

使用者指南

# Amazon Athena



# Amazon Athena: 使用者指南

Copyright © 2024 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、任從何貶低或使 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

什麼是 Amazon Athena ? .....	1
我應該在何時使用 Athena ? .....	1
Amazon Athena .....	1
Amazon EMR .....	2
Amazon Redshift .....	3
AWS 服務 與 Athena 的集成 .....	3
設定 .....	8
註冊一個 AWS 帳戶 .....	8
建立具有管理權限的使用者 .....	9
授與程式設計存取權 .....	10
連接 Athena 受管政策 .....	11
存取 Athena .....	12
使用 Athena SQL .....	13
了解資料表、資料庫和資料目錄 .....	14
開始使用 .....	16
必要條件 .....	16
步驟 1：建立資料庫 .....	17
步驟 2：建立資料表 .....	20
步驟 3：查詢資料 .....	25
儲存查詢 .....	28
鍵盤快速鍵和預先輸入的建議 .....	28
連接至其他資料來源 .....	29
連線至資料來源 .....	29
與整合 AWS Glue .....	30
使用 Hive 中繼存放區 .....	46
使用 Amazon Athena 聯合查詢 .....	80
用於存取資料目錄的 IAM 政策 .....	322
管理資料來源 .....	328
使用 DataZone .....	330
使用 ODBC 和 JDBC 驅動程式連接到 Amazon Athena .....	332
使用 JDBC 連接到 Athena .....	333
使用 ODBC 連接到 Athena .....	376
建立資料庫和資料表 .....	509
建立資料庫 .....	509

建立資料表 .....	513
資料表、資料庫和資料欄的名稱 .....	517
保留的關鍵字 .....	518
Amazon S3 中的資料表位置 .....	521
單欄式儲存格式 .....	523
轉換為單欄式格式 .....	524
分割資料 .....	525
分割區投影 .....	531
從查詢結果建立資料表 (CTAS) .....	552
CTAS 查詢的考量事項與限制 .....	553
在主控台中執行 CTAS 查詢 .....	555
分割和歸納 .....	557
CTAS 範例 .....	561
使用 CTAS 和 INSERT INTO 以進行 ETL .....	567
解決 100 分割區限制 .....	574
SerDe 參考 .....	579
使用 SerDe .....	579
支援的 SerDes 和資料格式 .....	581
執行查詢 .....	629
檢視查詢計劃 .....	630
查詢結果和近期查詢 .....	635
重複使用查詢結果 .....	650
檢視查詢統計資料 .....	654
使用檢視 .....	659
使用已儲存的查詢 .....	674
使用參數化查詢 .....	676
成本型最佳化工具 .....	684
查詢 S3 Express 單區域 .....	690
查詢 S3 Glacier .....	691
處理結構描述更新 .....	693
查詢陣列 .....	705
查詢地理空間資料 .....	730
查詢 JSON .....	754
搭配 Athena 使用機器學習 (ML) .....	766
使用 UDF 來查詢 .....	768
跨區域查詢 .....	779

查詢 AWS Glue Data Catalog .....	780
查詢 AWS 服務 記錄檔 .....	787
查詢 Web 伺服器日誌 .....	861
使用 ACID 交易 .....	872
查詢 Delta Lake 資料表 .....	872
查詢 Hudi 資料集 .....	877
使用 Iceberg 資料表 .....	886
安全 .....	907
資料保護 .....	908
身分與存取管理 .....	921
日誌記錄和監控 .....	985
合規驗證 .....	990
恢復能力 .....	991
基礎架構安全 .....	991
組態與漏洞分析 .....	994
搭配 Lake Formation 使用 Athena .....	995
工作負載管理 .....	1053
使用工作群組來控制查詢存取和成本 .....	1054
管理查詢處理容量 .....	1107
效能調校 .....	1122
壓縮支援 .....	1139
標記 資源 .....	1147
Service Quotas .....	1161
Athena 引擎版本控制 .....	1164
變更 Athena 引擎版本 .....	1165
Athena 引擎版本參考 .....	1169
Athena 的 SQL 參考 .....	1198
Athena 中的資料類型 .....	1198
DML 查詢、函數和運算子 .....	1206
DDL 陳述式 .....	1262
考量與限制 .....	1313
故障診斷 .....	1314
CREATE TABLE AS SELECT (CTAS) .....	1315
資料檔案問題 .....	1315
Linux Foundation Delta Lake 資料表 .....	1317
聯合查詢 .....	1317

JSON 相關錯誤 .....	1318
MSCK REPAIR TABLE .....	1319
輸出問題 .....	1320
Parquet 問題 .....	1320
分割問題 .....	1321
許可 .....	1323
查詢語法問題 .....	1325
查詢逾時問題 .....	1326
調節問題 .....	1327
檢視 .....	1327
工作群組 .....	1328
其他資源 .....	1328
Athena 錯誤目錄 .....	1328
程式碼範例 .....	1335
常數 .....	1336
建立用戶端以存取 Athena .....	1336
開始查詢執行 .....	1337
停止查詢執行 .....	1340
列出查詢執行 .....	1343
建立具名查詢 .....	1344
刪除具名查詢 .....	1345
列出具名查詢 .....	1347
使用 Apache Spark .....	1349
考量與限制 .....	1349
開始使用 .....	1350
在 Athena 中建立已啟用 Spark 的工作群組 .....	1350
開啟筆記本瀏覽器和切換工作群組 .....	1355
執行範例筆記本 .....	1355
編輯工作階段詳細資訊 .....	1356
檢視工作階段和計算詳細資訊 .....	1357
終止工作階段 .....	1358
建立您自己的筆記本 .....	1358
開啟先前建立的筆記本 .....	1360
使用筆記本 .....	1360
工作階段和計算 .....	1360
使用 Athena 筆記本編輯器 .....	1361

魔術 .....	1364
管理筆記本檔案 .....	1374
使用非 Hive 資料表格式 .....	1376
Python 程式庫支援 .....	1381
定義 .....	1381
生命週期管理 .....	1381
Python 程式庫 .....	1382
匯入檔案和程式庫 .....	1383
新增 JAR 檔案和自訂組態 .....	1396
使用 Athena 主控台 .....	1396
使用 AWS CLI 或 Athena API .....	1397
疑難排解 .....	1397
支援的資料和儲存格式 .....	1398
監控 Apache Spark 計算 .....	1399
Athena 中的 Apache Spark 計算的 CloudWatch 指標和維度清單 .....	1400
啟用請求者支付儲存貯體 .....	1400
1. 在 Amazon S3 儲存貯體上啟用請求者支付並新增儲存貯體政策 .....	1401
2. 建立 IAM 角色，並將其連接到 IAM 政策 .....	1402
3. 新增 Athena for Spark 工作階段屬性 .....	1403
啟用 Spark 加密 .....	1403
Athena 主控台 .....	1404
AWS CLI .....	1404
Athena API .....	1405
跨帳戶存取資料目錄 .....	1405
1. 在中 AWS Glue，提供對消費者角色的存取 .....	1406
2. 設定消費者帳戶，以用於存取 .....	1406
3. 設定工作階段並建立查詢 .....	1408
其他資源 .....	1409
Service Quotas .....	1409
Athena 筆記本 API .....	1410
已知問題 .....	1410
建立資料表時的非法引數例外狀況 .....	1411
在工作群組位置中建立的資料庫 .....	1412
預設資料庫中 Hive 管理資 AWS Glue 料表的問題 .....	1412
Athena for Spark 和 Athena SQL 之間的 CSV 和 JSON 檔案格式不相容 .....	1413
故障診斷 .....	1413

---

已啟用 Spark 的工作群組 .....	1414
使用 Spark EXPLAIN .....	1416
記錄應用程式事件 .....	1419
使用 CloudTrail 進行筆記本 API 呼叫 .....	1422
程式碼區塊大小限制 .....	1429
工作階段 .....	1431
資料表 .....	1432
取得支援 .....	1433
版本備註 .....	1435
2024 .....	1435
2024年6月26日 .....	1435
2024年4月26日 .....	1435
2024年4月24日 .....	1435
2024年4月16日 .....	1436
2024年4月10日 .....	1436
2024年4月8日 .....	1437
2024年3月15日 .....	1437
2024年2月15日 .....	1437
2024年1月31日 .....	1437
2023 .....	1438
2023年12月14日 .....	1438
2023年12月9日 .....	1438
2023年12月7日 .....	1439
2023年12月5日 .....	1439
2023年11月28日 .....	1439
2023年11月27日 .....	1439
2023年11月17日 .....	1440
2023年11月16日 .....	1441
2023年10月31日 .....	1441
2023年10月25日 .....	1442
2023年10月17日 .....	1442
2023年9月26日 .....	1442
2023年8月23日 .....	1442
2023年8月10日 .....	1442
2023年7月31日 .....	1443
2023年7月27日 .....	1443

---

2023 年 7 月 24 日 .....	1443
2023 年 7 月 20 日 .....	1444
2023 年 7 月 13 日 .....	1444
2023 年 7 月 3 日 .....	1445
2023 年 6 月 30 日 .....	1445
2023 年 6 月 29 日 .....	1445
2023 年 6 月 28 日 .....	1446
2023 年 6 月 12 日 .....	1446
2023 年 6 月 8 日 .....	1446
2023 年 6 月 2 日 .....	1447
2023 年 5 月 25 日 .....	1447
2023 年 5 月 18 日 .....	1448
2023 年 5 月 15 日 .....	1449
2023 年 5 月 10 日 .....	1449
2023 年 5 月 8 日 .....	1449
2023 年 4 月 28 日 .....	1451
2023 年 4 月 17 日 .....	1451
2023 年 4 月 14 日 .....	1451
2023 年 4 月 4 日 .....	1452
2023 年 3 月 30 日 .....	1452
2023 年 3 月 28 日 .....	1452
2023 年 3 月 27 日 .....	1453
2023 年 3 月 17 日 .....	1453
2023 年 3 月 8 日 .....	1454
2023 年 2 月 15 日 .....	1454
2023 年 1 月 31 日 .....	1454
2023 年 1 月 20 日 .....	1455
2023 年 1 月 3 日 .....	1455
2022 .....	1455
2022 年 12 月 14 日 .....	1455
2022 年 12 月 2 日 .....	1456
2022 年 11 月 30 日 .....	1456
2022 年 11 月 18 日 .....	1457
2022 年 11 月 17 日 .....	1457
2022 年 11 月 14 日 .....	1457
2022 年 11 月 11 日 .....	1458

2022 年 11 月 8 日 .....	1459
2022 年 10 月 13 日 .....	1459
2022 年 10 月 10 日 .....	1459
2022 年 9 月 23 日 .....	1460
2022 年 9 月 13 日 .....	1460
2022 年 8 月 31 日 .....	1460
2022 年 8 月 23 日 .....	1461
2022 年 8 月 3 日 .....	1461
2022 年 8 月 1 日 .....	1461
2022 年 7 月 21 日 .....	1462
2022 年 7 月 11 日 .....	1462
2022 年 7 月 8 日 .....	1463
2022 年 6 月 6 日 .....	1463
2022 年 5 月 25 日 .....	1463
2022 年 5 月 6 日 .....	1464
2022 年 4 月 22 日 .....	1464
2022 年 4 月 21 日 .....	1464
2022 年 4 月 13 日 .....	1465
2022 年 3 月 30 日 .....	1466
2022 年 3 月 18 日 .....	1466
2022 年 3 月 2 日 .....	1466
2022 年 2 月 23 日 .....	1467
2022 年 2 月 15 日 .....	1467
2022 年 2 月 14 日 .....	1468
2022 年 2 月 9 日 .....	1468
2022 年 2 月 8 日 .....	1468
2022 年 1 月 28 日 .....	1468
2022 年 1 月 13 日 .....	1469
2021 .....	1469
2021 年 11 月 26 日 .....	1469
2021 年 11 月 24 日 .....	1470
2021 年 11 月 22 日 .....	1470
2021 年 11 月 18 日 .....	1470
2021 年 11 月 17 日 .....	1471
2021 年 11 月 16 日 .....	1471
2021 年 11 月 12 日 .....	1472

2021 年 11 月 2 日 .....	1472
2021 年 10 月 29 日 .....	1472
2021 年 10 月 4 日 .....	1473
2021 年 9 月 16 日 .....	1474
2021 年 9 月 15 日 .....	1474
2021 年 8 月 31 日 .....	1475
2021 年 8 月 12 日 .....	1475
2021 年 8 月 6 日 .....	1476
2021 年 8 月 5 日 .....	1476
2021 年 7 月 30 日 .....	1476
2021 年 7 月 21 日 .....	1477
2021 年 7 月 16 日 .....	1477
2021 年 7 月 8 日 .....	1477
2021 年 7 月 1 日 .....	1478
2021 年 6 月 23 日 .....	1478
2021 年 5 月 12 日 .....	1478
2021 年 5 月 10 日 .....	1478
2021 年 5 月 5 日 .....	1479
2021 年 4 月 30 日 .....	1479
2021 年 4 月 29 日 .....	1479
2021 年 4 月 26 日 .....	1480
2021 年 4 月 21 日 .....	1480
2021 年 4 月 5 日 .....	1480
2021 年 3 月 30 日 .....	1481
2021 年 3 月 25 日 .....	1481
2021 年 3 月 5 日 .....	1481
2021 年 2 月 25 日 .....	1481
2020 .....	1481
2020 年 12 月 16 日 .....	1481
2020 年 11 月 24 日 .....	1482
2020 年 11 月 11 日 .....	1482
2020 年 10 月 22 日 .....	1484
2020 年 7 月 29 日 .....	1484
2020 年 7 月 9 日 .....	1484
2020 年 6 月 1 日 .....	1485
2020 年 5 月 21 日 .....	1485

---

2020 年 4 月 1 日 .....	1485
2020 年 3 月 11 日 .....	1486
2020 年 3 月 6 日 .....	1486
2019 .....	1486
2019 年 11 月 26 日 .....	1486
2019 年 11 月 12 日 .....	1489
2019 年 11 月 8 日 .....	1490
2019 年 10 月 8 日 .....	1490
2019 年 9 月 19 日 .....	1490
2019 年 9 月 12 日 .....	1490
2019 年 8 月 16 日 .....	1491
2019 年 8 月 9 日 .....	1491
2019 年 6 月 26 日 .....	1491
2019 年 5 月 24 日 .....	1491
2019 年 3 月 5 日 .....	1492
2019 年 2 月 22 日 .....	1492
2019 年 2 月 18 日 .....	1493
2018 .....	1495
2018 年 11 月 20 日 .....	1495
2018 年 10 月 15 日 .....	1495
2018 年 10 月 10 日 .....	1496
2018 年 9 月 6 日 .....	1496
2018 年 8 月 23 日 .....	1497
2018 年 8 月 16 日 .....	1497
2018 年 8 月 7 日 .....	1498
2018 年 6 月 5 日 .....	1498
2018 年 5 月 17 日 .....	1499
2018 年 4 月 19 日 .....	1500
2018 年 4 月 6 日 .....	1500
2018 年 3 月 15 日 .....	1500
2018 年 2 月 2 日 .....	1500
2018 年 1 月 19 日 .....	1500
2017 .....	1502
2017 年 11 月 13 日 .....	1502
2017 年 11 月 1 日 .....	1502
2017 年 10 月 19 日 .....	1502

---

2017 年 10 月 3 日 .....	1502
2017 年 9 月 25 日 .....	1502
2017 年 8 月 14 日 .....	1502
2017 年 8 月 4 日 .....	1503
2017 年 6 月 22 日 .....	1503
2017 年 6 月 8 日 .....	1503
2017 年 5 月 19 日 .....	1503
2017 年 4 月 4 日 .....	1504
2017 年 3 月 24 日 .....	1505
2017 年 2 月 20 日 .....	1506
文件歷史紀錄 .....	1509
AWS 詞彙表 .....	1524
.....	mdxxv

# 什麼是 Amazon Athena ?

Amazon Athena 是一種互動式查詢服務，可在 Amazon Simple Storage Service (Amazon S3) 中使用標準 [SQL](#) 輕鬆地直接分析資料。只要在中執行一些動作 AWS Management Console，您就可以將 Athena 指向存放在 Amazon S3 中的資料，然後開始使用標準 SQL 執行隨機操作查詢並在幾秒鐘內取得結果。

如需詳細資訊，請參閱 [開始使用](#)。

Amazon Athena 還可讓您輕鬆使用 Apache Spark 以互動方式執行資料分析，而不必規劃、設定或管理資源。當您在 Athena 執行 Apache Spark 應用程式時，可提交 Spark 程式碼進行處理，並直接接收結果。在 Amazon Athena 主控台中使用筆記本的簡化體驗，以使用 Python 或 [Athena 筆記本 API](#) 開發 Apache Spark 應用程式。

如需詳細資訊，請參閱 [開始在 Amazon Athena 上使用 Apache Spark](#)。

Amazon Athena 上的 Athena SQL 和 Apache Spark 為無伺服器，所以不需設定和管理基礎設施，而且您只需支付執行的查詢費用。Athena 會自動調整規模 (平行執行查詢)，因此即使在大型資料集和複雜的查詢中，也能快速獲得結果。

## 主題

- [我應該在何時使用 Athena ?](#)
- [AWS 服務 與 Athena 的集成](#)
- [設定](#)
- [存取 Athena](#)

## 我應該在何時使用 Athena ?

Amazon Athena 之類的查詢服務、Amazon Redshift 之類的資料倉儲以及 Amazon EMR 之類的複雜資料處理架構都可以滿足不同的需要和使用案例。以下指導可協助您根據您的要求選擇一項或多項服務。

## Amazon Athena

Athena 可協助您分析在 Amazon S3 中存放的非結構化、半結構化和結構化資料。範例包括 CSV、JSON 或單欄資料格式，例如 Apache Parquet 和 Apache ORC。您可以使用 Athena 以透過 ANSI SQL 執行隨機操作查詢，而不需要將資料彙總或載入到 Athena。

Athena 與 Amazon QuickSight 整合，可輕鬆實現資料視覺。您可以使用 Athena 產生報告，或搭配商業智慧工具或透過 JDBC 或 ODBC 驅動程式連接的 SQL 用戶端來探索資料。有關更多信息，請參閱 [Amazon 用 QuickSight 戶指南 QuickSight 中的 Amazon 是什麼使用 ODBC 和 JDBC 驅動程式連接到 Amazon Athena。](#)

Athena 與整合 AWS Glue Data Catalog，為您在 Amazon S3 中的資料提供永久的中繼資料存放區。這可讓您根據整個 Amazon Web Services 帳戶提供的中央中繼資料存放區，並與的 ETL 和資料探索功能整合，在 Athena 中建立表格和查詢資料。AWS Glue 如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的 [與 AWS Glue 整合](#) 以及 [什麼是 AWS Glue](#)。

Amazon Athena 讓您能直接在 Simple Storage Service (Amazon S3) 中輕鬆執行對資料的互動式查詢，而不需要格式化資料或管理基礎設施。例如，如果您想對 Web 日誌執行快速查詢以解決網站上的效能問題，Athena 非常有用。使用 Athena，您可以快速開始：只需要為資料定義一個資料表，即可使用標準 SQL 進行查詢開始進行查詢。

如果您想要對在 Simple Storage Service (Amazon S3) 上的資料執行互動式隨意 SQL 查詢，而不需要管理任何基礎設施或叢集，則應使用 Amazon Athena。Amazon Athena 為對在 Simple Storage Service (Amazon S3) 中的資料執行隨意查詢提供了最簡單的方法，而不需要設置或管理任何伺服器。

如需 Athena 運用或整合的 AWS 服務清單，請參閱 [the section called “AWS 服務與 Athena 的集成”](#)。

## Amazon EMR

與內部部署的部署方式相比，Amazon EMR 讓執行諸如 Hadoop、Spark 和 Presto 等高分散式處理架構變得簡單又符合成本效益。Amazon EMR 非常靈活 – 您可以執行自訂的應用程式和代碼，並定義特定的運算、記憶體、儲存和應用程式參數，以優化您的分析需求。

除了執行 SQL 查詢之外，Amazon EMR 還可以為應用程式執行各種水平擴展資料處理任務，例如機器學習、圖形分析、資料轉換、串流資料以及幾乎任何您可以編寫代碼的內容。如果您使用自訂的代碼搭配諸如 Spark、Hadoop、Presto 或 Hbase 等最新的大數據處理架構來處理和分析超大資料集，則應使用 Amazon EMR。Amazon EMR 帶給您對叢集及安裝在叢集上之軟體的組態的完整控制。

您可以使用 Amazon Athena 來查詢您用 Amazon EMR 處理的資料。Amazon Athena 支援許多與 Amazon EMR 相同的資料格式。Athena 的資料目錄與 Hive 中繼存放區相容。如果您使用 EMR 並且已經擁有 Hive 中繼存放區，即能在 Amazon Athena 上執行 DDL 陳述式並立即查詢您的資料，而不會影響您的 Amazon EMR 任務。

## Amazon Redshift

當您需要將來自許多不同來源的資料 ( 例如庫存系統、財務系統和零售銷售系統 ) 彙集成一個通用格式並長時間存放時，像 Amazon Redshift 這樣的資料倉儲是您的最佳選擇。如果您想根據歷史資料建置複雜的業務報告，那麼像 Amazon Redshift 這樣的資料倉儲是最佳選擇。Amazon Redshift 中的查詢引擎經過優化，在執行聯結大量非常龐大的資料庫資料表的複雜查詢時，表現尤佳。如果您需要對高度結構化且具有許多大量超大資料表聯結的資料執行查詢時，請選擇 Amazon Redshift。

如需有關何時使用 Athena 的詳細資訊，請參閱下列資源：

- [資源中心入門中的 AWS 上的分析服務決策指南](#)
- Amazon Athena 常見問答集中的 [何時使用 Athena 與其他大數據服務](#)
- [Amazon Athena 概觀](#)
- [Amazon Athena 功能](#)
- [Amazon Athena 常見問答集](#)
- [Amazon Athena 部落格帖文](#)

## AWS 服務 與 Athena 的集成

您可以使用 Athena 查詢本節中 AWS 服務 列出的資料。若要查看每項服務支援的區域，請參閱 Amazon Web Services 一般參考 中的 [區域與端點](#)。

### AWS 服務 與 Athena 整合

- [AWS CloudFormation](#)
- [Amazon CloudFront](#)
- [AWS CloudTrail](#)
- [Amazon DataZone](#)
- [Elastic Load Balancing](#)
- [Amazon EMR Studio](#)
- [AWS Glue Data Catalog](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [Amazon QuickSight](#)

- [Amazon S3 清查](#)
- [AWS Step Functions](#)
- [AWS Systems Manager 清查](#)
- [Amazon Virtual Private Cloud](#)

如需每項整合的資訊，請參閱下列各節。

## AWS CloudFormation

### 容量保留

參考主題：[AWS: Athena:: CapacityReservation](#) 在AWS CloudFormation 使用者指南中

使用提供的名稱和請求的資料處理單元數量來指定容量保留。如需詳細資訊，請參閱[管理查詢處理容量](#)亞 Amazon Athena 使用者指南和[CreateCapacityReservation](#)亞 Amazon Athena API 參考中的。

### 資料型錄

參考主題：[AWS: Athena:: DataCatalog](#) 在AWS CloudFormation 使用者指南中

指定 Athena 資料目錄，包括名稱、描述、類型、參數和標籤。如需詳細資訊，請參閱[了解資料表、資料庫和資料目錄](#)亞 Amazon Athena 使用者指南和[CreateDataCatalog](#)亞 Amazon Athena API 參考中的。

### 具名查詢

參考主題：[AWS: Athena:: NamedQuery](#) 在AWS CloudFormation 使用者指南中

使用指定具名的查詢，AWS CloudFormation 然後在 Athena 中執行。具名查詢允許您將查詢名稱映射到查詢，然後將其作為 Athena 主控台中已儲存的查詢來執行。如需詳細資訊，請參閱[使用已儲存的查詢](#)亞 Amazon Athena 使用者指南和[CreateNamedQuery](#)亞 Amazon Athena API 參考中的。

### 預備陳述式

參考主題：[AWS: Athena:: PreparedStatement](#) 在AWS CloudFormation 使用者指南中

指定已備妥的陳述式，以便與 Athena 中的 SQL 查詢搭配使用。預備陳述式包含參數預留位置，其值在執行時提供。如需詳細資訊，請參閱[使用參數化查詢](#)亞 Amazon Athena 使用者指南和[CreatePreparedStatement](#)亞 Amazon Athena API 參考中的。

## 工作群組

參考主題：[AWS: Athena: WorkGroup](#) 在AWS CloudFormation 使用者指南中

使用指定 Athena 工作群組 AWS CloudFormation。使用 Athena 工作群組將您或您群組的查詢與相同帳戶中的其他查詢隔離。如需詳細資訊，請參閱[使用工作群組來控制查詢存取和成本](#)亞 Amazon Athena 使用者指南和[CreateWorkGroup](#)亞 Amazon Athena API 參考中的。

## Amazon CloudFront

參考主題：[查詢 Amazon CloudFront 日誌](#)

使用 Athena 查詢 Amazon CloudFront 日誌。如需使用的詳細資訊 CloudFront，請參閱[Amazon CloudFront 開發人員指南](#)。

## AWS CloudTrail

參考主題：[查詢 AWS CloudTrail 記錄檔](#)

搭配 CloudTrail 日誌使用 Athena 是強化 AWS 服務活動分析的強大方法。例如，您可以使用查詢來識別趨勢，並依屬性 (例如來源 IP 地址或使用者) 進一步隔離活動。您可以直接從 CloudTrail 主控台建立用於查詢記錄的資料表，並使用這些資料表在 Athena 中執行查詢。如需詳細資訊，請參閱[使用主 CloudTrail 控制台為 CloudTrail 記錄建立 Athena 資料表](#)。

## Amazon DataZone

參考主題：[在 Athena 中使用 Amazon DataZone](#)

使用 [Amazon DataZone](#) 跨組織界限大規模共用、搜尋和探索資料。DataZone 簡化您跨 AWS 分析服務的體驗，例如 Athena AWS Glue、和 AWS Lake Formation。如果您在不同的資料來源中有大量資料，則可以使用 Amazon 建立 DataZone 以商業使用案例為基礎的人員、資料和工具群組。

在 Athena 中，您可以使用查詢編輯器來存取和查詢 DataZone 環境。如需詳細資訊，請參閱[在 Athena 中使用 Amazon DataZone](#)。

## Elastic Load Balancing

參考主題：[查詢 Application Load Balancer 日誌](#)

查詢 Application Load Balancer 日誌可讓您查看流量來源、延遲，以及往返於 Elastic Load Balancing 執行個體與後端應用程式傳輸的位元組。如需詳細資訊，請參閱[查詢 Application Load Balancer 日誌](#)。

參考主題：[查詢 Classic Load Balancer 日誌](#)

查詢 Classic Load Balancer 日誌來分析和了解往返於 Elastic Load Balancing 執行個體與後端應用程式的流量模式。您可以查看流量來源、延遲和傳輸的位元組。如需詳細資訊，請參閱[建立 ELB 日誌的資料表](#)。

## Amazon EMR Studio

參考主題：[在 EMR Studio 中使用 Amazon Athena SQL 編輯器](#)

您可以在 EMR Studio 中使用 Athena 來開發和執行互動式查詢。這讓您可以透過與 Spark、Scala 和其他工作負載相同的 Amazon EMR 介面，在 Athena 上使用 EMR Studio 進行 SQL 分析。您可以使用 EMR Studio 中的 Athena 整合進行下列任務：

- 進行 Athena SQL 查詢
- 檢視查詢結果
- 檢視查詢歷史記錄
- 檢視儲存的查詢
- 進行參數化查詢
- 檢視資料目錄的資料庫、資料表和檢視

下列 Athena 功能不適用於 Amazon EMR Studio：

- 建立或更新 Athena 工作群組、資料來源或容量保留等管理員功能
- Athena for Spark 或 Spark 筆記本
- DataZone 整合
- Step Functions

EMR 工作室與 Athena 的整合可在所有提供 EMR 工作室和 Athena 的 AWS 區域 地方使用。如需有關在 EMR Studio 中使用 Athena 的詳細資訊，請參閱《Amazon EMR 管理指南》中的在[EMR Studio 中使用 Amazon Athena SQL 編輯器](#)。

## AWS Glue Data Catalog

參考主題：[與整合 AWS Glue](#)

Athena 與整合 AWS Glue Data Catalog，為您在 Amazon S3 中的資料提供永久的中繼資料存放區。這可讓您根據整個 Amazon Web Services 帳戶提供的中央中繼資料存放區，並與的 ETL 和資料探索功能整合，在 Athena 中建立表格和查詢資料。AWS Glue 如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[與整合 AWS Glue](#) 和 [什麼是 AWS Glue](#)。

## AWS Identity and Access Management (IAM)

參考主題：[Amazon Athena 的動作](#)

您可以在 IAM 許可政策中使用 Athena API 動作。如需詳細資訊，請參閱 [Amazon Athena 的動作](#) 以及 [Athena 中的 Identity and Access Management](#)。

## Amazon QuickSight

參考主題：[使用 ODBC 和 JDBC 驅動程式連接到 Amazon Athena](#)

Athena 與 Amazon QuickSight 整合，可輕鬆實現資料視覺。您可以使用 Athena 產生報告，或搭配商業智慧工具或透過 JDBC 或 ODBC 驅動程式連接的 SQL 用戶端來探索資料。有關 Amazon 的更多信息 QuickSight，請參閱 [Amazon 用 QuickSight 戶指南 QuickSight 中的 Amazon 是什麼](#)。如需有關搭配 Athena 使用 JDBC 和 ODBC 驅動程式的資訊，請參閱 [使用 ODBC 和 JDBC 驅動程式連接至 Amazon Athena](#)。

## Amazon S3 清查

參考主題：《Amazon Simple Storage Service 使用者指南》中的 [使用 Athena 查詢清查](#)

您可以使用 Amazon Athena 來以標準 SQL 查詢 Amazon S3 清查。您可以使用 Amazon S3 清查來稽核與回報物件的複寫和加密狀態，以滿足業務、合規及法規需求。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [Amazon S3 清查](#)。

## AWS Step Functions

參考主題：《AWS Step Functions 開發人員指南》中的 [使用 Step Functions 呼叫 Athena](#)

打電話給 Athena AWS Step Functions。AWS Step Functions 可以使用 [Amazon 州語言 AWS 服務](#) 直接控制選擇。您可以搭配 Athena 使用 Step Functions 來啟動和停止查詢執行、取得查詢結果、執行隨機操作或排程的資料查詢，以及從 Simple Storage Service (Amazon S3) 中的資料湖擷取結果。Step Functions 角色必須擁有使用 Athena 的許可。如需詳細資訊，請參閱 [《AWS Step Functions 開發人員指南》](#)。

影片：使用編排 Amazon Athena 查詢 AWS Step Functions

以下影片示範如何使用 Amazon Athena，以及如 AWS Step Functions 何執行定期排程的 Athena 查詢並產生對應的報告。

[使用編排 Amazon Athena 查詢 AWS Step Functions](#)

如需使用 Step Functions 和 Amazon EventBridge 協調 AWS Glue DataBrew 雅典娜和亞馬遜的範例 QuickSight，請參閱 AWS 大數據部落格 AWS Step Functions 中的 [協調 AWS Glue DataBrew 任務和 Amazon Athena 查詢](#)。

## AWS Systems Manager 庫存

參考主題：《AWS Systems Manager 使用者指南》中的[查詢來自多個區域和帳戶的清查資料](#)

AWS Systems Manager 庫存與 Amazon Athena 整合，協助您查詢來自多個帳戶 AWS 區域和帳戶的庫存資料。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》<https://docs.aws.amazon.com/systems-manager/latest/userguide/>。

## Amazon Virtual Private Cloud

參考主題：[查詢 Amazon VPC 流程日誌](#)

Amazon Virtual Private Cloud 流量日誌擷取傳入和傳出 VPC 中網路界面之 IP 流量的相關資訊。在 Athena 中查詢日誌來調查網路流量模式，並識別整個 Amazon VPC 網路所遭受的威脅和風險。如需 Amazon VPC 的詳細資訊，請參閱《Amazon VPC 使用者指南》<https://docs.aws.amazon.com/vpc/latest/userguide/>。

# 設定

如果您已註冊 Amazon Web Services，便可以立即開始使用 Amazon Athena。如果您尚未註冊 AWS 或需要開始使用的協助，請務必完成下列工作。

## 註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

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

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊時 AWS 帳戶，會建立 AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 [root 使用者來執行需要 root 使用者存取權](#)的工作。

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

## 建立具有管理權限的使用者

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

### 保護您的 AWS 帳戶根使用者

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

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的[以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

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

### 建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM 身分中心中，將管理存取權授予使用者。

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

### 以具有管理權限的使用者身分登入

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

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

### 指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[建立權限集](#)」。

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

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

## 授與程式設計存取權

如果使用者想要與 AWS 之外的 AWS Management Console 授與程式設計存取權的方式取決於正在存取的使用者類型。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	By
人力身分 (IAM Identity Center 中管理的使用者)	使用臨時登入資料來簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> <li>如需詳細資訊 AWS CLI，請參閱 <a href="#">《使 AWS CLI 用 AWS Command Line Interface 者指南》</a> AWS IAM Identity Center 中的〈配置使用〉。</li> <li>如需 AWS SDK、工具和 AWS API，請參閱 AWS SDK 和工具參考指南中的 <a href="#">IAM 身分中心身分驗證</a>。</li> </ul>
IAM	使用臨時登入資料來簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	遵循 <a href="#">《IAM 使用者指南》</a> 中的〈 <a href="#">將臨時登入資料搭配 AWS 資源使用</a> 〉中的指示
IAM	(不建議使用) 使用長期認證簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> <li>如需相關資訊 AWS CLI，請參閱使用指南中的 <a href="#">使用 IAM 使用者登入資料進行驗證</a>。AWS Command Line Interface</li> </ul>

哪個使用者需要程式設計存取權？	到	By
		<ul style="list-style-type: none"><li>對於 AWS SDK 和工具，請參閱 AWS SDK 和工具參考指南中的<a href="#">使用長期憑據進行身份驗證</a>。</li><li>如需 AWS API，請參閱 IAM 使用者指南中的<a href="#">管理 IAM 使用者的存取金鑰</a>。</li></ul>

## 連接 Athena 受管政策

Athena 受管政策授予使用 Athena 功能的許可。您可以將這些受管政策連接到一或多個 IAM 角色，而使用者可以擔任這些角色以使用 Athena。

[IAM 角色](#)是您可以在帳戶中建立的另一種 IAM 身分，具有特定的許可。IAM 角色與 IAM 使用者類似，因為它是具有許可政策的 AWS 身分識別，可決定身分可以執行和不能在其中執行的操作 AWS。但是，角色的目的是讓需要它的任何人可代入，而不是單獨地與某個人員關聯。此外，角色沒有與之關聯的標準長期憑證，例如密碼或存取金鑰。反之，當您擔任角色時，其會為您的角色工作階段提供臨時安全性憑證。

如需有關角色的詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 角色](#)和[建立 IAM 角色](#)。

若要建立可授予 Athena 存取權的角色，請將 Athena 受管政策連接至該角色。有兩個適用於 Athena 的受管政策：AmazonAthenaFullAccess 和 AWSQuicksightAthenaAccess。這些政策會授權 Athena 查詢 Amazon S3，並代表您將查詢結果寫入單獨的儲存貯體。若要查看 Athena 的這些政策內容，請參閱 [AWS Amazon Athena 的受管政策](#)。

如需將 Athena 受管政策連接至角色的相關步驟，請依照《IAM 使用者指南》中的[新增 IAM 身分許可 \(主控台\)](#) 操作，然後新增 AmazonAthenaFullAccess 和 AWSQuicksightAthenaAccess 受管政策至您建立的角色。

### Note

您可能需要額外的許可，才能在 Amazon S3 中存取基礎資料集。如果您不是帳戶擁有者，或存取儲存貯體時受限，請聯絡儲存貯體擁有者使用以資源為基礎的政策來授與存取權，或聯絡帳戶管理員使用角色型政策來授與存取權。如需詳細資訊，請參閱 [從 Athena 訪問 Amazon](#)

**S3**。如果資料集或 Athena 查詢結果已加密，您可能需要額外的許可。如需更多詳細資訊，請參閱 [靜態加密](#)。

## 存取 Athena

您可以使用 AWS Management Console JDBC 或 ODBC 連線、Athena API、Athena CLI、AWS SDK 或 AWS Tools for Windows PowerShell

- 若要開始搭配使用 Athena SQL 和主控台，請參閱 [開始使用](#)。
- 若要開始建立與 Jupyter 相容的筆記本和 Apache Spark 應用程式使用 Python，請參閱 [在 Amazon Athena 使用 Apache Spark](#)
- 若要了解如何使用 JDBC 或 ODBC 驅動程式，請參閱 [使用 JDBC 連接到 Amazon Athena](#) 和 [使用 ODBC 連接到 Amazon Athena](#)。
- 若要使用 Athena API，請參閱 [《Amazon Athena API 參考》](#)。
- 若要使用 CLI，請安裝 [AWS CLI](#)，然後從命令列輸入 `aws athena help` 來查看可用的命令。如需有關可用命令的資訊，請參閱 [《Amazon Athena 命令列參考》](#)。
- 若要使用 AWS SDK for Java 2.x，請參閱 [AWS SDK for Java 2.x API 參考](#) 的 Athena 部分、GitHub .com 上的 [Athena Java V2 範例](#)，以及 [AWS SDK for Java 2.x 開發人員指南](#)。
- 若要使用 AWS SDK for .NET，請參閱 [AWS SDK for .NET API 參考](#) 中的 Amazon.Athena 命名空間、GitHub .com 上的 [.NET Athena 範例](#)，以及 [AWS SDK for .NET 開發人員指南](#)。
- 若要使用 AWS Tools for Windows PowerShell，請參閱 [AWS Tools for PowerShell -Amazon Athena](#) 指令程式參考資料、入 [AWS Tools for PowerShell](#) 口網站頁面和使用 [AWS Tools for Windows PowerShell 者指南](#)。
- 如需有關您可以透過程式設計方式連接的 Athena 服務端點的資訊，請參閱 [《Amazon Web Services 一般參考》](#) 中的 [Amazon Athena 端點和配額](#)。

# 使用 Athena SQL

您可以透過 Athena SQL 在 Amazon S3 中使用 [AWS Glue Data Catalog \(外部 Hive 中繼存放區\)](#) 就地查詢資料，或使用各種[預先建置的連接器](#)連接到其他資料來源進行[聯合查詢](#)。

您也可以：

- 使用 [Athena 的 JDBC 和 ODBC 驅動程式](#)連接到商業智慧工具和其他應用程式。
- 查詢 [AWS 服務日誌](#)。
- 查詢 [Apache Iceberg 資料表](#) (包括時間歷程查詢) 和 [Apache Hudi 資料集](#)。
- 查詢[地理空間資料](#)。
- 使用 Amazon SageMaker 提供的[機器學習推論](#)進行查詢。
- 使用您自己的[使用者定義的函數](#)查詢。
- 使用[分割區投影](#)可加速高度分割資料表的查詢處理，並自動化分割區管理。

## 主題

- [了解資料表、資料庫和資料目錄](#)
- [開始使用](#)
- [連線至資料來源](#)
- [使用 ODBC 和 JDBC 驅動程式連接到 Amazon Athena](#)
- [建立資料庫和資料表](#)
- [從查詢結果建立資料表 \(CTAS\)](#)
- [SerDe 參考](#)
- [使用 Amazon Athena 執行 SQL 查詢](#)
- [使用 Athena ACID 交易](#)
- [Amazon Athena 安全性](#)
- [工作負載管理](#)
- [Athena 引擎版本控制](#)
- [Athena 的 SQL 參考](#)
- [Athena 中的疑難排解](#)
- [程式碼範例](#)

## 了解資料表、資料庫和資料目錄

在 Athena 中，目錄、資料庫和資料表是中繼資料定義的容器，用於定義基礎來源資料的結構描述。

Athena 使用下列術語來表示資料物件的階層：

- 資料來源 – 一組資料庫
- 資料表 – 一組資料表
- 表格 – 組織為一組資料列或資料欄的資料

有時候，這些物件也會指稱為替代但對等的名稱，如下所示：

- 資料來源有時被稱為目錄。
- 資料庫有時被稱為結構描述。

### Note

該術語可能會因您搭配 Athena 使用的聯合資料來源而異。如需詳細資訊，請參閱 [Athena 和聯合資料表名稱限定詞](#)。

下列 Athena 主控台內的範例查詢會使用 `awsdatacatalog` 資料來源、`default` 資料庫和 `some_table` 資料表。

The screenshot displays the Amazon Athena Query Editor interface. At the top, there are tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings', along with a 'Workgroup' dropdown set to 'primary'. The main area is divided into a left sidebar and a right main panel. The sidebar contains a 'Data' section with a refresh icon and a 'Data source' dropdown set to 'AwsDataCatalog'. Below it is a 'Database' dropdown set to 'default'. Further down is a 'Tables and views' section with a 'Create' button and a search bar labeled 'Filter tables and views'. Underneath, there are two sections: 'Tables (2)' and 'Views (0)'. The 'Tables (2)' section lists 'some\_table' and 'some\_table2', with 'some\_table' highlighted in red. The main panel shows a query editor with a SQL query: `SELECT * FROM "awsdatacatalog"."default"."some_table" limit 10;`. The query is highlighted in red. Below the query editor, there are buttons for 'Run', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Run' button is highlighted in orange. To the right of the buttons, there is a toggle for 'Reuse query results' and a note '\*Athena engine version 3 only'. Below the buttons, there is a 'Query results' section showing 'Completed' status, 'Time in queue: 240 ms', 'Run time: 6.535 sec', and 'Data scanned: 0.91 KB'. Below this, there is a 'Results (5)' section with a search bar and a table of results. The table has columns '#', 'id', 'data', and 'category'. The results are: Row 1: id=1, data=a, category=A; Row 2: id=3, data=d, category=d1; Row 3: id=4, data=e, category=e1; Row 4: id=4, data=f, category=f1; Row 5: id=2, data=b, category=b1.

對於每個資料集，Athena 中必須存在資料表。資料表中的中繼資料可讓 Athena 知道資料位於 Amazon S3 中的何處，並指定資料的結構，例如資料欄名稱、資料類型和資料表名稱。資料庫是資料表的邏輯群組，也只保存資料集的中繼資料和結構描述資訊。

對於您想要查詢的每個資料集，Athena 必須有基礎資料表，以用於取得和傳回查詢結果。因此，在查詢資料之前，必須在 Athena 中註冊資料表。註冊是在您自動或手動建立資料表時進行。

您可以使用 AWS Glue 爬行者程式自動建立資料表。如需有關 AWS Glue 和爬行者程式的詳細資訊，請參閱與 [AWS Glue 整合](#)。當 AWS Glue 創建一個表，它在自己的 AWS Glue 數據目錄註冊它。Athena 使用 AWS Glue 資料目錄來存放和擷取此中繼資料，供您執行查詢以分析基礎資料集時使用。

無論資料表的建立方式為何，資料表建立程序都會向 Athena 註冊資料集。此註冊會發生在中 AWS Glue Data Catalog，可讓 Athena 對資料執行查詢。在 Athena 查詢編輯器中，此目錄 (或資料來源) 會指稱為標籤 `AwsDataCatalog`。

在建立資料表之後，您可以使用 [SQL SELECT](#) 陳述式來查詢該資料表，包括取得 [來源資料的特定檔案位置](#)。您的查詢結果會存放在 Amazon S3 中 [您指定的查詢結果位置](#)。

您可以在整個 Amazon Web Services 帳戶中存取 AWS Glue 資料目錄。其他人 AWS 服務 可以共用 AWS Glue 資料目錄，因此您可以查看使用 Athena 在整個組織中建立的資料庫和表格，反之亦然。

- 手動建立資料表：
  - 使用 Athena 主控台執行建立資料表精靈。
  - 使用 Athena 主控台在查詢編輯器中撰寫 Hive DDL 陳述式。
  - 使用 Athena API 或 CLI，以執行採用 DDL 陳述式的 SQL 查詢字串。
  - 使用 Athena JDBC 或 ODBC 驅動程式。

手動建立資料表和資料庫時，Athena 會在幕後使用 HiveQL 資料定義語言 (DDL) 陳述式 (例如 CREATE TABLE、CREATE DATABASE 和 DROP TABLE)，在 AWS Glue Data Catalog 中建立資料表和資料庫。

若要開始使用，您可以使用 Athena 主控台中的教學課程，或參閱 Athena 說明文件中的 step-by-step 指南。

- 若要使用 Athena 主控台中的教學課程，請選擇主控台右上角的資訊圖示，然後選擇教學課程索引標籤。
- 如需在 Athena 查詢編輯器中建立資料表和撰寫查詢的 step-by-step 教學課程，請參閱 [開始使用](#)。

## 開始使用

此教學課程會逐步解說如何使用 Amazon Athena 來查詢資料。您將會根據在 Amazon Simple Storage Service 中存放的範例資料來建立資料表、查詢資料表，並檢查查詢結果。

此教學課程中使用線上資源，因此將向您就執行的查詢收費。您不需為本教學課程所使用的位置中的範例資料付費，但如果您上傳自己的資料檔案至 Amazon S3，則需付費。

## 必要條件

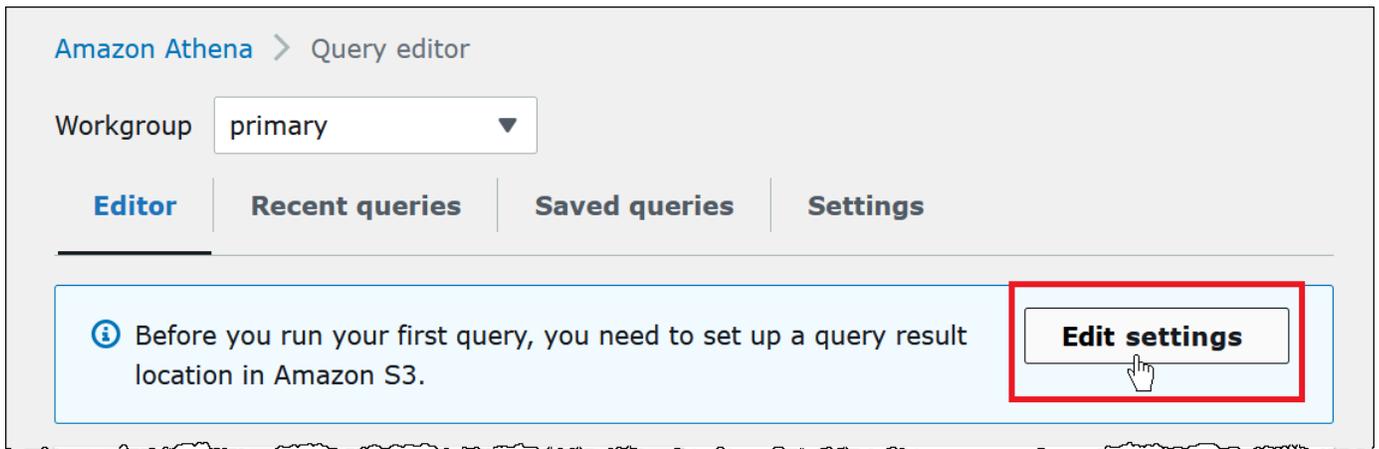
- 若您尚未註冊，[請先註冊 AWS 帳戶](#)。
- 使用您用於 Athena 的相同 AWS 區域 帳戶 (例如美國西部 (奧勒岡)) 和帳戶，請依照下列步驟在 [Amazon S3 中建立儲存貯體](#) 來保存您的 Athena 查詢結果。您可將此儲存貯體設定為您的查詢輸出位置。

## 步驟 1：建立資料庫

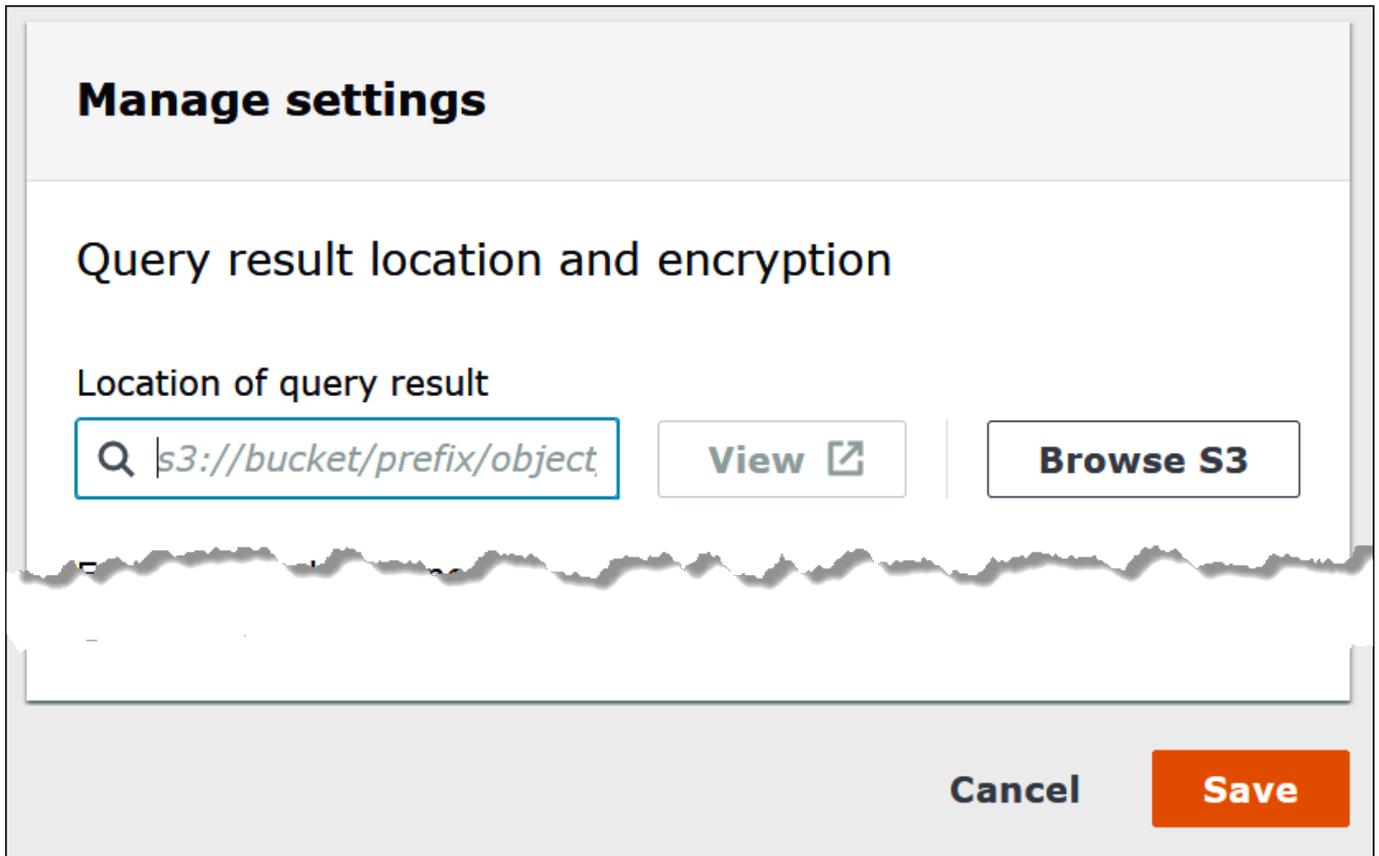
首先，您需要在 Athena 中建立資料庫。

### 建立 Athena 資料庫

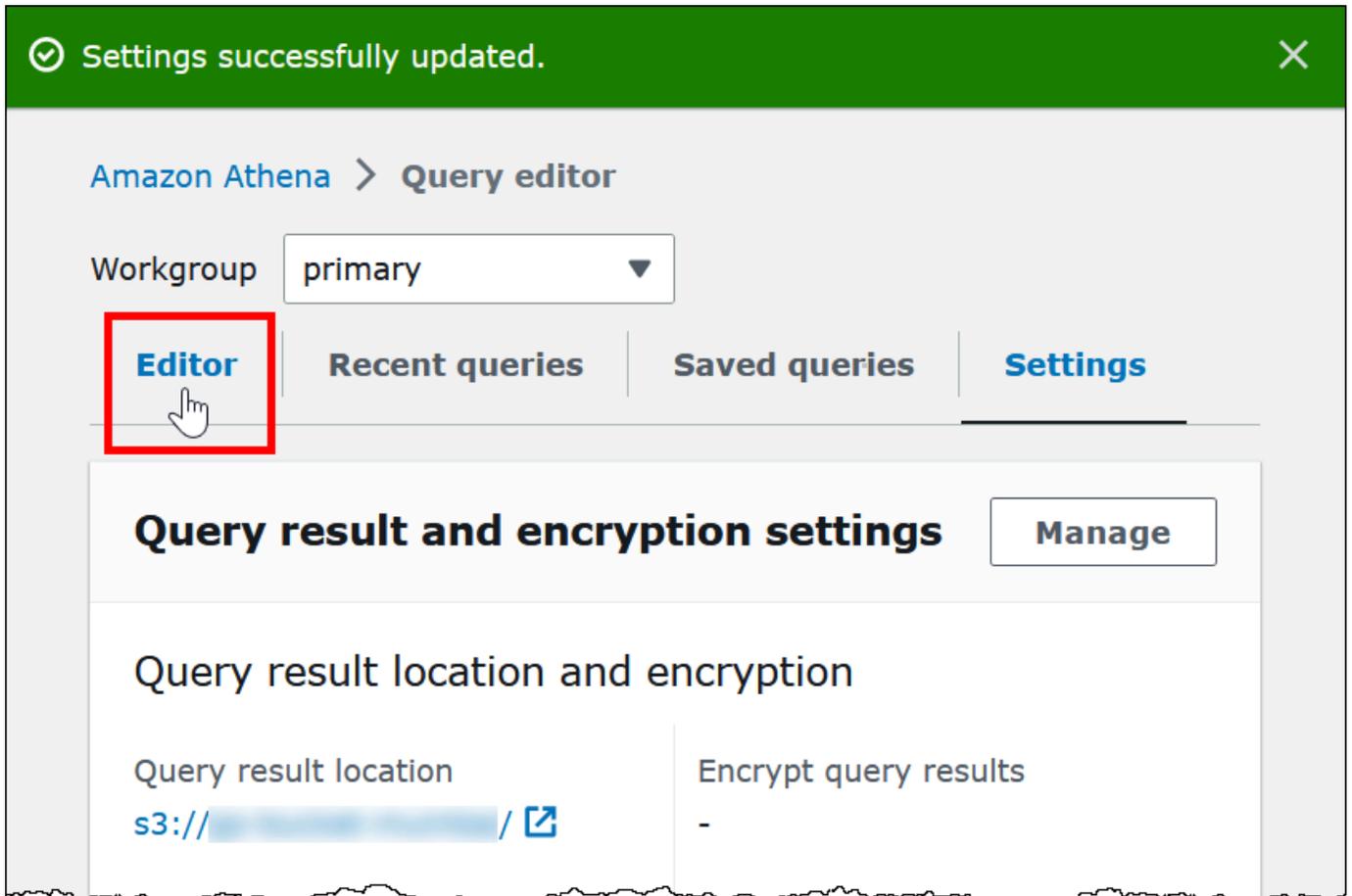
1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果這是您第一次造訪您目前 AWS 區域中的 Athena 主控台，請選擇 Explore the query editor (探索查詢編輯器) 以開啟查詢編輯器。此外，Athena 會於查詢編輯器中開啟。
3. 選擇 Edit Settings (編輯設定) 以在 Amazon S3 中設定查詢結果位置。



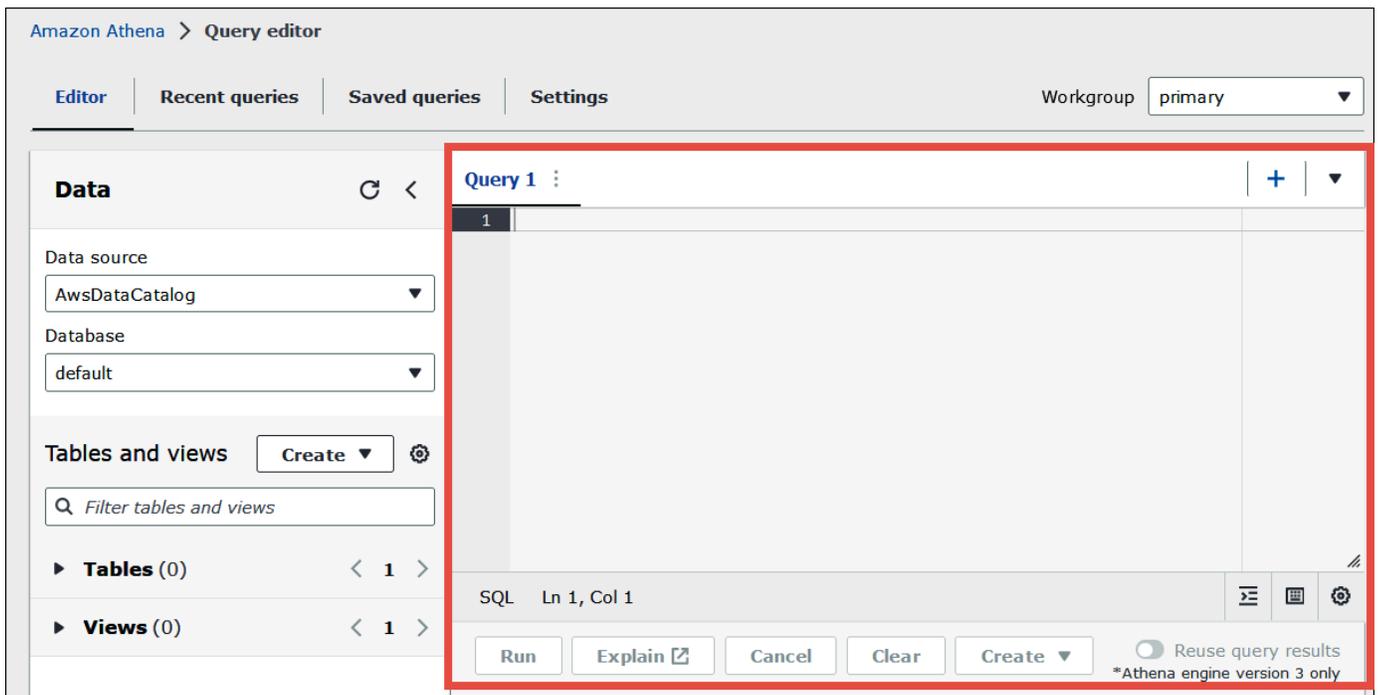
4. 對於 Manage settings (管理設定)，請執行下列其中一項動作：
  - 在 Location of query result (查詢結果的位置) 方塊中，輸入您在 Amazon S3 中為查詢結果建立的儲存貯體路徑。在路徑中加上 `s3://` 作為字首。
  - 選擇 Browse S3 (瀏覽 S3)，選擇您在目前區域建立的 Amazon S3 儲存貯體，然後選擇 Choose (選擇)。



5. 選擇儲存。
6. 選擇 Editor (編輯器) 以切換至查詢編輯器。



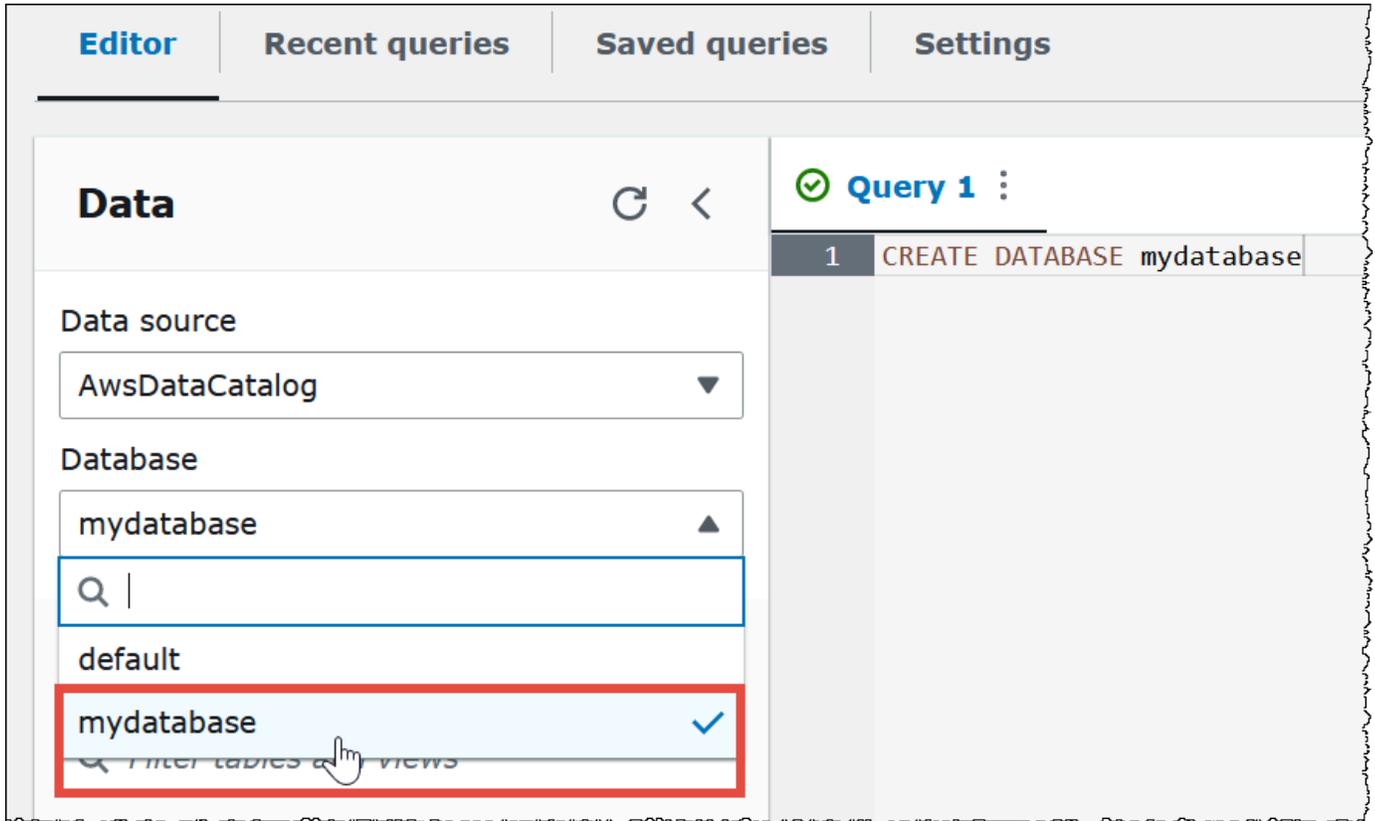
7. 在導覽窗格的右側，您可以使用 Athena 查詢編輯器來輸入和執行查詢和陳述式。



- 若要建立名為 `mydatabase` 的資料庫，請輸入以下 `CREATE DATABASE` 陳述式。

```
CREATE DATABASE mydatabase
```

- 選擇 `Run` (執行) 或按 `Ctrl+ENTER`。
- 從左側的 `Database` (資料庫) 清單中，選擇 `mydatabase` 設為您目前的資料庫。



## 步驟 2：建立資料表

現在您有一個資料庫，您可以為其建立 Athena 資料表。您建立的表格將根據位置中的範例 Amazon CloudFront 日誌資料 `s3://athena-examples-myregion/cloudfront/plaintext/`，*myregion* 是您目前 AWS 區域的位置。

日誌資料範例採用製表符分隔值 (TSV) 格式，這表示製表符字元會用作分隔符號來分隔欄位。資料看起來會如以下範例：出於可讀性考量，摘錄中的製表符已轉換為空格並縮短了最終欄位。

```
2014-07-05 20:00:09 DFW3 4260 10.0.0.15 GET eabcd12345678.cloudfront.net /test-  
image-1.jpeg 200 - Mozilla/5.0[...]  
2014-07-05 20:00:09 DFW3 4252 10.0.0.15 GET eabcd12345678.cloudfront.net /test-  
image-2.jpeg 200 - Mozilla/5.0[...]
```

```
2014-07-05 20:00:10 AMS1 4261 10.0.0.15 GET eabcd12345678.cloudfront.net /test-  
image-3.jpeg 200 - Mozilla/5.0[...]
```

若要讓 Athena 能夠讀取此資料，您可以建立如下所示的簡單 CREATE EXTERNAL TABLE 陳述式。建立資料表的陳述式會定義與資料對應的資料欄、指定資料的分隔方式，並指定 Amazon S3 中包含資料範例的位置。請注意，由於 Athena 希望掃描資料夾中的所有檔案，因此 LOCATION 子句指定了 Amazon S3 資料夾位置，而不是特定檔案。

不要使用這個例子，因為它有一個重要的限制，將很快解釋。

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (  
  `Date` DATE,  
  Time STRING,  
  Location STRING,  
  Bytes INT,  
  RequestIP STRING,  
  Method STRING,  
  Host STRING,  
  Uri STRING,  
  Status INT,  
  Referrer STRING,  
  ClientInfo STRING  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
LINES TERMINATED BY '\n'  
LOCATION 's3://athena-examples-my-region/cloudfront/plaintext/';
```

此範例會建立名為 cloudfront\_logs 的資料表，並為每個欄位指定名稱和資料類型。這些欄位會成為資料表中的資料欄。因為 date 是 [保留字](#)，所以會使用反引號 ( ` ) 字元逸出。ROW FORMAT DELIMITED 意味著 Athena 將使用調用的默認庫 [LazySimpleSerDe](#) 來完成解析數據的實際工作。此範例也會指定欄位以製表符分隔 ( FIELDS TERMINATED BY '\t' )，且檔案中的每個記錄以新行字元 ( LINES TERMINATED BY '\n' ) 結尾。最後，LOCATION 子句會指定要讀取的實際資料所在的 Amazon S3 中的路徑。

如果您有自己的定位鍵或以逗號分隔的資料，只要欄位不包含巢狀資訊，就可以使用如剛顯示的範例之類的 CREATE TABLE 陳述式。但是，如果您有類似的欄，其中包 ClientInfo 含使用不同分隔符號的巢狀資訊，則需要使用不同的方法。

從 ClientInfo 欄位中擷取資料

查看示例數據，以下是最終字段的完整示例 ClientInfo：

```
Mozilla/5.0%20(Android;%20U;%20Windows%20NT%205.1;%20en-US;%20rv:1.9.0.9)%20Gecko/2009040821%20IE/3.0.9
```

如您所見，此欄位為多重值。因為剛剛顯示的範例 CREATE TABLE 陳述式會將定位字元指定為欄位分隔符號，因此無法將欄 ClientInfo 位內的個別元件分解為不同的資料行。所以，一個新的 CREATE TABLE 聲明是必需的。

要從 ClientInfo 字段內的值創建列，可以使用包含 [正則表達式組的正則表達式](#) ( regex )。您指定的 regex 群組會成為單獨的資料表資料欄。若要在 CREATE TABLE 陳述式中使用 regex，則請使用如下所示的語法。此語法指示 Athena 使用 [正則表 SerDe](#) 程式庫和您指定的規則表達式。

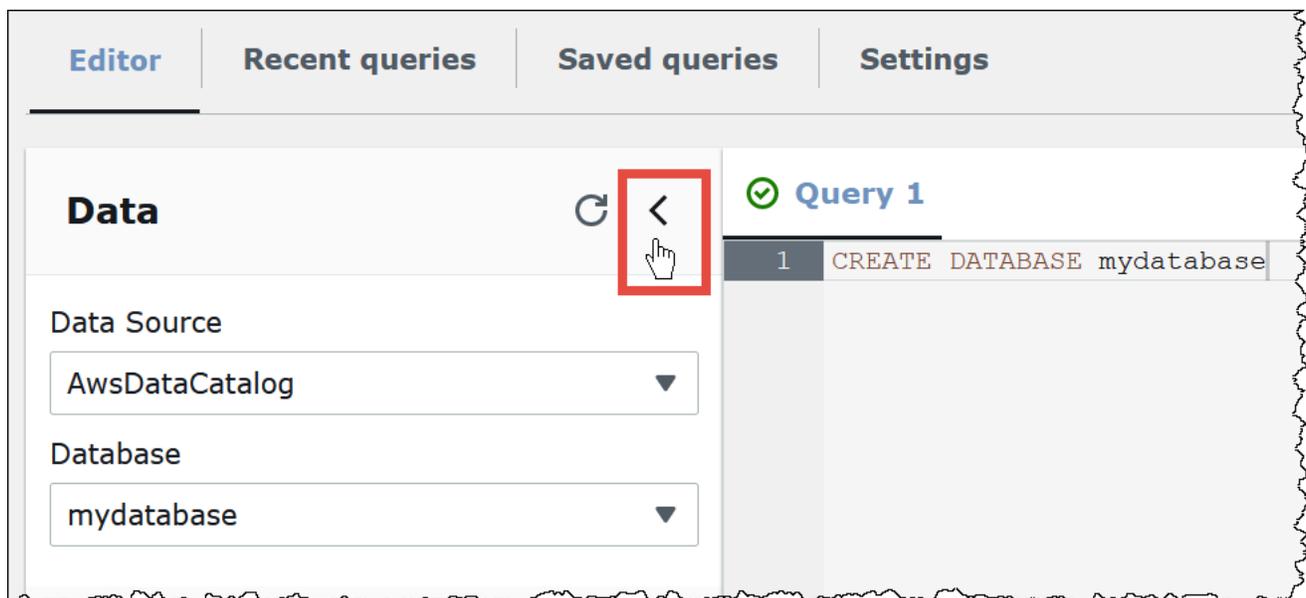
```
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'  
WITH SERDEPROPERTIES ("input.regex" = "regular_expression")
```

規則表達式對於根據複雜的 CSV 或 TSV 資料建立資料表非常有用，但是很難寫入和維護。幸運的是，還有其他程式庫可用於 JSON、Parquet 和 ORC 等格式。如需詳細資訊，請參閱 [支援的 SerDes 和資料格式](#)。

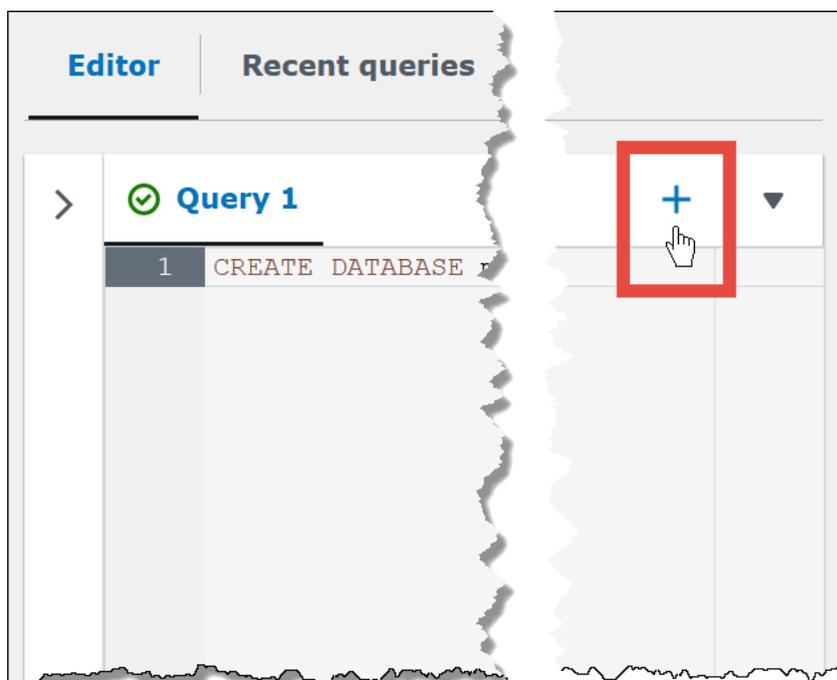
現在您已準備好要開始在 Athena 查詢編輯器中建立資料表。所以該 CREATE TABLE 陳述式和 regex 正好提供給您。

如何在 Athena 中建立資料表

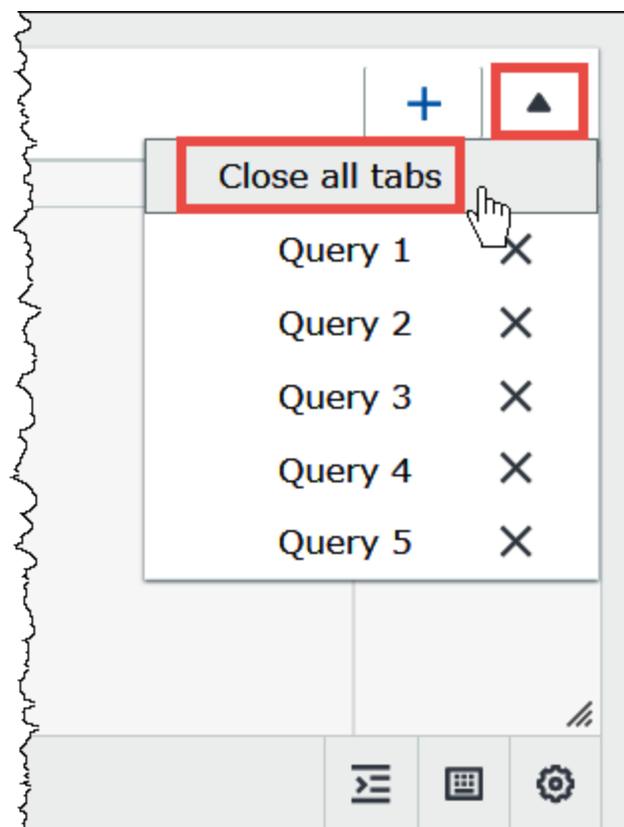
1. 在導覽窗格中，針對 Database (資料庫)，請確認 mydatabase 已選取。
2. 若要在查詢編輯器中給自己更多空間，您可以選擇箭頭圖示來收合導覽窗格。



3. 如需建立新查詢的索引標籤，請選擇加號 (+) 以登入查詢編輯器。一次最多可以開啟十個查詢標籤。



4. 如需關閉一個或多個查詢索引標籤，請選擇加號旁邊的箭頭。如需一次關閉所有索引標籤，請選擇箭頭，然後選擇 Close all tabs (關閉所有索引標籤)。





## 步驟 3：查詢資料

現在您已在 Athena 中根據 Amazon S3 的資料建立 `cloudfront_logs` 資料表，您可以對資料表執行 SQL 查詢，並在 Athena 中查看結果。如需有關在 Athena 中使用 SQL 的詳細資訊，請參閱 [Athena 的 SQL 參考](#)。

若要執行查詢

1. 選擇加號 (+) 打開一個新的查詢索引標籤，並在查詢窗格中輸入以下 SQL 陳述式。

```
SELECT os, COUNT(*) count
FROM cloudfront_logs
WHERE date BETWEEN date '2014-07-05' AND date '2014-08-05'
GROUP BY os
```

2. 選擇執行。

結果類似如下：

🟢 **Completed**  
Time in queue: 0.151 sec    Run time: 3.143 sec    Data scanned: 992.88 KB

**Results (6)**    [Copy](#)    [Download results](#)

🔍 *Search rows*

< **1** >    ⚙️

os	count
MacOS	852
Android	855
Linux	813
OSX	799
iOS	794
Windows	883

3. 若要儲存查詢結果到 .csv 檔案，請選擇 Download results (下載結果)。

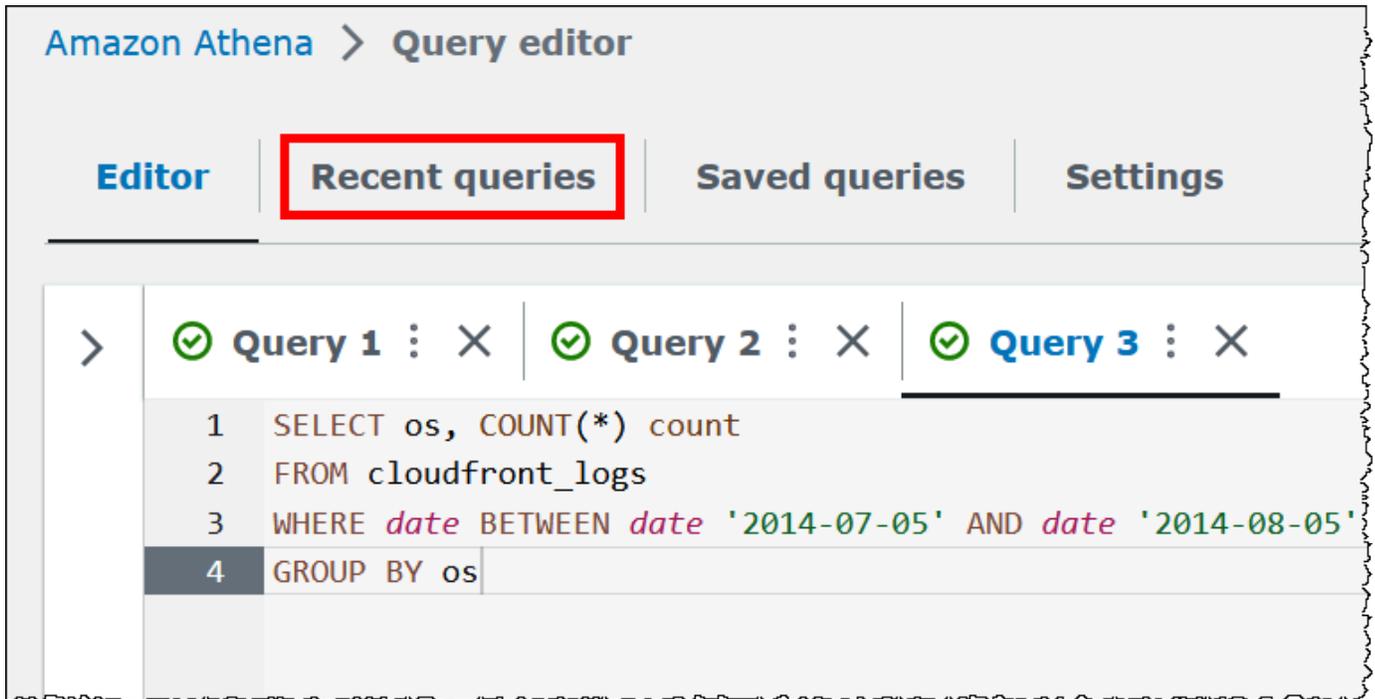
**Results (6)**    [Copy](#)    [Download results](#)

🔍 *Search rows*

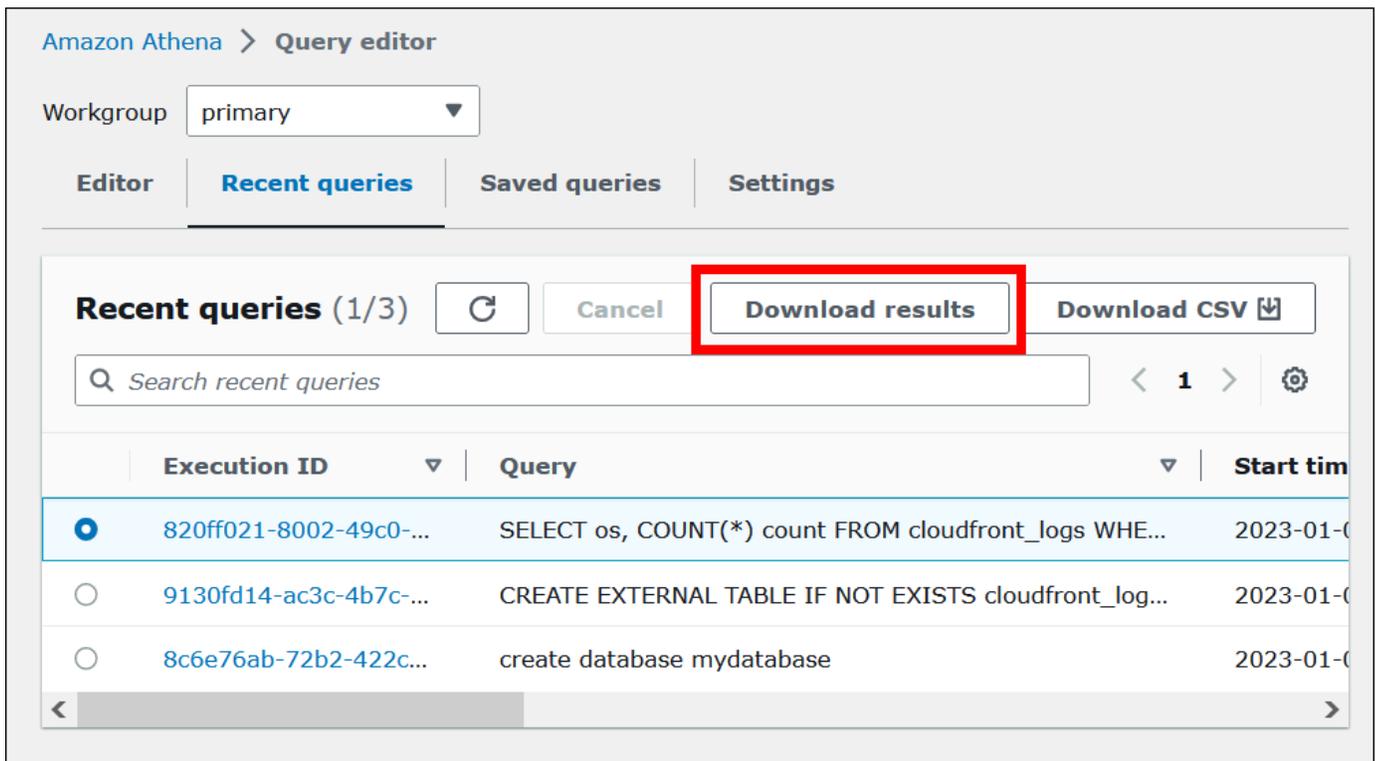
< **1** >    ⚙️

os	count
----	-------

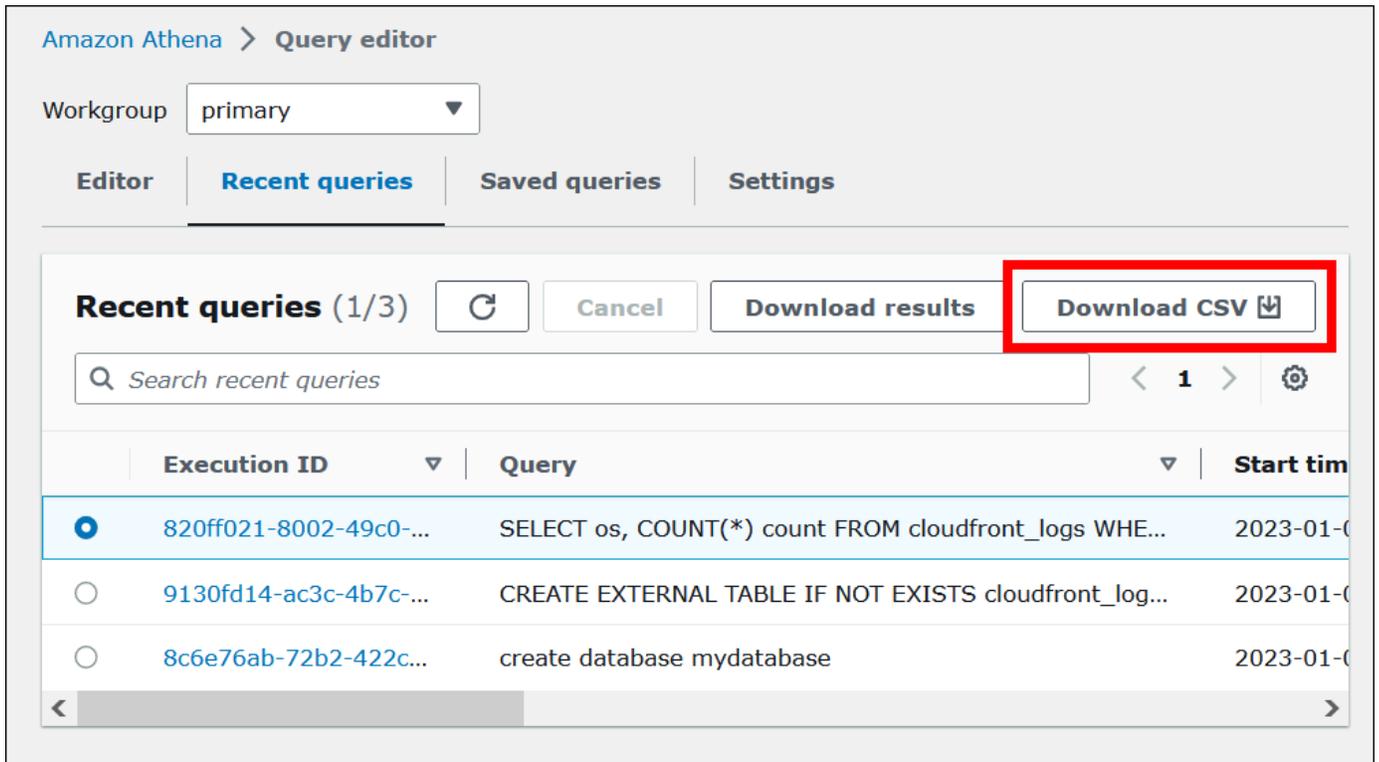
4. 若要檢視或執行先前的查詢，請選擇 Recent queries (近期查詢) 索引標籤。



- 若要從 Recent queries (近期查詢) 索引標籤下載先前查詢的結果，請選取查詢，然後選擇 Download results (下載結果)。查詢會保留 45 天。



- 若要將一個或多個最近使用的 SQL 查詢字串下載至 CSV 檔案，請選擇 Download CSV (下載 CSV)。



The screenshot shows the Amazon Athena Query editor interface. At the top, there's a breadcrumb 'Amazon Athena > Query editor'. Below that, a 'Workgroup' dropdown is set to 'primary'. There are four tabs: 'Editor', 'Recent queries' (which is selected), 'Saved queries', and 'Settings'. Under the 'Recent queries' tab, there are buttons for 'Download results' and 'Download CSV' (the latter is highlighted with a red box). Below the buttons is a search bar 'Search recent queries' and a pagination control showing '1'. A table of recent queries is displayed below, with columns for 'Execution ID', 'Query', and 'Start time'. The first row is selected and highlighted in blue.

Execution ID	Query	Start time
820ff021-8002-49c0-...	SELECT os, COUNT(*) count FROM cloudfront_logs WHE...	2023-01-0
9130fd14-ac3c-4b7c-...	CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_log...	2023-01-0
8c6e76ab-72b2-422c-...	create database mydatabase	2023-01-0

如需詳細資訊，請參閱 [使用查詢結果、近期查詢和輸出檔案](#)。

## 儲存查詢

您可以使用名稱儲存您在查詢編輯器中建立或編輯的查詢。Athena 會將這些查詢儲存在 Saved queries (已儲存的查詢) 索引標籤。您可以使用 Saved queries (已儲存的查詢) 索引標籤重新呼叫、執行、重新命名或刪除已儲存的查詢。如需詳細資訊，請參閱 [使用已儲存的查詢](#)。

## 鍵盤快速鍵和預先輸入的建議

Athena 查詢編輯器為各種動作提供了不同的鍵盤快速鍵，例如執行查詢、格式化查詢、列操作以及尋找及取代。如需詳細資訊和快速鍵的完整清單，請參閱 AWS 大數據部落格中的 [使用 Amazon Athena 查詢編輯器中的鍵盤快速鍵提高生產力](#)。

Athena 查詢編輯器支援預先輸入的程式碼建議，以獲得更快的查詢編寫體驗。為協助您更準確且更有效率地編寫 SQL 查詢，其提供了以下功能：

- 當您輸入時，將會即時顯示關鍵字、本機變數、程式碼片段和目錄項目。
- 當您輸入資料庫名稱或資料表名稱，並在其後加上一個點時，編輯器會輕鬆顯示可供選擇的資料表和資料欄清單。

- 將滑鼠游標移至程式碼片段建議上時，該概要會顯示程式碼片段語法和用法的簡要概觀。
- 為提高程式碼可讀性，關鍵字及其醒目提示規則也已更新，符合 Trino 和 Hive 的最新語法。

此功能預設為啟用。若要啟用或停用此功能，請使用查詢編輯器視窗右下角的程式碼編輯器偏好設定 (齒輪圖示)。

## 連接至其他資料來源

本教學課程使用 Amazon S3 中 CSV 格式的資料來源。如需搭配使用 Athena 的詳細資訊 AWS Glue，請參閱[用於連接 AWS Glue 到 Amazon S3 中的資料來源](#)。您也可以使用 ODBC 和 JDBC 驅動程式、外部 Hive 中繼存放區和 Athena 資料來源連接器，將 Athena 連接到各種資料來源。如需詳細資訊，請參閱[連線至資料來源](#)。

## 連線至資料來源

您可以使用 Amazon Athena 來查詢資料集中以不同位置和格式存放的資料。此資料集可能是 CSV，JSON，Avro、Parquet 或其他格式。

您在 Athena 中用來執行查詢的資料表和資料庫是以中繼資料為基礎。中繼資料是資料集中的基礎資料的相關資料。該中繼資料如何描述您的資料集，就稱為結構描述。例如，資料表名稱、資料表中的資料欄名稱及每一欄的資料類型都是結構描述 (儲存為中繼資料)，用於描述基礎資料集。在 Athena 中，用來組織中繼資料的系統稱為資料目錄或中繼存放區。資料集和描述它的資料目錄合稱為資料來源。

中繼資料與基礎資料集的關係取決於您使用的資料來源類型。關聯式資料來源 (例如 MySQL、PostgreSQL 和 SQL Server) 將中繼資料與資料集緊密整合。在這些系統中，中繼資料最常於寫入資料時寫入。其他資料來源 (例如使用 [Hive](#) 建置的資料來源) 可讓您在讀取資料集 on-the-fly 時定義中繼資料。資料集可以為多種格式，例如 CSV、JSON、Parquet 或 Avro。

Athena 本地支持 AWS Glue Data Catalog 這 AWS Glue Data Catalog 是建立在其他資料集和資料來源 (例如 Amazon S3、亞馬遜紅移和亞馬 Amazon DynamoDB) 之上的資料型錄。您也可以使用各種連接器，將 Athena 連接至其他資料來源。

### 主題

- [與整合 AWS Glue](#)
- [將 Athena 資料連接器用於外部 Hive 中繼存放區](#)
- [使用 Amazon Athena 聯合查詢](#)
- [用於存取資料目錄的 IAM 政策](#)

- [管理資料來源](#)
- [在 Athena 中使用 Amazon DataZone](#)

## 與整合 AWS Glue

[AWS Glue](#)是完全受管的 ETL (擷取、轉換和載入) AWS 服務。其關鍵能力之一是分析和分類資料。您可以使用 AWS Glue 檢索器從 Amazon S3 中的資料自動推斷資料庫和表格結構描述，並將相關的中繼資料存放在 AWS Glue Data Catalog

Athena 會使 AWS Glue Data Catalog 用在您的亞馬遜網路服務帳戶中存放和擷取 Amazon S3 資料的表格中繼資料。資料表中繼資料可讓 Athena 查詢引擎知道如何尋找、讀取和處理您想要查詢的資料。

若要在中建立資料庫和資料表結構描述 AWS Glue Data Catalog，您可以從 Athena 內部的資料來源執行 AWS Glue 爬行者程式，或直接在 Athena 查詢編輯器中執行資料定義語言 (DDL) 查詢。然後，透過您建立的資料庫和資料表結構描述，您可以使用 Athena 中的資料操作 (DML) 查詢來查詢資料。

您可以 AWS Glue Data Catalog 從您自己以外的帳戶註冊。設定所需的 IAM 許可後 AWS Glue，您可以使用 Athena 執行跨帳戶查詢。如需詳細資訊，請參閱 [跨帳戶存取 AWS Glue 資料目錄](#)。

如需有關的詳細資訊 AWS Glue Data Catalog，請參閱AWS Glue 開發人員指南 [AWS Glue 中的〈資料目錄和檢索器〉](#)。

如需說明如何使用 AWS Glue 和 Athena 處理 XML 資料的說明性文章，請參閱大數據部落格中的 [使用和 Amazon Athena 處理 AWS Glue 和分析高度巢狀和 AWS 大型 XML 檔案](#)。

另外收費適用於 AWS Glue。如需詳細資訊，請參閱 [AWS Glue 定價](#)。

### 主題

- [用於連接 AWS Glue 到 Amazon S3 中的資料來源](#)
- [AWS Glue Data Catalog 從另一個帳戶註冊](#)
- [搭配使用 Athena 時的最佳做法 AWS Glue](#)
- [使用重新 AWS CLI 建立 AWS Glue 資料庫及其表格](#)

## 用於連接 AWS Glue 到 Amazon S3 中的資料來源

Athena 可以使用存放中繼資料 (例如表格和欄名稱) 連線 AWS Glue Data Catalog 到存放在 Amazon S3 中的資料。建立連線之後，您的資料庫、資料表和檢視會出現在 Athena 的查詢編輯器中。

若要定義 AWS Glue 要使用的結構描述資訊，您可以建立 AWS Glue 爬行者程式來自動擷取資訊，或者手動新增表格並輸入結構描述資訊。

## 建立 AWS Glue 爬行者程式

您可以從 Athena 主控台開始建立爬蟲程式，然後以整合方式使用 AWS Glue 主控台。建立爬蟲程式時，需要在 Amazon S3 中指定要網路爬取的資料位置。

從 Athena 主控台 AWS Glue 開始建立爬蟲程式

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在查詢編輯器中，選擇位於資料表和檢視旁的建立，然後選擇 AWS Glue 爬蟲程式。
3. 在 AWS Glue 主控台的 Add crawler (新增爬蟲程式) 頁面上，依照步驟建立爬蟲程式。如需詳細資訊，請參閱本指南中的 [使用 AWS Glue 爬行程式](#) 和 [填入 AWS Glue](#) 開發人員指南 AWS Glue Data Catalog 中的。

### Note

Athena 無法辨識您為 AWS Glue 爬行者程 [式指定的排除模式](#)。例如，如果您的 Simple Storage Service (Amazon S3) 儲存貯體同時包含 .csv 和 .json 檔案，並且您將 .json 檔案從爬蟲程式中排除，Athena 會查詢這兩個檔案群組。若要避免這種情況，請將您要排除的檔案放在不同的位置。

## 使用表單新增資料表

下列程序會說明如何利用 Athena 主控台，使用 Create Table From S3 bucket data (從 S3 儲存貯體資料建立資料表) 表單新增資料表。

新增資料表和使用表單輸入結構描述資訊

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在查詢編輯器中，選擇位在 Tables and views (資料表和檢視) 旁的 Create (建立)，然後選擇 S3 bucket data (S3 儲存貯體資料)。
3. 在 Create Table From S3 bucket data (從 S3 儲存貯體資料建立資料表) 表單上，對於 Table name (資料表名稱)，輸入資料表的名稱。
4. 對於 Database configuration (資料庫組態)，選擇現有資料庫或建立新資料庫。

5. 對於 Location of Input Data Set (輸入資料集的位置)，請指定 Amazon S3 中的資料夾路徑，該資料夾包含您要處理的資料集。請勿在路徑中包含檔案名稱。Athena 會掃描您指定之資料夾中的所有檔案。如果您的資料已經進行分割 (例如，

s3://DOC-EXAMPLE-BUCKET/logs/year=2004/month=12/day=11/)，僅輸入基本路徑 (例如，s3://DOC-EXAMPLE-BUCKET/logs/)。

6. 在 Data Format (資料格式) 中，請選擇下列選項：

- 對於 Table type (資料表類型)，選擇 Apache Hive、Apache Iceberg 或 Delta Lake。Athena 預設使用 Apache Hive 資料表類型。如需有關在 Athena 中查詢 Apache Iceberg 的詳細資訊，請參閱 [使用 Apache Iceberg 資料表](#)。如需有關在 Athena 中使用 Delta Lake 資料表的詳細資訊，請參閱 [查詢 Linux Foundation Delta Lake 資料表](#)。
- 在 File format (檔案格式) 中，選擇資料所在的檔案或使用的日誌格式。
  - 對於 Text File with Custom Delimiters (含自訂分隔符號的文字檔) 選項，指定 Field terminator (欄位結束字元) (亦即資料欄分隔符號)。或者，您可以指定標記陣列類型結尾的 Collection terminator (集合結束符號)，或指定標記地圖資料類型結尾的 Collection terminator (集合結束符號)。
- SerDe 函式庫 — SerDe (序列化程式-還原序列化程式) 程式庫會剖析特定的資料格式，以便 Athena 可以為其建立資料表。對於大多數格式，系統會為您選擇預設 SerDe 資料庫。對於下列格式，請根據您的需求選擇程式庫：
  - Apache 網頁記錄 — 選擇 RegexSerDe 或程式 GrokSerDe 庫。對於 RegexSerDe，請在 [Regex 定義] 方塊中提供規則運算式。對於 GrokSerDe，提供屬性的一系列具名規則運算 input.format SerDe 式。具名的規則表達式比規則表達式更容易讀取和維護。如需詳細資訊，請參閱 [查詢存放在 Amazon S3 中的 Apache 日誌](#)。
  - CSV — 選擇 LazySimpleSerDe 逗號分隔資料是否不包含以雙引號括住的值，或是否使用該 java.sql.Timestamp 格式。SerDe 如果您的資料包含引號，或使用 UNIX 數字格式 TIMESTAMP (例如 1564610311)，請選擇 OpenCSV。如需詳細資訊，請參閱 [LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案](#) 及 [SerDe 用於處理 CSV 的開啟 CSV](#)。
  - JSON — 選擇任一 OpenX 或蜂巢 JSON SerDe 程式庫。這兩種格式都預期每個 JSON 文件都以單行文字表示，而且欄位不以換行字元分隔。OpenX SerDe 提供了一些額外的屬性。如需這些屬性的相關資訊，請參閱 [OpenX SerDe](#)。如需有關 Hive 的資訊 SerDe，請參閱 [蜂巢 SerDe](#)。

如需在 Athena 中使用 SerDe 程式庫的詳細資訊，請參閱 [支援的 SerDes 和資料格式](#)。

7. 對於 SerDe 性質，請根據您使用的 SerDe 資源庫和需求加入、編輯或移除性質和值。

- 若要新增 SerDe 屬性，請選擇 [新增 SerDe 屬性]。
  - 在 Name (名稱) 欄位中，輸入屬性的名稱。
  - 在 Value (值) 欄位中，輸入屬性的值。
  - 若要移除 SerDe 屬性，請選擇 [移除]。
8. 在 Table properties (資料表屬性) 中，請根據您的需求選擇或編輯資料表屬性。
- 在 Write compression (寫入壓縮) 中，選擇壓縮選項。是否提供寫入壓縮選項和可用壓縮選項取決於資料格式。如需詳細資訊，請參閱 [Athena 壓縮支援](#)。
  - 如果基礎資料是在 Amazon S3 中加密，在 Encryption (加密) 中，選取 Encrypted data set (加密的資料集)。此選項會在 CREATE TABLE 陳述式中將 has\_encrypted\_data 資料表屬性設定為 true。
9. 在 Column details (資料欄詳細資訊) 中，輸入要新增至資料表的資料欄名稱和資料類型。
- 若要一次新增更多資料欄，請選擇 Add a column (新增資料欄)。
  - 若要快速新增更多資料欄，請選擇 Bulk add columns (大量新增資料欄)。在文字方塊中，輸入逗號分隔的資料欄清單，格式為 *column\_name data\_type, column\_name data\_type*[, ...]，然後選擇 Add (新增)。
10. (選用) 對於 Partition details (分割區詳情)，新增一個或多個資料欄名稱和資料類型。分割會根據資料欄值將相關資料保存在一起，有助於減少每次查詢掃描的資料量。如需分割的相關資訊，請參閱 [在 Athena 中分割資料](#)。
11. (選用) 在 Bucketing (歸納) 中，您可以指定包含要組合在一起的資料列的一個或多個資料欄，然後將這些資料列放入多個儲存貯體中。這可允許您在指定已歸納的資料欄值時僅查詢要讀取的儲存貯體。
- 在 Buckets (儲存貯體) 中，請選取具有大量唯一值 (例如主索引鍵) 且經常用來篩選查詢中資料的一個或多個資料欄。
  - 在 Number of buckets (儲存貯體數量) 中，輸入可讓檔案具有最佳大小的儲存貯體數量。如需詳細資訊，請參閱 AWS 大數據部落格中的 [Amazon Athena 的十大效能調校秘訣](#)。
  - 若要指定歸納的資料欄，CREATE TABLE 陳述式會使用下列語法：

```
CLUSTERED BY (bucketed_columns) INTO number_of_buckets BUCKETS
```

**Note**

Bucketing (歸納) 選項不適用於 Iceberg 資料表類型。

12. Preview table query (預覽資料表查詢) 方塊顯示由您所輸入到表單中的資訊產生的 CREATE TABLE 陳述式。無法直接編輯預覽陳述式。若要變更陳述式，請修改預覽上方的表單欄位，或不使用表單，在查詢編輯器中[直接建立陳述式](#)。
13. 選擇 Create table (建立資料表) 以在查詢編輯器中執行產生的陳述式並建立資料表。

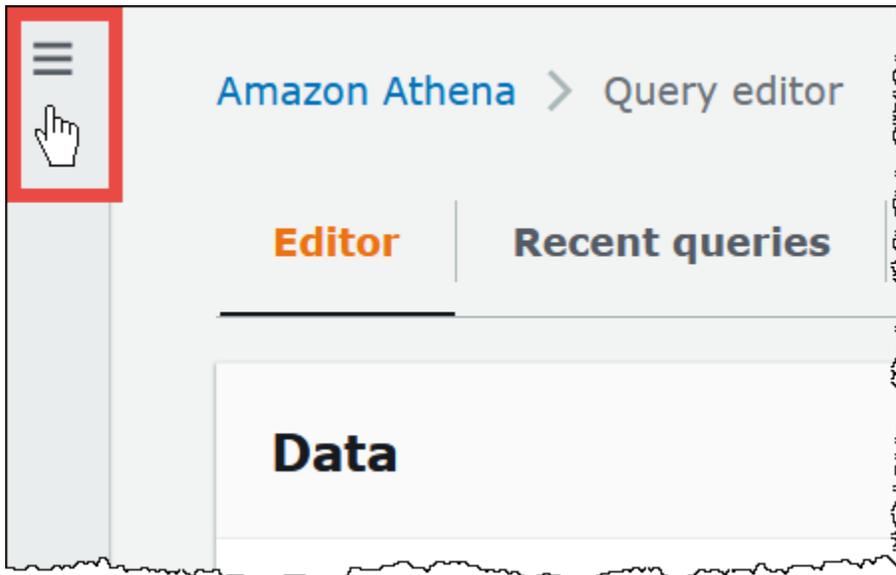
## AWS Glue Data Catalog 從另一個帳戶註冊

您可以使用 Athena 的跨帳戶 AWS Glue 目錄功能，從您自己以外的帳戶註冊 AWS Glue 目錄。在您為 AWS Glue 設定所需的 IAM 許可，並將目錄註冊為 Athena DataCatalog 資源後，您可以使用 Athena 來執行跨帳戶查詢。如需有關設定所需許可的詳細資訊，請參閱[跨帳戶存取 AWS Glue 資料目錄](#)。

下列程序說明如何使用 Athena 主控台，在您帳戶以外的 Amazon Web Services 帳戶中設定 AWS Glue Data Catalog 作為資料來源。

### AWS Glue Data Catalog 從另一個帳戶註冊

1. 請按照 [跨帳戶存取 AWS Glue 資料目錄](#) 中的步驟，確保您已獲得許可，能查詢其他帳戶中的資料目錄。
2. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
3. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



4. 選擇 Data Source (資料來源)。
5. 在右上角，選擇 Create data source (建立資料來源)。
6. 在 [選擇資料來源] 頁面上，針對 [資料來源] 選擇 [S3-] AWS Glue Data Catalog，然後選擇 [下一步]。
7. 在 Enter data source details (輸入資料來源詳細資訊) 頁面上的 AWS Glue Data Catalog 區段，針對選擇 AWS Glue Data Catalog，在另一個帳戶選擇 AWS Glue Data Catalog。
8. 在 Data source details (資料來源詳細資訊) 輸入以下資訊：
  - Data source name (資料來源名稱) – 輸入您希望在 SQL 查詢中使用的名稱，以參考其他帳戶中的資料目錄。
  - Description (描述) – (選填) 輸入其他帳戶裡資料目錄的描述。
  - Catalog ID (目錄 ID) – 輸入資料目錄所屬帳戶的 12 位數 Amazon Web Services 帳戶 ID。Amazon Web Services 帳戶 ID 即為目錄 ID。
9. (選用) 在 Tags (標籤) 中，輸入您希望與資料來源相關聯的鍵值對。如需標籤的詳細資訊，請參閱 [標記 Athena 資源](#)。
10. 選擇下一步。
11. 在 Review and create (檢閱並建立) 頁面上，檢閱您提供的資訊，然後選擇 Create data source (建立資料來源)。Data source details (資料來源詳細資訊) 頁面列出了註冊資料型錄的資料庫和標籤。
12. 選擇 Data Source (資料來源)。您註冊的資料型錄列於 Data Source Name (資料來源名稱) 資料欄。
13. 若要檢視或編輯資料型錄的相關資訊，請選擇型錄，然後選擇 Actions (動作)、Edit (編輯)。

14. 若要刪除新資料型錄，請選擇型錄，然後選擇 Actions (動作)、Delete (刪除)。

如需詳細資訊，請參閱AWS 大數據部落格中的[使用 Amazon Athena 查詢跨帳戶 AWS Glue Data Catalog](#)。

## 搭配使用 Athena 時的最佳做法 AWS Glue

搭配使用 Athena 時 AWS Glue Data Catalog，您可以使用建立 AWS Glue 要在 Athena 中查詢的資料庫和表格 (結構描述)，或者您可以使用 Athena 建立結構描述，然後在 AWS Glue 和相關服務中使用它們。本主題提供使用這兩個方法時的考量和最佳實務。

背後原理是，Athena 使用 Trino 來處理 DML 陳述式，並使用 Hive 來處理負責建立和修改結構描述的 DDL 陳述式。有了這些技術，有幾個慣例遵循，以便 Athena 和一起 AWS Glue 工作良好。

在本主題中

- [資料庫、資料表和資料欄名稱](#)
- [使用 AWS Glue 爬蟲](#)
  - [排程爬蟲程式以保持 AWS Glue Data Catalog 和 Amazon S3 同步](#)
  - [對爬蟲程式使用多個資料來源](#)
  - [同步分割區結構描述以避免 "HIVE\\_PARTITION\\_SCHEMA\\_MISMATCH"](#)
  - [更新資料表中繼資料](#)
- [使用 CSV 檔案](#)
  - [CSV 資料含括在引號中](#)
  - [具有標頭的 CSV 檔案](#)
- [AWS Glue 分割區索引和篩選](#)
- [使用地理空間資料](#)
- [使用與 AWS Glue Athena 的 ETL 工作](#)
  - [針對 AWS Glue ETL 工作使用 Athena 建立資料表](#)
  - [使用 ETL 任務來最佳化查詢效能](#)
  - [在轉換為 ORC 時將 SMALLINT 和 TINYINT 資料類型轉換為 INT](#)
  - [自動化 ET AWS Glue L 工作](#)

## 資料庫、資料表和資料欄名稱

在中建立結構描述以 AWS Glue 便在 Athena 中進行查詢時，請考慮下列事項：

- 中的資料庫名稱、表格名稱和欄名稱可接受的字元 AWS Glue 必須是 UTF-8 字串，且應小寫。請注意，Athena 會在建立資料庫、資料表或資料行時，自動降低 DDL 查詢中的任何大寫字母名稱。字串長度不得小於 1 或超過 255 個位元組。可以使用的字元包括空格。
- 目前，名稱開頭可能有前導空格。由於這些前導空格很難偵測，而且在建立之後可能會造成可用性問題，因此請避免意外建立含有前導空格的物件名稱。
- 如果您使用 [AWS::Glue::Database](#) AWS CloudFormation 範本來建立資 AWS Glue 料庫，但未指定資料庫名稱，則會 AWS Glue 自動產生與 Athena 不相容的 `##### - random_string #####` #稱。
- 您可以使用「目 AWS Glue 錄管理員」更名欄，但無法更名表格名稱或資料庫名稱。若要解決此限制，您必須使用舊資料庫的定義，建立具有新名稱的資料庫。然後，您可以使用舊資料庫中資料表的定義，重新建立新資料庫中的資料表。若要這麼做，您可以使用 AWS CLI 或 AWS Glue SDK。如需這些步驟，請參閱 [使用重新 AWS CLI 建立 AWS Glue 資料庫及其表格](#)。

如需有關中的資料庫和表格的詳細資訊 AWS Glue，請參閱 AWS Glue 開發人員指南中的 [資料庫](#) 和 [表格](#)。

### 使用 AWS Glue 爬蟲

AWS Glue 搜尋器可協助探索資料集的結構描述，並將其註冊為「AWS Glue 資料目錄」中的資料表。爬蟲程式可逐一瀏覽您的資料，並判斷結構描述。此外，爬蟲程式可偵測和登錄分割區。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的 [定義爬蟲程式](#)。可從 Athena 查詢來自己成功編目的資料的資料表。

#### Note

Athena 無法辨識您為 AWS Glue 爬行者程 [式指定的排除模式](#)。例如，如果您的 Simple Storage Service (Amazon S3) 儲存貯體同時包含 .csv 和 .json 檔案，並且您將 .json 檔案從爬蟲程式中排除，Athena 會查詢這兩個檔案群組。若要避免這種情況，請將您要排除的檔案放在不同的位置。

## 排程爬蟲程式以保持 AWS Glue Data Catalog 和 Amazon S3 同步

AWS Glue 爬行者程式可以設定為按排程或按需執行。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[任務和爬蟲程式的時間排程](#)。

如果您有固定時間送達分區資料表的資料，您可以設定 AWS Glue 爬行者程式按排程執行，以偵測和更新表格分割區。您不再需要花費大量時間和金錢執行 `MSCK REPAIR` 命令，也不需要再手動執行 `ALTER TABLE ADD PARTITION` 命令。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[資料表分割區](#)。

### 對爬蟲程式使用多個資料來源

當 AWS Glue 爬行者程式掃描 Amazon S3 並偵測到多個目錄時，它會使用啟發式來判斷表格的根目錄在目錄結構中的位置，以及哪些目錄是表格的分割區。在某些情況下，在兩個或多個目錄偵測到類似的結構描述時，爬蟲程式可能會將它們視為分割區，而不是個別的資料表。協助爬蟲程式探索個別資料表的其中一種方法是，將每個資料表的根目錄新增為爬蟲程式的資料存放區。

Amazon S3 中的下列分割區為範例：

```
s3://DOC-EXAMPLE-BUCKET/folder1/table1/partition1/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table1/partition2/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table1/partition3/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table2/partition4/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table2/partition5/file.txt
```

如果 `table1` 和 `table2` 的結構描述類似，並且 AWS Glue 中的單一資料來源設定為 `s3://DOC-EXAMPLE-BUCKET/folder1/`，則爬蟲程式可能會建立具有兩個分割區資料欄的單一資料表：一個分割區資料欄包含 `table1` 和 `table2`，第二個分割區資料欄包含 `partition1` 到 `partition5`。

若要讓 AWS Glue 爬行者程式建立兩個個別的表格，請將爬行者程式設定為具有兩個資料來源 `s3://DOC-EXAMPLE-BUCKET/folder1/table2`，`s3://DOC-EXAMPLE-BUCKET/folder1/table1/` 並如下列程序所示。

若要將 S3 資料存放區新增至中的現有爬行者程式 AWS Glue

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，網址為 <https://console.aws.amazon.com/glue/>。
2. 在導覽窗格中，選擇 Crawlers (爬蟲程式)。
3. 選擇您的爬蟲程式連結，然後選擇 Edit (編輯)。

4. 對於 Step 2: Choose data sources and classifiers (步驟 2：選擇資料來源和分類器)，選擇 Edit (編輯)。
5. 對於 Data sources (資料來源)，選擇 Add a data source (新增資料來源)。
6. 在 Add data source (新增資料來源) 對話方塊中，用於 S3 path (S3 路徑)，選擇 Browse (瀏覽)。
7. 選取您想要使用的儲存貯體，然後選擇 Choose (選擇)。

您新增的資料來源會顯示在 Data sources (資料來源) 的清單中。

8. 選擇下一步。
9. 在 Configure Security Settings (設定安全設定) 頁面上，建立或選擇爬蟲程式的 IAM 角色，然後選擇 Next (下一步)。
10. 確定 S3 路徑結尾為尾端斜線，然後選擇 Add an S3 data source (新增 S3 資料來源)。
11. 在 Set output and scheduling (設定輸出與排程) 頁面上，針對 Output configuration (輸出組態)，選擇目標資料庫。
12. 選擇下一步。
13. 在 Review (檢閱和更新) 頁面上，檢閱您所做的選擇。若要編輯步驟，請選擇 Edit (編輯)。
14. 選擇更新。

#### 同步分割區結構描述以避免 "HIVE\_PARTITION\_SCHEMA\_MISMATCH"

對於具有分割區 AWS Glue 資料行的「資料目錄」中的每個表格，結構定義會儲存在表格層級以及表格中的每個個別分割區。AWS Glue 爬行者程式會根據分割區內讀取的資料樣本，填入分割區的結構描述。如需詳細資訊，請參閱 [對爬蟲程式使用多個資料來源](#)。

當 Athena 執行查詢時，會驗證資料表的結構描述，以及查詢所需的任何分割區的結構描述。驗證會依序比較資料欄資料類型，並確認它們符合重疊的資料欄。這可避免意外的操作，例如從資料表中間新增或移除資料欄。如果 Athena 偵測到分割區的結構描述與資料表的結構描述不同，Athena 可能無法處理查詢，並且會在 HIVE\_PARTITION\_SCHEMA\_MISMATCH 時操作失敗。

您有一些方式可以修正這個問題。首先，如果資料是意外新增，您可以移除造成結構描述中差異的資料檔案，捨棄分割區，並重新編目資料。第二，您可以捨棄個別分割區，然後在 Athena 內執行 MSCK REPAIR，以使用資料表的結構描述重新建立分割區。這個第二個選項只有在您確定套用的結構描述將可繼續正確讀取資料時才適用。

#### 更新資料表中繼資料

AWS Glue 搜索之後，爬行者程式會自動指派特定資料表中繼資料，以協助使其與 Apache Hive、Presto 和 Spark 等其他外部技術相容。有時爬蟲程式可能不正確地指派中繼資料屬性。在使用

Athena 查詢資料表之 AWS Glue 前，請手動更正中的屬性。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[檢視與編輯資料表詳細資訊](#)。

AWS Glue 當 CSV 檔案在每個資料欄位周圍都有引號時，可能會錯誤地指派中繼資料，導致 serializationLib 屬性錯誤。如需詳細資訊，請參閱 [CSV 資料含括在引號中](#)。

## 使用 CSV 檔案

CSV 檔案中的資料值偶爾會有引號，目的是用於每個資料欄，並且 CSV 檔案可能會包含會標頭值，它不是要分析之資料的一部分。當您使用 AWS Glue 從這些檔案建立結構描述時，請遵循本節中的指導。

## CSV 資料含括在引號中

您可能有一個 CSV 檔案，其中的資料欄位以雙引號括住，如下列範例所示：

```
"John","Doe","123-555-1231","John said \"hello\""  
"Jane","Doe","123-555-9876","Jane said \"hello\""
```

若要在 Athena 中對使用引號值的 CSV 檔案建立的資料表執行查詢，您必須修改中的資料表屬性 AWS Glue 才能使用 OpenCSV SerDe。如需 OpenCSV 的詳細資訊 SerDe，請參閱[SerDe 用於處理 CSV 的開啟 CSV](#)。

## 在 AWS Glue 主控台中編輯表格屬性的步驟

1. 在 AWS Glue 主控台瀏覽窗格中，選擇 [表格]。
2. 選擇您想要編輯的資料表連結，然後選擇 Action (動作)、Edit table (編輯資料表)。
3. 在 Edit table (編輯資料表) 頁面上，進行下列變更：
  - 對於 Serialization lib (序列化程式庫)，請輸入 org.apache.hadoop.hive.serde2.OpenCSVSerde。
  - 對於 Serde parameters (SerDe 參數)，請輸入金鑰 escapeChar、quoteChar 以及 separatorChar 的值：
    - 對於 escapeChar，請輸入反斜線 (\)。
    - 對於 quoteChar，請輸入雙引號 (")。
    - 對於 separatorChar，請輸入逗號 (,)。
4. 選擇儲存。

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[檢視與編輯資料表詳細資訊](#)。

## 以程式設計方式更新 AWS Glue

您可以使用 AWS Glue [UpdateTable](#) API 作業或[更新資料表](#) CLI 命令來修改資料表定義中的 SerDeInfo 區塊，如下列範例 JSON 所示。

```
"SerDeInfo": {
  "name": "",
  "serializationLib": "org.apache.hadoop.hive.serde2.OpenCSVSerde",
  "parameters": {
    "separatorChar": ",",
    "quoteChar": "\""
    "escapeChar": "\\"
  }
},
```

## 具有標頭的 CSV 檔案

當您在 Athena 中以 CREATE TABLE 陳述式定義資料表時，您可以使用 skip.header.line.count 資料表屬性忽略 CSV 資料中的標頭，如下列範例所示。

```
...
STORED AS TEXTFILE
LOCATION 's3://DOC-EXAMPLE-BUCKET/csvdata_folder/';
TBLPROPERTIES ("skip.header.line.count"="1")
```

或者，您也可以事先移除 CSV 標頭，使標頭資訊不包含在 Athena 查詢結果中。實現這一目標的一種方法是使用 AWS Glue 工作，這些工作執行擷取、轉換和載入 (ETL) 工作。您可以 AWS Glue 使用 PySpark Python 方言延伸的語言來撰寫指令碼。如需詳細資訊，請參閱[AWS Glue 開發人員指南中的使用 AWS Glue 編寫工作](#)。

下列範例會示範 AWS Glue 指令碼中的函數，該函式會使用寫出動態影格 from\_options，並將 writeHeader 格式選項設定為 false，這會移除標頭資訊：

```
glueContext.write_dynamic_frame.from_options(frame = applymapping1, connection_type =
"s3", connection_options = {"path": "s3://DOC-EXAMPLE-BUCKET/MYTABLEDATA/"}, format =
"csv", format_options = {"writeHeader": False}, transformation_ctx = "datasink2")
```

## AWS Glue 分割區索引和篩選

當 Athena 查詢已分割的資料表時，會擷取和篩選可用資料表分割區至與您的查詢相關的子集。隨著新資料和分割區的增加，處理分割區需要更長的時間，執行查詢的時間可能會增加。如果您有一個具有大量分割區的資料表，並且分割區的數量隨著時間的推移而增加，請考慮使用 AWS Glue 分割區索引和篩選。分割區索引允許 Athena 最佳化分割區處理，並改善高度已分割的資料表的查詢效能。在資料表屬性中設定分割區篩選，一共有兩個步驟：

1. 在 AWS Glue 建立分割區索引。
2. 啟用資料表的分割區篩選。

### 建立分割區索引

如需在中建立分割區索引的步驟 AWS Glue，請參閱 [關閉 AWS Glue 發人員指南中的使用分割區索引](#)。  
如需中分割區索引的限制 AWS Glue，請參閱該頁面的 [關於分割區索引](#) 一節。

### 啟用分割區篩選

如需啟用資料表的分割區篩選，您必須在 AWS Glue 設定新資料表屬性。如需如何在中設定表格屬性的步驟 AWS Glue，請參閱 [設定分割區投影](#) 頁面。當您在中編輯表格詳細資訊時 AWS Glue，請將下列索引鍵值配對新增至表格特性段落：

- 對於 Key (索引鍵)，新增 `partition_filtering.enabled`。
- 對於 Value (數值)，新增 `true`

您可以將 `partition_filtering.enabled` 設定為 `false`，隨時停用此資料表上的分割區篩選。

完成上述步驟後，您可以返回 Athena 主控台以查詢資料。

如需有關使用分割區索引和篩選的詳細資訊，請參閱 AWS 大數據部落格中的 [使用 AWS Glue Data Catalog 分區索引改善 Amazon Athena 查詢效能](#)。

### 使用地理空間資料

AWS Glue 本身不支援已知文字 (WKT)、已知二進位檔 (WKB) 或其他 PostGIS 資料類型。AWS Glue 分類器會剖析地理空間資料，並使用格式支援的資料類型 (例如 `varchar CSV`) 對其進行分類。與其他 AWS Glue 表格一樣，您可能需要更新從空間資料建立的表格的性質，以允許 Athena 依原樣剖析這些資料類型。如需詳細資訊，請參閱 [使用 AWS Glue 爬蟲](#) 及 [使用 CSV 檔案](#)。Athena 可能無法按原

樣剖析 AWS Glue 表格中的某些地理空間資料類型。如需在 Athena 中使用地理空間資料的詳細資訊，請參閱[查詢地理空間資料](#)。

## 使用與 AWS Glue Athena 的 ETL 工作

AWS Glue 工作執行 ETL 操作。AWS Glue 工作會執行從來源擷取資料、轉換資料並將其載入目標的指令集。如需詳細資訊，請參閱[AWS Glue 開發人員指南中的使用 AWS Glue 編寫工作](#)。

## 針對 AWS Glue ETL 工作使用 Athena 建立資料表

您在 Athena 中建立的資料表必須新增一個資料表屬性，稱為 `classification`，它可以識別資料格式。這允許 AWS Glue 使用表進行 ETL 作業。分類值可以是 `avro`、`csv`、`json`、`orc`、`parquet` 或 `xml`。Athena 中的 CREATE TABLE 陳述式範例如下所示：

```
CREATE EXTERNAL TABLE sampleTable (  
  column1 INT,  
  column2 INT  
) STORED AS PARQUET  
TBLPROPERTIES (  
  'classification'='parquet')
```

如果在建立資料表時未新增 table 屬性，您可以使用 AWS Glue 主控台新增它。

若要使用 AWS Glue 主控台新增分類表屬性

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，[網址為 https://console.aws.amazon.com/glue/](https://console.aws.amazon.com/glue/)。
2. 在主控台導覽窗格中，選擇 Tables (資料表)。
3. 選擇您想要編輯的資料表連結，然後選擇 Action (動作)、Edit table (編輯資料表)。
4. 向下捲動到 Table properties (資料表屬性) 的區段中。
5. 選擇新增。
6. 在 Key (索引鍵) 欄位，輸入 **classification**。
7. 在 Value (值) 欄位，輸入資料類型 (例如，**json**)。
8. 選擇儲存。

在 Table details (資料表詳細資訊) 區段中，您輸入的資料類型會顯示在 Classification (分類) 表格的欄位。

如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[使用資料表](#)。

## 使用 ETL 任務來最佳化查詢效能

AWS Glue 工作可協助您將資料轉換為可最佳化 Athena 查詢效能的格式。資料格式對於 Athena 中的查詢效能和查詢成本有重大影響。

我們建議使用實木複合地板和 ORC 數據格式。AWS Glue 支援寫入這兩種資料格式，可讓您更輕鬆快速地將資料轉換為 Athena 的最佳格式。如需這些格式的詳細資訊以及其他改善效能的方法，請參閱 [Amazon Athena 的十大效能調整秘訣](#)。

在轉換為 ORC 時將 SMALLINT 和 TINYINT 資料類型轉換為 INT

為了減少 Athena 無法讀取 ETL 工作所產生之 SMALLINT 和 TINYINT 資料類型的可能性，請在使用精靈或撰寫 AWS Glue ETL 工作的指令碼 INT 時轉換 SMALLINT 和 TINYINT 為。

### 自動化 ETL AWS Glue 工作

您可以將 AWS Glue ETL 工作設定為根據觸發程序自動執行。當將來自外部 AWS 的資料以不理想的格式推送至 Amazon S3 儲存貯體以便在 Athena 進行查詢時，此功能非常理想。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的 [觸發 AWS Glue 任務](#)。

## 使用重新 AWS CLI 建立 AWS Glue 資料庫及其表格

您無法直接重新命名 AWS Glue 資料庫，但您可以複製其定義、修改定義，以及使用定義來重新建立具有不同名稱的資料庫。同樣地，您可以複製舊資料庫中資料表的定義、修改定義，並使用修改後的定義重新建立新資料庫中的資料表。

### Note

顯示的方法不會複製資料表分割區。

Windows 的下列程序假設您 AWS CLI 已針對 JSON 輸出進行設定。若要變更中的預設輸出格式 AWS CLI，請執行 `aws configure`。

若要使用複製 AWS Glue 資料庫 AWS CLI

1. 在命令提示字元中，執行下列 AWS CLI 命令以擷取要複製之 AWS Glue 資料庫的定義。

```
aws glue get-database --name database_name
```

- 如需有關 `get-database` 命令的詳細資訊，請參閱 [get-database](#)。
- 將 JSON 輸出儲存到桌面上的檔案中，該檔案包含新資料庫的名稱 (例如 `new_database_name.json`)。
  - 在文字編輯器中開啟 `new_database_name.json` 檔案。
  - 在 JSON 檔案中，執行下列步驟：
    - 刪除文件末尾的外部 `{ "Database":` 條目和相應的右括號 `}`。
    - 將 `Name` 項目變更為新的資料庫名稱。
    - 移除 `CatalogId` 欄位。
  - 儲存檔案。
  - 在命令提示字元中，執行下列 AWS CLI 命令，以使用修改後的資料庫定義檔案，以新名稱建立資料庫。

```
aws glue create-database --database-input "file://~/Desktop/new_database_name.json"
```

如需有關 `create-database` 命令的詳細資訊，請參閱 [create-database](#)。若要取得有關從檔案載入 AWS CLI 參數的資訊，請[AWS CLI 參閱《AWS Command Line Interface 使用指南》中的〈從檔案載入參數〉](#)。

- 若要驗證是否已在中建立新資料庫 AWS Glue，請執行下列命令：

```
aws glue get-database --name new_database_name
```

現在，您已準備好取得要複製到新資料庫的資料表的定義、修改定義，並使用修改後的定義在新資料庫中重新建立資料表。此程序不會變更資料表名稱。

若要使用複製 AWS Glue 表格 AWS CLI

- 在命令提示字元中，執行下列 AWS CLI 命令。

```
aws glue get-table --database-name database_name --name table_name
```

如需有關 `get-table` 命令的詳細資訊，請參閱 [get-table](#)。

- 將 JSON 輸出儲存到 Windows 桌面上的檔案中，該檔案包含資料表名稱 (例如 `table_name.json`)。

3. 在文字編輯器中開啟該檔案。
4. 在 JSON 檔案中，移除檔案結尾的外部 {"Table": 項目和對應的大括弧 }。
5. 在 JSON 檔案中，移除下列項目及其值：
  - DatabaseName – 因為 create-table CLI 命令使用 --database-name 參數，所以不需要此項目。
  - CreateTime
  - UpdateTime
  - CreatedBy
  - IsRegisteredWithLakeFormation
  - CatalogId
  - VersionId
6. 儲存資料表定義檔案。
7. 在命令提示字元中，執行下列 AWS CLI 命令，在新資料庫中重新建立資料表：

```
aws glue create-table --database-name new_database_name --table-input "file://~/Desktop/table_name.json"
```

如需有關 create-table 命令的詳細資訊，請參閱 [create-table](#)。

表格現在會顯示在中的新資料庫中，AWS Glue 並且可以從 Athena 查詢。

8. 重複步驟，以將每個其他資料表複製到 AWS Glue 中的新資料庫。

## 將 Athena 資料連接器用於外部 Hive 中繼存放區

您可以使用適用於外部 Hive 中繼存放區的 Amazon Athena 資料連接器，來查詢在 Simple Storage Service (Amazon S3) 中使用 Apache Hive 中繼存放區的資料集。不需要將中繼資料移轉至。AWS Glue Data Catalog 在 Athena 管理主控台中，您可以設定 Lambda 函數來與私有 VPC 中的 Hive 中繼存放區通訊，然後將其連接到該中繼存放區。Lambda 與您的 Hive 中繼存放區間的連線受到私有 Amazon VPC 頻道的保護，且不會使用公有網際網路。您可以提供自己的 Lambda 函數程式碼，也可以使用適用於外部 Hive 中繼存放區的 Athena 資料連接器預設實作。

### 主題

- [功能概觀](#)
- [工作流程](#)

- [考量與限制](#)
- [將 Athena 連接到 Apache Hive 中繼存放區](#)
- [使用部 AWS Serverless Application Repository 署 Hive 資料來源連接器](#)
- [使用現有 IAM 執行角色將 Athena 連接到 Hive 中繼存放區](#)
- [將 Athena 設定為使用已部署的 Hive 中繼存放區連接器](#)
- [在外部 Hive 中繼存放區查詢中使用預設資料來源](#)
- [使用 Hive 檢視](#)
- [使用 AWS CLI 與蜂巢中繼存儲](#)
- [參考實作](#)

## 功能概觀

您可以使用適用於外部 Hive 中繼存放區的 Athena 資料連接器執行下列任務：

- 使用 Athena 主控台註冊自訂目錄並使用它們執行查詢。
- 為不同的外部 Hive 中繼存放區定義 Lambda 函數，並在 Athena 查詢中加以聯結。
- 在相同的 Athena 查詢中使用 AWS Glue Data Catalog 和您的外部 Hive 中繼存放區。
- 在查詢執行內容中將目錄指定為目前預設目錄。如此就無需在查詢中將目錄名稱做為資料庫名稱的前綴。您可以使用 `database.table`，而不是用語法 `catalog.database.table`。
- 使用各種工具來執行參考外部 Hive 中繼存放區的查詢。您可以使用 Athena 主控台、AWS SDK、AWS CLI、Athena API，以及更新的 Athena JDBC 和 ODBC 驅動程式。更新的驅動程式具有自訂目錄的支援。

## API 支援

適用於外部 Hive 中繼存放區的 Athena 資料連接器，提供對目錄註冊 API 操作和中繼資料 API 操作的支援。

- 目錄註冊 – 註冊適用於外部 Hive 中繼存放區和[聯合資料來源](#)的自訂目錄。
- 中繼資料 — 使用中繼資料 API，為您向 Athena 註冊的任何目錄提供資料庫 AWS Glue 和表格資訊。
- Athena JAVA 軟體開發套件用戶端 – 在更新的 Athena Java 軟體開發套件用戶端中使用目錄註冊 API、中繼資料 API 以及支援 StartQueryExecution 操作中的目錄。

## 參考實作

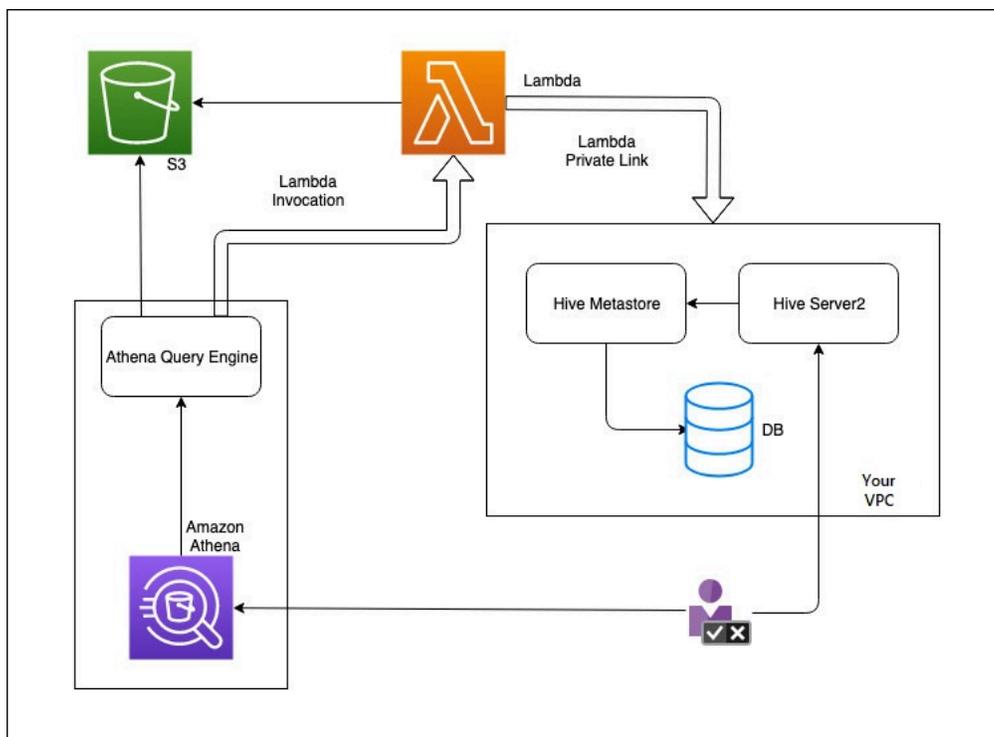
Athena 提供 Lambda 函數的參考實作，該函數會連接到外部 Hive 中繼存放區。在 [Athena Hive 中繼存放區 GitHub](#) 以開放原始碼專案的形式提供參考實作。

參考實施可作為 AWS Serverless Application Repository ( SAR ) 中的以下兩個 AWS SAM 應用程式使用。您可以在 SAR 中使用任一應用程式來建立自己的 Lambda 函數。

- **AthenaHiveMetastoreFunction** – Uber Lambda 功能 .jar 檔案。"uber" JAR (也稱為大量 JAR 或具有依存項目的 JAR) 是一種 .jar 檔案，該檔案會將 Java 程式和其依存項目包含在單一檔案中。
- **AthenaHiveMetastoreFunctionWithLayer** – Lambda 層和薄型 Lambda 函數 .jar 檔案。

## 工作流程

下列圖表說明 Athena 如何與您的外部 Hive 中繼存放區互動。



在此工作流程中，您連接資料庫的 Hive 中繼存放區位於 VPC 內。您使用 Hive Server2 透過 Hive CLI 來管理 Hive 中繼存放區。

從 Athena 使用外部 Hive 中繼存放區的工作流程包含下列步驟。

1. 您需要建立一個 Lambda 函數，以將 Athena 連接到 VPC 內部的 Hive 中繼存放區。

2. 您可以為 Hive 中繼存放區註冊唯一的目錄名稱，並在您的帳戶中註冊相對應的函數名稱。
3. 當您執行使用目錄名稱的 Athena DML 或 DDL 查詢時，Athena 查詢引擎會呼叫您與目錄名稱相關的 Lambda 函數名稱。
4. 使用 Lambda 函數與 VPC 中的外部 Hive 中繼存放區進行通訊 AWS PrivateLink，並接收中繼資料請求的回應。Athena 會使用來自外部 Hive 中繼存放區的中繼資料，就像它使用預設 AWS Glue Data Catalog 的中繼資料一樣。

## 考量與限制

當您使用適用於外部 Hive 中繼存放區的 Athena 資料連接器時，請考慮下列幾點：

- 您可以使用 CTAS 在外部 Hive 中繼存放區上建立資料表。
- 您可以使用 INSERT INTO 將資料插入外部 Hive 中繼存放區。
- 適用於外部 Hive 中繼存放區的 DDL 支援僅限於下列陳述式。
  - ALTER DATABASE SET DBPROPERTIES
  - ALTER TABLE ADD COLUMNS
  - ALTER TABLE ADD PARTITION
  - ALTER TABLE DROP PARTITION
  - ALTER TABLE RENAME PARTITION
  - 更改資料表的替換資料欄
  - ALTER TABLE SET LOCATION
  - ALTER TABLE SET TBLPROPERTIES
  - CREATE DATABASE
  - CREATE TABLE
  - CREATE TABLE AS
  - DESCRIBE TABLE
  - DROP DATABASE
  - DROP TABLE
  - SHOW COLUMNS
  - SHOW CREATE TABLE
  - SHOW PARTITIONS
  - SHOW SCHEMAS

- SHOW TABLES
- SHOW TBLPROPERTIES
- 您可以擁有的已註冊目錄數目上限為 1,000。
- Hive 中繼存放區不支援 Kerberos 身分驗證。
- 若要將 JDBC 驅動程式與外部 Hive 中繼存放區或[聯合查詢](#)搭配使用，請將 `MetadataRetrievalMethod=ProxyAPI` 包含在您的 JDBC 連接字串中。如需有關 JDBC 驅動程式的資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)。
- Hive 隱藏資料欄  
`$path`、`$bucket`、`$file_size`、`$file_modified_time`、`$partition`、`$row_id` 不能用於精細存取控制篩選。
- Hive 隱藏的系統資料表，例如 `example_table$properties` 或 `example_table$partitions` 不受精細存取控制支援。

## 許可

預先建置和自訂的資料連接器可能需要存取下列資源，才能正常運作。檢查您使用的連接器的資訊，以確保您已正確設定 VPC。如需有關在 Athena 中執行查詢和建立資料來源連接器所需 IAM 許可的資訊，請參閱[允許存取外部 Hive 中繼存放區的 Athena 資料連接器](#)和[允許 Lambda 函數存取外部 Hive 中繼存放區](#)。

- Simple Storage Service (Amazon S3) – 除了將查詢結果寫入 Simple Storage Service (Amazon S3) 中的 Athena 查詢結果位置外，資料連接器也會寫入 Simple Storage Service (Amazon S3) 中的溢出儲存貯體。對此 Simple Storage Service (Amazon S3) 位置，需有連線能力和許可。如需詳細資訊，請參閱本主題後面部分的[Amazon S3 中的溢出位置](#)。
- Athena – 需要存取權才能檢查查詢狀態，並防止溢出掃描。
- AWS Glue— 如果您的連接器用 AWS Glue 於補充或主要中繼資料，則需要存取權。
- AWS Key Management Service
- 政策 – Hive 中繼存放區、Athena Query Federation 和 UDF 還需要 [AWS 受管理的策略](#)：[AmazonAthenaFullAccess](#) 以外的政策。如需詳細資訊，請參閱[Athena 中的 Identity and Access Management](#)。

## Amazon S3 中的溢出位置

由於針對 Lambda 函數回應大小的[限制](#)，大於閾值的回應會溢入您在建立 Lambda 函數時指定的 Simple Storage Service (Amazon S3) 位置。Athena 會直接從 Simple Storage Service (Amazon S3) 讀取這些回應。

### Note

Athena 不會移除 Simple Storage Service (Amazon S3) 上的回應檔案。建議您設定保留原則，以自動刪除回應檔案。

## 將 Athena 連接到 Apache Hive 中繼存放區

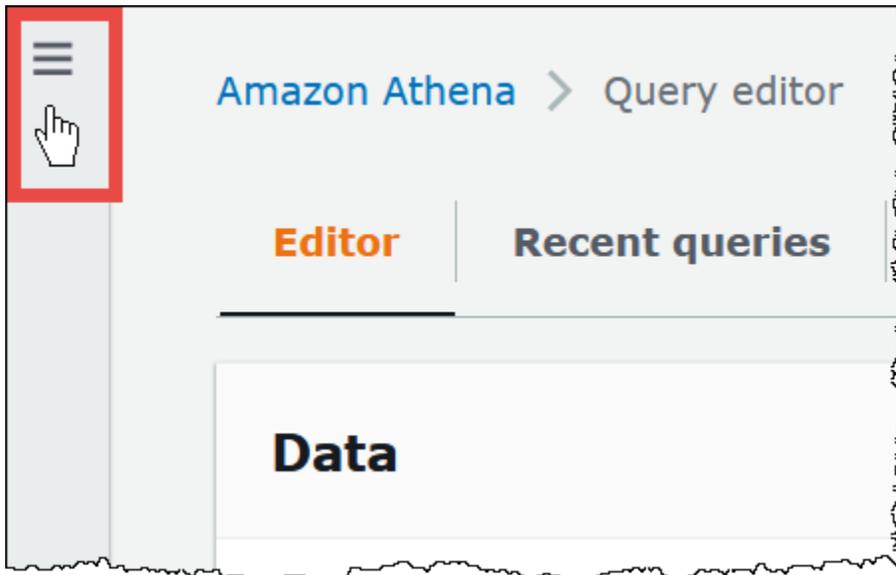
若要將 Athena 連接到 Apache Hive 中繼存放區，您必須建立和設定 Lambda 函數。關於基本實作，您可以從 Athena 管理主控台開始執行所有必要的步驟。

### Note

下列程序要求您具有為 Lambda 函數建立自訂 IAM 角色的許可。如果您沒有建立自訂角色的權限，可以使用 Athena [參考實作](#) 個別建立 Lambda 函數，然後使用 AWS Lambda 主控台為該函數選擇現有的 IAM 角色。如需詳細資訊，請參閱 [使用現有 IAM 執行角色將 Athena 連接到 Hive 中繼存放區](#)。

若要將 Athena 連接到 Hive 中繼存放區

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 選擇 Data Source (資料來源)。
4. 在主控台的右上角，選擇 Create data source (建立資料來源)。
5. 在 Choose a data source (選擇資料來源) 頁面上，針對 Data source (資料來源) 選擇 S3 - Apache Hive metastore (S3 - Apache Hive 中繼存放區)。
6. 選擇下一步。
7. 在 Data source details (資料來源詳細資訊) 區段中，當您從 Athena 查詢資料來源，請於 Data Source Name (資料來源名稱) 輸入要在 SQL 陳述式中使用的名稱。名稱最多可包含 127 個字元，且在您的帳戶中必須是唯一的。建立後便無法變更。有效字元包括 a-z、A-Z、0-9、\_ (底線)、@ (at 符號) 和 - (連字號)。Athena 已保留名稱 awsdatalog、hive、jmx 和 system，這些名稱不能用於資料來源名稱。
8. 對於 Lambda 函數，請選擇建立 Lambda 函數，然後選擇在中建立新的 Lambda 函數 AWS Lambda

AthenaHiveMetastoreFunction 頁面會在 AWS Lambda 主控台中開啟。此頁面包含連接器的詳細資訊。

Lambda > Functions > Create function > Review, configure and deploy

# AthenaHiveMetastoreFunction — version 1.0.1

Review, configure and deploy

 Copy as SAM Resource

## Application details

Author	Source code URL	Description	Report a vulnerability
default author	<a href="https://github.com/aws-labs/aws-athena-hive-metastore">https://github.com/aws-labs/aws-athena-hive-metastore</a>	An Athena Lambda function to interact with Hive Metastore	If you believe this application poses a security risk

## Readme file

Amazon Athena  
Hive Metastore  
Lambda Function

## Application settings

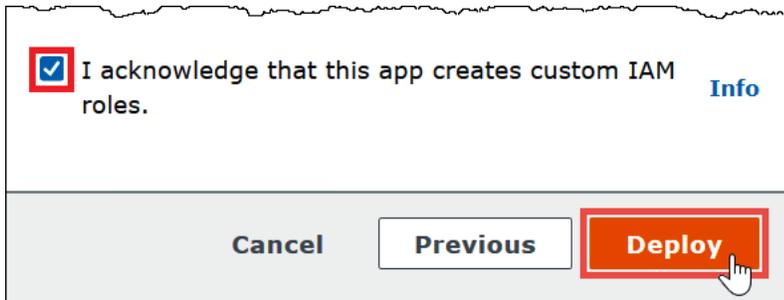
Application name  
The stack name of this application created via AWS CloudFormation

AthenaHiveMetastoreFunction

9. 在 Application settings (應用程式設定) 下，輸入您的 Lambda 函數的參數。

- LambdaFuncName— 提供函數的名稱。例如，myHiveMetastore.
- SpillLocation— 在此帳戶中指定 Amazon S3 位置，以便在 Lambda 函數回應大小超過 4 MB 時保留溢出的中繼資料。
- HMSUri – 輸入在連接埠 9083 上使用 Thrift 通訊協定的 Hive 中繼存放區主機的 URI。使用語法 `thrift://<host_name>:9083`。

- LambdaMemory— 指定介於 128 MB 到 3008 MB 之間的值。Lambda 函數分配到的 CPU 週期與您設定的記憶體數量成比例。預設值為 1024。
  - LambdaTimeout— 指定允許 Lambda 叫用的最大執行時間，以秒為單位，從 1 到 900 (900 秒為 15 分鐘)。預設值為 300 秒 (5 分鐘)。
  - VPC SecurityGroupIds — 輸入 Hive 中繼存放區的 VPC 安全性群組識別碼的逗號分隔清單。
  - VPC SubnetIds — 為 Hive 中繼存放區輸入以逗號分隔的 VPC 子網路識別碼清單。
10. 選擇 I acknowledge that this app creates custom IAM roles (我認可此應用程式建立自訂的 IAM 角色)，然後選擇 Deploy (部署)。



部署完成時，您的函數會顯示在 Lambda 應用程式清單中。現在 Hive 中繼存放區函數已部署到您的帳戶，您可以設定 Athena 以使用它。

11. 返回 Athena 主控台的 Enter data source details (輸入資料來源) 頁面。
12. 在 Lambda function (Lambda 函數) 區段中，選擇 Lambda 函數搜尋方塊旁的重新整理圖示。重新整理可用函數的清單可以使您新建立的函數出現在清單中。
13. 選擇您剛才在 Lambda 主控台中建立的函數名稱。隨即顯示 Lambda 函數的 ARN。
14. (選用) 在 Tags (標籤) 中，新增要與此資料來源相關聯的鍵值對。如需標籤的詳細資訊，請參閱[標記 Athena 資源](#)。
15. 選擇下一步。
16. 在 Review and create (檢閱並建立) 頁面上，檢閱資料來源詳細資訊，然後選擇 Create data source (建立資料來源)。
17. 該頁面的 Data source details (資料來源詳細資訊) 區段顯示了有關新連接器的資訊。

您現在可以使用您指定的 Data source name (資料來源名稱)，在 Athena 的 SQL 查詢中指定參考 Hive 中繼存放區。在 SQL 查詢中，使用下列範例中的語法，將 hms-catalog-1 取代為您先前指定的目錄名稱。

```
SELECT * FROM hms-catalog-1.CustomerData.customers
```

18. 如需有關查看、編輯或刪除您建立的資料來源的資訊，請參閱 [管理資料來源](#)。

## 使用部 AWS Serverless Application Repository 署 Hive 資料來源連接器

若要部署 Hive 的 Athena 資料來源連接器，您可以使用 [AWS Serverless Application Repository](#)，而無需從 Athena 主控台開始。使用 AWS Serverless Application Repository 尋找您要使用的連接器、提供連接器所需的參數，然後將連接器部署到您的帳戶。然後，部署連接器後，可以使用 Athena 主控台讓資料來源可供 Athena 使用。

使用將 Hive 的 AWS Serverless Application Repository 資料來源連接器部署到您的帳戶

1. 登入 AWS Management Console 並開啟無伺服器應用程式存放庫。
2. 選擇在導覽窗格中的 Available applications (可用的應用程式)。
3. 選取選項 Show apps that create custom IAM roles or resource policies (顯示建立自訂 IAM 角色或資源政策的應用程式)。
4. 在搜尋方塊中，輸入 **Hive**。出現的連接器包括以下兩個：
  - AthenaHiveMetastoreFunction – Uber Lambda 功能 .jar 檔案。
  - AthenaHiveMetastoreFunctionWithLayer-Lambda 層和瘦 Lambda 函數 .jar 文件。

這兩個應用程式具有相同的功能，唯一不同之處在其實作方式。您可以使用上述任一個應用程式來建立 Lambda 函數，將 Athena 連接到您的 Hive 中繼存放區。

5. 選擇您要使用的連接器名稱。本教學課程使用的是 AthenaHiveMetastoreFunction。

The screenshot shows the 'Serverless Application Repository' interface. On the left, there are navigation links for 'Available applications' (highlighted in orange) and 'Published applications'. The main area is titled 'Available applications' and is split into 'Public applications (1)' and 'Private applications'. A search bar contains the text 'AthenaHiveMetastoreFunction'. Below the search bar, there is a checked checkbox for 'Show apps that create custom IAM roles or resource policies' and a 'Sort by' dropdown menu set to 'Best Match'. A pagination indicator shows '1' of 1 results. The search result for 'AthenaHiveMetastoreFunction' is displayed, featuring a warning icon and the text 'Creates custom IAM roles or resource policies'. Below this, it says 'An Athena Lambda function to interact with Hive Metastore'. A blue button labeled 'athena-hive-metastore' is visible, along with the author 'default author' and '5 deployments'.

6. 在 Application settings (應用程式設定) 下，輸入您的 Lambda 函數的參數。

- LambdaFuncName— 提供函數的名稱。例如，myHiveMetastore。
- SpillLocation— 在此帳戶中指定 Amazon S3 位置，以便在 Lambda 函數回應大小超過 4 MB 時保留溢出的中繼資料。
- HMSUri – 輸入在連接埠 9083 上使用 Thrift 通訊協定的 Hive 中繼存放區主機的 URI。使用語法 `thrift://<host_name>:9083`。
- LambdaMemory— 指定介於 128 MB 到 3008 MB 之間的值。Lambda 函數分配到的 CPU 週期與您設定的記憶體數量成比例。預設值為 1024。
- LambdaTimeout— 指定允許 Lambda 叫用的最大執行時間，以秒為單位，從 1 到 900 (900 秒為 15 分鐘)。預設值為 300 秒 (5 分鐘)。
- VPC SecurityGroupIds — 輸入 Hive 中繼存放區的 VPC 安全性群組識別碼的逗號分隔清單。

- VPC SubnetIds — 為 Hive 中繼存放區輸入以逗號分隔的 VPC 子網路識別碼清單。
7. 在 Application details (應用程式詳細資訊) 頁面的右下方，選取 I acknowledge that this app creates custom IAM roles (我確認要以此應用程式建立自訂的 IAM 角色)，然後選擇 Deploy (部署)。

此時，您可以將 Athena 設定為使用 Lambda 函數連接到您的 Hive 中繼存放區。如需這些步驟，請參閱 [將 Athena 設定為使用已部署的 Hive 中繼存放區連接器](#)。

## 使用現有 IAM 執行角色將 Athena 連接到 Hive 中繼存放區

若要透過使用現有 IAM 角色的 Lambda 函數將您的外部 Hive 中繼存放區連接到 Athena，您可以使用適用於外部 Hive 中繼存放區的 Athena 連接器參考實作。

三個主要步驟如下：

1. [複製與建置](#) – 複製 Athena 參考實作，並建置包含 Lambda 函數程式碼的 JAR 檔案。
2. [AWS Lambda 主控台](#) — 在 AWS Lambda 主控台中建立 Lambda 函數，為其指派現有的 IAM 執行角色，然後上傳您產生的函數程式碼。
3. [Amazon Athena 主控台](#) – 在 Amazon Athena 主控台中建立資料來源名稱，您可以使用此名稱來參考 Athena 查詢中的外部 Hive 中繼存放區。

如果您已經擁有建立自訂 IAM 角色的許可，則可以使用更簡單的工作流程，該工作流程使用 Athena 主控台，AWS Serverless Application Repository 以及建立和設定 Lambda 函數。如需詳細資訊，請參閱 [將 Athena 連接到 Apache Hive 中繼存放區](#)。

### 必要條件

- Git 必須安裝在您的系統上。
- 您必須已安裝 [Apache Maven](#)。
- 您擁有可以指派給 Lambda 函數的 IAM 執行角色。如需詳細資訊，請參閱 [允許 Lambda 函數存取外部 Hive 中繼存放區](#)。

### 複製並建置 Lambda 函數

Athena 參考實現的函數代碼是一個位於 [aws aws-athena-hive-metastore labs/](#) 上 GitHub 的 Maven 項目。如需有關專案的詳細資訊，請參閱上的對應 README 檔案 GitHub 或本文件中的 [參考實作](#) 主題。

## 若要複製並建置 Lambda 函數程式碼

1. 請輸入下列命令來複製 Athena 參考實作：

```
git clone https://github.com/aws-labs/aws-athena-hive-metastore
```

2. 執行下列命令來建置適用於 Lambda 函數的 .jar 檔案：

```
mvn clean install
```

專案成功建置後，會將下列 .jar 檔案建立在您專案的目標資料夾中：

```
hms-lambda-func-1.0-SNAPSHOT-withdep.jar
```

在下一節中，您可以使用主 AWS Lambda 控制台將此檔案上傳到您的 Amazon Web Services 帳戶。

## 在 AWS Lambda 主控台中建立和設定 Lambda 函數

在本節中，您可以使用 AWS Lambda 主控台建立使用現有 IAM 執行角色的函數。設定函數的 VPC 後，您可以上傳函數程式碼並設定函數的環境變數。

### 建立 Lambda 函式

在此步驟中，您會在 AWS Lambda 主控台中建立使用現有 IAM 角色的函數。

### 若要建立使用現有 IAM 角色的 Lambda 函數

1. 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 <https://console.aws.amazon.com/lambda/>。
2. 在導覽視窗中，選擇函數。
3. 選擇 Create function (建立函數)。
4. 選擇從頭開始撰寫。
5. 在 Function name (函數名稱) 中，輸入您的 Lambda 函數名稱 (例如 **EHMSBasedLambda**)。
6. 在 Runtime (執行時間) 中，選擇 Java 8。
7. 在 Permissions (許可) 下，展開 Change default execution role (變更預設執行角色)。
8. 針對 Execution role (執行角色)，選擇 Use an existing role (使用現有的角色)。

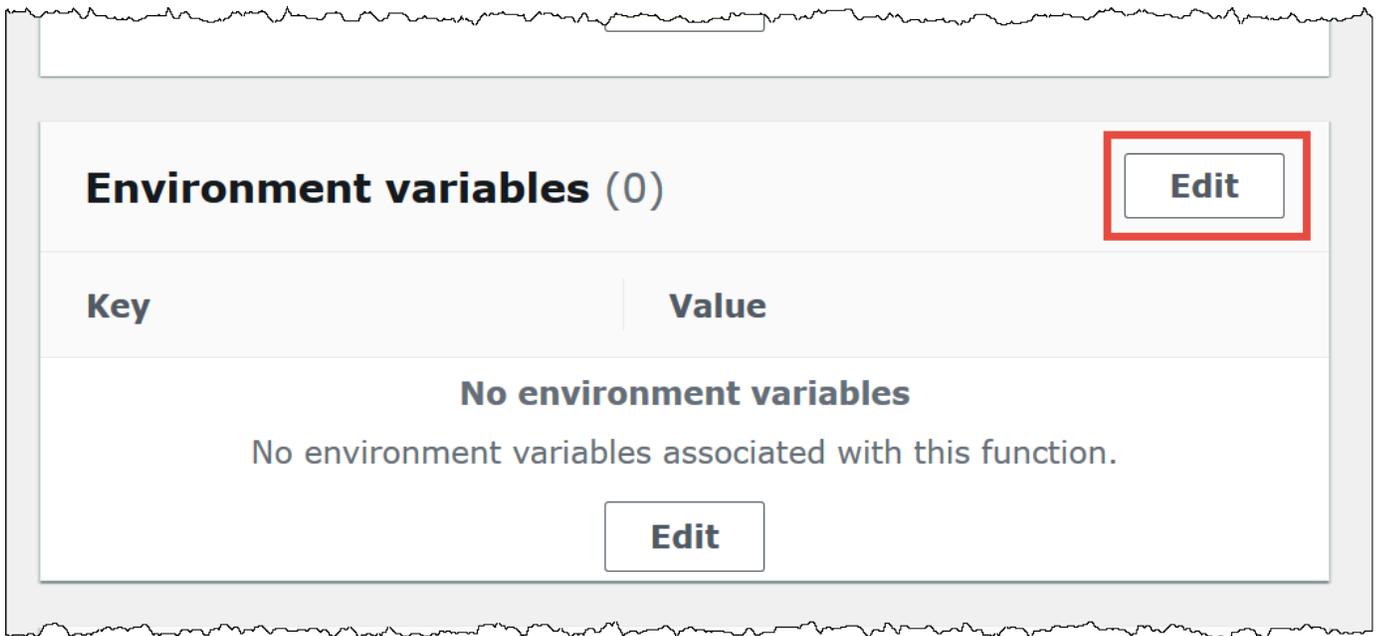
9. 針對 Existing role (現有角色)，選擇您的 Lambda 函數將用於 Athena 的 IAM 執行角色 (此範例使用名為 AthenaLambdaExecutionRole 的角色)。
10. 展開 Advanced settings (進階設定)。
11. 選取 Enable Network (啟用網路)。
12. 對於 VPC，選擇您的函數可以存取的 VPC。
13. 對於 Subnets (子網路)，選擇 Lambda 要使用的 VPC 子網路。
14. 對於 Security groups (安全群組)，選擇 Lambda 要使用的 VPC 安全群組。
15. 選擇建立函數。AWS Lambda 控制台並打開功能的配置頁面並開始創建函數。

### 上傳程式碼並設定 Lambda 函數

當主控台通知您已成功建立函數時，您便可以上傳函數程式碼並設定其環境變數。

若要上傳您的 Lambda 函數程式碼並設定其環境變數

1. 在 Lambda 主控台中，請確認您在指定功能之頁面的 Code (程式碼) 索引標籤上。
2. Code source (程式碼來源)：選擇 Upload from (上傳來源)，然後選擇 .zip or .jar file (.zip 或 .jar 檔案)。
3. 上傳您先前產生的 hms-lambda-func-1.0-SNAPSHOT-withdep.jar 檔案。
4. 在 Lambda 函數頁面上，選擇 Configuration (組態) 標籤。
5. 從左側的窗格中選擇 Environment variables (環境變數)。
6. 在 Environment variables (環境變數) 區段中，選擇 Edit (編輯)。



7. 在 Edit environment variables (編輯環境變數) 的頁面上，使用 Add environment variable (新增環境變數) 的選項新增下列環境變數索引鍵和值：

- HMS\_URIS – 以下列語法在連接埠 9083 上輸入使用 Thrift 通訊協定的 Hive 中繼存放區主機的 URI。

```
thrift://<host_name>:9083
```

- SPILL\_LOCATION – 指定您的 Amazon Web Services 帳戶中的 Amazon S3 位置，如果 Lambda 函數回應大小超過 4 MB，此位置會保留溢出的中繼資料。

Lambda > Functions > EHMSBasedLambda > Edit environment variables

## Edit environment variables

### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#) 

Key	Value	
<input type="text" value="HMS_URIS"/>	<input type="text"/>	<input type="button" value="Remove"/>
<input type="text" value="SPILL_LOCATION"/>	<input type="text"/>	<input type="button" value="Remove"/>

► **Encryption configuration**

#### 8. 選擇儲存。

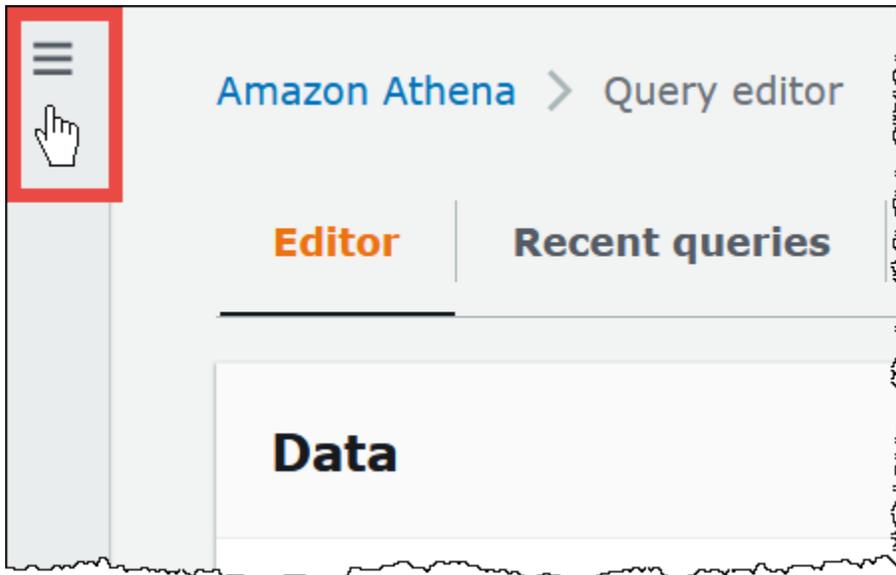
此時，您可以將 Athena 設定為使用 Lambda 函數連接到您的 Hive 中繼存放區。如需這些步驟，請參閱 [將 Athena 設定為使用已部署的 Hive 中繼存放區連接器](#)。

#### 將 Athena 設定為使用已部署的 Hive 中繼存放區連接器

將 Lambda 資料來源連接器 (例如 AthenaHiveMetastoreFunction) 部署到您的帳戶之後，您即可設定 Athena 以使用它。在這個步驟中，您需建立資料來源名稱，以便在參考外部 Hive 中繼存放區的 Athena 查詢中使用。

使用現有 Lambda 函數，將 Athena 連接至您的 Hive 中繼存放區

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 選擇 Data Source (資料來源)。
4. 在 Data sources (資料來源) 頁面上，選擇 Create data source (建立資料來源)。
5. 在 Choose a data source (選擇資料來源) 頁面上，針對 Data source (資料來源) 選擇 S3 - Apache Hive metastore (S3 - Apache Hive 中繼存放區)。
6. 選擇下一步。
7. 在 Data source details (資料來源詳細資訊) 區段中，當您從 Athena 查詢資料來源 (例如 MyHiveMetastore)，請於 Data Source Name (資料來源名稱) 輸入要在 SQL 陳述式中使用的名稱。名稱最多可包含 127 個字元，且在您的帳戶中必須是唯一的。建立後便無法變更。有效字元包括 a-z、A-Z、0-9、\_ (底線)、@ (at 符號) 和 - (連字號)。Athena 已保留名稱 awsdatalog、hive、jmx 和 system，這些名稱不能用於資料來源名稱。
8. 在 Connection details (連線詳細資訊) 區段中，使用 Select or enter a Lambda function (選取或輸入 Lambda 函數) 方塊，以選擇您剛才建立的函數名稱。隨即顯示 Lambda 函數的 ARN。
9. (選用) 在 Tags (標籤) 中，新增要與此資料來源相關聯的鍵值對。如需標籤的詳細資訊，請參閱 [標記 Athena 資源](#)。
10. 選擇下一步。
11. 在 Review and create (檢閱並建立) 頁面上，檢閱資料來源詳細資訊，然後選擇 Create data source (建立資料來源)。
12. 該頁面的 Data source details (資料來源詳細資訊) 區段顯示了有關新連接器的資訊。

您現在可以使用您指定的 Data source name (資料來源名稱)，在 Athena 的 SQL 查詢中指定參考 Hive 中繼存放區。

在您的 SQL 查詢中使用下列範例中的語法，將 ehms-catalog 取代為您先前指定的資料來源名稱。

```
SELECT * FROM ehms-catalog.CustomerData.customers
```

13. 若要檢視、編輯或刪除您建立的資料來源，請參閱[管理資料來源](#)。

## 在外部 Hive 中繼存放區查詢中使用預設資料來源

當您在外部 Hive 中繼存放區上執行 DML 和 DDL 查詢時，可以藉由省略目錄名稱 (如果查詢編輯器中已選取該名稱) 來簡化查詢語法。此功能有某些限制。

### DML 陳述式

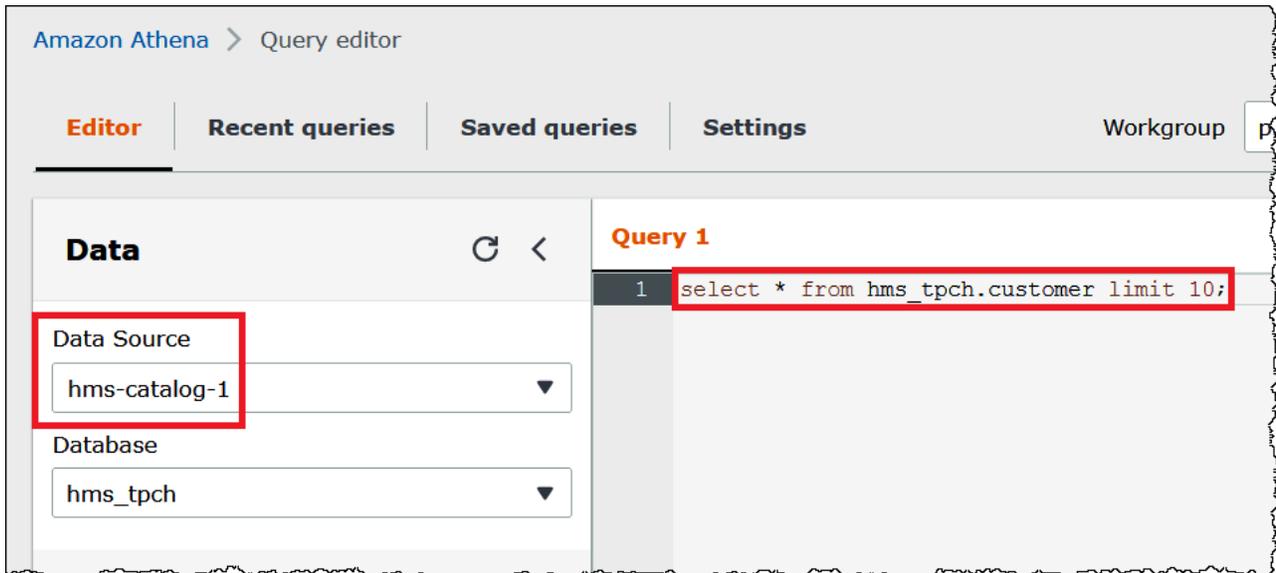
#### 使用已註冊的目錄執行查詢

1. 您可以在資料庫使用語法 `[[data_source_name].database_name].table_name` 之前放置資料來源，如下列範例所示。

```
select * from "hms-catalog-1".hms_tpch.customer limit 10;
```

2. 如果您想要使用的資料來源已在查詢編輯器中選取，則可以省略查詢中的名稱，如下列範例所示。

```
select * from hms_tpch.customer limit 10;
```



- 當您在查詢中使用多個資料來源時，只能省略預設資料來源名稱，並且必須為任何非預設資料來源指定完整名稱。

例如，假設已在查詢編輯器中選取 `AwsDataCatalog` 作為預設資料來源。下列查詢摘錄中的 FROM 陳述式完全限定前兩個資料來源名稱，但由於第三個資料來源位於資料目錄中，所以忽略第三個資料來源的名稱。AWS Glue

```
...
FROM ehms01.hms_tpch.customer,
      "hms-catalog-1".hms_tpch.orders,
      hms_tpch.lineitem
...
```

## DDL 陳述式

下列 Athena DDL 陳述式支援目錄名稱字首。其他 DDL 陳述式中的目錄名稱字首會導致語法錯誤。

```
SHOW TABLES [IN [catalog_name.]database_name] ['regular_expression']

SHOW TBLPROPERTIES [[catalog_name.]database_name.]table_name [('property_name')]

SHOW COLUMNS IN [[catalog_name.]database_name.]table_name

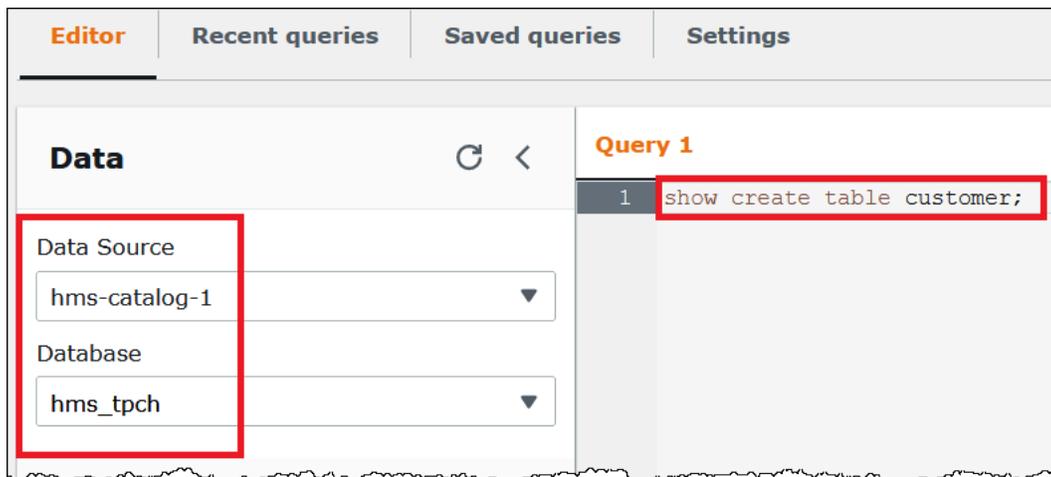
SHOW PARTITIONS [[catalog_name.]database_name.]table_name

SHOW CREATE TABLE [[catalog_name.][database_name.]table_name
```

```
DESCRIBE [EXTENDED | FORMATTED] [[catalog_name.][database_name.]table_name [PARTITION
partition_spec] [col_name ( [.field_name] | [.'$elem$'] | [.'$key$'] | [.'$value$'] )]
```

如同 DML 陳述式，若已在查詢編輯器中選取資料來源和資料庫，則可以在查詢編輯器中省略資料來源和資料庫字首。

於下圖可見，已在查詢編輯器中選取 `hms-catalog-1` 資料來源和 `hms_tpch` 資料庫。`show create table customer` 陳述式成立，即便查詢本身省略了 `hms-catalog-1` 字首和 `hms_tpch` 資料庫名稱。



在 JDBC 連接字串中指定預設資料來源

當使用 Athena JDBC 驅動程式將 Athena 連接到外部 Hive 中繼存放區時，您可以使用 `Catalog` 參數在 SQL 編輯器 (例如 [SQL Workbench](#)) 中的連接字串中指定預設資料來源名稱。

#### Note

如需下載最新的 Athena JDBC 驅動程式，請參閱 [透過 JDBC 驅動程式使用 Athena](#)。

下列連接字串會指定預設資料來源 `hms-catalog-name`。

```
jdbc:awsathena://AwsRegion=us-east-1;S3outputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results/;Workgroup=AmazonAthenaPreviewFunctionality;Catalog=hms-catalog-name;
```

下圖即為在 SQL Workbench 中設定的 JDBC 連接 URL 範例。

## 使用 Hive 檢視

您可以使用 Athena 來查詢在外部 Apache Hive 中繼存放區中的現有檢視。Athena 會 on-the-fly 在執行階段為您翻譯檢視，而不需變更原始檢視或儲存翻譯。

例如，假設您有一個 Hive 檢視 (如以下所示)，其使用了 Athena 不支援的語法，如 LATERAL VIEW explode()：

```
CREATE VIEW team_view AS
SELECT team, score
FROM matches
LATERAL VIEW explode(scores) m AS score
```

Athena 將 Hive 檢視查詢字串翻譯為 Athena 可以執行的以下陳述式：

```
SELECT team, score
```

```
FROM matches
CROSS JOIN UNNEST(scores) AS m (score)
```

如需將外部 Hive 中繼存放區連線到 Athena 的相關資訊，請參閱[將 Athena 資料連接器用於外部 Hive 中繼存放區](#)。

## 考量與限制

當從 Athena 查詢 Hive 檢視時，請考慮下列幾點：

- Athena 不支援建立 Hive 檢視。您可以在外部 Hive 中繼存放區中建立 Hive 檢視，然後您可以從 Athena 查詢該檢視。
- Athena 不支援 Hive 檢視的自訂 UDF。
- 由於 Athena 主控台中存在已知問題，Hive 檢視顯示在資料表清單下，而不是檢視清單下。
- 雖然翻譯過程是自動的，但某些 Hive 函數在 Hive 檢視中不受支援或需要特殊處理。如需詳細資訊，請參閱下一節。

## Hive 函數支援限制

本節重點介紹了 Athena 不支援 Hive 檢視或需要特殊處理的 Hive 函數。目前，由於 Athena 主要支援 Hive 2.2.0 中的函數，因此無法使用僅在更高版本 (如 Hive 4.0.0) 中可用的函數。如需 Hive 函數的完整清單，請參閱 [Hive language manual UDF](#) (Hive 語言手冊 UDF)。

## 彙總函數

### 需要特殊處理的彙總函數

Hive 檢視的以下彙總函數需要特殊處理。

- Avg – 使用 `avg(CAST(i AS DOUBLE))` 而不是 `avg(INT i)`。

### 不支援彙總函數

在 Athena 中，Hive 檢視不支援以下 Hive 彙總函數。

```
covar_pop
histogram_numeric
ntile
percentile
percentile_approx
```

在 Athena 中，Hive 檢視不支援迴歸函數，例如 `regr_count`、`regr_r2` 和 `regr_sxx`。

### 不支援日期函數

在 Athena 中，Hive 檢視不支援以下 Hive 日期函數。

```
date_format(date/timestamp/string ts, string fmt)
day(string date)
dayofmonth(date)
extract(field FROM source)
hour(string date)
minute(string date)
month(string date)
quarter(date/timestamp/string)
second(string date)
weekofyear(string date)
year(string date)
```

### 不支援遮罩函數

在 Athena 中，Hive 檢視不支援 Hive 遮罩函數，例如 `mask()` 和 `mask_first_n()`。

### 其他函數

#### 需要特殊處理的其他函數

Hive 檢視的以下其他函數需要特殊處理。

- `md5` – Athena 支援 `md5(binary)` 而非 `md5(varchar)`。
- `Explode` – 在以下語法中使用時，Athena 支援 `explode`：

```
LATERAL VIEW [OUTER] EXplode(<argument>)
```

- `Posexplode` – 在以下語法中使用時，Athena 支援 `posexplode`：

```
LATERAL VIEW [OUTER] POSEXplode(<argument>)
```

在 `(pos, val)` 輸出中，Athena 將 `pos` 資料欄視為 `BIGINT`。因此，您可能需要將 `pos` 資料欄轉換為 `BIGINT`，以避免檢視過時。以下範例說明此技術。

```
SELECT CAST(c AS BIGINT) AS c_bigint, d
FROM table LATERAL VIEW POSEXplode(<argument>) t AS c, d
```

## 不支援的其他函數

在 Athena 中，Hive 檢視不支援以下 Hive 函數。

```
aes_decrypt
aes_encrypt
current_database
current_user
inline
java_method
logged_in_user
reflect
sha/sha1/sha2
stack
version
```

## 運算子

### 需要特殊處理的運算子

Hive 檢視的以下運算子需要特殊處理。

- 求餘運算子 (%) – 因為 DOUBLE 類型隱含地轉換為 DECIMAL(x, y)，以下語法可能會導致 View is stale (檢視過時) 錯誤訊息：

```
a_double % 1.0 AS column
```

若要解決此問題，請使用 CAST，如下列範例所示。

```
CAST(a_double % 1.0 as DOUBLE) AS column
```

- 除法運算子 (/) – 在 Hive 中，int 除以 int 等於 double。在 Athena 中，相同的運算會產生一個遭截斷的 int。

### 不支援的運算子

Athena 不支援 Hive 檢視的以下運算子。

~A – 位元 NOT

A ^ b – 位元 XOR

A & b – 位元 AND

A | b – 位元 OR

A <=> b – 對於非 null 運算元，會傳回與等於 (=) 運算子相同的結果。如果兩個皆為 NULL，則會傳回 TRUE；如果其中一個是 NULL，則會傳回 FALSE。

字串函數

需要特殊處理的字串函數

Hive 檢視的以下 Hive 字串函數需要特殊處理。

- chr(bigint|double a) – Hive 允許負值引數；Athena 則不允許。
- instr(string str, string substr) – 因為 instr 函數的 Athena 映射會傳回 BIGINT 而不是 INT，請使用以下語法：

```
CAST(instr(string str, string substr) as INT)
```

如果沒有這個步驟，檢視將被視為過時。

- length(string a) – 因為 length 函數的 Athena 映射會傳回 BIGINT 而不是 INT，請使用以下語法，以使檢視不會被視為過時：

```
CAST(length(string str) as INT)
```

不支援字串函數

在 Athena 中，Hive 檢視不支援以下 Hive 字串函數。

```
ascii(string str)
character_length(string str)
decode(binary bin, string charset)
encode(string src, string charset)
elt(N int, str1 string, str2 string, str3 string, ...)
field(val T, val1 T, val2 T, val3 T, ...)
find_in_set(string str, string strList)
initcap(string A)
levenshtein(string A, string B)
locate(string substr, string str[, int pos])
octet_length(string str)
```

```
parse_url(string urlString, string partToExtract [, string keyToExtract])
printf(String format, Obj... args)
quote(String text)
regexp_extract(string subject, string pattern, int index)
repeat(string str, int n)
sentences(string str, string lang, string locale)
soundex(string A)
space(int n)
str_to_map(text[, delimiter1, delimiter2])
substring_index(string A, string delim, int count)
```

## 不支援 XPath 函數

在 Athena 中，Hive 檢視不支援 Hive XPath 函數，例如 `xpath`、`xpath_short` 和 `xpath_int`。

## 故障診斷

當您在 Athena 中使用 Hive 檢視時，您可能會遇到以下問題：

- 檢視 `<####>` 已過時 – 此訊息通常表示 Hive 和 Athena 中的檢視類型不符。如果 [Hive LanguageManual UDF](#) 和 [Presto 函數和操作](#) 符文檔中的相同函數具有不同的簽名，請嘗試轉換不匹配的數據類型。
- 未註冊函數 – Athena 目前不支援此函數。如需詳細資訊，請參閱本文件的前述資訊。

## 使用 AWS CLI 與蜂巢中繼存儲

您可以使用 `aws athena` CLI 命令來管理您搭配 Athena 使用的 Hive 中繼存放區資料目錄。定義一或多個搭配 Athena 使用的目錄之後，您可以在 `aws athena` DDL 和 DML 命令中參考這些目錄。

### 使用管理 Hive 中 AWS CLI 繼存放區目錄

註冊目錄：C `reate-data-catalog`

若要註冊資料目錄，請使用 `create-data-catalog` 命令。使用 `name` 參數指定要用於目錄的名稱。將 Lambda 函數的 ARN 傳遞給 `parameters` 引數的 `metadata-function` 選項。若要為新目錄建立標籤，請將 `tags` 參數與一或多個以空格分隔的 `Key=key, Value=value` 引數組搭配使用。

以下為註冊名為 `hms-catalog-1` 的 Hive 中繼存放區的範例。為了方便閱讀，命令已經過格式化處理。

```
$ aws athena create-data-catalog
  --name "hms-catalog-1"
```

```
--type "HIVE"  
--description "Hive Catalog 1"  
--parameters "metadata-function=arn:aws:lambda:us-  
east-1:111122223333:function:external-hms-service-v3, sdk-version=1.0"  
--tags Key=MyKey, Value=MyValue  
--region us-east-1
```

### 顯示目錄詳細資料:G et-data-catalog

若要顯示目錄的詳細資訊，請將目錄的名稱傳遞給 `get-data-catalog` 命令，如下列範例所示。

```
$ aws athena get-data-catalog --name "hms-catalog-1" --region us-east-1
```

以下樣本結果為 JSON 格式。

```
{  
  "DataCatalog": {  
    "Name": "hms-catalog-1",  
    "Description": "Hive Catalog 1",  
    "Type": "HIVE",  
    "Parameters": {  
      "metadata-function": "arn:aws:lambda:us-  
east-1:111122223333:function:external-hms-service-v3",  
      "sdk-version": "1.0"  
    }  
  }  
}
```

### 上市註冊目錄:L ist-data-catalogs

若要列示已註冊的目錄，請使用 `list-data-catalogs` 命令並選擇性地指定區域，如下列範例所示。列示的目錄永遠包括 AWS Glue 在內。

```
$ aws athena list-data-catalogs --region us-east-1
```

以下樣本結果為 JSON 格式。

```
{  
  "DataCatalogs": [  
    {  
      "CatalogName": "AwsDataCatalog",  
      "Type": "GLUE"  
    }  
  ]  
}
```

```
    },
    {
      "CatalogName": "hms-catalog-1",
      "Type": "HIVE",
      "Parameters": {
        "metadata-function": "arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-v3",
        "sdk-version": "1.0"
      }
    }
  ]
}
```

### 更新目錄 : U pdate-data-catalog

若要更新資料目錄，請使用 `update-data-catalog` 命令，如下列範例所示。為了方便閱讀，命令已經過格式化處理。

```
$ aws athena update-data-catalog
--name "hms-catalog-1"
--type "HIVE"
--description "My New Hive Catalog Description"
--parameters "metadata-function=arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-new,sdk-version=1.0"
--region us-east-1
```

### 刪除目錄 : D elete-data-catalog

若要刪除資料目錄，請使用 `delete-data-catalog` 命令，如下列範例所示。

```
$ aws athena delete-data-catalog --name "hms-catalog-1" --region us-east-1
```

### 顯示資料庫詳細資訊 : Get-database

若要顯示資料庫的詳細資訊，請將目錄和資料庫的名稱傳遞給 `get-database` 命令，如下列範例所示。

```
$ aws athena get-database --catalog-name hms-catalog-1 --database-name mydb
```

以下樣本結果為 JSON 格式。

```
{
```

```
"Database": {
  "Name": "mydb",
  "Description": "My database",
  "Parameters": {
    "CreatedBy": "Athena",
    "EXTERNAL": "TRUE"
  }
}
```

## 列出目錄中的資料庫：List-databases

若要列出目錄中的資料庫，請使用 `list-databases` 命令並選擇性地指定區域，如下列範例所示。

```
$ aws athena list-databases --catalog-name AwsDataCatalog --region us-west-2
```

以下樣本結果為 JSON 格式。

```
{
  "DatabaseList": [
    {
      "Name": "default"
    },
    {
      "Name": "mycrawlerdatabase"
    },
    {
      "Name": "mydatabase"
    },
    {
      "Name": "sampledb",
      "Description": "Sample database",
      "Parameters": {
        "CreatedBy": "Athena",
        "EXTERNAL": "TRUE"
      }
    },
    {
      "Name": "tpch100"
    }
  ]
}
```

## 顯示表格詳細資訊：Get-table-metadata

若要顯示資料表的中繼資料，包括資料行名稱和資料類型，請將目錄、資料庫的名稱和資料表名稱傳遞給 `get-table-metadata` 命令，如下列範例所示。

```
$ aws athena get-table-metadata --catalog-name AwsDataCatalog --database-name mydb --table-name cityuseragent
```

以下樣本結果為 JSON 格式。

```
{
  "TableMetadata": {
    "Name": "cityuseragent",
    "CreateTime": 1586451276.0,
    "LastAccessTime": 0.0,
    "TableType": "EXTERNAL_TABLE",
    "Columns": [
      {
        "Name": "city",
        "Type": "string"
      },
      {
        "Name": "useragent1",
        "Type": "string"
      }
    ],
    "PartitionKeys": [],
    "Parameters": {
      "COLUMN_STATS_ACCURATE": "false",
      "EXTERNAL": "TRUE",
      "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
      "last_modified_by": "hadoop",
      "last_modified_time": "1586454879",
      "location": "s3://DOC-EXAMPLE-BUCKET/",
      "numFiles": "1",
      "numRows": "-1",
      "outputformat":
"org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat",
      "rawDataSize": "-1",
      "serde.param.serialization.format": "1",
      "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
      "totalSize": "61"
    }
  }
}
```

```
    }  
  }  
}
```

顯示數據庫中所有表的元數據：List-table-metadata

若要顯示資料庫中所有資料表的中繼資料，請將目錄的名稱和資料庫名稱傳遞給 list-table-metadata 命令。list-table-metadata 命令與 get-table-metadata 命令類似，但是您不指定資料表名稱。若要限制結果的數目，您可以使用 --max-results 選項，如下列範例所示。

```
$ aws athena list-table-metadata --catalog-name AwsDataCatalog --database-name sampledb  
--region us-east-1 --max-results 2
```

以下樣本結果為 JSON 格式。

```
{  
  "TableMetadataList": [  
    {  
      "Name": "cityuseragent",  
      "CreateTime": 1586451276.0,  
      "LastAccessTime": 0.0,  
      "TableType": "EXTERNAL_TABLE",  
      "Columns": [  
        {  
          "Name": "city",  
          "Type": "string"  
        },  
        {  
          "Name": "useragent1",  
          "Type": "string"  
        }  
      ],  
      "PartitionKeys": [],  
      "Parameters": {  
        "COLUMN_STATS_ACCURATE": "false",  
        "EXTERNAL": "TRUE",  
        "inputformat": "org.apache.hadoop.mapred.TextInputFormat",  
        "last_modified_by": "hadoop",  
        "last_modified_time": "1586454879",  
        "location": "s3://DOC-EXAMPLE-BUCKET/",  
        "numFiles": "1",  
        "numRows": "-1",  
      }  
    }  
  ]  
}
```

```

        "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
        "rawDataSize": "-1",
        "serde.param.serialization.format": "1",
        "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "totalSize": "61"
    }
},
{
    "Name": "clearinghouse_data",
    "CreateTime": 1589255544.0,
    "LastAccessTime": 0.0,
    "TableType": "EXTERNAL_TABLE",
    "Columns": [
        {
            "Name": "location",
            "Type": "string"
        },
        {
            "Name": "stock_count",
            "Type": "int"
        },
        {
            "Name": "quantity_shipped",
            "Type": "int"
        }
    ],
    "PartitionKeys": [],
    "Parameters": {
        "EXTERNAL": "TRUE",
        "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
        "location": "s3://DOC-EXAMPLE-BUCKET/",
        "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
        "serde.param.serialization.format": "1",
        "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "transient_lastDdlTime": "1589255544"
    }
}
],
"NextToken":
"eyJzYXN0RXZhbHVhdGVkS2V5Ijpw7IkhBU0hfS0VZiJp7InMi0iJ0Ljk0YWZjYjk1MjJjNTQ1YmU4Y2I50WE5NTg0MjFjYjY"

```

```
}
```

## 執行 DDL 和 DML 陳述式

當您使用執行 DDL 和 DML 陳述式時，您可以使用下列兩種方式之一 AWS CLI 來傳遞 Hive 中繼儲存區目錄的名稱：

- 直接傳入支援它的陳述式。
- 傳遞給 `--query-execution-context Catalog` 參數。

## DDL 陳述式

下列範例會直接做為 `show create table` DDL 陳述式的一部分傳入目錄名稱。為了方便閱讀，命令已經過格式化處理。

```
$ aws athena start-query-execution
  --query-string "show create table hms-catalog-1.hms_tpch_partitioned.lineitem"
  --result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

下面的範例 DDL `show create table` 陳述式使用 `--query-execution-context` 的 `Catalog` 參數來傳遞 Hive 中繼存放區目錄名稱 `hms-catalog-1`。為了方便閱讀，命令已經過格式化處理。

```
$ aws athena start-query-execution
  --query-string "show create table lineitem"
  --query-execution-context "Catalog=hms-catalog-1,Database=hms_tpch_partitioned"
  --result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

## DML 陳述式

下列範例 DML `select` 陳述式將目錄名稱直接傳遞到查詢中。為了方便閱讀，命令已經過格式化處理。

```
$ aws athena start-query-execution
  --query-string "select * from hms-catalog-1.hms_tpch_partitioned.customer limit 100"
  --result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

下列範例 DML `select` 陳述式使用 `--query-execution-context` 的 `Catalog` 參數傳入 Hive 中繼存放區目錄名稱 `hms-catalog-1`。為了方便閱讀，命令已經過格式化處理。

```
$ aws athena start-query-execution
--query-string "select * from customer limit 100"
--query-execution-context "Catalog=hms-catalog-1,Database=hms_tpch_partitioned"
--result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

## 參考實作

Athena 在 .com 上為外部 Hive 中繼存放區提供其連接器的參考實作，GitHub 網址為 <https://github.com/aws-labs/aws-athena-hive-metastore>。

參考實作是具有以下模組的 [Apache Maven](#) 專案：

- **hms-service-api** – 包含 Lambda 函數和 Athena 服務用戶端之間的 API 操作。這些 API 操作定義於 HiveMetaStoreService 介面中。因為這是服務合約，所以您不應該變更此模組中的任何內容。
- **hms-lambda-handler** – 一組預設 Lambda 處理常式，用於處理所有 Hive 中繼存放區 API 呼叫。類別 MetadataHandler 是所有 API 呼叫的發送器。您不需要變更此套件。
- **hms-lambda-func** – 具有下列元件的範例 Lambda 函數。
  - **HiveMetaStoreLambdaFunc** – Lambda 函數擴展 MetadataHandler 的範例。
  - **ThriftHiveMetaStoreClient** – 與 Hive 中繼存放區通訊的 Thrift 用戶端。此用戶端是為 Hive 2.3.0 編寫的。如果您使用不同的 Hive 版本，可能需要更新這個類別，以確保回應物件是相容的。
  - **ThriftHiveMetaStoreClientFactory** – 控制 Lambda 函數的行為。例如，您可以透過覆寫 getHandlerProvider() 方法來提供自己的一組處理常式提供者。
- **hms.properties** – 設定 Lambda 函數。大多數情況下只需要更新下列兩個屬性。
  - **hive.metastore.uris** – Hive 中繼存放區的 URI，格式為 `thrift://<host_name>:9083`。
  - **hive.metastore.response.spill.location**：當回應物件的大小超過指定的閾值 (例如 4 MB) 時，儲存回應物件的 Amazon S3 位置。閾值定義於 `hive.metastore.response.spill.threshold` 屬性中。不建議變更預設值。

### Note

這兩個屬性可由 [Lambda 環境變數](#) HMS\_URIS 和 SPILL\_LOCATION 覆寫。當您想要使用具有不同 Hive 中繼存放區或溢出位置的函數時，請使用這些變數，而不要重新編譯 Lambda 函數的來源程式碼。

- **hms-lambda-layer** – 將 `hms-service-api`、`hms-lambda-handler` 和其依存項目放入 `.zip` 檔案中的 Maven 組合專案。`.zip` 檔案會註冊為 Lambda 層，以供多個 Lambda 函數使用。
- **hms-lambda-rnp** – 記錄來自 Lambda 函數的回應，然後使用其重播回應。您可以使用此模型來模擬 Lambda 回應以進行測試。

## 自行建置成品

大多數使用案例並不需要修改參考實作。不過，如有必要，您可以修改來源程式碼，自行建置成品，然後將它們上傳到 Amazon S3 位置。

在建置成品之前，請先更新 `hms-lambda-func` 模組中 `hive.metastore.response.spill.location` 檔案內的屬性 `hive.metastore.uris` 和 `hms.properties`。

要建置成品，您必須安裝 Apache Maven 並執行命令 `mvn install`。如此會在模組 `hms-lambda-layer` 中名為 `target` 的輸出資料夾中產生層 `.zip` 檔案，以及在 `hms-lambda-func` 模組中產生 Lambda 函數 `.jar` 檔案。

## 使用 Amazon Athena 聯合查詢

如果您有資料位於 Simple Storage Service (Amazon S3) 以外的來源中，您可以使用 Athena 聯合查詢就地查詢資料，或建置管道以從多個資料來源擷取資料並將它們存放在 Simple Storage Service (Amazon S3) 中。Athena 聯合查詢可讓您針對存放在關聯式、非關聯式、物件和自訂資料來源中的資料執行 SQL 查詢。

Athena 使用執行的資料來源連接器 AWS Lambda 來執行聯合查詢。資料來源連接器是可在目標資料來源和 Athena 之間進行轉譯的一段程式碼。您可以將連接器視為 Athena 查詢引擎的延伸。預先建立的 Athena 資料來源連接器可用於像是 Amazon CloudWatch 日誌、Amazon 動態 B、亞馬遜 RDS 和亞馬遜 RDS 等資料來源，以及符合 JDBC 規範的關聯式資料來源 (例如 MySQL) 和 Apache 2.0 授權下的 PostgreSQL。您也可以使用 Athena Query Federation 軟體開發套件來撰寫自訂連接器。若要選擇、設定並將資料來源連接器部署到您的帳戶，您可以使用 Athena 和 Lambda 主控台或 AWS Serverless Application Repository。部署資料來源連接器之後，連接器會與一個目錄產生關聯，而您可以在 SQL 查詢中指定此目錄。您可以結合來自多個目錄的 SQL 陳述式，並在單一查詢中跨越多個資料來源。

對資料來源提交查詢時，Athena 會叫用對應的連接器，以識別需要讀取的資料表部分，並會管理平行處理和下推篩選條件述詞。根據提交查詢的使用者，連接器可能允許或禁止存取特定的資料元素。連接器使用 Apache Arrow 作為格式，以傳回查詢中要求的資料，因此能夠以 C、C++、Java、Python

和 Rust 等語言實作連接器。連接器是在 Lambda 中進行處理，因此可以從雲端或內部部署中可供 Lambda 存取的任何資料來源存取資料。

若要撰寫您的資料來源連接器，您可以使用 Athena Query Federation 軟體開發套件，自訂 Amazon Athena 提供和維護的其中一個預先建置的連接器。您可以從[GitHub 存放庫](#)修改原始程式碼的副本，然後使用[連接器發行工具](#)建立您自己的 AWS Serverless Application Repository 套件。

### Note

第三方開發人員可能已經使用 Athena Query Federation 軟體開發套件來撰寫資料來源連接器。如對這些資料來源連接器的支援或授權有疑問，請洽詢您的連接器供應商。這些連接器並未受到測試或支援 AWS。

如需 Athena 撰寫和測試的資料來源連接器之清單，請參閱[可用資料來源連接器](#)。

如需撰寫自己資料來源連接器的相關資訊，請參閱上的[範例 Athena 連接器](#) GitHub。

## 考量與限制

- 引擎版本 – 只有 Athena 引擎版本 2 及更新版本才支援 Athena 聯合查詢。如需有關 Athena 引擎版本的資訊，請參閱[Athena 引擎版本控制](#)。
- 檢視 – 您可以在聯合資料來源上建立和查詢檢視。聯合檢視會儲存在中 AWS Glue，而非基礎資料來源。如需詳細資訊，請參閱[查詢聯合檢視](#)。
- 寫入操作 – 不支援 [INSERT INTO](#) 等寫入操作。嘗試執行這項操作可能會產生錯誤訊息：外部目錄目前不支援此操作。
- 定價 – 如需定價資訊，請參閱 [Amazon Athena 定價](#)。

JDBC 驅動程式 – 若要將 JDBC 驅動程式與聯合查詢或[外部 Hive 中繼存放區](#)搭配使用，請在您的 JDBC 連線字串中包含 `MetadataRetrievalMethod=ProxyAPI`。如需有關 JDBC 驅動程式的資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

- Secrets Manager – 若要搭配 AWS Secrets Manager 使用 Athena 聯合查詢功能，您必須為 Secrets Manager 設定 Amazon VPC 私有端點。如需詳細資訊，請參閱《AWS Secrets Manager 使用者指南》中的[建立 Secrets Manager VPC 私有端點](#)。

資料來源連接器可能需要存取下列資源，才能正常運作。如果您使用預先建置的連接器，請檢查連接器的資訊，以確保您已正確設定 VPC。此外，請確定執行查詢和建立連接器的 IAM 委託人具有必要動作的權限。如需詳細資訊，請參閱[允許 Athena 聯合查詢的 IAM 許可政策範例](#)。

- Simple Storage Service (Amazon S3) – 除了將查詢結果寫入 Simple Storage Service (Amazon S3) 中的 Athena 查詢結果位置外，資料連接器也會寫入 Simple Storage Service (Amazon S3) 中的溢出儲存貯體。對此 Simple Storage Service (Amazon S3) 位置，需有連線能力和許可。
- Athena – 資料來源需要連線到 Athena (反之亦然)，以檢查查詢狀態和防止溢出掃描。
- AWS Glue Data Catalog – 如果您的連接器將資料目錄用於補充或主要中繼資料，則需要連線和許可。

## 影片

請觀看下列影片，進一步了解如何使用 Athena 聯合查詢。

影片：分析 Amazon 雅典娜在亞馬遜的聯合查詢結果 QuickSight

以下影片示範如何在 Amazon QuickSight 分析 Athena 聯合查詢的結果。

### [分析 Amazon Athena 的聯合查詢結果 QuickSight](#)

影片：遊戲分析管道

下列影片說明如何使用 Amazon Athena 聯合查詢來部署可擴展的無伺服器資料管道，以擷取、存放並分析遊戲和服務中的遙測資料。

### [遊戲分析管道](#)

## 可用資料來源連接器

本節列出預先建置的 Athena 資料來源連接器，供您用來查詢 Simple Storage Service (Amazon S3) 外部的各種資料來源。若要在 Athena 查詢中使用連接器，請設定連接器並將其部署到您的帳戶。

### 考量與限制

- 某些預先建置的連接器要求您先建立 VPC 和安全群組，然後才能使用連接器。如需建立 VPC 的相關資訊，請參閱 [為資料來源連接器建立 VPC](#)。
- 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，您必須為 Secrets Manager 設定 Amazon VPC 私有端點。如需詳細資訊，請參閱《AWS Secrets Manager 使用者指南》中的 [建立 Secrets Manager VPC 私有端點](#)。
- 對於不支援述詞下推的連接器，包含述詞的查詢需要更長的時間才能執行。對於小型資料集，只會掃描極少的資料，而查詢平均需要大約 2 分鐘。但是，對於大型資料集，許多查詢可能會逾時。
- 有些聯合資料來源使用術語來描述與 Athena 不同的資料物件。如需詳細資訊，請參閱 [Athena 和聯合資料表名稱限定詞](#)。

- 針對列出資料表時不支援分頁的連接器，如果您的資料庫有許多資料表和中繼資料，Web 服務可能會逾時。下列連接器提供所列資料表的分頁支援：
  - DocumentDB
  - DynamoDB
  - MySQL
  - OpenSearch
  - Oracle
  - PostgreSQL
  - Redshift
  - SQL Server

## 其他資訊

- 如需有關部署 Athena 資料來源連接器的資訊，請參閱[部署資料來源連接器](#)。
- 如需有關使用 Athena 資料來源連接器查詢的資訊，請參閱[執行聯合查詢](#)。
- 如需 Athena 資料來源連接器的深入資訊，請參閱上的[可用連接器](#) GitHub。

## Athena 資料來源連接器

- [Amazon Athena Azure Data Lake Storage \(ADLS\) Gen2 連接器](#)
- [Amazon Athena Azure Synapse 連接器](#)
- [Amazon Athena Cloudera Hive 連接器](#)
- [Amazon Athena Cloudera Impala 連接器](#)
- [Amazon Athena CloudWatch 連](#)
- [Amazon Athena CloudWatch 指標連](#)
- [Amazon Athena AWS CMDB 連接器](#)
- [Amazon Athena IBM Db2 連接器](#)
- [Amazon Athena IBM Db2 AS/400 \(Db2 系列\) 連接器](#)
- [Amazon Athena DocumentDB 連接器](#)
- [Amazon Athena DynamoDB 連接器](#)
- [Amazon Athena 谷歌 BigQuery 連](#)
- [Amazon Athena Google Cloud Storage 連接器](#)

- [Amazon Athena HBase 連接器](#)
- [Amazon Athena Hortonworks 連接器](#)
- [Amazon Athena Apache Kafka 連接器](#)
- [Amazon Athena MSK 連接器](#)
- [Amazon Athena MySQL 連接器](#)
- [Amazon Athena Neptune 連接器](#)
- [Amazon Athena OpenSearch 連](#)
- [Amazon Athena Oracle 連接器](#)
- [Amazon Athena PostgreSQL 連接器](#)
- [Amazon Athena Redis 連接器](#)
- [Amazon Athena Redshift 連接器](#)
- [Amazon Athena SAP HANA 連接器](#)
- [Amazon Athena Snowflake 連接器](#)
- [Amazon Athena Microsoft SQL Server 連接器](#)
- [Amazon Athena Teradata 連接器](#)
- [Amazon Athena Timestream 連接器](#)
- [Amazon Athena TPC benchmark DS \(TPC-DS\) 連接器](#)
- [Amazon Athena Vertica 連接器](#)

 Note

[AthenaJdbcConnector](#) (最新版本 2022.4.1) 已被棄用。相反地，請使用資料庫特定連接器，例如適用於 [MySQL](#)、[Redshift](#) 或 [PostgreSQL](#) 的連接器。

## Amazon Athena Azure Data Lake Storage (ADLS) Gen2 連接器

適用於 [Azure Data Lake Storage \(ADLS\) Gen2](#) 的 Amazon Athena 連接器可讓 Amazon Athena 能夠對存放在 ADLS 上的資料執行 SQL 查詢。Athena 無法直接存取資料湖中儲存的檔案。

- 工作流程 – 連接器實作 JDBC 介面，該介面使用 `com.microsoft.sqlserver.jdbc.SQLServerDriver` 驅動程式。連接器會將查詢傳遞至 Azure Synapse 引擎，然後其會存取資料湖。

- 資料處理和 S3 – 一般而言，Lambda 連接器會直接查詢資料，而無需傳輸至 Amazon S3。但是，當 Lambda 函數傳回的資料超過 Lambda 限制時，資料會寫入您指定的 Amazon S3 溢出儲存貯體，以便 Athena 可以讀取多餘的資料。
- AAD 身分驗證 – AAD 可用作 Azure Synapse 連接器的身分驗證方法。若要使用 AAD，連接器使用的 JDBC 連線字串必須包含 URL 參數 `authentication=ActiveDirectoryServicePrincipal`、`AADSecurePrincipalId` 和 `AADSecurePrincipalSecret`。這些參數可以直接傳遞，也可以由 Secrets Manager 傳遞。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 篩選條件中的日期和時間戳記資料類型必須轉換為適當的資料類型。

## 條款

下列術語與 Azure Data Lake Storage Gen2 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。

- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是`connection_string`屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Azure Data Lake Storage Gen2 連接器。

## 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
datalakegentwo://${jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	DataLakeGen2MuxCompositeHandler
中繼資料處理常式	DataLakeGen2MuxMetadataHandler
記錄處理常式	DataLakeGen2MuxRecordHandler

## 多工處理常式參數

參數	描述
<code>\$catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mydatalakegentwocatalog</code> ，則環境變數名稱為 <code>mydatalakegentwocatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 DataLakeGen 2 MUX Lambda 函數：datalakegentwo1(預設值) 和datalakegentwo2。

屬性	Value
default	datalakegentwo://jdbc:sqlserver://adlsgentwo1. . . <i>hostname:port</i> ;databaseName= <i>database_name</i> ; \${ <i>secret1_name</i> }
datalakegentwo_cat alog1_connection_s tring	datalakegentwo://jdbc:sqlserver://adlsgentwo1. . . <i>hostname:port</i> ;databaseName= <i>database_name</i> ; \${ <i>secret1_name</i> }
datalakegentwo_cat alog2_connection_s tring	datalakegentwo://jdbc:sqlserver://adlsgentwo2. . . <i>hostname:port</i> ;databaseName= <i>database_name</i> ; \${ <i>secret2_name</i> }

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用指南》AWS Secrets Manager中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入秘密名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

### 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${secret1_name}`。

```
datalakegentwo://jdbc:sqlserver://hostname:port;databaseName=database_name;  
${secret1_name}
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
datalakegentwo://  
jdbc:sqlserver://  
hostname:port;databaseName=database_name;user=user_name;password=password
```

### 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Azure Data Lake Storage Gen2 執行個體。

處理常式類型	類別
複合處理常式	DataLakeGen2CompositeHandler
中繼資料處理常式	DataLakeGen2MetadataHandler
記錄處理常式	DataLakeGen2RecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 Azure Data Lake Storage Gen2 執行個體。

屬性	Value
default	<code>datalakegentwo://jdbc:sqlserver:// <i>hostname:port</i>;database Name=;\${ <i>secret_name</i> }</code>

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 ADLS Gen2 和 Arrow 的相應資料類型。

ADLS Gen2	Arrow
bit	TINYINT
tinyint	SMALLINT
smallint	SMALLINT

ADLS Gen2	Arrow
int	INT
bigint	BIGINT
decimal	DECIMAL
numeric	FLOAT8
smallmoney	FLOAT8
money	DECIMAL
float[24]	FLOAT4
float[53]	FLOAT8
real	FLOAT4
datetime	Date(MILLISECOND)
datetime2	Date(MILLISECOND)
smalldatetime	Date(MILLISECOND)
date	Date(DAY)
time	VARCHAR
datetimeoffset	Date(MILLISECOND)
char[n]	VARCHAR
varchar[n/max]	VARCHAR

## 分割區和分隔

Azure Data Lake Storage Gen2 使用與 Hadoop 相容的 Gen2 Blob 儲存體來儲存資料檔案。從 Azure Synapse 引擎中查詢這些檔案的資料。Azure Synapse 引擎會將儲存在檔案系統中的 Gen2 資料視為

外部資料表。根據資料類型實作分割區。如果已在 Gen 2 儲存系統中分割和分配資料，則連接器會以單一分割方式擷取資料。

## 效能

當一次執行多個查詢時，Azure Data Lake Storage Gen2 連接器會顯示較慢的查詢效能，且會受到限流的影響。

Athena Azure Data Lake Storage Gen2 連接器會執行述詞下推，以減少查詢掃描的資料。簡單述詞和複雜表達式會下推至連接器，以減少掃描的資料量並減少查詢執行時間。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Azure Data Lake Storage Gen2 連接器可以合併這些表達式，並將其直接推送至 Azure Data Lake Storage Gen2，以增強功能並減少掃描的資料量。

下列 Athena Azure Data Lake Storage Gen2 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## 直通查詢

Azure 資料湖儲存體 Gen2 連接器支援[傳遞](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要搭配 Azure 資料湖儲存體 Gen2 使用傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

下列範例查詢會將查詢向下推送至 Azure 資料湖儲存體 Gen2 中的資料來源。查詢會選取資料 `customer` 表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

### 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

### 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上 GitHub Azure 資料湖儲存體第二代連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

### Amazon Athena Azure Synapse 連接器

適用於 [Azure Synapse 分析](#) 的 Amazon Athena 連接器可讓 Amazon Athena 使用 JDBC 對 Azure Synapse 資料庫執行 SQL 查詢。

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 限制

- 不支援寫入 DDL 操作。

- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 在篩選條件中，您必須將 Date 和 Timestamp 資料類型轉換為適當的資料類型。
- 若要搜尋類型 Real 和 Float 的負值，請使用 <= 或者 >= 運算子。
- 不支援 binary、varbinary、image、和 rowversion 資料類型。

## 條款

下列術語與 Synapse 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是connection\_string屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Synapse 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
synapse://${jdbc_connection_string}
```

### 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	SynapseMuxCompositeHandler
中繼資料處理常式	SynapseMuxMetadataHandler
記錄處理常式	SynapseMuxRecordHandler

## 多工處理常式參數

參數	描述
<code>\$catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mysynapse catalog</code> ，則環境變數名稱為 <code>mysynapsecatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Synapse MUX Lambda 函數：synapse1 (預設) 和 synapse2。

屬性	Value
<code>default</code>	<code>synapse://jdbc:synapse://synapse1.hostname:port;databaseName= &lt;database_name&gt; ;\${secret1_name }</code>
<code>synapse_catalog1_connection_string</code>	<code>synapse://jdbc:synapse://synapse1.hostname:port;databaseName= &lt;database_name&gt; ;\${secret1_name }</code>
<code>synapse_catalog2_connection_string</code>	<code>synapse://jdbc:synapse://synapse2.hostname:port;databaseName= &lt;database_name&gt; ;\${secret2_name }</code>

屬性	Value
connection_string	

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代之為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

## 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${secret_name}`。

```
synapse://jdbc:synapse://hostname:port;databaseName=<database_name>;${secret_name}
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
synapse://jdbc:synapse://
hostname:port;databaseName=<database_name>;user=<user>;password=<password>
```

## 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Synapse 執行個體。

處理常式類型	類別
複合處理常式	SynapseCompositeHandler
中繼資料處理常式	SynapseMetadataHandler
記錄處理常式	SynapseRecordHandler

## 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 Synapse 執行個體。

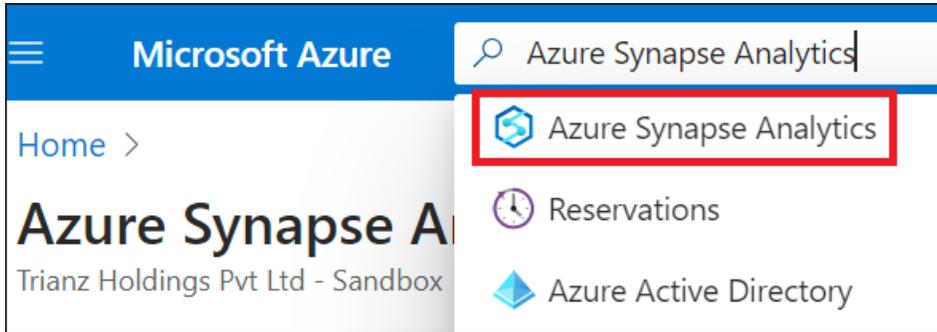
屬性	Value
default	synapse://jdbc:sqlserver://hostname:port;databaseName=<database_name>;\${secret_name }

## 設定 Active Directory 身分驗證

Amazon Athena Azure Synapse 連接器支援 Microsoft Active Directory 身分驗證。在開始之前，您必須在 Microsoft Azure 入口網站中設定系統管理使用者，然後使用 AWS Secrets Manager 來建立密碼。

## 若要設定 Active Directory 管理使用者

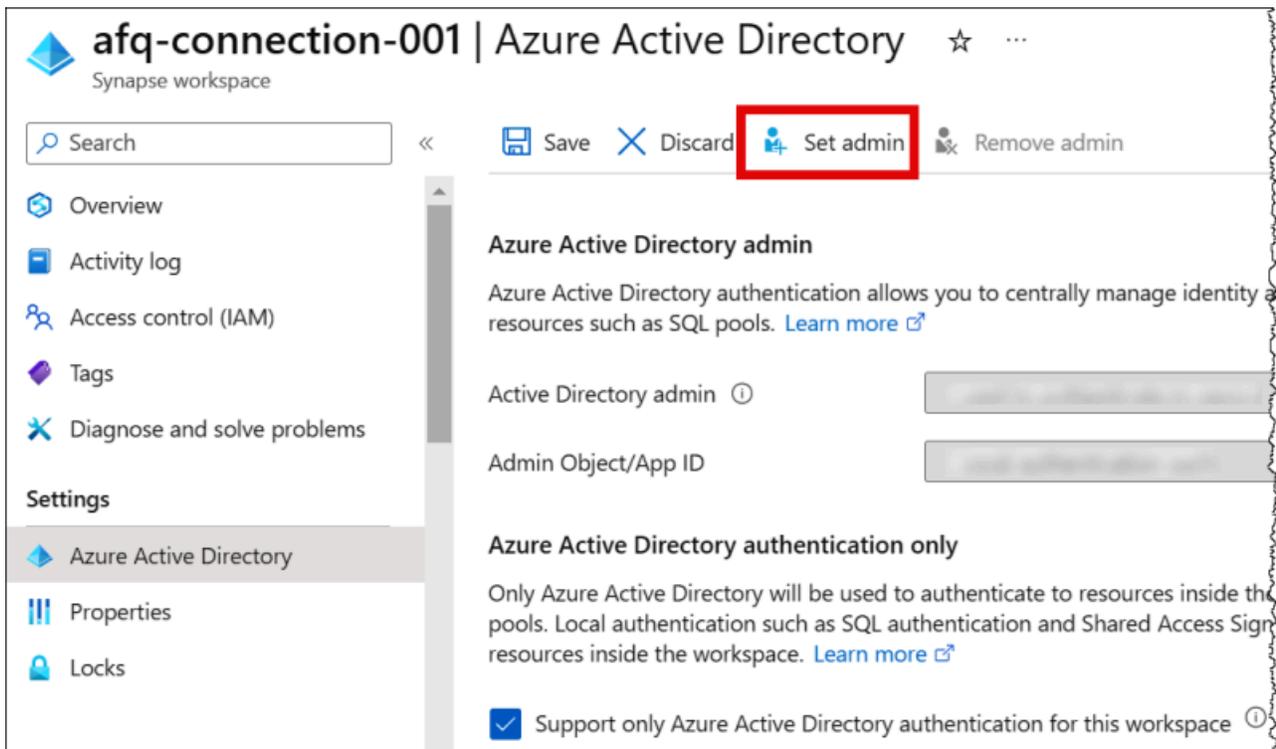
1. 使用具有管理權限的帳戶，登入 Microsoft Azure 入口網站 <https://portal.azure.com/>。
2. 在搜尋方塊中，輸入 Azure Synapse Analytics，然後選擇 Azure Synapse Analytics。



3. 開啟左側的選單。



4. 在導覽窗格中，依序選擇 Azure Active Directory。
5. 在設定管理員索引標籤上，將 Active Directory 管理員設定為新的或現有的使用者。



6. 在中 AWS Secrets Manager，儲存管理員使用者名稱和密碼認證。如需有關在 Secrets Manager 中建立祕密的詳細資訊，請參閱[建立 AWS Secrets Manager 祕密](#)。

若要檢視 Secrets Manager 中的祕密

1. 前往以下位置開啟機密管理員控制台：<https://console.aws.amazon.com/secretsmanager/>。
2. 在導覽窗格中，選擇 Secrets (祕密)。
3. 在 Secrets (祕密) 頁面中，選擇祕密的連結。
4. 在祕密的詳細資訊頁面上，選擇 Retrieve secret value (擷取祕密值)。

Key/value	Plaintext
Secret key	Secret value
username	
password	

## 修改連線字串

若要啟用連接器的 Active Directory 身分驗證，請使用下列語法修改連線字串：

```
synapse://jdbc:synapse://
hostname:port;databaseName=database_name;authentication=ActiveDirectoryPassword;
{secret_name}
```

## 使用 ActiveDirectoryServicePrincipal

Amazon Athena Azure Synapse 連接器也支援 ActiveDirectoryServicePrincipal。若要啟用此功能，請依照下列方式修改連線字串。

```
synapse://jdbc:synapse://
hostname:port;databaseName=database_name;authentication=ActiveDirectoryServicePrincipal;
{secret_name}
```

對於 secret\_name，指定應用程式或用戶端 ID 做為密碼中服務主體識別的使用者名稱和秘密。

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 Synapse 和 Apache Arrow 的相應資料類型。

Synapse	Arrow
bit	TINYINT
tinyint	SMALLINT
smallint	SMALLINT
int	INT
bigint	BIGINT
decimal	DECIMAL
numeric	FLOAT8
smallmoney	FLOAT8
money	DECIMAL
float[24]	FLOAT4
float[53]	FLOAT8
real	FLOAT4
datetime	Date(MILLISECOND)
datetime2	Date(MILLISECOND)
smalldatetime	Date(MILLISECOND)
date	Date(DAY)
time	VARCHAR
datetimeoffset	Date(MILLISECOND)
char[n]	VARCHAR
varchar[n/max]	VARCHAR

Synapse	Arrow
nchar[n]	VARCHAR
nvarchar[n/max]	VARCHAR

## 分割區和分隔

分割區由 `varchar` 類型的單一分割區資料欄表示。Synapse 支援範圍分割，因此透過從 Synapse 中繼資料表中擷取分割區資料欄和分割區範圍來執行分割。這些範圍值用於建立分割。

## 效能

選取資料欄子集會大幅拖慢查詢執行時間。連接器會顯示因為並行而導致的嚴重限流。

Athena Synapse 連接器執行述詞下推，以減少查詢掃描的資料。簡單述詞和複雜表達式會下推至連接器，以減少掃描的資料量並減少查詢執行時間。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Synapse 連接器可以合併這些表達式，並將其直接推送至 Synapse，以增強功能並減少掃描的資料量。

下列 Athena Synapse 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
```

```
AND ((col_a + col_b) > (col_c % col_d))
AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## 直通查詢

突觸連接器支持[直通](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

要使用 Synapse 的直通查詢，您可以使用以下語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

以下示例查詢將查詢向下推送到 Synapse 中的數據源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

- 有關如何使用 Amazon QuickSight 和亞馬 Amazon Athena 聯合查詢對存儲在 Microsoft Azure Synapse 數據庫中的數據構建儀表板和可視化的文章，請參閱大數據博客中的[使用 Amazon QuickSight](#)，[亞馬遜雅典娜聯合查詢](#)和 [Microsoft Azure 突觸執行多雲分析](#)。AWS
- 如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上的 Synapse 連接器 [pom.xml](#) 檔案。GitHub
- 如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena Cloudera Hive 連接器

適用於 Cloudera Hive 的 Amazon Athena 連接器可讓 Athena 能夠對 [Cloudera Hive](#) Hadoop 分發執行 SQL 查詢。連接器可將您的 Athena SQL 查詢轉換為其同等的 HiveQL 語法。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。
- 您必須先設定 VPC 和安全群組，才能使用此連接器。如需詳細資訊，請參閱 [為資料來源連接器建立 VPC](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。

## 條款

下列術語與 Cloudera Hive 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是connection\_string屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Cloudera Hive 連接器。

## 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
hive://${jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	HiveMuxCompositeHandler
中繼資料處理常式	HiveMuxMetadataHandler
記錄處理常式	HiveMuxRecordHandler

## 多工處理常式參數

參數	描述
<code>\$catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>myhivecatalog</code> ，則環境變數名稱為 <code>myhivecatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Hive MUX Lambda 函數：hive1 (預設) 和 hive2。

屬性	Value
<code>default</code>	<code>hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}</code>

屬性	Value
hive2_catalog1_connection_string	hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}
hive2_catalog2_connection_string	hive://jdbc:hive2://hive2:10000/default?UID=sample&PWD=sample

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入秘密名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

## 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 \${Test/RDS/hive1}。

```
hive://jdbc:hive2://hive1:10000/default?...&${Test/RDS/hive1}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
hive://jdbc:hive2://hive1:10000/default?...&UID=sample2&PWD=sample2&...
```

目前，Cloudera Hive 連接器可辨識 UID 和 PWD JDBC 屬性。

### 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Cloudera Hive 執行個體。

處理常式類型	類別
複合處理常式	HiveCompositeHandler
中繼資料處理常式	HiveMetadataHandler
記錄處理常式	HiveRecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 Cloudera Hive 執行個體。

屬性	Value
預設	hive://jdbc:hive2://hive1:10000/default?secret=\${Test/RDS/hive1}

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC、Cloudera Hive 和 Arrow 的相應資料類型。

JDBC	Cloudera Hive	Arrow
Boolean	Boolean	位元
Integer	TINYINT	Tiny
Short	SMALLINT	Smallint
Integer	INT	Int
Long	BIGINT	Bigint
float	float4	Float4
Double	float8	Float8
日期	日期	DateDay
時間戳記	timestamp	DateMilli

JDBC	Cloudera Hive	Arrow
字串	VARCHAR	Varchar
位元組	位元組	Varbinary
BigDecimal	Decimal (小數)	Decimal (小數)
ARRAY	N/A (請參閱備註)	清單

### Note

目前，Cloudera Hive 不支援彙總類型 ARRAY、MAP、STRUCT 或 UNIONTYPE。彙總類型的資料欄在 SQL 中被視為 VARCHAR 資料欄。

## 分割區和分隔

分割區用於決定如何產生連接器的分割。Athena 建構了類型 varchar 的合成資料欄，表示資料表的分割結構，以幫助連接器產生分割。連接器不會修改實際的資料表定義。

## 效能

Cloudera Hive 支援靜態分割區。Athena Cloudera Hive 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大的資料集，強烈建議使用靜態分割。Cloudera Hive 連接器對由於並行而導致的限流保有彈性。

Athena Cloudera Hive 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Cloudera Hive 連接器可以結合這些表達式，並將它們直接推送到 Cloudera Hive，以增強功能並減少掃描的資料量。

下列 Athena Cloudera Hive 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

### 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

### 直通查詢

Cloudera 蜂巢連接器支援[直通查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

要使用 Cloudera Hive 的直通查詢，您可以使用以下語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 Cloudera Hive 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上 GitHub 雲端硬碟連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena Cloudera Impala 連接器

Amazon Athena Cloudera 黑斑羚連接器使 Athena 能夠在 [Cloudera](#) 黑斑羚分佈上運行 SQL 查詢。連接器會將您的 Athena SQL 查詢轉換為同等的黑斑羚語法。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。
- 您必須先設定 VPC 和安全群組，才能使用此連接器。如需詳細資訊，請參閱 [為資料來源連接器建立 VPC](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。

## 條款

下列術語與 Cloudera Impala 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。

- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Cloudera Impala 連接器。

## 連接字串

使用下列格式的 JDBC 連接字串來連接 Impala 叢集。

```
impala://${jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	ImpalaMuxCompositeHandler
中繼資料處理常式	ImpalaMuxMetadataHandler
記錄處理常式	ImpalaMuxRecordHandler

## 多工處理常式參數

參數	描述
<code>\$catalog_connection_string</code>	必要。Athena 目錄的 Impala 叢集連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的

參數	描述
	目錄為 <code>myimpalacatalog</code> ，則環境變數名稱為 <code>myimpalacatalog_connection_string</code> 。
default	必要。預設的连接字串。目錄為 <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Impala MUX Lambda 函數：impala1 (預設) 和 impala2。

屬性	Value
default	<code>impala://jdbc:impala://some.impala.host.name:21050/?\${Test/impala1}</code>
<code>impala_catalog1_connection_string</code>	<code>impala://jdbc:impala://someother.impala.host.name:21050/?\${Test/impala1}</code>
<code>impala_catalog2_connection_string</code>	<code>impala://jdbc:impala://another.impala.host.name:21050/?UID=sample&amp;PWD=sample</code>

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

### 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${Test/impala1host}`。

```
impala://jdbc:impala://Impala1host:21050/?...&${Test/impala1host}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
impala://jdbc:impala://Impala1host:21050/?...&UID=sample2&PWD=sample2&...
```

目前，Cloudera Impala 可辨識 UID 和 PWD JDBC 屬性。

### 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Cloudera Impala 執行個體。

處理常式類型	類別
複合處理常式	ImpalaCompositeHandler
中繼資料處理常式	ImpalaMetadataHandler
記錄處理常式	ImpalaRecordHandler

## 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 Cloudera Impala 執行個體。

屬性	Value
default	impala://jdbc:impala://Impala1host:21050/?secret=\${Test/impala1host}

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC、Cloudera Impala 和 Arrow 的相應資料類型。

JDBC	Cloudera Impala	Arrow
Boolean	Boolean	位元
Integer	TINYINT	Tiny
Short	SMALLINT	Smallint
Integer	INT	Int
Long	BIGINT	Bigint
float	float4	Float4
Double	float8	Float8
日期	日期	DateDay
時間戳記	timestamp	DateMilli
字串	VARCHAR	Varchar
位元組	位元組	Varbinary
BigDecimal	Decimal (小數)	Decimal (小數)
ARRAY	N/A (請參閱備註)	清單

### Note

目前，Cloudera Impala 不支援彙總類型 ARRAY、MAP、STRUCT 或 UNIONTYPE。彙總類型的資料欄在 SQL 中被視為 VARCHAR 資料欄。

## 分割區和分隔

分割區用於決定如何產生連接器的分割。Athena 建構了類型 varchar 的合成資料欄，表示資料表的分割結構，以幫助連接器產生分割。連接器不會修改實際的資料表定義。

## 效能

Cloudera Impala 支援靜態分割區。Athena Cloudera Impala 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大的資料集，強烈建議使用靜態分割。Cloudera Impala 連接器由於並行而對限流保有彈性。

Athena Cloudera Impala 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。

### LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

### 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Cloudera Impala 連接器可以結合這些表達式，並將它們直接推送到 Cloudera Impala，以增強功能並減少掃描的資料量。

下列 Athena Cloudera Impala 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

### 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## 直通查詢

Cloudera 黑斑羚連接器支持 [直通](#) 查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

要使用 Cloudera 黑斑羚直通查詢，您可以使用以下語法：

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

下面的示例查詢向下推一個查詢到 Cloudera 黑斑羚的數據源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上 Cloudera 黑斑羚連接器的 [pom.xml](#) 檔案。  
GitHub

如需有關此連接器的其他資訊，請造訪 GitHub .com 上 [的對應網站](#)。

## Amazon Athena CloudWatch 連

Amazon Athena CloudWatch 連接器可讓 Amazon Athena 進行通訊，以 CloudWatch 便您可以使用 SQL 查詢日誌資料。

連接器會將您的資料架構對映 LogGroups 為資料架構，並 LogStream 將每個連接 連接器也會對映包含 LogStreams 中所有內容的特殊all\_log\_streams視圖 LogGroup。此檢視可讓您一次查詢所有記錄檔，而不是 LogStream 逐一搜尋每個記錄。 LogGroup

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 參數

使用本節中的 Lambda 環境變數來設定 CloudWatch 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- spill\_put\_request\_headers - (選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"})。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- kms\_key\_id - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 a7e63k4b-8loc-40db-a2a1-4d0en2cd8331，您可以指定 KMS 金鑰 ID。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。

此連接器也支援 [AIMD 擁塞控制](#)，以處理 CloudWatch 透過 [Amazon Athena 查詢聯合](#) 開發套件建構的節流事件。ThrottlingInvoker 您可以透過設定下列任何選用的環境變數來調整預設限流行為：

- throttle\_initial\_delay\_ms - 在第一個擁塞事件之後套用的初始呼叫延遲。預設值為 10 毫秒。
- throttle\_max\_delay\_ms - 呼叫之間的最大延遲。您可以透過將其分成 1000 毫秒來衍生 TPS。預設值為 1000 毫秒。
- throttle\_decrease\_factor - Athena 降低呼叫速率的係數。預設值為 0.5
- throttle\_increase\_ms — Athena 降低呼叫延遲的速率。預設值為 10 毫秒。

## 資料庫和資料表

Athena CloudWatch 連接器會將您 LogGroups 的結構描述 (也就是資料庫) 對應 LogStream 為資料表。連接器也會對映包含 LogStreams 中所有內容的特殊 `all_log_streams` 視圖 LogGroup。此檢視可讓您一次查詢所有記錄檔，而不是 LogStream 逐一搜尋每個記錄。 LogGroup

Athena CloudWatch 連接器對應的每個表格都具有下列結構描述。此結構描述與記 CloudWatch 錄檔提供的欄位相符。

- `log_stream` - VARCHAR 個包含該行所在 LogStream 的名稱。
- `time` - 一個 INT64，它包含產生日誌行時的 epoch 時間。
- `message` - 包含日誌消息的 VARCHAR。

## 範例

下列範例顯示如何對指定的執行 SELECT 查詢 LogStream。

```
SELECT *
FROM "lambda:cloudwatch_connector_lambda_name".log_group_path".log_stream_name"
LIMIT 100
```

下面的例子演示了如何使用 `all_log_streams` 視圖來對所有 LogStreams 在指定的執行查詢 LogGroup。

```
SELECT *
FROM "lambda:cloudwatch_connector_lambda_name".log_group_path".all_log_streams"
LIMIT 100
```

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-cloudwatch.yaml](#) 檔案的 Policies 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- CloudWatch 防護記錄讀取/寫入 — 連接器會使用此權限來讀取您的防護記錄資料，並寫入其診斷記錄。

## 效能

Athena CloudWatch 連接器會平行處理查 CloudWatch 詢所需的記錄串流掃描，嘗試將查詢最佳化。對於特定時間週期篩選器，述詞下推會在 Lambda 函數和日誌內 CloudWatch 執行。

為了獲得最佳效能，請僅使用小寫作為日誌群組名稱和日誌串流名稱。使用混合大小寫會導致連接器執行運算密集程度較高的不區分大小寫搜尋。

## 直通查詢

CloudWatch 連接器支援使用[CloudWatch 記錄見解查詢語法的傳遞查詢](#)。如需有關 CloudWatch 日誌洞見的詳細資訊，請參閱 [Amazon CloudWatch 日誌使用者指南中的使用日誌洞察分析 CloudWatch 日誌資料](#)。

若要使用建立傳遞查詢 CloudWatch，請使用下列語法：

```
SELECT * FROM TABLE(  
    system.query(  
        STARTTIME => 'start_time',  
        ENDTIME => 'end_time',  
        QUERYSTRING => 'query_string',  
        LOGGROUPNAMES => 'log_group-names',  
        LIMIT => 'max_number_of_results'  
    ))
```

下列範例 CloudWatch 傳遞查詢會在duration欄位不等於 1000 時篩選該欄位。

```
SELECT * FROM TABLE(  
    system.query(  
        STARTTIME => '1710918615308',  
        ENDTIME => '1710918615972',  
        QUERYSTRING => 'fields @duration | filter @duration != 1000',  
        LOGGROUPNAMES => '/aws/lambda/cloudwatch-test-1',  
        LIMIT => '2'  
    ))
```

## 授權資訊

Amazon Athena CloudWatch 連接器專案是根據[阿帕奇-2.0](#) 授權進行授權。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena CloudWatch 指標連

Amazon Athena CloudWatch 指標連接器可讓 Amazon Athena 使用 SQL 查詢 CloudWatch 指標資料。

如需將查詢量度 CloudWatch 從 Athena 本身發佈至的相關資訊，請參閱[使用指標和事件控制成本並 CloudWatch 監控查詢](#)。

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 參數

使用本節中的 Lambda 環境變數來設定指 CloudWatch 標連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- spill\_put\_request\_headers - (選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"})。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- kms\_key\_id - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 a7e63k4b-8loc-40db-a2a1-4d0en2cd8331，您可以指定 KMS 金鑰 ID。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。

此連接器也支援 [AIMD 擁塞控制](#)，以處理 CloudWatch 透過 [Amazon Athena 查詢聯合](#) 開發套件建構的節流事件。ThrottlingInvoker 您可以透過設定下列任何選用的環境變數來調整預設限流行為：

- throttle\_initial\_delay\_ms - 在第一個擁塞事件之後套用的初始呼叫延遲。預設值為 10 毫秒。
- throttle\_max\_delay\_ms - 呼叫之間的最大延遲。您可以透過將其分成 1000 毫秒來衍生 TPS。預設值為 1000 毫秒。

- `throttle_decrease_factor` - Athena 降低呼叫速率的係數。預設值為 0.5
- `throttle_increase_ms`— Athena 降低呼叫延遲的速率。預設值為 10 毫秒。

## 資料庫和資料表

Athena CloudWatch Metric 連接器會將您的命名空間、維度、量度和量度值對應到名為 `default` 的單一結構描述中的兩個表格。

## 指標資料表

`metrics` 資料表包含由命名空間、集合和名稱組合唯一定義的可用指標。`metrics` 資料表包含以下資料欄。

- `namespace` - 包含命名空間的 VARCHAR。
- `metric_name` - 包含指標名稱的 VARCHAR。
- `dimensions` - STRUCT 物件的 LIST，由 `dim_name` (VARCHAR) 和 `dim_value` (VARCHAR) 組成。
- `statistic` - VARCHAR 統計資料的 LIST (例如，`p90`、`AVERAGE ...`)，可供指標使用。

## `metric_samples` 資料表

`metric_samples` 資料表包含 `metrics` 資料表中每個指標的可用指標範例。`metric_samples` 資料表包含以下資料欄。

- `namespace` - 包含命名空間的 VARCHAR。
- `metric_name` - 包含指標名稱的 VARCHAR。
- `dimensions` - STRUCT 物件的 LIST，由 `dim_name` (VARCHAR) 和 `dim_value` (VARCHAR) 組成。
- `dim_name`— 一個 VARCHAR 方便欄位，可用於輕鬆篩選單個維度名稱。
- `dim_value`— 一個 VARCHAR 方便欄位，可用於輕鬆篩選單個維度值。
- `period` - 一個表示指標「時段」的 INT 欄位 (以秒為單位) (例如，60 秒指標)。
- `timestamp` - 表示指標範例所用 epoch 時間的 BIGINT 欄位 (以秒為單位)。
- `value` - 包含範例值的 FLOAT8 欄位。
- `statistic` - 包含範例統計類型的 VARCHAR (例如，`AVERAGE` 或者 `p90`)。

## 所需的許可

如需此連接器所需 IAM 政策的完整詳細資訊，請檢閱 [athena-cloudwatch-metrics.yaml](#) 檔案的 Policies 章節。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- CloudWatch 度量 ReadOnly — 連接器會使用此權限來查詢您的指標資料。
- CloudWatch 記錄寫入 — 連接器會使用此存取權來寫入其診斷記錄。

## 效能

Athena CloudWatch Metric 連接器會平行處理查詢所需的記錄串流掃描，嘗試將 CloudWatch 量度的查詢最佳化。對於特定期間、量度、命名空間和維度篩選器，述詞下推會同時在 Lambda 函數和日誌內 CloudWatch 執行。

## 授權資訊

Amazon Athena CloudWatch 指標連接器專案是根據 [Apache-2.0](#) 授權進行授權。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena AWS CMDB 連接器

Amazon Athena AWS CMDB 連接器可讓 Athena 與各種 AWS 服務進行通訊，以便您可以使用 SQL 進行查詢。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 參數

您可以使用本節中的 Lambda 環境變數來設定 AWS CMDB 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。

- `spill_prefix` - (選用) 預設為指定的 `spill_bucket` 中名為 `athena-federation-spill` 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- `spill_put_request_headers` - (選用) 用於溢出的 Amazon S3 `putObject` 請求的請求標頭和值的 JSON 編碼映射 (例如，`{"x-amz-server-side-encryption" : "AES256"}`)。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- `kms_key_id` - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`，您可以指定 KMS 金鑰 ID。
- `disable_spill_encryption` - (選用) 當設定為 `True` 時，停用溢出加密。預設為 `False`，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。
- `default_ec2_image_owner` - (選用) 設定後，控制篩選 [Amazon Machine Image \(AMI\)](#) 的預設 Amazon EC2 映像擁有者。如果您未設定此值，且您針對 EC2 映像資料表的查詢不包含擁有者的篩選條件，則您的結果將包含所有公有映像。

## 資料庫和資料表

Athena AWS CMMDB 連接器會提供下列資料庫和表格，以便查詢您的 AWS 資源清查。如需有關每個資料表中可用資料欄的詳細資訊，請使用 Athena 主控台或 API 執行 `DESCRIBE database.table` 陳述式。

- `ec2` - 此資料庫包含與 Amazon EC2 相關的資源，包括下列資源。
  - `ebs_volumes` - 包含您的 Amazon EBS 磁碟區的詳細資訊。
  - `ec2_instances` - 包含您的 EC2 執行個體的詳細資訊。
  - `ec2_images` - 包含您的 EC2 執行個體映像的詳細資訊。
  - `routing_tables` - 包含您的 VPC 路由表的詳細資訊。
  - `security_groups` - 包含安全性群組的詳細資訊。
  - `subnets` - 包含 VPC 子網的詳細資訊。
  - `vpcs` - 包含 VPC 的詳細資訊。
- `emr` - 此資料庫包含與 Amazon EMR 相關的資源，包括下列資源。
  - `emr_clusters` - 包含 EMR 叢集的詳細資訊。

- rds - 此資料庫包含與 Amazon RDS 相關的資源，包括下列資源。
- rds\_instances - 包含您的 RDS 執行個體的詳細資訊。
- s3 - 此資料庫包含與 RDS 相關的資源，包括下列資源。
- buckets - 包含您的 Amazon S3 儲存貯體的詳細資訊。
- objects - 包含 Amazon S3 物件的詳細資訊，但不包含其內容。

## 所需的許可

如需此連接器所需 IAM 政策的完整詳細資訊，請檢閱 [athena-aws-cmdb.yaml](#) 檔案的 Policies 章節。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- S3 List - 連接器使用此許可列出您的 Amazon S3 儲存貯體和物件。
- EC2 Describe - 連接器使用此權限來描述資源，例如您的 Amazon EC2 執行個體、安全群組、VPC 和 Amazon EBS 磁碟區。
- EMR Describe / List - 連接器使用此權限來描述您的 EMR 叢集。
- RDS Describe - 連接器使用此許可來描述您的 RDS 執行個體。

## 效能

目前，Athena AWS CMDB 連接器不支援 parallel 掃描。在 Lambda 函數內執行述詞下推。如果可能，部分述詞會被推送到正在查詢的服務。例如，查詢特定 Amazon EC2 執行個體的詳細資訊會使用特定執行個體 ID 呼叫 EC2 API，以執行目標描述操作。

## 授權資訊

Amazon Athena AWS CMDB 連接器專案是根據 [阿帕奇 2.0](#) 授權進行授權。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena IBM Db2 連接器

適用於 Db2 的 Amazon Athena 連接器可讓 Amazon Athena 使用 JDBC 對 IBM Db2 資料庫執行 SQL 查詢。

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。
- 您必須先設定 VPC 和安全群組，才能使用此連接器。如需詳細資訊，請參閱 [為資料來源連接器建立 VPC](#)。

### 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 篩選條件中的日期和時間戳記資料類型必須轉換為適當的資料類型。

### 條款

下列術語與 Db2 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Db2 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
dbtwo://${jdbc_connection_string}
```

### 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	Db2MuxCompositeHandler
中繼資料處理常式	Db2MuxMetadataHandler
記錄處理常式	Db2MuxRecordHandler

### 多工處理常式參數

參數	描述
<code>catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mydbtwocatalog</code> ，則環境變數名稱為 <code>mydbtwocatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Db2 MUX Lambda 函數：dbtwo1 (預設) 和 dbtwo2。

屬性	Value
default	dbtwo://jdbc:db2://dbtwo1.hostname:port/ <i>database_name</i> :\${secret1_name }
dbtwo_catalog1_connection_string	dbtwo://jdbc:db2://dbtwo1. hostname:port/ <i>database_name</i> :\${secret1_name }
dbtwo_catalog2_connection_string	dbtwo://jdbc:db2://dbtwo2. hostname:port/ <i>database_name</i> :\${secret2_name }

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代之為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

## 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${secret_name}`。

```
dbtwo://jdbc:db2://hostname:port/database_name:${secret_name}
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
dbtwo://jdbc:db2://hostname:port/database_name:user=user_name;password=password;
```

## 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Db2 執行個體。

處理常式類型	類別
複合處理常式	Db2CompositeHandler
中繼資料處理常式	Db2MetadataHandler
記錄處理常式	Db2RecordHandler

## 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 Db2 執行個體。

屬性	Value
default	dbtwo://jdbc:db2://hostname:port/ <i>database_name</i> :\${secret_name}

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC 和 Arrow 的相應資料類型。

Db2	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR
DATE	DATEDAY
TIME	VARCHAR
TIMESTAMP	DATEMILLI
DATETIME	DATEMILLI
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT

Db2	Arrow
BIGINT	BIGINT
DECIMAL	DECIMAL
REAL	FLOAT8
DOUBLE	FLOAT8
DECFLOAT	FLOAT8

## 分割區和分隔

分割區由 `varchar` 類型的一或多個分割區資料欄表示。Db2 連接器使用下列組織結構描述建立分割區。

- 依雜湊分配
- 依範圍分割
- 依維度組織

連接器會從一或多個 Db2 中繼資料資料表擷取分割區詳細資訊，例如分割區數目和資料欄名稱。根據識別的分割區數目建立分割。

## 效能

Athena Db2 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Db2 連接器可以合併這些表達式，並將其直接推送至 Db2，以增強功能並減少掃描的資料量。

下列 Athena Db2 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

### 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

### 直通查詢

Db2 連接器支援[直通](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要使用 Db2 的傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 Db2 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上 GitHub Db2 連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena IBM Db2 AS/400 (Db2 系列) 連接器

適用於 Db2 AS/400 的亞馬遜雅典娜連接器可讓亞馬遜雅典娜使用 JDBC 在您的 IBM Db2 AS/400 (Db2 系列) 資料庫上執行 SQL 查詢。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。
- 您必須先設定 VPC 和安全群組，才能使用此連接器。如需詳細資訊，請參閱 [為資料來源連接器建立 VPC](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 篩選條件中的日期和時間戳記資料類型必須轉換為適當的資料類型。

## 條款

下列術語與 Db2 AS/400 連接器有關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。

- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

您可以使用本節中的 Lambda 環境變數來設定 Db2 AS/400 連接器。

## 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
db2as400://${jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	Db2MuxCompositeHandler
中繼資料處理常式	Db2MuxMetadataHandler
記錄處理常式	Db2MuxRecordHandler

## 多工處理常式參數

參數	描述
<code>\$catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mydb2as40</code>

參數	描述
	0catalog ，則環境變數名稱為 mydb2as400catalog_connection_string 。
default	必要。預設的连接字串。目錄為 lambda:\${ <i>AWS_LAMBDA_FUNCTION_NAME</i> } 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Db2 MUX Lambda 函數：db2as4001 (預設) 和 db2as4002。

屬性	Value
default	db2as400://jdbc:as400:// <i>&lt;ip_address&gt;</i> ; <i>&lt;properties&gt;</i> ;:\${ <i>&lt;secret name&gt;</i> };
db2as400_catalog1_connection_string	db2as400://jdbc:as400://db2as4001. hostname/ :\${ <i>secret1_name</i> }
db2as400_catalog2_connection_string	db2as400://jdbc:as400://db2as4002. hostname/ :\${ <i>secret2_name</i> }
db2as400_catalog3_connection_string	db2as400://jdbc:as400:// <i>&lt;ip_address&gt;</i> ;user= <i>&lt;username&gt;</i> ;password= <i>&lt;password&gt;</i> ; <i>&lt;properties&gt;</i> ;

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

**⚠ Important**

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

**帶有秘密名稱的連接字串範例**

以下字串具有秘密名稱 `${secret_name}`。

```
db2as400://jdbc:as400://<ip_address>;<properties>;:${<secret_name>};
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
db2as400://jdbc:as400://<ip_address>;user=<username>;password=<password>;<properties>;
```

**使用單一連接處理常式**

您可以使用下列單一連線中繼資料和記錄處理常式來連線至單一 Db2 AS/400 執行個體。

處理常式類型	類別
複合處理常式	Db2CompositeHandler
中繼資料處理常式	Db2MetadataHandler

處理常式類型	類別
記錄處理常式	Db2RecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數支援的單一 Db2 AS/400 執行個體。

屬性	Value
default	db2as400://jdbc:as400:// <i>&lt;ip_address&gt;</i> ; <i>&lt;properties&gt;</i> ;: \${ <i>&lt;secret_name&gt;</i> };

### 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC 和 Apache Arrow 的相應資料類型。

定位系統	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR
DATE	DATEDAY
TIME	VARCHAR
TIMESTAMP	DATEMILLI
DATETIME	DATEMILLI
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT
BIGINT	BIGINT
DECIMAL	DECIMAL
REAL	FLOAT8
DOUBLE	FLOAT8
DECFLOAT	FLOAT8

## 分割區和分隔

分割區由 varchar 類型的一或多個分割區資料欄表示。Db2 AS/400 連接器會使用下列組織配置建立分割區。

- 依雜湊分配
- 依範圍分割

- 依維度組織

連接器會從一或多個 Db2 AS/400 中繼資料表格擷取分割區詳細資訊，例如分割區數目和資料行名稱。根據識別的分割區數目建立分割。

### 效能

若要提升效能，請使用述詞下推來從 Athena 進行查詢，如下列範例所示。

```
SELECT * FROM "lambda:<LAMBDA_NAME>"."<SCHEMA_NAME>"."<TABLE_NAME>"
WHERE integercol = 2147483647
```

```
SELECT * FROM "lambda: <LAMBDA_NAME>"."<SCHEMA_NAME>"."<TABLE_NAME>"
WHERE timestampcol >= TIMESTAMP '2018-03-25 07:30:58.878'
```

### 直通查詢

Db2 AS/400 連接器支援[直通](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要使用 Db2 AS/400 的傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 Db2 AS/400 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

### 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上 Db2 AS/400 連接器的 [pom.xml](#) 檔案。GitHub

如需有關此連接器的其他資訊，請造訪 GitHub .com 上 [的對應網站](#)。

## Amazon Athena DocumentDB 連接器

Amazon Athena DocumentDB 連接器讓 Athena 能夠與您的 DocumentDB 執行個體通訊，以便使用 SQL 來查詢您的 DocumentDB 資料。此連接器也適用於 MongoDB 相容的任何端點。

與傳統的關聯式資料存放區不同，Amazon DocumentDB 集合沒有集結構描述。DocumentDB 沒有中繼資料存放區。DocumentDB 集合中的每個項目可以有不同的欄位和資料類型。

DocumentDB 連接器支援兩種機制來產生資料表結構描述資訊：基本結構描述推論和中繼資料。AWS Glue Data Catalog

結構描述推斷是預設值。此選項會掃描集合中的少量文件，形成所有欄位的聯集，並指定具有非重疊資料類型的欄位。此選項適用於大多數具有統一條目的集合。

對於具有更多資料類型的集合，連接器支援從 AWS Glue Data Catalog 中擷取中繼資料。如果連接器看到與 DocumentDB 資料 AWS Glue 料庫和集合名稱相符的資料庫和資料表，它會從對應 AWS Glue 的資料表取得其結構描述資訊。當您創建您的 AWS Glue 表，我們建議您使它成為您可能想要從 DocumentDB 集合訪問的所有字段的超集。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 [AWS Glue Data Catalog](#)

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 參數

使用本節中的 Lambda 環境變數來設定 DocumentDB 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。

- `spill_put_request_headers` – (選用) 用於溢出的 Amazon S3 `putObject` 請求的請求標頭和值的 JSON 編碼映射 (例如, `{"x-amz-server-side-encryption" : "AES256"}`)。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- `kms_key_id` - (選用) 依預設, 任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰, 例如 `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, 您可以指定 KMS 金鑰 ID。
- `disable_spill_encryption` - (選用) 當設定為 `True` 時, 停用溢出加密。預設為 `False`, 因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能, 尤其是如果溢出位置使用 [伺服器端加密](#)。
- `disabled_le_glue` — (選擇性) 如果存在且設定為 `true`, 則連接器不會嘗試從中擷取補充中繼資料。  
AWS Glue
- `glue_catalog` - (選用) 使用此選項可指定 [跨帳戶 AWS Glue 目錄](#)。依預設, 連接器會嘗試從其自己的 AWS Glue 帳戶取得中繼資料。
- `default_docdb` - 如果存在, 請指定當不存在目錄特定環境變數時要使用的 DocumentDB 連接字串。
- `disable_projection_and_casing` - (選用) 停用投影和大小寫。如果想要查詢使用區分大小寫資料欄名稱的 Amazon DocumentDB 資料表, 請使用此選項。`disable_projection_and_casing` 參數使用下列值來指定大小寫和資料欄映射的行為:
  - `false` – 這是預設設定。投影已啟用, 連接器預期所有資料欄名稱都為小寫字母。
  - `true` – 停用投影和大小寫。使用 `disable_projection_and_casing` 參數時, 請謹記以下幾點:
    - 使用該參數可能會導致更高的頻寬用量。此外, 如果您的 Lambda 函數與資料來源位於不同的 AWS 區域, 則將因為頻寬用量增加而產生較高的標準 AWS 跨區域傳輸成本。如需跨區域傳輸成本的詳細資訊, 請參閱 AWS 合作夥伴網路部落格中的 [的伺服器和無伺服器架構的資料傳輸費用](#)。
    - 由於傳輸的位元組數目較多, 而且位元組數目越多, 需要越長的還原序列化時間, 因此整體延遲可能會增加。
- `啟用案例不敏感_匹配` — (選用) 何時 `true`, 對 Amazon DocumentDB 中的模式和表名執行不區分大小寫的搜索。預設值為 `false`。如果您的查詢包含大寫的結構描述或資料表名稱, 則

## 指定連接字串

您可以提供一個或多個屬性, 以定義與連接器搭配使用的 DocumentDB 執行個體的 DocumentDB 連接詳細資訊。因此, 請設定您要在 Athena 使用的目錄名稱所對應的 Lambda 環境變數。例如, 假設您想要使用下列查詢來查詢來自 Athena 的兩個不同 DocumentDB 執行個體:

```
SELECT * FROM "docdb_instance_1".database.table
```

```
SELECT * FROM "docdb_instance_2".database.table
```

您必須先將兩個環境變數 `docdb_instance_1` 和 `docdb_instance_2` 新增至 Lambda 函數，才能使用這兩個 SQL 陳述式。每個陳述式的值應該為以下格式的 DocumentDB 連接字串：

```
mongodb://:@/?ssl=true&ssl_ca_certs=rds-combined-ca-bundle.pem&replicaSet=rs0
```

## 使用秘密

您可以選擇性地用 AWS Secrets Manager 於連接字串詳細資料的部分或全部值。若要搭配使用 Athena 聯合查詢功能和 Secrets Manager，連接到您的 Lambda 函數的 VPC 應該具有[網際網路存取權](#)或 [VPC 端點](#)，以連接到 Secrets Manager。

如果您使用語法 `${my_secret}` 將 Secrets Manager 秘密的名稱放入連接字串中，連接器會以 Secret Manager 中的相應純文字值來取代 `${my_secret}`。秘密應儲存為具有值 `<username>:<password>` 的純文字秘密。儲存為 `{username:<username>,password:<password>}` 的秘密將不會正確地傳遞給連接字串。

秘密也可以完整地用於整個連接字串，並且可以在秘密中定義使用者名稱和密碼。

例如，假設您將 `docdb_instance_1` 的 Lambda 環境變數設定為以下值：

```
mongodb://${docdb_instance_1_creds}@myhostname.com:123/?ssl=true&ssl_ca_certs=rds-combined-ca-bundle.pem&replicaSet=rs0
```

Athena Query Federation 軟體開發套件會自動嘗試從 Secrets Manager 中擷取名為 `docdb_instance_1_creds` 的秘密，並將該值放入 `${docdb_instance_1_creds}` 位置。由 `{ }` 字元組合括住的連接字串的任何部分會被解譯為 Secrets Manager 的秘密。如果您指定連接器在 Secrets Manager 中找不到的秘密名稱，則連接器不會取代該文字。

## 設定資料庫和資料表 AWS Glue

由於連接器的內建結構描述推論功能會掃描有限數量的文件，而且只支援資料類型的子集，因此您可能想要改用 AWS Glue 於中繼資料。

若要啟用與 Amazon DocumentDB 搭配使用的 AWS Glue 表格，您必須擁有要為其提供補充中繼 AWS Glue 資料的 DocumentDB 資料庫和集合的資料庫和表格。

## 若要將 AWS Glue 表格用於補充描述資料

1. 當您在 AWS Glue 主控台中編輯表格和資料庫時，請新增下表屬性。
  - `docdb-metadata-flag`— 此內容向 DocumentDB 連接器指示連接器可將表格用於補充中繼資料。您可以為 `docdb-metadata-flag` 提供任何值，只要 `docdb-metadata-flag` 屬性存在於資料表屬性清單中。
2. (選用) 新增 `sourceTable` 資料表屬性。此屬性定義 Amazon DocumentDB 中的來源資料表名稱。如果 AWS Glue 表格命名規則無法建立與 Amazon DocumentDB AWS Glue 表格名稱相同的資料表，請使用此屬性。例如，AWS Glue 資料表名稱中不允許大寫字母，但在 Amazon DocumentDB 資料表名稱中允許。
3. (選用) 新增 `columnMapping` 資料表屬性。此屬性可定義資料欄名稱映射。如果資料行命名規則阻止您建立的資料行名稱與 Amazon DocumentDB AWS Glue 表格中的資料行名稱相同，請使用此屬性。這非常有用，因為 Amazon DocumentDB 資料欄名稱中允許大寫字母，但 AWS Glue 資料欄名稱中不允許。

`columnMapping` 屬性值應該是格式 `col1=Col1,col2=Col2` 中的一組映射。

### Note

資料欄映射僅適用於頂層資料欄名稱，而不適用於巢狀欄位。

新增資料表 `columnMapping` 屬性之後，您可以移除 `disable_projection_and_casing` Lambda 環境變數。

4. 請確定您使用本文件中所列適 AWS Glue 用的資料類型。

## 支援的資料類型

本節列出 DocumentDB 連接器用於結構描述推論的資料類型，以及使用 AWS Glue 中繼資料時的資料類型。

### 結構描述推斷資料類型

DocumentDB 連接器的結構描述推斷功能會嘗試將值推斷為以下資料類型之一。該資料表顯示 Amazon DocumentDB、Java 和 Apache Arrow 的相應資料類型。

Apache Arrow	Java 或 DocDB
VARCHAR	字串
INT	Integer
BIGINT	Long
BIT	Boolean
FLOAT4	Float
FLOAT8	Double
TIMESTAMPSEC	日期
VARCHAR	ObjectId
LIST	清單
STRUCT	文件

### AWS Glue 資料類型

如果您用 AWS Glue 於補充描述資料，您可以設定下列資料類型。該表顯示了對應的數據類型 AWS Glue 和 Apache 箭頭。

AWS Glue	Apache Arrow
int	INT
bigint	BIGINT
double	FLOAT8
float	FLOAT4
boolean	BIT
binary	VARBINARY

AWS Glue	Apache Arrow
string	VARCHAR
清單	LIST
Struct	STRUCT

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-docdb.yaml](#) 檔案的 Policies 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- AWS Glue Data Catalog— DocumentDB 連接器需要的唯讀存取權，才能取得結 AWS Glue Data Catalog 構描述資訊。
- CloudWatch 防護記錄 — 連接器需要存取 CloudWatch 記錄檔以儲存防護記錄。
- AWS Secrets Manager 讀取存取權 — 如果您選擇將 DocumentDB 端點詳細資料儲存在 Secrets Manager 中，您必須授與連接器存取這些密碼。
- VPC 存取 - 連接器需要能夠將介面連接到 VPC 並能進行分離，以便可進行連接並與 DocumentDB 執行個體通訊。

## 效能

Athena Amazon DocumentDB 連接器目前不支援平行掃描，但會嘗試下推述詞作為其 DocumentDB 查詢的一部分，而針對 DocumentDB 集合上之索引的述詞會導致掃描的資料大幅減少。

Lambda 函數執行投影下推，以減少查詢掃描的資料。不過，選取資料欄子集有時會導致較長的查詢執行時間。LIMIT 子句可減少掃描的資料量，但是如果您未提供述詞，則應期望具有 LIMIT 子句的 SELECT 查詢掃描至少 16MB 的資料。

## 直通查詢

Athena Amazon DocumentDB 連接器支持[直通查詢](#)，並且是基於 NoSQL。如需查詢 Amazon 文件資料庫的相關資訊，請參閱 Amazon 文件資料庫開發人員指南中的[查詢](#)。

若要搭配 Amazon DocumentDB 使用傳遞查詢，請使用下列語法：

```
SELECT * FROM TABLE(  
    system.query(  
        database => 'database_name',  
        collection => 'collection_name',  
        filter => '{query_syntax}'  
    ))
```

下列範例會查詢TPCDS集中的example資料庫，篩選標題為「權利法案」的所有書籍。

```
SELECT * FROM TABLE(  
    system.query(  
        database => 'example',  
        collection => 'tpcds',  
        filter => '{title: "Bill of Rights"}'  
    ))
```

## 其他資源

- 有關使用 [Amazon Athena 聯合查詢](#) 將 MongoDB 數據庫連接到 [Amazon QuickSight](#) 以構建儀表板和可視化的文章，請參閱在大數據博客中 [QuickSight 使用 Amazon 雅典娜聯合查詢將來自亞馬遜的 MongoDB 數據可視化](#)。AWS
- 如需有關此連接器的其他資訊，請造訪 GitHub .com 上 [的對應網站](#)。

## Amazon Athena DynamoDB 連接器

Amazon Athena DynamoDB 連接器讓 Amazon Athena 能夠與 DynamoDB 通訊，以便使用 SQL 來查詢您的資料表。不支援 [INSERT INTO](#) 等寫入操作。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 [AWS Glue Data Catalog](#)

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 參數

使用本節中的 Lambda 環境變數來設定 DynamoDB 連接器。

- `spill_bucket` - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- `spill_prefix` - (選用) 預設為指定的 `spill_bucket` 中名為 `athena-federation-spill` 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- `spill_put_request_headers` - (選用) 用於溢出的 Amazon S3 `putObject` 請求的請求標頭和值的 JSON 編碼映射 (例如，`{"x-amz-server-side-encryption" : "AES256"}`)。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- `kms_key_id` - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`，您可以指定 KMS 金鑰 ID。
- `disable_spill_encryption` - (選用) 當設定為 `True` 時，停用溢出加密。預設為 `False`，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。
- `disable_glue` — (選擇性) 如果存在且設定為 `true`，則連接器不會嘗試從中擷取補充中繼資料。  
AWS Glue
- `glue_catalog` - (選用) 使用此選項可指定 [跨帳戶 AWS Glue 目錄](#)。依預設，連接器會嘗試從其自己的 AWS Glue 帳戶取得中繼資料。
- `disable_projection_and_casing` - (選用) 停用投影和大小寫。如果您想要查詢在其欄名稱中有大小寫的 DynamoDB 表格，且您不想在資料表上指定 `columnMapping` 屬性，請使用此選 AWS Glue 項。

`disable_projection_and_casing` 參數使用下列值來指定大小寫和資料欄映射的行為：

- `auto` (自動) - 當偵測到先前不支援的類型且未在資料表中設定資料欄名稱映射時，停用投影和大小寫。這是預設設定。
- `always` (永遠) - 無條件停用投影和大小寫。當 DynamoDB 資料欄名稱中有大小寫，但您不想指定任何資料欄名稱映射時，此功能非常有用。

使用 `disable_projection_and_casing` 參數時，請謹記以下幾點：

- 使用該參數可能會導致更高的頻寬用量。此外，如果您的 Lambda 函數與資料來源位於不同的 AWS 區域，則將因為頻寬用量增加而產生較高的標準 AWS 跨區域傳輸成本。如需跨區域傳輸成本的詳細資訊，請參閱 AWS 合作夥伴網路部落格中的 [的伺服器和無伺服器架構的資料傳輸費用](#)。
- 由於傳輸的位元組數目較多，而且位元組數目越多，需要越長的還原序列化時間，因此整體延遲可能會增加。

## 設定資料庫和資料表 AWS Glue

由於連接器的內建結構描述推論功能受到限制，因此您可能想要用 AWS Glue 於中繼資料。要做到這一點，你必須有一個數據庫和表 AWS Glue。若要啟用資料庫和資料表以與 DynamoDB 搭配使用，您必須編輯其屬性。

若要在 AWS Glue 主控台中編輯資料庫屬性

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，網址為 <https://console.aws.amazon.com/glue/>。
2. 在瀏覽窗格中，展開 [資料目錄]，然後選擇 [資料庫]。

在 Databases (資料庫) 頁面中，您可以編輯現有的資料庫，或選擇 Add database (新增資料庫) 來建立資料庫。

3. 在資料庫清單中，選擇您要編輯的資料庫連結。
4. 選擇編輯。
5. 在 [更新資料庫] 頁面的 [資料庫設定] 下，對於 [位置]，新增字串 **dynamo-db-flag**。此關鍵字表示 AWS Glue 資料庫包含 Athena DynamoDB 連接器用於補充中繼資料的表格，而且對於以外的資料庫則是必需的。defaultdynamo-db-flag 屬性有助於在具有許多資料庫的帳戶中篩選資料庫。
6. 選擇 Update Database (更新資料庫)。

在 AWS Glue 主控台中編輯表格屬性的步驟

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，網址為 <https://console.aws.amazon.com/glue/>。
2. 在導覽窗格中，展開 [資料目錄]，然後選擇 [表格]。
3. 在「表格」頁面的表格清單中，選擇要編輯之表格的連結名稱。
4. 選擇 Actions (動作)、Edit table (編輯資料表)。
5. 在 Edit table (編輯資料表) 頁面的 Table properties (資料表屬性) 部分中，根據需要新增以下資料表屬性。如果您使用 AWS Glue DynamoDB 爬行者程式，則會自動設定這些屬性。
  - dynamodb - 向 Athena DynamoDB 連接器發出指示的字串，指示資料表可用於補充中繼資料。在名為 classification (分類) 的欄位下的資料表屬性中輸入 dynamodb (完全匹配)。

**Note**

作為主控台中表格建立程序一部分的 [設定表格屬性] 頁 AWS Glue 面具有 [分類] 欄位的 [資料格式] 區段。在這裡您無法進入或選擇 dynamodb。相反，在建立資料表後，請依照下列步驟來編輯資料表，並在資料表屬性區段中輸入 classification 和 dynamodb 做為索引鍵值配對。

- **sourceTable** - 在 DynamoDB 中定義來源資料表名稱的選用資料表屬性。如果 AWS Glue 表格命名規則無法建立與 DynamoDB AWS Glue 表格名稱相同的資料表，請使用此選項。例如，資料表名稱中不允許使用大寫字母，但 DynamoDB AWS Glue 表格名稱中允許使用這些字母。
- **columnMapping** - 定義資料欄名稱映射的選用資料表屬性。如果 AWS Glue 欄命名規則無法建立與 DynamoDB AWS Glue 表格具有相同欄名稱的資料表，請使用此選項。例如，欄名稱不允許使用大寫字母，但 DynamoDB AWS Glue 欄名稱中允許使用。屬性值的格式應為：col1=Col1, col2=Col2。請注意，資料欄映射僅適用於頂層資料欄名稱，而不適用於巢狀欄位。
- **defaultTimeZone**— 套用至的選擇性表格屬性 date 或沒有明確時區的 datetime 值。若要避免資料來源預設時區與 Athena 工作階段時區之間的差異，設定此值是一個很好的方法。
- **datetimeFormatMapping**— 選用的資料表屬性，指定剖析 date 或資料類型之欄的資料時要使用的 AWS Glue date 或 datetime 格式。如果未指定此屬性，連接器會嘗試推斷 ISO-8601 格式。如果連接器無法推斷 date 或者 datetime 的格式，或無法剖析原始字串，則會從結果中省略該值。

datetimeFormatMapping 值的格式應為 col1=someformat1,col2=someformat2。以下是一些範例格式：

```
yyyyMMdd'T'HHmmss  
ddMMyyyy'T'HH:mm:ss
```

如果您的資料欄具有沒有時區的 date 或 datetime 值，並且您想要在 WHERE 子句中使用該資料欄，請設定該資料欄的 datetimeFormatMapping 屬性。

6. 如果您手動定義資料欄，請務必使用適當的資料類型。如果您使用爬蟲程式，請驗證爬蟲程式發現的資料欄和類型。
7. 選擇儲存。

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-dynamodb.yaml](#) 檔案的 Policies 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- AWS Glue Data Catalog— DynamoDB 連接器需要的唯讀存取權，才能取得結 AWS Glue Data Catalog 構描述資訊。
- CloudWatch 防護記錄 — 連接器需要存取 CloudWatch 記錄檔以儲存防護記錄。
- DynamoDB 讀取權限 - 連接器使用 DescribeTable、ListSchemas、ListTables、Query 以及 Scan API 操作。

## 效能

Athena DynamoDB 連接器支援平行掃描，並嘗試下推述詞作為 DynamoDB 查詢的一部分。具有 X 個不同值的雜湊索引鍵述詞會導致對 DynamoDB 發出 X 個查詢呼叫。所有其他述詞案例都會導致 Y 個掃描呼叫，其中 Y 是根據資料表大小及其佈建的輸送量啟發式決定的。不過，選取資料欄子集有時會導致查詢執行期更長。

LIMIT 子句和簡單述詞會下推，可以減少掃描的資料量，並會導致查詢執行時間縮短。

### LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

### 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。為了增強功能及減少掃描的資料量，Athena DynamoDB 連接器可以合併這些表達式並將其直接推送至 DynamoDB。

下列 Athena DynamoDB 連接器運算子支援述詞下推：

- 布林值：AND
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10 and col_b < 10
LIMIT 10
```

如需有關使用述詞下推以改善聯合查詢中的效能 (包括 DynamoDB) 的文章，請參閱 AWS 大數據部落格中的[使用 Amazon Athena 中的述詞下推改善聯合查詢](#)。

## 直通查詢

DynamoDB 連接器支援[傳遞查詢](#)，並使用 PartiQL 語法。不支援 [GetItem](#) API 作業。如需使用 PartiQL 查詢 DynamoDB 的相關資訊，請參閱 Amazon [DynamoDB 開發人員指南](#)中的 [PartiQL 選取陳述式](#)。

若要搭配 DynamoDB 使用傳遞查詢，請使用下列語法：

```
SELECT * FROM TABLE(
    system.query(
        query => 'query_string'
    ))
```

下列 DynamoDB 直通查詢範例使用 PartiQL 傳回具有早於 12/24/22 DateWatched 屬性的消防電視棒裝置清單。

```
SELECT * FROM TABLE(
    system.query(
        query => 'SELECT Devices
                FROM WatchList
                WHERE Devices.FireStick.DateWatched[0] > '12/24/22''
    ))
```

## 故障診斷

排序索引鍵資料欄上的多個篩選條件

錯誤訊息：每KeyConditionExpressions個金鑰只能包含一個條件

原因：在 DynamoDB 排序索引鍵資料欄上同時具有下限篩選條件和上限篩選條件的查詢中，Athena 引擎第 3 版可能會發生此問題。由於 DynamoDB 不支援排序索引鍵上的多個篩選條件，因此當連接器嘗試下推套用這兩個條件的查詢時，即會擲回錯誤。

解決方案：將連接器更新到 2023.11.1 版本或更新版本。如需有關更新連接器的說明，請參閱 [更新資料來源連接器](#)。

## 成本

連接器的使用成本取決於所使用的基礎 AWS 資源。因為使用掃描的查詢可能會消耗大量的 [讀取容量單位 \(RCU\)](#)，所以請仔細考慮 [Amazon DynamoDB 定價](#) 資訊。

## 其他資源

- 如需有關使用 Amazon Athena DynamoDB 連接器的說明，請參閱《AWS 規範性指引模式》指南中的 [使用 Athena 存取、查詢和聯結 Amazon DynamoDB 資料表](#)。
- 如需有關如何使用雅典娜 DynamoDB 連接器使用 SQL 查詢 DynamoDB 中的資料，並在 Amazon 中視覺化洞察的文章 QuickSight，請參閱 AWS 大數據部落格文章 [QuickSight 使用 Amazon Athena DynamoDB 連接器在亞馬遜中視覺化亞馬遜 DynamoDB 深入解析](#)。AWS Glue
- 如需使用 Amazon Athena DynamoDB 連接器搭配 Amazon DynamoDB、Athena 和亞馬遜建立簡單控管儀表板的文章，請參閱使用亞馬遜 QuickSight 雅典娜聯合查詢的 AWS 大數據部落格文章 [查詢跨帳戶 Amazon DynamoDB 表](#)。
- 如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena 谷歌 BigQuery 連

Amazon Athena 谷歌連接器 [BigQuery](#) 使 Amazon Athena 能夠對您的谷歌 BigQuery 數據運行 SQL 查詢。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- Lambda 函數的逾時值上限為 15 分鐘。每次分割都會執行查詢，BigQuery 並且必須有足夠的時間來儲存 Athena 才能讀取結果。如果 Lambda 函數逾時，查詢就會失敗。

- 谷歌 BigQuery 是區分大小寫的。連接器會嘗試更正資料集名稱和資料表名稱的大小寫，但不會對專案 ID 進行任何大小寫更正。這是必要的，因為 Athena 會使所有中繼資料變為小寫。這些更正使得許多額外的調用谷歌 BigQuery。
- 不支援二進位資料類型。
- 由於 Google 並發 BigQuery 性和配額限制，連接器可能會遇到 Google 配額限制問題。為了避免這些問題，請 BigQuery 盡可能多地向 Google 推送約束。如需 BigQuery 配額的詳細資訊，請參閱 Google BigQuery 說明文件中的[配額和限制](#)。

## 參數

使用本節中的 Lambda 環境變數來設定 Google BigQuery 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- spill\_put\_request\_headers - (選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"})。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考[PutObject](#)中的。
- kms\_key\_id - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 a7e63k4b-81oc-40db-a2a1-4d0en2cd8331，您可以指定 KMS 金鑰 ID。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用[伺服器端加密](#)。
- gcp\_project\_id - 專案 ID (非專案名稱)，它包含連接器應讀取的資料集 (例如，semiotic-primer-1234567)。
- 秘密名稱 — 密碼的名稱，其中包含您以 JSON 格式的 AWS Secrets Manager 認證 (例如，)。BigQuery GoogleCloudPlatformCredentials
- 端點 — (選用) 私有端點的網址。BigQuery 當您想要透過私有端點存取 BigQuery 時，請使用此參數。

## 分割和檢視

由於 BigQuery 連接器使用 BigQuery 儲存區讀取 API 來查詢資料表，而 BigQuery 儲存區 API 不支援檢視，因此連接器會使用 BigQuery 戶端進行檢視的單一分割。

## 效能

若要查詢資料表，BigQuery 連接器會使用 BigQuery 儲存區讀取 API，該 API 使用 RPC 型通訊協定，可快速存取 BigQuery 受管理儲存區。如需有關 BigQuery 儲存區讀取 API 的詳細資訊，請參閱 Google Cloud 說明文件中的 [使用 BigQuery 儲存空間讀取 API 讀取資料表](#) 資料。

選取資料欄子集可大幅加速查詢執行期，並減少掃描的資料。隨著並行數目增加，連接器容易出現查詢失敗，且連接器速度通常較慢。

Athena Google BigQuery 連接器會執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、ORDER BY 子句、簡單述詞和複雜運算式會向下推送至連接器，以減少掃描的資料量並減少查詢執行時間。

### LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

### 前 N 個查詢

前 N 個查詢會指定結果集的順序，以及傳回的資料列數目的限制。您可以使用這種查詢類型，判斷資料集的前 N 個最大值或前 N 個最小值。透過前 N 個下推，連接器只會向 Athena 傳回 N 個排序的資料列。

### 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Google BigQuery 連接器可以結合這些表達式，並將其直接推送到 Google BigQuery 以增強功能並減少掃描的數據量。

以下 Athena 谷歌 BigQuery 連接器運營商支持謂詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

### 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
ORDER BY col_a DESC
LIMIT 10;
```

## 直通查詢

Google 連 BigQuery 接器支援[直通查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

要在 Google 上使用直通查詢 BigQuery，可以使用以下語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  )
)
```

以下示例查詢將查詢向下推送到 Google BigQuery 中的數據源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  )
)
```

## 授權資訊

Amazon Athena 谷歌 BigQuery 連接器項目是根據[阿帕奇 -2.0](#) 許可證授權。

使用此連接器即表示您確認包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的各自協力廠商授權中的條款。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena Google Cloud Storage 連接器

Amazon Athena Google Cloud Storage 連接器讓 Amazon Athena 能夠對存放在 Google Cloud Storage (GCS) 儲存貯體中的 Parquet 檔案和 CSV 檔案執行查詢。將一個或多個 Parquet 或 CSV 檔案分組到 GCS 儲存貯體中的未分割或分割資料夾之後，您可以在 [AWS Glue](#) 資料庫資料表中對其進行整理。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 `AWS Glue Data Catalog`

如需瞭解如何使用 Athena 對 GCS 儲存貯體中的實木複合地板或 CSV 檔案執行查詢的文章，請參閱 AWS 大數據部落格文章 [使用 Amazon Athena 查詢儲存在 Google 雲端平台中的資料](#)。

### 必要條件

- 在 GCP 儲存空間中設定對應儲存貯體和資料夾的資料 AWS Glue 庫和資料表。如需這些步驟，請參閱本文件後面的 [設置數據庫和表 AWS Glue](#)。
- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 限制

- 不支援寫入 DDL 操作。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 目前，連接器僅支援磁碟分割資料行 (string 或 varchar 資料 AWS Glue 表結構描述) 的 VARCHAR 類型。當您在 Athena 中查詢其他分割區欄位類型時，會引發錯誤。

### 條款

下列術語與 GCS 連接器相關。

- 處理常式 - 存取 GCS 儲存貯體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從 GCS 儲存貯體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從 GCS 儲存貯體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從 GCS 儲存貯體中擷取中繼資料和資料記錄的 Lambda 處理常式。

## 支援的檔案類型

GCS 連接器支援 Parquet 和 CSV 檔案類型。

### Note

請確定不要將 CSV 和 Parquet 檔案同時放在相同的 GCS 儲存貯體或路徑中。如果嘗試將 Parquet 檔案讀取為 CSV，可能會導致執行階段錯誤，反之亦然。

## 參數

使用本節中的 Lambda 環境變數來設定 GCS 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- spill\_put\_request\_headers - (選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"})。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- kms\_key\_id - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 a7e63k4b-8loc-40db-a2a1-4d0en2cd8331，您可以指定 KMS 金鑰 ID。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。
- 秘密名稱 — 密碼的名稱，其中包含您的 JSON 格式的 GCS AWS Secrets Manager 認證 (例如)。GoogleCloudPlatformCredentials

## 設置數據庫和表 AWS Glue

由於 GCS 連接器的內建結構描述推論功能受到限制，因此建議您將中繼資料用 AWS Glue 於中繼資料。下列程序顯示如何建立可從 Athena 存取的 AWS Glue 資料庫和表格。

### 在 AWS Glue 建立資料庫

您可以使用主 AWS Glue 控制台建立與 GCS 連接器搭配使用的資料庫。

## 若要在中建立資料庫 AWS Glue

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，網址為 <https://console.aws.amazon.com/glue/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 選擇新增資料庫。
4. 對於 Name (名稱)，輸入您要搭配 GCS 連接器使用的資料庫名稱。
5. 對於「位置」，請指定 `s3://google-cloud-storage-flag`。此位置告知 GCS 連接器資料 AWS Glue 庫包含要在 Athena 中查詢之 GCS 資料的表格。連接器可辨識 Athena 中具有此標記的資料庫，並忽略沒有此標記的資料庫。
6. 選擇建立資料庫。

## 在 AWS Glue 中建立資料表

現在，您可以為資料庫建立資料表。當您建立要與 GCS 連接器搭配使用的資料 AWS Glue 表時，必須指定其他中繼資料。

## 若要在 AWS Glue 主控台中建立資料表

1. 在 AWS Glue 主控台中，從導覽窗格中選擇 [表格]。
2. 在 Tables (資料表) 頁面中，選擇 Add table (新增資料表)。
3. 在 Set table properties (設定資料表屬性) 頁面中，輸入以下資訊。
  - 名稱 - 資料表的唯一名稱。
  - 資料庫 - 選擇您為 GCS 連接器建立的 AWS Glue 資料庫。
  - 包含路徑 - 在 Data store (資料存放區) 區段中，對於 Include path (包含路徑)，輸入字首為 `gs://` 的 GCS URI 位置 (例如，`gs://gcs_table/data/`)。如果您有一個或多個分割區資料夾，則請不要將其包含在路徑中。

### Note

當您輸入非 `s3://` 資料表路徑時，AWS Glue 主控台會顯示錯誤。您可以忽略此錯誤。將成功建立資料表。

- 資料格式 - 對於 Classification (分類)，請選取 CSV 或 Parquet。
4. 選擇下一步。

5. 強烈建議您在 Choose or define schema (選擇或定義結構描述) 頁面中定義資料表結構描述，但不是必要的。如果您未定義結構描述，則 GCS 連接器會嘗試為您推斷結構描述。

執行以下任意一項：

- 如果要讓 GCS 連接器嘗試為您推斷結構描述，則請選擇 Next (下一步)，然後選擇 Create (建立)。
- 若要自行定義結構描述，請遵循下一節中的步驟。

## 定義資料表結構定義於 AWS Glue

在中定義資料表結構定義 AWS Glue 需要更多步驟，但可讓您更好地控制資料表建立程序。

若要在中定義資料表的結構定義 AWS Glue

1. 在 Choose or define schema (選擇或定義結構描述) 頁面中，選擇 Add (新增)。
2. 使用 Add schema entry (新增結構描述項目) 對話方塊提供資料欄名稱和資料類型。
3. 若要將資料欄指定為分割區資料欄，請選取 Set as partition key (設定為分割區索引鍵) 選項。
4. 選擇 Save (儲存) 可儲存資料欄。
5. 選擇 Add (新增) 可新增另一個資料欄。
6. 當您完成新增資料欄時，選擇 Next (下一步)。
7. 在 Review and create (檢閱和建立) 頁面中，檢閱資料表，然後選擇 Create (建立)。
8. 如果您的結構描述包含分割區資訊，則請遵循下一節中的步驟，將分割區模式新增至 AWS Glue 中的資料表屬性。

## 將分區模式添加到表屬性 AWS Glue

如果您的 GCS 儲存貯體具有分割區，則必須將分割區模式新增至 AWS Glue 中的資料表屬性。

若要將分割區資訊新增至表格屬性 AWS Glue

1. 在您在中建立之表格的詳細資訊頁面上 AWS Glue，選擇動作，編輯表格。
2. 在 Edit table (編輯資料表) 頁面中，向下捲動至 Table properties (資料表屬性) 區段。
3. 選擇 Add (新增) 以新增分割區索引鍵。
4. 在 Key (索引鍵) 欄位，輸入 **partition.pattern**。此索引鍵定義資料夾路徑模式。

5. 在 Value (值) 中，輸入資料夾路徑樣式，例如 **StateName=\${statename}/ZipCode=\${zipcode}/**，其中由 **\${}** 括起來的 **statename** 和 **zipcode** 是分割區資料欄名稱。GCS 連接器支援 Hive 和非 Hive 分割區結構描述。
6. 完成時，請選擇 Save (儲存)。
7. 若要檢視您剛剛建立的資料表屬性，請選擇 Advanced properties (進階屬性) 索引標籤。

此時，您可以導覽至 Athena 主控台。您在其中建立的資料庫和資料表可供 AWS Glue 在 Athena 查詢。

## 支援的資料類型

下表顯示 CSV 和 Parquet 支援的資料類型。

### CSV

Nature of data (資料性質)	Inferred Data Type (推斷的資料類型)
資料看起來像數字	BIGINT
資料看起來像字串	VARCHAR
資料看起來像浮點值 (浮點數、雙精度或十進制)	DOUBLE
資料看起來像日期	時間戳記
包含真/假值的資料	BOOL

### Parquet

PARQUET	Athena (Arrow) (Athena (箭頭))
BINARY	VARCHAR
BOOLEAN	BOOL
DOUBLE	DOUBLE
ENUM	VARCHAR

PARQUET	Athena (Arrow) (Athena (箭頭))
FIXED_LEN _BYTE_ARRAY	DECIMAL
FLOAT	FLOAT (32 位元)
INT32	1. INT32 2. DATEDAY (當 Parquet 資料欄邏輯類型為 DATE 時)
INT64	1. INT64 2. TIMESTAMP (當 Parquet 資料欄邏輯類型為 TIMESTAMP 時)
INT96	時間戳記
MAP	MAP
STRUCT	STRUCT
LIST	LIST

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-gcs.yaml](#) 檔案的 Policies 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- AWS Glue Data Catalog— GCS 連接器需要的唯讀存取權才能取 AWS Glue Data Catalog 得結構描述資訊。
- CloudWatch 防護記錄 — 連接器需要存取 CloudWatch 記錄檔以儲存防護記錄。

## 效能

當資料表結構描述包含分割區欄位且 `partition.pattern` 資料表屬性設定正確時，您可以在查詢的 WHERE 子句中包含分割區欄位。對於此類查詢，GCS 連接器會使用分割區資料欄來精簡 GCS 資料夾路徑，並避免掃描 GCS 資料夾中不需要的檔案。

對於 Parquet 資料集，選取資料欄子集會導致掃描的資料較少。當套用資料欄投影時，這通常會導致較短的查詢執行階段。

對於 CSV 資料集，不支援資料欄投影，並且不會減少掃描的資料量。

LIMIT 子句可減少掃描的資料量，但是如果您未提供述詞，則應期望具有 LIMIT 子句的 SELECT 查詢掃描至少 16 MB 的資料。與較小資料集相比，GCS 連接器可為較大資料集掃描更多資料，無論是否套用 LIMIT 子句。例如，與較小的基礎資料集相比，SELECT \* LIMIT 10000 查詢可為較大的基礎資料集掃描更多資料。

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena HBase 連接器

Amazon Athena HBase 連接器讓 Amazon Athena 能夠與您的 Apache HBase 執行個體通訊，以便使用 SQL 來查詢您的 HBase 資料。

與傳統的關聯式資料存放區不同，HBase 集合沒有集結構描述。HBase 沒有中繼資料存放區。HBase 集合中的每個條目可以有不同的欄位和資料類型。

HBase 連接器支援兩種機制來產生資料表結構描述資訊：基本結構描述推論和 AWS Glue Data Catalog 中繼資料。

結構描述推論是預設值。此選項會掃描集合中的少量文件，形成所有欄位的聯集，以及強制具有非重疊資料類型的欄位。此選項適用於大多數具有統一條目的集合。

對於具有更多資料類型的集合，連接器支援從 AWS Glue Data Catalog 中擷取中繼資料。如果連接器看到與 HBase 命名空間和集合名稱匹配的 AWS Glue 數據庫和表，則它會從相應的 AWS Glue 表中獲取其模式信息。當您建立 AWS Glue 資料表時，我們建議您讓它成為您可能想要從 HBase 集合存取的所有欄位的超集。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 `AWS Glue Data Catalog`

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 參數

使用本節中的 Lambda 環境變數來設定 HBase 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- spill\_put\_request\_headers - (選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"})。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- kms\_key\_id - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 a7e63k4b-81oc-40db-a2a1-4d0en2cd8331，您可以指定 KMS 金鑰 ID。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。
- disabled le\_glue — (選擇性) 如果存在且設定為 true，則連接器不會嘗試從中擷取補充中繼資料。  
AWS Glue
- glue\_catalog - (選用) 使用此選項可指定 [跨帳戶 AWS Glue 目錄](#)。依預設，連接器會嘗試從其自己的 AWS Glue 帳戶取得中繼資料。
- default\_hbase - 如果存在，則指定當不存在目錄特定環境變數時要使用的 HBase 連接字串。
- 啟用案例不敏感\_匹配 — (可選) 當 true，對 HBase 中的表名執行不區分大小寫的搜索。預設值為 false。如果您的查詢包含大寫資料表名稱，則使用

## 指定連接字串

您可以提供一個或多個屬性，以定義與連接器搭配使用的 HBase 執行個體的 HBase 連接詳細資訊。因此，請設定您要在 Athena 使用的目錄名稱所對應的 Lambda 環境變數。例如，假設您想要使用下列查詢來查詢來自 Athena 的兩個不同 HBase 執行個體：

```
SELECT * FROM "hbase_instance_1".database.table
```

```
SELECT * FROM "hbase_instance_2".database.table
```

您必須先將兩個環境變數 `hbase_instance_1` 和 `hbase_instance_2` 新增至 Lambda 函數，才能使用這兩個 SQL 陳述式。每個的值應該為以下格式的 HBase 連接字串：

```
master_hostname:hbase_port:zookeeper_port
```

## 使用秘密

您可以選擇性地用 AWS Secrets Manager 於連接字串詳細資料的部分或全部值。若要搭配使用 Athena 聯合查詢功能和 Secrets Manager，連接到您的 Lambda 函數的 VPC 應該具有[網際網路存取權](#)或 [VPC 端點](#)，以連接到 Secrets Manager。

如果您使用語法 `${my_secret}` 將 Secret Manager 秘密的名稱放入連接字串中，連接器會使用您在 Secret Manager 中的使用者名稱和密碼值來取代秘密名稱。

例如，假設您將 `hbase_instance_1` 的 Lambda 環境變數設定為以下值：

```
${hbase_host_1}:${hbase_master_port_1}:${hbase_zookeeper_port_1}
```

Athena Query Federation 軟體開發套件會自動嘗試從 Secrets Manager 中擷取名為 `hbase_instance_1_creds` 的秘密，並將該值放入 `${hbase_instance_1_creds}` 位置。由 `${ }` 字元組合括住的連接字串的任何部分會被解譯為 Secrets Manager 的秘密。如果您指定連接器在 Secrets Manager 中找不到的秘密名稱，則連接器不會取代該文字。

## 設定資料庫和資料表 AWS Glue

連接器的內建結構描述推斷僅支援在 HBase 中序列化為字串的值 (例如，`String.valueOf(int)`)。由於連接器的內建結構描述推斷功能受到限制，因此您可能需要對中繼資料使用 AWS Glue。要啟用 AWS Glue 表與 HBase 的使用，你必須有一個 AWS Glue 數據庫和表的名稱匹配 HBase 的命名空間和表，你想提供補充元數據。使用 HBase 資料欄系列命名慣例是可選的，但不是必需的。

若要將 AWS Glue 表格用於補充描述資料

1. 在 AWS Glue 主控台中編輯表格和資料庫時，請新增下列表格屬性：

- `hbase-metadata-flag`— 此內容向 HBase 連接器指示連接器可將表格用於補充中繼資料。您可以為 `hbase-metadata-flag` 提供任何值，只要 `hbase-metadata-flag` 屬性存在於資料表屬性清單中。
- `hbase-native-storage-flag`— 使用此旗標可切換連接器支援的兩個值序列化模式。根據預設，當此欄位不存在時，連接器假定所有值都作為字串儲存在 HBase 中。因此，它會嘗試將 HBase 中的 INT、BIGINT 和 DOUBLE 等資料類型作為字串進行剖析。如果使用中的資料表上的任何值設定此欄位 AWS Glue，則連接器會切換至「原生」儲存模式 INTBIGINT，並使用下列函數嘗試讀取 BIT、和 DOUBLE 作為位元組：

```
ByteBuffer.wrap(value).getInt()  
ByteBuffer.wrap(value).getLong()  
ByteBuffer.wrap(value).get()  
ByteBuffer.wrap(value).getDouble()
```

## 2. 請確定您使用本文件中所列適 AWS Glue 用的資料類型。

### 建立資料欄系列模型

Athena HBase 連接器支援兩種方式來建立 HBase 資料欄系列模型：完全限定 (扁平化) 命名 (例如 `family:column`)，或使用 STRUCT 物件。

在 STRUCT 模型中，STRUCT 欄位的名稱應與資料欄系列相符，而 STRUCT 的子項應該與該系列的資料欄名稱相符。但是，由於諸如 STRUCT 等複雜類型尚不完全支援述詞下推和資料欄讀取，因此目前不建議使用 STRUCT。

下列影像顯示在中設定的表格，AWS Glue 該表格使用兩種方法的組合。

Edit table
Delete table
View properties
Compare versions
Edit schema

**Name** transactions

**Description**

**Database** hbase\_payments

**Classification** Unknown

**Location** s3:// /

**Connection**

**Deprecated** No

**Last updated** Wed Oct 23 12:30:00 GMT-400 2019

**Serde parameters** serialization.format 1

**Table properties** hbase-metadata-flag hbase-metadata-flag

Schema Showing: 1 - 13 of 13 < >

	Column name	Data type	Partition key	Comment
1	summary:order_id	string		summary family, id of the order that this transaction is for
2	summary:customer_id	bigint		summary family, id of the customer that this transaction is for
3	summary:status	string		summary family, status of the transaction
4	summary:auth	string		summary family, auth code for the transaction
5	summary:cc_id	int		summary family, last for of the credit card used for the transaction
6	summary:amount	double		summary family, the amount of the transaction
7	details:fee	double		details family, Fee the transaction network charged to process the tx
8	details:bank	string		details family, the bank baking the transaction
9	details:network	string		details family, the network that was used to clear the tx
10	details:days_payable	int		details family, the number of days this transaction will likely spend in accounts receivable
11	details:latency	int		details family, the latency (millis) of the transaction
12	details:fraud_score	int		details family, the score given to this tx by our fraud algo
13	struct_family	STRUCT		sample column family modeled as a STRUCT and containing two columns (col1, col2)

## 支援的資料類型

連接器會擷取所有 HBase 值作為基本位元組類型。然後，根據您在 AWS Glue 資料目錄中定義資料表的方式，它會將值對應至下表中其中一個 Apache Arrow 資料類型。

AWS Glue 資料類型	Apache Arrow 資料類型
int	INT

AWS Glue 資料類型	Apache Arrow 資料類型
bigint	BIGINT
double	FLOAT8
float	FLOAT4
boolean	BIT
binary	VARBINARY
string	VARCHAR

#### Note

如果您不使用 AWS Glue 來補充中繼資料，則連接器的結構描述推斷僅會使用 BIGINT、FLOAT8、和的資料類型。VARCHAR

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-hbase.yaml](#) 檔案的 Policies 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- AWS Glue Data Catalog— HBase 連接器需要的唯讀存取權，才能取 AWS Glue Data Catalog 得結構描述資訊。
- CloudWatch 防護記錄 — 連接器需要存取 CloudWatch 記錄檔以儲存防護記錄。
- AWS Secrets Manager 讀取存取權 — 如果您選擇將 HBase 端點詳細資料儲存在 Secrets Manager 中，則必須將這些密碼的存取權授與連接器。
- VPC 存取 - 連接器需要能夠將介面連接到 VPC 並能進行分離，以便可進行連接並與 HBase 執行個體通訊。

## 效能

Athena HBase 連接器透過平行讀取每個區域伺服器，嘗試對 HBase 執行個體平行處理查詢。Athena HBase 連接器執行述詞下推，以減少查詢掃描的資料。

Lambda 函數還執行投影下推，以減少查詢掃描的資料。不過，選取資料欄子集有時會導致較長的查詢執行時間。LIMIT 子句可減少掃描的資料量，但是如果您未提供述詞，則應期望具有 LIMIT 子句的 SELECT 查詢掃描至少 16MB 的資料。

HBase 容易導致查詢失敗和可變的查詢執行時間。您可能必須重試查詢多次才能成功。HBase 連接器由於並行而對限流保有彈性。

## 直通查詢

HBase 的連接器支援[直通查詢](#)，並且是以 NoSQL 為基礎。如需使用篩選來查詢 Apache HBase 的詳細資訊，請參閱 Apache 文件中的[篩選器語言](#)。

若要使用 HBase 的直通查詢，請使用下列語法：

```
SELECT * FROM TABLE(  
    system.query(  
        database => 'database_name',  
        collection => 'collection_name',  
        filter => '{query_syntax}'  
    ))
```

下列範例 HBase 的直通查詢篩選器適用於資料庫 employee 集中年齡在 24 或 30 歲的 default 員工。

```
SELECT * FROM TABLE(  
    system.query(  
        DATABASE => 'default',  
        COLLECTION => 'employee',  
        FILTER => 'SingleColumnValueFilter('personaldata', 'age', =,  
'binary:30')' ||  
                ' OR SingleColumnValueFilter('personaldata', 'age', =,  
'binary:24')'  
    ))
```

## 授權資訊

Amazon Athena HBase 連接器專案是依據 [Apache-2.0 License](#) 來授權的。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena Hortonworks 連接器

適用於 Hortonworks 的 Amazon Athena 連接器可讓 Amazon Athena 能夠在 Cloudera [Hortonworks](#) 資料平台上執行 SQL 查詢。連接器可將您的 Athena SQL 查詢轉換為其同等的 HiveQL 語法。

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。

### 條款

下列術語與 Hortonworks Hive 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Hortonworks Hive 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
hive://${jdbc_connection_string}
```

### 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	HiveMuxCompositeHandler
中繼資料處理常式	HiveMuxMetadataHandler
記錄處理常式	HiveMuxRecordHandler

### 多工處理常式參數

參數	描述
<code>\$<i>catalog</i>_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>myhivecatalog</code> ，則環境變數名稱為 <code>myhivecatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${ <i>AWS_LAMBDA_FUNCTION_NAME</i> }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Hive MUX Lambda 函數：hive1 (預設) 和 hive2。

屬性	Value
default	hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}
hive_catalog1_connection_string	hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}
hive_catalog2_connection_string	hive://jdbc:hive2://hive2:10000/default?UID=sample&PWD=sample

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

## 帶有秘密名稱的连接字串範例

以下字串具有秘密名稱 `${Test/RDS/hive1host}`。

```
hive://jdbc:hive2://hive1host:10000/default?...&${Test/RDS/hive1host}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
hive://jdbc:hive2://hive1host:10000/default?...&UID=sample2&PWD=sample2&...
```

目前，Hortonworks Hive 連接器可辨識 UID 和 PWD JDBC 屬性。

## 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Hortonworks Hive 執行個體。

處理常式類型	類別
複合處理常式	HiveCompositeHandler
中繼資料處理常式	HiveMetadataHandler
記錄處理常式	HiveRecordHandler

## 單一連接處理常式參數

參數	描述
default	必要。預設的连接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 Hortonworks Hive 執行個體。

屬性	Value
default	hive://jdbc:hive2://hive1host:10000/default?secret=\${Test/RDS/hive1host}

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC、Hortonworks Hive 和 Arrow 的相應資料類型。

JDBC	Hortonworks Hive	Arrow
Boolean	Boolean	位元
Integer	TINYINT	Tiny
Short	SMALLINT	Smallint
Integer	INT	Int
Long	BIGINT	Bigint
float	float4	Float4

JDBC	Hortonworks Hive	Arrow
Double	float8	Float8
日期	日期	DateDay
時間戳記	timestamp	DateMilli
字串	VARCHAR	Varchar
位元組	位元組	Varbinary
BigDecimal	Decimal (小數)	Decimal (小數)
ARRAY	N/A (請參閱備註)	清單

#### Note

目前，Hortonworks Hive 不支援彙總類型 ARRAY、MAP、STRUCT 或 UNIONTYPE。彙總類型的資料欄在 SQL 中被視為 VARCHAR 資料欄。

## 分割區和分隔

分割區用於決定如何產生連接器的分割。Athena 建構了類型 `varchar` 的合成資料欄，表示資料表的分割結構，以幫助連接器產生分割。連接器不會修改實際的資料表定義。

## 效能

Hortonworks Hive 支援靜態分割區。Athena Hortonworks Hive 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大型的資料集，強烈建議使用靜態分割。選取資料欄子集可大幅加速查詢執行期，並減少掃描的資料。Hortonworks Hive 連接器由於並行而對限流保有彈性。

Athena Hortonworks Hive 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Hortonworks Hive 連接器可以結合這些表達式，並將它們直接推送到 Hortonworks Hive，以增強功能並減少掃描的資料量。

下列 Athena Hortonworks Hive 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## 直通查詢

Hive 連接器支援[直通](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要使用傳遞查詢搭配 Hortonworks 配置單元，您可以使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 Hortonworks Hive 中的資料來源。查詢會選取資料 `customer` 表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱網站上的 [pom.xml](#) 檔案。GitHub

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena Apache Kafka 連接器

適用於 Apache Kafka 的 Amazon Athena 連接器可讓 Amazon Athena 能夠對 Apache Kafka 主題執行 SQL 查詢。使用此連接器可以在 Athena 中以資料表的形式檢視 [Apache Kafka](#) 主題，並以資料列的形式檢視訊息。

## 必要條件

使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 篩選條件中的日期和時間戳記資料類型必須轉換為適當的資料類型。
- 日期和時間戳記資料類型不受 CSV 檔案類型支援，且會被視為 varchar 值。
- 不支援映射至巢狀 JSON 欄位。連接器僅映射最上層欄位。

- 連接器不支援複雜類型。複雜類型會轉譯為字串。
- 若要擷取或使用複雜的 JSON 值，請使用 Athena 中提供的 JSON 相關函數。如需詳細資訊，請參閱 [從字串中提取 JSON 數據](#)。
- 連接器不支援存取 Kafka 訊息中繼資料。

## 條款

- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- Kafka 端點 - 與 Kafka 執行個體建立連線的文字字串。

## 叢集相容性

Kafka 連接器可搭配下列叢集類型使用。

- 獨立 Kafka - 直接連接至 Kafka (已進行身分驗證或未進行身分驗證)。
- Confluent - 直接連線至 Confluent Kafka。如需將 Athena 搭配 Confluent Kafka 資料搭配使用的相關資訊，請參閱商業智慧部落格中的 [QuickSight 使用 Amazon Athena 在亞馬遜中視覺化統一資料](#)。AWS

## 連線至 Confluent

連線到 Confluent 需要以下步驟：

1. 從 Confluent 產生一個 API 金鑰。
2. 將 Confluent API 金鑰的使用者名稱和密碼儲存到 AWS Secrets Manager。
3. 提供 Kafka 連接器中 `secrets_manager_secret` 環境變數的密碼名稱。
4. 請遵循本文件 [設定 Kafka 連接器](#) 一節中的步驟。

## 支援的身分驗證方法

連接器支援下列身分驗證方法。

- [SSL](#)
- [SASL/SCRAM](#)

- SASL/PLAIN
- SASL/PLAINTEXT
- NO\_AUTH
- 自我管理的 Kafka 和 Confluent 平台 – SSL、SASL/SCRAM、SASL/PLAINTEXT、NO\_AUTH
- 自我管理的 Kafka 和 Confluent 雲端 – SASL/PLAIN

如需詳細資訊，請參閱 [設定 Athena Kafka 連接器的身分驗證](#)。

### 支援的輸入資料格式

連接器支援以下輸入資料格式。

- JSON
- CSV

### 參數

使用本節提及的 Lambda 環境變數來設定 Athena Kafka 連接器。

- `auth_type` – 指定叢集的身分驗證類型。連接器支援下列身分驗證類型：
  - `NO_AUTH` – 直接連接至 Kafka (例如，連接至透過無需使用身分驗證之 EC2 執行個體部署的 Kafka 叢集)。
  - `SASL_SSL_PLAIN` – 此方法使用 `SASL_SSL` 安全通訊協定和 PLAIN SASL 機制。如需詳細資訊，請參閱 Apache Kafka 文件中的 [SASL 組態](#)。
  - `SASL_PLAINTEXT_PLAIN` – 此方法使用 `SASL_PLAINTEXT` 安全通訊協定和 PLAIN SASL 機制。如需詳細資訊，請參閱 Apache Kafka 文件中的 [SASL 組態](#)。
  - `SASL_SSL_SCRAM_SHA512` – 您可以使用此身分驗證類型來控制對 Apache Kafka 叢集的存取。此方法將使用者名稱和密碼儲存在中 AWS Secrets Manager。秘密必須與 Kafka 叢集相關聯。如需詳細資訊，請參閱 Apache Kafka 文件中的 [使用 SASL/SCRAM 進行身分驗證](#)。
  - `SASL_PLAINTEXT_SCRAM_SHA512` – 此方法使用 `SASL_PLAINTEXT` 安全通訊協定和 `SCRAM_SHA512` SASL 機制。此方法使用儲存在中的使用者名稱和密碼 AWS Secrets Manager。如需詳細資訊，請參閱 Apache Kafka 文件中的 [SASL 組態](#) 一節。
  - `SSL` – SSL 身分驗證使用金鑰存放區和信任存放區檔案來連接 Apache Kafka 叢集。您必須產生信任存放區和金鑰存放區檔案，將其上傳至 Amazon S3 儲存貯體，並在部署連接器時提供 Amazon S3 參考。金鑰存放區、信任存放區和 SSL 金鑰會儲存在中 AWS Secrets Manager。部署連接器

時，您的用戶端必須提供 AWS 秘密金鑰。如需詳細資訊，請參閱 Apache Kafka 文件中的[使用 SSL 進行加密和身分驗證](#)。

如需詳細資訊，請參閱 [設定 Athena Kafka 連接器的身分驗證](#)。

- `certificates_s3_reference` – 包含憑證 (金鑰存放區和信任存放區檔案) 的 Amazon S3 位置。
- `disable_spill_encryption` - (選用) 當設定為 `True` 時，停用溢出加密。預設為 `False`，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用[伺服器端加密](#)。
- `kafka_endpoint` – 提供給 Kafka 的端點詳細資訊。
- `secrets_manager_secret` – 用來儲存憑證的 AWS 秘密的名稱。
- 溢出參數 – Lambda 函數會將不適用記憶體體的資料暫時存放 (「溢出」) 至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。請使用以下資料表中的參數來指定溢出位置。

參數	描述
<code>spill_bucket</code>	必要。Lambda 函數可在其中溢出資料的 Amazon S3 儲存貯體的名稱。
<code>spill_prefix</code>	必要。Lambda 函數可在其中溢出資料的溢出儲存貯體的字首。
<code>spill_put_request_headers</code>	(選用) 用於溢出的 Amazon S3 <code>putObject</code> 請求的請求標頭和值的 JSON 編碼映射 (例如， <code>{"x-amz-server-side-encryption" : "AES256"}</code> )。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

- Subnet IDs – 一個或多個子網路 ID，其對應於 Lambda 函數可用來存取資料來源的子網路。
  - 公有 Kafka 叢集或標準 Confluent 雲端叢集 – 將連接器與具有 NAT 閘道的私有子網路建立關聯。
  - Confluent 雲端叢集與私有連線 – 將連接器與具有對 Confluent 雲端叢集的路由的私有子網路建立關聯。
    - 對於 [AWS Transit Gateway](#)，子網路必須位於連接至 Confluent 雲端使用的同一傳輸閘道的 VPC 中。
    - 對於 [VPC 對等連線](#)，子網路必須位於對等連線至 Confluent 雲端 VPC 的 VPC 中。
    - 對於 [AWS PrivateLink](#)，子網路必須位於已路由至連線至 Confluent 雲端的 VPC 端點的 VPC 中。

**Note**

如果您將連接器部署到 VPC 中以存取私有資源，並且還想要連接到可公開存取的服務 (例如 Confluent)，則必須將連接器與具有 NAT 閘道的私有子網路建立關聯。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [NAT 閘道](#)。

## 支援的資料類型

下表顯示 Kafka 和 Apache Arrow 支援的相應資料類型。

Kafka	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR
TIMESTAMP	毫秒
DATE	DAY
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT
BIGINT	BIGINT
DECIMAL	FLOAT8
DOUBLE	FLOAT8

## 分割區和分隔

Kafka 主題會分為多個分割區。每個分割區都已排序。分割區中的每個訊息都有一個稱為位移的增量 ID。每個 Kafka 分割區可再細分為多個分隔，用於並行處理。資料在 Kafka 叢集中設定的保留期間內可供使用。

## 最佳實務

最佳實務是在您查詢 Athena 時使用述詞下推，如以下範例所示。

```
SELECT *
FROM "kafka_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE integercol = 2147483647
```

```
SELECT *
FROM "kafka_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE timestampcol >= TIMESTAMP '2018-03-25 07:30:58.878'
```

## 設定 Kafka 連接器

您必須先設定 Apache Kafka 叢集、使用 [AWS Glue 結構描述登錄檔](#) 來定義結構描述，以及為連接器設定身分驗證，才能使用連接器。

使用 AWS Glue 結構描述登錄檔時，請注意以下幾點：

- 請確定 AWS Glue 結構描述登錄檔的 Description (描述) 欄位中的文字包含字串 {AthenaFederationKafka}。對於搭配 Amazon Athena 卡夫卡連接器使用的 AWS Glue 登錄，需要此標記字串。
- 為了獲得最佳效能，請僅使用小寫作為資料庫名稱和資料表名稱。使用混合大小寫會導致連接器執行運算密集程度較高的不區分大小寫搜尋。

若要設定您的 Apache 卡夫卡環境和 AWS Glue 結構描述登錄

1. 設定 Apache Kafka 環境。
2. 將 JSON 格式的 Kafka 主題描述檔案 (也就是其結構描述) 上傳至結 AWS Glue 構描述登錄。如需詳細資訊，請參閱 AWS Glue 開發人員指南中的 [與 AWS Glue 結構描述登錄整合](#)。如需範例結構描述，請參閱下一節。

## AWS Glue 結構描述登錄檔的結構描述範例

將結構描述上傳至 [AWS Glue 結構描述登錄檔](#) 時，請使用本節中的範例格式。

### JSON 類型結構描述範例

在下列範例中，要在結構描述登錄檔中建立的 AWS Glue 結構描述指定 json 為的值，dataFormat 並使用 datatypejson 於 topicName。

**Note**

topicName 的值應使用與 Kafka 中的主題名稱相同的大小寫。

```
{
  "topicName": "datatypejson",
  "message": {
    "dataFormat": "json",
    "fields": [
      {
        "name": "intcol",
        "mapping": "intcol",
        "type": "INTEGER"
      },
      {
        "name": "varcharcol",
        "mapping": "varcharcol",
        "type": "VARCHAR"
      },
      {
        "name": "booleancol",
        "mapping": "booleancol",
        "type": "BOOLEAN"
      },
      {
        "name": "bigintcol",
        "mapping": "bigintcol",
        "type": "BIGINT"
      },
      {
        "name": "doublecol",
        "mapping": "doublecol",
        "type": "DOUBLE"
      },
      {
        "name": "smallintcol",
        "mapping": "smallintcol",
        "type": "SMALLINT"
      },
      {
        "name": "tinyintcol",
```

```
    "mapping": "tinyintcol",
    "type": "TINYINT"
  },
  {
    "name": "datecol",
    "mapping": "datecol",
    "type": "DATE",
    "formatHint": "yyyy-MM-dd"
  },
  {
    "name": "timestampcol",
    "mapping": "timestampcol",
    "type": "TIMESTAMP",
    "formatHint": "yyyy-MM-dd HH:mm:ss.SSS"
  }
]
}
}
```

## CSV 類型結構描述範例

在下列範例中，要在結構描述登錄檔中建立的 AWS Glue 結構描述指定 csv 為的值，dataFormat 並使用 datatypecsvbulk 於 topicName。topicName 的值應使用與 Kafka 中的主題名稱相同的大小寫。

```
{
  "topicName": "datatypecsvbulk",
  "message": {
    "dataFormat": "csv",
    "fields": [
      {
        "name": "intcol",
        "type": "INTEGER",
        "mapping": "0"
      },
      {
        "name": "varcharcol",
        "type": "VARCHAR",
        "mapping": "1"
      },
      {
        "name": "booleancol",
        "type": "BOOLEAN",

```

```

    "mapping": "2"
  },
  {
    "name": "bigintcol",
    "type": "BIGINT",
    "mapping": "3"
  },
  {
    "name": "doublecol",
    "type": "DOUBLE",
    "mapping": "4"
  },
  {
    "name": "smallintcol",
    "type": "SMALLINT",
    "mapping": "5"
  },
  {
    "name": "tinyintcol",
    "type": "TINYINT",
    "mapping": "6"
  },
  {
    "name": "floatcol",
    "type": "DOUBLE",
    "mapping": "7"
  }
]
}
}

```

## 設定 Athena Kafka 連接器的身分驗證

您可以使用各種方法對 Apache Kafka 叢集進行身分驗證，包括 SSL、SASL/SCRAM、SASL/PLAIN 和 SASL/PLAINTEXT。

下表顯示了連接器的身分驗證類型，以及每種連接器的安全通訊協定和 SASL 機制。如需有關詳細資訊，請參閱 Apache Kafka 文件的[安全性](#)一節。

auth_type	security.protocol	saslmecanism	叢集類型相容性
SASL_SSL_PLAIN	SASL_SSL	PLAIN	• 自我管理的 Kafka

auth_type	security.protocol	sasl.mechanism	叢集類型相容性
			<ul style="list-style-type: none"> <li>Confluent 平台</li> <li>Confluent 雲端</li> </ul>
SASL_PLAINTEXT_PLAIN	SASL_PLAINTEXT	PLAIN	<ul style="list-style-type: none"> <li>自我管理的 Kafka</li> <li>Confluent 平台</li> </ul>
SASL_SSL_SCRAM_SHA512	SASL_SSL	SCRAM-SHA-512	<ul style="list-style-type: none"> <li>自我管理的 Kafka</li> <li>Confluent 平台</li> </ul>
SASL_PLAINTEXT_SCRAM_SHA512	SASL_PLAINTEXT	SCRAM-SHA-512	<ul style="list-style-type: none"> <li>自我管理的 Kafka</li> <li>Confluent 平台</li> </ul>
SSL	SSL	N/A	<ul style="list-style-type: none"> <li>自我管理的 Kafka</li> <li>Confluent 平台</li> </ul>

## SSL

如果叢集經過 SSL 身分驗證，您必須產生信任存放區和金鑰存放區檔案，並將其上傳到 Amazon S3 儲存貯體。部署連接器時，您必須提供此 Amazon S3 參考。金鑰存放區、信任存放區和 SSL 金鑰會儲存在 AWS Secrets Manager 中。您可以在部署連接器時提供 AWS 私密金鑰。

如需有關在 Secrets Manager 中建立祕密的詳細資訊，請參閱 [建立 AWS Secrets Manager 祕密](#)。

若要使用此身分驗證類型，請設定環境變數，如下表所示。

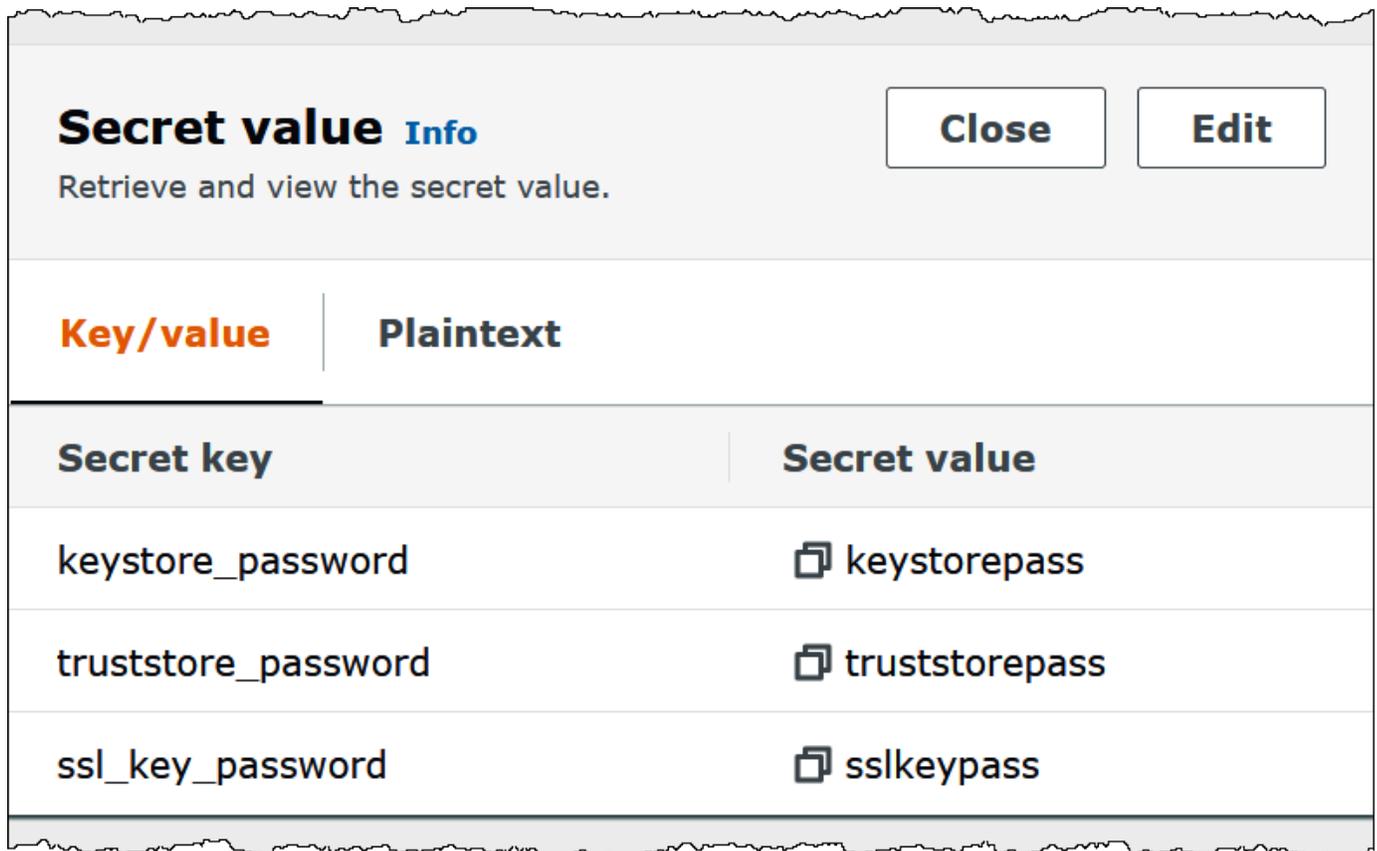
參數	Value
auth_type	SSL
certificates_s3_reference	包含憑證的 Amazon S3 位置。
secrets_manager_secret	您的 AWS 密鑰的名稱。

在 Secrets Manager 中建立祕密之後，您可以在 Secrets Manager 主控台中進行檢視。

若要檢視 Secrets Manager 中的祕密

1. 前往以下位置開啟機密管理員控制台：<https://console.aws.amazon.com/secretsmanager/>。
2. 在導覽窗格中，選擇 Secrets (祕密)。
3. 在 Secrets (祕密) 頁面中，選擇祕密的連結。
4. 在祕密的詳細資訊頁面上，選擇 Retrieve secret value (擷取祕密值)。

下圖顯示了具有三個金鑰/值對的祕密範例：keystore\_password、truststore\_password 和 ssl\_key\_password。



如需有關使用 SSL with Kafka 的詳細資訊，請參閱 Apache Kafka 文件中的[使用 SSL 進行加密和身分驗證](#)。

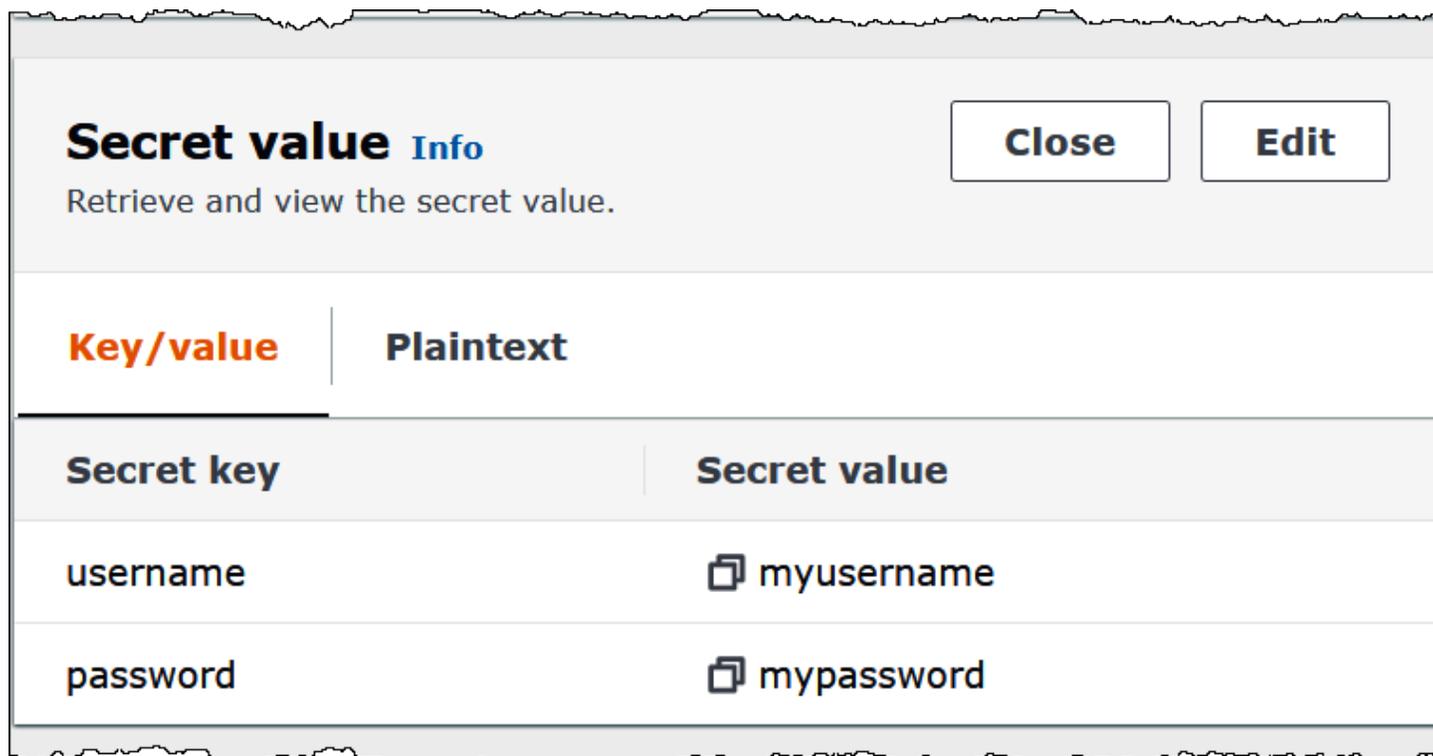
## SASL/SCRAM

如果您的叢集使用 SCRAM 身分驗證，則請在部署連接器時提供與叢集相關聯的 Secrets Manager 金鑰。使用者的 AWS 憑證 (祕密金鑰和存取金鑰) 可用於向叢集進行身分驗證。

設定環境變數，如下表所示。

參數	Value
auth_type	SASL_SSL_SCRAM_SHA512
secrets_manager_secret	您的 AWS 密鑰的名稱。

下圖顯示了 Secrets Manager 主控台中含有兩個金鑰/值對的祕密範例：一個用於 username，另一個用於 password。



如需有關使用 SASL/SCRAM with Kafka 的詳細資訊，請參閱 Apache Kafka 文件中的[使用 SASL/SCRAM 進行身分驗證](#)。

### 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

### 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena MSK 連接器

適用於 [Amazon MSK](#) 的 Amazon Athena 連接器可讓 Amazon Athena 能夠對 Apache Kafka 主題執行 SQL 查詢。使用此連接器可以在 Athena 中以資料表的形式檢視 [Apache Kafka](#) 主題，並以資料列的形式檢視訊息。如需詳細資訊，請參閱 [AWS 大數據部落格中的使用 Amazon Athena 在 Amazon MSK 中分析即時串流](#) 資料。

### 必要條件

使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 限制

- 不支援寫入 DDL 操作。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 篩選條件中的日期和時間戳記資料類型必須轉換為適當的資料類型。
- 日期和時間戳記資料類型不受 CSV 檔案類型支援，且會被視為 varchar 值。
- 不支援映射至巢狀 JSON 欄位。連接器僅映射最上層欄位。
- 連接器不支援複雜類型。複雜類型會轉譯為字串。
- 若要擷取或使用複雜的 JSON 值，請使用 Athena 中提供的 JSON 相關函數。如需詳細資訊，請參閱 [從字符串中提取 JSON 數據](#)。
- 連接器不支援存取 Kafka 訊息中繼資料。

### 條款

- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- Kafka 端點 - 與 Kafka 執行個體建立連線的文字字串。

### 叢集相容性

MSK 連接器可搭配下列叢集類型使用。

- MSK 佈建的叢集 – 您可以手動指定、監控和調整叢集容量。
- MSK 無伺服器叢集 – 提供隨需容量，可隨應用程式 I/O 擴展而自動擴展。
- 獨立 Kafka – 直接連接至 Kafka (已進行身分驗證或未進行身分驗證)。

## 支援的身分驗證方法

連接器支援下列身分驗證方法。

- [SASL/IAM](#)
- [SSL](#)
- [SASL/SCRAM](#)
- SASL/PLAIN
- SASL/PLAINTEXT
- NO\_AUTH

如需詳細資訊，請參閱 [設定 Athena MSK 連接器的身分驗證](#)。

## 支援的輸入資料格式

連接器支援以下輸入資料格式。

- JSON
- CSV

## 參數

使用本節提及的 Lambda 環境變數來設定 Athena MSK 連接器。

- auth\_type – 指定叢集的身分驗證類型。連接器支援下列身分驗證類型：
  - NO\_AUTH – 直接連接至 Kafka，而不需要身分驗證 (例如，連接至透過無需使用身分驗證之 EC2 執行個體部署的 Kafka 叢集)。
  - SASL\_SSL\_PLAIN – 此方法使用 SASL\_SSL 安全通訊協定和 PLAIN SASL 機制。
  - SASL\_PLAINTEXT\_PLAIN – 此方法使用 SASL\_PLAINTEXT 安全通訊協定和 PLAIN SASL 機制。

**Note**

Apache Kafka 支援 SASL\_SSL\_PLAIN 和 SASL\_PLAINTEXT\_PLAIN 身分驗證類型，但 Amazon MSK 不支援。

- SASL\_SSL\_AWS\_MSK\_IAM – 適用於 Amazon MSK 的 IAM 存取控制可讓您能夠處理 MSK 叢集的身分驗證和授權。使用者的 AWS 認證 (秘密金鑰和存取金鑰) 會用來連線叢集。如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的 [IAM 存取控制](#)。
- SASL\_SSL\_SCRAM\_SHA512 – 您可以使用此身分驗證類型來控制對 Amazon MSK 叢集的存取。此方法存儲用戶名和密碼 AWS Secrets Manager。祕密必須與 Amazon MSK 叢集相關聯。如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的 [設定 Amazon MSK 叢集的 SASL/SCRAM 身分驗證](#)。
- SSL – SSL 身分驗證使用金鑰存放區和信任存放區檔案來連接 Amazon MSK 叢集。您必須產生信任存放區和金鑰存放區檔案，將其上傳至 Amazon S3 儲存貯體，並在部署連接器時提供 Amazon S3 參考。金鑰存放區、信任存放區和 SSL 金鑰會儲存在 AWS Secrets Manager 中。部署連接器時，您的用戶端必須提供 AWS 秘密金鑰。如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的 [相互 TLS 身分驗證](#)。

如需詳細資訊，請參閱 [設定 Athena MSK 連接器的身分驗證](#)。

- certificates\_s3\_reference – 包含憑證 (金鑰存放區和信任存放區檔案) 的 Amazon S3 位置。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。
- kafka\_endpoint – 提供給 Kafka 的端點詳細資訊。例如，對於 Amazon MSK 叢集，您可以為叢集提供 [引導 URL](#)。
- secrets\_manager\_secret – 用來儲存憑證的 AWS 祕密的名稱。IAM 身分驗證不需要此參數。
- 溢出參數 – Lambda 函數會將不適用記憶體資料暫時存放 (「溢出」) 至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。請使用以下資料表中的參數來指定溢出位置。

參數	描述
spill_bucket	必要。Lambda 函數可在其中溢出資料的 Amazon S3 儲存貯體的名稱。
spill_prefix	必要。Lambda 函數可在其中溢出資料的溢出儲存貯體的字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 Kafka 和 Apache Arrow 支援的相應資料類型。

Kafka	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR
TIMESTAMP	毫秒
DATE	DAY
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT
BIGINT	BIGINT
DECIMAL	FLOAT8
DOUBLE	FLOAT8

## 分割區和分隔

Kafka 主題會分為多個分割區。每個分割區都已排序。分割區中的每個訊息都有一個稱為位移的增量 ID。每個 Kafka 分割區可再細分為多個分隔，用於並行處理。資料在 Kafka 叢集中設定的保留期間內可供使用。

## 最佳實務

最佳實務是在您查詢 Athena 時使用述詞下推，如以下範例所示。

```
SELECT *
FROM "msk_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE integercol = 2147483647
```

```
SELECT *
FROM "msk_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE timestampcol >= TIMESTAMP '2018-03-25 07:30:58.878'
```

## 設定 MSK 連接器

您必須先設定 Amazon MSK 叢集、使用 [AWS Glue 結構描述登錄檔](#) 來定義結構描述，以及為連接器設定身分驗證，才能使用連接器。

### Note

如果您將連接器部署到 VPC 中以存取私有資源，並且還想要連接到可公開存取的服務 (例如 Confluent)，則必須將連接器與具有 NAT 閘道的私有子網路建立關聯。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [NAT 閘道](#)。

使用 AWS Glue 結構描述登錄檔時，請注意以下幾點：

- 請確定 AWS Glue 結構描述登錄檔的 Description (描述) 欄位中的文字包含字串 {AthenaFederationMSK}。對於搭配 Amazon Athena MSK 連接器使用的 AWS Glue 登錄，需要此標記字串。
- 為了獲得最佳效能，請僅使用小寫作為資料庫名稱和資料表名稱。使用混合大小寫會導致連接器執行運算密集程度較高的不區分大小寫搜尋。

## 若要設定您的 Amazon MSK 環境和 AWS Glue 結構描述登錄

1. 設定您的 Amazon MSK 環境。如需相關資訊和步驟，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的[設定 Amazon MSK](#) 和[開始使用 Amazon MSK](#)。
2. 將 JSON 格式的 Kafka 主題描述檔案 (也就是其結構描述) 上傳至結 AWS Glue 構描述登錄。如需詳細資訊，請參閱 AWS Glue 開發人員指南中的[與 AWS Glue 結構描述登錄整合](#)。如需範例結構描述，請參閱下一節。

### AWS Glue 結構描述登錄檔的結構描述範例

將結構描述上傳至 [AWS Glue 結構描述登錄檔](#) 時，請使用本節中的範例格式。

### JSON 類型結構描述範例

在下列範例中，要在結構描述登錄檔中建立的 AWS Glue 結構描述指定 json 為的值，dataFormat 並使用 datatypejson 於 topicName。

#### Note

topicName 的值應使用與 Kafka 中的主題名稱相同的大小寫。

```
{
  "topicName": "datatypejson",
  "message": {
    "dataFormat": "json",
    "fields": [
      {
        "name": "intcol",
        "mapping": "intcol",
        "type": "INTEGER"
      },
      {
        "name": "varcharcol",
        "mapping": "varcharcol",
        "type": "VARCHAR"
      },
      {
        "name": "booleancol",
        "mapping": "booleancol",

```

```
    "type": "BOOLEAN"
  },
  {
    "name": "bigintcol",
    "mapping": "bigintcol",
    "type": "BIGINT"
  },
  {
    "name": "doublecol",
    "mapping": "doublecol",
    "type": "DOUBLE"
  },
  {
    "name": "smallintcol",
    "mapping": "smallintcol",
    "type": "SMALLINT"
  },
  {
    "name": "tinyintcol",
    "mapping": "tinyintcol",
    "type": "TINYINT"
  },
  {
    "name": "datecol",
    "mapping": "datecol",
    "type": "DATE",
    "formatHint": "yyyy-MM-dd"
  },
  {
    "name": "timestampcol",
    "mapping": "timestampcol",
    "type": "TIMESTAMP",
    "formatHint": "yyyy-MM-dd HH:mm:ss.SSS"
  }
]
}
```

## CSV 類型結構描述範例

在下列範例中，要在結構描述登錄檔中建立的 AWS Glue 結構描述指定 csv 為的值，dataFormat 並使用 datatypecsvbulk 於 topicName。topicName 的值應使用與 Kafka 中的主題名稱相同的大小寫。

```
{
  "topicName": "datatypecsvbulk",
  "message": {
    "dataFormat": "csv",
    "fields": [
      {
        "name": "intcol",
        "type": "INTEGER",
        "mapping": "0"
      },
      {
        "name": "varcharcol",
        "type": "VARCHAR",
        "mapping": "1"
      },
      {
        "name": "booleancol",
        "type": "BOOLEAN",
        "mapping": "2"
      },
      {
        "name": "bigintcol",
        "type": "BIGINT",
        "mapping": "3"
      },
      {
        "name": "doublecol",
        "type": "DOUBLE",
        "mapping": "4"
      },
      {
        "name": "smallintcol",
        "type": "SMALLINT",
        "mapping": "5"
      },
      {
        "name": "tinyintcol",
        "type": "TINYINT",
        "mapping": "6"
      },
      {
        "name": "floatcol",
        "type": "DOUBLE",
```

```

        "mapping": "7"
    }
]
}
}

```

## 設定 Athena MSK 連接器的身分驗證

您可以使用各種方法對 Amazon MSK 叢集進行身分驗證，包括 IAM、SSL、SCRAM 和獨立 Kafka。

下表顯示了連接器的身分驗證類型，以及每種連接器的安全通訊協定和 SASL 機制。如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的[Apache Kafka API 的身分驗證和授權](#)。

auth_type	security.protocol	sasl.mechanism
SASL_SSL_PLAIN	SASL_SSL	PLAIN
SASL_PLAINTEXT_PLAIN	SASL_PLAINTEXT	PLAIN
SASL_SSL_AWS_MSK_IAM	SASL_SSL	AWS_MSK_IAM
SASL_SSL_SCRAM_SHA512	SASL_SSL	SCRAM-SHA-512
SSL	SSL	N/A

### Note

Apache Kafka 支援 SASL\_SSL\_PLAIN 和 SASL\_PLAINTEXT\_PLAIN 身分驗證類型，但 Amazon MSK 不支援。

## SASL/IAM

如果叢集使用 IAM 身分驗證，則必須在設定叢集時為使用者設定 IAM 政策。如需詳細資訊，請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的[IAM 存取控制](#)。

若要使用此身分驗證類型，請將連接器的 auth\_type Lambda 環境變數設定為 SASL\_SSL\_AWS\_MSK\_IAM。

## SSL

如果叢集經過 SSL 身分驗證，您必須產生信任存放區和金鑰存放區檔案，並將其上傳到 Amazon S3 儲存貯體。部署連接器時，您必須提供此 Amazon S3 參考。金鑰存放區、信任存放區和 SSL 金鑰會儲存在 AWS Secrets Manager 中。您可以在部署連接器時提供 AWS 私密金鑰。

如需有關在 Secrets Manager 中建立祕密的詳細資訊，請參閱 [建立 AWS Secrets Manager 祕密](#)。

若要使用此身分驗證類型，請設定環境變數，如下表所示。

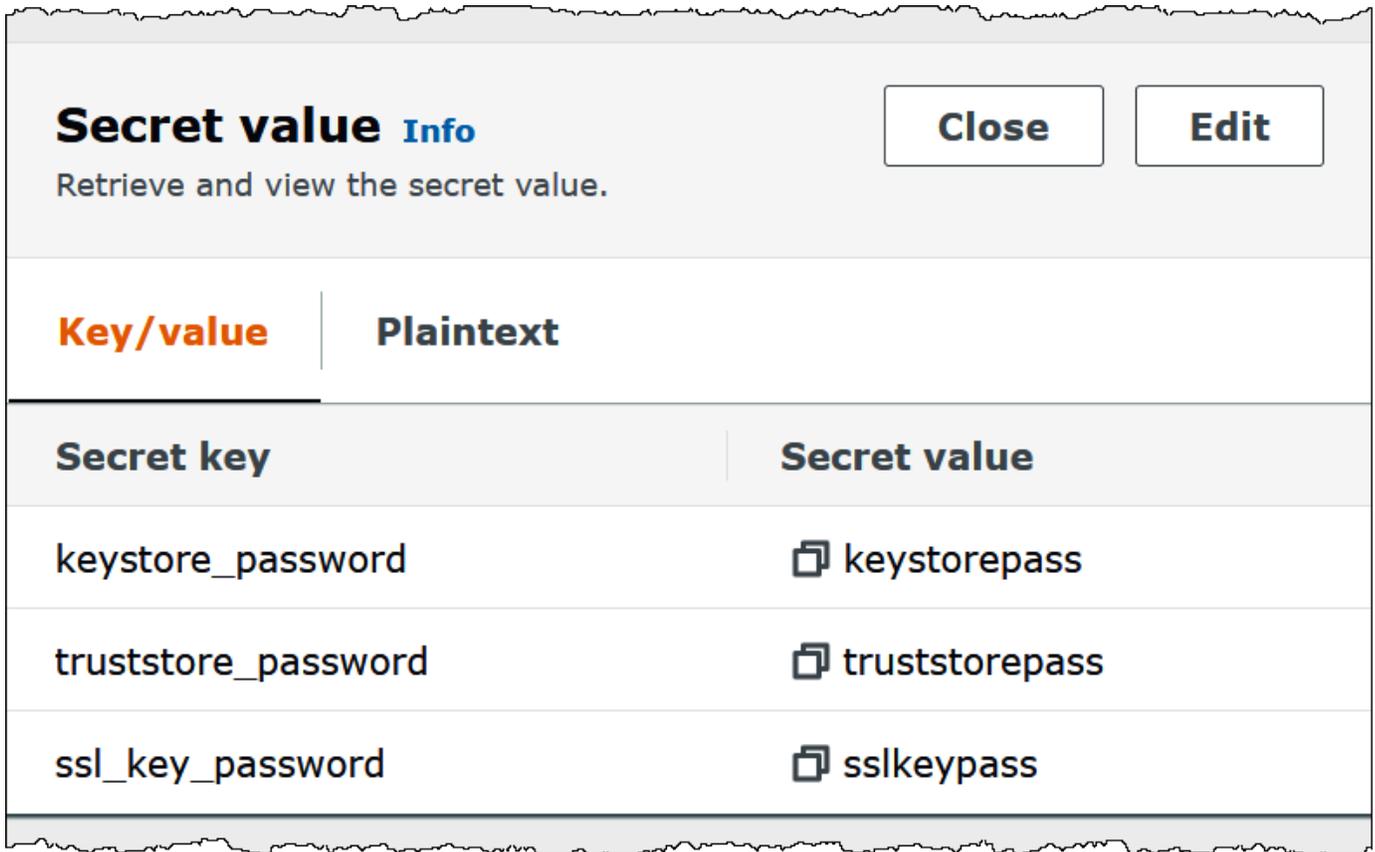
參數	Value
auth_type	SSL
certificates_s3_reference	包含憑證的 Amazon S3 位置。
secrets_manager_secret	您的 AWS 密鑰的名稱。

在 Secrets Manager 中建立祕密之後，您可以在 Secrets Manager 主控台中進行檢視。

若要檢視 Secrets Manager 中的祕密

1. 前往以下位置開啟機密管理員控制台：<https://console.aws.amazon.com/secretsmanager/>。
2. 在導覽窗格中，選擇 Secrets (祕密)。
3. 在 Secrets (祕密) 頁面中，選擇祕密的連結。
4. 在祕密的詳細資訊頁面上，選擇 Retrieve secret value (擷取祕密值)。

下圖顯示了具有三個金鑰/值對的祕密範例：keystore\_password、truststore\_password 和 ssl\_key\_password。



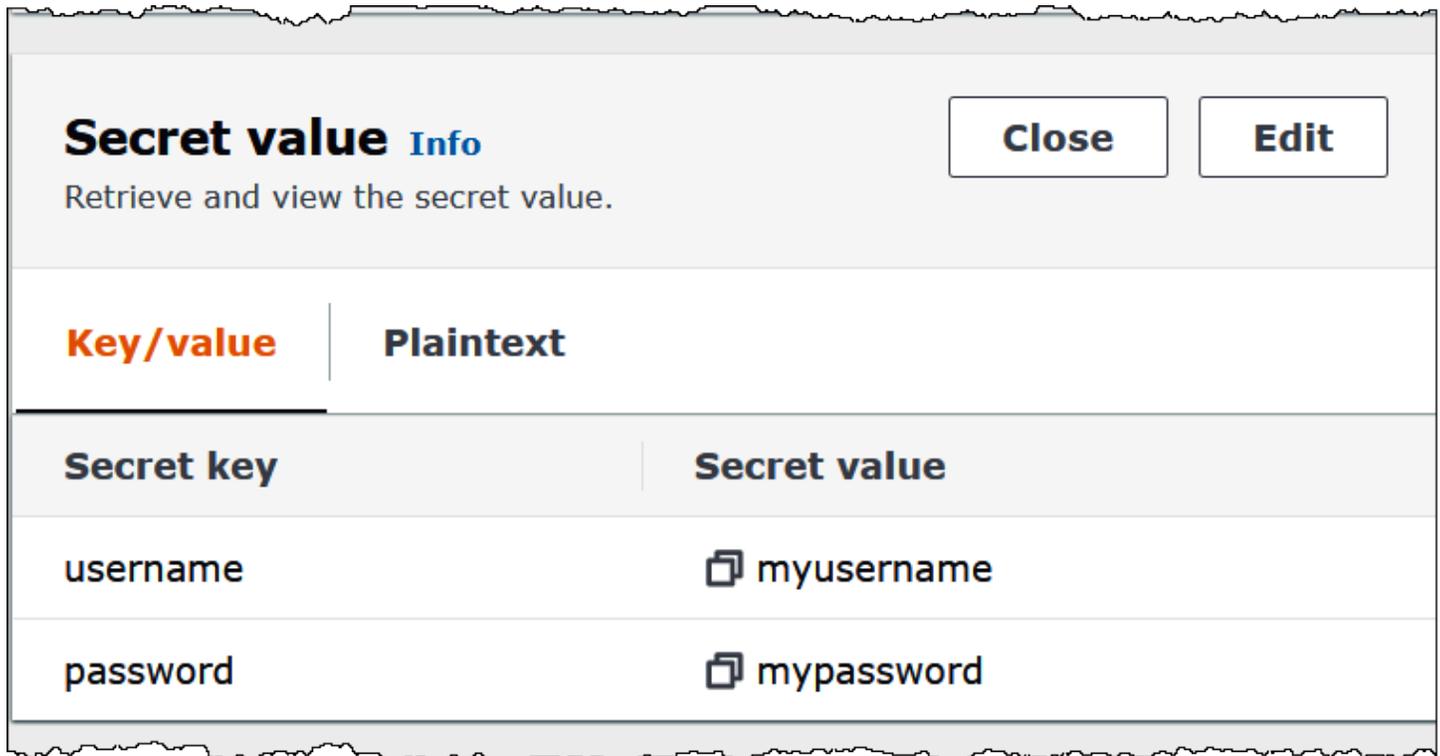
## SASL/SCRAM

如果您的叢集使用 SCRAM 身分驗證，則請在部署連接器時提供與叢集相關聯的 Secrets Manager 金鑰。使用者的 AWS 憑證 (祕密金鑰和存取金鑰) 可用於向叢集進行身分驗證。

設定環境變數，如下表所示。

參數	Value
auth_type	SASL_SSL_SCRAM_SHA512
secrets_manager_secret	您的 AWS 密鑰的名稱。

下圖顯示了 Secrets Manager 主控台中含有兩個金鑰/值對的祕密範例：一個用於 username，另一個用於 password。



### 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

### 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena MySQL 連接器

Amazon Athena Lambda MySQL 連接器讓 Amazon Athena 能夠存取 MySQL 資料庫。

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。
- 您必須先設定 VPC 和安全群組，才能使用此連接器。如需詳細資訊，請參閱 [為資料來源連接器建立 VPC](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 由於 Athena 將查詢轉換為小寫，因此 MySQL 資料表名稱必須為小寫。例如，Athena 針對名為 myTable 的資料表進行查詢會失敗。

## 條款

下列術語與 MySQL 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是connection\_string屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 MySQL 連接器。

## 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
mysql://${jdbc_connection_string}
```

**Note**

如果在對 MySQL 資料表進行 SELECT 查詢時，收到錯誤 `java.sql.SQLException: Zero date value prohibited (java.sql.SQLException : 禁止零日期值)`，請將以下參數新增到您的連接字串中：

```
zeroDateTimeBehavior=convertToNull
```

[有關更多信息，請參閱嘗試從 .com 上的 MySQL 表中選擇時錯誤「禁止零日期值」。GitHub](#)

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	MySqlMuxCompositeHandler
中繼資料處理常式	MySqlMuxMetadataHandler
記錄處理常式	MySqlMuxRecordHandler

## 多工處理常式參數

參數	描述
<code><i>\$catalog_connection_string</i></code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mymysqlcatalog</code> ，則環境變數名稱為 <code>mymysqlcatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 MySQL MUX Lambda 函數：mysql1(預設值) 和mysql2。

屬性	Value
default	mysql://jdbc:mysql://mysql2 .host:3333/default? user=samp le2&password=sample2
mysql_catalog1_connection_string	mysql://jdbc:mysql://mysql1 .host:3306/default?\${Test/RDS/ MySQL1}
mysql_catalog2_connection_string	mysql://jdbc:mysql://mysql2 .host:3333/default? user=samp le2&password=sample2

### 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

#### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

### 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${Test/RDS/MySQL1}`。

```
mysql://jdbc:mysql://mysql1.host:3306/default?...&${Test/RDS/MySQL1}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
mysql://jdbc:mysql://mysql1host:3306/default?...&user=sample2&password=sample2&...
```

目前，MySQL 連接器可辨識 `user` 和 `password` JDBC 屬性。

### 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 MySQL 執行個體。

處理常式類型	類別
複合處理常式	MySQLCompositeHandler
中繼資料處理常式	MySQLMetadataHandler
記錄處理常式	MySQLRecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 MySQL 執行個體。

屬性	Value
default	mysql://mysql11.host:3306/default?secret=Test/RDS/ MySQL1

### 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

### 支援的資料類型

下表顯示 JDBC 和 Arrow 的相應資料類型。

JDBC	Arrow
Boolean	位元
Integer	Tiny
Short	Smallint
Integer	Int

JDBC	Arrow
Long	Bigint
float	Float4
Double	Float8
日期	DateDay
時間戳記	DateMilli
字串	Varchar
位元組	Varbinary
BigDecimal	Decimal (小數)
ARRAY	清單

## 分割區和分隔

分割區用於決定如何產生連接器的分割。Athena 建構了類型 `varchar` 的合成資料欄，表示資料表的分割結構，以幫助連接器產生分割。連接器不會修改實際的資料表定義。

## 效能

MySQL 支援原生分割區。Athena MySQL 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大的資料集，強烈建議使用原生分割。

Athena MySQL 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena MySQL 連接器可以合併這些表達式，並將其直接推送至 MySQL，以增強功能並減少掃描的資料量。

下列 Athena MySQL 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

如需有關使用述詞下推以改善聯合查詢中的效能 (包括 MySQL) 的文章，請參閱 AWS 大數據部落格中的[使用 Amazon Athena 中的述詞下推改善聯合查詢](#)。

## 直通查詢

MySQL 連接器支援[直通查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

要使用 MySQL 的直通查詢，您可以使用以下語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

以下示例查詢將查詢向下推送到 MySQL 中的數據源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 GitHub 網站上 MySQL 連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena Neptune 連接器

Amazon Neptune 是快速、可靠、全受管的圖形資料庫服務，可讓您輕鬆建置和執行搭配高度連線資料集使用的應用程式。專門打造的 Neptune 高效能圖形資料庫引擎可將數十億筆關係以最佳化方式存放，且查詢圖形時只會產生數毫秒的延遲。如需詳細資訊，請參閱《[Neptune 使用者指南](#)》。

Amazon Athena Neptune 連接器讓 Athena 能夠與您的 Neptune 圖形資料庫執行個體通訊，以便使用 SQL 查詢來存取您的 Neptune 圖形資料。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 `AWS Glue Data Catalog`

## 必要條件

使用 Neptune 連接器需要以下三個步驟。

- 設定 Neptune 叢集
- 設置一個 AWS Glue Data Catalog
- 將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。如需部署 Neptune 連接器的其他詳細資訊，請參閱在 GitHub .com 上的[部署 Amazon Athena Neptune 連接器](#)。

## 限制

目前，Neptune 連接器具有以下限制。

- 不支援投影資料欄，包括主索引鍵 (ID)。

## 設定 Neptune 叢集

如果您沒有想要使用的現有 Amazon Neptune 叢集和屬性圖資料集，則必須設定一個。

確保在託管 Neptune 叢集的 VPC 中具有網際網路閘道和 NAT 閘道。Neptune 連接器 Lambda 函數使用的私有子網應具有透過此 NAT 閘道連線至網際網路的路由。Neptune 連接器 Lambda 函數使用 NAT 閘道與之通訊 AWS Glue。

如需設定新 Neptune 叢集並使用範例資料集載入該叢集的指示，請參閱 GitHub .com 上的[範例 Neptune 叢集設定](#)。

## 設定 AWS Glue Data Catalog

與傳統的關聯式資料存放區不同，Neptune 圖形資料庫節點和邊緣不使用集結構描述。每個條目可以有不同的欄位和資料類型。但是，由於 Neptune 連接器會從擷取中繼資料 AWS Glue Data Catalog，因此您必須建立具有具有所需結構描述之表格的資料 AWS Glue 庫。建立 AWS Glue 資料庫和資料表之後，連接器會填充可從 Athena 查詢的資料表清單。

## 啟用不區分大小寫的資料欄比對

若要使用正確的大小寫來解析 Neptune 資料表中的資料行名稱，即使資料行名稱都是較低的大小寫 AWS Glue，您可以設定 Neptune 連接器以進行不區分大小寫的比對。

若要啟用此功能，請將 Neptune 連接器 Lambda 函數環境變數 `enable_caseinsensitivematch` 設定為 `true`。

## 為套管表名稱指定 AWS Glue glabel 表參數

因為只 AWS Glue 支援小寫資料表名稱，因此當您為 Neptune 建立 `glabel` AWS Glue 資料表且 Neptune AWS Glue 資料表名稱包含大小寫時，請務必指定 `table` 參數。

在您的 AWS Glue 表定義中，包括 `glabel` 參數，並使用其原始大小寫將其值設置為表名。這可確保在與 Neptune 表 AWS Glue 互動時保留正確的外殼。下列範例會將 `glabel` 的值設定為資料表名稱 `Airport`。

```
glabel = Airport
```

Table properties (3)	
Key	Value
separatorChar	,
componenttype	vertex
glabel	Airport

如需有關設定與 Neptune 搭配 AWS Glue Data Catalog 使用的詳細資訊，請參閱在 GitHub .com 上[設定 AWS Glue 目錄](#)。

## 效能

Athena Neptune 連接器執行述詞下推，以減少查詢掃描的資料。不過，使用主索引鍵的述詞會導致查詢失敗。LIMIT 子句可減少掃描的資料量，但是如果您未提供述詞，則應期望具有 LIMIT 子句的 SELECT 查詢掃描至少 16 MB 的資料。Neptune 連接器由於並行而對限流保有彈性。

## 直通查詢

Neptune 連接器支援[直通查詢](#)。您可以使用此功能在屬性圖上執行 Gemlin 查詢，以及對 RDF 資料執行 SPARQL 查詢。

若要使用 Neptune 建立直通查詢，請使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    DATABASE => 'database_name',
    COLLECTION => 'collection_name',
    QUERY => 'query_string'
  ))
```

下面的例子 Neptune 直通查詢過濾器過濾器與代碼ATL的機場。雙引號用於轉義。

```
SELECT * FROM TABLE(
  system.query(
    DATABASE => 'graph-database',
    COLLECTION => 'airport',
    QUERY => 'g.V().has(''airport'', ''code'', ''ATL'').valueMap()'
  ))
```

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena OpenSearch 連

### OpenSearch 服務

Amazon Athena OpenSearch 連接器可讓 Amazon Athena 與您的 OpenSearch 執行個體通訊，以便您可以使用 SQL 查詢 OpenSearch 資料。

#### Note

由於已知問題，OpenSearch 連接器無法與 VPC 搭配使用。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 `AWS Glue Data Catalog`

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 條款

下列與 OpenSearch 連接器有關的術語。

- 網域 — 此連接器與執行個體端點關聯的名 OpenSearch 稱。該網域也用作資料庫名稱。對於 OpenSearch Amazon OpenSearch 服務中定義的執行個體，該網域是可自動探索的。對於其他執行個體，您必須提供網域名稱和端點之間的映射。
- Index — 在 OpenSearch 執行個體中定義的資料庫表格。
- 映射 - 如果索引是資料庫資料表，則映射為其結構描述 (亦即其欄位和屬性的定義)。

此連接器支援從執行個 OpenSearch 體和從中擷取中繼資料 AWS Glue Data Catalog。如果連接器找到與您的 OpenSearch 網域和索引名稱相符的 AWS Glue 資料庫和資料表，則連接器會嘗試將它們用於結構描述定義。我們建議您建立 AWS Glue 資料表，使其成為 OpenSearch 索引中所有欄位的超集合。

- 文件 - 資料庫資料表中的記錄。

- 資料串流 – 由多個後備索引組成的以時間為基礎的資料。如需詳細資訊，請參閱 OpenSearch 文件中的[資料串流](#)和 Amazon Ser OpenSearch vice 開發人員指南中的[資料串流入門](#)。

### Note

由於資料串流索引是透過開放式搜尋在內部建立和管理的，因此連接器會從第一個可用索引中選擇結構描述映射。因此，我們強烈建議您將 AWS Glue 表格設定為補充中繼資料來源。如需詳細資訊，請參閱 [設定資料庫和資料表 AWS Glue](#)。

## 參數

使用本節中的 Lambda 環境變數來設定 OpenSearch 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- spill\_put\_request\_headers - (選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"})。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- kms\_key\_id - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 a7e63k4b-81oc-40db-a2a1-4d0en2cd8331，您可以指定 KMS 金鑰 ID。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。
- disabled\_le\_glue — (選擇性) 如果存在且設定為 true，則連接器不會嘗試從中擷取補充中繼資料。AWS Glue
- query\_timeout\_cluster - 產生平行掃描時使用的叢集運作狀態查詢的逾時時間 (以秒為單位)。
- query\_timeout\_search - 從索引中擷取文件時使用的搜尋查詢的逾時時間 (以秒為單位)。
- auto\_discover\_endpoint - 布林值。預設值為 true。當您使用 Amazon OpenSearch 服務並將此參數設定為 true 時，連接器可以透過在服 OpenSearch 務上呼叫適當的描述或列出 API 操作來自動探索您的網域和端點。對於任何其他類型的 OpenSearch 執行個體 (例如，自我託管)，您必須在 domain\_mapping 變數中指定關聯的網域端點。如果 auto\_discover\_endpoint=true，連接器會使用 AWS 認證向 OpenSearch 服務進行驗證。否則，連接器會 AWS Secrets Manager 透過 domain\_mapping 變數擷取使用者名稱和密碼認證。

- `domain_mapping` - 僅在 `auto_discover_endpoint` 設定為 `false` 並定義網域名稱與其關聯端點之間的映射時使用。此 `domain_mapping` 變數可以容納下列格式的多個 OpenSearch 端點：

```
domain1=endpoint1,domain2=endpoint2,domain3=endpoint3,...
```

為了向 OpenSearch 端點進行驗證，連接器支援使用從中 AWS Secrets Manager 擷取的使用者名稱和密碼 `${SecretName}` 的格式插入的替代字串。運算式結尾的冒號 (:) 可做為端點其餘部分的分隔符號。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

下列範例使用 `opensearch-creds` 秘密。

```
movies=https://${opensearch-creds}:search-movies-ne...qu.us-east-1.es.amazonaws.com
```

在執行時，`${opensearch-creds}` 呈現為使用者名稱和密碼，如以下範例所示。

```
movies=https://myusername@mypassword:search-movies-ne...qu.us-east-1.es.amazonaws.com
```

在 `domain_mapping` 參數中，每個網域端點對可以使用不同的秘密。秘密本身必須指定為 `user_name@password` 格式。雖然密碼可能包含內嵌式 @ 符號，但第一個 @ 用作 `user_name` 的分隔符號。

還需要注意的是，此連接器使用逗號 (,) 和等號 (=) 作為網域端點對的分隔符號。因此，您不應該在儲存的秘密中的任何位置使用它們。

## 設定資料庫和資料表 AWS Glue

連接器使用 AWS Glue 或 OpenSearch 取得中繼資料資訊。您可以將 AWS Glue 表格設定為補充中繼資料定義來源。若要啟用此功能，請定義與您要補充之來源的網域和索引相符的 AWS Glue 資料庫和表格。連接器也可以擷取指定索引的對應，以利用 OpenSearch 執行個體中儲存的中繼資料定義。

## 定義陣列的中繼資料 OpenSearch

OpenSearch 沒有專用的陣列資料類型。任何欄位都可以包含零個或多個值，只要它們是相同的資料類型。如果要用 OpenSearch 作中繼資料定義來源，則必須為 Athena 搭配使用的所有索引定義 `_meta` 屬性，以便視為清單或陣列的欄位。如果您無法完成此步驟，查詢只會傳回串列欄位中的第一個元素。當您指定 `_meta` 屬性時，欄位名稱應完全符合巢狀 JSON 結構 (例如，`address.street`，其中 `street` 是 `address` 結構內的巢狀欄位)。

以下範例定義 `movies` 資料表中的 `actor` 和 `genre` 清單。

```
PUT movies/_mapping
{
  "_meta": {
    "actor": "list",
    "genre": "list"
  }
}
```

## 資料類型

OpenSearch 連接器可以從執行個體 AWS Glue 或 OpenSearch 執行個體擷取中繼資料定義。連接器使用下表中的映射，將定義轉換為 Apache Arrow 資料類型，包括下一節中所述的幾點。

OpenSearch	Apache Arrow	AWS Glue
文字、關鍵字、二進位	VARCHAR	string
長整數	BIGINT	bigint
scaled_float	BIGINT	SCALED_FLOAT(...)
integer	INT	int
short	SMALLINT	smallint
byte	TINYINT	tinyint
double	FLOAT8	double
float, half_float	FLOAT4	float

OpenSearch	Apache Arrow	AWS Glue
boolean	BIT	boolean
date, date_nanos	DATEMILLI	timestamp
JSON 結構	STRUCT	STRUCT
_meta (如需詳細資訊，請參閱 <a href="#">定義陣列的中繼資料 OpenSearch</a> 一節)。	LIST	ARRAY

### 資料類型的注意事項

- 目前，連接器僅支援上表中列出的 OpenSearch 和 AWS Glue 資料類型。
- `scaled_float` 是由固定雙精度比例係數調整的浮點數，在 Apache Arrow 中表示為 `BIGINT`。例如，若比例係數為 100，則 0.756 會四捨五入為 76。
- **##scaled\_float#### AWS Glue#####array##### SCALD\_FLOAT (####) #####**

以下範例有效：

```
SCALD_FLOAT(10.51)
SCALD_FLOAT(100)
SCALD_FLOAT(100.0)
```

以下範例無效：

```
SCALD_FLOAT(10.)
SCALD_FLOAT(.5)
```

- 從 `date_nanos` 轉換為 `DATEMILLI` 時，奈秒四捨五入為最接近的毫秒。`date` 和 `date_nanos` 的有效值包括但不限於下列格式：

```
"2020-05-18T10:15:30.123456789"
"2020-05-15T06:50:01.123Z"
"2020-05-15T06:49:30.123-05:00"
1589525370001 (epoch milliseconds)
```

- A OpenSearch binary 是使用編碼的二進制值的字符串表示，Base64並轉換為VARCHAR.

## 執行 SQL 查詢

以下是您可以搭配此連接器使用的 DDL 查詢範例。在這些範例中，*function\_name* 對應於 Lambda 函數的名稱，*domain* 是您要查詢的網域名稱，*index* 是索引的名稱。

```
SHOW DATABASES in `lambda:function_name`
```

```
SHOW TABLES in `lambda:function_name`.`domain`
```

```
DESCRIBE `lambda:function_name`.`domain`.`index`
```

## 效能

Athena OpenSearch 連接器支援以碎片為基礎的 parallel 掃描。連接器會使用從 OpenSearch 執行個體擷取的叢集健全狀況資訊，針對文件搜尋查詢產生多個要求。會針對每個碎片分割請求並同時執行。

連接器也會下推述詞，作為其文件搜尋查詢的一部分。下列範例查詢和述詞顯示連接器如何使用述詞下推。

## 查詢

```
SELECT * FROM "lambda:elasticsearch".movies.movies  
WHERE year >= 1955 AND year <= 1962 OR year = 1996
```

## 述詞

```
(_exists_:year) AND year:([1955 TO 1962] OR 1996)
```

## 直通查詢

OpenSearch 連接器支援[傳遞查詢](#)，並使用查詢 DSL 語言。如需有關使用查詢 DSL 進行查詢的詳細資訊，請參閱 Elasticsearch 文件中的[查詢 DSL](#) 或文件中的[查詢 DSL](#)。OpenSearch

若要將傳遞查詢與 OpenSearch 連接器搭配使用，請使用下列語法：

```
SELECT * FROM TABLE(  
    system.query(  
        schema => 'schema_name',
```

```
    index => 'index_name',
    query => "{query_string}"
  ))
```

下列 OpenSearch 範例傳遞查詢篩選 default 結構描述 employee 索引中具有作用中雇用狀態的員工。

```
SELECT * FROM TABLE(
  system.query(
    schema => 'default',
    index => 'employee',
    query => "{ 'bool':{ 'filter':{ 'term':{ 'status': 'active' } } } }"
  ))
```

## 其他資源

- 如需使用 Amazon Athena OpenSearch 連接器在單一查詢中查詢 Amazon OpenSearch 服務和 Amazon S3 中的資料的文章，請參閱 [AWS 大數據部落格中的使用亞馬遜 Athena 的 SQL 查詢亞馬遜 OpenSearch 服務](#) 中的資料。
- 如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena Oracle 連接器

適用於 Oracle 的 Amazon Athena 連接器可讓 Amazon Athena 能夠對內部部署或在 Amazon EC2 或 Amazon RDS 上執行的 Oracle 中存放的資料執行 SQL 查詢。您也可以使用連接器來查詢 [Oracle exadata](#) 上的資料。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。

## 條款

下列術語與 Oracle 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Oracle 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
oracle://${jdbc_connection_string}
```

#### Note

如果您的密碼包含特殊字元 (例如, `some.password`)，當您將密碼傳遞給連線字串時，請以雙引號括住密碼 (例如, `"some.password"`)。如果沒有這樣做，會導致指定的 Oracle URL 無效錯誤。

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	OracleMuxCompositeHandler
中繼資料處理常式	OracleMuxMetadataHandler
記錄處理常式	OracleMuxRecordHandler

## 多工處理常式參數

參數	描述
<code>\$<i>catalog</i>_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>myoraclecatalog</code> ，則環境變數名稱為 <code>myoraclecatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${ <i>AWS_LAMBDA_FUNCTION_NAME</i> }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Oracle MUX Lambda 函數：`oracle1` (預設) 和 `oracle2`。

屬性	Value
<code>default</code>	<code>oracle://jdbc:oracle:thin:\${Test/RDS/Oracle1}@//oracle1.hostname:port/serviceName</code>
<code>oracle_catalog1_connection_string</code>	<code>oracle://jdbc:oracle:thin:\${Test/RDS/Oracle1}@//oracle1.hostname:port/serviceName</code>
<code>oracle_catalog2_connection_string</code>	<code>oracle://jdbc:oracle:thin:\${Test/RDS/Oracle2}@//oracle2.hostname:port/serviceName</code>

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入秘密名稱。連接器將秘密名稱取代之為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

### Note

如果您的密碼包含特殊字元 (例如，some.password)，當您將密碼儲存在 Secrets Manager 時，請以雙引號括住密碼 (例如，"some.password")。如果沒有這樣做，會導致指定的 Oracle URL 無效錯誤。

## 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${Test/RDS/Oracle}`。

```
oracle://jdbc:oracle:thin:${Test/RDS/Oracle}@//hostname:port/servicename
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
oracle://jdbc:oracle:thin:username/password@//hostname:port/servicename
```

目前，Oracle 連接器可辨識 UID 和 PWD JDBC 屬性。

使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Oracle 執行個體。

處理常式類型	類別
複合處理常式	OracleCompositeHandler
中繼資料處理常式	OracleMetadataHandler
記錄處理常式	OracleRecordHandler

單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。
IsFIPSEnabled	選用。設定為true啟用 FIPS 模式時。預設值為 false。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

該連接器支援基於 SSL 的 Amazon RDS 執行個體連接。支援範圍僅限於 Transport Layer Security (TLS) 通訊協定，以及用戶端對伺服器執行的身分驗證作業。Amazon RDS 中不支援相互身分驗證。下表中的第二列顯示使用 SSL 的語法。

下列範例屬性適用於 Lambda 函數所支援的單一 Oracle 執行個體。

屬性	Value
default	oracle://jdbc:oracle:thin:\${Test/RDS/Oracle}@//hostname:port/servicename
	oracle://jdbc:oracle:thin:\${Test/RDS/Oracle}@((DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS) (HOST=<HOST_NAME>)(PORT=)))(CONNECT_DATA=(SID=))(SECURITY=(SSL_SERVER_CERT_DN=)))

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC、Oracle 和 Arrow 的相應資料類型。

JDBC	Oracle	Arrow
Boolean	boolean	位元
Integer	N/A	Tiny
Short	smallint	Smallint

JDBC	Oracle	Arrow
Integer	integer	Int
Long	bigint	Bigint
float	float4	Float4
Double	float8	Float8
日期	日期	DateDay
時間戳記	timestamp	DateMilli
字串	text	Varchar
位元組	位元組	Varbinary
BigDecimal	numeric(p,s)	Decimal (小數)
ARRAY	N/A (請參閱備註)	清單

## 分割區和分隔

分割區用於決定如何產生連接器的分割。Athena 建構了類型 `varchar` 的合成資料欄，表示資料表的分割結構，以幫助連接器產生分割。連接器不會修改實際的資料表定義。

## 效能

Oracle 支援原生分割區。Athena Oracle 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大的資料集，強烈建議使用原生分割。選取資料欄子集可大幅加速查詢執行期，並減少掃描的資料。Oracle 連接器由於並行而對限流保有彈性。但是，查詢執行時間往往會很長。

Athena Oracle 連接器執行述詞下推，以減少查詢掃描的資料。簡單述詞和複雜表達式會下推至連接器，以減少掃描的資料量並減少查詢執行時間。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Oracle 連接器可以合併這些表達式，並將其直接推送至 Oracle，以增強功能並減少掃描的資料量。

下列 Athena Oracle 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## 直通查詢

Oracle 連接器支援[傳遞查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要使用 Oracle 的傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

以下示例查詢將查詢向下推送到 Oracle 中的數據源。該查詢選擇表中的所有customer列。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer'
```

```
))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 GitHub .com 上 Oracle 連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena PostgreSQL 連接器

Amazon Athena PostgreSQL 連接器讓 Athena 能夠存取您的 PostgreSQL 資料庫。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。

## 條款

下列術語與 PostgreSQL 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。

- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是connection\_string屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 PostgreSQL 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
postgres://${jdbc_connection_string}
```

### 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	PostGreSqlMuxCompositeHandler
中繼資料處理常式	PostGreSqlMuxMetadataHandler
記錄處理常式	PostGreSqlMuxRecordHandler

### 多工處理常式參數

參數	描述
<code>catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mypostgrescatalog</code> ，則環境變數名稱為 <code>mypostgrescatalog_connection_string</code> 。

參數	描述
default	必要。預設的连接字串。目錄為 <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 PostgreSQL MUX Lambda 函數：postgres1(預設值) 和 postgres2。

屬性	Value
default	postgres://jdbc:postgresql://postgres1.host:5432/default?\${Test/RDS/PostGres1}
postgres_catalog1_connection_string	postgres://jdbc:postgresql://postgres1.host:5432/default?\${Test/RDS/PostGres1}
postgres_catalog2_connection_string	postgres://jdbc:postgresql://postgres2.host:5432/default?user=sample&password=sample

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代之為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

### 帶有秘密名稱的連接字符串範例

以下字符串具有秘密名稱 `${Test/RDS/PostGres1}`。

```
postgres://jdbc:postgresql://postgres1.host:5432/default?...&${Test/RDS/PostGres1}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
postgres://jdbc:postgresql://postgres1.host:5432/  
default?...&user=sample2&password=sample2&...
```

目前，PostgreSQL 連接器可辨識 user 和 password JDBC 屬性。

### 正在啟用 SSL

若要在 PostgreSQL 連線中支援 SSL，請將下列內容附加到您的連線字符串中：

```
&sslmode=verify-ca&sslfactory=org.postgresql.ssl.DefaultJavaSSLFactory
```

### 範例

下列範例連線字符串並未使用 SSL。

```
postgres://jdbc:postgresql://example-asdf-aurora-postgres-endpoint:5432/asdf?  
user=someuser&password=somepassword
```

若要啟用 SSL，請依照下列方式修改字符串。

```
postgres://jdbc:postgresql://example-asdf-aurora-postgres-
endpoint:5432/asdf?user=someuser&password=somepassword&sslmode=verify-
ca&sslfactory=org.postgresql.ssl.DefaultJavaSSLFactory
```

## 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 PostgreSQL 執行個體。

處理常式類型	類別
複合處理常式	PostGreSqlCompositeHandler
中繼資料處理常式	PostGreSqlMetadataHandler
記錄處理常式	PostGreSqlRecordHandler

## 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 PostgreSQL 執行個體。

屬性	Value
default	postgres://jdbc:postgresql://postgres1.host:5432/default?secret=\${Test/RDS/PostgreSQL1}

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示了 JDBC , PostgreSQL 和箭頭的相應數據類型。

JDBC	PostgreSQL	Arrow
Boolean	Boolean	位元
Integer	N/A	Tiny
Short	smallint	Smallint
Integer	integer	Int
Long	bigint	Bigint
float	float4	Float4
Double	float8	Float8
日期	日期	DateDay
時間戳記	timestamp	DateMilli
字串	text	Varchar
位元組	位元組	Varbinary
BigDecimal	numeric(p,s)	Decimal (小數)

JDBC	PostgreSQL	Arrow
ARRAY	N/A (請參閱備註)	清單

### Note

PostgreSQL 連接器支援 ARRAY 類型，但有以下限制：不支援多維陣列 (`<data_type>[] []` 或巢狀陣列)。具有不受支援之 ARRAY 資料類型的資料欄被轉換為字串元素陣列 (`array<varchar>`)。

## 分割區和分隔

分割區用於決定如何產生連接器的分割。Athena 建構了類型 `varchar` 的合成資料欄，表示資料表的分割結構，以幫助連接器產生分割。連接器不會修改實際的資料表定義。

## 效能

PostgreSQL 支援原生分割區。Athena PostgreSQL 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大的資料集，強烈建議使用原生分割。

Athena PostgreSQL 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。不過，選取資料欄子集有時會導致查詢執行期更長。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena PostgreSQL 連接器可以合併這些表達式，並將其直接推送至 PostgreSQL，以增強功能並減少掃描的資料量。

下列 Athena PostgreSQL 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT

- 等

式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL

- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE

- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## 直通查詢

PostgreSQL 連接器支援[直通查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

要使用 PostgreSQL 的傳遞查詢，您可以使用以下語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 PostgreSQL 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 [.com](#) 上之 PostgreSQL 連接器的 [pom.xml](#) 檔案。

### GitHub

如需有關此連接器的其他資訊，請造訪 [GitHub .com](#) 上的[對應網站](#)。

## Amazon Athena Redis 連接器

Amazon Athena Redis 連接器讓 Amazon Athena 能夠與您的 Redis 執行個體通訊，以便使用 SQL 來查詢您的 Redis 資料。您可以使用將 AWS Glue Data Catalog Redis 索引鍵值組對映到虛擬資料表中。

與傳統的關聯式資料存放區不同，Redis 沒有資料表或資料欄概念。相反，Redis 提供索引鍵/值存取模式，其中索引鍵本質上是一個 string，而值是 string、z-set 或 hmap。

您可以使用建立結構描述和設定虛擬資料表。AWS Glue Data Catalog 特殊資料表屬性會告知 Athena Redis 連接器如何將 Redis 索引鍵和值映射到資料表中。如需詳細資訊，請參閱本文件稍後的[設定資料庫和資料表 AWS Glue](#)。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 `AWS Glue Data Catalog`

Amazon Athena Redis 連接器支持 Amazon 記憶數據庫的 Redis 和 Amazon 的 Redis。ElastiCache 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。
- 您必須先設定 VPC 和安全群組，才能使用此連接器。如需詳細資訊，請參閱 [為資料來源連接器建立 VPC](#)。

### 參數

使用本節中的 Lambda 環境變數來設定 Redis 連接器。

- `spill_bucket` - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- `spill_prefix` - (選用) 預設為指定的 `spill_bucket` 中名為 `athena-federation-spill` 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- `spill_put_request_headers` - (選用) 用於溢出的 Amazon S3 `putObject` 請求的請求標頭和值的 JSON 編碼映射 (例如，`{"x-amz-server-side-encryption" : "AES256"}`)。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- `kms_key_id` - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`，您可以指定 KMS 金鑰 ID。

- `disable_spill_encryption` - (選用) 當設定為 `True` 時，停用溢出加密。預設為 `False`，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用[伺服器端加密](#)。
- `glue_catalog` - (選用) 使用此選項可指定[跨帳戶 AWS Glue 目錄](#)。依預設，連接器會嘗試從其自己的 AWS Glue 帳戶取得中繼資料。

## 設定資料庫和資料表 AWS Glue

若要啟用與 Redis 搭配使用的 AWS Glue 表格，您可以在表格上設定下列表格屬性：`redis-endpointredis-value-type`、和`redis-keys-zset`或`redis-key-prefix`。

此外，任何包含 Redis 資料表的 AWS Glue 資料庫都必須`redis-db-flag`在資料庫的 URI 屬性中具有。若要設定 `redis-db-flag` URI 屬性，請使用 AWS Glue 主控台編輯資料庫。

以下清單說明資料表屬性。

- `redis-endpoint` - (必要) 包含此資料表資料之 Redis 伺服器的#機名稱：`###:##` (例如`athena-federation-demo.cache.amazonaws.com:6379`) 或者，您也可以使用 `${Secret_Name}` 作為資料表內容值，將端點或部 AWS Secrets Manager 分端點儲存在中。

### Note

若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權](#)，或具有連線至 Secrets Manager 的 [VPC 端點](#)。

- `redis-keys-zset`— (如果未使用，`redis-key-prefix`則需要) 以逗號分隔的索引鍵清單，其值為 [zset](#) (例如，`active-orders,pending-orders`)。zset 中的每個值都會被視為屬於資料表的索引鍵。必須設定 `redis-keys-zset` 屬性或 `redis-key-prefix` 屬性。
- `redis-key-prefix`— (如果未使用，`redis-keys-zset`則需要) 用逗號分隔的金鑰前置詞清單，以掃描表格中的值 (例如，`accounts-*,acct-`)。必須設定 `redis-key-prefix` 屬性或 `redis-keys-zset` 屬性。
- `redis-value-type`— (必要) 定義`redis-key-prefix`或`redis-keys-zset`所定義之索引鍵值對應至資料表的方式。常值映射到單個欄。zset 也映射到單個欄，但每個索引鍵可以存儲許多列。雜湊可讓每個索引鍵成為包含多個欄的資料列 (例如雜湊、常值或 zset)。
- `redis-ssl-flag`— (選用) 何時`True`，會建立使用 SSL/TLS 的 Redis 連線。預設值為 `False`。
- `redis-cluster-flag`— (選擇性) 啟`True`用叢集 Redis 執行個體的支援時機。預設值為 `False`。

- `redis-db-number`— (選擇性) 僅適用於獨立的非叢集執行個體。) 設定此數字 (例如 1、2 或 3) 以便從非預設 Redis 資料庫中讀取。預設值為 Redis 邏輯資料庫 0。這個數字不是指 Athena 中的資料庫 AWS Glue，而是指 Redis 邏輯資料庫。如需詳細資訊，請參閱 Redis 文件中的 [SELECT 索引](#)。

## 資料類型

Redis 連接器支援以下資料類型。不支援 Redis 資料串流。

- [: 字串](#)
- [雜湊](#)
- [排序集 \(ZSet\)](#)

所有 Redis 值被擷取為 `string` 資料類型。然後根據在 AWS Glue Data Catalog 中定義資料表的方式，將它們轉換為以下 Apache Arrow 資料類型之一。

AWS Glue 資料類型	Apache Arrow 資料類型
<code>int</code>	<code>INT</code>
<code>string</code>	<code>VARCHAR</code>
<code>bigint</code>	<code>BIGINT</code>
<code>double</code>	<code>FLOAT8</code>
<code>float</code>	<code>FLOAT4</code>
<code>smallint</code>	<code>SMALLINT</code>
<code>tinyint</code>	<code>TINYINT</code>
<code>boolean</code>	<code>BIT</code>
<code>binary</code>	<code>VARBINARY</code>

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-redis.yaml](#) 檔案的 `Policies` 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- AWS Glue Data Catalog— Redis 連接器需要的唯讀存取權才能取 AWS Glue Data Catalog 得結構描述資訊。
- CloudWatch 防護記錄 — 連接器需要存取 CloudWatch 記錄檔以儲存防護記錄。
- AWS Secrets Manager 讀取存取權 — 如果您選擇將 Redis 端點詳細資料儲存在 Secrets Manager 中，則必須將這些密碼的存取權授與連接器。
- VPC 存取 - 連接器需要能夠將介面連接到 VPC 並能進行分離，以便可進行連接並與 Redis 執行個體通訊。

## 效能

Athena Redis 連接器會嘗試根據您定義的資料表類型 (例如，zset 索引鍵或字首索引鍵)，對 Redis 執行個體進行並行查詢。

Athena Redis 連接器執行述詞下推，以減少查詢掃描的資料。不過，包含針對主索引鍵的述詞的查詢因逾時而失敗。LIMIT 子句可減少掃描的資料量，但是如果您未提供述詞，則應期望具有 LIMIT 子句的 SELECT 查詢掃描至少 16 MB 的資料。Redis 連接器由於並行而對限流保有彈性。

## 授權資訊

Amazon Athena Redis 連接器專案是依據 [Apache-2.0 License](#) 來授權的。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上[的對應網站](#)。

## Amazon Athena Redshift 連接器

Amazon Athena Redshift 連接器可讓 Amazon Athena 存取您的 Amazon Redshift 和 Amazon Redshift 無伺服器資料庫，包括 Redshift 無伺服器視圖。您可以使用此頁面所述的 JDBC 連線字串組態設定值來連線至任一服務。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 由於 Redshift 不支援外部分割區，因此每次都會擷取由查詢指定的所有資料。

## 條款

下列術語與 Redshift 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Redshift 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
redshift://${jdbc_connection_string}
```

### 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	RedshiftMuxCompositeHandler
中繼資料處理常式	RedshiftMuxMetadataHandler
記錄處理常式	RedshiftMuxRecordHandler

## 多工處理常式參數

參數	描述
<code><i>\$catalog_connection_string</i></code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>myredshif</code> <code>tcatalog</code> ，則環境變數名稱為 <code>myredshif</code> <code>tcatalog_connection_string</code> 。
default	必要。預設的連接字串。目錄為 <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Redshift MUX Lambda 函數：redshift1 (預設) 和 redshift2。

屬性	Value
default	<code>redshift://jdbc:redshift://redshift1.host:5439/dev?user=sample2&amp;password=sample2</code>
<code>redshift_catalog1_connection_string</code>	<code>redshift://jdbc:redshift://redshift1.host:3306/default?\${Test/RDS/Redshift1}</code>
<code>redshift_catalog2_connection_string</code>	<code>redshift://jdbc:redshift://redshift2.host:3333/default?user=sample2&amp;password=sample2</code>

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的 [VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

### 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${Test/RDS/Redshift1}`。

```
redshift://jdbc:redshift://redshift1.host:3306/default?...&${Test/RDS/Redshift1}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
redshift://jdbc:redshift://redshift1.host:3306/  
default?...&user=sample2&password=sample2&...
```

目前，Redshift 連接器可辨識 user 和 password JDBC 屬性。

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC 和 Apache Arrow 的相應資料類型。

JDBC	Arrow
Boolean	位元
Integer	Tiny
Short	Smallint
Integer	Int
Long	Bigint
float	Float4
Double	Float8
日期	DateDay
時間戳記	DateMilli

JDBC	Arrow
字串	Varchar
位元組	Varbinary
BigDecimal	Decimal (小數)
ARRAY	清單

## 分割區和分隔

Redshift 不支援外部分割區。如需效能相關問題的資訊，請參閱 [效能](#)。

## 效能

Athena Redshift 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、ORDER BY 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。不過，選取資料欄子集有時會導致查詢執行期更長。同時執行多個查詢時，Amazon Redshift 特別容易受到查詢執行速度減慢的影響。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 前 N 個查詢

前 N 個查詢會指定結果集的順序，以及傳回的資料列數目的限制。您可以使用這種查詢類型，判斷資料集的前 N 個最大值或前 N 個最小值。透過前 N 個下推，連接器只會向 Athena 傳回 N 個排序的資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Redshift 連接器可以合併這些表達式，並將其直接推送至 Redshift，以增強功能並減少掃描的資料量。

下列 Athena Redshift 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT

- 等

式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL

- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE

- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
ORDER BY col_a DESC
LIMIT 10;
```

如需有關使用述詞下推以改善聯合查詢中的效能 (包括 Amazon Redshift) 的文章，請參閱 AWS 大數據部落格中的[使用 Amazon Athena 中的述詞下推改善聯合查詢](#)。

## 直通查詢

Redshift 連接器支援[傳遞](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要搭配 Redshift 使用傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 Redshift 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 [.com](#) 上 GitHub Redshift 連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub [.com](#) 上的 [對應網站](#)。

## Amazon Athena SAP HANA 連接器

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 在 SAP HANA 中，當物件名稱儲存在 SAP HANA 資料庫中時會轉換為大寫。但是，由於引號中的名稱區分大小寫，因此兩個資料表的名稱可能會相同，分別為小寫和大寫 (例如，EMPLOYEE 和 employee)。

在 Athena 聯合查詢中，結構描述資料表名稱會以小寫形式提供給 Lambda 函數。若要解決此問題，您可以提供 @schemaCase 查詢提示以便從名稱區分大小寫的資料表中擷取資料。以下是具有查詢提示的兩個查詢範例。

```
SELECT *  
FROM "lambda:saphanaconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=upper"
```

```
SELECT *  
FROM "lambda:saphanaconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=lower"
```

## 條款

下列術語與 SAP HANA 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是`connection_string`屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 SAP HANA 連接器。

## 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
saphana://${jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	<code>SaphanaMuxCompositeHandler</code>
中繼資料處理常式	<code>SaphanaMuxMetadataHandler</code>
記錄處理常式	<code>SaphanaMuxRecordHandler</code>

## 多工處理常式參數

參數	描述
<code>\$catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mysaphana catalog</code> ，則環境變數名稱為 <code>mysaphanacatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Saphana MUX Lambda 函數：saphana1 (預設) 和 saphana2。

屬性	Value
<code>default</code>	<code>saphana://jdbc:sap://saphana1.host:port/?\${Test/RDS/Saphana1}</code>
<code>saphana_catalog1_connection_string</code>	<code>saphana://jdbc:sap://saphana1.host:port/?\${Test/RDS/Saphana1}</code>
<code>saphana_catalog2_connection_string</code>	<code>saphana://jdbc:sap://saphana2.host:port/?user=sample2&amp;password=sample2</code>

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

**⚠ Important**

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

**帶有秘密名稱的連接字串範例**

以下字串具有秘密名稱 `${Test/RDS/Saphana1}`。

```
saphana://jdbc:sap://saphana1.host:port/?${Test/RDS/Saphana1}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
saphana://jdbc:sap://saphana1.host:port/?user=sample2&password=sample2&...
```

目前，SAP HANA 連接器可辨識 user 和 password JDBC 屬性。

**使用單一連接處理常式**

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 SAP HANA 執行個體。

處理常式類型	類別
複合處理常式	SaphanaCompositeHandler

處理常式類型	類別
中繼資料處理常式	SaphanaMetadataHandler
記錄處理常式	SaphanaRecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 SAP HANA 執行個體。

屬性	Value
default	saphana://jdbc:sap://saphana1.host:port/?secret=Test/RDS/Saphana1

### 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-

參數	描述
	encryption" : "AES256"} )。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC 和 Apache Arrow 的相應資料類型。

JDBC	Arrow
Boolean	位元
Integer	Tiny
Short	Smallint
Integer	Int
Long	Bigint
float	Float4
Double	Float8
日期	DateDay
時間戳記	DateMilli
字串	Varchar
位元組	Varbinary
BigDecimal	Decimal (小數)
ARRAY	清單

## 資料類型轉換

除了 JDBC 到 Arrow 轉換之外，連接器還會執行一些其他轉換，以使 SAP HANA 來源和 Athena 資料類型相容。這些轉換有助於確保成功執行查詢。下表顯示了這些轉換。

來源資料類型 (SAP HANA)	轉換後的資料類型 (Athena)
DECIMAL	BIGINT
INTEGER	INT
DATE	DATEDAY
TIMESTAMP	DATEMILLI

所有其他不受支援的資料類型都會轉換為 VARCHAR。

## 分割區和分隔

分割區由 Integer 類型的單一分割區資料欄表示。此資料欄包含 SAP HANA 資料表中所定義之分割區的分割區名稱。對於沒有分割區名稱的資料表，會傳回 \*，這相當於一個單一的分割區。分割區等同於分隔。

名稱	Type	描述
PART_ID	Integer	SAP HANA 中的已命名分割區。

## 效能

SAP HANA 支援原生分割區。Athena SAP HANA 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大的資料集，強烈建議使用原生分割。選取資料欄子集可大幅加速查詢執行期，並減少掃描的資料。由於並行，連接器會顯示嚴重限流，且有時會顯示查詢失敗。

Athena SAP HANA 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行時間。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena SAP HANA 連接器可以合併這些表達式，並將其直接推送至 SAP HANA，以增強功能並減少掃描的資料量。

下列 Athena SAP HANA 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## 直通查詢

SAP HANA 連接器支援[直通查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要搭配 SAP HANA 使用傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 SAP HANA 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
```

```
system.query(  
    query => 'SELECT * FROM customer LIMIT 10'  
))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 GitHub .com 上針對 SAP HANA 連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena Snowflake 連接器

適用於 [Snowflake](#) 的 Amazon Athena 連接器可讓 Amazon Athena 使用 JDBC 對存放在 Snowflake SQL 資料庫或 RDS 執行個體中的資料執行 SQL 查詢。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 目前支援採用單一分割方式的 Snowflake 檢視。
- 在 Snowflake 中，由於物件名稱區分大小寫，因此兩個資料表的名稱可能會相同，分別為小寫和大寫 (例如，EMPLOYEE 和 employee)。在 Athena 聯合查詢中，結構描述資料表名稱會以小寫形式提供給 Lambda 函數。若要解決此問題，您可以提供 @schemaCase 查詢提示以便從名稱區分大小寫的資料表中擷取資料。以下是具有查詢提示的兩個查詢範例。

```
SELECT *
```

```
FROM "lambda:snowflakeconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=upper"
```

```
SELECT *  
FROM "lambda:snowflakeconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=lower"
```

## 條款

下列術語與 Snowflake 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 Snowflake 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
snowflake://{jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	SnowflakeMuxCompositeHandler
中繼資料處理常式	SnowflakeMuxMetadataHandler
記錄處理常式	SnowflakeMuxRecordHandler

### 多工處理常式參數

參數	描述
<code>\$catalog_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mysnowflakecatalog</code> ，則環境變數名稱為 <code>mysnowflakecatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Snowflake MUX Lambda 函數：`snowflake1` (預設) 和 `snowflake2`。

屬性	Value
<code>default</code>	<code>snowflake://jdbc:snowflake://snowflake1.host:port/?warehouse=warehousename&amp;db=db1&amp;schema=schema1&amp;\${Test/RDS/Snowflake1}</code>
<code>snowflake_catalog1_connection_string</code>	<code>snowflake://jdbc:snowflake://snowflake1.host:port/?warehouse=warehousename&amp;db=db1&amp;schema=schema1\${Test/RDS/Snowflake1}</code>

屬性	Value
snowflake _catalog2 _connecti on_string	snowflake://jdbc:snowflake://snowflake2.host: port/?warehouse=warehousename&db=db1&schema=s chema1&user=sample2&password=sample2

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

## 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 \${Test/RDS/Snowflake1}。

```
snowflake://jdbc:snowflake://snowflake1.host:port/?
warehouse=warehousename&db=db1&schema=schema1${Test/RDS/Snowflake1}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
snowflake://jdbc:snowflake://snowflake1.host:port/
warehouse=warehousename&db=db1&schema=schema1&user=sample2&password=sample2&...
```

目前，Snowflake 可辨識 `user` 和 `password` JDBC 屬性。它也接受格式為 `username/password` 的使用者名稱和密碼，而不使用金鑰 `user` 或 `password`。

### 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Snowflake 執行個體。

處理常式類型	類別
複合處理常式	SnowflakeCompositeHandler
中繼資料處理常式	SnowflakeMetadataHandler
記錄處理常式	SnowflakeRecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 Snowflake 執行個體。

屬性	Value
default	snowflake://jdbc:snowflake://snowflake1.host:port/?sec ret=Test/RDS/Snowflake1

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC 和 Apache Arrow 的相應資料類型。

JDBC	Arrow
Boolean	位元
Integer	Tiny
Short	Smallint
Integer	Int
Long	Bigint
float	Float4

JDBC	Arrow
Double	Float8
日期	DateDay
時間戳記	DateMilli
字串	Varchar
位元組	Varbinary
BigDecimal	Decimal (小數)
ARRAY	清單

### 資料類型轉換

除了 JDBC 到 Arrow 轉換之外，連接器還會執行一些其他轉換，以使 Snowflake 來源和 Athena 資料類型相容。這些轉換有助於確保成功執行查詢。下表顯示了這些轉換。

來源資料類型 (Snowflake)	轉換後的資料類型 (Athena)
TIMESTAMP	TIMESTAMPMILLI
DATE	TIMESTAMPMILLI
INTEGER	INT
DECIMAL	BIGINT
TIMESTAMP_NTZ	TIMESTAMPMILLI

所有其他不受支援的資料類型都會轉換為 VARCHAR。

### 分割區和分隔

分割區用於決定如何產生連接器的分割。Athena 建構了類型 varchar 的合成資料欄，表示資料表的分割結構，以幫助連接器產生分割。連接器不會修改實際的資料表定義。

若要建立這個合成資料欄和分割區，Athena 需要定義主索引鍵。但是，由於 Snowflake 不會強制執行主索引鍵條件約束，您必須自行強制執行唯一。如果不這樣做會導致 Athena 預設為單一分割。

## 效能

為了獲得最佳效能，請盡可能在查詢中使用篩選條件。此外，我們強烈建議使用原生分割來擷取具有統一分割區分佈的龐大資料集。選取資料欄子集可大幅加速查詢執行期，並減少掃描的資料。Snowflake 連接器由於並行而對限流保有彈性。

Athena Snowflake 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句、簡單述詞和複雜的表達式會下推至連接器，以減少掃描的資料量並縮短查詢執行期。

## LIMIT 子句

LIMIT N 陳述句可減少查詢掃描的資料。透過 LIMIT N 下推，連接器只會向 Athena 傳回 N 個資料列。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Snowflake 連接器可以合併這些表達式，並將其直接推送至 Snowflake，以增強功能並減少掃描的資料量。

下列 Athena Snowflake 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
```

```
LIMIT 10;
```

## 直通查詢

「雪花」連接器支援[傳遞查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要搭配 Snowflake 使用直通查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

下列範例查詢會將查詢向下推送至 Snowflake 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 GitHub .com 上雪花連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上[的對應網站](#)。

## Amazon Athena Microsoft SQL Server 連接器

適用於 [Microsoft SQL Server](#) 的 Amazon Athena 連接器可讓 Amazon Athena 使用 JDBC 對存放在 Microsoft SQL Server 中的資料執行 SQL 查詢。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。
- 在篩選條件中，您必須將 Date 和 Timestamp 資料類型轉換為適當的資料類型。
- 若要搜尋類型 Real 和 Float 的負值，請使用 <= 或者 >= 運算子。
- 不支援 binary、varbinary、image、和 rowversion 資料類型。

## 條款

下列術語與 SQL Server 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是connection\_string屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## 參數

使用本節中的 Lambda 環境變數來設定 SQL Server 連接器。

### 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
sqlserver://${jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	SqlServerMuxCompositeHandler
中繼資料處理常式	SqlServerMuxMetadataHandler
記錄處理常式	SqlServerMuxRecordHandler

## 多工處理常式參數

參數	描述
<code><i>\$catalog_connection_string</i></code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>mysqlservercatalog</code> ，則環境變數名稱為 <code>mysqlservercatalog_connection_string</code> 。
default	必要。預設的連接字串。目錄為 <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 SqlServer MUX Lambda 函數：`sqlserver1`(預設值)和`sqlserver2`。

屬性	Value
default	<code>sqlserver://jdbc:sqlserver://sqlserver1.<i>hostname:port</i>;databaseName= &lt;database_name&gt; ;\${secret1_name }</code>
<code>sqlserver_catalog1_connection_string</code>	<code>sqlserver://jdbc:sqlserver://sqlserver1.<i>hostname:port</i>;databaseName= &lt;database_name&gt; ;\${secret1_name }</code>

屬性	Value
sqlserver_catalog2 _connection_string	sqlserver://jdbc:sqlserver://sqlserv er2. <i>hostname:port</i> ;databaseName= <i>&lt;database _name&gt;</i> ; <i>\${secret2_name }</i>

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

## 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${secret_name}`。

```
sqlserver://jdbc:sqlserver://hostname:port;databaseName=<database_name>;${secret_name}
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
sqlserver://
jdbc:sqlserver://
hostname:port;databaseName=<database_name>;user=<user>;password=<password>
```

### 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 SQL Server 執行個體。

處理常式類型	類別
複合處理常式	SqlServerCompositeHandler
中繼資料處理常式	SqlServerMetadataHandler
記錄處理常式	SqlServerRecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數所支援的單一 SQL Server 執行個體。

屬性	Value
default	sqlserver://jdbc:sqlserver:// hostname:port;database Name= <database_name> ;\${secret_name }

## 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 SQL Server 和 Apache Arrow 的相應資料類型。

SQL Server	Arrow
bit	TINYINT
tinyint	SMALLINT
smallint	SMALLINT
int	INT
bigint	BIGINT
decimal	DECIMAL
numeric	FLOAT8
smallmoney	FLOAT8
money	DECIMAL

SQL Server	Arrow
float[24]	FLOAT4
float[53]	FLOAT8
real	FLOAT4
datetime	Date(MILLISECOND)
datetime2	Date(MILLISECOND)
smalldatetime	Date(MILLISECOND)
date	Date(DAY)
time	VARCHAR
datetimeoffset	Date(MILLISECOND)
char[n]	VARCHAR
varchar[n/max]	VARCHAR
nchar[n]	VARCHAR
nvarchar[n/max]	VARCHAR
text	VARCHAR
ntext	VARCHAR

## 分割區和分隔

分割區由 `varchar` 類型的單一分割區資料欄表示。如果是 SQL Server 連接器，分割區函數會決定在資料表上套用分割區的方式。從 SQL Server 中繼資料表中擷取分割區函數和資料欄名稱資訊。然後，自訂查詢可獲取分割區。根據收到的不同分割區數目建立分割。

## 效能

選取資料欄子集可大幅加速查詢執行期，並減少掃描的資料。SQL Server 連接器由於並行而對限流保有彈性。

Athena SQL Server 連接器執行述詞下推，以減少查詢掃描的資料。簡單述詞和複雜表達式會下推至連接器，以減少掃描的資料量並減少查詢執行時間。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena SQL Server 連接器可以合併這些表達式，並將其直接推送至 SQL Server，以增強功能並減少掃描的資料量。

下列 Athena SQL Server 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## 直通查詢

SQL Server 連接器支援[傳遞查詢](#)。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要搭配 SQL Server 使用傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(
```

```
system.query(  
    query => 'query string'  
))
```

下列範例查詢會將查詢向下推送至 SQL Server 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 GitHub .com 上 SQL 伺服器連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena Teradata 連接器

適用於 Teradata 的 Amazon Athena 連接器可讓 Athena 對存放在 Teradata 資料庫中的資料執行 SQL 查詢。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 限制

- 不支援寫入 DDL 操作。
- 在多工器設定中，在所有資料庫執行個體之間共用溢出儲存貯體和字首。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。

## 條款

下列術語與 Teradata 連接器相關。

- 資料庫執行個體 - 在內部部署、Amazon EC2 或 Amazon RDS 上部署的任何資料庫執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是 `connection_string` 屬性的必要前置詞。
- 多工處理常式 - 可以接受和使用多個資料庫連接的 Lambda 處理常式。

## Lambda 層先決條件

若要搭配使用 Teradata 連接器與 Athena，您必須建立包含 Teradata JDBC 驅動程式的 Lambda 層。Lambda 層是包含其他 Lambda 函數的程式碼的 .zip 封存檔。將 Teradata 連接器部署到您的帳戶時，您需要指定層 ARN。這樣一來，會將具有 Teradata JDBC 驅動程式的 Lambda 層連接到 Teradata 連接器，以便您可以將其與 Athena 搭配使用。

如需有關 Lambda 層的詳細資訊，請參閱《AWS Lambda 開發人員指南》中的[建立和共用 Lambda 層](#)。

## 建立 Teradata 連接器的 Lambda 層

1. 瀏覽 Teradata JDBC 驅動程式下載頁面：<https://downloads.teradata.com/download/connectivity/jdbc-driver>。
2. 下載 Teradata JDBC 驅動程式。網站要求您建立一個帳戶並接受授權協議才能下載該檔案。
3. 從下載的封存檔解壓縮出 `terajdbc4.jar` 檔案。
4. 建立以下資料夾結構，並將 `.jar` 檔案放在其中。

```
java\lib\terajdbc4.jar
```

5. 建立整個資料夾結構的 .zip 檔案，其中包含 `terajdbc4.jar` 檔案。

- 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 <https://console.aws.amazon.com/lambda/>。
- 在導覽窗格中，選擇 Layers (層)，然後選擇 Create layer (建立層)。
- 在 Name (名稱) 中，輸入層的名稱 (例如 TeradataJava11LambdaLayer)。
- 確認已選取 Upload a .zip file (上傳 .zip 檔案) 選項。
- 選擇 Upload (上傳)，然後上傳包含 Teradata JDBC 驅動程式的壓縮檔資料夾。
- 選擇 Create (建立)。
- 在層的詳細資訊頁面上，選擇頁面頂部的剪貼簿圖示，複製層 ARN。
- 儲存 ARN 以供參考。

## 參數

使用本節中的 Lambda 環境變數來設定 Teradata 連接器。

## 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
teradata://${jdbc_connection_string}
```

## 使用多工處理常式

您可以使用多工器透過單一 Lambda 函數連接到多個資料庫執行個體。按目錄名稱路由請求。在 Lambda 中使用下列類別。

處理常式	類別
複合處理常式	TeradataMuxCompositeHandler
中繼資料處理常式	TeradataMuxMetadataHandler
記錄處理常式	TeradataMuxRecordHandler

## 多工處理常式參數

參數	描述
<code>\$<i>catalog</i>_connection_string</code>	必要。資料庫執行個體連接字串。在環境變數前面加上 Athena 中使用的目錄名稱。例如，如果向 Athena 註冊的目錄為 <code>myteradatacatalog</code> ，則環境變數名稱為 <code>myteradatacatalog_connection_string</code> 。
<code>default</code>	必要。預設的連接字串。目錄為 <code>lambda:\${ <i>AWS_LAMBDA_FUNCTION_NAME</i> }</code> 時，使用此字串。

下列範例屬性適用於支援兩個資料庫執行個體的 Teradata MUX Lambda 函數：teradata1 (預設) 和 teradata2。

屬性	Value
<code>default</code>	<code>teradata://jdbc:teradata://teradata2.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,user=sample2&amp;password=sample2</code>
<code>teradata_catalog1_connection_string</code>	<code>teradata://jdbc:teradata://teradata1.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,\${Test/RDS/Teradata1}</code>
<code>teradata_catalog2_connection_string</code>	<code>teradata://jdbc:teradata://teradata2.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,user=sample2&amp;password=sample2</code>

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

### ⚠ Important

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

### 帶有秘密名稱的連接字串範例

以下字串具有秘密名稱 `${Test/RDS/Teradata1}`。

```
teradata://jdbc:teradata1.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,${Test/RDS/Teradata1}&...
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
teradata://jdbc:teradata://teradata1.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,...&user=sample2&password=sample2&...
```

目前，Teradata 可辨識 user 和 password JDBC 屬性。它也接受格式為 *username/password* 的使用者名稱和密碼，而不使用金鑰 user 或 password。

### 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Teradata 執行個體。

處理常式類型	類別
複合處理常式	TeradataCompositeHandler
中繼資料處理常式	TeradataMetadataHandler
記錄處理常式	TeradataRecordHandler

### 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

下列範例屬性適用於 Lambda 函數支援的單一 Teradata 執行個體。

屬性	Value
default	teradata://jdbc:teradata:// teradata1.host/TMODE=ANSI,C HARSET=UTF8,DATABASE=TEST,s ecret=Test/RDS/Teradata1

### 溢出參數

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。

參數	描述
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 JDBC 和 Apache Arrow 的相應資料類型。

JDBC	Arrow
Boolean	位元
Integer	Tiny
Short	Smallint
Integer	Int
Long	Bigint
float	Float4
Double	Float8
日期	DateDay
時間戳記	DateMilli
字串	Varchar
位元組	Varbinary
BigDecimal	Decimal (小數)
ARRAY	清單

## 分割區和分隔

分割區由 Integer 類型的單一分割區資料欄表示。此資料欄包含 Teradata 資料表中所定義之分割區的分割區名稱。對於沒有分割區名稱的資料表，會傳回 \*，這相當於一個單一的分割區。分割區等同於分隔。

名稱	Type	描述
分割區	Integer	Teradata 中的已命名分割區。

## 效能

Teradata 支援原生分割區。Athena Teradata 連接器可以平行地從這些分割區中擷取資料。如果您想要查詢具有統一分割區分佈的非常大的資料集，強烈建議使用原生分割。選取資料欄子集會大幅拖慢查詢執行時間。連接器會顯示因為並行而導致的部分限流。

Athena Teradata 連接器執行述詞下推，以減少查詢掃描的資料。簡單述詞和複雜表達式會下推至連接器，以減少掃描的資料量並減少查詢執行時間。

## 述詞

述詞是 SQL 查詢的 WHERE 子句中的一種表達式，它會評估為布林值，並根據多個條件篩選資料列。Athena Teradata 連接器可以合併這些表達式，並將其直接推送至 Teradata，以增強功能並減少掃描的資料量。

下列 Athena Teradata 連接器運算子支援述詞下推：

- 布林值：AND、OR、NOT
- 等式：EQUAL、NOT\_EQUAL、LESS\_THAN、LESS\_THAN\_OR\_EQUAL、GREATER\_THAN、GREATER\_THAN\_OR\_EQUAL
- 算術：ADD、SUBTRACT、MULTIPLY、DIVIDE、MODULUS、NEGATE
- 其他：LIKE\_PATTERN、IN

## 合併下推範例

如需增強的查詢功能，請合併下推類型，如以下範例所示：

```
SELECT *
```

```
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## 直通查詢

Teradata 連接器支援**傳遞**查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要搭配 Teradata 使用傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

下列範例查詢會將查詢向下推送至 Teradata 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上 GitHub Teradata 連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請造訪 GitHub .com 上[的對應網站](#)。

## Amazon Athena Timestream 連接器

Amazon Athena Timestream 連接器讓 Amazon Athena 能夠與 [Amazon Timestream](#) 進行通訊，從而可透過 Amazon Athena 存取您的時間序列資料。您可以選擇性地用 AWS Glue Data Catalog 作補充中繼資料的來源。

Amazon Timestream 是快速、可擴展、全受管、專門打造的時間序列資料庫，可讓您每天輕鬆存放和分析數兆個時間序列資料點。Timestream 會將最近的資料保存在記憶體中，並根據使用者定義的政策將歷史資料移到成本最佳化的儲存層，藉此節省您管理時間序列資料生命週期的時間和成本。

如果您的帳戶中啟用了 Lake Formation，則您部署在其中部署的 Athena 聯合 Lambda 連接器的 IAM 角色 AWS Serverless Application Repository 必須具有 `AWS Glue Data Catalog`

### 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

### 參數

使用本節中的 Lambda 環境變數來設定 Timestream 連接器。

- `spill_bucket` - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- `spill_prefix` - (選用) 預設為指定的 `spill_bucket` 中名為 `athena-federation-spill` 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- `spill_put_request_headers` - (選用) 用於溢出的 Amazon S3 `putObject` 請求的請求標頭和值的 JSON 編碼映射 (例如，`{"x-amz-server-side-encryption" : "AES256"}`)。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- `kms_key_id` - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`，您可以指定 KMS 金鑰 ID。
- `disable_spill_encryption` - (選用) 當設定為 `True` 時，停用溢出加密。預設為 `False`，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。
- `glue_catalog` - (選用) 使用此選項可指定 [跨帳戶 AWS Glue 目錄](#)。依預設，連接器會嘗試從其自己的 AWS Glue 帳戶取得中繼資料。

### 設定資料庫和資料表 AWS Glue

您可以選擇性地使用 AWS Glue Data Catalog 作為補充中繼資料的來源。若要啟用與 Timestream 搭配使用的 AWS Glue 表格，您必須擁有一個資料 AWS Glue 庫和表格，其名稱與您要提供補充描述資料的 Timestream 資料庫和表格相符。

**Note**

為了獲得最佳效能，請僅使用小寫作為資料庫名稱和資料表名稱。使用混合大小寫會導致連接器執行運算密集程度較高的不區分大小寫搜尋。

若要設定 AWS Glue 表格與 Timestream 搭配使用，您必須在 AWS Glue 中設定其表格屬性。

若要將 AWS Glue 表格用於補充描述資料

1. 在 AWS Glue 主控台中編輯表格，以新增下列表格屬性：

- `timestream-metadata-flag`— 此內容向 Timestream 連接器指示連接器可將表格用於補充中繼資料。您可以為 `timestream-metadata-flag` 提供任何值，只要 `timestream-metadata-flag` 屬性存在於資料表屬性清單中。
- `_view_template` - 當您使用 AWS Glue 作為補充中繼資料時，可以使用此資料表屬性，並指定任何 Timestream SQL 作為檢視。Athena Timestream 連接器會使用檢視中的 SQL 以及 Athena 中的 SQL 來執行查詢。如果您想要使用 Athena 中沒有的 Timestream SQL 功能，這會非常有用。

2. 請確定您使用本文件中所列適 AWS Glue 用的資料類型。

### 資料類型

目前，Timestream 連接器僅支援 Timestream 中部份的可用資料類型，尤其是：純量值 `varchar`、`double` 和 `timestamp`。

若要查詢 `timeseries` 資料類型，您必須在 AWS Glue 資料表屬性中設定一個使用 Timestream `CREATE_TIME_SERIES` 函數的檢視。您還需要為檢視提供結構描述，它使用語法 `ARRAY<STRUCT<time:timestamp,measure_value::double:double>>` 作為任何時間序列欄的類型。請務必將 `double` 取代為適合您的資料表的純量類型。

下列影像顯示設定為在時間序列中設定檢視 AWS Glue 表的表格屬性範例。

Tables > my\_timeseries Last updated 6 May 2020 Table Version (Current version) ▾

[Edit table](#) [Delete table](#) [View properties](#) [Compare versions](#) [Edit schema](#)

**Name** my\_timeseries  
**Description**  
**Database** virtuoso  
**Classification** parquet  
**Location** [s3://fake-path/](#)  
**Connection**  
**Deprecated** No  
**Last updated** Wed May 06 16:01:00 GMT-400 2020  
**Input format** org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat  
**Output format** org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat  
**Serde serialization lib** org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe

**Serde parameters**

serialization.format	1
----------------------	---

**Table properties**

timestream-metadata-flag	timestream-metadata-flag
_view_template	select az, hostname, region, CREATE_TIME_SERIES(time, measure_value::double) as cpu_utilization from virtuoso.virtuoso WHERE measure_name = 'cpu_utilization' GROUP BY measure_name, az, hostname, region

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-timestream.yaml](#) 檔案的 Policies 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。
- AWS Glue Data Catalog— 時間流連接器需要的唯讀存取權，才能取 AWS Glue Data Catalog 得結構描述資訊。
- CloudWatch 防護記錄 — 連接器需要存取 CloudWatch 記錄檔以儲存防護記錄。
- Timestream Access - 用於執行 Timestream 查詢。

## 效能

建議您使用 LIMIT 子句將傳回的資料 (非掃描的資料) 限制在 256 MB 以下，以確保互動式查詢效能良好。

Athena Timestream 連接器執行述詞下推，以減少查詢掃描的資料。LIMIT 子句可減少掃描的資料量，但是如果未提供述詞，則應預期含有 LIMIT 子句的 SELECT 查詢可掃描至少 16 MB 的資料。選取資料欄子集可大幅加速查詢執行時間，並減少掃描的資料。Timestream 連接器由於並行而對限流保有彈性。

## 直通查詢

時間流連接器支援[傳遞](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

若要搭配 Timestream 使用傳遞查詢，您可以使用下列語法：

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

下列範例查詢會將查詢向下推送至時間串流中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## 授權資訊

Amazon Athena Timestream 連接器專案是依據 [Apache-2.0 License](#) 來授權的。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的[對應網站](#)。

## Amazon Athena TPC benchmark DS (TPC-DS) 連接器

Amazon Athena TPC-DS 連接器讓 Amazon Athena 能夠與隨機產生的 TPC Benchmark DS 資料來源通訊，以用於 Athena Federation 的基準化分析和功能測試。Athena TPC-DS 連接器會以四種縮放比例之一產生符合 TPC-DS 規範的資料庫。我們不建議使用此連接器作為 Amazon S3 資料湖效能測試的替代方案。

## 必要條件

- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。

## 參數

使用本節中的 Lambda 環境變數來設定 TPC-DS 連接器。

- spill\_bucket - 針對超過 Lambda 函數限制的資料，指定 Amazon S3 儲存貯體。
- spill\_prefix - (選用) 預設為指定的 spill\_bucket 中名為 athena-federation-spill 的子資料夾。我們建議您在此位置設定 Amazon S3 [儲存生命週期](#)，以刪除超過預定天數或小時數的溢出。
- spill\_put\_request\_headers - (選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如，{"x-amz-server-side-encryption" : "AES256"})。如需其他可能的標頭，請參閱 Amazon 簡單儲存服務 API 參考 [PutObject](#) 中的。
- kms\_key\_id - (選用) 依預設，任何溢出到 Amazon S3 的資料都會使用 AES-GCM 驗證加密模式和隨機產生的金鑰進行加密。為了讓您的 Lambda 函數使用 KMS 產生的更強大的加密金鑰，例如 a7e63k4b-81oc-40db-a2a1-4d0en2cd8331，您可以指定 KMS 金鑰 ID。
- disable\_spill\_encryption - (選用) 當設定為 True 時，停用溢出加密。預設為 False，因此溢出 S3 的資料會使用 AES-GCM 進行加密 — 使用隨機產生的金鑰或 KMS 來產生金鑰。停用溢出加密可以提高效能，尤其是如果溢出位置使用 [伺服器端加密](#)。

## 測試資料庫和資料表

Athena TPC-DS 連接器會以四種縮放比例 (tpcds1、tpcds10、tpcds100、tpcds250 或 tpcds1000) 之一產生符合 TPC-DS 規範的資料庫。

## 資料表摘要

如需測試資料表和資料欄的完整清單，請執行 SHOW TABLES 或者 DESCRIBE TABLE 查詢。為了方便起見，提供以下資料表摘要。

1. call\_center
2. catalog\_page
3. catalog\_returns
4. catalog\_sales

5. customer
6. customer\_address
7. customer\_demographics
8. date\_dim
9. dbgen\_version
- 10.household\_demographics
- 11.income\_band
- 12.庫存
- 13.item
- 14.promotion
- 15.reason
- 16.ship\_mode
- 17.存放
- 18.store\_returns
- 19.store\_sales
- 20.time\_dim
- 21.warehouse
- 22.web\_page
- 23.web\_returns
- 24.web\_sales
- 25.web\_site

如需與此產生的結構描述和資料相容的 TPC-DS 查詢，請參閱上的「[雅典娜-tpcs/src/主要/資源/查詢/目錄](#)」。GitHub

### 查詢範例

下列 SELECT 查詢範例會查詢特定郡的客戶人口統計的 tpcds 目錄。

```
SELECT
  cd_gender,
  cd_marital_status,
  cd_education_status,
  count(*) cnt1,
```

```

cd_purchase_estimate,
count(*) cnt2,
cd_credit_rating,
count(*) cnt3,
cd_dep_count,
count(*) cnt4,
cd_dep_employed_count,
count(*) cnt5,
cd_dep_college_count,
count(*) cnt6
FROM
  "lambda:tpcds".tpcds1.customer c, "lambda:tpcds".tpcds1.customer_address ca,
  "lambda:tpcds".tpcds1.customer_demographics
WHERE
  c.c_current_addr_sk = ca.ca_address_sk AND
  ca_county IN ('Rush County', 'Toole County', 'Jefferson County',
               'Dona Ana County', 'La Porte County') AND
  cd_demo_sk = c.c_current_cdemo_sk AND
  exists(SELECT *
         FROM "lambda:tpcds".tpcds1.store_sales, "lambda:tpcds".tpcds1.date_dim
         WHERE c.c_customer_sk = ss_customer_sk AND
              ss_sold_date_sk = d_date_sk AND
              d_year = 2002 AND
              d_moy BETWEEN 1 AND 1 + 3) AND
  (exists(SELECT *
         FROM "lambda:tpcds".tpcds1.web_sales, "lambda:tpcds".tpcds1.date_dim
         WHERE c.c_customer_sk = ws_bill_customer_sk AND
              ws_sold_date_sk = d_date_sk AND
              d_year = 2002 AND
              d_moy BETWEEN 1 AND 1 + 3) OR
   exists(SELECT *
         FROM "lambda:tpcds".tpcds1.catalog_sales, "lambda:tpcds".tpcds1.date_dim
         WHERE c.c_customer_sk = cs_ship_customer_sk AND
              cs_sold_date_sk = d_date_sk AND
              d_year = 2002 AND
              d_moy BETWEEN 1 AND 1 + 3))
GROUP BY cd_gender,
         cd_marital_status,
         cd_education_status,
         cd_purchase_estimate,
         cd_credit_rating,
         cd_dep_count,
         cd_dep_employed_count,
         cd_dep_college_count

```

```
ORDER BY cd_gender,  
         cd_marital_status,  
         cd_education_status,  
         cd_purchase_estimate,  
         cd_credit_rating,  
         cd_dep_count,  
         cd_dep_employed_count,  
         cd_dep_college_count  
LIMIT 100
```

## 所需的許可

如需詳細了解此連接器所需的 IAM 政策，請檢閱 [athena-tpcds.yaml](#) 檔案的 Policies 部分。以下清單摘要說明所需的許可。

- Amazon S3 寫入存取 - 連接器需要 Amazon S3 中某個位置的寫入存取權，以便從大型查詢中溢寫結果。
- Athena GetQueryExecution — 當上游 Athena 查詢終止時，連接器會使用此權限快速失敗。

## 效能

Athena TPC-DS 連接器會嘗試根據您選擇的比例因數平行處理查詢。在 Lambda 函數內執行述詞下推。

## 授權資訊

Amazon Athena TPC-DS 連接器專案是依據 [Apache-2.0 License](#) 來授權的。

## 其他資源

如需有關此連接器的其他資訊，請造訪 GitHub .com 上的 [對應網站](#)。

## Amazon Athena Vertica 連接器

Vertica 是一種單欄式資料庫平台，可部署在雲端或進行內部部署，以支援 Exabyte 艾位元組規模的資料倉儲。您可以在聯合查詢中使用 Amazon Athena Vertica 連接器，查詢來自 Athena 的 Vertica 資料來源。例如，您可以在 Vertica 上的資料倉儲和 Simple Storage Service (Amazon S3) 中的資料湖上執行分析查詢。

## 必要條件

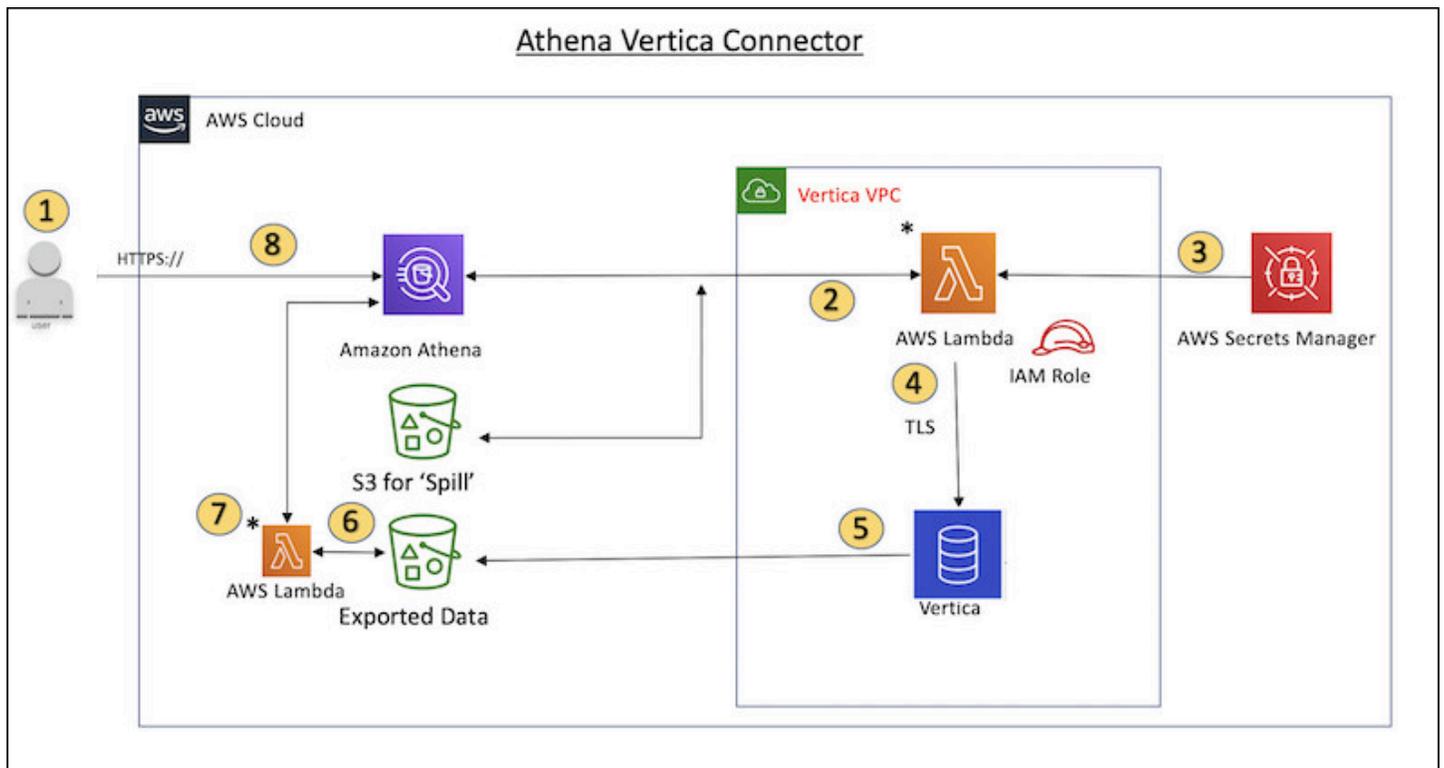
- 使用 Athena 主控台或 AWS Serverless Application Repository，將連接器部署到您的 AWS 帳戶。如需詳細資訊，請參閱 [部署資料來源連接器](#) 或 [使用 AWS Serverless Application Repository 部署資料來源連接器](#)。
- 您必須先設定 VPC 和安全群組，才能使用此連接器。如需詳細資訊，請參閱 [為資料來源連接器建立 VPC](#)。

## 限制

- 因為 Athena Vertica 連接器使用 [Amazon S3 Select](#) 從 Amazon S3 中讀取 Parquet 檔案，所以連接器的效能可能會很慢。查詢大型資料表時，建議您使用 [CREATE TABLE AS \(SELECT ...\)](#) 查詢和 SQL 述詞。
- 目前，由於 Athena Federated Query 中存在已知問題，連接器會讓 Vertica 將已查詢資料表的所有資料欄匯出至 Amazon S3，但在 Athena 主控台的結果中只能看到已查詢的資料欄。
- 不支援寫入 DDL 操作。
- 任何相關的 Lambda 限制。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的 [Lambda 配額](#)。

## 工作流程

下圖顯示使用 Vertica 連接器的查詢工作流程。



1. 針對 Vertica 中的一個或多個資料表發出 SQL 查詢。
2. 連接器會解析 SQL 查詢，以透過 JDBC 連接將相關部分傳送至 Vertica。
3. 連接字串會使用儲存在中 AWS Secrets Manager 的使用者名稱和密碼來存取 Vertica。
4. 連接器使用 Vertica EXPORT 命令包裝 SQL 查詢，如下例所示。

```
EXPORT TO PARQUET (directory = 's3://DOC-EXAMPLE-BUCKET/folder_name,
    Compression='Snappy', fileSizeMB=64) OVER() as
SELECT
PATH_ID,
...
SOURCE_ITEMIZED,
SOURCE_OVERRIDE
FROM DELETED_OBJECT_SCHEMA.FORM_USAGE_DATA
WHERE PATH_ID <= 5;
```

5. Vertica 處理 SQL 查詢，並將結果集傳送至 Amazon S3 儲存貯體。為了獲得更好的輸送量，Vertica 使用 EXPORT 選項來平行處理多個 Parquet 檔案的寫入操作。
6. Athena 會掃描 Amazon S3 儲存貯體，以確定要為結果集讀取的檔案數。
7. Athena 對 Lambda 函數發出多個呼叫並使用 [Amazon S3 Select](#) 從結果集中讀取 Parquet 檔案。多個呼叫可讓 Athena 平行讀取 Amazon S3 檔案，並達到每秒高達 100GB 的輸送量。

8. Athena 會使用從資料湖掃描的資料來處理從 Vertica 傳回的資料，並傳回結果。

## 條款

下列術語與 Vertica 連接器相關。

- 資料庫執行個體 - 部署在 Amazon EC2 上的 Vertica 資料庫的任何執行個體。
- 處理常式 - 存取資料庫執行個體的 Lambda 處理常式。處理常式可以用於中繼資料或資料記錄。
- 中繼資料處理常式 - 從資料庫執行個體中擷取中繼資料的 Lambda 處理常式。
- 記錄處理常式 - 從資料庫執行個體中擷取資料記錄的 Lambda 處理常式。
- 複合處理常式 - 從資料庫執行個體中擷取中繼資料和資料記錄的 Lambda 處理常式。
- 屬性或參數 - 處理常式用來擷取資料庫資訊的資料庫屬性。您可以將這些屬性設定為 Lambda 環境變數。
- 連接字串 - 用來建立資料庫執行個體連線的文字字串。
- 目錄 — 在 Athena 註冊的非AWS Glue 目錄，是`connection_string`屬性的必要前置詞。

## 參數

Amazon Athena Vertica 連接器透過 Lambda 環境變數公開組態選項。您可以使用下列 Lambda 環境變數來設定連接器。

- AthenaCatalogName-Lambda 函數名
- ExportBucket— 匯出 Vertica 查詢結果的 Amazon S3 儲存貯體。
- SpillBucket— 此函數可能會洩漏資料的 Amazon S3 儲存貯體的名稱。
- SpillPrefix— 此函數可能溢滿資料之SpillBucket位置的前置詞。
- SecurityGroupIds— 對應至應套用至 Lambda 函數的安全性群組的一或多個 ID (例如sg1sg2、或sg3)。
- SubnetIds— 對應至子網路的一或多個子網路 ID，Lambda 函數可用來存取您的資料來源 (例如subnet1，或subnet2)。
- SecretNameOrPrefix— 這個函數可以存取的 Secrets Manager 中一組名稱的名稱或前置詞 (例如vertica-\*)
- VerticaConnectionString— 如果未定義目錄特定連線，預設會使用 Vertica 連線詳細資料。字串可以選擇性地使用 AWS Secrets Manager 語法 (例如，`${secret_name}`)。
- VPC ID - 連接至 Lambda 函數的 VPC ID。

## 連接字串

使用下列格式的 JDBC 連接字串來連接資料庫執行個體。

```
vertica://jdbc:vertica://host_name:port/database?user=vertica-username&password=vertica-password
```

## 使用單一連接處理常式

您可以使用下列單一連接中繼資料和記錄處理常式來連接至單一 Vertica 執行個體。

處理常式類型	類別
複合處理常式	VerticaCompositeHandler
中繼資料處理常式	VerticaMetadataHandler
記錄處理常式	VerticaRecordHandler

## 單一連接處理常式參數

參數	描述
default	必要。預設的連接字串。

單一連接處理常式支援一個資料庫執行個體，並且必須提供 default 連接字串參數。忽略所有其他連接字串。

## 提供憑證

要在 JDBC 連接字串中提供資料庫的使用者名稱和密碼，可以使用連接字串屬性或 AWS Secrets Manager。

- 連接字串 - 可以將使用者名稱和密碼指定為 JDBC 連接字串中的屬性。

**⚠ Important**

作為安全最佳實務，請勿在您的環境變數或連線字串中使用硬式編碼憑證。如需有關將硬式編碼密碼移至的資訊 AWS Secrets Manager，請參閱《AWS Secrets Manager 使用者指南》AWS Secrets Manager 中的「[將硬式編碼密碼移至](#)」。

- AWS Secrets Manager— 若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，連線至 Lambda 函數的虛擬私人雲端應具有[網際網路存取權限](#)，或具有連線至 Secrets Manager 的[VPC 端點](#)。

您可以在 JDBC 連接字符串 AWS Secrets Manager 中輸入密碼的名稱。連接器將秘密名稱取代為 Secrets Manager 中的 username 和 password 值。

對於 Amazon RDS 資料庫執行個體，已緊密整合此支援。如果您使用 Amazon RDS，我們強烈建議您使用 AWS Secrets Manager 和登入資料輪替。如果您的資料庫不使用 Amazon RDS，請以下列格式將憑證儲存為 JSON：

```
{"username": "${username}", "password": "${password}"}
```

**帶有秘密名稱的連接字符串範例**

以下字串具有秘密名稱 `${vertica-username}` 和 `${vertica-password}`。

```
vertica://jdbc:vertica://host_name:port/database?user=${vertica-username}&password=${vertica-password}
```

連接器會使用秘密名稱來擷取秘密並提供使用者名稱和密碼，如下列範例所示。

```
vertica://jdbc:vertica://host_name:port/database?user=sample-user&password=sample-password
```

目前，Vertica 連接器可辨識 `vertica-username` 和 `vertica-password` JDBC 屬性。

**溢出參數**

Lambda SDK 可能會將資料溢出至 Amazon S3。由相同 Lambda 函數存取的所有資料庫執行個體溢出到相同的位置。

參數	描述
spill_bucket	必要。溢出儲存貯體名稱。
spill_prefix	必要。溢出儲存貯體金鑰字首。
spill_put_request_headers	(選用) 用於溢出的 Amazon S3 putObject 請求的請求標頭和值的 JSON 編碼映射 (例如, {"x-amz-server-side-encryption" : "AES256"} )。如需其他可能的標頭, 請參閱 Amazon 簡單儲存服務 API 參考 <a href="#">PutObject</a> 中的。

## 支援的資料類型

下表顯示 Vertica 連接器支援的資料類型。

Boolean
BigInt
Short
Integer
Long
Float
Double
日期
Varchar
位元組
BigDecimal
TimeStamp 作為瓦爾

## 效能

Lambda 函數執行投影下推，以減少查詢掃描的資料。LIMIT 子句可減少掃描的資料量，但是如果未提供述詞，則應預期含有 LIMIT 子句的 SELECT 查詢可掃描至少 16 MB 的資料。Vertica 連接器由於並行而對限流保有彈性。

## 直通查詢

Vertica 連接器支援[直通](#)查詢。傳遞查詢使用資料表函數將完整查詢推送至資料來源以進行執行。

要將傳遞查詢與 Vertica 一起使用，您可以使用以下語法：

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

下列範例查詢會將查詢向下推送至 Vertica 中的資料來源。查詢會選取資料customer表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## 授權資訊

使用此連接器即表示您確認已包含協力廠商元件，其清單可在此連接器的 [pom.xml](#) 檔案中找到，並同意 GitHub .com 上 [LICENSE.txt](#) 檔案中提供的個別協力廠商授權中的條款。

## 其他資源

如需最新的 JDBC 驅動程式版本資訊，請參閱 .com 上 GitHub 適用於 Vertica 連接器的 [pom.xml](#) 檔案。

如需有關此連接器的其他資訊，請參閱 GitHub .com 上的[對應網站](#)和[使用AWS 大數據部落格中的 Athena 聯合查詢 SDK 在 Amazon Athena 中查詢 Vertica 資料來源](#)。

## 部署資料來源連接器

準備建立聯合查詢的過程分成兩部分：部署 Lambda 函數資料來源連接器，以及將 Lambda 函數連接到資料來源。在此過程中，請指定 Lambda 函數的名稱，以便稍後在 Athena 主控台中選擇該函數，同時為連接器指定名稱，以便在 SQL 查詢中引用該連接器。

### Note

若要搭配使用 Athena 聯合查詢功能 AWS Secrets Manager，您必須為 Secrets Manager 設定 Amazon VPC 私有端點。如需詳細資訊，請參閱《AWS Secrets Manager 使用者指南》中的[建立 Secrets Manager VPC 私有端點](#)。

### 主題

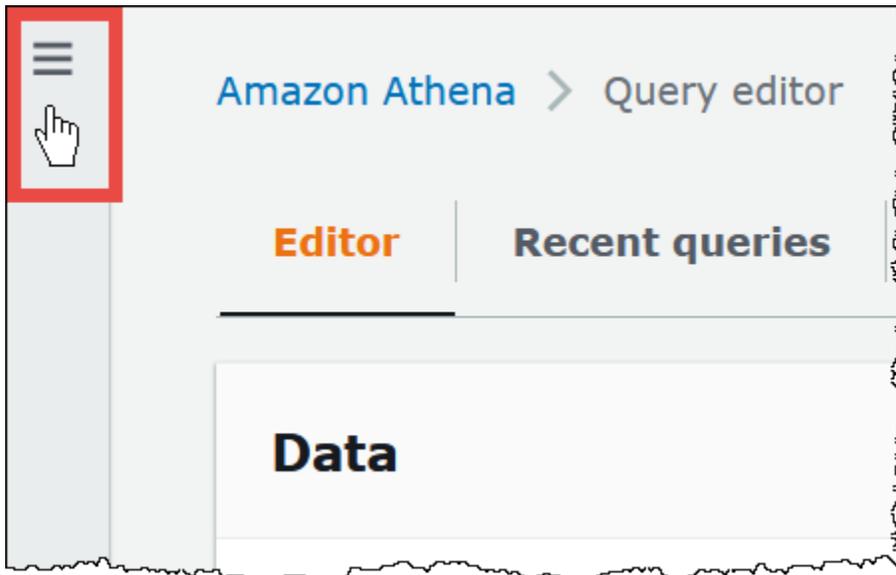
- [使用 Athena 主控台](#)
- [使用 AWS Serverless Application Repository 部署資料來源連接器](#)
- [為資料來源連接器建立 VPC](#)
- [啟用跨帳戶聯合查詢](#)
- [更新資料來源連接器](#)

### 使用 Athena 主控台

若要選擇、命名和部署資料來源連接器，請在整合程序中使用 Athena 和 Lambda 主控台。

### 部署資料來源連接器

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中選擇 Data sources (資料來源)。
4. 在 Data sources (資料來源) 頁面上，選擇 Create data source (建立資料來源)。
5. 針對 Choose a data source (選擇資料來源)，請參考下列指導方針，選擇您希望 Athena 查詢的資料來源：
  - 選擇與您的資料來源對應的聯合查詢選項。使用 Athena 預先建置的資料來源連接器可以設定包括 MySQL、Amazon DocumentDB 和 PostgreSQL 在內的來源。
  - 如果想要查詢 Simple Storage Service (Amazon S3) 中的資料，而且沒有使用 Apache Hive 中繼存放區或此頁面上的其他任何一個聯合查詢資料來源選項，請選擇 S3 - AWS Glue Data Catalog。Athena 使用 AWS Glue Data Catalog 以將資料來源的中繼資料和結構描述資訊存放在 Simple Storage Service (Amazon S3) 中。這是預設的 (非聯合) 選項。如需詳細資訊，請參閱 [用於連接 AWS Glue 到 Amazon S3 中的資料來源](#)。
  - 選擇 S3 - Apache Hive metastore (S3 - Apache Hive 中繼存放區) 可查詢使用 Apache Hive 中繼存放區的 Simple Storage Service (Amazon S3) 資料集。如需有關此選項的詳細資訊，請參閱 [將 Athena 連接到 Apache Hive 中繼存放區](#)。
  - 如果想要建立用於 Athena 的資料來源連接器，請選擇 Custom or shared connector (自訂或共用連接器)。如需寫入資料來源連接器的相關資訊，請參閱 [使用 Athena Query Federation SDK 開發資料來源連接器](#)。

本教學會選擇 Amazon CloudWatch 日誌做為聯合資料來源。

6. 選擇下一步。

7. 在 Enter data source details (輸入資料來源詳細資訊) 頁面中，當您從 Athena 查詢資料來源 (例如 CloudWatchLogs)，請於 Data Source Name (資料來源名稱) 輸入要在 SQL 陳述式中使用的名稱。名稱最多可包含 127 個字元，且在您的帳戶中必須是唯一的。建立後便無法變更。有效字元包括 a-z、A-Z、0-9、\_ (底線)、@ (at 符號) 和 - (連字號)。Athena 已保留名稱 awsdatalog、hive、jmx 和 system，這些名稱不能用於資料來源名稱。
8. 針對 Lambda function (Lambda 函數)，選擇 Create Lambda function (建立 Lambda 函數)。您所選擇之連接器的功能頁會在 AWS Lambda 主控台中開啟。此頁面包含連接器的詳細資訊。
9. 在 Application settings (應用程式設定) 下，仔細閱讀每個應用程式設定的說明，然後輸入符合您需求的數值。

您看到的應用程式設定會根據資料來源的連接器而有所不同。所需最低設定包括：

- AthenaCatalogName— Lambda 函數的名稱，以小寫表示其定位的資料來源，例如 cloudwatchlogs。
- SpillBucket— 您帳戶中的 Amazon S3 儲存貯體，用於存放超過 Lambda 函數回應大小限制的資料。

#### Note

溢出的資料不會在後續執行中重複使用，並且可以在 12 小時後安全刪除。Athena 不會替您刪除此資料。若要管理這些物件，請考慮新增一個會刪除您 Simple Storage Service (Amazon S3) 溢出儲存貯體中的舊資料的物件生命週期政策。如需詳細資訊，請參閱 Simple Storage Service (Amazon S3) 使用者指南中的 [管理儲存生命週期](#)。

10. 選擇 I acknowledge that this app creates custom IAM roles and resource policies (我認可此應用程式建立自訂的 IAM 角色和資源政策)。如需詳細資訊，請選擇 Info (資訊) 連結。
11. 選擇部署。部署完成後，Lambda 函數會出現在 Lambda 主控台的 Resources (資源) 區段中。

## 連接至資料來源

將資料來源連接器部署到您的帳戶之後，即可使 Athena 與其連接。

使用已部署至帳戶的連接器讓 Athena 與資料來源連接

1. 返回 Athena 主控台的 Enter data source details (輸入資料來源) 頁面。
2. 在 Connection details (連線詳細資訊) 區段中，選擇 Select or enter a Lambda function (選取或輸入 Lambda 函數) 搜尋方塊旁的重新整理圖示。

3. 選擇您剛才在 Lambda 主控台中建立的函數名稱。隨即顯示 Lambda 函數的 ARN。
4. (選用) 在 Tags (標籤) 中，新增要與此資料來源相關聯的鍵值對。如需標籤的詳細資訊，請參閱[標記 Athena 資源](#)。
5. 選擇下一步。
6. 在 Review and create (檢閱並建立) 頁面上，檢閱資料來源詳細資訊，然後選擇 Create data source (建立資料來源)。
7. 該頁面的 Data source details (資料來源詳細資訊) 區段顯示了有關新連接器的資訊。您現在可以在 Athena 查詢中使用此連接器。

如需有關在查詢中使用資料連接器的資訊，請參閱[執行聯合查詢](#)。

### 使用 AWS Serverless Application Repository 部署資料來源連接器

若要部署資料來源連接器，可以使用 [AWS Serverless Application Repository](#)，無需從 Athena 主控台開始。請使用 AWS Serverless Application Repository 來尋找您想使用的連接器、提供連接器所需的參數，然後將連接器部署至您的帳戶。然後，部署連接器後，可以使用 Athena 主控台讓資料來源可供 Athena 使用。

#### 將連接器部署到您的帳戶

#### 使用 AWS Serverless Application Repository 將資料來源連接器部署至您的帳戶

1. 登入 AWS Management Console 並開啟 Serverless App Repository (無伺服器無應用程式儲存庫)。
2. 選擇在導覽窗格中的 Available applications (可用的應用程式)。
3. 選取選項 Show apps that create custom IAM roles or resource policies (顯示建立自訂 IAM 角色或資源政策的應用程式)。
4. 在搜尋方塊中，輸入連接器的名稱。如需預先建置的 Athena 資料連接器之清單，請參閱[可用資料來源連接器](#)。
5. 選擇連接器的名稱。在 AWS Lambda 主控台中選擇連接器會開啟 Lambda 函數的 Application details (應用程式詳細資訊) 頁面。
6. 在此詳細資訊頁面右側的 Application settings (應用程式設定) 中，輸入必要資訊。至少需設定下列內容。如需 Athena 所建立之資料連接器的其餘可設定選項的相關資訊，請參閱 GitHub 上相關的[可用的連接器](#)主題。
  - AthenaCatalogName – 小寫的 Lambda 函數名稱，表示其目標資料來源，例如 cloudwatchlogs。

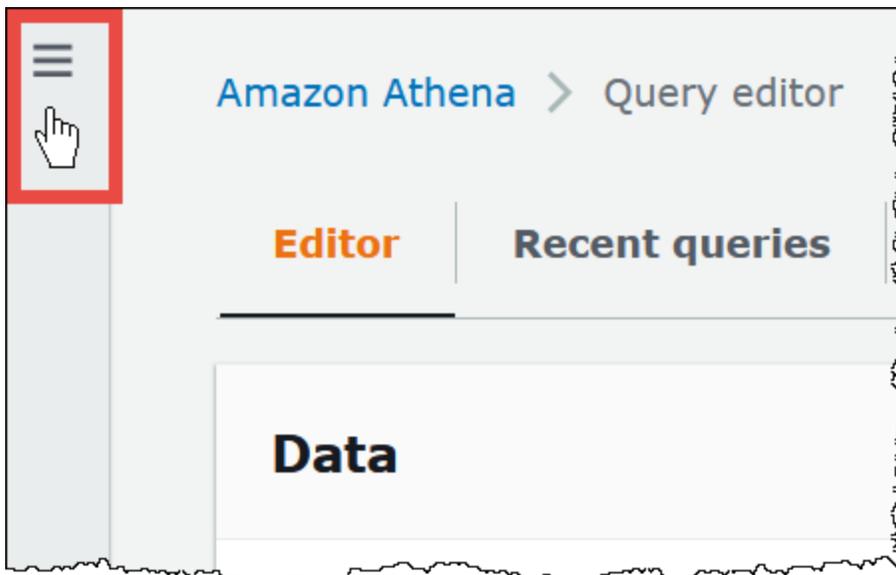
- SpillBucket – 指定您帳戶中用來接收所有大型回應承載資料 (超過 Lambda 函數回應大小限制的資料) 的 Simple Storage Service (Amazon S3) 儲存貯體。
7. 選擇 I acknowledge that this app creates custom IAM roles and resource policies (我認可此應用程式建立自訂的 IAM 角色和資源政策)。如需詳細資訊，請選擇 Info (資訊) 連結。
  8. 在 Application settings (應用程式設定) 部分的右下方，選擇 Deploy (部署)。部署完成後，Lambda 函數會出現在 Lambda 主控台的 Resources (資源) 區段中。

## 使連接器可供 Athena 使用

現在，您可以所示使用 Athena 主控台使資料來源連接器可供 Athena 使用。

## 使資料來源連接器可供 Athena 使用

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中選擇 Data sources (資料來源)。
4. 在 Data sources (資料來源) 頁面上，選擇 Create data source (建立資料來源)。
5. 針對 Choose a data source (選擇資料來源)，選擇您在 AWS Serverless Application Repository 中為其建立連接器的資料來源。本教學使用 Amazon CloudWatch Logs 作為聯合資料來源。
6. 選擇 Next (下一步)。
7. 在 Enter data source details (輸入資料來源詳細資訊) 頁面中，當您從 Athena 查詢資料來源 (例如 CloudWatchLogs)，請於 Data Source Name (資料來源名稱) 輸入要在 SQL 陳述式中使用的名稱。名稱最多可包含 127 個字元，且在您的帳戶中必須是唯一的。建立後便無法變

- 更。有效字元包括 a-z、A-Z、0-9、\_ (底線)、@ (at 符號) 和 - (連字號)。Athena 已保留名稱 `awsdatacatalog`、`hive`、`jmx` 和 `system`，這些名稱不能用於資料來源名稱。
- 在 Connection details (連線詳細資訊) 區段中，使用 Select or enter a Lambda function (選取或輸入 Lambda 函數) 方塊，以選擇您剛才建立的函數名稱。隨即顯示 Lambda 函數的 ARN。
  - (選用) 在 Tags (標籤) 中，新增要與此資料來源相關聯的鍵值對。如需標籤的詳細資訊，請參閱 [標記 Athena 資源](#)。
  - 選擇 Next (下一步)。
  - 在 Review and create (檢閱並建立) 頁面上，檢閱資料來源詳細資訊，然後選擇 Create data source (建立資料來源)。
  - 該頁面的 Data source details (資料來源詳細資訊) 區段顯示了有關新連接器的資訊。您現在可以在 Athena 查詢中使用此連接器。

如需有關在查詢中使用資料連接器的資訊，請參閱 [執行聯合查詢](#)。

## 為資料來源連接器建立 VPC

某些 Athena 資料來源連接器需要 VPC 和安全群組。本主題將說明如何建立具有子網路的 VPC 和 VPC 的安全群組。在此程序中，您可擷取您建立的 VPC、子網路和安全群組的 ID。設定用於 Athena 的連接器時，這些 ID 是必要項目。

## 為 Athena 資料來源連接器建立 VPC

- 登入 AWS Management Console 並開啟 Amazon VPC 主控台，網址為 <https://console.aws.amazon.com/vpc/>。
- 選擇建立 VPC。
- 在 [建立 VPC] 頁面的 [VPC 設定] 下，對於 [要建立的資源]，選擇 [VPC] 等。
- 在「自動產生名稱標籤」下，對於「自動產生」，輸入將用於為 VPC 中的所有資源產生名稱標籤的值。
- 選擇建立 VPC。
- 該過程完成後，選擇查看 VPC。
- 在 Details (詳細資訊) 區段的 VPC ID 中，複製您的 VPC ID 以供日後參考。

現在，您已準備好擷取剛建立的 VPC 的子網路 ID。

## 擷取您的 VPC 子網路 ID

1. 在 VPC 主控台的導覽窗格中，選擇 Subnets (子網路)。
2. 選取子網路的名稱，其 VPC 資料行具有您記下的 VPC ID。
3. 在 Details (詳細資訊) 區段的 Subnet ID (子網路 ID) 中，複製您的子網路 ID 以供日後參考。

接著，為您的 VPC 建立安全群組。

## 為 VPC 建立安全群組

1. 在 VPC 主控台的導覽窗格中，選擇 Security (安全)、Security Groups (安全群組)。
2. 選擇建立安全群組。
3. 在 Create Security Group (建立安全群組) 頁面，輸入下列資訊：
  - 在 Security group name (安全群組名稱) 中，輸入您的安全群組名稱。
  - 在 Description (說明) 中，輸入安全群組的說明。需要描述。
  - 對於 VPC，請選擇您為資料來源連接器建立的 VPC 識別碼。
  - 在 Inbound rules (傳入規則) 和 Outbound rules (傳出規則) 中，新增您所需的任何傳入和傳出規則。
4. 選擇建立安全群組。
5. 在安全群組的 Details (詳細資訊) 頁面，複製 Security group ID (安全群組 ID) 以供日後參考。

## 啟用跨帳戶聯合查詢

聯合查詢允許您使用部署在 AWS Lambda 上的資料來源連接器查詢 Amazon S3 以外的資料來源。跨帳戶聯合查詢功能允許 Lambda 函數和要查詢的資料來源位於不同的帳戶中。

作為資料管理員，您可以透過與資料分析師的帳戶共用資料連接器來啟用跨帳戶聯合查詢，或者作為資料分析師，透過使用資料管理員提供的共用 Lambda ARN 新增至您的帳戶，來啟用此功能。當對原始帳戶中的連接器進行組態變更時，更新的組態將自動套用於其他使用者帳戶中的連接器共用執行個體。

## 考量與限制

- 跨帳戶聯合查詢功能適用於使用 Lambda 型資料來源的非 Hive 中繼儲存資料連接器。
- 此功能不適用於 AWS Glue Data Catalog 資料來源類型。如需跨帳戶存取 AWS Glue Data Catalog 的相關資訊，請參閱[跨帳戶存取 AWS Glue 資料目錄](#)。

- 如果來自連接器 Lambda 函數的回應超過 6 MB 的 Lambda 回應大小限制，Athena 會自動對您設定的 Amazon S3 儲存貯體進行加密、批次處理和溢出回應。執行 Athena 查詢的實體必須能夠存取溢出位置，這樣 Athena 才能讀取溢出的資料。建議您設定 Amazon S3 生命週期政策，從溢出位置刪除物件，因為查詢完成後就不需要資料了。
- 不支援跨 AWS 區域 使用聯合查詢。

## 所需的許可

- 若資料管理員帳戶 A 要與資料分析師帳戶 B 共用 Lambda 函數，帳戶 B 需要 Lambda 叫用函數和溢出儲存貯體存取權限。因此，帳戶 A 應將[資源型政策](#)新增至 Lambda 函數中，並將[主體](#)存取權限新增至 Amazon S3 中的溢出儲存貯體。
  1. 以下政策向帳戶 B 授予帳戶 A 中 Lambda 函數的 Lambda 叫用函數許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountInvocationStatement",
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::account-B-id:user/username"]
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:aws-region:account-A-id:function:lambda-function-name"
    }
  ]
}
```

2. 以下政策允許溢出儲存貯體存取帳戶 B 中的主體。

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::account-B-id:user/username"]
      },
      "Action": [
        "s3:GetObject",

```

```

        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::spill-bucket",
        "arn:aws:s3:::spill-bucket/*"
      ]
    }
  ]
}

```

3. 如果 Lambda 函數使用 AWS KMS 金鑰而非聯合 SDK 提供的預設加密來加密溢出儲存貯體，則帳戶 A 中的 AWS KMS 金鑰原則必須將存取權授與帳戶 B 中的使用者，如下列範例所示。

```

{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": ["arn:aws:iam::account-B-id:user/username"]
  },
  "Action": [ "kms:Decrypt" ],
  "Resource": "*" // Resource policy that gets placed on the KMS key.
}

```

- 若要讓帳戶 A 與帳戶 B 共用其連接器，帳戶 B 必須透過呼叫 AWS 安全性權杖服務 [AssumeRole](#) API 動作來建立名為 `AthenaCrossAccountCreate-account-A-id` 該帳戶 A 所承擔的角色。

以下政策允許 `CreateDataCatalog` 動作，應在帳戶 B 中建立，並新增至帳戶 B 為帳戶 A 建立的 `AthenaCrossAccountCreate-account-A-id` 角色。

```

{
  "Effect": "Allow",
  "Action": "athena:CreateDataCatalog",
  "Resource": "arn:aws:athena:*:account-B-id:datacatalog/*"
}

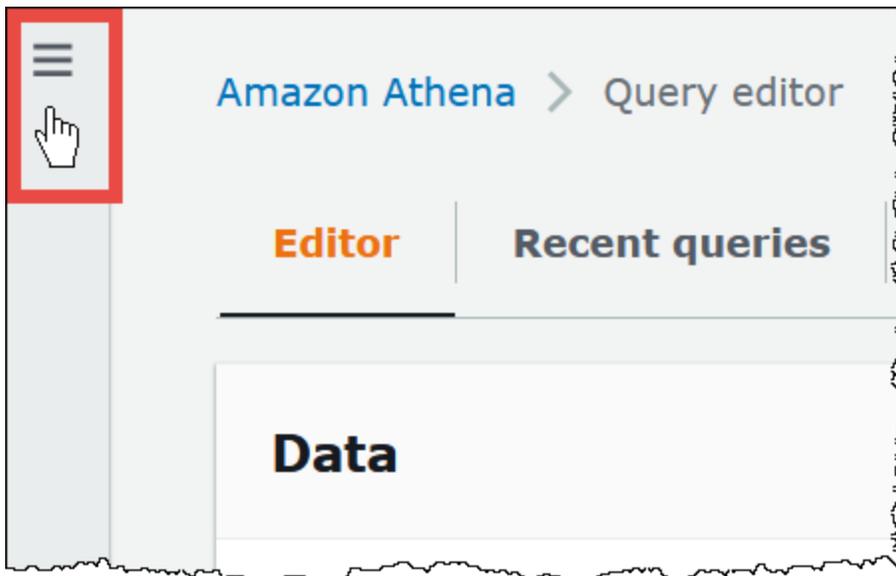
```

## 與帳戶 B 共享帳戶 A 中的資料來源

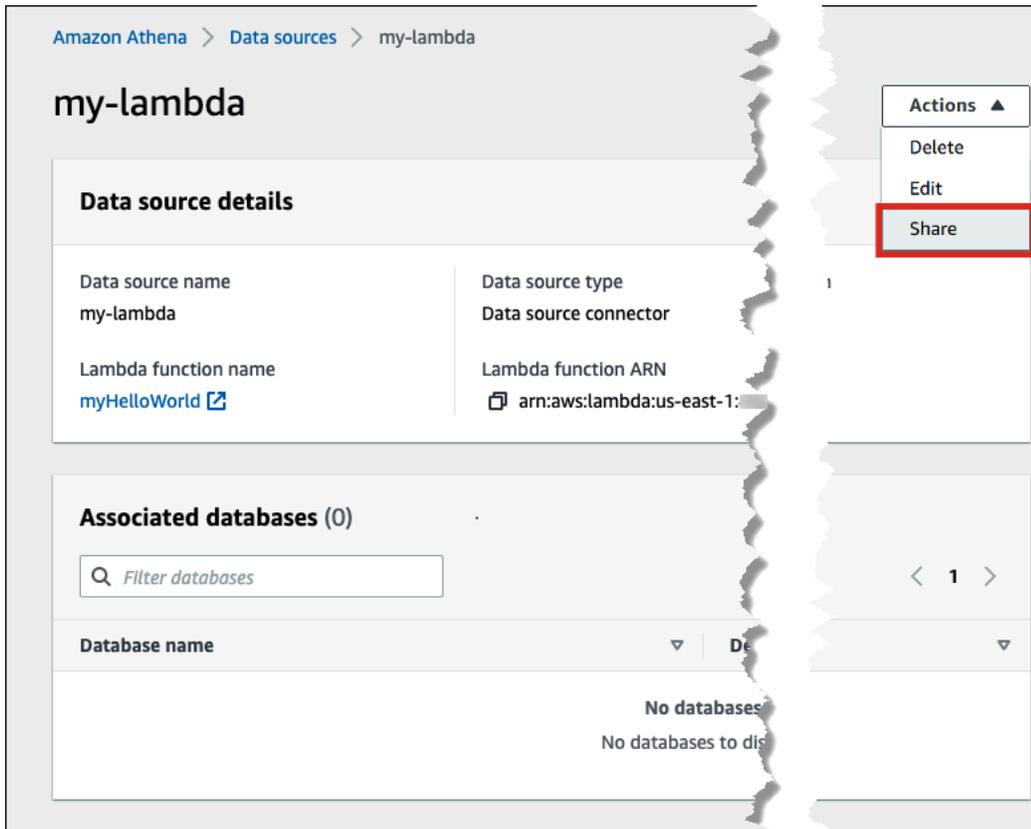
設定許可後，即可使用 Athena 主控台中的 Data sources (資料來源) 頁面將您的帳戶 (帳戶 A) 中的資料連接器與另一個帳戶 (帳戶 B) 共用。帳戶 A 保留連接器的完全控制權和擁有權。當帳戶 A 對連接器進行組態變更時，更新的組態將套用於帳戶 B 中的共用連接器。

## 與帳戶 B 共用帳戶 A 中的 Lambda 資料來源

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 選擇 Data Source (資料來源)。
4. 在 Data sources (資料來源) 頁面上，選擇您要共用的連接器連結。
5. 在 Lambda 資料來源的詳細資訊頁面上，選擇右上角的 Share (共用) 選項。



6. 在 Share **Lambda-name** with another account (與另一個帳戶共用 Lambda-name) 對話方塊中，輸入所需的資訊。
  - 針對 Data source name (資料來源名稱) 中，輸入您希望複製的資料來源在其他帳戶中顯示的名稱。
  - 針對 Account ID (帳戶 ID) 中，輸入要與其共用資料來源的帳戶 ID (在本例中為帳戶 B)。

## Share my-lambda with another account? [Learn more](#) ✕

**Data source name**  
Create a unique name to specify this data source within a SQL statement. For example, `SELECT * from <catalogName>.<database>.<table>`

The name cannot be changed after creation. It can be up to 127 characters. Valid characters are a-z, A-Z, 0-9, \_(underscore), @(at sign) and -(hyphen).

**Account ID**

Account ID can only be numbers (0-9) and 12 characters.

**Cancel** **Share**

7. 選擇共用。您指定的共用資料連接器會建立在帳戶 B 中。帳戶 A 中連接器的組態變更會套用於帳戶 B 中的連接器。

將共用資料來源從帳戶 A 新增至帳戶 B

作為資料分析師，您可能會從資料管理員處獲得連接器的 ARN，以將其新增至您的帳戶中。您可以使用 Athena 主控台的 Data sources (資料來源) 頁面將管理員提供的 Lambda ARN 新增至您的帳戶。

將共用資料連接器的 Lambda ARN 新增至您的帳戶

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果您正在使用新的主控台體驗，且看不到導覽窗格，請選擇左側的展開選單。
3. 選擇 Data Source (資料來源)。
4. 在 Data sources (資料來源) 頁面上，選擇 Connect data source (連接資料來源)。

The screenshot shows the Amazon Athena console interface. On the left is a navigation sidebar with options like 'New Athena experience', 'Query editor', 'Workgroups', 'Data sources', and 'Density Settings'. The main content area is titled 'Data sources' and contains a list of existing data sources. A red box highlights the 'Connect data source' button in the top right corner of the main area.

	Data source name	Data source type
<input type="radio"/>	AwsDataCatalog	AWS Glue Data Catalog
<input type="radio"/>	cw_logs_catalog	Data source connector
<input type="radio"/>	cw_metrics_catalog	Data source connector
<input type="radio"/>	dynamo_db_catalog	Data source connector
<input type="radio"/>	hms-catalog-1	Hive metastore

5. 選擇 Custom or shared connector (自訂或共用連接器)。

The screenshot shows the 'Connect data sources' page in the Amazon Athena console. It features a 'Data source selection' section with the instruction 'Choose the data source to query with Athena'. There are four options displayed as cards, each with a radio button. The 'Custom or shared connector' option is selected and highlighted with a red box.

Radio Button	Icon	Data Source Name	Description
<input type="radio"/>		S3 - AWS Glue Data Catalog	Queries data from S3.
<input type="radio"/>		S3 - Apache Hive metastore	Queries data from S3.
<input type="radio"/>		Redis	Queries data from Redis.
<input checked="" type="radio"/>		Custom or shared connector	Use a custom or another account's connector.

- 在 Lambda function (Lambda 函數) 區段中，請確認選取 Use an existing Lambda function (使用現有 Lambda 函數) 選項。

The screenshot shows the 'Data source details' section of the Amazon Athena console. At the top, there are two options for the data source: 'Redis' (Queries data from Redis) and 'Custom or shared connector' (Use a custom or another account's connector). The 'Custom or shared connector' option is selected. Below this is the 'Lambda function' section, which includes the following text: 'Choose or enter a Lambda function for your data source, or create and configure a Lambda function for the connection.' Underneath, there are two radio buttons: 'Use an existing Lambda function' (selected and highlighted with a red box) and 'Create a new Lambda function'. Below the radio buttons, there is a text input field with the placeholder 'Choose or enter a Lambda function' and the instruction 'Choose a Lambda function to connect to your data source, or enter the ARN for a cross-account Lambda data source function. To manage the Lambda function details, use the Lambda console. [Info](#)'. The input field contains the ARN 'arn:aws:lambda:us-west-2:123456789012:function:Account-A-function' and is highlighted with a red box. To the right of the input field is a search icon and a refresh icon. At the bottom right of the console, there are two buttons: 'Cancel' and 'Connect data source' (highlighted in red).

- 針對 Choose or enter a Lambda function (選擇或輸入 Lambda 函數)，輸入帳戶 A 的 Lambda ARN。
- 選擇 Connect data source (連接資料來源)。

## 故障診斷

如果您收到錯誤訊息，指出帳戶 A 沒有在帳戶 B 中擔任角色的許可，請確認在帳戶 B 中建立的角色名稱拼寫正確，並且連接了正確的政策。

## 更新資料來源連接器

Athena 建議您定期將使用的資料來源連接器更新到最新版本，以利用新功能和增強功能。若要開始使用，您必須找到最新的版本編號。

### 尋找最新的 Athena Query Federation 版本

Athena 資料來源連接器的最新版本編號對應於最新的 Athena Query Federation 版本。在某些情況下，GitHub 發行版本可能會比 AWS Serverless Application Repository (SAR) 上提供的版本稍新。

若要找到最新的 Athena Query Federation 版本編號

1. 造訪 GitHub URL <https://github.com/awslabs/aws-athena-query-federation/releases/latest>。
2. 請注意主頁面標題中的版本編號，格式如下：

Release v *year.week\_of\_year.iteration\_of\_week* of Athena Query Federation

例如，Release v2023.8.3 of Athena Query Federation 的版本編號為 2023.8.3。

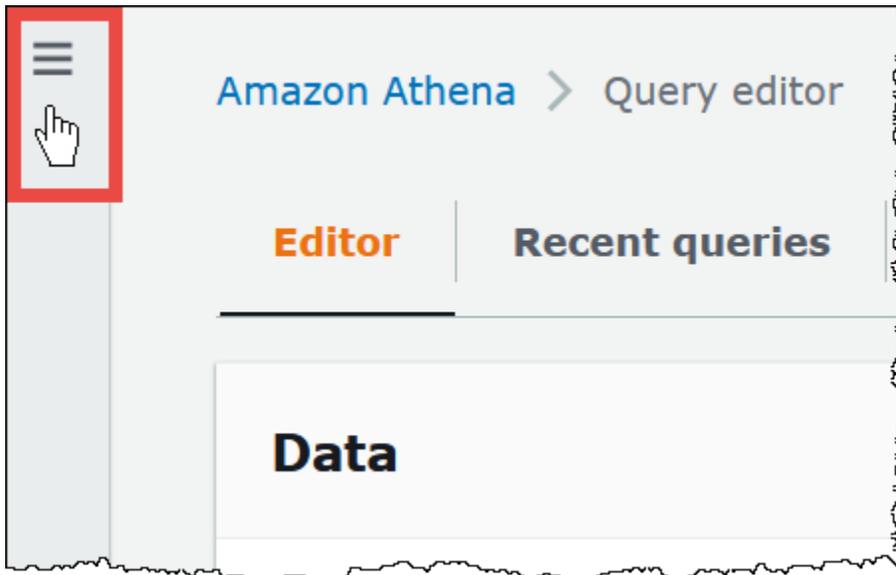
### 尋找並記下資源名稱

在準備升級時，您必須尋找並記下以下資訊：

1. 連接器的 Lambda 函數名稱。
2. Lambda 函數環境變數。
3. Lambda 應用程式名稱，可管理連接器的 Lambda 函數。

若要從 Athena 主控台尋找資源名稱

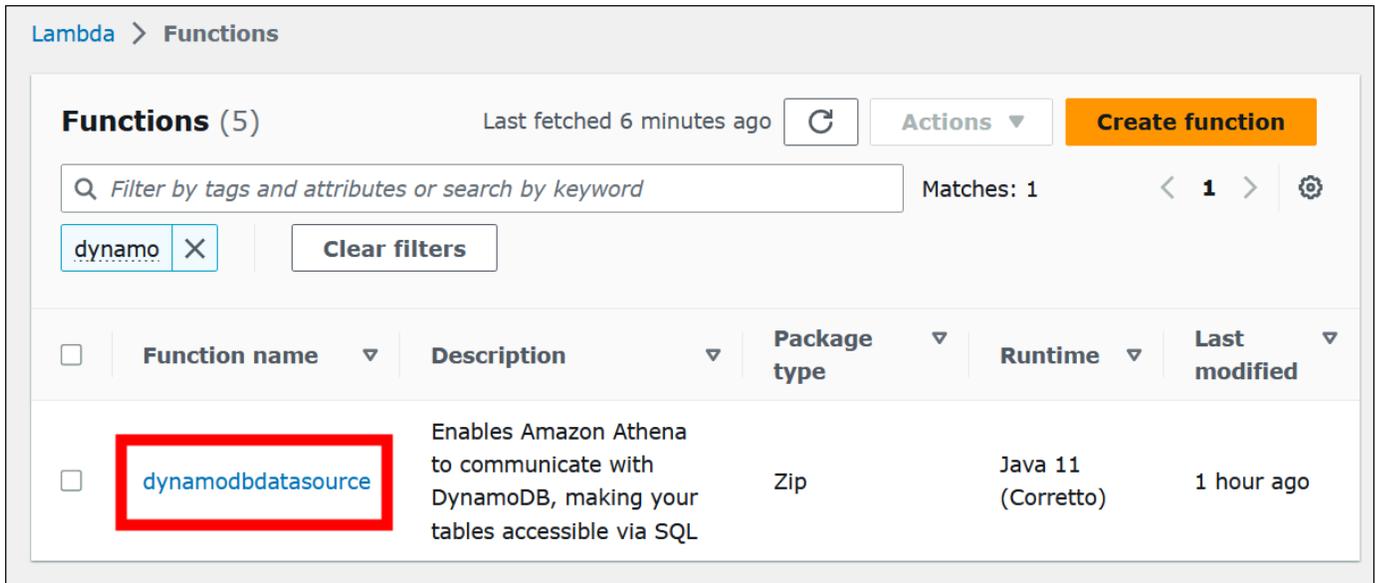
1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中選擇 Data sources (資料來源)。
4. 在資料來源名稱資料欄中，選擇連接器資料來源的連結。
5. 在資料來源詳細資訊區段的 Lambda 函數下，選擇 Lambda 函數的連結。

Data source details		
Data source name dynamo_db_catalog	Data source type Data source connector	Description DynamoDB Catalog
Lambda function <a href="#">dynamo_db_lambda</a>	Lambda function ARN <a href="#">arn:aws:lambda:us-west-2: [redacted] :function:dynamo_db_lambda</a>	

6. 在函數頁面的函數名稱資料欄中，記下連接器的函數名稱。



- 選擇函數名稱連結。
- 在函數概觀區段下，選擇組態索引標籤。
- 在左側的窗格中，選擇環境變數。
- 在環境變數區段中，記下索引鍵及其對應值。
- 捲動至頁面頂端。
- 在訊息此函數屬於一個應用程式。中 按一下此處進行管理，選擇按一下此處連結。
- 在 serverlessrepo-*your\_application\_name* 頁面上，記下不帶 serverlessrepo 的應用程式名稱。例如，如果應用程式名稱是 serverlessrepo-DynamoDbTestApp，則您的應用程式名稱就是 DynamoDbTestApp。
- 停留在應用程式的 Lambda 主控台頁面上，然後繼續執行尋找正在使用的連接器版本中的步驟。

### 尋找正在使用的連接器版本

依照以下步驟，尋找正在使用的連接器版本。

#### 若要尋找正在使用的連接器版本

- 在 Lambda 應用程式的 Lambda 主控台頁面上，選擇部署索引標籤。
- 在部署索引標籤上，展開 SAM 範本。
- 搜尋 CodeUri。
- 在 CodeUri 下的索引鍵欄位中，尋找下列字串：

```
applications-connector_name-  
versions-year.week_of_year.iteration_of_week/hash_number
```

下列範例顯示 CloudWatch 連接器的字串：

```
applications-AthenaCloudwatchConnector-versions-2021.42.1/15151159...
```

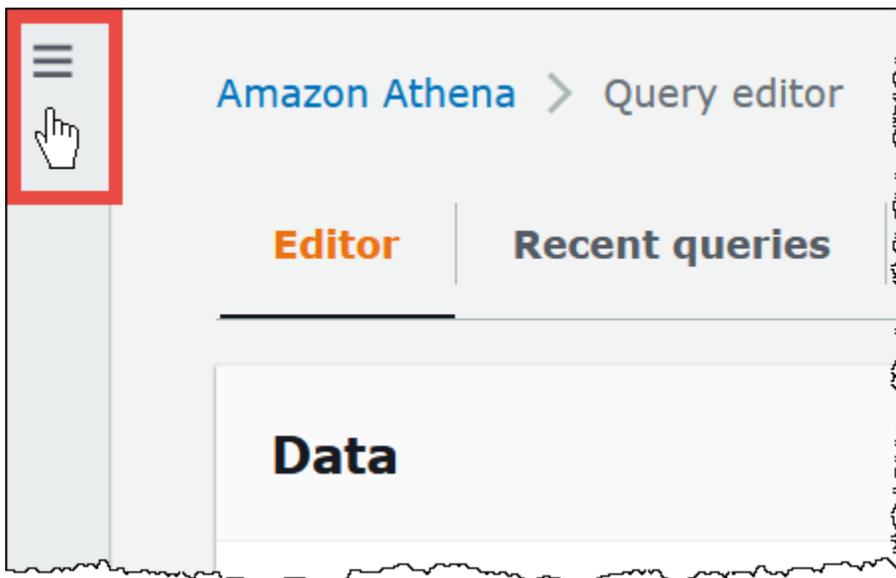
5. 記錄 *year.week\_of\_year.iteration\_of\_week* 的值 (例如 2021.42.1)。這是連接器的版本。

## 部署連接器的新版本

依照以下步驟，部署連接器的新版本。

若要部署連接器的新版本

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中選擇 Data sources (資料來源)。
4. 在 Data sources (資料來源) 頁面上，選擇 Create data source (建立資料來源)。
5. 選擇您想要升級的資料來源，然後選擇下一步。
6. 在連線詳細資訊區段中，選擇建立 Lambda 函數。這會開啟 Lambda 主控台，您將能在其中部署更新的應用程式。

Lambda > Applications > Review, configure and deploy

# AthenaDynamoDBConnector — version 2023.6.1

Review, configure and deploy

Copy as SAM Resource

## Application details

Author	Source code URL	Description	Report a vulnerability
<a href="#">Amazon Athena Federation</a>  AWS verified author	<a href="https://github.com/aws-labs/aws-athena-query-federation">https://github.com/aws-labs/a ws-athena-query-federation</a>	This connector enables Amazon Athena to communicate with DynamoDB, making your tables accessible via SQL.	If you believe this application poses a security risk, please file a vulnerability report.

▶ **Template**

▶ **Permissions**

▶ **License**

### Readme file

View on the [AWS Serverless Application Repository site](#).

### Application settings

Application name  
The stack name of this application created via AWS CloudFormation

AthenaDynamoDBConnector

- 由於您實際上並未建立新的資料來源，因此您可以關閉 Athena 主控台索引標籤。
- 在連接器的 Lambda 主控台頁面上，執行以下步驟：
  - 請確定您已從應用程式名稱中移除 serverlessrepo- 字首，然後將應用程式名稱複製到應用程式名稱欄位。
  - 將您的 Lambda 函數名稱複製到 AthenaCatalogName 欄位。某些連接器將此欄位稱為 LambdaFunctionName。
  - 將您記錄的環境變數複製到其對應欄位中。
- 選擇選項我認可此應用程式建立自訂的 IAM 角色和資源政策，然後選擇部署。

10. 若要確認您的應用程式是否已更新，請選擇部署索引標籤。

部署歷史記錄區段會顯示您的更新已完成。

Lambda > Applications > serverlessrepo-AthenaDynamoDBConnector

serverlessrepo-AthenaDynamoDBConnector

Overview | **Deployments** | Monitoring

► SAM template CloudFormation stack

Deployment history View stack events

Deployment	Resource type	Last updated time	Status
2 minutes ago	Lambda application	2 minutes ago	Update complete
last year	Lambda application	last year	Update complete
3 years ago	Lambda application	3 years ago	Update complete

11. 若要確認新的版本編號，您可以像以前一樣展開 SAM 範本，尋找 CodeUri，然後在索引鍵欄位中檢查連接器版本編號。

您現在可以使用更新的連接器來建立 Athena 聯合查詢。

## 執行聯合查詢

設定一或多個資料連接器並部署到您的帳戶之後，您就可以在 Athena 查詢中使用這些資料連接器。

### 查詢單一資料來源

本節中的範例假設您已設定 [Amazon Athena CloudWatch 連](#) 並將其部署到您的帳戶。使用其他連接器時，請使用相同的方法進行查詢。

若要建立使用 CloudWatch 連接器的 Athena 查詢

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在 Athena 查詢編輯器中，建立在 FROM 子句中使用下列語法的 SQL 查詢。

```
MyCloudwatchCatalog.database_name.table_name
```

## 範例

下列範例使用 Athena 連 CloudWatch 接器來連線至 `/var/e-commerce-engine/order-processor` CloudWatch 錄日誌群組中的 `all_log_streams` 檢視。 `all_log_streams` 檢視是日誌群組中所有日誌串流的檢視。範例查詢將傳回的列數限制為 100。

```
SELECT *
FROM "MyCloudwatchCatalog"."/var/e-commerce-engine/order-processor".all_log_streams
LIMIT 100;
```

下列範例剖析的資訊來自前一個範例的檢視。此範例擷取訂單 ID 和日誌等級，並篩選掉等級為 INFO 的任何訊息。

```
SELECT
  log_stream as ec2_instance,
  Regexp_extract(message '.*orderId=(\d+) .*', 1) AS orderId,
  message AS order_processor_log,
  Regexp_extract(message, '(.*):.*', 1) AS log_level
FROM MyCloudwatchCatalog."/var/e-commerce-engine/order-processor".all_log_streams
WHERE Regexp_extract(message, '(.*):.*', 1) != 'INFO'
```

## 查詢多個資料來源

舉一個更複雜的範例，假設一家電子商務公司使用以下資料來源來儲存與客戶購買相關的資料：

- [Amazon RDS for MySQL](#)，儲存產品目錄資料
- [Amazon DocumentDB](#)，儲存客戶帳戶資料，例如電子郵件和收貨地址
- [Amazon DynamoDB](#)，儲存訂單運送和追蹤資料

假設此電子商務應用程式的資料分析師了解到，某些區域的運送時間受到當地天氣狀況的影響。分析師想知道有多少訂單已延誤，受影響客戶所在的位置，以及哪些產品受影響最大。分析師使用 Athena 在單一聯合查詢中將資料連結起來，而不是個別調查資料來源。

## Example

```
SELECT
  t2.product_name AS product,
  t2.product_category AS category,
  t3.customer_region AS region,
  count(t1.order_id) AS impacted_orders
```

```
FROM my_dynamodb.default.orders t1
JOIN my_mysql.products.catalog t2 ON t1.product_id = t2.product_id
JOIN my_documentdb.default.customers t3 ON t1.customer_id = t3.customer_id
WHERE
    t1.order_status = 'PENDING'
    AND t1.order_date between '2022-01-01' AND '2022-01-05'
GROUP BY 1, 2, 3
ORDER BY 4 DESC
```

## 查詢聯合檢視

查詢聯合來源時，您可以使用檢視來混淆基礎資料來源，或隱藏來自其他查詢資料分析師的複雜聯結。

## 考量與限制

- 聯合檢視需要 Athena 引擎版本 3。
- 聯合檢視會儲存在中 AWS Glue，而非與基礎資料來源一起儲存。
- 使用聯合目錄建立的檢視必須使用完全合格的名稱語法，如下列範例所示：

```
"ddbcatalog"."default"."customers"
```

- 在聯合來源上執行查詢的使用者，必須擁有查詢聯合來源的許可。
- 聯合檢視需要 `athena:GetDataCatalog` 許可。如需詳細資訊，請參閱 [允許 Athena 聯合查詢的 IAM 許可政策範例](#)。

## 範例

以下範例為儲存在聯合資料來源中的資料建立一個名為 `customers` 的檢視。

### Example

```
CREATE VIEW customers AS
SELECT *
FROM my_federated_source.default.table
```

以下範例顯示一個查詢，其參考了 `customers` 檢視而非基礎聯合資料來源。

### Example

```
SELECT id, SUM(order_amount)
```

```
FROM customers
GROUP by 1
ORDER by 2 DESC
LIMIT 50
```

以下範例建立一個名為 `order_summary` 的檢視，該檢視結合了來自聯合資料來源和 Amazon S3 資料來源的資料。從已在 Athena 建立的聯合來源中，檢視會使用 `person` 和 `profile` 資料表。從 Amazon S3，檢視使用 `purchase` 和 `payment` 資料表。要參閱 Amazon S3，陳述式需使用關鍵字 `awsdatacatalog`。請注意，聯合資料來源會使用完整名稱語法 `federated_source_name.federated_source_database.federated_source_table`。

### Example

```
CREATE VIEW default.order_summary AS
SELECT *
FROM federated_source_name.federated_source_database."person" p
JOIN federated_source_name.federated_source_database."profile" pr ON pr.id = p.id
JOIN awsdatacatalog.default.purchase i ON p.id = i.id
JOIN awsdatacatalog.default.payment pay ON pay.id = p.id
```

### 其他資源

- 如需與原始來源分離並可在多使用者模型中進行隨選分析的聯合檢視範例，請參閱 AWS 大數據部落格中的 [使用 Amazon Athena 和聯合檢視擴展資料網格](#)。
- 如需有關在 Athena 中使用檢視的詳細資訊，請參閱 [使用檢視](#)。

### 執行同盟傳遞查詢

在 Athena 中，您可以使用資料來源本身的查詢語言對聯合資料來源執行查詢，並將完整查詢向下推送至資料來源以執行。這些查詢稱為直通查詢。若要執行傳遞查詢，請在 Athena 查詢中使用資料表函數。您可以將要在資料來源上執行的傳遞查詢包含在資料表函數的其中一個引數中。透過查詢傳回您可以使用 Athena SQL 進行分析的資料表。

### 支援連接器

下列 Athena 資料來源連接器支援傳遞查詢。

- [Azure Data Lake Storage](#)
- [Azure Synapse](#)

- [Cloudera Hive](#)
- [Cloudera Impala](#)
- [CloudWatch](#)
- [Db2](#)
- [Db2 集团](#)
- [DocumentDB](#)
- [DynamoDB](#)
- [HBase](#)
- [谷歌 BigQuery](#)
- [Hortonworks](#)
- [MySQL](#)
- [Neptune](#)
- [OpenSearch](#)
- [Oracle](#)
- [PostgreSQL](#)
- [Redshift](#)
- [SAP HANA](#)
- [Snowflake](#)
- [SQL Server](#)
- [Teradata](#)
- [Timestream](#)
- [Vertica](#)

## 考量與限制

在 Athena 中使用直通查詢時，請考慮以下幾點：

- 只有 Athena SELECT 陳述式或讀取作業才支援查詢傳遞。
- 傳遞查詢必須在外部查詢 (也就是呼叫 table 函數的查詢) 的目錄內容中執行。
- 查詢效能可能會因資料來源的組態而有所不同。

- 檢視不支援傳遞查詢。

## 語法

一般的 Athena 查詢傳遞語法如下。

```
SELECT * FROM TABLE(system.function_name(arg1 => 'arg1Value'[, arg2 =>
'arg2Value', ...]))
```

對於大多數資料來源，第一個也是唯一引數query後面接著箭頭運算子=>和查詢字串。

```
SELECT * FROM TABLE(system.query(query => 'query string'))
```

為了簡單起見，您可以省略可選的命名參數query和箭頭運算符=>。

```
SELECT * FROM TABLE(system.query('query string'))
```

如果資料來源所需的數量超過查詢字串，請按資料來源預期的順序使用具名引數。例如，運算式`arg1 => 'arg1Value'`包含第一個引數及其值。名稱`arg1`專屬於資料來源，且可能因連接器到連接器不同而有所不同。

```
SELECT * FROM TABLE(
  system.query(
    arg1 => 'arg1Value',
    arg2 => 'arg2Value',
    arg3 => 'arg3Value'
  ));
```

如需與特定連接器搭配使用的確切語法的相關資訊，請參閱個別連接器頁面。

## 引號用法

引數值 (包括您傳遞的查詢字串) 必須以單引號括住，如下列範例所示。

```
SELECT * FROM TABLE(system.query(query => 'SELECT * FROM testdb.persons LIMIT 10'))
```

當查詢字串被雙引號包圍時，查詢會失敗。以下查詢失敗，並出現錯誤消息列 `_NOT_FOUND`：行 1:43：列「從測試數據庫中選擇 \*。人員限制 10」無法解析。

```
SELECT * FROM TABLE(system.query(query => "SELECT * FROM testdb.persons LIMIT 10"))
```

若要逸出單引號，請在原始引號中加入單引號 (例如，`terry's_group` 到 `terry''s_group`)。

## 範例

下列範例查詢會將查詢向下推送至資料來源。查詢會選取資料 `customer` 表中的所有資料行，將結果限制為 10。

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10;' )  
    ))
```

下列陳述式會執行相同的查詢，但會排除選用的具名引數 `query` 和箭頭運算子 `=>`。

```
SELECT * FROM TABLE(  
    system.query(  
        'SELECT * FROM customer LIMIT 10;' )  
    ))
```

## Athena 和聯合資料表名稱限定詞

Athena 使用下列術語來表示資料物件的階層：

- 資料來源 – 一組資料庫
- 資料表 – 一組資料表
- 表格 – 組織為一組資料列或資料欄的資料

有時候，這些物件也會指稱為替代但對等的名稱，如下所示：

- 資料來源有時被稱為目錄。
- 資料庫有時被稱為結構描述。

下列 Athena 主控台內的範例查詢會使用 `awsdatacatalog` 資料來源、`default` 資料庫和 `some_table` 資料表。

The screenshot displays the Amazon Athena console interface. On the left, the 'Data' panel shows the 'Data source' set to 'AwsDataCatalog' and the 'Database' set to 'default'. Under 'Tables and views', 'some\_table' is selected. The SQL editor shows the query: `SELECT * FROM "awsdatacatalog"."default"."some_table" limit 10;`. The 'Query results' panel indicates the query is 'Completed' with a run time of 6.535 seconds and 0.91 KB of data scanned. The results table has 5 rows with columns '#', 'id', 'data', and 'category'.

#	id	data	category
1	1	a	A
2	3	d	d1
3	4	e	e1
4	4	f	f1
5	2	b	b1

## 聯合資料來源中的術語

查詢聯合資料來源時，請注意，基礎資料來源可能不會使用與 Athena 相同的術語。寫入聯合查詢時，請牢記這項差異性。下列各節說明 Athena 中的資料物件術語如何與聯合資料來源中的術語進行對應。

### Amazon Redshift

Amazon Redshift 資料庫是一組 Redshift 結構描述，其中包含一組 Redshift 資料表。

Athena	Redshift
Redshift 資料來源	設定為指向 Redshift database 的 Redshift 連接器 Lambda 函數。
<code>data_source.database.table</code>	<code>database.schema.table</code>

## 查詢範例

```
SELECT * FROM
Athena_Redshift_connector_data_source.Redshift_schema_name.Redshift_table_name
```

如需有關此連接器的詳細資訊，請參閱 [Amazon Athena Redshift 連接器](#)。

## Cloudera Hive

Cloudera Hive 伺服器或叢集是一組 Cloudera Hive 資料庫，其中包含一組 Cloudera Hive 資料表。

Athena	Hive
Cloudera Hive 資料來源	Cloudera Hive 連接器 Lambda 函數設定為指向 Cloudera Hive server。
<code>data_source.database.table</code>	<code>server.database.table</code>

## 查詢範例

```
SELECT * FROM
Athena_Cloudera_Hive_connector_data_source.Cloudera_Hive_database_name.Cloudera_Hive_table_name
```

如需有關此連接器的詳細資訊，請參閱 [Amazon Athena Cloudera Hive 連接器](#)。

## Cloudera Impala

Impala 伺服器或叢集是一組 Impala 資料庫，其中包含一組 Impala 資料表。

Athena	Impala
Impala 資料來源	Impala 連接器 Lambda 函數設定為指向 Impala server。
<code>data_source.database.table</code>	<code>server.database.table</code>

## 查詢範例

```
SELECT * FROM
Athena_Impala_connector_data_source.Impala_database_name.Impala_table_name
```

如需有關此連接器的詳細資訊，請參閱 [Amazon Athena Cloudera Impala 連接器](#)。

## MySQL

MySQL 伺服器是一組 MySQL 資料庫，其中包含一組 MySQL 資料表。

Athena	MySQL
MySQL 資料來源	MySQL 連接器 Lambda 函數設定為指向 MySQL server。
data_source.database.table	server.database.table

### 查詢範例

```
SELECT * FROM
Athena_MySQL_connector_data_source.MySQL_database_name.MySQL_table_name
```

如需有關此連接器的詳細資訊，請參閱 [Amazon Athena MySQL 連接器](#)。

## Oracle

Oracle 伺服器 (或資料庫) 是一組 Oracle 結構描述，其中包含一組 Oracle 資料表。

Athena	Oracle
Oracle 資料來源	Oracle 連接器 Lambda 函數設定為指向 Oracle server。
data_source.database.table	server.schema.table

### 查詢範例

```
SELECT * FROM
```

```
Athena_Oracle_connector_data_source.Oracle_schema_name.Oracle_table_name
```

如需有關此連接器的詳細資訊，請參閱 [Amazon Athena Oracle 連接器](#)。

## Postgres

Postgres 伺服器 (或叢集) 是一組 Postgres 資料庫。Postgres 資料庫是一組 Postgres 結構描述，其中包含一組 Postgres 資料表。

Athena	Postgres
Postgres 資料來源	Postgres 連接器 Lambda 函數設定為指向 Postgres server 和 database。
<code>data_source.database.table</code>	<code>server.database.schema.table</code>

## 查詢範例

```
SELECT * FROM
Athena_Postgres_connector_data_source.Postgres_schema_name.Postgres_table_name
```

如需有關此連接器的詳細資訊，請參閱 [Amazon Athena PostgreSQL 連接器](#)。

## 使用 Athena Query Federation SDK 開發資料來源連接器

若要撰寫您的 [資料來源連接器](#)，您可以使用 [Athena Query Federation 軟體開發套件](#)。Athena Query Federation 軟體開發套件定義一組界面和連線通訊協定，可讓 Athena 將部分查詢執行計劃委派給您撰寫和部署的程式碼。SDK 包含連接器套件和範例連接器。

您也可以自訂 Amazon Athena [預先建置的連接器](#) 供您使用。您可以從中修改原始程式碼的副本，GitHub 然後使用 [連接器發行工具](#) 建立您自己的 AWS Serverless Application Repository 套件。以這種方式部署連接器之後，您就可以在 Athena 查詢中使用連接器。

如需如何下載 SDK 的詳細資訊，以及撰寫自己連接器的詳細指示，請參閱開放 [範例 Athena 連接器 GitHub](#)。

## 適用於 Apache Spark 的 Athena 資料來源連接器

有些 Athena 資料來源連接器可做為 Spark DSV2 連接器使用。Spark DSV2 連接器名稱具有 `-dsv2` 字尾 (例如，`athena-dynamodb-dsv2`)。

以下是目前可用的 DSV2 連接器、其 Spark `.format()` 類別名稱，以及對應的 Amazon Athena 聯合查詢文件的連結：

DSV2 連接器	Spark <code>.format()</code> 類別名稱	文件
athena-cloudwatch-dsv2	<code>com.amazonaws.athena.connectors.dsv2.cloudwatch.CloudwatchTableProvider</code>	<a href="#">CloudWatch</a>
athena-cloudwatch-metrics-dsv2	<code>com.amazonaws.athena.connectors.dsv2.cloudwatch.metrics.CloudwatchMetricsTableProvider</code>	<a href="#">CloudWatch 度量</a>
athena-aws-cmdb-dsv2	<code>com.amazonaws.athena.connectors.dsv2.aws.cmdb.AwsCmdbTableProvider</code>	<a href="#">CMDB</a>
athena-dynamodb-dsv2	<code>com.amazonaws.athena.connectors.dsv2.dynamodb.DDBTableProvider</code>	<a href="#">DynamoDB</a>

若要下載 DSV2 連接器的 `.jar` 檔案，請瀏覽 [Amazon Athena 查詢聯盟 DSV2](#) GitHub 頁面，並參閱「發行版本、發行、資產」一節。 <version>

將 jar 指定至 Spark

若要搭配使用 Athena DSV2 連接器與 Spark，請將連接器的 `.jar` 檔案提交至您正在使用的 Spark 環境。下列各節說明了特定情況。

Athena for Spark

如需有關新增自訂 `.jar` 檔案和自訂組態至 Amazon Athena for Apache Spark 的資訊，請參閱 [新增 JAR 檔案和自訂 Spark 組態](#)。

## General Spark

若要將連接器 .jar 檔案傳遞給 Spark，請使用 `spark-submit` 命令並在 `--jars` 選項中指定 .jar 檔案，如下列範例所示：

```
spark-submit \  
  --deploy-mode cluster \  
  --jars https://github.com/awslabs/aws-athena-query-federation-dsv2/releases/  
download/some_version/athena-dynamodb-dsv2-some_version.jar
```

## Amazon EMR Spark

為在 Amazon EMR 上使用 `--jars` 參數執行 `spark-submit` 命令，您必須新增步驟至您的 Amazon EMR Spark 叢集。如需有關如何使用 Amazon EMR 上的 `spark-submit`，請參閱《Amazon EMR 版本指南》中的[新增 Spark 步驟](#)。

## AWS Glue ETL 星火

對於 AWS Glue ETL，您可以將 .jar 檔案的 GitHub .com URL 傳遞給 `aws glue start-job-run` 命令的 `--extra-jars` 引數。本 AWS Glue 文件將 `--extra-jars` 參數描述為採用 Amazon S3 路徑，但該參數也可以採用 HTTPS 網址。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的[任務參數參考](#)。

## 在 Spark 上查詢連接器

若要在 Apache Spark 上提交您現有的 Athena 聯合查詢的等式，請使用 `spark.sql()` 函數。例如，假設您想要在 Apache Spark 上使用如下所示的 Athena 查詢。

```
SELECT somecola, somecolb, somecolc  
FROM ddb_datasource.some_schema_or_glue_database.some_ddb_or_glue_table  
WHERE somecola > 1
```

若要使用 Amazon Athena DynamoDB DSV2 連接器在 Spark 上執行相同的查詢，請使用下列程式碼：

```
dynamoDf = (spark.read  
  .option("athena.connectors.schema", "some_schema_or_glue_database")  
  .option("athena.connectors.table", "some_ddb_or_glue_table")  
  .format("com.amazonaws.athena.connectors.dsv2.dynamodb.DDBTableProvider")  
  .load())
```

```
dynamoDf.createOrReplaceTempView("ddb_spark_table")

spark.sql('''
SELECT somecola, somecolb, somecolc
FROM ddb_spark_table
WHERE somecola > 1
''')
```

## 指定參數

Athena 資料來源連接器的 DSV2 版本使用的參數，與對應 Athena 資料來源連接器使用的參數相同。如需參數資訊，請參閱對應 Athena 資料來源連接器的文件。

在您的 PySpark 程式碼中，使用下列語法來設定參數。

```
spark.read.option("athena.connectors.conf.parameter", "value")
```

例如，下列程式碼會將 Amazon Athena DynamoDB 連接器 `disable_projection_and_casing` 參數設定為 `always`。

```
dynamoDf = (spark.read
    .option("athena.connectors.schema", "some_schema_or_glue_database")
    .option("athena.connectors.table", "some_ddb_or_glue_table")
    .option("athena.connectors.conf.disable_projection_and_casing", "always")
    .format("com.amazonaws.athena.connectors.dsv2.dynamodb.DDBTableProvider")
    .load())
```

## 用於存取資料目錄的 IAM 政策

若要控制對資料目錄的存取，請使用資源層級 IAM 許可或身分型 IAM 政策。

下列程序專用於 Athena。

如需 IAM 特定的資訊，請參閱本節最後列出的連結。如需範例 JSON 資料目錄政策的相關資訊，請參閱[資料目錄範例政策](#)。

若要在 IAM 主控台中使用視覺化編輯器來建立資料目錄政策

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在左邊的導覽窗格中，選擇 Policies (政策)，然後選擇 Create policy (建立政策)。

3. 在 Visual editor (視覺化編輯器) 標籤上，選擇 Choose a service (選擇一項服務)。接著選擇 Athena 以新增到政策。
4. 選擇 Select actions (選取動作)，然後選擇要新增到政策的動作。視覺化編輯器會顯示 Athena 中可用的動作。如需詳細資訊，請參閱《服務授權參考》中的 [Amazon Athena 的動作、資源和條件索引鍵](#)。
5. 選擇 add actions (新增動作) 來輸入特定動作，或使用萬用字元 (\*) 來指定多個動作。  
  
預設情況下，您建立的政策允許執行選擇的操作。對於 Athena 中的 datacatalog 資源，如果您選擇的一個或多個動作支援資源層級許可，則編輯器會列出 datacatalog 資源。
6. 選擇 Resources (資源) 來為您的政策指定資料目錄。如需 JSON 資料目錄政策範例，請參閱[資料目錄範例政策](#)。
7. 如下所示指定 datacatalog 資源：

```
arn:aws:athena:<region>:<user-account>:datacatalog/<datacatalog-name>
```

8. 選擇 Review policy (檢閱政策)，然後為您正在建立的政策輸入 Name (名稱) 與 Description (描述) (選用)。檢閱政策摘要來確認您已授予想要的許可。
9. 選擇 Create policy (建立政策) 儲存您的新政策。
10. 將此以身分為基礎的政策連接到使用者、群組或角色，並指定其可存取的 datacatalog 資源。

如需詳細資訊，請參閱《服務授權參考》與《IAM 使用者指南》中的下列主題：

- [Amazon Athena 的操作、資料和條件索引鍵](#)
- [使用視覺化編輯器來建立政策](#)
- [新增和移除 IAM 政策](#)
- [控制資源的存取](#)

如需 JSON 資料目錄政策範例，請參閱[資料目錄範例政策](#)。

如需有關 AWS Glue 權限和 AWS Glue 爬行者程式許可的詳細資訊，請參閱開發人員指南中的[設定 IAM 許可 AWS Glue](#)和[爬行AWS Glue 者程式先決條件](#)。

如需 Amazon Athena 動作的完整清單，請參閱《[Amazon Athena API 參考](#)》中的 API 動作名稱。

## 資料目錄範例政策

本節包含可讓您在資料目錄上用來啟用各種動作的政策範例。

資料目錄是由 Athena 管理的 IAM 資源。因此，如果您的資料目錄政策使用 `datacatalog` 做為輸入的動作，您必須指定資料目錄的 ARN，如下所示：

```
"Resource": [arn:aws:athena:<region>:<user-account>:datacatalog/<datacatalog-name>]
```

<datacatalog-name> 是資料目錄的名稱。例如，對於名為 `test_datacatalog` 的資料目錄，將其指定為資源，如下所示：

```
"Resource": ["arn:aws:athena:us-east-1:123456789012:datacatalog/test_datacatalog"]
```

如需 Amazon Athena 動作的完整清單，請參閱《[Amazon Athena API 參考](#)》中的 API 動作名稱。如需有關 IAM 政策的詳細資訊，請參閱《IAM 使用者指南》中的[使用視覺化編輯器來建立政策](#)。如需有關為工作群組建立 IAM 政策的詳細資訊，請參閱[用於存取資料目錄的 IAM 政策](#)。

- [Example Policy for Full Access to All Data Catalogs](#)
- [Example Policy for Full Access to a Specified Data Catalog](#)
- [Example Policy for Querying a Specified Data Catalog](#)
- [Example Policy for Management Operations on a Specified Data Catalog](#)
- [Example Policy for Listing Data Catalogs](#)
- [Example Policy for Metadata Operations on Data Catalogs](#)

Example 適用於完整存取所有資料目錄的政策範例

以下政策允許完整存取帳戶中可能存在的所有資料目錄資源。對於您的帳戶中必須為所有其他使用者管理資料目錄的那些使用者，我們建議您使用此政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

### Example 適用於完整存取指定資料目錄的政策範例

以下政策允許完整存取名為 `datacatalogA` 的單一特定的資料目錄資源。對於能夠完全控制特定資料目錄的使用者，您可以使用此政策。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "athena:ListDataCatalogs",  
        "athena:ListWorkGroups",  
        "athena:GetDatabase",  
        "athena:ListDatabases",  
        "athena:ListTableMetadata",  
        "athena:GetTableMetadata"  
      ],  
      "Resource": "*"   
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "athena:StartQueryExecution",  
        "athena:GetQueryResults",  
        "athena>DeleteNamedQuery",  
        "athena:GetNamedQuery",  
        "athena:ListQueryExecutions",  
        "athena:StopQueryExecution",  
        "athena:GetQueryResultsStream",  
        "athena:ListNamedQueries",  
        "athena:CreateNamedQuery",  
        "athena:GetQueryExecution",  
        "athena:BatchGetNamedQuery",  
        "athena:BatchGetQueryExecution",  
        "athena>DeleteWorkGroup",  
        "athena:UpdateWorkGroup",  
        "athena:GetWorkGroup",  
        "athena:CreateWorkGroup"  
      ]  
    }  
  ]  
}
```

```

    ],
    "Resource": [
      "arn:aws:athena:us-east-1:123456789012:workgroup/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "athena:CreateDataCatalog",
      "athena>DeleteDataCatalog",
      "athena:GetDataCatalog",
      "athena:GetDatabase",
      "athena:GetTableMetadata",
      "athena>ListDatabases",
      "athena>ListTableMetadata",
      "athena:UpdateDataCatalog"
    ],
    "Resource": "arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA"
  }
]
}

```

### Example 適用於查詢指定資料目錄的政策範例

在以下政策中，允許使用者在指定的 datacatalogA 中執行查詢。不允許使用者對資料目錄本身執行管理任務，例如更新或刪除。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:StartQueryExecution"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetDataCatalog"
      ]
    }
  ]
}

```

```
    ],
    "Resource": [
      "arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA"
    ]
  }
]
```

### Example 適用於指定資料目錄上管理操作的策略範例

在以下政策中，允許使用者建立、刪除、取得詳細資訊和更新資料目錄 datacatalogA。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateDataCatalog",
        "athena:GetDataCatalog",
        "athena>DeleteDataCatalog",
        "athena:UpdateDataCatalog"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA"
      ]
    }
  ]
}
```

### Example 適用於列出資料目錄的政策範例

以下政策可讓所有使用者列出所有資料目錄：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListDataCatalogs"
      ],

```

```
        "Resource": "*"
    }
]
}
```

## Example 適用於資料目錄上中繼資料操作的政策範例

下列政策允許對資料目錄進行中繼資料操作：

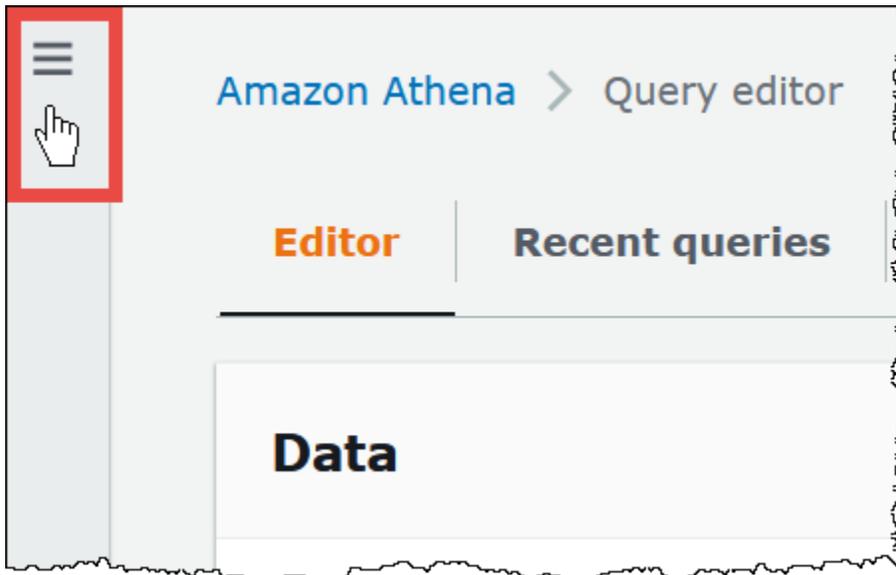
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetDatabase",
        "athena:GetTableMetadata",
        "athena:ListDatabases",
        "athena:ListTableMetadata"
      ],
      "Resource": "*"
    }
  ]
}
```

## 管理資料來源

您可以使用 Athena 主控台的 Data Sources (資料來源) 頁面來管理您建立的資料來源。

### 檢視資料來源

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中選擇 Data sources (資料來源)。
4. 從資料來源清單中選擇您想要檢視的資料來源的名稱。

#### Note

Data source name (資料來源名稱) 資料欄中的項目對應至 [ListDataCatalogs](#) API 動作和 [list-data-catalogs](#) CLI 命令的輸出。

## 編輯資料來源

1. 在 Data Source (資料來源) 頁面上執行下列其中一項動作：
  - 選擇目錄名稱旁邊的按鈕，然後選擇 Actions (動作)、Edit (編輯)。
  - 選擇資料來源的名稱。然後在詳細資訊頁面上，選擇 Actions (動作)、Edit (編輯)。
2. 在 Edit (編輯) 頁面上，您可以為資料來源選擇不同的 Lambda 函數，變更其描述或新增自訂標籤。如需標籤的詳細資訊，請參閱 [標記 Athena 資源](#)。
3. 選擇 Save (儲存)。
4. 如需編輯 AwsDataCatalog 資料來源，請選擇 AwsDataCatalog 連結以開啟其詳細資訊頁面。接著，在詳細資訊頁面上，選擇 AWS Glue 主控台的連結，即可在其中編輯目錄。

## 共用資料來源

如需有關共用資料來源的資訊，請造訪下列連結。

- 如需瞭解非 Hive Lambda 型資料來源，請參閱 [啟用跨帳戶聯合查詢](#)。
- 如需瞭解 AWS Glue Data Catalog，請參閱 [跨帳戶存取 AWS Glue 資料目錄](#)。

## 刪除資料來源

1. 在 Data source (資料來源) 頁面上執行下列其中一項動作：
  - 選取目錄名稱旁的按鈕，然後選擇 Actions (動作)、Delete (刪除)。
  - 選擇資料來源的名稱，然後在詳細資訊頁面上，依序選擇 Actions (動作)、Delete (刪除)。

### Note

AwsDataCatalog 是帳戶中的預設資料來源且無法刪除。

當您刪除資料來源時，會收到系統的警告，其對應的資料目錄、資料表和檢視會從查詢編輯器中移除。使用資料來源的已儲存查詢不再於 Athena 中執行。

2. 如需確認刪除，請輸入資料來源的名稱，然後選擇 Delete (刪除)。

## 在 Athena 中使用 Amazon DataZone

您可以使用 [Amazon DataZone](#) 來跨組織界限大規模共用、搜尋和探索資料。DataZone 可簡化您在各種 AWS 分析服務 (例如 Athena、AWS Glue 和 AWS Lake Formation) 的體驗。例如，如果您在不同的資料來源中有數 PB 的資料，則可以使用 Amazon DataZone 建立以商業使用案例為基礎的人員、資料和工具群組。如需詳細資訊，請參閱 [什麼是 Amazon DataZone ?](#)。

在 Athena，您可以使用查詢編輯器來存取和查詢 DataZone 環境。一個 DataZone 環境指定一個 DataZone 專案和域的組合。當您從 Athena 主控台使用 DataZone 環境時，您會擔任 DataZone 環境的 IAM 角色，而且只會看到屬於該環境的資料庫和資料表。許可由您在 DataOne 中指定的角色決定。

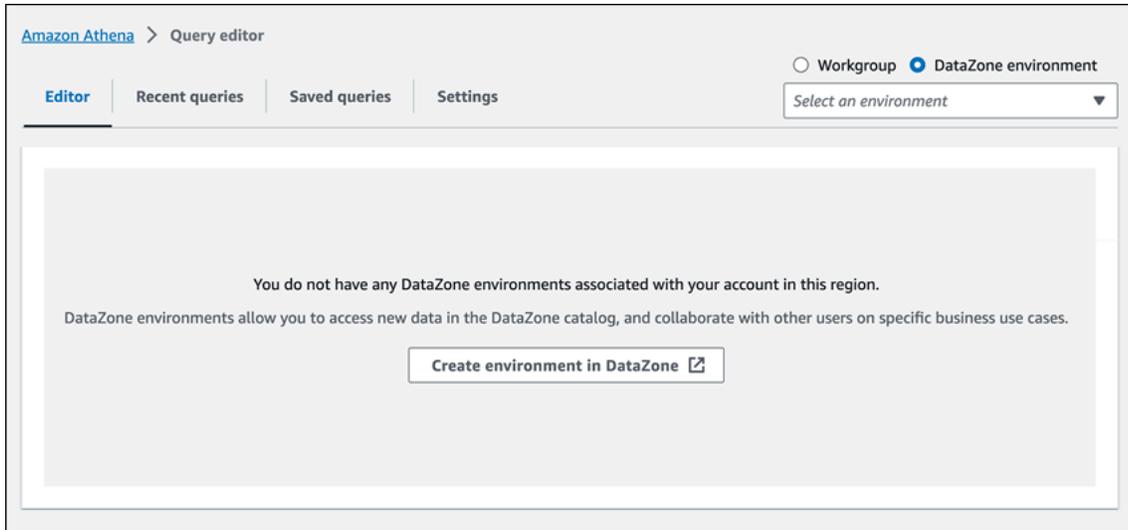
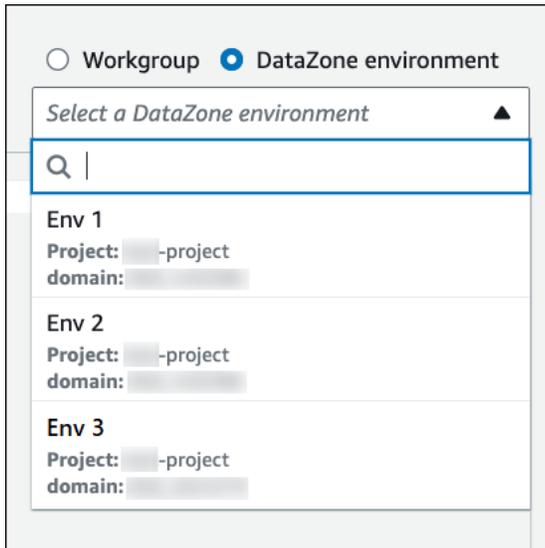
在 Athena，您可以使用查詢編輯器頁面上的 DataZone 環境選擇器來選擇 DataZone 環境。

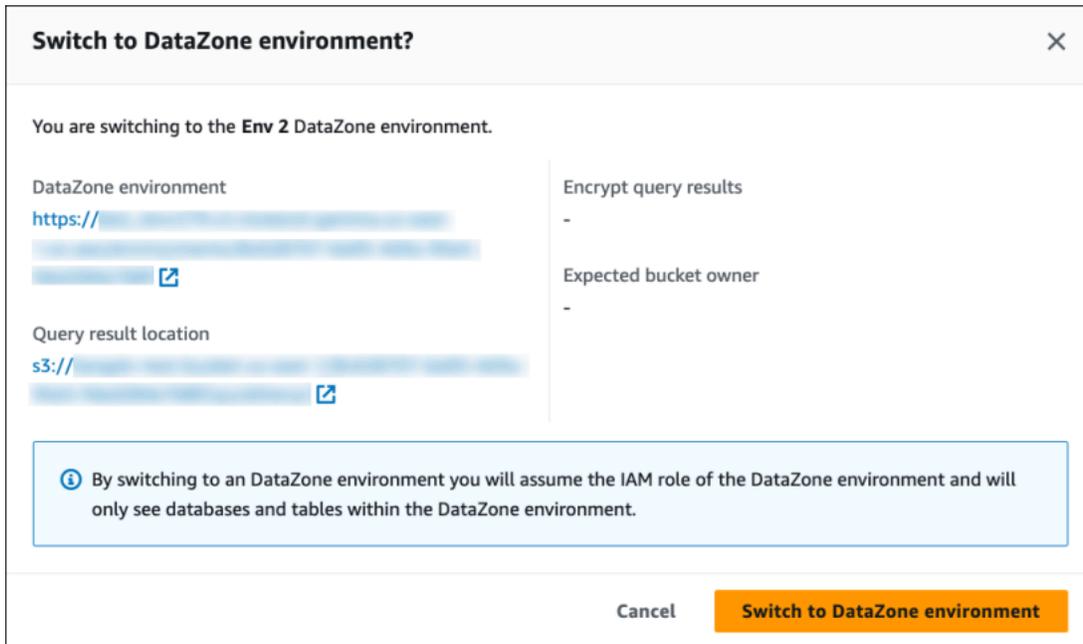
若要在 Athena 中開啟 DataZone 環境

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在 Athena 主控台右上角的工作群組旁邊，選擇 DataZone 環境。

**Note**

只有當您在 DataOne 中有一或多個域可用時，才會顯示 DataZone 環境選項。

**3. 使用 DataZone 環境選擇器來選擇 DataZone 環境。****4. 在切換至 DataZone 環境對話方塊中，確認該環境是您想要的環境，然後選擇切換至 DataZone 環境。**



如需開始使用 DataZone 和 Athena 的相關詳細資訊，請參閱《Amazon DataZone 使用者指南》中的 [入門教學課程](#)。

## 使用 ODBC 和 JDBC 驅動程式連接到 Amazon Athena

為了使用商業智慧工具探索您的資料並加以和視覺化，請下載、安裝並設定 ODBC (開放資料庫連線) 或 JDBC (Java 資料庫連線) 驅動程式。

### 主題

- [使用 JDBC 連接到 Amazon Athena](#)
- [使用 ODBC 連接到 Amazon Athena](#)

另請參閱下列 AWS 知識中心和 AWS 大數據部落格主題：

- [使用 JDBC 驅動程式連接至 Athena 時，如何使用 IAM 角色憑證或切換至其他 IAM 角色？](#)
- [在 ADFS 和使用活動目錄登入資料之間設定信任，以 AWS 及使用 ODBC 驅動程式連接到 Amazon Athena](#)

## 使用 JDBC 連接到 Amazon Athena

Amazon Athena 提供兩個 JDBC 驅動程式：版本 2.x 和 3.x。Athena JDBC 3.x 驅動程式是新一代驅動程式，可提供更好的效能和相容性。JDBC 3.x 驅動程式支援直接從 Amazon S3 讀取查詢結果，可改善使用大量查詢結果的應用程式效能。新的驅動程式也有較少的協力廠商相依性，這使得與 BI 工具和自訂應用程式的整合更容易。在大多數情況下，您可以使用新的驅動程序，而不需要對現有配置進行任何更改或最少的

- 若要下載 JDBC 3.x 驅動程式，請參閱 [Athena JDBC 3.x 驅動程式](#)。
- 若要下載 JDBC 2.x 驅動程式，請參閱 [Athena JDBC 2.x 驅動程式](#)。

### 主題

- [Athena JDBC 3.x 驅動程式](#)
- [Athena JDBC 2.x 驅動程式](#)

## Athena JDBC 3.x 驅動程式

您可以從許多第三方 SQL 用戶端工具和自訂應用程式使用 Athena ODBC 驅動程式連接至 Amazon Athena。

### 系統要求

- Java 8 (或更高版本) 執行期環境
- 至少 20 MB 的可用磁碟空間

### 考量與限制

以下是 Athena JDBC 3.x 驅動程式的一些考量與限制。

- 記錄日誌 – 3.x 驅動程式使用 [SLF4J](#)，這是可在執行期啟用數個記錄日誌系統其中之一的抽象層。
- 加密 – 搭配使用 Amazon S3 擷取程式與 CSE\_KMS 加密選項時，Amazon S3 用戶端無法解密存放於 Amazon S3 儲存貯體中的結果。如果您需要 CSE\_KMS 加密，則可以繼續使用串流擷取程式。已計劃支援搭配使用 CSE\_KMS 加密與 Amazon S3 擷取程式。

## JDBC 3.x 驅動程式下載

本節包含 JDBC 3.x 驅動程式的下載和授權資訊。

### Important

當您使用 JDBC 3.x 驅動程式時，務必注意以下要求：

- 打開連接埠 444 – 將 Athena 用來串流查詢結果的連接埠 444 保持開放給輸出流量。當您使用 PrivateLink 端點連線至 Athena 時，請確定連接到 PrivateLink 端點的安全群組對連接埠 444 的輸入流量開放。
- 雅典娜/ GetQueryResultsStream 政策 — 將athena:GetQueryResultsStream政策動作新增至使用 JDBC 驅動程式的 IAM 主體。此政策動作不會直接透過 API 公開。它僅與 ODBC 和 JDBC 驅動程式搭配使用，做為串流結果支援的一部分。如需政策範例，請參閱 [AWS 受管理的策略：AWSQuicksightAthenaAccess](#)。

若要下載 Amazon Athena 3.x JDBC 驅動程式，請造訪以下連結。

#### JDBC 驅動程式 uber jar

下面的下載將驅動程式及其所有相依性包在同一個 .jar 檔案中。此下載通常用於第三方 SQL 用戶端。

#### [3.2.0 尤伯杯罐 尤伯杯罐](#)

#### JDBC 驅動程式 lean jar

下面的下載是一個 .zip 檔案，其中包含驅動程式的 lean .jar 和驅動程式相依性的單獨 .jar 檔案。此下載通常用於可能具有相依性與驅動程式使用的相依性衝突的自訂應用程式。如果您想選擇要包含在 lean jar 中的驅動程式相依性，以及要排除的驅動程式相依性 (如果您的自訂應用程式已包含一或多個驅動程式相依性)，則此下載非常有用。

#### [瘦肉罐子 瘦肉罐](#)

#### 授權

下列連結包含 JDBC 3.x 驅動程式的授權合約。

#### [授權](#)

#### 主題

- [JDBC 3.x 驅動程式入門](#)

- [Amazon Athena JDBC 3.x 連線參數](#)
- [其他 JDBC 3.x 組態](#)
- [Amazon Athena JDBC 3.x 版本資訊](#)
- [先前版本的 Athena JDBC 3.x 驅動程序](#)

## JDBC 3.x 驅動程式入門

使用本節中的資訊，開始使用 Amazon Athena JDBC 3.x 驅動程式。

### 主題

- [安裝說明](#)
- [執行驅動程式](#)
- [設定驅動程式](#)
- [從 Athena JDBC v2 驅動程式升級](#)

## 安裝說明

您可以在自訂應用程式或第三方 SQL 用戶端中，使用 JDBC 3.x 驅動程式。

### 在自訂應用程式中

下載包含驅動程式 jar 及其相依性的 .zip 檔案。每個相依性都有自己的 .jar 檔案。在自訂應用程式中新增驅動程式 jar 作為相依性。根據您是否已經從另一個來源新增這些相依性至應用程式中，選擇性地新增驅動程式 jar 的相依性。

### 在第三方 SQL 用戶端中

下載驅動程式 uber jar 檔案，並依照該用戶端的指示將其新增至第三方 SQL 用戶端。

### 執行驅動程式

若要執行驅動程式，您可以使用自訂應用程式或第三方 SQL 用戶端。

### 在自訂應用程式中

使用 JDBC 介面與程式的 JDBC 驅動程式進行互動。下列程式碼顯示了一個自訂 Java 應用程式範例。

```
public static void main(String args[]) throws SQLException {
```

```
Properties connectionParameters = new Properties();
connectionParameters.setProperty("Workgroup", "primary");
connectionParameters.setProperty("Region", "us-east-2");
connectionParameters.setProperty("Catalog", "AwsDataCatalog");
connectionParameters.setProperty("Database", "sampledatabase");
connectionParameters.setProperty("OutputLocation", "s3://DOC-EXAMPLE-BUCKET");
connectionParameters.setProperty("CredentialsProvider", "DefaultChain");
String url = "jdbc:athena://";
AthenaDriver driver = new AthenaDriver();
Connection connection = driver.connect(url, connectionParameters);
Statement statement = connection.createStatement();
String query = "SELECT * from sample_table LIMIT 10";
ResultSet resultSet = statement.executeQuery(query);
printResults(resultSet); // A custom-defined method for iterating over a
                          // result set and printing its contents
}
```

### 在第三方 SQL 用戶端中

遵循您所使用之 SQL 用戶端的文件。一般而言，您可以使用 SQL 用戶端的圖形化使用者介面來輸入及提交查詢，而查詢結果會顯示在相同的介面中。

### 設定驅動程式

您可以使用連線參數來設定 Amazon Athena JDBC 驅動程式。如需支援的連線參數，請參閱 [Amazon Athena JDBC 3.x 連線參數](#)。

### 在自訂應用程式中

若要在自訂應用程式中設定 JDBC 驅動程式的連線參數，請執行下列其中一個動作：

- 將參數名稱及其值新增至 Properties 物件。當您呼叫 Connection#connect 時，將該物件與 URL 一起傳遞。如需範例，請參閱 [執行驅動程式](#) 中的 Java 範例應用程式。
- 在連線字串 (URL) 中，使用下列格式將參數名稱及其值直接新增至通訊協定字首之後。

```
<parameterName>=<parameterValue>;
```

在每個參數名稱/參數值配對的末尾使用分號，分號後不留空格，如下列範例所示。

```
String url = "jdbc:athena://WorkGroup=primary;Region=us-east-1;...";AthenaDriver
driver = new AthenaDriver();Connection connection = driver.connect(url, null);
```

**Note**

如果在連線字串和 Properties 物件中同時指定參數，則連線字串中的值優先。不建議在兩個位置指定相同的參數。

- 將參數值作為引數新增至 AthenaDataSource 的方法，如下列範例所示。

```
AthenaDataSource dataSource = new AthenaDataSource();
dataSource.setWorkGroup("primary");
dataSource.setRegion("us-east-2");
...
Connection connection = dataSource.getConnection();
...
```

### 在第三方 SQL 用戶端中

依照您所使用之 SQL 用戶端的指示進行。通常，用戶端會提供圖形化使用者介面來輸入參數名稱及其值。

### 從 Athena JDBC v2 驅動程式升級

大多數 JDBC 第 3 版連接參數都與第 2 版 ( 辛巴 ) JDBC 驅動程序向後兼容。這意味著第 2 版連線字串可以與第 3 版驅動程式重複使用。但是，某些連線參數已更改。這些變更會在這裡說明。當您升級至第 3 版 JDBC 驅動程式時，請視需要更新現有的組態。

### 驅動程式類別

有些 BI 工具會要求您從 JDBC 驅動程式 .jar 檔案提供驅動程式類別。大多數工具都會自動尋找這個類別。第 3 版驅動程式中完全合格的類別名稱為 `com.amazon.athena.jdbc.AthenaDriver`。在第 2 版驅動程式中，該類別為 `com.simba.athena.jdbc.Driver`。

### 連接字串

版本 3 驅動程式用 `jdbc:athena://` 於 JDBC 連接字串 URL 開頭的通訊協定。版本 3 驅動程式也支援版本 2 通訊協定 `jdbc:awsathena://`，但版本 2 通訊協定的使用已被棄用。為了避免未定義的行為，版本 3 不接受以版本 2 開頭的連接字符串 ( 或任何其他接受以連接字符串開頭的驅動程序 `jdbc:awsathena://` ) 已在 [DriverManager](#) 類中註冊的連接字符串。 `jdbc:awsathena://`

## 憑證提供者

第 2 版驅動程式會使用完全合格的名稱來辨識不同的憑證提供者 (例如 `com.simba.athena.amazonaws.auth.DefaultAWSCredentialsProviderChain`)。第 3 版驅動程式則使用較短的名稱 (例如 `DefaultChain`)。新名稱會在每個憑證提供者的對應章節中說明。

您必須修改為第 2 版驅動程式撰寫的自訂認證提供者，才能使版本 3 驅動程式實作新的 [AwsCredentialsProvider](#) 介面，AWS SDK for Java 而不是先前的 [AWSCredentialsProvider](#) 介面 AWS SDK for Java。

JDBC 3.x 驅動程式不支援 `PropertiesFileCredentialsProvider` 該提供程序在 JDBC 2.x 驅動程式中使用，但屬於即將結束支持的 AWS SDK for Java 的以前版本。若要在 JDBC 3.x 驅動程式中達到相同的功能，請改用 [AWS 組態設定檔憑證](#) 提供者。

## 日誌層級

下表顯示 JDBC 第 2 版和第 3 版驅動程式中的 `LogLevel` 參數差異。

JDBC 驅動程式版本	參數名稱	參數類型	預設值	可能的值	連線字串範例
v2	<code>LogLevel</code>	選用	0	0-6	<code>LogLevel=6;</code>
v3	<code>LogLevel</code>	選用	TRACE	OFF、ERROR、WARN、INFO、DEBUG、TRACE	<code>LogLevel=INFO;</code>

## 查詢 ID 擷取

在第 2 版驅動程式中，您可以將 `Statement` 執行個體取消包裝至 `com.interfaces.core.IStatementQueryInfoProvider`，一個具有兩種方法 `#getPReparedQueryId` 和 `#getQueryId` 的介面。您可以使用這些方法來取得已執行之查詢的查詢執行 ID。

在第 3 版驅動程式中，您可以將 `Statement`、`PreparedStatement` 和 `ResultSet` 執行個體取消包裝至 `com.amazon.athena.jdbc.AthenaResultSet` 介面。該介面有一種方法：`#getQueryExecutionId`。

## Amazon Athena JDBC 3.x 連線參數

支援的連線參數在此分為三個區段：[基本連線參數](#)、[進階連線參數](#) 和 [身分驗證連線參數](#)。「進階」連線參數和「身分驗證」連線參數區段都有將相關參數群組在一起的子區段。

### 主題

- [基本連線參數](#)
- [進階連線參數](#)
- [身分驗證連線參數](#)

### 基本連線參數

下列各節說明 JDBC 3.x 驅動程式的基本連線參數。

#### 區域

在 AWS 區域 哪裡查詢將被運行。如需區域清單，請參閱 [Amazon Athena 端點和配額](#)。

參數名稱	別名	參數類型	預設值
區域	AwsRegion (已取代)	強制性 (但如果未提供，將使用搜尋 <a href="#">DefaultAwsRegionProviderChain</a> )	無

#### 目錄

包含將透過驅動程式存取的資料庫和資料表的目錄。如需目錄的相關資訊，請參閱[DataCatalog](#)。

參數名稱	別名	參數類型	預設值
目錄	無	選用	AwsDataCatalog

### 資料庫

將在其中執行查詢的資料庫。未明確限定資料庫名稱的資料表將解析至此資料庫。如需有關資料庫的資訊，請參閱[資料庫](#)。

參數名稱	別名	參數類型	預設值
資料庫	結構描述	選用	預設

## 工作群組

將在其中執行查詢的工作群組。如需工作群組的相關資訊，請參閱[WorkGroup](#)。

參數名稱	別名	參數類型	預設值
WorkGroup	無	選用	主要

## 輸出位置

Amazon S3 中儲存查詢結果的位置。如需有關輸出位置的資訊，請參閱[ResultConfiguration](#)。

參數名稱	別名	參數類型	預設值
OutputLocation	S3 OutputLocation (已淘汰)	強制性 (除非工作群組指定了輸出位置)	無

## 進階連線參數

下列各節說明 JDBC 3.x 驅動程式的進階連線參數。

### 主題

- [結果加密參數](#)
- [結果擷取參數](#)
- [查詢結果重複使用參數](#)
- [查詢執行輪詢參數](#)
- [端點覆寫參數](#)
- [代理組態參數](#)
- [記錄日誌參數](#)
- [應用程式名稱](#)

- [連線測試](#)
- [重試次數](#)

## 結果加密參數

請注意以下重點：

- AWS KMS 金鑰必須在EncryptionOption為SSE\_KMS或時指定CSE\_KMS。
- 如果未指定或未指定 AWS KMS 金鑰，EncryptionOption則無法指定金鑰SSE\_S3。EncryptionOption

## 加密選項

查詢結果存放在 Amazon S3 時所使用的加密類型。如需查詢結果加密的相關資訊，請參閱 Amazon Athena API 參考[EncryptionConfiguration](#)中的。

參數名稱	別名	參數類型	預設值	可能的值
EncryptionOption	S3 OutputEnc Option (已淘汰)	選用	無	SSE_S3、SSE_KMS、CSE_KMS

## KMS 金鑰

如果選擇 SSE\_KMS 或 CSE\_KMS 做為加密選項，則為 KMS 金鑰 ARN 或 ID。如需詳細資訊，請參閱[EncryptionConfiguration](#)亞 Amazon Athena API 參考中的。

參數名稱	別名	參數類型	預設值
KmsKey	S3 OutputEnc 金鑰 (已停用)	選用	無

## 結果擷取參數

### 結果擷取程式

將用於下載查詢結果的擷取程式。

預設的結果擷取程式 S3 會直接從 Amazon S3 下載查詢結果，而無需使用 Athena API。在大多數情況下，這是最快的選擇。如果您的查詢結果已使用 CSE\_KMS 加密，或允許使用者存取查詢結果的政策只允許使用 s3:CalledVia 的 Athena 呼叫，則無法使用此選項。

參數名稱	別名	參數類型	預設值	可能的值
ResultFetcher	無	選用	S3	中三, GetQueryResults, GetQueryResultsStream

### Note

在 JDBC 2.x 驅動程式中，此 UseResultSetStreaming = 1 設定會將驅動程式設定為使用結果集串流 API。在 JDBC 3.x 驅動程式中，對等的設定為。ResultFetcher=GetQueryResultsStream

## 擷取大小

這個參數的值用做內部緩衝區的最小值，以及擷取結果時的目標頁面大小。值 0 (零) 表示驅動程式應使用其預設值，如下所述。最大值為 1,000,000。

參數名稱	別名	參數類型	預設值
FetchSize	RowsToFetchPerBlock (已取代)	選用	0

- GetQueryResults 擷取程式一律會使用 1,000 的頁面大小，這是 API 呼叫所支援的最大值。當擷取大小大於 1,000 時，會進行多次連續的 API 呼叫，以填滿高於最小值的緩衝區。
- GetQueryResultsStream 擷取程式會使用設定的擷取大小做為頁面大小，或依預設使用 10,000。
- S3 擷取程式會使用設定的擷取大小做為頁面大小，或依預設使用 10,000。

## 查詢結果重複使用參數

### 啟用結果重複使用

指定執行查詢時是否可以重複使用同一查詢之前的結果。如需查詢結果重複使用的資訊，請參閱[ResultReuseByAgeConfiguration](#)。

參數名稱	別名	參數類型	預設值
EnableResultReuseByAge	無	選用	FALSE

### 結果重複使用最長期限

Athena 應考慮重複使用的之前查詢結果的最長期限 (以分鐘為單位)。如需結果重複使用年限上限的資訊，請參閱[ResultReuseByAgeConfiguration](#)。

參數名稱	別名	參數類型	預設值
MaxResultReuseAgeInMinutes	無	選用	60

## 查詢執行輪詢參數

### 最小查詢執行輪詢間隔

輪詢 Athena 的查詢執行狀態之前要等待的最短時間 (以毫秒為單位)。

參數名稱	別名	參數類型	預設值
MinQueryExecutionPollingIntervalMillis	MinQueryExecutionPollingInterval (已取代)	選用	100

### 最大查詢執行輪詢間隔

輪詢 Athena 的查詢執行狀態之前要等待的最長時間 (以毫秒為單位)。

參數名稱	別名	參數類型	預設值
MaxQueryExecutionPollingIntervalMillis	MaxQueryExecutionPollingInterval (已取代)	選用	5000

### 查詢執行輪詢間隔倍數

增加輪詢期間的因數。依預設，輪詢會以 `MinQueryExecutionPollingIntervalMillis` 的值開始，並且每次輪詢都會加倍，直至達到 `MaxQueryExecutionPollingIntervalMillis` 的值為止。

參數名稱	別名	參數類型	預設值
QueryExecutionPollingIntervalMultiplier	無	選用	2

### 端點覆寫參數

#### Athena 端點覆寫

驅動程式將用來對 Athena 進行 API 呼叫的端點。

請注意以下重點：

- 如果提供的 URL 中未指定 `https://` 或 `http://` 通訊協定，則驅動程式會插入 `https://` 字首。
- 若沒有指定此參數，則驅動程式會使用預設端點。

參數名稱	別名	參數類型	預設值
AthenaEndpoint	EndpointOverride (已取代)	選用	無

## Athena 串流服務端點覆寫

當驅動程式使用 Athena 串流服務時，將用來下載查詢結果的端點。Athena 串流服務可在連接埠 444 上使用。

請注意以下重點：

- 如果提供的 URL 中未指定 `https://` 或 `http://` 通訊協定，則驅動程式會插入 `https://` 字首。
- 如果未在提供的 URL 中指定連接埠，則驅動程式會插入串流服務連接埠 444。
- 如果未指定 `AthenaStreamingEndpoint` 參數，驅動程式會使用 `AthenaEndpoint` 覆寫。如果 `AthenaStreamingEndpoint` 或 `AthenaEndpoint` 覆寫均未指定，則驅動程式會使用預設的串流端點。

參數名稱	別名	參數類型	預設值
<code>AthenaStreamingEndpoint</code>	<code>StreamingEndpointOverride</code> (已取代)	選用	無

## LakeFormation 端點取代

使用 AWS Lake Formation [AssumeDecoratedRoleWithSAML API](#) 擷取暫時認證時，驅動程式將用於 Lake Formation 服務的端點。若沒有指定此參數，則驅動程式會使用預設的 Lake Formation 端點。

請注意以下重點：

- 如果提供的 URL 中未指定 `https://` 或 `http://` 通訊協定，則驅動程式會插入 `https://` 字首。

參數名稱	別名	參數類型	預設值
<code>LakeFormationEndpoint</code>	<code>LfEndpointOverride</code> (已取代)	選用	無

## S3 端點覆寫

當驅動程式使用 Amazon S3 擷取程式時，將用來下載查詢結果的端點。若沒有指定此參數，則驅動程式會使用預設的 Amazon S3 端點。

請注意以下重點：

- 如果提供的 URL 中未指定 `https://` 或 `http://` 通訊協定，則驅動程式會插入 `https://` 字首。

參數名稱	別名	參數類型	預設值
S3Endpoint	無	選用	無

## STS 端點覆寫

使用 AWS STS [AssumeRoleWithSAML](#) API 擷取暫時認證時，驅動程式將用於 AWS STS 服務的端點。如果未指定此參數，驅動程式會使用預設 AWS STS 端點。

請注意以下重點：

- 如果提供的 URL 中未指定 `https://` 或 `http://` 通訊協定，則驅動程式會插入 `https://` 字首。

參數名稱	別名	參數類型	預設值
StsEndpoint	StsEndpointOverride(已取代)	選用	無

## 代理組態參數

### 代理主機

代理主機的 URL。如果您需要 Athena 請求透過代理傳送，請使用此參數。

**Note**

確定在 ProxyHost 的 URL 開頭包含通訊協定 https:// 或 http://。

參數名稱	別名	參數類型	預設值
ProxyHost	無	選用	無

**代理連接埠**

要在代理主機上使用的連接埠。如果您需要 Athena 請求透過代理傳送，請使用此參數。

參數名稱	別名	參數類型	預設值
ProxyPort	無	選用	無

**代理使用者名稱**

在代理伺服器上進行身分驗證的使用者名稱。如果您需要 Athena 請求透過代理傳送，請使用此參數。

參數名稱	別名	參數類型	預設值
ProxyUsername	ProxyUID (已廢除)	選用	無

**代理密碼**

在代理伺服器上進行身分驗證的密碼。如果您需要 Athena 請求透過代理傳送，請使用此參數。

參數名稱	別名	參數類型	預設值
ProxyPassword	ProxyPWD (已廢除)	選用	無

## 免除代理的主機

啟用代理後 (也就是設定 ProxyHost 和 ProxyPort 連線參數後), 驅動程式在不使用代理的情況下連線到的一組主機名稱。主機應以縱線 (|) 字元分隔 (例如, host1.com|host2.com)。

參數名稱	別名	參數類型	預設值
ProxyExemptHosts	NonProxyHosts	選用	無

## 為身分提供者啟用代理

指定當驅動程式連線至身分提供者時, 是否應使用代理。

參數名稱	別名	參數類型	預設值
ProxyEnabledForIdP	UseProxyForIdP	選用	FALSE

## 記錄日誌參數

本節說明與記錄日誌相關的參數。

### 日誌層級

指定驅動程式記錄日誌的層級。除非還設定了 LogPath 參數, 否則不會記錄任何內容。

#### Note

除非您有特殊需求, 否則建議您僅設定 LogPath 參數。僅設定 LogPath 參數會啟用記錄日誌, 並使用預設的 TRACE 日誌層級。TRACE 日誌層級會提供最詳細的記錄。

參數名稱	別名	參數類型	預設值	可能的值
LogLevel	無	選用	TRACE	OFF、ERROR、WARN、INFO、DEBUG、TRACE

## 日誌路徑

執行驅動程式的電腦上儲存驅動程式日誌的目錄路徑。將在指定的目錄中建立具有唯一名稱的日誌檔。如果設定，則啟用驅動程式記錄日誌。

參數名稱	別名	參數類型	預設值
LogPath	無	選用	無

## 應用程式名稱

使用驅動程式的應用程式的名稱。如果指定此參數的值，則該值會包含在驅動程式對 Athena 進行之 API 呼叫的使用者代理字串中。

### Note

您也可以透過呼叫 `setApplicationName` 物件上的 `DataSource` 來設定應用程式名稱。

參數名稱	別名	參數類型	預設值
ApplicationName	無	選用	無

## 連線測試

如果設定為 `TRUE`，則即使未在連線上執行查詢，驅動程式也會在每次建立 JDBC 連線時進行連線測試。

參數名稱	別名	參數類型	預設值
ConnectionTest	無	選用	TRUE

**Note**

連線測試會將 SELECT 1 查詢提交至 Athena，以確認連線設定正確。這表示兩個檔案將儲存在 Amazon S3 中 (結果集和中繼資料)，而且可能會根據 [Amazon Athena 定價政策](#) 收取額外費用。

## 重試次數

驅動程式將可重試請求重新傳送至 Athena 的次數上限。

參數名稱	別名	參數類型	預設值
NumRetries	MaxErrorRetry (已取代)	選用	無

## 身分驗證連線參數

Athena JDBC 3.x 驅動程式支援多種身分驗證方法。所需的連線參數視您使用的身分驗證方法而定。

### 主題

- [IAM 憑證](#)
- [預設憑證](#)
- [AWS 組態設定檔憑證](#)
- [執行個體設定檔憑證](#)
- [自訂憑證](#)
- [JWT 憑證](#)
- [Azure AD 憑證](#)
- [Okta 憑證](#)
- [Ping 憑證](#)
- [AD FS 認證](#)
- [瀏覽器 Azure AD 憑證](#)
- [瀏覽器 SAML 憑證](#)

## IAM 憑證

您可以將 IAM 憑證與 JDBC 驅動程式配合使用，透過設定以下連線參數連線至 Amazon Athena。

### 使用者

您的 AWS 存取金鑰 ID。如需有關存取金鑰的資訊，請參閱《IAM 使用者指南》中的 [AWS 安全憑證](#)。

參數名稱	別名	參數類型	預設值
使用者	AccessKeyId	必要	無

### 密碼

您的 AWS 秘密金鑰 ID。如需有關存取金鑰的資訊，請參閱《IAM 使用者指南》中的 [AWS 安全憑證](#)。

參數名稱	別名	參數類型	預設值
密碼	SecretAccessKey	選用	無

### 工作階段字串

如果您使用臨時 AWS 認證，則必須指定工作階段 Token。如需有關暫時憑證的資訊，請參閱《IAM 使用者指南》中的 [IAM 中的暫時安全憑證](#)。

參數名稱	別名	參數類型	預設值
SessionToken	無	選用	無

### 預設憑證

您可以使用在用戶端系統上設定的預設憑證，透過設定以下連線參數連線至 Amazon Athena。如需有關使用預設憑證的資訊，請參閱《AWS SDK for Java 開發人員指南》中的 [使用預設憑證提供者鏈結](#)。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 DefaultChain。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProviderClass(已取代)	必要	無	DefaultChain

## AWS 組態設定檔憑證

您可以設定下列連線參數，使用儲存在 AWS 組態設定檔中的認證。AWS 組態設定檔通常儲存在~/ .aws目錄中的檔案中)。如需 AWS 組態設定檔的相關資訊，請參閱《AWS SDK for Java 開發人員指南》中的[使用設定檔](#)。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 ProfileCredentials。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProviderClass(已取代)	必要	無	ProfileCredentials

### 設定檔名稱

設定設 AWS 定檔的名稱，其認證應用於向 Athena 驗證要求。

參數名稱	別名	參數類型	預設值
ProfileName	無	必要	無

#### Note

設定檔名稱也可以指定為 CredentialsProviderArguments 參數的值，雖然此用法已廢除。

## 執行個體設定檔憑證

此身分驗證類型在 Amazon EC2 執行個體上使用。執行個體設定檔是連接至 Amazon EC2 執行個體的設定檔。使用執行個體設定檔登入資料提供者將 AWS 登入資料的管理委派給 Amazon EC2 執行個體中繼資料服務。如此一來，開發人員就無需在 Amazon EC2 執行個體上永久儲存憑證，也不用擔心輪換或管理臨時憑證。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 InstanceProfile。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProviderClass(已取代)	必要	無	InstanceProfile

### 自訂憑證

您可以使用此驗證類型，透過使用實作 [AwsCredentialsProvider](#) 介面的 Java 類別來提供您自己的認證。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設定為實作 [AwsCredentialsProvider](#) 介面之自訂類別的完整類別名稱。在執行期，該類別必須位於使用 JDBC 驅動程式之應用程式的 Java 類別路徑上。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProviderClass(已取代)	必要	無	AwsCredentialsProvider 自訂實作的完全合格類別名稱

### 憑證提供者引數

自訂認證提供者建構函數的逗號分隔字串引數清單。

參數名稱	別名	參數類型	預設值
CredentialsProviderArguments	AwsCredentialsProviderArguments (已取代)	選用	無

## JWT 憑證

藉由此身分驗證類型，您可以使用從外部身分提供者獲得的 JSON Web Token (JWT) 作為連線參數，向 Athena 進行身分驗證。外部認證提供者必須已與 AWS 聯合。

## 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 JWT。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AwsCredentialsProviderClass (已取代)	必要	無	JWT

## JWT web 身分權杖

從外部聯合身分提供者獲得的 JWT 權杖。此權杖將用於向 Athena 進行身分驗證。

參數名稱	別名	參數類型	預設值
JwtWebIdentityToken	web_identity_token (已廢除)	必要	無

## JWT 角色 ARN

要擔任之角色的 Amazon Resource Name (ARN)。有關假設角色的信息，請參閱 AWS Security Token Service API 參考 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
JwtRoleArn	role_arn (已廢除)	必要	無

### JWT 角色工作階段名稱

使用 JWT 憑證進行身分驗證時的工作階段名稱。該名稱可以是您選擇的任何名稱。

參數名稱	別名	參數類型	預設值
JwtRoleSessionName	role_session_name (已廢除)	必要	無

### Azure AD 憑證

SAML 型身分驗證機制，可讓您使用 Azure AD 身分提供者向 Athena 進行身分驗證。此方法假設已經在 Athena 和 Azure AD 之間建立了聯合。

#### Note

本節中的某些參數名稱具有別名。別名是參數名稱的功能對等項目，並已提供與 JDBC 2.x 驅動程式的向後相容性。由於參數名稱已經過改進，以遵循更清晰、更一致的命名慣例，因此我們建議您使用參數名稱，而不是已被取代的別名。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 AzureAD。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProviderClass (已取代)	必要	無	AzureAD

## 使用者

Azure AD 使用者的電子郵件地址，用來透過 Azure AD 進行身分驗證。

參數名稱	別名	參數類型	預設值
使用者	UID (已廢除)	必要	無

## 密碼

Azure AD 使用者的密碼。

參數名稱	別名	參數類型	預設值
密碼	PWD (已廢除)	必要	無

## Azure AD 租用戶 ID

Azure AD 應用程式的租用戶 ID。

參數名稱	別名	參數類型	預設值
AzureAdTenantId	tenant_id (已廢除)	必要	無

## Azure AD 用戶端 ID

Azure AD 應用程式的用戶端 ID。

參數名稱	別名	參數類型	預設值
AzureAdClientId	client_id (已廢除)	必要	無

## Azure AD 用戶端秘密

Azure AD 應用程式的用戶端秘密。

參數名稱	別名	參數類型	預設值
AzureAdClientSecret	client_secret (已廢除)	必要	無

### 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需 ARN 角色的相關資訊，請參閱 AWS Security Token Service API 參考資料 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
PreferredRole	preferred_role (已廢除)	選用	無

### 角色工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱 AWS Security Token Service API 參考 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
RoleSessionDuration	Duration (已廢除)	選用	3600

### Lake Formation 啟用

指定是否使用 S [AssumeDecoratedRoleWithAML](#) Lake Formation API 動作擷取臨時 IAM 登入資料，而非 S [AssumeRoleWithAML](#) AWS STS API 動作。

參數名稱	別名	參數類型	預設值
LakeFormationEnabled	無	選用	FALSE

## Okta 憑證

SAML 型身分驗證機制，可讓您使用 Okta 身分提供者向 Athena 進行身分驗證。此方法假設已經在 Athena 和 Okta 之間設定了聯合。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 Okta。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProviderClass (已取代)	必要	無	Okta

### 使用者

Okta 使用者的電子郵件地址，用來透過 Okta 進行身分驗證。

參數名稱	別名	參數類型	預設值
使用者	UID (已廢除)	必要	無

### 密碼

Okta 使用者的密碼。

參數名稱	別名	參數類型	預設值
密碼	PWD (已廢除)	必要	無

### Okta 主機名稱

您的 Okta 組織的 URL。您可以從 Okta 應用程式中的內嵌連結 URL 擷取 idp\_host 參數。如需這些步驟，請參閱 [從 Okta 擷取 ODBC 組態資訊](#)。https:// 後的第一個區段、直到並包括

okta.com，即為您的 IdP 主機 (例如，以 <https://trial-1234567.okta.com> 開頭的 URL 為 [trial-1234567.okta.com](https://trial-1234567.okta.com))。

參數名稱	別名	參數類型	預設值
OktaHostName	IdP_Host (已廢除)	必要	無

### Okta 應用程式 ID

您應用程式的兩部分識別符。您可以從 Okta 應用程式中的內嵌連結 URL 擷取應用程式 ID。如需這些步驟，請參閱 [從 Okta 擷取 ODBC 組態資訊](#)。應用程式 ID 是 URL 的最後兩個區段，包括中間的正斜線。這些區段是兩個由 20 個字元組成的字串，其中包含數字和大小寫字母的混合 (例如 [Abc1de2fghi3J45kL678/abc1defghij2klmNo3p4](#))。

參數名稱	別名	參數類型	預設值
OktaAppId	App_ID (已廢除)	必要	無

### Okta 應用程式名稱

Okta 應用程式的名稱。

參數名稱	別名	參數類型	預設值
OktaAppName	App_Name (已廢除)	必要	無

### Okta MFA 類型

如果您已將 Okta 設定為需要多重要素驗證 (MFA)，則需要根據您要使用的第二要素指定 Okta MFA 類型和其他參數。

Okta MFA 類型是用以透過 Okta 進行身分驗證的第二身分驗證要素類型 (在密碼之後)。支援的第二要素包括透過 Okta Verify 應用程式傳送的推送通知以及由 Okta Verify、Google Authenticator 生成的或透過簡訊發送的臨時一次性密碼 (TOTP)。個別組織安全政策會判斷使用者登入是否需要 MFA。

參數名稱	別名	參數類型	預設值	可能的值
OktaMfaType	okta_mfa_type (已廢除)	如果將 Okta 設定為需要 MFA，則為必要	無	oktaverifywithpush, oktaverifywithtotp, googleauthenticator, smsauthentication

### Okta 電話號碼

選擇 smsauthentication MFA 類型時，Okta 將使用簡訊將臨時一次性密碼傳送到的電話號碼。該電話號碼必須是美國或加拿大的電話號碼。

參數名稱	別名	參數類型	預設值
OktaPhoneNumber	okta_phone_number (已廢除)	必要，如果 OktaMfaType 是 smsauthentication	無

### Okta MFA 等待時間

在驅動程式擲出逾時例外狀況之前，等待使用者確認來自 Okta 的推送通知的持續時間 (以秒為單位)。

參數名稱	別名	參數類型	預設值
OktaMfaWaitTime	okta_mfa_wait_time (已廢除)	選用	60

### 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需 ARN 角色的相關資訊，請參閱 AWS Security Token Service API 參考資料 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
PreferredRole	preferred_role (已廢除)	選用	無

### 角色工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱 [AWS Security Token Service API 參考 AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
RoleSessionDuration	Duration (已廢除)	選用	3600

### Lake Formation 啟用

指定是否使用 S [AssumeDecoratedRoleWithAML](#) Lake Formation API 動作擷取臨時 IAM 登入資料，而非 S [AssumeRoleWithAML](#) AWS STS API 動作。

參數名稱	別名	參數類型	預設值
LakeFormationEnabled	無	選用	FALSE

### Ping 憑證

SAML 型身分驗證機制，可讓您使用 Ping Federate 身分提供者向 Athena 進行身分驗證。此方法假設已經在 Athena 和 Ping Federate 之間設定了聯合。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 Ping。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProv	必要	無	Ping

參數名稱	別名	參數類型	預設值	將使用的值
	IdPClass (已取代)			

## 使用者

Ping Federate 使用者的電子郵件地址，用於透過 Ping Federate 進行身分驗證。

參數名稱	別名	參數類型	預設值
使用者	UID (已廢除)	必要	無

## 密碼

Ping Federate 使用者的密碼。

參數名稱	別名	參數類型	預設值
密碼	PWD (已廢除)	必要	無

## PingHostName

Ping 伺服器的地址。若要尋找您的地址，請造訪下列 URL 並檢視 SSO 應用程式端點欄位。

```
https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections
```

參數名稱	別名	參數類型	預設值
PingHostName	IdP_Host (已廢除)	必要	無

## PingPortNumber

用於連線至 IdP 主機的連接埠號碼。

參數名稱	別名	參數類型	預設值
PingPortNumber	IdP_Port (已廢除)	必要	無

### PingPartnerSpId

服務提供者地址。若要尋找服務提供者地址，請造訪下列 URL 並檢視 SSO 應用程式端點欄位。

```
https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections
```

參數名稱	別名	參數類型	預設值
PingPartnerSpId	Partner_SPID (已廢除)	必要	無

### 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需 ARN 角色的相關資訊，請參閱 AWS Security Token Service API 參考資料 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
PreferredRole	preferred_role (已廢除)	選用	無

### 角色工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱 AWS Security Token Service API 參考 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
RoleSessionDuration	Duration (已廢除)	選用	3600

## Lake Formation 啟用

指定是否使用 S [AssumeDecoratedRoleWithAML](#) Lake Formation API 動作擷取臨時 IAM 登入資料，而非 S [AssumeRoleWithAML](#) AWS STS API 動作。

參數名稱	別名	參數類型	預設值
LakeFormationEnabled	無	選用	FALSE

## AD FS 認證

一種 SAML 型驗證機制，可使用 Microsoft 活動目錄同盟服務 (AD FS) 向 Athena 進行驗證。此方法假設使用者已設定 Athena 與 AD FS 之間的聯合。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 ADFS。

參數名稱	別名	參數類型	預設值	將使用的值
CredentialsProvider	AWSCredentialsProviderClass (已取代)	必要	無	ADFS

### 使用者

AD FS 使用者的電子郵件地址，用於驗證 AD FS。

參數名稱	別名	參數類型	預設值
使用者	UID (已廢除)	表單式驗證所需。視窗整合式驗證是選用的。	無

## 密碼

AD FS 使用者的密碼。

參數名稱	別名	參數類型	預設值
密碼	PWD (已廢除)	表單式驗證所需。視窗整合式驗證是選用的。	無

## ADFS 主機名稱

AD FS 伺服器的位址。

參數名稱	別名	參數類型	預設值
AdfsHostName	IdP_Host (已廢除)	必要	無

## ADFS 連接埠號碼

用來連線到 AD FS 伺服器的連接埠號碼。

參數名稱	別名	參數類型	預設值
AdfsPortNumber	IdP_Port (已廢除)	必要	無

## 依賴方

可信依賴方。使用此參數來覆寫 AD FS 依賴方端點 URL。

參數名稱	別名	參數類型	預設值
AdfsRelyingParty	LoginToRP (已淘汰)	選用	urn:amazon:webservices

## 已啟用 ADFS WIA

布林值。使用此參數可透過 AD FS 啟用視窗整合式驗證 (WIA)。

參數名稱	別名	參數類型	預設值
AdfsWiaEnabled	none	選用	FALSE

## 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需 ARN 角色的相關資訊，請參閱 AWS Security Token Service API 參考資料 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
PreferredRole	preferred_role (已廢除)	選用	無

## 角色工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱 AWS Security Token Service API 參考中的 [AssumeRole](#)。

參數名稱	別名	參數類型	預設值
RoleSessionDuration	Duration (已廢除)	選用	3600

## Lake Formation 啟用

指定是否使用 [AssumeDecoratedRoleWithSAML](#) Lake Formation API 動作擷取臨時 IAM 登入資料，而非 [AssumeRoleWithSAML](#) AWS STS API 動作。

參數名稱	別名	參數類型	預設值
LakeFormationEnabled	none	選用	FALSE

## 瀏覽器 Azure AD 憑證

瀏覽器 Azure AD 是 SAML 型身分驗證機制，可與 Azure AD 身分提供者搭配使用並支援多重因素認證。與標準 Azure AD 身分驗證不同，此機制不需要連線參數中的使用者名稱、密碼或用戶端密碼。如同標準 Azure AD 身分驗證機制，瀏覽器 Azure AD 也會假設使用者已經設定了 Athena 與 Azure AD 之間的聯合。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 `BrowserAzureAD`。

參數名稱	別名	參數類型	預設值	將使用的值
<code>CredentialsProvider</code>	<code>AWSCredentialsProviderClass</code> (已取代)	必要	無	<code>BrowserAzureAD</code>

### Azure AD 租用戶 ID

#### Azure AD 應用程式的租用戶 ID

參數名稱	別名	參數類型	預設值
<code>AzureAdTenantId</code>	<code>tenant_id</code> (已廢除)	必要	無

### Azure AD 用戶端 ID

#### Azure AD 應用程式的用戶端 ID

參數名稱	別名	參數類型	預設值
<code>AzureAdClientId</code>	<code>client_id</code> (已廢除)	必要	無

### 識別提供者回應逾時

驅動程式停止等待來自 Azure AD 之 SAML 回應之前的持續時間 (以秒為單位)。

參數名稱	別名	參數類型	預設值
IdpResponseTimeout	idp_response_timeout (已廢除)	選用	120

### 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需 ARN 角色的相關資訊，請參閱 AWS Security Token Service API 參考資料 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
PreferredRole	preferred_role (已廢除)	選用	無

### 角色工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱 AWS Security Token Service API 參考 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
RoleSessionDuration	Duration (已廢除)	選用	3600

### Lake Formation 啟用

指定是否使用 S [AssumeDecoratedRoleWithAML](#) Lake Formation API 動作擷取臨時 IAM 登入資料，而非 S [AssumeRoleWithAML](#) AWS STS API 動作。

參數名稱	別名	參數類型	預設值
LakeFormationEnabled	無	選用	FALSE

## 瀏覽器 SAML 憑證

瀏覽器 SAML 是一般身分驗證外掛程式，可與 SAML 型身分提供者搭配使用並支援多重因素認證。

### 憑證提供者

將用來驗證對 AWS 的請求的憑證提供者。將此參數的值設為 `BrowserSaml`。

參數名稱	別名	參數類型	預設值	將使用的值
<code>CredentialsProvider</code>	<code>AWSCredentialsProviderClass</code> (已取代)	必要	無	<code>BrowserSaml</code>

### 單一登入登入 URL

您的應用程式在 SAML 型身分提供者上的單一登入 URL。

參數名稱	別名	參數類型	預設值
<code>SsoLoginUrl</code>	<code>login_url</code> (已廢除)	必要	無

### 接聽連接埠

用來接聽 SAML 回應的連接埠號碼。此值應與您設定 SAML 型身分提供者所使用的 URL 相符 (例如, `http://localhost:7890/athena`)。

參數名稱	別名	參數類型	預設值
<code>ListenPort</code>	<code>listen_port</code> (已廢除)	選用	7890

### 識別提供者回應逾時

驅動程式停止等待來自 Azure AD 之 SAML 回應之前的持續時間 (以秒為單位)。

參數名稱	別名	參數類型	預設值
IdpResponseTimeout	idp_response_timeout (已廢除)	選用	120

### 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需 ARN 角色的相關資訊，請參閱 AWS Security Token Service API 參考資料 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
PreferredRole	preferred_role (已廢除)	選用	無

### 角色工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱 AWS Security Token Service API 參考 [AssumeRole](#) 中的。

參數名稱	別名	參數類型	預設值
RoleSessionDuration	Duration (已廢除)	選用	3600

### Lake Formation 啟用

指定是否使用 S [AssumeDecoratedRoleWithAML](#) Lake Formation API 動作擷取臨時 IAM 登入資料，而非 S [AssumeRoleWithAML](#) AWS STS API 動作。

參數名稱	別名	參數類型	預設值
LakeFormationEnabled	無	選用	FALSE

## 其他 JDBC 3.x 組態

下列各節說明 JDBC 3.x 驅動程式的其他組態設定。

### 網路逾時

驅動程式在向 Athena 進行 API 呼叫時等待回應的時間量，以毫秒為單位。在此時間之後，驅動程式會擲出逾時例外狀況。

網路逾時無法設定為連線參數。要進行設定，請在 JDBC Connection 物件上呼叫 `setNetworkTimeout` 方法。此值可以在 JDBC 連線的生命週期中變更。此參數的預設值為 `infinity`。

下列範例會將網路逾時設定為 5000 毫秒。

```
...
AthenaDriver driver = new AthenaDriver();
Connection connection = driver.connect(url, connectionParameters);
connection.setNetworkTimeout(null, 5000);
...
```

### 查詢逾時

提交查詢之後，驅動程式會在 Athena 上等待查詢完成的時間量，以秒為單位。在此時間之後，驅動程式會嘗試取消提交的查詢並擲出逾時例外狀況。

查詢逾時無法設定為連線參數。要進行設定，請在 JDBC Statement 物件上呼叫 `setQueryTimeout` 方法。這個值可以在 JDBC 陳述式的生命週期中變更。此參數的預設值為 0 (零)。值 0 表示查詢可以執行直到完成 (受限於 [Service Quotas](#))。

下列範例會將查詢逾時設定為 5 秒。

```
...
AthenaDriver driver = new AthenaDriver();
Connection connection = driver.connect(url, connectionParameters);
Statement statement = connection.createStatement();
statement.setQueryTimeout(5);
...
```

## Amazon Athena JDBC 3.x 版本資訊

這些版本說明提供 Amazon Athena JDBC 3.x 驅動程式中改進和修正的詳細資訊。

### 3.2.0

二零二四年四月二十六日

#### 改善項目

- 目錄作業效能 — 已改善不使用萬用字元之目錄作業的效能。
- 最小輪詢間隔變更 — 已修改最小輪詢間隔預設值，以減少驅動程式對 Athena 進行的 API 呼叫次數。仍會儘快偵測到查詢完成。
- BI 工具可發現性 — 驅動程序已經變得更容易發現商業智能工具。
- 資料類型對應 — 已改善與 Athena 的資料類型對應 `binaryarray`，以及 `struct DDL` 資料類型。
- AWS SDK 版本 — 驅動程式中使用的 AWS SDK 版本已更新為 2.25.34。

#### 修正項目

- 同盟目錄表格清單 — 修正造成聯合目錄傳回空白資料表清單的問題。
- `getSchema` — 修正導致 JDBC [DatabaseMetaData#getSchemas](#) 方法僅從預設目錄而非從所有目錄擷取資料庫的問題。(
- `GetColumns` — 修正當以空目錄名稱呼叫 JDBC [DatabaseMetaData#getColumns](#) 方法時，造成傳回空目錄的問題。

### 3.1.0

二零二四年二月二十五日

#### 改善項目

- 增加了對 Microsoft 活動目錄聯合服務 ( AD FS ) 的 Support Windows 集成身份驗證和基於表單的身份驗證。
- 為了與版本 2.x 的向後兼容性，`awsathenaJDBC` 子協議現在被接受，但會產生棄用警告。請改用 `athena JDBC` 子協議。
- `AwsDataCatalog` 現在是目錄參數的預設值，並且 `default` 是資料庫參數的預設值。這些變更可確保傳回目前目錄和資料庫的正確值，而非 `null`。
- 在與 JDBC 規範一致，`IS_GENERATEDCOLUMN` 現在返回一個空字符串，`IS_AUTOINCREMENT` 而不是 `NO`。
- Athena `int` 資料類型現在會對應至與 Athena 相同的 JDBC 類型，`integer` 而不是對應至 `other`。

- 當 Athena 的欄中繼資料不包含選擇性 precision 和 scale 欄位時，驅動程式現在會針對 ResultSet 欄中對應的值傳回零。
- AWS 開發套件版本已經更新為 2.21.39 版。

## 修正項目

- 修正當 Athena GetQueryResultsStream 的純文字結果的欄數與 Athena 結果中繼資料中的欄計數不一致時，會發生例外狀況的問題。(

## 3.0.0

### 二零二三至一六年發表

Athena JDBC 3.x 驅動程式是新一代驅動程式，可提供更好的效能和相容性。JDBC 3.x 驅動程式支援直接從 Amazon S3 讀取查詢結果，可改善使用大量查詢結果的應用程式效能。新的驅動程式也有較少的協力廠商相依性，這使得與 BI 工具和自訂應用程式的整合更容易。

### 先前版本的 Athena JDBC 3.x 驅動程序

我們強烈建議您使用 [最新版本](#) 的 JDBC 3.x 驅動程式。最新版本的驅動程式包含最新的改進和修正。只有在您的應用程式與最新版本不相容時，才使用舊版本。

### JDBC 驅動程式 uber jar

下面的下載將驅動程式及其所有相依性包在同一個 .jar 檔案中。此下載通常用於第三方 SQL 用戶端。

- [3.1.0 尤伯杯罐](#) 尤伯杯罐
- [3.0.0 尤伯杯罐](#) 尤伯杯罐

### JDBC 驅動程式 lean jar

下面的下載是一個 .zip 檔案，其中包含驅動程式的 lean .jar 和驅動程式相依性的單獨 .jar 檔案。此下載通常用於可能具有相依性與驅動程式使用的相依性衝突的自訂應用程式。如果您想選擇要包含在 lean jar 中的驅動程式相依性，以及要排除的驅動程式相依性 (如果您的自訂應用程式已包含一或多個驅動程式相依性)，則此下載非常有用。

- [3.1.0 瘦肉罐子](#) 瘦肉罐

- [3.0.0 瘦肉罐子 精益罐](#)

## Athena JDBC 2.x 驅動程式

您可以使用 JDBC 連接來將 Athena 連接到商業智慧工具和其他應用程式，例如 [SQL Workbench](#)。若要這樣做，請使用此頁面上的 Amazon S3 連結，以下載、安裝和設定 Athena JDBC 2.x 驅動程式。有關建置 JDBC 連接 URL 的資訊，請參閱可下載的[JDBC 驅動程式安裝和組態指南](#)。如需許可的相關資訊，請參閱[透過 JDBC 和 ODBC 連接存取](#)。要提交有關 JDBC 驅動程式的意見回饋，請傳送電子郵件至 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。從 2.0.24 版開始，有兩個版本的驅動程式可供使用：一個包含 AWS SDK，另一個則沒有。

### Important

當您使用 JDBC 驅動程式時，務必注意以下要求：

- 打開連接埠 444 – 將 Athena 用來串流查詢結果的連接埠 444 保持開放給輸出流量。當您使用 PrivateLink 端點連線至 Athena 時，請確定連接到 PrivateLink 端點的安全群組對連接埠 444 的輸入流量開放。如果連接埠 444 遭到封鎖，您可能會收到錯誤訊息 [Simba][AthenaJDBC](100123) An error has occurred. ([Simba][AthenaJDBC](100123) 發生錯誤。) 資料欄初始化期間發生例外狀況。
- 雅典娜/ GetQueryResultsStream 政策 — 將athena:GetQueryResultsStream政策動作新增至使用 JDBC 驅動程式的 IAM 主體。此政策動作不會直接透過 API 公開。它僅與 ODBC 和 JDBC 驅動程式搭配使用，做為串流結果支援的一部分。如需政策範例，請參閱 [AWS 受管理的策略：AWSQuicksightAthenaAccess](#)。
- 使用用於多個資料目錄的 JDBC 驅動程式 – 若要將用於多個資料目錄的 JDBC 驅動程式與 Athena 搭配使用 (例如，在使用[外部 Hive 中繼存放區](#)或[聯合查詢](#)時)，在您的 JDBC 連線字串中包含 MetadataRetrievalMethod=ProxyAPI。
- 4.1 驅動程式 – 從 2023 年起，將停止對 JDBC 4.1 版提供驅動程式支援。將不再發布進一步更新。如果您仍在使用 JDBC 4.1 驅動程式，強烈建議您改用 4.2 驅動程式。

## 使用 SDK 的 JDBC 2.x 驅動程式 AWS

JDBC 驅動程式 2.1.5 版本符合 JDBC API 4.2 資料標準，且需要 JDK 8.0 或更新版本。如需檢查您使用之 Java 執行階段環境 (JRE) 版本的相關資訊，請參閱 Java [文件](#)。

使用下列連結來下載 JDBC 4.2 驅動程式 .jar 檔案。

- <https://downloads.athena.us-east-1.amazonaws.com/drivers/JDBC/SimbaAthenaJDBC-2.1.5.1000/AthenaJDBC42-2.1.5.1000.jar>

下列 .zip 檔案下載包含 JDBC 4.2 的 .jar 檔案，並包含 AWS SDK 和隨附的文件、版本說明、授權和合約。

- [SimbaAthena](#)

不含 AWS SDK 的 JDBC 2.x 驅動程式

JDBC 驅動程式 2.1.5 版本符合 JDBC API 4.2 資料標準，且需要 JDK 8.0 或更新版本。如需檢查您使用之 Java 執行階段環境 (JRE) 版本的相關資訊，請參閱 Java [文件](#)。

使用以下鏈接下載不使用 AWS SDK 的 JDBC 4.2 驅動程序 .jar 文件。

- <https://downloads.athena.us-east-1.amazonaws.com/drivers/JDBC/SimbaAthenaJDBC-2.1.5.1001/AthenaJDBC42-2.1.5.1001.jar>

以下 .zip 檔案下載包含 JDBC 4.2 .jar 檔案及隨附的文件、版本備註、授權和協議。它不包括 AWS SDK。

- [SimbaAthena](#)

JDBC 2.x 驅動程式版本備註、授權合約和聲明

下載您需要的版本後，請閱讀版本備註，並檢閱授權合約和聲明。

- [版本備註](#)
- [授權合約](#)
- [通知](#)
- [第三方授權](#)

JDBC 2.x 驅動程式文件

下載驅動程式的下列文件：

- [JDBC 驅動程式安裝和組態指南](#)。使用此指南來安裝和設定驅動程式。

- [JDBC 驅動程式遷移指南](#)。使用此指南從舊版遷移到目前的版本。

## 使用 ODBC 連接到 Amazon Athena

Amazon Athena 提供兩個 ODBC 驅動程序：版本 1.x 和 2.x。Athena ODBC 2.x 驅動程序是一個新的替代方案，支持 Linux，macOS 果系統的手臂，macOS 果系統英特爾和視窗 64 位系統。Athena 2.x 驅動程式支援 1.x ODBC 驅動程式支援的所有身分驗證外掛程式，而且幾乎所有連線參數均可回溯相容。

- 若要下載 ODBC 2.x 驅動程式，請參閱 [Amazon Athena ODBC 2.x](#)。
- 若要下載 ODBC 1.x 驅動程式，請參閱 [Athena ODBC 1.x 驅動程式](#)。

### 主題

- [Amazon Athena ODBC 2.x](#)
- [Athena ODBC 1.x 驅動程式](#)
- [使用 Amazon Athena Power BI 連接器](#)

## Amazon Athena ODBC 2.x

您可以從許多第三方 SQL 用戶端工具和應用程式使用 ODBC 連線來連接至 Amazon Athena。在您的用戶端電腦上設定 ODBC 連線。

### 考量與限制

- 如需有關從 Athena ODBC 1.x 驅動程式遷移至 Athena 2.x ODBC 驅動程式的資訊，請參閱 [遷移至 ODBC 2.x 驅動程式](#)。
- 搭配使用 [S3 擷取程式](#)與CSE\_KMS[加密選項](#)時，Amazon S3 用戶端無法解密存放於 Amazon S3 儲存貯體中的結果。解決方法是使用 [Athena 串流 API](#) 選項來擷取結果集。

### ODBC 2.x 驅動程式下載

要下載 Amazon Athena 2.x ODBC 驅動程序，請訪問此頁面上的鏈接。

#### Important

當您使用 ODBC 2.x 驅動程式時，務必注意以下要求：

- 打開連接埠 444 – 將 Athena 用來串流查詢結果的連接埠 444 保持開放給輸出流量。當您使用 PrivateLink 端點連線至 Athena 時，請確定連接到 PrivateLink 端點的安全群組對連接埠 444 的輸入流量開放。
- 雅典娜/ GetQueryResultsStream 政策 — 將athena:GetQueryResultsStream政策動作新增至使用 ODBC 驅動程式的 IAM 主體。此政策動作不會直接透過 API 公開。它僅與 ODBC 和 JDBC 驅動程式搭配使用，做為串流結果支援的一部分。如需政策範例，請參閱 [AWS 受管理的策略：AWSQuicksightAthenaAccess](#)。

## Linux

驅動程式版本	下載連結
適用於 64 位元的 ODBC 2.0.3.0	<a href="#">64 位元硬碟驅動式</a>

## macOS (臂)

驅動程式版本	下載連結
適用於 macOS 64 位元 (ARM)	<a href="#">macOS 64 位驅動程序 2.0.3.0 ( 臂 )</a> 2.0.3.0 ( 臂 )

## macOS 特爾

驅動程式版本	下載連結
適用於 macOS 64 位 (英特爾)	<a href="#">macOS 64 位 ODBC 驅動程序 2.0.3.0 ( 英特爾 )</a> 2.0.3.0 ( 英特爾 )

## Windows

驅動程式版本	下載連結
視窗 64 位元版本 2.0.3.0	<a href="#">視窗 64 位元驅動程式 2.0.3.0 驅動程式</a>

### 主題

- [ODBC 2.x 驅動程式入門](#)
- [Athena ODBC 2.x 連線參數](#)
- [遷移至 ODBC 2.x 驅動程式](#)
- [疑難排解 ODBC 2.x 驅動程式](#)
- [Amazon Athena ODBC 2.x 版本備註](#)

### ODBC 2.x 驅動程式入門

使用本節中的資訊，開始使用 Amazon Athena ODBC 2.x 驅動程式。視窗、Linux 和 macOS 作業系統支援此驅動程式。

### 主題

- [Windows](#)
- [Linux](#)
- [macOS](#)

## Windows

如果您想使用 Windows 客戶端計算機訪問 Amazon Athena，則需要 Amazon Athena ODBC 驅動程式。

### Windows 系統要求

在將直接存取 Amazon Athena 資料庫的用戶端電腦上安裝 Amazon Athena ODBC 驅動程式，而不是使用 Web 瀏覽器。

您使用的 Windows 系統必須符合下述要求：

- 您擁有管理員權限

- 下列其中一個操作系統：
  - Windows 11、10 或 8.1
  - Windows Server 2019、2016 或 2012
- 至少 100 MB 的可用磁碟空間
- 已安裝 [Microsoft Visual C++ Redistributable for Visual Studio](#)，適用於 Windows 64 位元。

## 安裝 Amazon Athena ODBC 驅動程式

若要下載並安裝適用於 Windows 的 Amazon Athena ODBC 驅動程式

1. [下載](#) AmazonAthenaODBC-2.x.x.x.msi 安裝檔案。
2. 啟動安裝檔案，然後選擇下一步。
3. 若要接受授權協議條款，請選取核取方塊，然後選擇下一步。
4. 若要變更安裝位置，請選擇瀏覽，瀏覽至所需的資料夾，然後選擇確定。
5. 若要接受安裝位置，請選擇下一步。
6. 選擇 Install (安裝)。
7. 完成安裝時，請選擇完成。

## 設定驅動程式組態選項的方法

若要控制 Windows 中 Amazon Athena ODBC 驅動程式的行為，您可以使用下列方式設定驅動程式組態選項：

- 在 ODBC 資料來源管理員中設定資料來源名稱 (DSN) 時。
- 透過在下列位置新增或變更 Windows 登錄機碼：

```
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\YOUR_DSN_NAME
```

- 以程式設計方式連線時，在連線字串中設定驅動程式選項。

## 在 Windows 上設定資料來源名稱

在下載並安裝 ODBC 驅動程式之後，您必須將資料來源名稱 (DSN) 項目新增至用戶端電腦或 Amazon EC2 執行個體。SQL 用戶端工具使用此資料來源以連接至並查詢 Amazon Athena。

## 建立系統 DSN 項目

1. 在 Windows 的開始選單中，以滑鼠右鍵按一下 ODBC 資料來源 (64 位元)，然後選擇更多，以管理員身分執行。
2. 在 ODBC 資料來源管理員中，選擇驅動程式索引標籤。
3. 在名稱資料欄中，確認是否存在 Amazon Athena ODBC (x64)。
4. 執行以下任意一項：
  - 若要為電腦上的所有使用者設定驅動程式，請選擇系統 DSN 索引標籤。由於使用不同帳戶載入資料的應用程式可能無法偵測來自其他帳戶的 DSN，因此建議您使用系統 DSN 組態選項。

### Note

使用系統 DSN 選項需要管理權限。

- 若要僅為您的使用者帳戶設定驅動程式，請選擇使用者 DSN 索引標籤。
5. 選擇新增。建立新資料來源對話方塊隨即開啟。
  6. 選擇 Amazon Athena ODBC (x64)，然後選擇完成。
  7. 請在 Amazon Athena ODBC 組態對話方塊中輸入下列資訊。如需這些選項的詳細資訊，請參閱 [主要 ODBC 2.x 連線參數](#)。
    - 對於資料來源名稱，輸入您要用來識別資料來源的名稱。
    - 對於描述，輸入描述，以協助您識別資料來源。
    - 對於區域，輸入您將在 Athena 中使用的 AWS 區域 的名稱 (例如 **us-west-1**)。
    - 對於目錄，輸入 Amazon Athena 目錄的名稱。預設值為 `AwsDataCatalog`，由使用 AWS Glue。
    - 對於資料庫，輸入 Amazon Athena 資料庫的名稱。預設值為預設。
    - 對於工作群組，輸入 Amazon Athena 工作群組的名稱。預設值為主要。
    - 對於 S3 輸出位置，請在 Amazon S3 中輸入將存放查詢結果的位置 (例如 **s3://DOC-EXAMPLE-BUCKET/**)。
    - (選用) 對於加密選項，請選擇加密選項。預設值為 `NOT_SET`。
    - (選用) 對於 KMS 金鑰，請視需要選擇加密 KMS 金鑰。
  8. 若要指定 IAM 身分驗證的組態選項，請選擇身分驗證選項。
  9. 輸入下列資訊：

- 對於身分驗證類型，選擇 IAM 憑證。此為預設值。如需有關可用身分驗證類型的詳細資訊，請參閱 [身分驗證選項](#)。
  - 對於使用者名稱，輸入使用者名稱。
  - 對於密碼，輸入密碼。
  - 對於工作階段權杖，如果您要使用臨時 AWS 認證，請輸入工作階段權杖。如需臨時登入資料的相關資訊，請參閱 IAM [使用指南中的將臨時登入資料搭配 AWS 資源使用](#)
10. 選擇確定。
  11. 請在 Amazon Athena ODBC 組態對話方塊底部，選擇測試。如果用戶端電腦成功連線至 Amazon Athena，連線測試方塊會報告連線成功。如果沒有，方塊會報告連線失敗，並顯示對應的錯誤資訊。
  12. 選擇確定以關閉連線測試。您建立的資料來源現在會顯示在資料來源清單中。

## 在 Windows 上使用無 DSN 連線

您可以使用無 DSN 連線來連接至沒有資料來源名稱 (DSN) 的資料庫。下列範例顯示連線至 Amazon Athena 的 Amazon Athena ODBC (x64) ODBC 驅動程式的連線字串。

```
DRIVER={Amazon Athena ODBC (x64)};Catalog=AwsDataCatalog;AwsRegion=us-west-1;Schema=test_schema;S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/;AuthenticationType=IAM Credentials;UID=YOUR_UID;PWD=YOUR_PWD;
```

## Linux

如果您想使用 Linux 客戶端計算機訪問 Amazon Athena，則需要 Amazon Athena ODBC 驅動程序。

### 系統需求

安裝驅動程式的每部 Linux 用戶端電腦都必須符合下列需求。

- 您有根存取權限。
- 請使用下列其中一個 Linux 發行版：
  - 紅帽企業版 (RHEL) 7 或 8
  - CentOS 7 或 8。
- 有 100 MB 的可用磁碟空間。
- 使用版本 2.3.1 或 [uni](#) xODBC。

- 使用版本 2.26 或更高版本的 [GNU C 函式庫 \(glibc\)](#)。

在 Linux 上安裝 ODBC 資料連接器

使用下列程序在 Linux 作業系統上安裝 Amazon Athena ODBC 驅動程式。

在 Linux 上安裝 Amazon Athena ODBC 驅動程序

1. 輸入下列其中一個命令：

```
sudo rpm -Uvh AmazonAthenaODBC-2.X.Y.Z.rpm
```

或

```
sudo yum --nogpgcheck localinstall AmazonAthenaODBC-2.X.Y.Z.rpm
```

2. 安裝完成後，輸入下列其中一個指令，以確認驅動程式是否已安裝：

- ```
yum list | grep amazon-athena-odbc-driver
```

輸出：

```
amazon-athena-odbc-driver.x86_64 2.0.2.1-1.amzn2int installed
```

- ```
rpm -qa | grep amazon
```

輸出：

```
amazon-athena-odbc-driver-2.0.2.1-1.amzn2int.x86_64
```

在 Linux 上配置資料來源名稱

安裝驅動程式之後，您可以在下列位置找到範例 `.odbc.ini` 和 `.odbcinst.ini` 檔案：

- `/opt/athena/odbc/ini/`。

使用此位置中的 `.ini` 檔案做為設定 Amazon Athena ODBC 驅動程式和資料來源名稱 (DSN) 的範例。

**Note**

默認情況下，ODBC 驅動程序管理器使用隱藏的配置文件 `.odbc.ini.odbcinst.ini`，它們位於主目錄中。

若要使用 UnixODBC 指定 `.odbc.ini` 和 `.odbcinst.ini` 檔案的路徑，請執行以下步驟。

**使用 unixODBC ODBC 指定 ODBC `.ini` 檔案位置的步驟**

1. 設定 ODBCINI 為檔案的完整路徑和 `odbc.ini` 檔案名稱，如下列範例所示。

```
export ODBCINI=/opt/athena/odbc/ini/odbc.ini
```

2. 設定 ODBCSYSINI 為包含 `odbcinst.ini` 檔案之目錄的完整路徑，如下列範例所示。

```
export ODBCSYSINI=/opt/athena/odbc/ini
```

3. 輸入以下命令以確認您使用的是 UnixODBC 驅動程式管理員和正確的檔案：`odbc*.ini`

```
username % odbcinst -j
```

**範例輸出**

```
unixODBC 2.3.1
DRIVERS.....: /opt/athena/odbc/ini/odbcinst.ini
SYSTEM DATA SOURCES: /opt/athena/odbc/ini/odbc.ini
FILE DATA SOURCES..: /opt/athena/odbc/ini/ODBCDataSources
USER DATA SOURCES..: /opt/athena/odbc/ini/odbc.ini
SQLULEN Size.....: 8
SQLLEN Size.....: 8
SQLSETPOSIR0W Size.: 8
```

4. 如果要使用資料來源名稱 (DSN) 連接至資料倉庫，請規劃 `odbc.ini` 檔案以定義資料來源名稱 (DSN)。設定 `odbc.ini` 檔案中的性質以建立 DSN，以指定資料倉庫的連接資訊，如以下範例所示。

```
[ODBC Data Sources]
athena_odbc_test=Amazon Athena ODBC (x64)
```

```
[ATHENA_WIDE_SETTINGS] # Special DSN-name to signal driver about logging
configuration.
LogLevel=0             # To enable ODBC driver logs, set this to 1.
UseAwsLogger=0        # To enable AWS-SDK logs, set this to 1.
LogPath=/opt/athena/odbc/logs/ # Path to store the log files. Permissions to the
location are required.

[athena_odbc_test]
Driver=/opt/athena/odbc/lib/libathena-odbc.so
AwsRegion=us-west-1
Workgroup=primary
Catalog=AwsDataCatalog
Schema=default
AuthenticationType=IAM Credentials
UID=
PWD=
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/
```

5. 設定odbcinst.ini檔案，如下列範例所示。

```
[ODBC Drivers]
Amazon Athena ODBC (x64)=Installed

[Amazon Athena ODBC (x64)]
Driver=/opt/athena/odbc/lib/libathena-odbc.so
Setup=/opt/athena/odbc/lib/libathena-odbc.so
```

6. 安裝並設定 Amazon Athena ODBC 驅動程式之後，請使用 UnixODBC isql 命令列工具來驗證連線，如下列範例所示。

```
username % isql -v "athena_odbc_test"
+-----+
| Connected! |
|           |
| sql-statement |
| help [tablename] |
| quit |
|           |
+-----+
SQL>
```

## macOS

如果您想使用 macOS 客戶端計算機訪問 Amazon Athena，則需要 Amazon Athena ODBC 驅動程序。

### macOS 系統需求

安裝驅動程式的每部 macOS 電腦都必須符合下列需求。

- 使用 macOS 版本 14 或更新版本。
- 有 100 MB 的可用磁碟空間。
- [使用版本 3.52.16 或更高版本的 iODBC。](#)

### 在 macOS 上安裝 ODBC 資料連接器

使用下列程序來下載並安裝適用於 macOS 作業系統的 Amazon Athena ODBC 驅動程式。

#### 下載並安裝適用於 macOS 的 Amazon Athena ODBC 驅動程序

1. 下載 .pkg 套件檔案。
2. 按兩下 .pkg 檔案。
3. 依照精靈中的步驟安裝驅動程式。
4. 在 [授權合約] 頁面上，按 [繼續]，然後選擇 [同意]。
5. 選擇 Install (安裝)。
6. 完成安裝時，請選擇完成。
7. 輸入以下命令以確認驅動程式是否已安裝：

```
> pkgutil --pkgs | grep athenaodbc
```

根據您的系統而定，輸出可能如下所示之一。

```
com.amazon.athenaodbc-x86_64.Config  
com.amazon.athenaodbc-x86_64.Driver
```

或

```
com.amazon.athenaodbc-arm64.Config
```

```
com.amazon.athenaodbc-arm64.Driver
```

在 macOS 上設定資料來源名稱

安裝驅動程式之後，您可以在下列位置找到範例 `.odbc.ini` 和 `.odbcinst.ini` 檔案：

- 英特爾處理器電腦： `/opt/athena/odbc/x86_64/ini/`
- ARM 處理器電腦： `/opt/athena/odbc/arm64/ini/`

使用此位置中的 `.ini` 檔案做為設定 Amazon Athena ODBC 驅動程式和資料來源名稱 (DSN) 的範例。

#### Note

默認情況下，ODBC 驅動程序管理器使用隱藏的配置文件 `.odbc.ini` 和 `.odbcinst.ini`，它們位於主目錄中。

若要使用 iODBC 驅動程式管理員指定 `.odbc.ini` 和 `.odbcinst.ini` 檔案的路徑，請執行下列步驟。

使用 iODBC 驅動程式管理員指定 ODBC `.ini` 檔案位置的步驟

1. 將 `ODBCINI` 設定為 `odbc.ini` 檔案的完整路徑和檔案名稱。

- 對於具有 Intel 處理器的 macOS 電腦，請使用下列語法。

```
export ODBCINI=/opt/athena/odbc/x86_64/ini/odbc.ini
```

- 對於具有 ARM 處理器的 macOS 電腦，請使用下列語法。

```
export ODBCINI=/opt/athena/odbc/arm64/ini/odbc.ini
```

2. 將 `ODBCSYSINI` 設定為 `odbcinst.ini` 檔案的完整路徑和檔案名稱。

- 對於具有 Intel 處理器的 macOS 電腦，請使用下列語法。

```
export ODBCSYSINI=/opt/athena/odbc/x86_64/ini/odbcinst.ini
```

- 對於具有 ARM 處理器的 macOS 電腦，請使用下列語法。

```
export ODBCSYSINI=/opt/athena/odbc/arm64/ini/odbcinst.ini
```

3. 如果要使用資料來源名稱 (DSN) 連接至資料倉庫，請規劃odbc.ini檔案以定義資料來源名稱 (DSN)。設定odbc.ini檔案中的性質以建立 DSN，以指定資料倉庫的連接資訊，如以下範例所示。

```
[ODBC Data Sources]
athena_odbc_test=Amazon Athena ODBC (x64)

[ATHENA_WIDE_SETTINGS] # Special DSN-name to signal driver about logging
configuration.
LogLevel=0             # set to 1 to enable ODBC driver logs
UseAwsLogger=0        # set to 1 to enable AWS-SDK logs
LogPath=/opt/athena/odbc/logs/ # Path to store the log files. Permissions to the
location are required.

[athena_odbc_test]
Description=Amazon Athena ODBC (x64)
# For ARM:
Driver=/opt/athena/odbc/arm64/lib/libathena-odbc-arm64.dylib
# For Intel:
# Driver=/opt/athena/odbc/x86_64/lib/libathena-odbc-x86_64.dylib
AwsRegion=us-west-1
Workgroup=primary
Catalog=AwsDataCatalog
Schema=default
AuthenticationType=IAM Credentials
UID=
PWD=
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/
```

4. 設定odbcinst.ini檔案，如下列範例所示。

```
[ODBC Drivers]
Amazon Athena ODBC (x64)=Installed

[Amazon Athena ODBC (x64)]
# For ARM:
Driver=/opt/athena/odbc/arm64/lib/libathena-odbc-arm64.dylib
Setup=/opt/athena/odbc/arm64/lib/libathena-odbc-arm64.dylib
# For Intel:
# Driver=/opt/athena/odbc/x86_64/lib/libathena-odbc-x86_64.dylib
```

```
# Setup=/opt/athena/odbc/x86_64/lib/libathena-odbc-x86_64.dylib
```

5. 安裝並設定 Amazon Athena ODBC 驅動程式之後，請使用 `iodbctest` 命令列工具驗證連線，如下列範例所示。

```
username@ % iodbctest
iODBC Demonstration program
This program shows an interactive SQL processor
Driver Manager: 03.52.1623.0502

Enter ODBC connect string (? shows list): ?

DSN                                | Driver
-----|-----
athena_odbc_test                   | Amazon Athena ODBC (x64)

Enter ODBC connect string (? shows list): DSN=athena_odbc_test;
Driver: 2.0.2.1 (Amazon Athena ODBC Driver)

SQL>
```

## Athena ODBC 2.x 連線參數

Amazon Athena ODBC 組態對話訪客選項包括身分驗證選項、進階選項、記錄選項、端點覆寫和代理選項。如需有關各項內容的詳細資訊，請造訪相應的連結。

- [主要 ODBC 2.x 連線參數](#)
- [身分驗證選項](#)
- [進階選項](#)
- [記錄選項](#)
- [端點覆寫](#)
- [代理選項](#)

## 主要 ODBC 2.x 連線參數

下列各節會描述每個主要連線參數。

### Data Source Name (資料來源名稱)

指定資料來源的名稱。

連線字串名稱	參數類型	預設值	連線字串範例
DSN	無 DSN 連線類型的選用項目	none	DSN=AmazonAthena0dbcUsWest1;

## 描述

包含資料來源的描述。

連線字串名稱	參數類型	預設值	連線字串範例
描述	選用	none	Description=Connection to Amazon Athena us-west-1;

## 目錄

指定資料目錄名稱。如需有關目錄的詳細資訊，請參閱 Amazon Athena API 參考 [DataCatalog](#) 中的。

連線字串名稱	參數類型	預設值	連線字串範例
目錄	選用	AwsDataCatalog	Catalog=AwsDataCatalog;

## 區域

指定 AWS 區域。如需相關資訊 AWS 區域，請參閱 [區域和可用區域](#)。

連線字串名稱	參數類型	預設值	連線字串範例
AwsRegion	強制性	none	AwsRegion =us-west-1;

## 資料庫

指定資料庫名稱。如需有關資料庫的詳細資訊，請參閱《Amazon Athena API 參考》中的[資料庫](#)。

連線字串名稱	參數類型	預設值	連線字串範例
結構描述	選用	default	Schema=default;

## 工作群組

指定工作群組名稱。如需有關工作群組的詳細資訊，請參閱 Amazon Athena API 參考[WorkGroup](#)中的。

連線字串名稱	參數類型	預設值	連線字串範例
工作群組	選用	primary	Workgroup =primary;

## 輸出位置

指定 Amazon S3 中儲存查詢結果的位置。如需有關輸出位置的詳細資訊，請參閱 Amazon Athena API 參考[ResultConfiguration](#)中的。

連線字串名稱	參數類型	預設值	連線字串範例
S3 OutputLocation	強制性	none	S3outputL ocation=s3:// DOC-EXAMPLE-BU CKET/;

## 加密選項

對話方塊參數名稱：加密選項

指定加密選項。如需有關加密選項的詳細資訊，請參閱 Amazon Athena API 參考[EncryptionConfiguration](#)中的。

連線字串名稱	參數類型	預設值	可能的值	連線字串範例
S3 OutputEnc Option	選用	none	NOT_SET, SSE_S3, SSE_KMS, CSE_KMS	S3OutputEncOption=SSE_S3;

## KMS 金鑰

指定要加密的 KMS 金鑰。如需 KMS 金鑰加密組態的詳細資訊，請參閱 Amazon Athena API 參考 [EncryptionConfiguration](#) 中的。

連線字串名稱	參數類型	預設值	連線字串範例
S3 OutputEnc 肯姆斯基	選用	none	S3OutputEncKMSKey=your_key;

## 連線測試

ODBC 資料來源管理員提供測試選項，您可以用來測試 ODBC 2.x 與 Amazon Athena 的連線。如需這些步驟，請參閱 [在 Windows 上設定資料來源名稱](#)。當您測試連線時，ODBC 驅動程式會呼叫 [GetWorkGroup](#) Athena API 動作。呼叫會使用您指定的身分驗證類型和對應的憑證提供者來擷取憑證。使用 ODBC 2.x 驅動程式時，可免費使用連線測試。測試不會在您的 Amazon S3 儲存貯體中產生查詢結果。

## 身分驗證選項

您可以使用下列身分驗證類型連線至 Amazon Athena。對於所有類型，連線字串名稱為 `AuthenticationType`，參數類型為 `Required`，預設值為 `IAM Credentials`。如需有關每種身分驗證類型的參數的資訊，請造訪對應的連結。如需常用身分驗證參數，請參閱 [常用身分驗證參數](#)。

身分驗證類型	連線字串範例
<a href="#">IAM 憑證</a>	<code>AuthenticationType=IAM Credentials;</code>

身分驗證類型	連線字串範例
<a href="#">IAM 設定檔</a>	AuthenticationType=IAM Profile;
<a href="#">AD FS</a>	AuthenticationType=ADFS;
<a href="#">Azure AD</a>	AuthenticationType=AzureAD;
<a href="#">瀏覽器 Azure AD</a>	AuthenticationType=BrowserAzureAD;
<a href="#">瀏覽器 SAML</a>	AuthenticationType=BrowserSAML;
<a href="#">瀏覽器 SSO OIDC</a>	AuthenticationType=BrowserSSOIDC;
<a href="#">預設憑證</a>	AuthenticationType=Default Credentials;
<a href="#">外部憑證</a>	AuthenticationType=External Credentials;
<a href="#">執行個體描述檔</a>	AuthenticationType=Instance Profile;
<a href="#">JWT</a>	AuthenticationType=JWT;
<a href="#">Okta</a>	AuthenticationType=Okta;
<a href="#">Ping</a>	AuthenticationType=Ping;

## IAM 憑證

您可以使用 IAM 憑證，以使用本節所述的連線字串參數，透過 ODBC 驅動程式連線至 Amazon Athena。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType=IAM Credentials;

## 使用者 ID

您的 AWS 存取金鑰 ID。如需有關存取金鑰的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS 安全憑證](#)。

連線字串名稱	參數類型	預設值	連線字串範例
UID	必要	none	UID=AKIAI OSFODNN7E XAMPLE;

## 密碼

您的 AWS 秘密金鑰 ID。如需有關存取金鑰的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS 安全憑證](#)。

連線字串名稱	參數類型	預設值	連線字串範例
PWD	必要	none	PWD=wJalr XUtnFEMI/ K7MDENG/b PxRfiCYEX AMPLEKE;

## 工作階段字串

如果您使用臨時 AWS 認證，則必須指定工作階段 Token。如需有關暫時憑證的資訊，請參閱《IAM 使用者指南》中的 [IAM 中的暫時安全憑證](#)。

連線字串名稱	參數類型	預設值	連線字串範例
SessionToken	選用	none	SessionTo ken=AQoDY XdzEJr... <remainder of session token>;

## IAM 設定檔

您可以將具名設定檔設定為使用 ODBC 驅動程式連線到 Amazon Athena。若要使用託管 Amazon EC2 執行個體設定檔中可用的憑證，請將 `credential_source` 參數設定為 `Ec2InstanceMetadata`。如果您要在具名設定檔中使用自訂憑證提供者，請在設定檔組態中指定 `plugin_name` 參數的值。

### 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credential s	AuthenticationType=IAM Profile;

## AWS 設定檔

用於 ODBC 連線的設定檔名稱。如需有關描述檔的詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的[使用具名描述檔](#)。

連線字串名稱	參數類型	預設值	連線字串範例
AWS設定檔	必要	none	AWSProfile=default;

## 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。當設定檔組態中的 `plugin_name` 參數指定自訂憑證提供者時，將使用偏好角色參數。如需有關 ARN 角色的詳細資訊，請參閱《AWS Security Token Service API 參考》中的[AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn: aws:IAM: :12345678 9012:id/user1;

## 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需有關工作階段持續時間的詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。當設定檔組態中的 `plugin_name` 參數指定自訂憑證提供者時，將使用工作階段持續時間參數。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	<code>duration=900;</code>

## 外掛程式名稱

指定在具名設定檔中使用的自訂憑證提供者名稱。此參數可以採用與「ODBC 資料來源管理員」的「驗證類型」欄位中的值相同，但僅供 `AWSProfile` 組態使用。

連線字串名稱	參數類型	預設值	連線字串範例
<code>plugin_name</code>	選用	none	<code>plugin_name=AzureAD;</code>

## AD FS

AD FS 是 SAML 型身分驗證外掛程式，可與 Active Directory Federation Service (AD FS) 身分提供者搭配使用。該外掛程式支援 [整合式 Windows 身分驗證](#) 和表單型身分驗證。如果您使用整合式 Windows 身分驗證，則可以省略使用者名稱和密碼。如需有關設定 AD FS 和 Athena 的資訊，請參閱 [使用 ODBC 用戶端為 Microsoft AD FS 使用者設定對 Amazon Athena 的聯合存取權](#)。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
<code>AuthenticationType</code>	必要	IAM Credentials	<code>AuthenticationType=ADFS;</code>

## 使用者 ID

連線至 AD FS 伺服器的使用者名稱。對於整合式 Windows 身分驗證，您可以省略使用者名稱。如果您的 AD FS 設定需要使用者名稱，您必須在連線參數中予以提供。

連線字串名稱	參數類型	預設值	連線字串範例
UID	對 Windows 整合式身分驗證為選用	none	UID=domain \username;

## 密碼

連線至 AD FS 伺服器的密碼。如同使用者名稱欄位，如果您使用整合式 Windows 身分驗證，您可以省略使用者名稱。如果您的 AD FS 設定需要密碼，您必須在連線參數中予以提供。

連線字串名稱	參數類型	預設值	連線字串範例
PWD	對 Windows 整合式身分驗證為選用	none	PWD=password_3EXAMPLE;

## 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如果您的 SAML 聲明具有多個角色，您可以指定此參數來選擇要擔任的角色。此角色應出現在 SAML 聲明中。如需有關 ARN 角色的詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn: aws:IAM: :12345678 9012:id/user1;

## 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需有關工作階段持續時間的詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	duration=900;

## IdP 主機

AD FS 服務主機的名稱。

連線字串名稱	參數類型	預設值	連線字串範例
idp_host	必要	none	idp_host=<server-name>.<company.com>;

## IdP 連接埠

用於連線至 AD FS 主機的連接埠。

連線字串名稱	參數類型	預設值	連線字串範例
idp_port	必要	none	idp_port=443;

## LoginToRP

可信依賴方。使用此參數來覆寫 AD FS 依賴方端點 URL。

連線字串名稱	參數類型	預設值	連線字串範例
LoginToRP	選用	urn:amazon:webservices	LoginToRP=trustedparty;

## Azure AD

Azure AD 是 SAML 型身分驗證外掛程式，可與 Azure AD 身分提供者搭配使用。本外掛程式不支援多重要素驗證 (MFA)。如果您需要 MFA 支援，請考慮改用 BrowserAzureAD 外掛程式。

### 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType =AzureAD;

### 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需有關 ARN 角色的資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn:aws:iam: :123456789012:id/user1;

### 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	duration=900;

### 租用戶 ID

指定您的應用程式租用戶 ID。

連線字串名稱	參數類型	預設值	連線字串範例
idp_tenant	必要	none	idp_tenant=123zz112z-z12d-1z1f-11zz-f111aa111234;

## 用戶端 ID

指定您的應用程式用戶端 ID。

連線字串名稱	參數類型	預設值	連線字串範例
client_id	必要	none	client_id=9178ac27-a1bc-1a2b-1a2b-a123abcd1234;

## Client secret (用戶端密碼)

指定您的用戶端密碼。

連線字串名稱	參數類型	預設值	連線字串範例
client_secret	必要	none	client_secret=zG12q~.xzG1xxZ1wX1.~ZzXXX1XxkHZizeT1zzZ;

## 瀏覽器 Azure AD

瀏覽器 Azure AD 是 SAML 型身分驗證外掛程式，可與 Azure AD 身分提供者搭配使用並支援多重因素認證。與標準 Azure AD 外掛程式不同，此外掛程式不需要連線參數中的使用者名稱、密碼或用戶端密碼。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentia ls	AuthenticationType =BrowserAzureAD;

## 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如果您的 SAML 聲明具有多個角色，您可以指定此參數來選擇要擔任的角色。指定的角色應出現在 SAML 聲明中。如需有關 ARN 角色的詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn :aws:IAM::12345678 9012:id/user1;

## 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需有關工作階段持續時間的詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	duration=900;

## 租用戶 ID

指定您的應用程式租用戶 ID。

連線字串名稱	參數類型	預設值	連線字串範例
idp_tenant	必要	none	idp_tenant=123zz112z-z12d-1z1f-11zz-f111aa111234;

## 用戶端 ID

指定您的應用程式用戶端 ID。

連線字串名稱	參數類型	預設值	連線字串範例
client_id	必要	none	client_id=9178ac27-a1bc-1a2b-1a2b-a123abcd1234;

## 逾時

外掛程式停止等待來自 Azure AD 之 SAML 回應之前的持續時間 (以秒為單位)。

連線字串名稱	參數類型	預設值	連線字串範例
timeout	選用	120	timeout=90;

## 啟用 Azure 檔案快取

啟用臨時憑證快取。此連線參數可讓您在多個程序之間快取，及重複使用臨時憑證。使用此選項可減少當您使用 BI 工具 (如 Microsoft Power BI) 時，開啟瀏覽器視窗的數目。

連線字串名稱	參數類型	預設值	連線字串範例
browser_azure_cache	選用	1	browser_azure_cache=0;

## 瀏覽器 SAML

瀏覽器 SAML 是一般身分驗證外掛程式，可與 SAML 型身分提供者搭配使用並支援多重因素認證。如需詳細的組態資訊，請參閱 [使用 ODBC、SAML 2.0 和 Okta 身分提供者設定單一登入](#)。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType=BrowserSAML;

## 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如果您的 SAML 聲明具有多個角色，您可以指定此參數來選擇要擔任的角色。此角色應出現在 SAML 聲明中。如需有關 ARN 角色的詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn:aws:IAM::123456789012:id/user1;

## 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	duration=900;

## 登入 URL

為您的應用程式顯示的單一登入 URL。

連線字串名稱	參數類型	預設值	連線字串範例
login_url	必要	none	login_url =https:// trial-123 4567.okta.com/ app/trial-123 4567_okta browsersaml_1/ zzz4izzzAzDFB zZz1234/sso/ saml;

### 接聽連接埠

用來接聽 SAML 回應的連接埠號碼。此值應與您設定 IdP 時使用的 IAM Identity Center URL 相符 (例如, http://localhost:7890/athena)。

連線字串名稱	參數類型	預設值	連線字串範例
listen_port	選用	7890	listen_po rt=7890;

### 逾時

外掛程式停止等待來自身分提供者的 SAML 回應之前的持續時間 (以秒為單位)。

連線字串名稱	參數類型	預設值	連線字串範例
timeout	選用	120	timeout=90;

### 瀏覽器 SSO OIDC

瀏覽器 SSO OIDC 是一個身份驗證插件，可與 AWS IAM Identity Center 如需有關啟用和使用 IAM Identity Center 的資訊，請參閱《AWS IAM Identity Center 使用者指南》中的 [步驟 1：啟用 IAM Identity Center](#)

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentia ls	AuthenticationType =BrowserSSOIDC;

## IAM Identity Center 啟動 URL

AWS 存取入口網站的 URL。IAM 身分中心 [StartDeviceAuthorization](#) API 動作會將此值用於 `startUrl` 參數。

若要複製 AWS 存取入口網站 URL

1. 請登入 AWS Management Console 並開啟 AWS IAM Identity Center 主控台，網址為 <https://console.aws.amazon.com/singlesignon/>。
2. 在導覽窗格中，選擇設定。
3. 在設定頁面的身分來源下，選擇 AWS 存取入口網站 URL 的剪貼簿圖示。

連線字串名稱	參數類型	預設值	連線字串範例
sso_oidc_start_url	必要	none	sso_oidc_start_url=https:// app_id.awsapps.com/start;

## IAM Identity Center 區域

您 AWS 區域的 SSO 的設定位置。SSOIDCClient 和 SSOClient AWS SDK 用戶端會使用此值做為 `region` 參數。

連線字串名稱	參數類型	預設值	連線字串範例
sso_oidc_region	必要	none	sso_oidc_region=us- east-1;

## 範圍

用戶端所定義的範圍清單。授權後，此清單會在授予存取字符時限制許可。IAM 身分中心 [RegisterClient](#) API 動作會將此值用於 `scopes` 參數。

連線字串名稱	參數類型	預設值	連線字串範例
<code>sso_oidc_scopes</code>	選用	none	<code>sso_oidc_scopes=scope1,scope2,scope3;</code>

## 帳戶 ID

指派給使用者的識別碼。AWS 帳戶 IAM 身分中心 [GetRoleCredentials](#) API 使用此值做為 `accountId` 參數。

連線字串名稱	參數類型	預設值	連線字串範例
<code>sso_oidc_account_id</code>	必要	none	<code>sso_oidc_account_id=123456789123;</code>

## 角色名稱

指派給使用者的角色的易記名稱。您為此權限集指定的名稱會以可用角色的形式出現在 AWS 存取入口網站中。IAM 身分中心 [GetRoleCredentials](#) API 動作會將此值用於 `roleName` 參數。

連線字串名稱	參數類型	預設值	連線字串範例
<code>sso_oidc_role_name</code>	必要	none	<code>sso_oidc_role_name=AthenaReadAccess;</code>

## 逾時

輪詢 SSO API 應檢查存取字符的秒數。

連線字串名稱	參數類型	預設值	連線字串範例
sso_oidc_timeout	選用	120	sso_oidc_timeout=60;

### 啟用檔案快取

啟用臨時憑證快取。此連線參數可讓您在多個程序之間快取，及重複使用臨時憑證。使用此選項可減少當您使用 BI 工具 (如 Microsoft Power BI) 時，開啟瀏覽器視窗的數目。

連線字串名稱	參數類型	預設值	連線字串範例
sso_oidc_cache	選用	1	sso_oidc_cache=0;

### 預設憑證

您可以使用在用戶端系統上設定的預設憑證連線至 Amazon Athena。如需有關使用預設憑證的資訊，請參閱《AWS SDK for Java 開發人員指南》中的[使用預設憑證提供者鏈結](#)。

### 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType=DefaultCredentials;

### 外部憑證

外部憑證的一般身分驗證外掛程式，可用來連線至任何外部 SAML 型身分提供者的外掛程式。若要使用外掛程式，您可以傳遞可傳回 SAML 回應的可執行檔。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType =External Credentials;

## 可執行路徑

包含自訂 SAML 型憑證提供者的邏輯的可執行檔路徑。可執行檔的輸出必須是身分提供者所解析的 SAML 回應。

連線字串名稱	參數類型	預設值	連線字串範例
ExecutablePath	必要	none	ExecutablePath=C:\Users <i>\user_name \external_</i> <i>credential.exe</i>

## 引數清單

您要傳遞至可執行檔的引數清單。

連線字串名稱	參數類型	預設值	連線字串範例
ArgumentList	選用	none	ArgumentList= <i>arg1 arg2</i> <i>arg3</i>

## 執行個體描述檔

此身份驗證類型用於 EC2 執行個體，並透過 Amazon EC2 中繼資料服務傳遞。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType= Instance Profile;

## JWT

JWT (JSON Web Token) 外掛程式提供了一個介面，該介面使用 JSON Web Tokens 來擔任 Amazon IAM 角色。組態取決於身分提供者。如需有關設定 Google Cloud 和 AWS 的聯合的資訊，請參閱 Google Cloud 文件中的[使用 AWS 或 Azure 設定工作負載聯合身分](#)。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType=JWT;

## 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需有關 ARN 角色的詳細資訊，請參閱《AWS Security Token Service API 參考》中的[AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn: aws:IAM: :12345678 9012:id/user1;

## 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需有關工作階段持續時間的詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [AssumeRole](#)。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	duration=900;

## JSON Web Token

JSON Web Token，用於使用 [AssumeRoleWithWebIdentity](#) AWS STS API 動作來擷取 IAM 暫時登入資料。如需有關為 Google Cloud Platform (GCP) 使用者產生 JSON Web Token 的資訊，請參閱 Google Cloud 文件中的 [使用 JWT OAuth 字符](#)。

連線字串名稱	參數類型	預設值	連線字串範例
web_identity_token	必要	none	web_identity_token=eyJhbGc. ..<remainder of token>;

## 角色工作階段名稱

工作階段的名稱。一種常見技術是將應用程式使用者的名稱或識別符做為角色工作階段名稱使用。這可輕鬆地將您應用程式使用的暫時安全登入資料與對應的使用者建立關聯。

連線字串名稱	參數類型	預設值	連線字串範例
role_session_name	必要	none	role_session_name=familiarn ame;

## Okta

Okta 是 SAML 型身分驗證外掛程式，可與 Okta 身分提供者搭配使用。如需有關設定 Okta 和 Amazon Athena 的聯合的資訊，請參閱 [使用 Okta 外掛程式和 Okta 身分提供者設定 ODBC 的 SSO](#)。

## 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType=Okta;

## 使用者 ID

您的 Okta 使用者名稱。

連線字串名稱	參數類型	預設值	連線字串範例
UID	必要	none	UID=jane.doe@org.com;

## 密碼

您的 Okta 使用者密碼。

連線字串名稱	參數類型	預設值	連線字串範例
PWD	必要	none	PWD=oktauserpasswordexample;

## 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如需 ARN 角色的詳細資訊，請參閱 AWS Security Token Service API 參考資料 [AssumeRole](#) 中的。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn

連線字串名稱	參數類型	預設值	連線字串範例
			:aws:IAM: :12345678 9012:id/user1;

### 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需詳細資訊，請參閱 [AWS Security Token Service API 參考 AssumeRole](#) 中的。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	duration=900;

### IdP 主機

您的 Okta 組織的 URL。您可以從 Okta 應用程式中的內嵌連結 URL 擷取 idp\_host 參數。如需這些步驟，請參閱 [從 Okta 擷取 ODBC 組態資訊](#)。https:// 後的第一個區段、直到並包括 okta.com 即為您的 IdP 主機 (例如 http://trial-1234567.okta.com)。

連線字串名稱	參數類型	預設值	連線字串範例
idp_host	必要	None	idp_host= dev-99999 999.okta.com;

### IdP 連接埠

用於連線至 IdP 主機的連接埠號碼。

連線字串名稱	參數類型	預設值	連線字串範例
idp_port	必要	None	idp_port=443;

## Okta 應用程式 ID

您應用程式的兩部分識別符。您可以從 Okta 應用程式中的內嵌連結 URL 擷取 app\_id 參數。如需這些步驟，請參閱 [從 Okta 擷取 ODBC 組態資訊](#)。應用程式 ID 是 URL 的最後兩個區段，包括中間的正斜線。這些區段是兩個由 20 個字元組成的字串，其中包含數字和大小寫字母的混合 (例如 Abc1de2fghi3J45kL678/abc1defghij2klmNo3p4)。

連線字串名稱	參數類型	預設值	連線字串範例
app_id	必要	None	app_id=0o a25kx8ze9 A3example /alnexamp lea0piaWa0g7;

## Okta 應用程式名稱

Okta 應用程式的名稱。

連線字串名稱	參數類型	預設值	連線字串範例
app_name	必要	None	app_name= amazon_aw s_redshift;

## Okta 等待時間

指定等待多重要素驗證 (MFA) 程式碼的持續時間 (以秒為單位)。

連線字串名稱	參數類型	預設值	連線字串範例
okta_mfa_wait_time	選用	10	okta_mfa_ wait_time=20;

## Okta MFA 類型

MFA 因素類型。支援的類型包括 Google Authenticator、SMS (Okta)、Okta Verify with Push 和 Okta Verify with TOTP。個別組織安全政策會判斷使用者登入是否需要 MFA。

連線字串名稱	參數類型	預設值	可能的值	連線字串範例
okta_mfa_type	Optional	None	googleauthenticator, smsauthentication, oktaverifywithpush, oktaverifywithtotp	okta_mfa_type=oktaverifywithpush;

## Okta 電話號碼

驗證時使用的電話號 AWS SMS 碼。僅多重要素註冊需要此參數。如果您的手機號碼已經註冊，或者安全政策未使用 AWS SMS 身分驗證，則可以忽略此欄位。

連線字串名稱	參數類型	預設值	連線字串範例
okta_mfa_phone_number	需要 MFA 註冊，否則為選用	None	okta_mfa_phone_number=19991234567;

## 啟用 Okta 檔案快取

啟用臨時憑證快取。此連線參數可讓您在 BI 應用程式開啟的多個程序之間快取及重複使用臨時憑證。使用此選項可避免 Okta API 限流限制。

連線字串名稱	參數類型	預設值	連線字串範例
okta_cache	選用	0	okta_cache=1;

## Ping

Ping 是一個基於 SAML 的插件，可與[PingFederate](#)身份提供者一起使用。

### 身分驗證類型

連線字串名稱	參數類型	預設值	連線字串範例
AuthenticationType	必要	IAM Credentials	AuthenticationType=Ping;

### 使用者 ID

伺服器的使用者名 PingFederate 稱。

連線字串名稱	參數類型	預設值	連線字串範例
UID	必要	none	UID=pingusername@domain.com;

### 密碼

PingFederate 伺服器的密碼。

連線字串名稱	參數類型	預設值	連線字串範例
PWD	必要	none	PWD=pingpassword;

### 偏好的角色

要擔任之角色的 Amazon Resource Name (ARN)。如果您的 SAML 聲明具有多個角色，您可以指定此參數來選擇要擔任的角色。此角色應出現在 SAML 聲明中。如需 ARN 角色的詳細資訊，請參閱 AWS Security Token Service API 參考資料[AssumeRole](#)中的。

連線字串名稱	參數類型	預設值	連線字串範例
preferred_role	選用	none	preferred_role=arn:aws:iam: :123456789012:id/user1;

### 工作階段持續時間

角色工作階段的持續時間 (以秒為單位)。如需工作階段持續時間的詳細資訊，請參閱 AWS Security Token Service API 參考 [AssumeRole](#) 中的。

連線字串名稱	參數類型	預設值	連線字串範例
持續時間	選用	900	duration=900;

### IdP 主機

Ping 伺服器的地址。若要尋找您的地址，請造訪下列 URL 並檢視 SSO 應用程式端點欄位。

```
https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections
```

連線字串名稱	參數類型	預設值	連線字串範例
idp_host	必要	none	idp_host=ec2-1-83-65-12.com pute-1.amazonaws.com;

### IdP 連接埠

用於連線至 IdP 主機的連接埠號碼。

連線字串名稱	參數類型	預設值	連線字串範例
idp_port	必要	None	idp_port=443;

### 合作夥伴 SPID

服務提供者地址。若要尋找服務提供者地址，請造訪下列 URL 並檢視 SSO 應用程式端點欄位。

```
https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections
```

連線字串名稱	參數類型	預設值	連線字串範例
partner_spid	必要	None	partner_spid=https://us-east-1.signin.aws.amazon.com/platform/saml/<...>;

### Ping URI 參數

將身分驗證請求的 URI 引數傳遞給 Ping。使用此參數可繞過 Lake Formation 的單一角色限制。設定 Ping 以識別傳遞的參數，並驗證傳遞的角色存在於分配給使用者的角色清單中。然後，在 SAML 聲明中傳送單個角色。

連線字串名稱	參數類型	預設值	連線字串範例
ping_uri_param	選用	None	ping_uri_param=role=my_iam_role;

### 常用身分驗證參數

如上所述，本節中的參數對於驗證類型而言非常常見。

### 將代理用於 IdP

透過代理 Proxy 啟用驅動程式與 IdP 之間的通訊。此選項適用於下列身分驗證外掛程式：

- AD FS
- Azure AD
- 瀏覽器 Azure AD
- 瀏覽器 SSO OIDC
- JWT
- Okta
- Ping

連線字串名稱	參數類型	預設值	連線字串範例
UseProxyForIdP	選用	0	UseProxyForIdP=1;

### 使用 Lake Formation

使用 [AssumeDecoratedRoleWithSAML](#) Lake Formation API 動作來擷取臨時 IAM 憑證，而非使用 [AssumeRoleWithSAML](#) AWS STS API 動作。此選項適用於 Azure AD、瀏覽器 Azure AD、瀏覽器 SAML、Okta、Ping 和 AD FS 身分驗證外掛程式。

連線字串名稱	參數類型	預設值	連線字串範例
LakeformationEnabled	選用	0	LakeformationEnabled=1;

### SSL 不安全 (IdP)

與 IdP 通訊時停用 SSL。此選項適用於 Azure AD、瀏覽器 Azure AD、Okta、Ping 和 AD FS 身分驗證外掛程式。

連線字串名稱	參數類型	預設值	連線字串範例
SSL_Insecure	選用	0	SSL_Insecure=1;

### 端點覆寫

#### Athena 端點覆寫

`endpointOverride` `ClientConfiguration` 類別會使用此值覆寫 Amazon Athena 用戶端的預設 HTTP 端點。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
EndpointOverride	選用	none	EndpointOverride=athena.us-west-2.amazonaws.com;

## Athena 串流端點覆寫

`ClientConfiguration.endpointOverride` 方法會使用此值覆寫 Amazon Athena 串流用戶端的預設 HTTP 端點。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。Athena 串流服務可透過連接埠 444 使用。

連線字串名稱	參數類型	預設值	連線字串範例
StreamingEndpointOverride	選用	none	<code>StreamingEndpointOverride=athena.us-west-1.amazonaws.com:444;</code>

## AWS STS 端點取代

此方 `ClientConfiguration.endpointOverride` 法會使用此值覆寫用 AWS STS 戶端的預設 HTTP 端點。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
StsEndpointOverride	選用	none	<code>StsEndpointOverride=sts.us-west-1.amazonaws.com;</code>

## Lake Formation 端點覆寫

`ClientConfiguration.endpointOverride` 方法會使用此值覆寫 Lake Formation 用戶端的預設 HTTP 端點。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
LakeFormationEndpointOverride	選用	none	<code>LakeFormationEndpointOverride=lakeformation.us-west-1.amazonaws.com;</code>

## SSO 端點覆寫

`ClientConfiguration.endpointOverride` 方法會使用此值覆寫 SSO 用戶端的預設 HTTP 端點。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
SSO EndpointOverride	選用	none	SSOEndpointOverride=portal.sso.us-east-2.amazonaws.com;

### SSO OIDC 端點覆寫

`ClientConfiguration.endpointOverride` 方法會使用此值覆寫 SSO OIDC 用戶端的預設 HTTP 端點。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
索伊德 EndpointOverride	選用	none	SSOOIDCEndpointOverride=oidc.us-east-2.amazonaws.com

### 進階選項

#### 擷取大小

本次請求傳回結果 (列) 的最大數量。如需參數資訊，請參閱 [GetQuery MaxResults](#)。對於串流 API，最大值為 10000000。

連線字串名稱	參數類型	預設值	連線字串範例
RowsToFetchPerBlock	選用	非串流為 1000 串流為 20000	RowsToFetchPerBlock=20000;

#### 啟用結果重複使用

指定執行查詢時是否可以重複使用之前的查詢結果。如需參數資訊，請參閱 [ResultReuseByAgeConfiguration](#)。

連線字串名稱	參數類型	預設值	連線字串範例
EnableResultReuse	選用	0	EnableResultReuse=1;

### 結果重複使用最長期限

指定 Athena 應考慮重複使用的之前查詢結果的最長期限 (以分鐘為單位)。如需參數資訊，請參閱 [ResultReuseByAgeConfiguration](#)。

連線字串名稱	參數類型	預設值	連線字串範例
ReusedResultMaxAgeInMinutes	選用	60	ReusedResultMaxAgeInMinutes=90;

### 啟用串流 API

選擇是否使用 Athena 串流 API 來擷取結果集。

連線字串名稱	參數類型	預設值	連線字串範例
UseResultsetStreaming	選用	0	UseResultsetStreaming=1;

### 啟用 S3 擷取程式

直接與 Amazon S3 互動，從 Amazon S3 儲存貯體擷取 Athena 產生的結果集。

連線字串名稱	參數類型	預設值	連線字串範例
EnableS3Fetcher	選用	1	EnableS3Fetcher=1;

## 使用多個 S3 執行緒

使用多個執行緒從 Amazon S3 擷取資料。啟用此選項時，會使用多個執行緒平行擷取存放在 Amazon S3 儲存貯體中的結果檔案。

只有在網路頻寬良好時，才會啟用此選項。例如，在我們對 EC2 [c5.2xlarge](#) 執行個體的測量中，單一執行緒 S3 用戶端達到 1 Gbps，而多執行緒 S3 用戶端則達到 4 Gbps 的網路輸送量。

連線字串名稱	參數類型	預設值	連線字串範例
UseMultipleS3 執行緒	選用	0	UseMultipleS3Threads=1;

## 使用單一目錄和結構描述

依預設，ODBC 驅動程式會查詢 Athena，以取得可用目錄和結構描述的清單。此選項會強制驅動程式使用「ODBC 資料來源管理員組態」對話方塊或連線參數指定的目錄和結構描述。

連線字串名稱	參數類型	預設值	連線字串範例
UseSingleCatalogAndSchema	選用	0	UseSingleCatalogAndSchema=1;

## 使用查詢列出表

對於 LAMBDA 目錄類型，可讓 ODBC 驅動程式提交 [SHOW TABLES](#) 查詢以取得可用資料表的清單。此設定是預設值。如果此參數設定為 0，ODBC 驅動程式會使用 Athena [ListTableMetadata](#) API 取得可用表格的清單。請注意，對於 LAMBDA 目錄類型，使用 `ListTableMetadata` 會導致效能回歸。

連線字串名稱	參數類型	預設值	連線字串範例
UseQueryToListTables	選用	1	UseQueryToListTables=1;

## 使用 WCHAR 的字符串類型

根據預設，ODBC 驅動程式會SQL\_VARCHAR針對 Athena 使用SQL\_CHAR和字串資料類型charvarcharstring、array、map<>、struct<>、和row。設定此參數以1強制驅動程式使SQL\_WVARCHAR用SQL\_WCHAR和字串資料類型。使用寬字元和寬變數字元類型，以確保可以正確儲存和擷取來自不同語言的字元。

連線字串名稱	參數類型	預設值	連線字串範例
使用 W CharForStringTypes	選用	0	UseWCharForStringTypes=1;

## 查詢外部目錄

指定驅動程式是否需要從 Athena 查詢外部目錄。如需詳細資訊，請參閱 [遷移至 ODBC 2.x 驅動程式](#)。

連線字串名稱	參數類型	預設值	連線字串範例
QueryExternalCatalogs	選用	0	QueryExternalCatalogs=1;

## 驗證 SSL

控制是否在使用 AWS SDK 時驗證 SSL 憑證。這個值會傳遞給 ClientConfiguration.verifySSL 參數。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
VerifySSL	選用	1	VerifySSL=0;

## S3 結果區塊大小

指定單一 Amazon S3 [GetObject](#) API 請求下載的區塊大小 (以位元組為單位)。預設值為 67108864 (64 MB)。允許的最小值和最大值為 10485760 (10 MB) 和 2146435072 (約 2 GB)。

連線字串名稱	參數類型	預設值	連線字串範例
S3 ResultBlockSize	選用	67108864	S3ResultBlockSize= 268435456;

### 字符串列長度

指定具有string資料類型之欄的欄長度。由於 Athena 使用沒有定義精確度的 [Apache Hive 字串資料類型](#)，所以 Athena 報告的預設長度是 2147483647 ()。INT\_MAX因為 BI 工具通常會預先配置資料行的記憶體，因此可能會導致高記憶體消耗量。為了避免這種情況，Athena ODBC 驅動程式會限制string資料類型資料行的報告精確度，並公開StringColumnLength連線參數，以便變更預設值。

連線字串名稱	參數類型	預設值	連線字串範例
StringColumnLength	選用	255	StringColumnLength =65535;

### 複雜類型欄長

指定具有複雜資料類型 (例如map、struct和) 的資料行的欄長度array。例如 [StringColumnLength](#)，Athena 針對具有複雜資料類型的資料行報告 0 精確度。Athena ODBC 驅動程式會為具有複雜資料類型的資料行設定預設精確度，並公開ComplexTypeColumnLength連線參數，以便變更預設值。

連線字串名稱	參數類型	預設值	連線字串範例
ComplexTypeColumnLength	選用	65535	ComplexTypeColumnL ength=123456;

### 可信 CA 憑證

指示 HTTP 用戶端在何處找到您的 SSL 憑證信任存放區。這個值會傳遞給 ClientConfiguration.caFile 參數。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
TrustedCerts	選用	%INSTALL_PATH%/bin	TrustedCerts=C:\\Program Files\\Amazon Athena ODBC Driver\\bin\\cacert.pem;

### 最短輪詢期間

指定輪詢 Athena 查詢執行狀態之前要等待的最小值 (以毫秒為單位)。

連線字串名稱	參數類型	預設值	連線字串範例
MinQueryExecutionPollingInterval	選用	100	MinQueryExecutionPollingInterval=200;

### 最長輪詢期間

指定輪詢 Athena 查詢執行狀態之前要等待的最大值 (以毫秒為單位)。

連線字串名稱	參數類型	預設值	連線字串範例
MaxQueryExecutionPollingInterval	選用	60000	MaxQueryExecutionPollingInterval=1000;

### 輪詢乘數

指定增加輪詢期間的因數。依預設，輪詢會以最短輪詢期間的值開始，並且每次輪詢都會加倍，直到達到最長輪詢期間的值為止。

連線字串名稱	參數類型	預設值	連線字串範例
QueryExecutionPollingIntervalMultiplier	選用	2	QueryExecutionPollingIntervalMultiplier=2;

## 最長輪詢持續時間

指定驅動程式可輪詢 Athena 查詢執行狀態的最大值 (以毫秒為單位)。

連線字串名稱	參數類型	預設值	連線字串範例
MaxPollDuration	選用	1800000	MaxPollDuration=1800000;

## 連線逾時

HTTP 連線等待建立連線的時間量 (以毫秒為單位)。此值是針對 `ClientConfiguration.connectTimeoutMs` Athena 用戶端設定的。若未指定，則使用 `curl` 預設值。如需有關連線參數的資訊，請參閱《AWS SDK for Java 開發人員指南》中的[用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
ConnectionTimeout	選用	0	ConnectionTimeout=2000;

## 請求逾時

指定 HTTP 用戶端的通訊端讀取逾時。此值是為 Athena 用戶端的 `ClientConfiguration.requestTimeoutMs` 參數設定的。如需參數資訊，請參閱《AWS SDK for Java 開發人員指南》中的[用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
RequestTimeout	選用	10000	RequestTimeout=30000;

## 代理選項

### 代理主機

若您要求使用者通過代理，請使用此參數來設定代理主機。此參數對應至 AWS SDK 中的 `ClientConfiguration.proxyHost` 參數。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
ProxyHost	選用	none	ProxyHost=127.0.0.1;

### 代理連接埠

使用此參數來設定代理連接埠。此參數對應至 AWS SDK 中的 `ClientConfiguration.proxyPort` 參數。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
ProxyPort	選用	none	ProxyPort=8888;

### 代理使用者名稱

使用此參數來設定代理使用者名稱。此參數對應至 AWS SDK 中的 `ClientConfiguration.proxyUserName` 參數。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
ProxyUID	選用	none	ProxyUID=username;

## 代理密碼

使用此參數來設定代理密碼。此參數對應至 AWS SDK 中的 `ClientConfiguration.proxyPassword` 參數。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

連線字串名稱	參數類型	預設值	連線字串範例
ProxyPWD	選用	none	ProxyPWD=password;

## 非代理主機

使用此選用參數，指定驅動程式在不使用代理的情況下連線至的主機。此參數對應至 AWS SDK 中的 `ClientConfiguration.nonProxyHosts` 參數。如需詳細資訊，請參閱《AWS SDK for C++ 開發人員指南》中的 [AWS 用戶端組態](#)。

該 `NonProxyHost` 連線參數會傳遞給 `CURLOPT_NOPROXY` curl 選項。如需 `CURLOPT_NOPROXY` 格式的相關資訊，請參閱 curl 文件中的 [CURLOPT\\_NOPROXY](#)。

連線字串名稱	參數類型	預設值	連線字串範例
NonProxyHost	選用	none	NonProxyHost=.amazonaws.com,localhost,.example.net,.example.com;

## 使用代理

啟用通過指定代理的使用者流量。

連線字串名稱	參數類型	預設值	連線字串範例
UseProxy	選用	none	UseProxy=1;

## 記錄選項

需要管理員權限才能修改此處所述的設定。若要進行變更，您可以使用 ODBC 資料來源管理員記錄選項對話方塊或直接修改 Windows 登錄檔。

### 日誌層級

此選項會啟用 ODBC 驅動程式日誌。在 Windows 中，您可以使用登錄檔或對話方塊啟用或停用記錄。此選項位於以下登錄檔路徑：

```
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Amazon Athena\ODBC\Driver
```

連線字串名稱	參數類型	預設值	連線字串範例
LogLevel	選用	0	LogLevel=1;

### 日誌路徑

指定儲存 ODBC 驅動程式日誌的檔案路徑。您可以使用登錄檔或對話方塊來設定此值。此選項位於以下登錄檔路徑：

```
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Amazon Athena\ODBC\Driver
```

連線字串名稱	參數類型	預設值	連線字串範例
LogPath	選用	none	LogPath=C:\Users\ <i>username</i> \projects\internal\trunk\;

### 使用 AWS 記錄器

指定是否啟用 AWS SDK 記錄。指定 1 表示啟用，0 表示停用。

連線字串名稱	參數類型	預設值	連線字串範例
UseAwsLogger	選用	0	UseAwsLogger=1;

## 遷移至 ODBC 2.x 驅動程式

由於大多數的 Athena ODBC 2.x 連線參數可向下相容 ODBC 1.x 驅動程式，因此您可以使用 Athena ODBC 2.x 驅動程式重複使用多數現有的連線字串。但是，下列連線參數需要修改。

### 日誌層級

雖然目前的 ODBC 驅動程式提供多種可用的記錄選項，但從 LOG\_OFF (0) 開始到 LOG\_TRACE (6)，Amazon Athena ODBC 驅動程式只有兩個值：0 (已停用) 和 1 (已啟用)。

如需有關記錄 ODBC 2.x 驅動程式的詳細資訊，請參閱 [記錄選項](#)。

	ODBC 1.x 驅動程式	ODBC 2.x 驅動程式
連線字串名稱	LogLevel	LogLevel
參數類型	選用	選用
預設值	0	0
可能的值	0-6	0, 1
連線字串範例	LogLevel=6;	LogLevel=1;

### MetadataRetrievalMethod

目前的 ODBC 驅動程式提供數個選項，以便從 Athena 擷取中繼資料。Amazon Athena ODBC 驅動程序取代了 MetadataRetrievalMethod，並始終使用 Amazon Athena API 來擷取中繼資料。

Athena 推出旗標 QueryExternalCatalogs，用於查詢外部目錄。若要使用目前的 ODBC 驅動程式查詢外部目錄，請將 MetadataRetrievalMethod 設定為 ProxyAPI。若要使用 Athena ODBC 驅動程式查詢外部目錄，請將 QueryExternalCatalogs 設定為 1。

	ODBC 1.x 驅動程式	ODBC 2.x 驅動程式
連線字串名稱	MetadataRetrievalMethod	QueryExternalCatalogs
參數類型	選用	選用

	ODBC 1.x 驅動程式	ODBC 2.x 驅動程式
預設值	Auto	0
可能的值	Auto, AWS Glue, ProxyAPI, Query	0,1
連線字串範例	MetadataRetrievalMethod=ProxyAPI;	QueryExternalCatalogs=1;

## 連線測試

當您測試 ODBC 1.x 驅動程式連線時，驅動程式會執行 SELECT 1 查詢，在 Amazon S3 儲存貯體中產生兩個檔案：一個用於結果集，另一個用於中繼資料。測試連線是根據 [Amazon Athena 定價](#) 政策收費。

當您測試 ODBC 2.x 驅動程式連線時，驅動程式會呼叫 [GetWorkGroup](#) Athena API 動作。呼叫會使用您指定的身分驗證類型和對應的憑證提供者來擷取憑證。使用 ODBC 2.x 驅動程式時，可免費使用連線測試，且測試不會在您的 Amazon S3 儲存貯體中產生查詢結果。

## 疑難排解 ODBC 2.x 驅動程式

如果您在使用 Amazon Athena ODBC 驅動程式時遇到問題，您可以聯絡 AWS Support (在 AWS Management Console 中，選擇支援、支援中心)。

請確保包含下列資訊，並提供將協助支援團隊了解您的使用案例的任何其他詳細資訊。

- 描述 – (必要) 包含以下詳細資訊的描述：您的使用案例以及預期行為和觀察到的行為之間的差異。包括任何可幫助支援工程師輕鬆解決問題的資訊。如果問題是間歇性出現，則指定發生問題的日期、時間戳記或間隔點。
- 版本資訊 – (必要) 驅動程式版本、作業系統以及您使用的應用程式的相關資訊。例如，「ODBC 驅動程序 1.2.3 版、Windows 10 (x64)、Power BI。」
- 日誌檔案 – (必要) 了解問題所需的 ODBC 驅動程式日誌檔案的數目下限。如需有關 ODBC 2.x 驅動程式的記錄選項的資訊，請參閱 [記錄選項](#)。
- 連線字串 – (必要) ODBC 連線字串或顯示您使用的連線參數的對話方塊的螢幕擷取畫面。如需有關連線參數的資訊，請參閱 [Athena ODBC 2.x 連線參數](#)。
- 問題步驟 – (選用) 如有可能，請包含可協助重現問題的步驟或獨立程式。
- 查詢錯誤資訊 – (選用) 如果出現涉及 DML 或 DDL 查詢的錯誤，請包含以下資訊：

- 失敗的 DML 或 DDL 查詢的完整版本或簡化版本。
- 使用的帳戶 ID 和 AWS 區域，以及查詢執行 ID。
- SAML 錯誤 – (選用) 如果您遇到有關使用 SAML 聲明進行身分驗證的問題，請包含以下資訊：
  - 使用的身分提供者和身分驗證外掛程式。
  - 使用 SAML 字符的範例。

## Amazon Athena ODBC 2.x 版本備註

這些版本備註提供了 Amazon Athena ODBC 2.x 驅動程式中的增強功能、功能、已知問題和工作流程變更的詳細資訊。

### 2.0.3.0

二零二四年四月八日發表

Amazon Athena ODBC v2.0.3.0 驅動程式包含以下改進和修正。

#### 改善項目

- 在 Linux 和 Mac 平台上添加了對 Okta 身份驗證插件的 MFA 支持。
- Windows 的程式 `athena-odbc.dll` 庫和 `AmazonAthenaODBC-2.x.x.x.msi` 安裝程式現在都已簽署。
- 更新了隨驅動程式一起安裝的 CA 憑證 `cacert.pem` 檔案。
- 改善在 Lambda 目錄下列出表格所需的時間。對於 LAMBDA 目錄類型，ODBC 驅動程式現在可以提交 [SHOW TABLES](#) 查詢以取得可用資料表的清單。如需詳細資訊，請參閱 [使用查詢列出表](#)。
- 引入使用 SQL\_WCHAR 和報告字串資料類型的 `UseWCharForStringTypes` 連接參數 SQL\_WVARCHAR。如需詳細資訊，請參閱 [使用 WCHAR 的字符串類型](#)。

#### 修正項目

- 修正使用 `Get-OdbcDsn` PowerShell 工具時發生的登錄損毀警告。
- 已更新剖析邏輯，以便在查詢字串開頭處理註解。
- 日期和時間戳記資料類型現在允許在 `year` 欄位中為零。

若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## 2.0.2.2

二零二一四年二月十三日

Amazon Athena ODBC v2.0.2.2 驅動程序包含以下改進和修復。

### 改善項目

- 已新增兩個連線參數ComplexTypeColumnLength，以StringColumnLength及，您可以用來變更字串和複雜資料類型的預設欄長度。如需詳細資訊，請參閱 [字串列長度](#) 及 [複雜類型欄長](#)。
- 已為 Linux 和 macOS ( 英特爾和 ARM ) 操作系統添加了 Support。如需詳細資訊，請參閱 [Linux](#) 及 [macOS](#)。
- AWS-SDK-標籤版本已更新至 1.11.245 標籤版本。
- 捲曲程式庫已更新至 8.6.0 版本。

### 修正項目

- 已解決造成有效位數欄中字串類似字串資料類型的結果集中繼資料中報告不正確值的問題。

若要下載 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## 2.0.2.1

2023-12-07 發布

Amazon Athena ODBC v2.0.2.1 驅動程式包含下列改善和修正。

### 改善項目

- 改進了所有介面的 ODBC 驅動程式執行緒安全性。
- 啟用日誌記錄時，現在會以毫秒級精確度記錄日期時間值。
- 在使用 [瀏覽器 SSO OIDC](#) 外掛程式進行身分驗證期間，現在會開啟終端以向使用者顯示裝置代碼。

### 修正項目

- 解決了剖析串流 API 結果時發生的記憶體釋放問題。
- 介面 SQLTablePrivileges()、SQLSpecialColumns()、SQLProcedureColumns() 和 SQLProcedures() 的請求現在會傳回空白結果集。

若要下載 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## 2.0.2.0

2023-10-17 發布

Amazon Athena ODBC v2.0.2.0 驅動程式包含下列改善和修正。

### 改善項目

- 為瀏覽器 Azure AD、瀏覽器 SSO OIDC 和 Okta 基於瀏覽器的身分驗證外掛程式新增了檔案快取功能。

BI 工具 (如 Power BI) 和基於瀏覽器的外掛程式使用多個瀏覽器視窗。新的檔案快取連線參數可讓您在 BI 應用程式開啟的多個程序之間快取及重複使用臨時憑證。

- 應用程式現在可以在陳述式準備好後查詢結果集的相關資訊。
- 預設連線和請求逾時已增加，以使用於速度較慢的用戶端網路。如需詳細資訊，請參閱 [連線逾時](#) 及 [請求逾時](#)。
- 已針對 SSO 和 SSO OIDC 新增端點覆寫。如需詳細資訊，請參閱 [端點覆寫](#)。
- 已新增連線參數，將身分驗證請求的 URI 引數傳遞至 Ping。您可以使用此參數繞過 Lake Formation 的單一角色限制。如需詳細資訊，請參閱 [Ping URI 參數](#)。

### 修正項目

- 已修正使用資料列型繫結機制時發生的整數溢位問題。
- 已從瀏覽器 SSO OIDC 身分驗證外掛程式的必要連線參數清單中移除逾時。
- 為 `SQLStatistics()`、`SQLPrimaryKeys()`、`SQLForeignKeys()` 和 `SQLColumnPrivileges()` 新增了缺少的介面，並新增了依請求傳回空白結果集的功能。

若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## 2.0.1.1

2023-08-10 發布

Amazon Athena ODBC v2.0.1.1 驅動程式包含下列改進和修正。

## 改善項目

- 新增 URI 記錄至 Okta 身分驗證外掛程式。
- 新增偏好的角色參數至外部憑證提供者外掛程式
- 在 AWS 組態檔案的描述檔名稱中新增對描述當字首的處理。

## 修正項目

- 更正了與 Lake Formation 和 AWS STS 客戶合作時發生的 AWS 區域 使用問題。
- 將缺少的分割區索引鍵還原至資料表欄清單中。
- 新增缺少的 BrowserSSOIDC 身分驗證類型至 AWS 描述檔。

若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。

### 2.0.1.0

2023-06-29 發布

Amazon Athena 發布 ODBC v2.0.1.0 驅動程式。

Athena 發布了新的 ODBC 驅動程式，可改善從相容 SQL 開發和商業智慧應用程式連線、查詢和視覺化資料的體驗。Athena ODBC 驅動程式最新版本支援現有驅動程式的功能，且可輕鬆升級。最新版本包含透過 [AWS IAM Identity Center](#) 對使用者進行身分驗證的支援。它還提供了讀取來自 Amazon S3 的查詢結果的選項，可讓您更快取得查詢結果。

如需更多詳細資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## Athena ODBC 1.x 驅動程式

使用本頁面中的連結下載 Amazon Athena 1.x ODBC 驅動程式授權合約、ODBC 驅動程式和 ODBC 文件。有關 ODBC 連接字串的相關資訊，請參閱《ODBC 驅動程式安裝和組態指南》PDF 檔案，可從本頁面下載該檔案。如需許可的相關資訊，請參閱[透過 JDBC 和 ODBC 連接存取](#)。

### Important

當您使用 ODBC 1.x 驅動程式時，務必注意以下要求：

- 打開連接埠 444 – 將 Athena 用來串流查詢結果的連接埠 444 保持開放給輸出流量。當您使用 PrivateLink 端點連線至 Athena 時，請確定連接到 PrivateLink 端點的安全群組對連接埠 444 的輸入流量開放。

- 雅典娜/ GetQueryResultsStream 政策 — 將athena:GetQueryResultsStream政策動作新增至使用 ODBC 驅動程式的 IAM 主體。此政策動作不會直接透過 API 公開。它僅與 ODBC 和 JDBC 驅動程式搭配使用，做為串流結果支援的一部分。如需政策範例，請參閱 [AWS 受管理的策略：AWSQuicksightAthenaAccess](#)。

## Windows

驅動程式版本	下載連結
視窗 32 位元版本 1.2.3.1000	<a href="#">視窗 32 位元驅動程式 1.2.3.1000</a> 位驅動程式
視窗 64 位元版本 1.2.3.1000	<a href="#">視窗 64 位元驅動程式 1.2.3.1000</a> 位元硬碟驅動程式

## Linux

驅動程式版本	下載連結
適用於 32 位元的 ODBC 1.2.3.1000	<a href="https://downloads.athena.us-east-1.amazonaws.com/drivers/ODBC/SimbaAthenaODBC_1.2.3.1000/Linux/simbaathena-1.2.3.1000-1.el7.i686.rpm">https://downloads.athena.us-east-1.amazonaws.com/drivers/ODBC/SimbaAthenaODBC_1.2.3.1000/Linux/simbaathena-1.2.3.1000-1.el7.i686.rpm</a>
適用於 64 位元的 ODBC 1.2.3.1000	<a href="#">64 位元硬碟驅動程式 1.2.3.1000</a> 程式

## OSX

驅動程式版本	下載連結
適用於 OSX 的 ODBC 1.2.3.1000	<a href="https://downloads.athena.us-east-1.amazonaws.com/drivers/ODBC/SimbaAthenaODBC_1.2.3.1000/OSX/SimbaAthena_1.2.3.1000.dmg">https://downloads.athena.us-east-1.amazonaws.com/drivers/ODBC/SimbaAthenaODBC_1.2.3.1000/OSX/SimbaAthena_1.2.3.1000.dmg</a>

## 文件

內容	文件連結
Amazon Athena ODBC 驅動程式授權合約	<a href="#">授權合約</a>
一千年二月三十一日的文件	<a href="#">驅動程式安裝與設定指南版本 1.2.3.1000 與設定指南版本 1.2.3.1000</a>
版本 1.2.3.1000 的發行公告	<a href="#">驅動程式版本說明：版本 1.2.3.1000 驅動程式版本說明版本 1.2.3.1000</a>

## ODBC 驅動程式備註

## 不使用 Proxy 之連線

如果您要指定驅動程式在不使用 Proxy 的情況下連線的特定主機，您可以使用 ODBC 連線字串中的選用 NonProxyHost 屬性。

該 NonProxyHost 屬性會指定一份以逗號分隔的主機清單，讓連接器可以在啟用 Proxy 連接時，無需經過代理伺服器即可存取，如下列範例所示：

```
.amazonaws.com,localhost,.example.net,.example.com
```

該 NonProxyHost 連線參數會傳遞給 CURLOPT\_NOPROXY curl 選項。如需 CURLOPT\_NOPROXY 格式的相關資訊，請參閱 curl 文件中的 [CURLOPT\\_NOPROXY](#)。

## 使用 ODBC 用戶端為 Microsoft AD FS 使用者設定對 Amazon Athena 的聯合存取權

若要使用 ODBC 用戶端為 Microsoft Active Directory Federation Services (AD FS) 使用者設定 Amazon Athena 的聯合存取權，您必須先在 AD FS 和您的 AWS 帳戶之間建立信任。有了此信任，您的 AD 使用者就可以使用其 AD 憑證[聯合](#)到 AWS，並取得 [AWS Identity and Access Management \(IAM\)](#) 角色存取 Athena API 等 AWS 資源的許可。

若要建立此信任，您可將 AD FS 作為 SAML 提供者新增至您的 AWS 帳戶，並建立聯合身分使用者可以擔任的 IAM 角色。在 AD FS 端，您可以將 AWS 新增為依存方，並撰寫 SAML 宣告規則，以將正確的使用者屬性傳送至 AWS 進行授權 (特別是 Athena 和 Amazon S3)。

設定 AD FS 存取 Athena，主要步驟如下所示：

## 1. 設定 IAM SAML 提供者和角色

## 2. 設定 AD FS

## 3. 建立 Active Directory 使用者和群組

## 4. 設定 AD FS ODBC 與 Athena 之間的連線

### 1. 設定 IAM SAML 提供者和角色

在本節中，您可將 AD FS 作為 SAML 提供者新增至您的 AWS 帳戶，並建立聯合身分使用者可以擔任的 IAM 角色。

若要設定 SAML 提供者

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在導覽窗格中，請選擇 Identity providers (身分提供者)。
3. 選擇 Add provider (新增提供者)。
4. 在 Provider type (提供者類型) 選擇 SAML。

The screenshot displays the AWS IAM console interface for creating a new identity provider. On the left, the navigation menu is visible, with 'Identity providers' highlighted in a red box. The main content area shows the 'Create Identity Provider' page. The 'Configure provider' section is active, and the 'SAML' provider type is selected. The 'Provider name' field is filled with 'adfs-saml-provider'. Under the 'Metadata document' section, the 'Choose file' button is visible, and the 'FederationMetadata.xml' file is selected with a green checkmark.

5. 對於 Provider name (供應商名稱)，輸入 **adfs-saml-provider**。
6. 在瀏覽器中，輸入下列位址，以下載 AD FS 伺服器的聯合 XML 檔案。若要執行此步驟，您的瀏覽器必須能夠存取 AD FS 伺服器。

```
https://adfs-server-name/federationmetadata/2007-06/federationmetadata.xml
```

7. 在 IAM 主控台中，對於 Metadata document (中繼資料文件)，選擇 Choose file (選擇檔案)，然後將聯合中繼資料檔案上傳至 AWS。
8. 若要完成，請選擇 Add provider (新增提供者)。

接著，建立聯合身分使用者可擔任的 IAM 角色。

## 若要為聯合身分使用者建立 IAM 角色

1. 在 IAM 主控台導覽窗格中，選擇 Roles (角色)。
2. 選擇 建立角色。
3. 在 Trusted entity type (可信實體類型) 中，選擇 SAML 2.0 federation (SAML 2.0 聯合)。
4. 對於 SAML 2.0-based provider (以 SAML 2.0 為基礎的提供者)，請選擇您建立的 adfs-saml-provider 提供者。
5. 選擇允許程式設計和 AWS 管理主控台存取，然後選擇下一步。

**Select trusted entity**

**Trusted entity type**

- AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this a

**SAML 2.0-based provider**

adfs-saml-provider ▼ [Refresh] [Create n]

- Allow programmatic access only
- Allow programmatic and AWS Management Console access

Attribute

6. 在 Add permissions (新增許可) 頁面上，篩選此角色所需的 IAM 許可政策，然後選取對應的核取方塊。本教學課程會連接 AmazonAthenaFullAccess 和 AmazonS3FullAccess 政策。

## Add permissions [Info](#)

### Permissions policies

(Selected 1/838)



Create policy [↗](#)

#### Info

Choose one or more policies to attach to your new role.

Q Filter policies by property or policy name and pres 1 match < 1 >

"AmazonAthenaFull" X

Clear filters

<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input checked="" type="checkbox"/>	AmazonAthenaFullAccess	AWS managed	Provide full access to

### ▶ Set permissions boundary - optional [Info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

## Add permissions [Info](#)

**Permissions policies**  
(Selected 2/838)

[Info](#)

Choose one or more policies to attach to your new role.

1 match < 1 > [Settings](#)

"AmazonS3FullAccess" [X](#) [Clear filters](#)

<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input checked="" type="checkbox"/>	<a href="#">+</a> <a href="#">📦</a> AmazonS3FullAccess	AWS managed	Provides full access

**▶ Set permissions boundary - optional** [Info](#)  
 Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

[Cancel](#) [Previous](#) [Next](#)

- 選擇 Next (下一步)。
- 在 Name, review, and create (名稱、檢閱和建立) 頁面的 Role name (角色名稱) 中，輸入角色的名稱。本教學課程使用名稱 adfs-data-access。

在 Step 1: Select trusted entities (步驟 1：選取可信實體) 中，應使用 "Federated:"  
 "arn:aws:iam::*account\_id*:saml-provider/adfs-saml-provider" 自動  
 填入 Principal (主體) 欄位。Condition 欄位應包含 "SAML:aud" 和 "https://  
 signin.aws.amazon.com/saml"。

Step 1: Select trusted entities Edit

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRoleWithSAML",
7       "Principal": {
8         "Federated": "arn:aws:iam::[redacted]:saml-provider/adfs-saml-provider"
9       },
10      "Condition": {
11        "StringEquals": {
12          "SAML:aud": [
13            "https://signin.aws.amazon.com/saml"
14          ]
15        }
16      }
17    }
18  ]
19 }

```

Step 2: Add permissions (步驟 2：新增許可) 會說明您已連接至角色的政策。

Step 2: Add permissions Edit

Permissions policy summary

Policy name <a href="#">↗</a>	Type	Attached as
<a href="#">AmazonAthenaFullAccess</a>	AWS managed	Permissions policy
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy

- 選擇 建立角色。橫幅訊息會確認角色的建立。
- 請在 Roles (角色) 頁面上，選擇您剛剛建立的角色名稱。角色的摘要頁面會說明已連接的政策。

IAM > Roles > adfs-data-access

# adfs-data-access

## Summary

Creation date August 30, 2022, 16:33 (UTC-07:00)	ARN arn:aws:iam::
Last activity 1 hour ago	Maximum session duration 1 hour

Permissions | Trust relationships | Tags | Access Advisor | Revoke session

### Permissions policies (2)

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

<input type="checkbox"/>	Policy name	Type
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed
<input type="checkbox"/>	AmazonAthenaFullAccess	AWS managed

## 2. 設定 AD FS

現在，您已準備好將 AWS 新增為依存方並撰寫 SAML 宣告規則，因此您可以將正確的使用者屬性傳送至 AWS 進行授權。

以 SAML 為基礎的聯合有兩個參與者：IdP (Active Directory) 和依存方 (AWS)，這是使用 IdP 身分驗證的服務或應用程式。

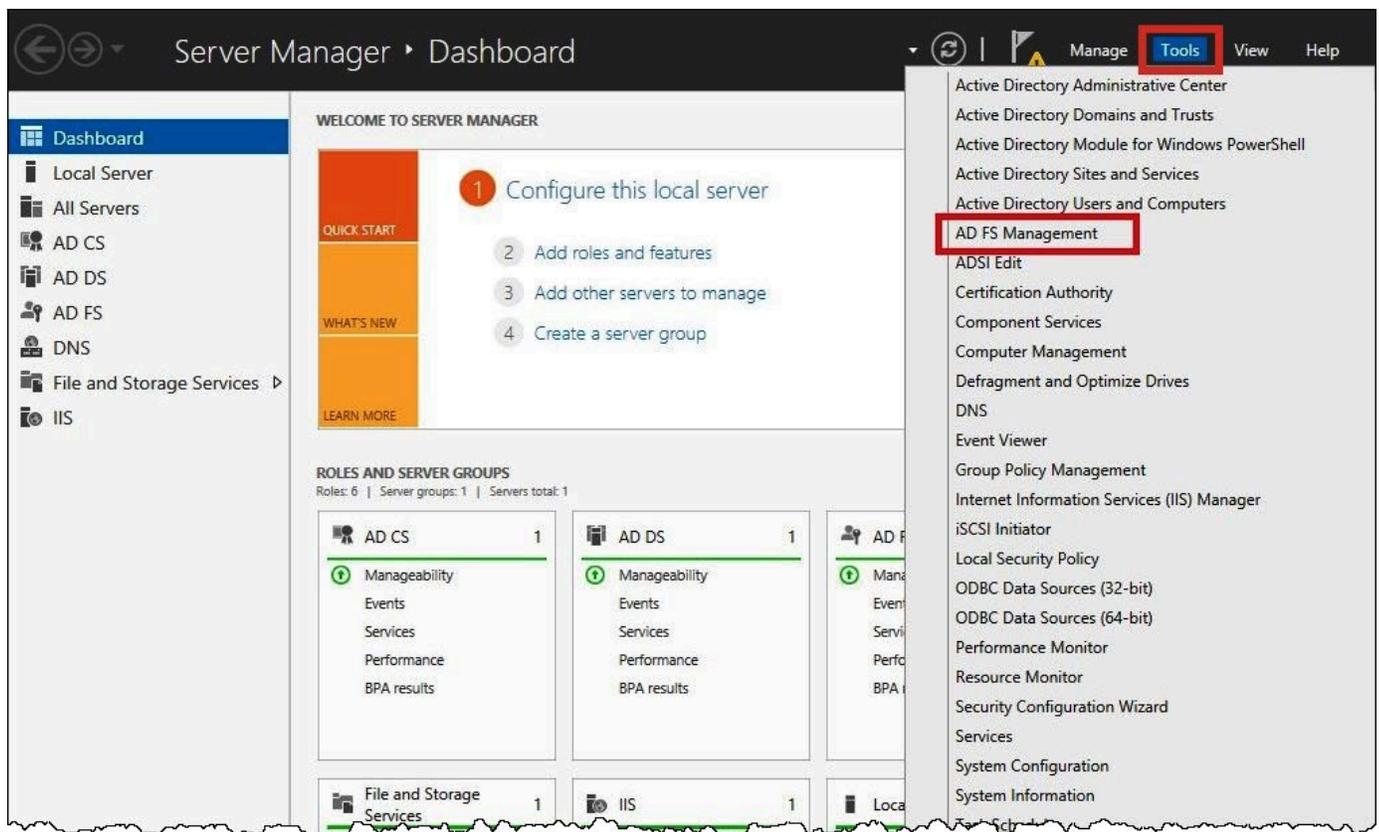
若要設定 AD FS，請先新增依存方信任，然後為依存方設定 SAML 宣告規則。AD FS 使用宣告規則來形成傳送給依存方的 SAML 聲明。SAML 聲明指出 AD 使用者的相關資訊為 true，並且已對使用者進行身分驗證。

## 新增依存方信任

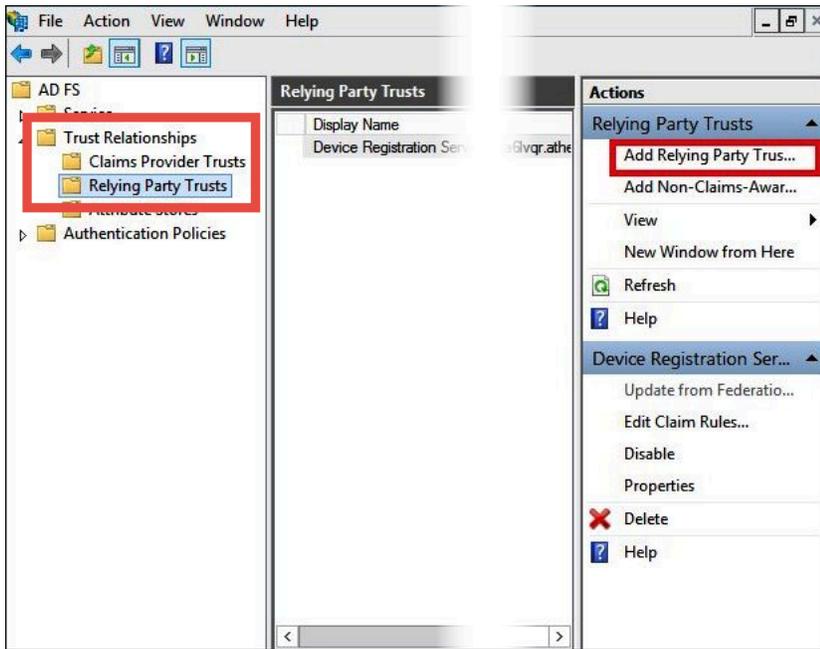
若要在 AD FS 中新增依存方信任，您可以使用 AD FS 伺服器管理員。

若要在 AD FS 中新增依存方信任

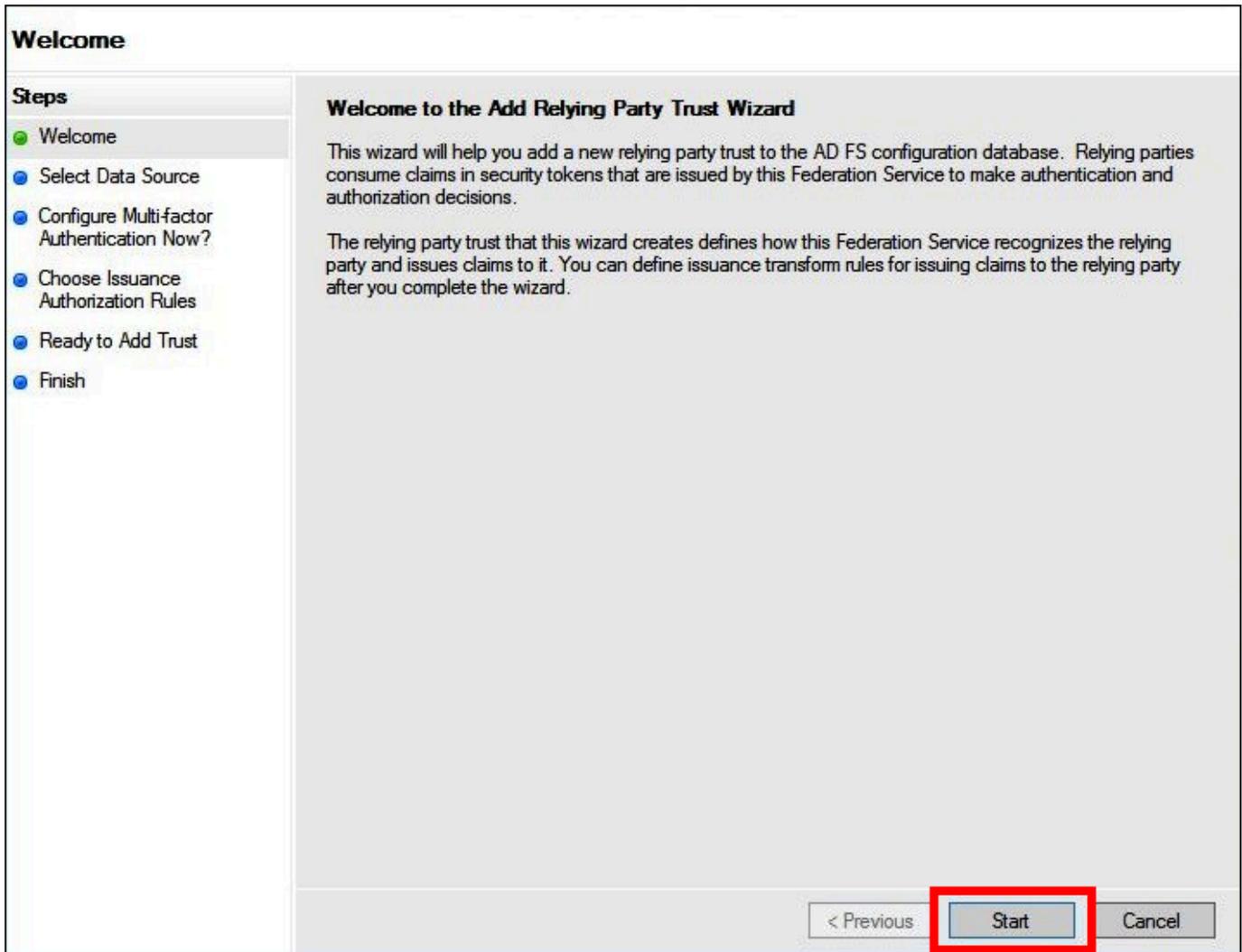
1. 登入 AD FS 伺服器。
2. 在 Start (開始) 選單上，開啟 Server Manager (伺服器管理員)。
3. 選擇 Tools (工具)，然後選擇 AD FS Management (AD FS 管理)。



4. 在導覽窗格的 Trust Relationships (信任關係) 下，選擇 Relying Party Trusts (依存方信任)。
5. 在 Actions (動作) 下，選擇 Add Relying Party Trust (新增依存方信任)。



6. 在 Add Relying Party Trust Wizard (新增信賴方信任精靈) 頁面上，選擇 Start (開始)。



7. 在 Select Data Source (選取資料來源) 頁面上，選取 Import data about the relying party published online or on a local network (匯入關於在線上或本機網路上發佈依存方的資料) 選項。
8. 針對 Federation metadata address (host name or URL) (聯合中繼資料地址 (主機名稱或 URL))，輸入 URL **`https://signin.aws.amazon.com/static/saml-metadata.xml`**
9. 選擇 Next (下一步)。

### Select Data Source

**Steps**

- Welcome
- Select Data Source
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Select an option that this wizard will use to obtain data about this relying party:

Import data about the relying party published online or on a local network

Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.

Federation metadata address (host name or URL):

Example: ts.contoso.com or https://www.contoso.com/app

Import data about the relying party from a file

Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.

Federation metadata file location:

Enter data about the relying party manually

Use this option to manually input the necessary data about this relying party organization.

< Previous 

- 在 Specify Display Name (指定顯示名稱) 頁面上，對於 Display name (顯示名稱)，輸入您的依存方的顯示名稱，然後選擇 Next (下一步)。

**Specify Display Name**

Enter the display name and any optional notes for this relying party.

Display name:  
signin.aws.amazon.com

Notes:

< Previous   **Next >**   Cancel

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

11. 在 Configure Multi-factor Authentication Now (現在設定多重要素驗證) 頁面上，此教學課程會選取 I do not want to configure multi-factor authentication for this relying party trust at this time (我不想此時針對該依存方設定多重要素驗證)。

為了提高安全性，建議您設定多重要素驗證以協助保護您的 AWS 資源。因為使用範例資料集，因此本教學課程不會啟用多重要素驗證。

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Configure multi-factor authentication settings for this relying party trust. Multi-factor authentication is required if there is a match for any of the specified requirements.

Multi-factor Authentication	Global Settings
Requirements	Not configured
Users/Groups	Not configured
Device	Not configured
Location	Not configured

I do not want to configure multi-factor authentication settings for this relying party trust at this time.

Configure multi-factor authentication settings for this relying party trust.

You can also configure multi-factor authentication settings for this relying party trust by navigating to the Authentication Policies node. For more information, see [Configuring Authentication Policies](#).

< Previous   **Next >**   Cancel

- 選擇 Next (下一步)。
- 在 Choose Issuance Authorization Rules (選擇發行授權規則) 頁面上，選取 Permit all users to access this relying party (允許所有使用者存取此依存方)。

此選項允許 Active Directory 中的所有使用者搭配使用 AD FS 和 AWS 作為依存方。您應該考量您的安全要求，並適當地對此組態進行調整。

### Choose Issuance Authorization Rules

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules**
- Ready to Add Trust
- Finish

Issuance authorization rules determine whether a user is permitted to receive claims for the relying party. Choose one of the following options for the initial behavior of this relying party's issuance authorization rules.

Permit all users to access this relying party

The issuance authorization rules will be configured to permit all users to access this relying party. The relying party service or application may still deny the user access.

Deny all users access to this relying party

The issuance authorization rules will be configured to deny all users access to this relying party. You must later add issuance authorization rules to enable any users to access this relying party.

You can change the issuance authorization rules for this relying party trust by selecting the relying party trust and clicking Edit Claim Rules in the Actions pane.

< Previous **Next >** Cancel

14. 選擇 Next (下一步)。

15. 在 Ready to Add Trust (準備新增信任) 頁面上，選擇 Next (下一步) 以將依存方信任新增至 AD FS 組態資料庫。

### Ready to Add Trust

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust**
- Finish

The relying party trust has been configured. Review the following settings, and then click Next to add the relying party trust to the AD FS configuration database.

Monitoring Identifiers Encryption Signature Accepted Claims Organization Endpoints Notes < >

Specify the monitoring settings for this relying party trust.

Relying party's federation metadata URL:

Monitor relying party

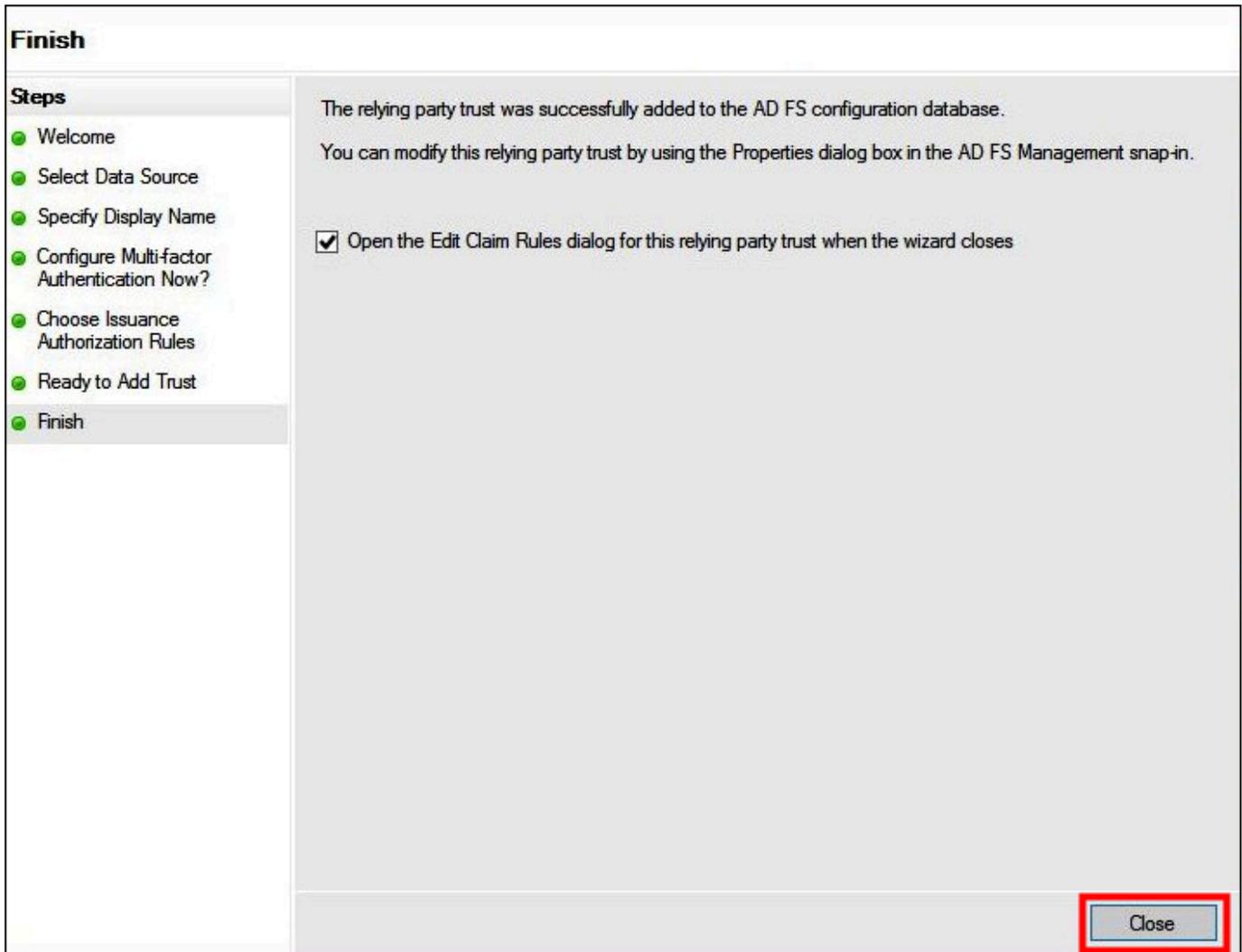
Automatically update relying party

This relying party's federation metadata data was last checked on:  
9/1/2022

This relying party was last updated from federation metadata on:  
9/1/2022

< Previous **Next >** Cancel

16. 在 Finish (完成) 頁面上，選擇 Close (關閉)。



## 設定依存方的 SAML 宣告規則

在此任務中，您會建立兩組宣告規則。

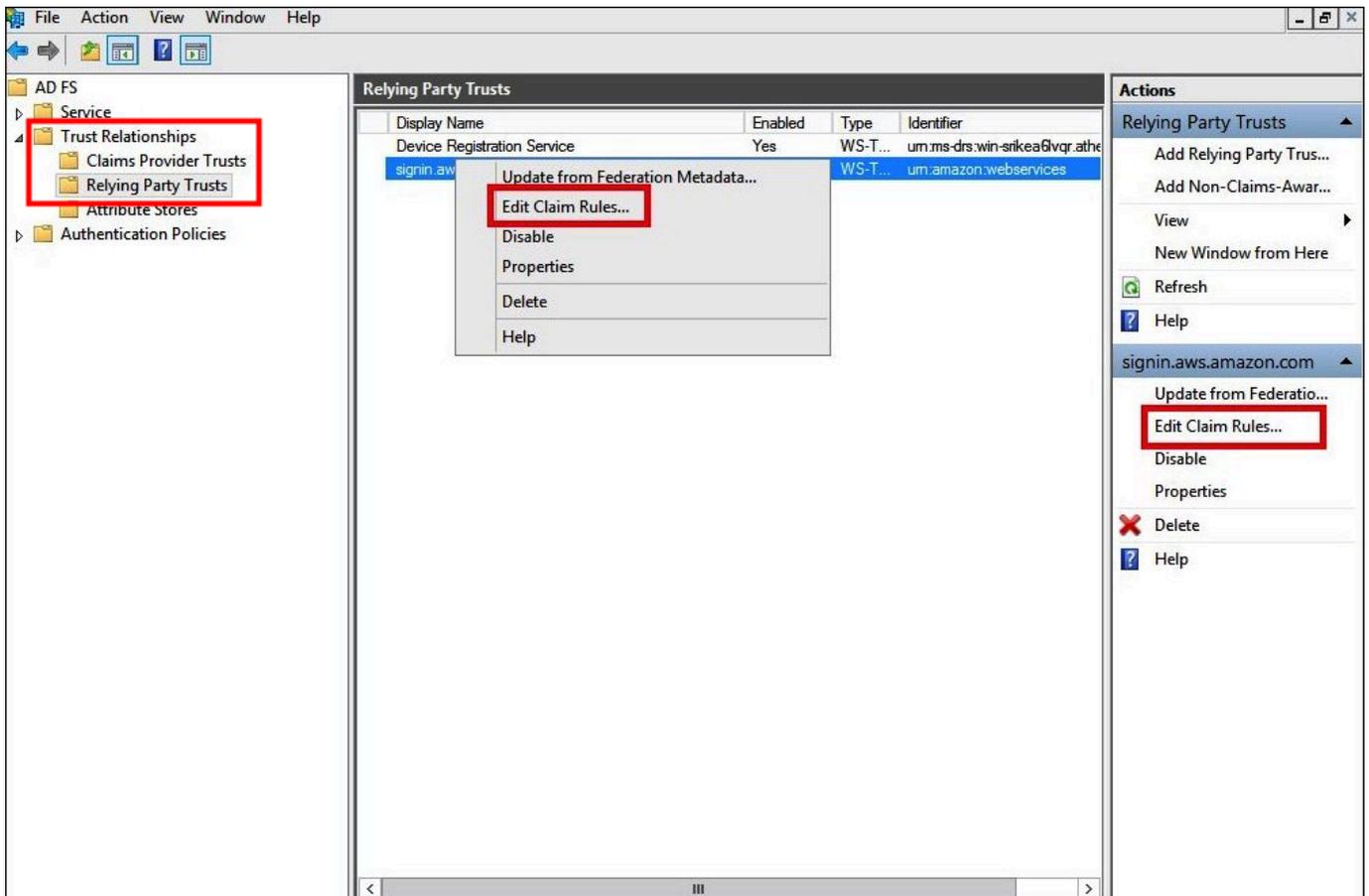
第一組規則 1–4, 包含基於 AD 群組成員資格擔任 IAM 角色所需的 AD FS 宣告規則。如果您想要建立 [AWS Management Console](#) 的聯合存取權，這些規則與您建立的規則相同。

第二組規則 5–6 是 Athena 存取控制所需的宣告規則。

### 若要建立 AD FS 宣告規則

1. 在 AD FS 管理主控台導覽窗格中，選擇 Trust Relationships (信任關係)、Relying Party Trusts (依存方信任)。
2. 尋找您在上一節建立的依存方。

3. 在依存方上按一下滑鼠右鍵，選擇 Edit Claim Rules (編輯宣告規則)，或從 Actions (動作) 選單中選擇 Edit Claim Rules (編輯宣告規則)。



4. 選擇 Add Rule (新增規則)。
5. 在 Add Transform Claim Rule Wizard (新增轉換宣告規則精靈) 的 Configure Rule (設定規則) 頁面上，輸入下列資訊以建立宣告規則 1，然後選擇 Finish (完成)。
  - 針對 Claim Rule name (宣告規則名稱)，輸入 **NameID**。
  - 針對 Rule template (規則範本)，使用 Transform an Incoming Claim (轉換傳入宣告)。
  - 針對 Incoming claim type (傳入宣告類型)，選擇 Windows account name (Windows 帳戶名稱)。
  - 針對 Outgoing claim type (傳出宣告類型)，選擇 Name ID (名稱 ID)。
  - 針對 Outgoing name ID format (傳出名稱 ID 格式)，選擇 Persistent Identifier (持久性標識符)。
  - 選取 Pass through all claim values (傳遞所有宣告值)。

### Configure Rule

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name:

Rule template: Transform an Incoming Claim

Incoming claim type:

Incoming name ID format:

Outgoing claim type:

Outgoing name ID format:

Pass through all claim values

Replace an incoming claim value with a different outgoing claim value

Incoming claim value:

Outgoing claim value:

Replace incoming e-mail suffix claims with a new e-mail suffix

New e-mail suffix:

Example: fabrikam.com

6. 選擇 Add Rule (新增規則)，然後輸入下列資訊以建立宣告規則 2，然後選擇 Finish (完成)。

- 針對 Claim rule name (宣告規則名稱)，輸入 **RoleSessionName**。
- 針對 Rule template (規則範本)，使用 Send LDAP Attribute as Claims (將 LDAP 屬性傳送為宣告)。
- 針對 Attribute store (屬性存放區)，選擇 Active Directory。
- 針對 Mapping of LDAP attributes to outgoing claim types (將 LDAP 屬性映射至傳出宣告類型)，新增屬性 **E-Mail-Addresses**。針對 Outgoing Claim Type (傳出宣告類型)，輸入 **https://aws.amazon.com/SAML/Attributes/RoleSessionName**。

### Configure Rule

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Rule template: Send LDAP Attributes as Claims

Attribute store:

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	<input type="text" value="E-Mail-Addresses"/>	<input type="text" value="aws.amazon.com/SAML/Attributes/RoleSessionName"/>
*	<input type="text"/>	<input type="text"/>

< Previous   Finish   Cancel

7. 選擇 Add Rule (新增規則)，然後輸入下列資訊以建立宣告規則 3，然後選擇 Finish (完成)。

- 針對 Claim rule name (宣告規則名稱)，輸入 **Get AD Groups**。
- 針對 Rule template (規則範本)，使用 Send Claims Using a Custom Rule (使用自訂規則傳送宣告)。
- 針對 Custom rule (自訂規則)，輸入下列程式碼。

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
  Issuer == "AD AUTHORITY"]=> add(store = "Active Directory", types = ("http://temp/variable"),
  query = ";tokenGroups;{0}", param = c.Value);
```

### Configure Rule

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS claim rule language.

Claim rule name:  
Get AD Groups

Rule template: Send Claims Using a Custom Rule

Custom rule:

```
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccount
name", Issuer == "AD AUTHORITY"]
=> add(store = "Active Directory", types = ("http://temp/variable"),
query = ";tokenGroups;{0}", param = c.Value);
```

< Previous   Finish   Cancel

8. 選擇 Add Rule (新增規則)。輸入下列資訊以建立宣告規則 4，然後選擇 Finish (完成)。

- 針對 Claim rule name (宣告規則名稱)，輸入 **Role**。
- 針對 Rule template (規則範本)，使用 Send Claims Using a Custom Rule (使用自訂規則傳送宣告)。
- 針對 Custom rule (自訂規則)，輸入下列含有您先前建立的帳戶號碼和 SAML 提供者的名稱的程式碼：

```
c:[Type == "http://temp/variable", Value =~ "(?i)^aws-"]=> issue(Type = "https://
aws.amazon.com/SAML/Attributes/Role",
Value = RegExReplace(c.Value, "aws-", "arn:aws:iam::AWS_ACCOUNT_NUMBER:saml-
provider/adfs-saml-provider,arn:aws:iam:: AWS_ACCOUNT_NUMBER:role/"));
```

### Configure Rule

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS claim rule language.

Claim rule name:

Rule template: Send Claims Using a Custom Rule

Custom rule:

```
c:[Type == "http://temp/variable", Value =~ "(?i)^aws-"]
=> issue(Type = "https://aws.amazon.com/SAML/Attributes/Role", Value =
RegexReplace(c.Value, "aws-", "arn:aws:iam::123456789012:saml-
provider/adfs-saml-provider,arn:aws:iam::123456789012:role/");
```

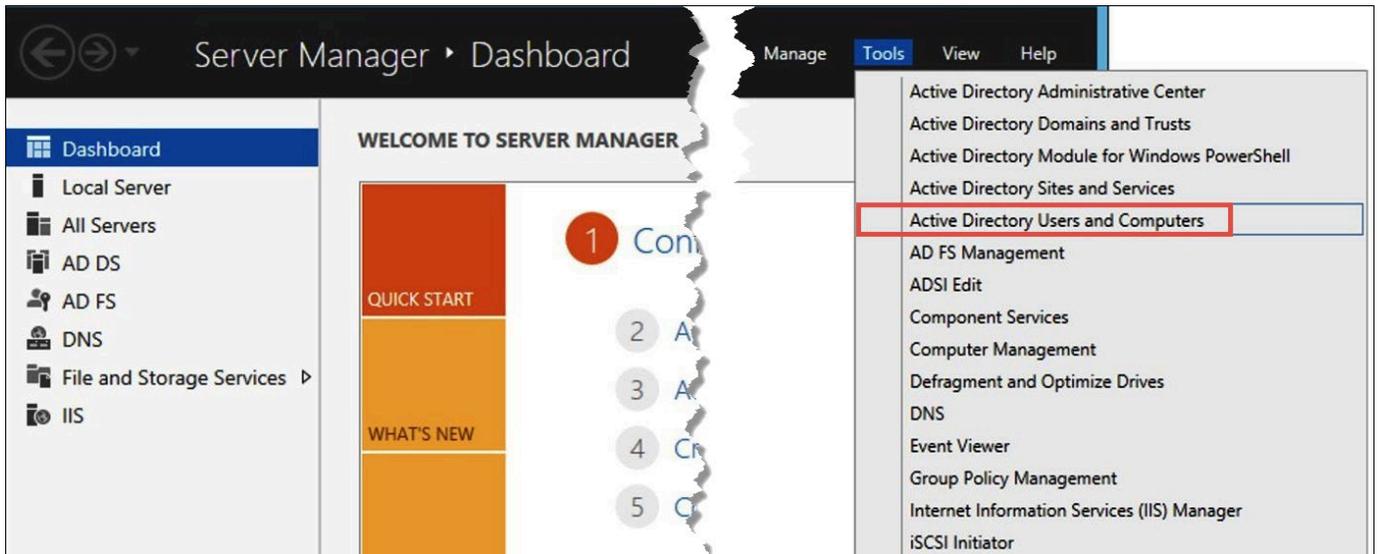
< Previous   Finish   Cancel

### 3. 建立 Active Directory 使用者和群組

現在，您已準備好建立可存取 Athena 的 AD 使用者，以及可將其放入的 AD 群組，以便您可以依群組控制存取層級。建立可分類資料存取模式的 AD 群組之後，您可以將使用者新增至這些群組。

若要建立 AD 使用者以存取 Athena

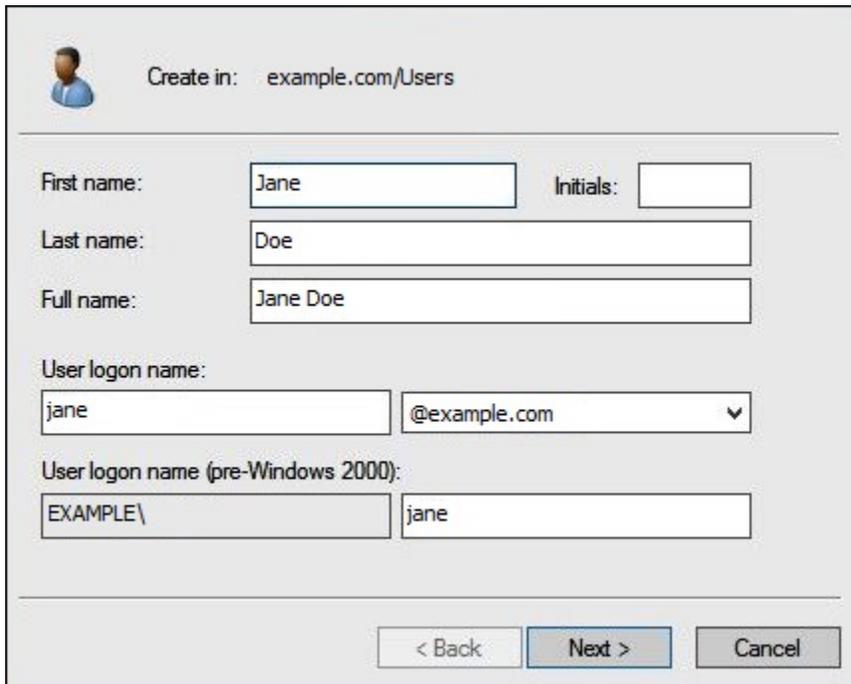
- 在 Server Manager Dashboard (伺服器管理員儀表板) 上，選擇 Tools (工具)，然後選擇 Active Directory Users and Computers (Active Directory 使用者和電腦)。



2. 在導覽窗格中，選擇 使用者。
3. 在 Active Directory Users and Computers (Active Directory 使用者和電腦) 工具列上，選擇 Create user (建立使用者) 選項。



4. 在 New Object – User (新建 – 使用者) 對話方塊中，對於 First name (名字)、Last name (姓氏) 和 Full name (全名)，輸入名稱。本教學課程使用的是 **Jane Doe**。



Create in: example.com/Users

First name: Jane Initials:

Last name: Doe

Full name: Jane Doe

User logon name: jane @example.com

User logon name (pre-Windows 2000): EXAMPLE\ jane

< Back Next > Cancel

5. 選擇 Next (下一步)。
6. 對於 Password (密碼)，輸入密碼，然後再次輸入以確認。

為了簡單起見，本教學課程取消選取 User must change password at next sign on (使用者必須在下次登入時變更密碼)。在真實世界的案例中，您應該要求新建立的使用者變更其密碼。



Create in: example.com/Users

Password:

Confirm password:

User must change password at next logon

User cannot change password

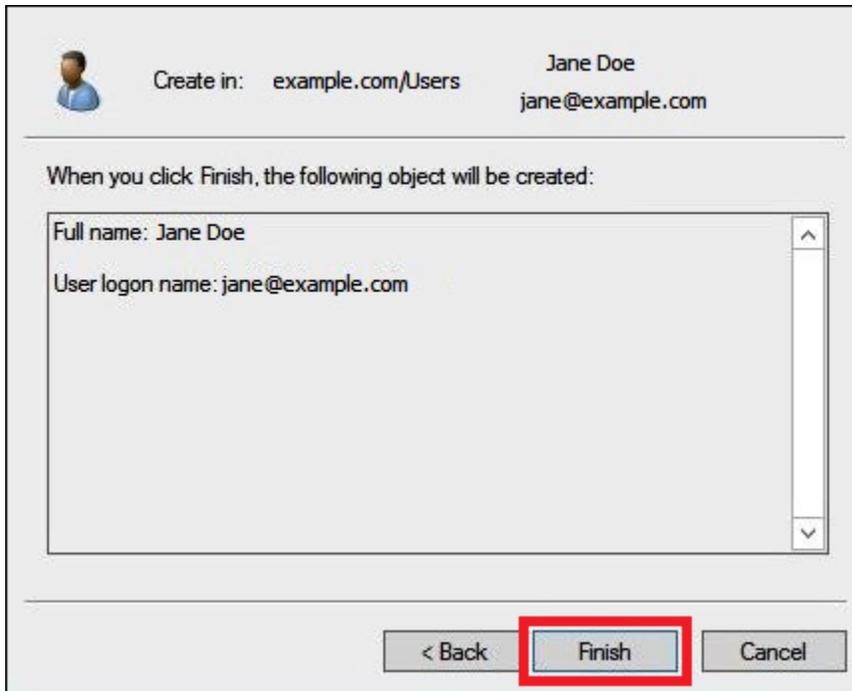
Password never expires

Account is disabled

< Back Next > Cancel

7. 選擇 Next (下一步)。

## 8. 選擇 Finish (完成)。



9. 在 Active Directory Users and Computers (Active Directory 使用者和電腦) 上，選擇使用者名稱。
10. 在使用者的 Properties (屬性) 對話方塊中，對於 E-mail (電子郵件)，輸入電子郵件地址。本教學課程使用的是 **jane@example.com**。

The image shows a user profile dialog box with the following fields and values:

Member Of	Dial-in	Environment	Sessions
Remote control	Remote Desktop Services Profile	COM+	
General	Address	Account	Profile
	Telephones	Organization	

**Jane Doe**

First name: Jane Initials:

Last name: Doe

Display name: Jane Doe

Description:

Office:

Telephone number:  Other...

E-mail: jane@example.com

Web page:  Other...

Buttons: OK, Cancel, Apply, Help

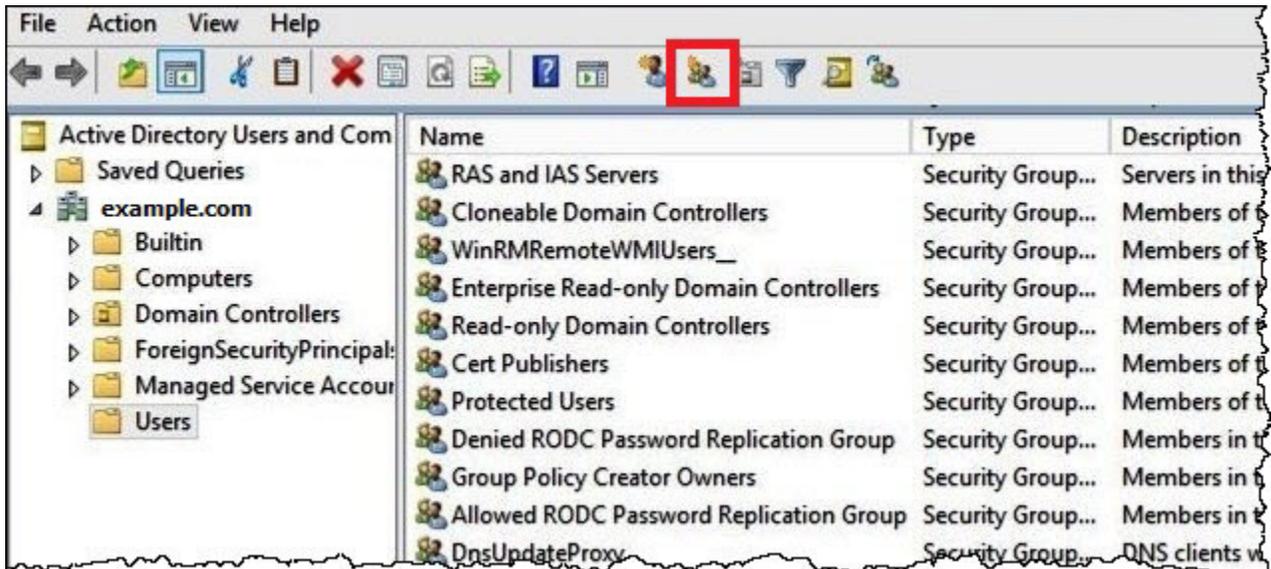
11. 選擇 OK (確定)。

### 建立 AD 群組，以代表資料存取模式

您可以建立 AD 群組，其成員在登入 AWS 時可擔任 `adfs-data-access` IAM 角色。下列範例會建立一個名為 `aws-adfs-data-access` 的 AD 群組。

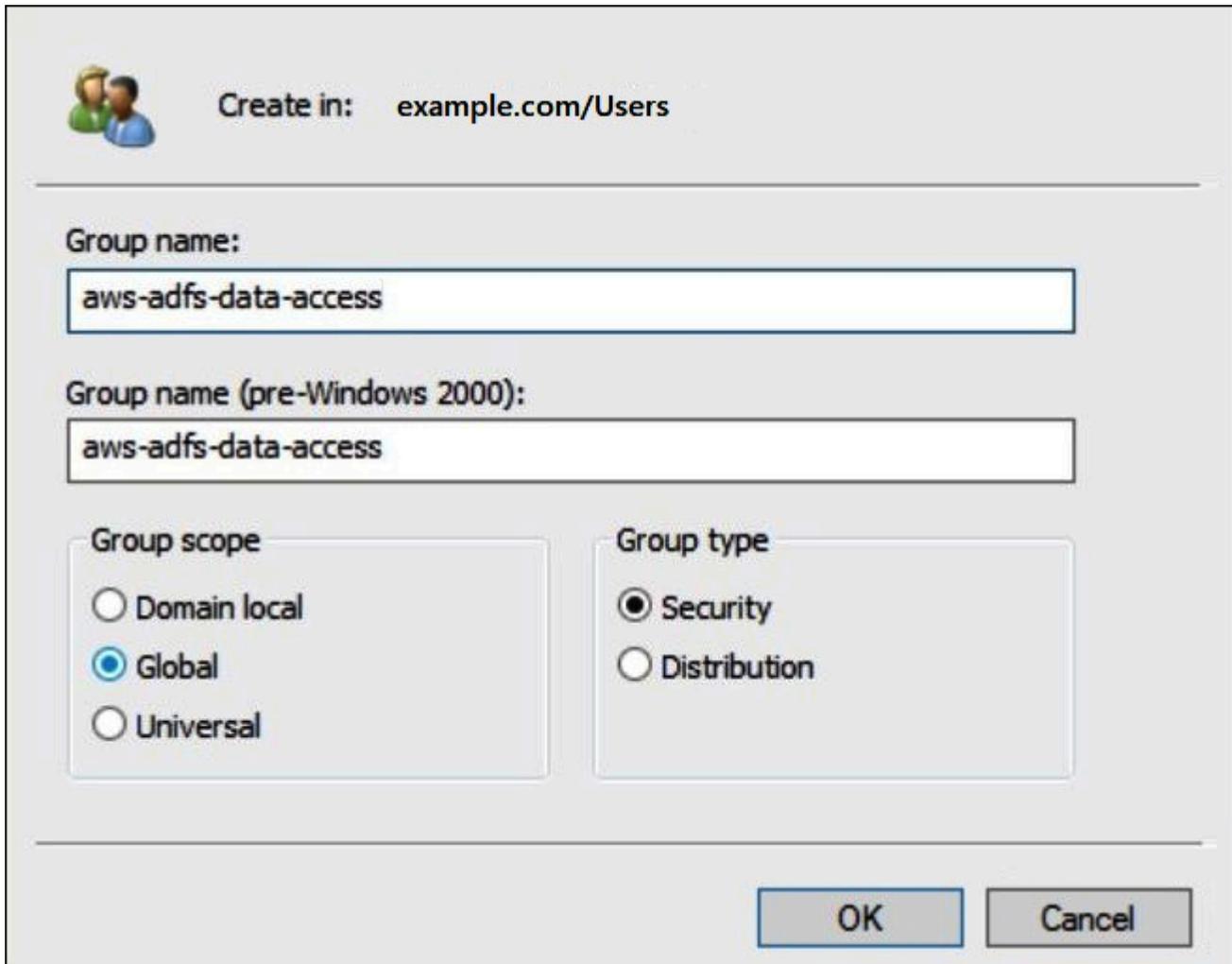
#### 若要建立 AD 群組

1. 在 Server Manager Dashboard (伺服器管理員儀表板) 上，從 Tools (工具) 選單上，選擇 Active Directory Users and Computers (Active Directory 使用者和電腦)。
2. 在工具列上，選擇 Create new group (建立新的群組) 選項。



3. 在 New Object - Group (新增物件 - 群組) 對話方塊中，輸入下列資訊：

- 針對 Group name (群組名稱)，輸入 **aws-adfs-data-access**。
- 針對 Group scope (群組範圍)，選取 Global (全域)。
- 針對 Group type (群組類型)，選取 Security (安全)。



Create in: example.com/Users

Group name:  
aws-adfs-data-access

Group name (pre-Windows 2000):  
aws-adfs-data-access

Group scope

- Domain local
- Global
- Universal

Group type

- Security
- Distribution

OK Cancel

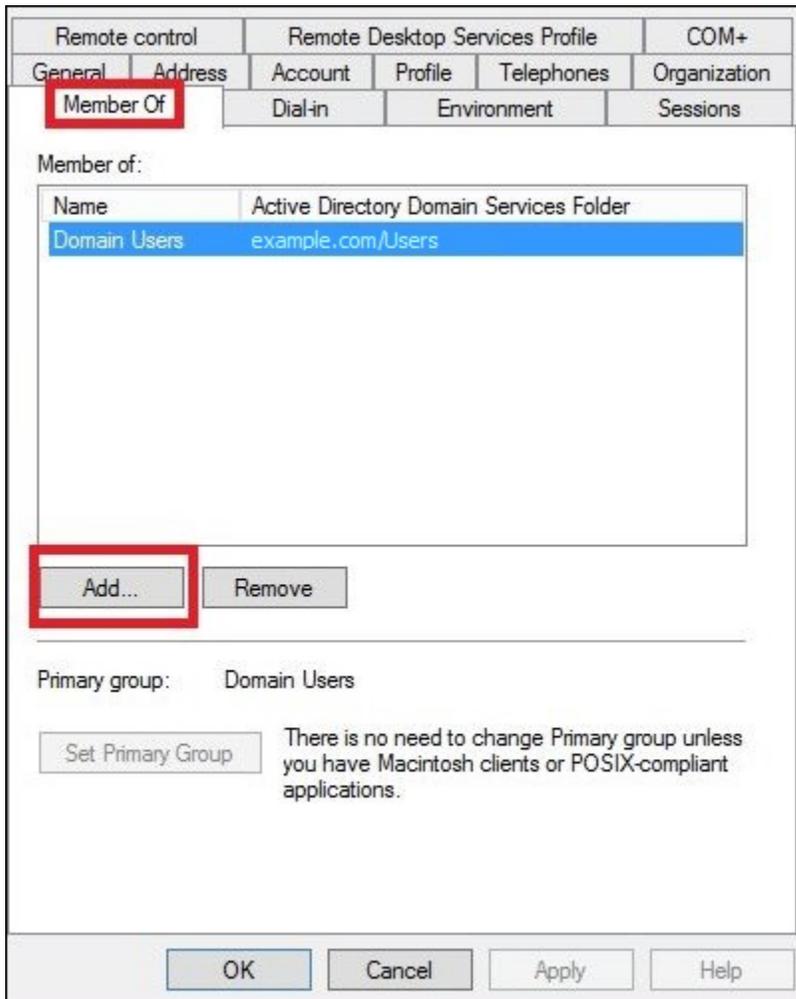
4. 選擇 OK (確定)。

將 AD 使用者新增至相應群組

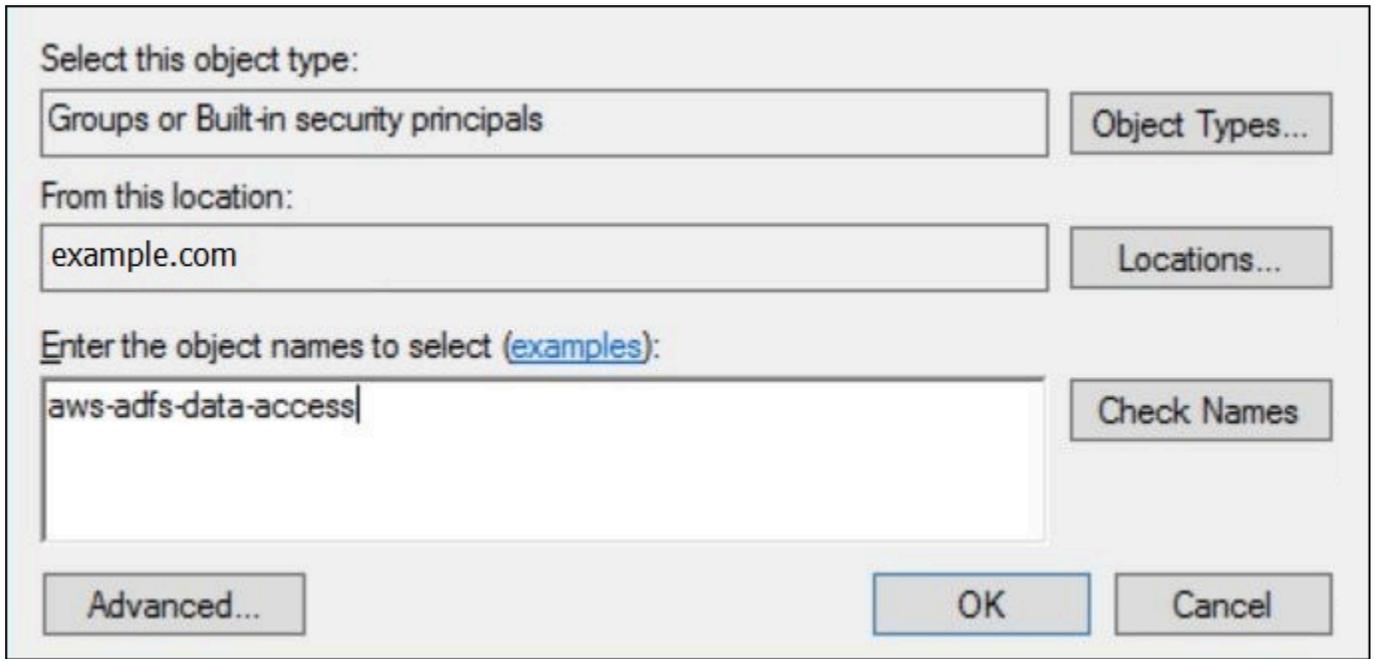
既然您已建立 AD 使用者和 AD 群組，您可以將使用者新增至群組。

若要將 AD 使用者新增至 AD 群組

1. 在 Server Manager Dashboard (伺服器管理員儀表板) 上的 Tools (工具) 選單上，選擇 Active Directory Users and Computers (Active Directory 使用者和電腦)。
2. 針對 First name (名字) 和 Last name (姓氏)，選擇使用者 (例如，Jane Doe)。
3. 在使用者的 Properties (屬性) 對話框中的 Member Of (成員群組) 標籤上，選擇 Add (新增)。



4. 根據您的需求新增一或多個 AD FS 群組。本教學課程新增了 `aws-ads-data-access` 群組。
5. 在 Select Groups (選取群組) 對話方塊中，針對 Enter the object names to select (輸入要選取的物件名稱)，輸入您建立的 AD FS 群組的名稱 (例如，**`aws-ads-data-access`**)，然後選擇 Check Names (檢查名稱)。

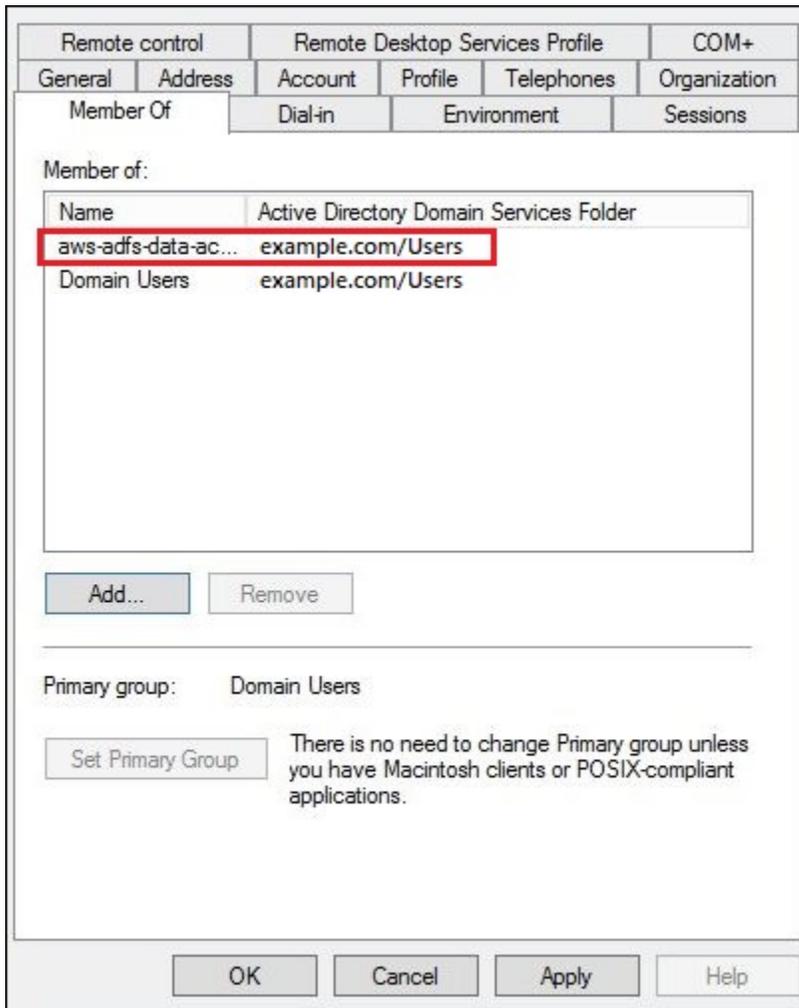


The screenshot shows a dialog box with the following elements:

- Select this object type:** A text box containing "Groups or Built-in security principals" and a button labeled "Object Types...".
- From this location:** A text box containing "example.com" and a button labeled "Locations...".
- Enter the object names to select (examples):** A text box containing "aws-ads-data-access" and a button labeled "Check Names".
- At the bottom, there are three buttons: "Advanced...", "OK", and "Cancel".

6. 選擇 OK (確定)。

在使用者的 Properties (屬性) 對話方塊中，AD 群組的名稱會顯示在 Member of (成員群組) 清單中。



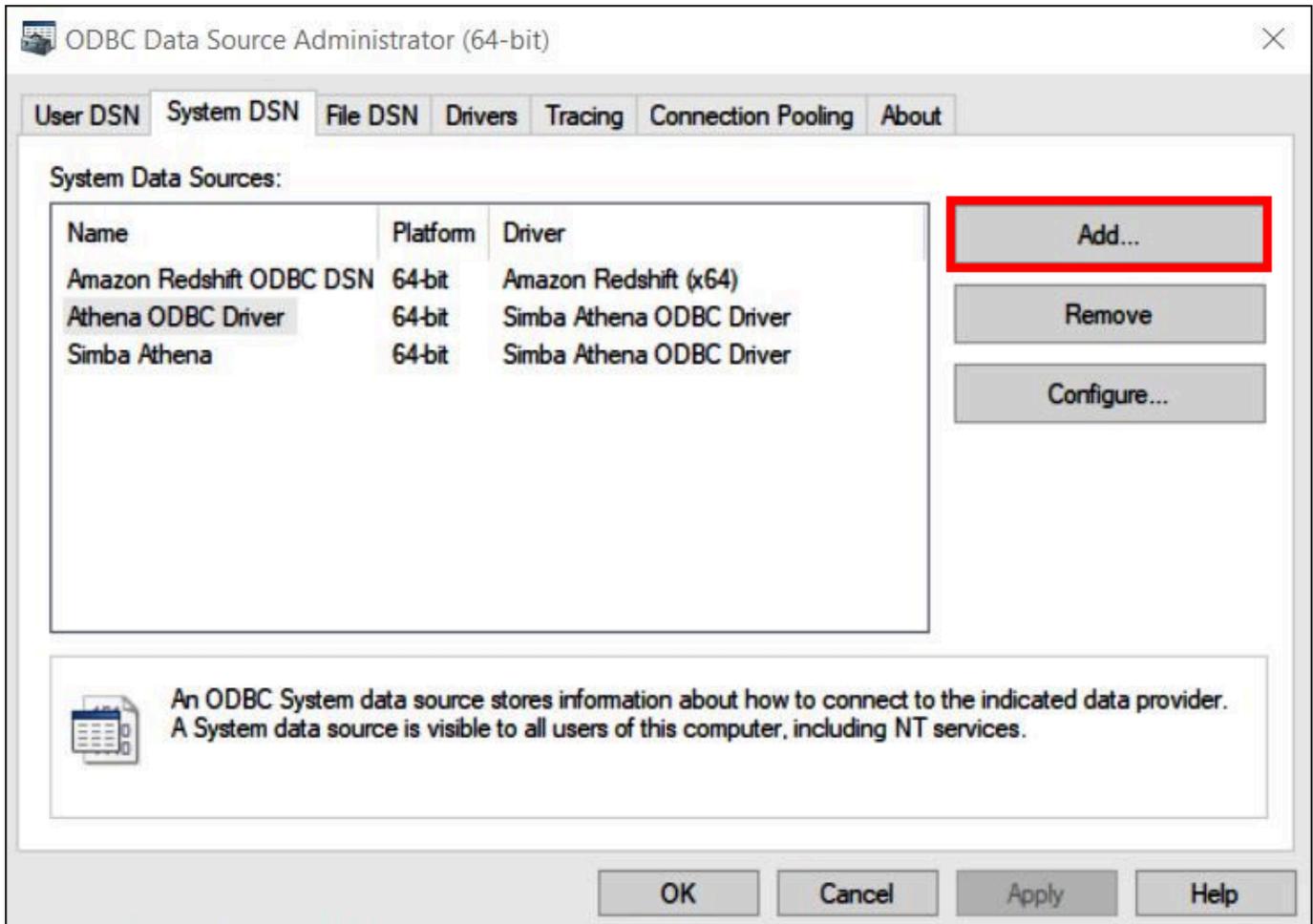
7. 選擇 Apply (套用)，然後選擇 OK (確定)。

#### 4. 設定 AD FS ODBC 與 Athena 之間的連線

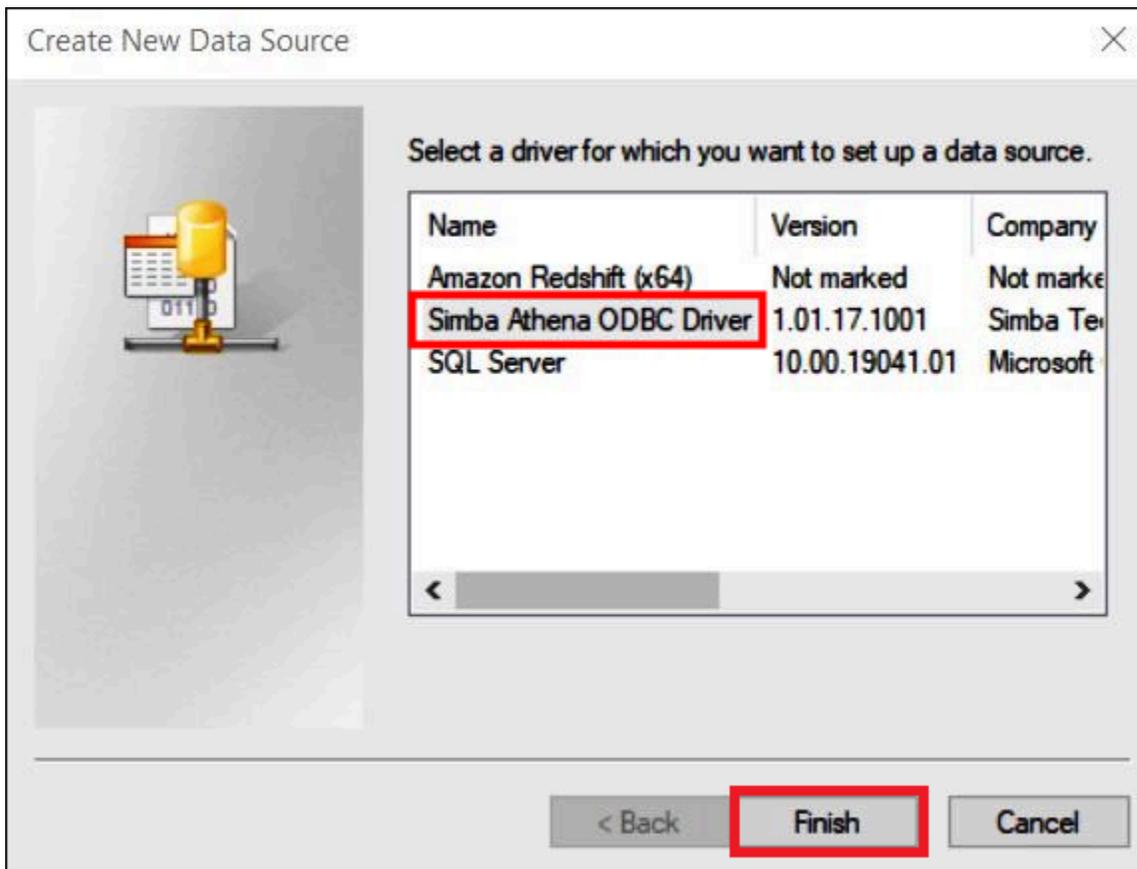
建立 AD 使用者和群組之後，您便準備好使用 Windows 中的 ODBC 資料來源程式來設定 Athena ODBC 與 AD FS 的連線。

若要設定 AD FS ODBC 與 Athena 之間的連線

1. 安裝適用於 Athena 的 ODBC 驅動程式。如需下載連結，請參閱[使用 ODBC 連接到 Amazon Athena](#)。
2. 在 Windows 中，選擇 Start (開始)，ODBC Data Sources (ODBC 資料來源)。
3. 在 ODBC Data Source Administrator (ODBC 資料來源管理員) 程式中，選擇 Add (新增)。



4. 在 Create New Data Source (建立新的資料來源) 對話方塊中，選擇 Simba Athena ODBC Driver (Simba Athena ODBC 驅動程式)，然後選擇 Finish (完成)。



5. 在 Simba Athena ODBC Driver DSN Setup (Simba Athena ODBC 驅動程式 DSN 設定) 對話方塊中，輸入以下值：
  - 在 Data Source Name (資料來源名稱)，為您的資料來源輸入名稱，例如 **Athena-odbc-test**。
  - 在 Description (描述) 輸入對資料來源的描述。
  - AWS 區域：輸入您正在使用的 AWS 區域 (例如 **us-west-1**)。
  - S3 Output Location (S3 輸出位置)：輸入您要存放輸出的 Simple Storage Service (Amazon S3) 路徑。

The screenshot shows the 'Simba Athena ODBC Driver DSN Setup' dialog box. The fields are filled with the following values:

- Data Source Name: Athena-odbc-test
- Description: (empty)
- AWS Region: us-west-1
- Catalog: AwsDataCatalog
- Schema: default
- Workgroup: odbc-test-group
- Metadata Retrieval Method: Auto
- Output Options:
  - S3 Output Location: s3://DOC-EXAMPLE-BUCKET/
  - Encryption Options: NOT\_SET
  - KMS Key: (empty)
- Endpoint Override: (empty)
- Streaming Endpoint Override: (empty)

The 'Authentication Options...' button is highlighted with a red box. Other buttons include 'Advanced Options...', 'Logging Options...', 'Proxy Options...', 'Test...', 'OK', and 'Cancel'. The version 'v1.1.17.1001 (64 bit)' is displayed at the bottom left.

6. 選擇 Authentication Options (身分驗證選項)。
7. 在 Authentication Options (身分驗證選項) 對話方塊中，指定以下值：
  - 在 Authentication Type (身分驗證類型) 中，選擇 ADFS。
  - 在 User (使用者) 中，輸入使用者的電子郵件地址 (例如，**jane@example.com**)。
  - 在 Password (密碼) 中，輸入使用者的 ADFS 密碼。
  - 在 IdP Host (IdP 主機) 中，輸入 AD FS 伺服器名稱 (例如，**ads.example.com**)。
  - 在 IdP Port (IdP 主機) 中，使用預設值 443。
  - 選取 SSL Insecure (SSL 不安全) 選項。

Authentication Type: ADFS

User: jane@example.com

Password: [masked]

Password Options...

Session Token:

Preferred Role:

Session Duration:

IdP Host: adfs.example.com

IdP Port: 443

Use HTTP Proxy For IdP Host       SSL Insecure

OK      Cancel

8. 選擇 OK (確定)，以關閉 Authentication Options (身分驗證選項)。
9. 選擇 Test (測試) 以測試連接，或 OK (確定) 以完成操作。

## 使用 Okta 外掛程式和 Okta 身分提供者設定 ODBC 的 SSO

本頁面說明如何設定 Amazon Athena ODBC 驅動程式和 Okta 外掛程式，以使用 Okta 身分提供者新增單一登入 (SSO) 功能。

### 必要條件

完成本教學課程中的步驟需要以下項目：

- Amazon Athena ODBC 驅動程式。如需下載連結，請查看[使用 ODBC 連接到 Amazon Athena](#)。
- 您要搭配 SAML 使用的 IAM 角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[為 SAML 2.0 聯合身分建立角色](#)。
- Okta 帳戶。如需詳細資訊，請前往[Okta.com](#)。

### 在 Okta 中建立應用程式整合作業

首先，使用 Okta 儀表板建立及設定 SAML 2.0 應用程式，以透過單一登入機制存取 Athena。您可以使用 Okta 中現有的 Redshift 應用程式，設定對 Athena 的存取權。

### 在 Okta 中建立應用程式整合作業

1. 在 [Okta.com](#) 上登入您帳戶的管理頁面。
2. 在導覽窗格中，依序選擇 Applications (應用程式) 和 Applications (應用程式)。
3. 在 Applications (應用程式) 頁面上，選擇 Browse App Catalog (瀏覽應用程式目錄)。
4. 在 Browse App Integration Catalog (瀏覽應用程式整合目錄) 頁面的 Use Case (使用案例) 區段中，選擇 All Integrations (所有整合)。
5. 在搜尋方塊中輸入 Amazon Web Services Redshift，然後選擇 Amazon Web Services Redshift SAML。
6. 選擇 Add Integration (新增整合作業)。

The screenshot displays the Okta Admin console interface. On the left is a navigation sidebar with categories: Dashboard, Directory, Customizations, Applications, Security, Workflow, Reports, and Settings. The 'Applications' section is expanded, showing 'Applications', 'Self Service', and 'Settings'. The main content area shows a breadcrumb trail: Applications > Catalog > Single Sign-On > Amazon Web Services Redshift. Below the breadcrumb, it says 'Last updated: August 27, 2019'. A blue button with a plus icon and the text 'Add Integration' is highlighted with a red rectangular box. Below this, there is a card for 'Amazon Web Services Redshift' with the AWS logo and a 'SAML' tag. At the bottom, there are two sections: 'Okta Verified' and 'Overview'. The 'Okta Verified' section states: 'The integration was either created by Okta or by'. The 'Overview' section states: 'Okta's integration with Amazon Web Services (AWS) Redshift allows end users to authenticate to AWS'.

7. 在 General Settings Required (必要的一般設定) 區段中，針對 Application (應用程式) 標籤輸入應用程式的名稱。本教學課程使用的名稱為 Athena-ODBC-Okta。

# Add Amazon Web Services Redshift

**1** General Settings

## General settings- Required

Application label

This label displays under the app on your home page

Application Visibility

- Do not display application icon to users
- Do not display application icon in the Okta Mobile App

[Cancel](#) [Done](#)

- 選擇 Done (完成)。
- 在 Okta 應用程式的頁面上 (例如 Athena-ODBC-Okta) , 選擇 Sign On (登入)。

← Back to Applications



# Athena-ODBC-Okta

**Active**  [View Logs](#) [Monitor Imports](#)

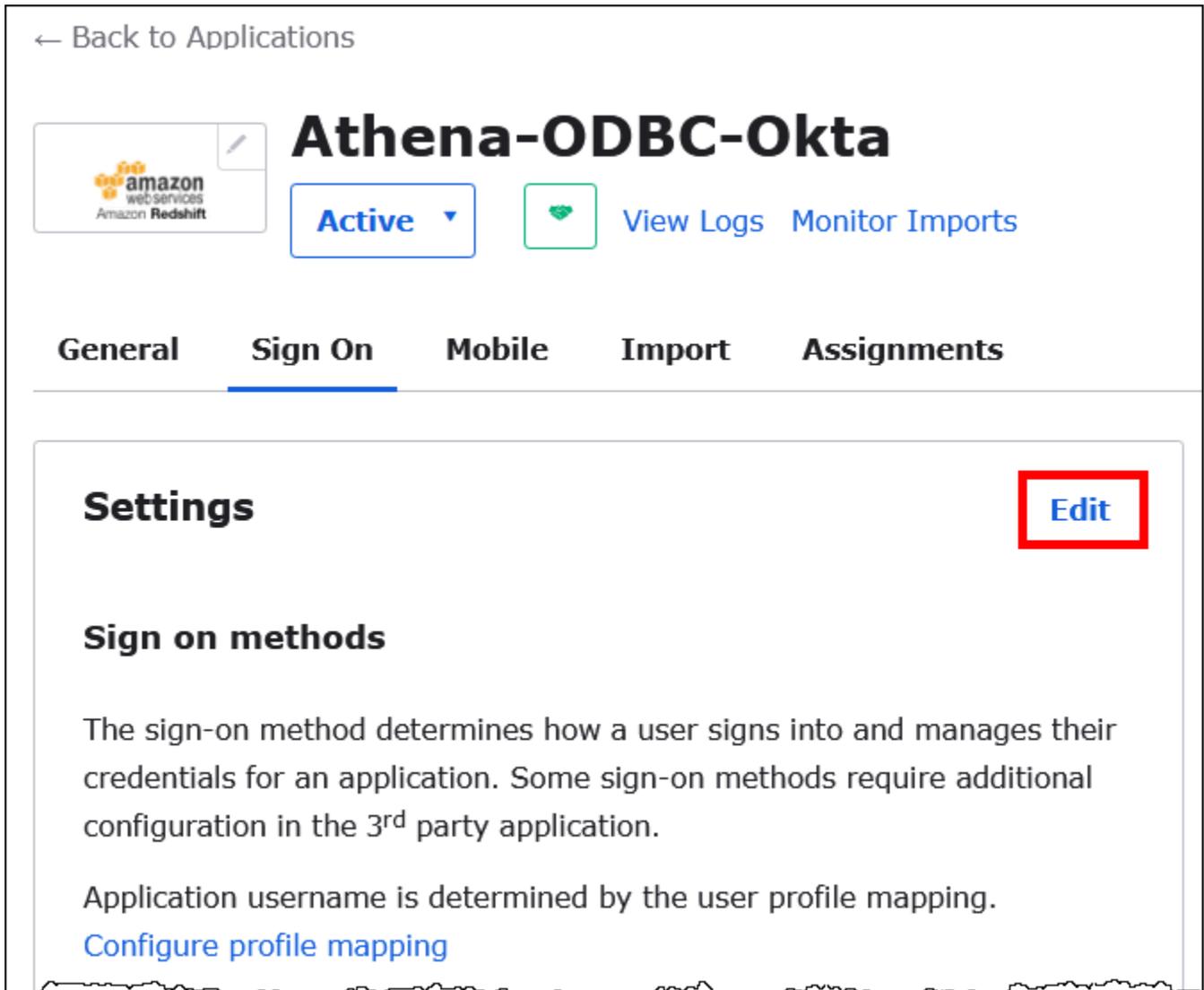
**General** **Sign On** **Mobile** **Import** **Assignments**

**Assign** **Convert assignments**

Search... **People**

Filters	Person	Type
People		
Groups		
		01101110
		01101111
		01110100
		01101000
		01101001
		01101110
		01100111
		No users found

10. 在 Settings (設定) 區段中，選擇 Edit (編輯)。



← Back to Applications

 **Athena-ODBC-Okta**

Active  [View Logs](#) [Monitor Imports](#)

**General** **Sign On** **Mobile** **Import** **Assignments**

**Settings** [Edit](#)

**Sign on methods**

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3<sup>rd</sup> party application.

Application username is determined by the user profile mapping.  
[Configure profile mapping](#)

11. 在 Advanced Sign-on Settings (進階登入設定) 區段設定下列值。

- 對於 IdP ARN 和角色 ARN，請以逗號分隔的值輸入您的 AWS IDP ARN 和角色 ARN。如需 IAM 角色格式的相關資訊，請參閱 [《IAM 使用者指南》](#) 的為身分驗證回應設定 SAML 聲明。
- 在 Session Duration (工作階段持續時間) 輸入 900 秒到 43200 秒之間的值。本教學課程使用預設值 3600 (亦即 1 小時)。

## Advanced Sign-on Settings

These fields may be required for a Amazon Web Services Redshift proprietary sign-on option or general setting.

Idp ARN and Role ARN

arn:aws:iam::1234567890:saml-provid

Enter your AWS IDP ARN and Role ARN as comma separated values (e.g. "arn:aws:iam::1234567890:saml-provider/OKTA,arn:aws:iam::1234567890:role/SAML\_ROLE").

Session Duration

3600

Set the user's session duration in seconds here.

Valid range is 900 to 43200.

DB User Format (Redshift)

\${user.username}

EL expression to get DB User value (e.g. "\${user.username}", "\${user.firstName}\${user.lastName}@acme.com")

Auto Create (Redshift)



AutoCreate Redshift property (Create a new database user if one does not exist)

Allowed DB Groups (Redshift)

Comma separated list of allowed user groups. Use "\*" to allow all groups, "\" to escape comma in group name

Athena 不會使用 [DbUser 格式] 和 [允許的資料庫群組] 設定。AutoCreate 因此您不必設定這些欄位。

## 12. 選擇儲存。

### 從 Okta 擷取 ODBC 組態資訊

既然您已建立 Okta 應用程式，您可以準備擷取應用程式的 ID 和 IdP 主機 URL。稍後設定 ODBC 以連線至 Athena 時，您會需要這些資訊。

### 從 Okta 擷取 ODBC 組態資訊

1. 選擇 Okta 應用程式的 General (一般) 標籤，接著向下捲動至 App Embed Link (應用程式內嵌連結) 區段。

**Athena-ODBC-Okta**

Active View Logs Monitor Imports

**General** Sign On Mobile Import Assignments

**App Settings** Edit

**App Embed Link** Edit

**Embed Link**

You can use the URL below to sign into Amazon Web Services Redshift from a portal or other location outside of Okta.

`https://[redacted].okta.com/home/amazon_aws_redshift/[redacted]/[redacted]`

Embed Link (內嵌連結) URL 的格式如下：

```
https://trial-1234567.okta.com/home/amazon_aws_redshift/Abc1de2fghi3J45kL678/abc1defghij2klmNo3p4
```

2. 從您的 Embed Link (內嵌連結) URL 擷取及儲存以下部分：

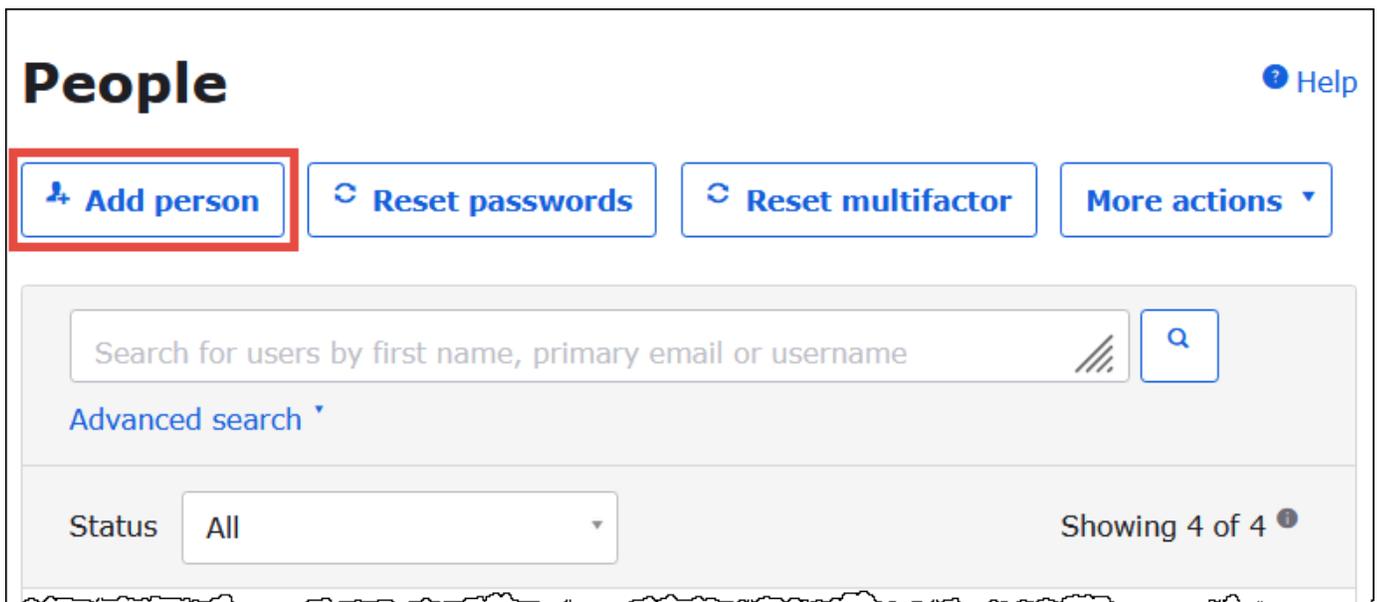
- 第一個區段中 `https://` 之後到 `okta.com` (含) 的內容 (例如 `trial-1234567.okta.com`)。這是您的 IdP 主機。
- URL 的最後兩小段，包括中間的正斜線。這些區段是兩個由 20 個字元組成的字串，其中包含數字和大小寫字母的混合 (例如 `Abc1de2fghi3J45kL678/abc1defghij2klmNo3p4`)。這是您的應用程式 ID。

將使用者新增至 Okta 應用程式

現在您可以準備將使用者新增至 Okta 應用程式。

將使用者新增至 Okta 應用程式

1. 在左側導覽窗格中，選擇 Directory (目錄)，然後選擇 People (人員)。
2. 選擇 Add Person (新增人員)。



3. 在 Add Person (新增人員) 對話方塊中輸入下列資訊。
  - 輸入 First name (名字) 和 Last name (姓氏) 的值。本教學課程使用的是 **test user**。
  - 輸入 Username (使用者名稱) 和 Primary email (主要電子郵件) 的值。對於這兩個欄位，本教學課程皆使用 **test@amazon.com**。您對密碼的安全要求可能不盡相同。

## Add Person

User type <sup>?</sup>

First name

Last name

Username

Primary email

Secondary email (optional)

Groups (optional)

Password <sup>?</sup>

Send user activation email now <sup>?</sup>

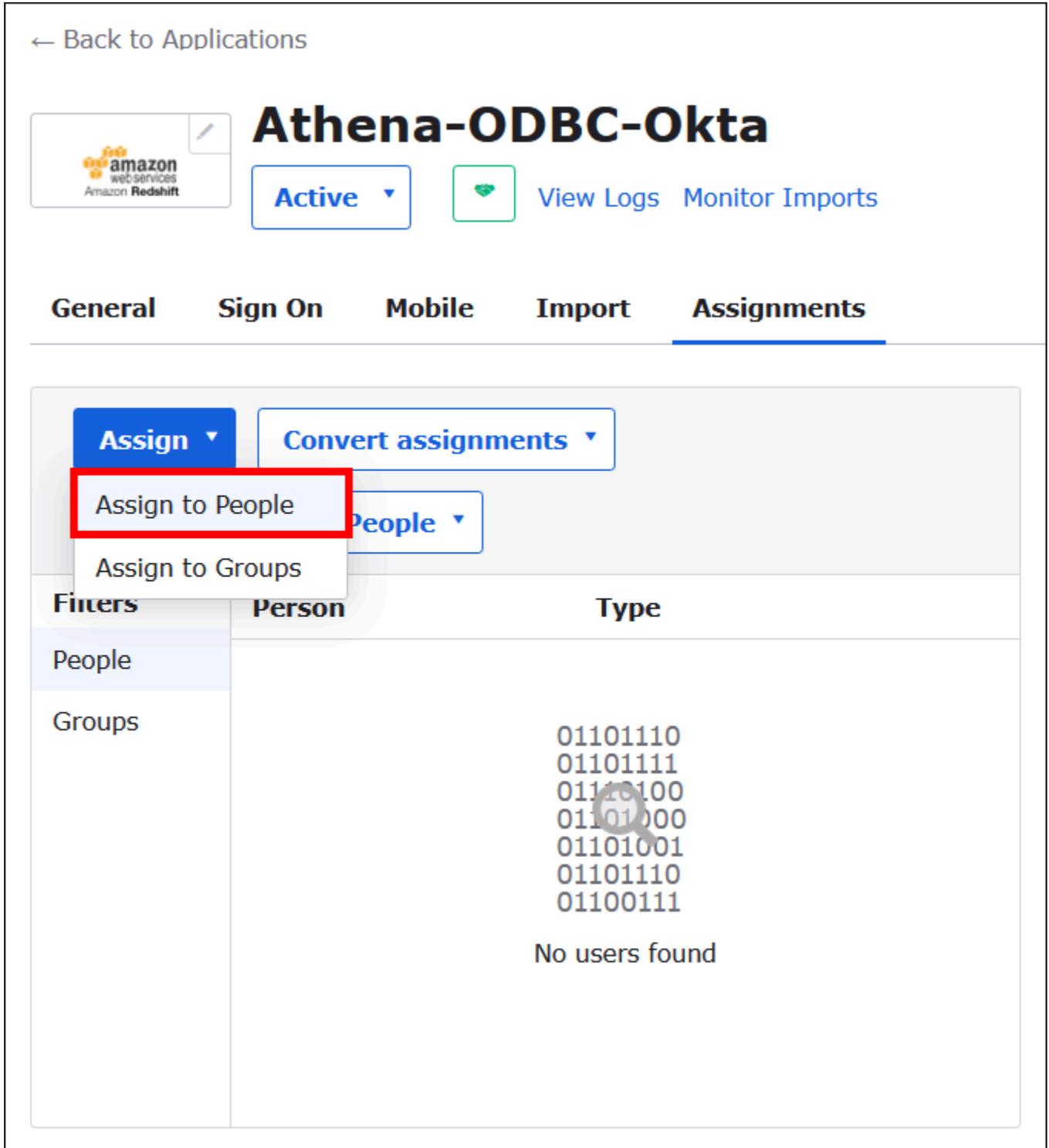
**Save** **Save and Add Another** **Cancel**

#### 4. 選擇儲存。

現在，您可以準備將所建立的使用者指派給您的應用程式。

## 將使用者指派給應用程式

1. 在導覽窗格中，依序選擇 Applications (應用程式) 和 Applications (應用程式)，然後選擇您應用程式的名稱 (例如 Athena-ODBC-Okta)。
2. 選擇 Assign (指派)，然後選擇 Assign to People (指派給人員)。



← Back to Applications

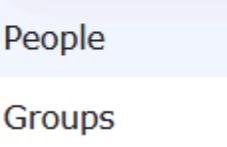
 **Athena-ODBC-Okta**

Active  View Logs  Monitor Imports

General Sign On Mobile Import **Assignments**

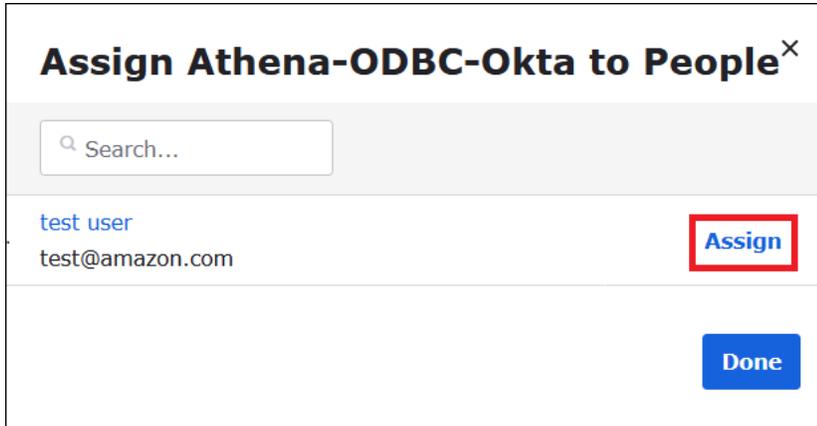
Assign 

Assign to People 

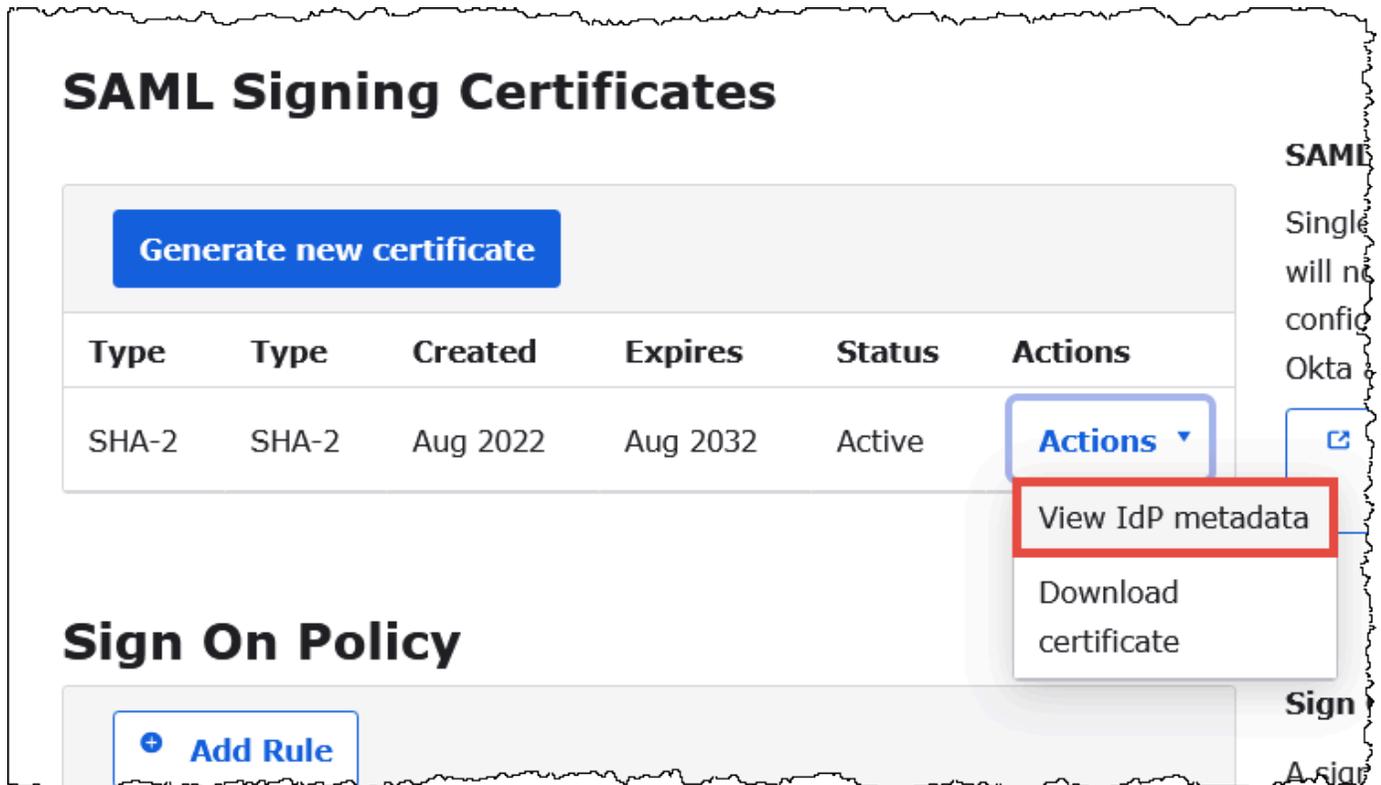
People 

Filters	Person	Type
People		01101110
		01101111
		01101100
		01101000
		01101001
		01101110
		01100111
		No users found

3. 為您的使用者選擇 Assign (指派) 選項，然後選擇 Done (完成)。



4. 畫面出現提示時，選擇 Save and Go Back (儲存並返回)。對話方塊會顯示使用者的狀態為 Assigned (已指派)。
5. 選擇完成。
6. 選擇 Sign On (登入) 標籤。
7. 向下捲動至 SAML Signing Certificates (SAML 簽署憑證) 區段。
8. 選擇 Actions (動作)。
9. 開啟內容功能表 (按一下滑鼠右鍵)，選擇 View IdP metadata (檢視 IdP 中繼資料)，然後選擇瀏覽器選項以儲存檔案。
10. 以 .xml 為副檔名儲存檔案。

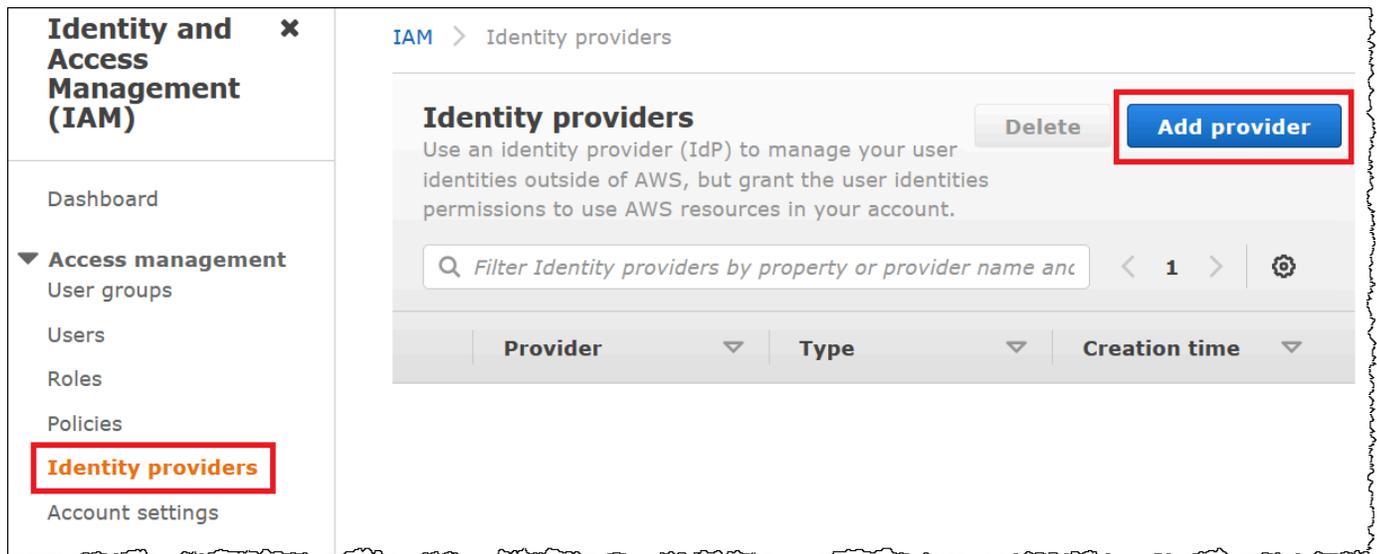


## 建立 AWS SAML 身分識別提供者和角色

現在，您已準備好將中繼資料 XML 檔案上傳到中的 IAM 主控台 AWS。您將使用此檔案來建立 AWS SAML 身分識別提供者和角色。請使用 AWS 服務管理員帳戶執行這些步驟。

若要在中建立 SAML 身分識別提供者和角色 AWS

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <https://console.aws.amazon.com/IAM/>。
2. 在導覽窗格中，選擇 Identity providers (身分提供者)，然後選擇 Add provider (新增提供者)。



3. 在 Add an Identity provider (新增身分提供者) 頁面的 Configure provider (設定提供者) 位置，輸入下列資訊。
  - 在 Provider type (提供者類型) 選擇 SAML。
  - 在 Provider name (提供者名稱) 輸入提供者的名稱，例如 **AthenaODBCOkta**。
  - 在 Metadata document (中繼資料文件) 使用 Choose file (選擇檔案) 選項，上傳您所下載的身分提供者 (IdP) 中繼資料 XML 檔案。

# Add an Identity provider

## Configure provider

Provider type

**SAML**  
Establish trust between your AWS account and a SAML 2.0 compatible Identity Provider such as Shibboleth or Active Directory Federation Services.

**OpenID Connect**  
Establish trust between your AWS account and an Identity Provider such as Google or Salesforce.

Provider name  
Enter a meaningful name to identify this provider

AthenaODBCOkta

Maximum 128 characters. Use alphanumeric or '.', '\_' characters.

Metadata document  
This document is issued by your IdP.

 **Choose file**

File needs to be a valid UTF-8 XML document.

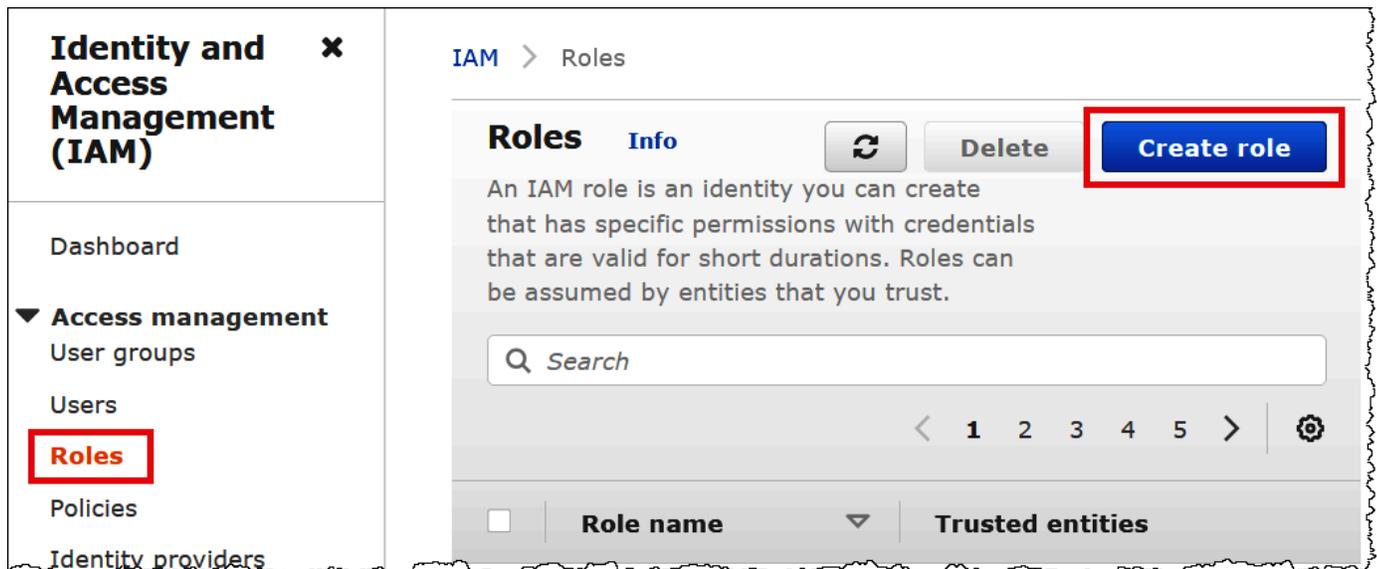
#### 4. 選擇 Add provider (新增提供者)。

為 Athena 和 Amazon S3 存取作業建立 IAM 角色

現在，您可以準備為 Athena 和 Amazon S3 存取作業建立 IAM 角色。您會將這個角色指派給您的使用者，如此一來，該使用者就能透過單一登入機制存取 Athena。

為使用者建立 IAM 角色

1. 在 IAM 主控台導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。



2. 在 Create role (建立角色) 頁面上選擇下列選項：

- 在 Select type of trusted entity (選取信任的實體類型) 選擇 SAML 2.0 Federation (SAML 2.0 聯合)。
- 在 SAML 2.0-based provider (SAML 2.0 提供者) 選擇您所建立的 SAML 身分提供者，例如 AthenaODBCOkta。
- 選取允許程式設計和 AWS Management Console 存取。

SAML 2.0 federation  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy  
Create a custom trust policy to enable others to perform actions in this account.

### SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

SAML 2.0-based provider

AthenaODBCOkta

Allow programmatic access only

Allow programmatic and AWS Management Console access

Attribute

SAML:aud

Value

https://signin.aws.amazon.com/saml

Condition - (optional)

3. 選擇下一步。
4. 在 Add Permissions (新增許可) 頁面上的 Filter policies (篩選政策) 位置，輸入 **AthenaFull**，然後按 Enter 鍵。
5. 選取 AmazonAthenaFullAccess 受管政策，然後選擇 Next (下一步)。

## Add permissions

**Permissions policies** (Selected 1/819)



Create policy

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press

1 match



1



"AthenaFull"

Clear filters



Policy name



Type



Description



AmazonAthenaFullAccess

AWS managed

Provide full access to

### ► Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

6. 在 Name, review, and create (命名、檢閱及建立) 頁面上的 Role name (角色名稱) 位置，輸入角色的名稱 (例如 **Athena-ODBC-OktaRole**)，然後選擇 Create role (建立角色)。

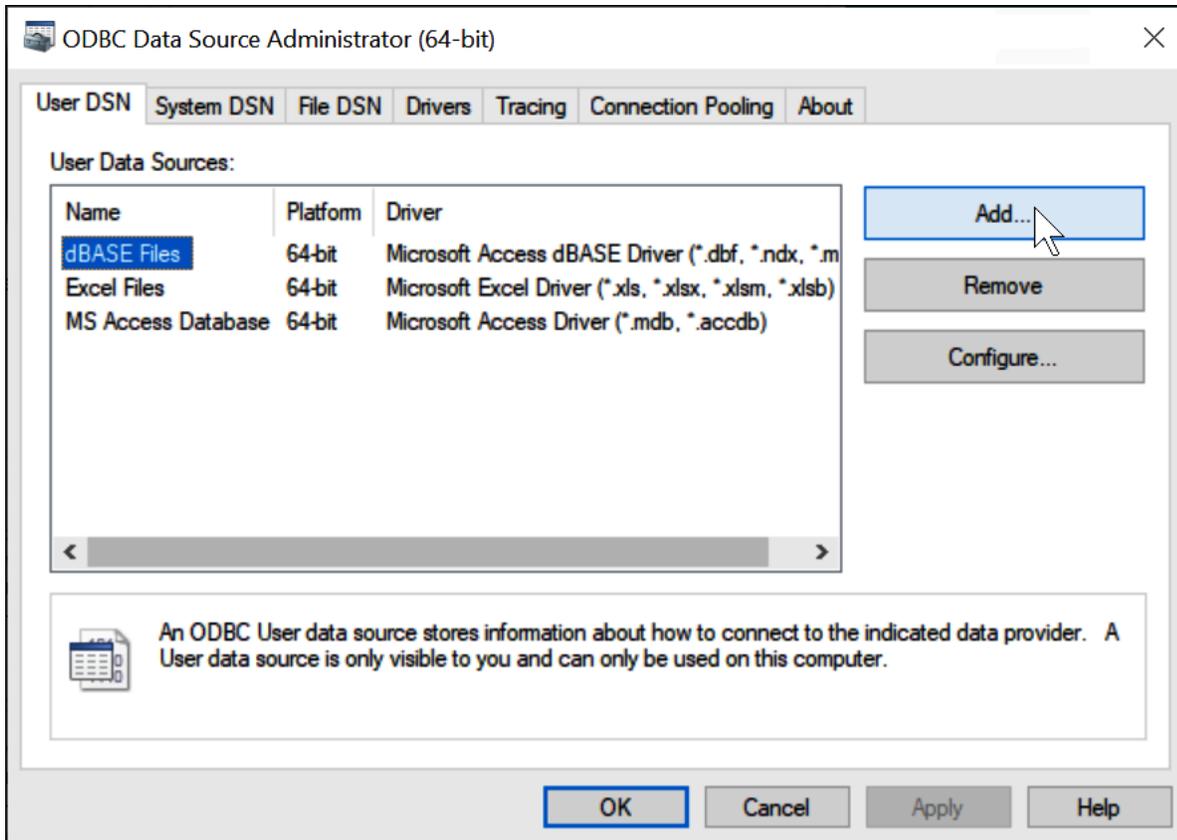
設定 Okta ODBC 與 Athena 之間的連線

現在，您可以準備使用 Windows 的 ODBC 資料來源程式，設定 Okta ODBC 與 Athena 的連線。

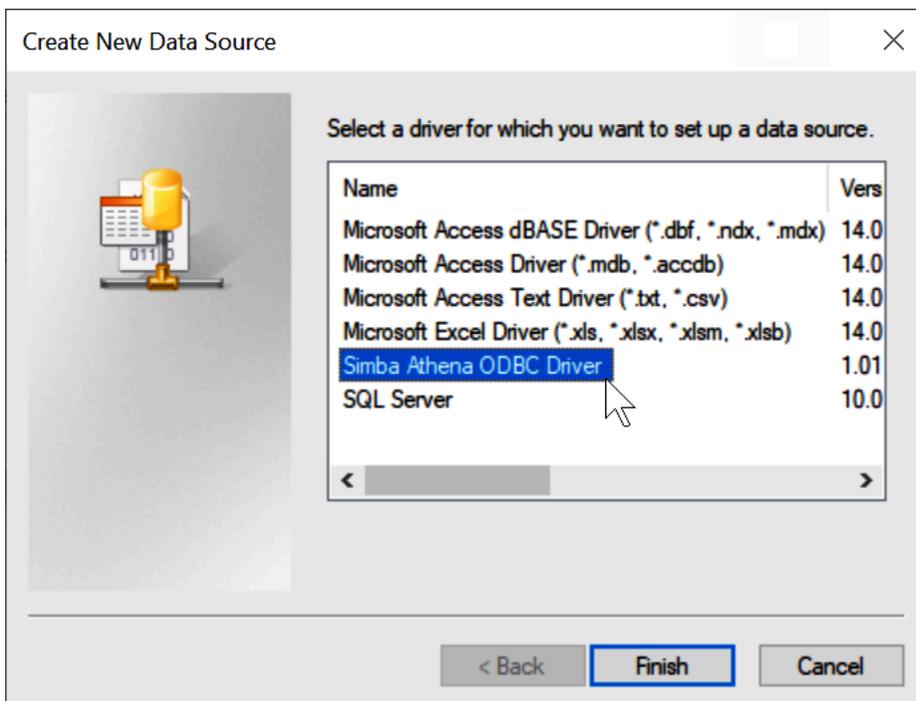
設定 Okta ODBC 與 Athena 的連線

1. 啟動 Windows 的 ODBC 資料來源程式。

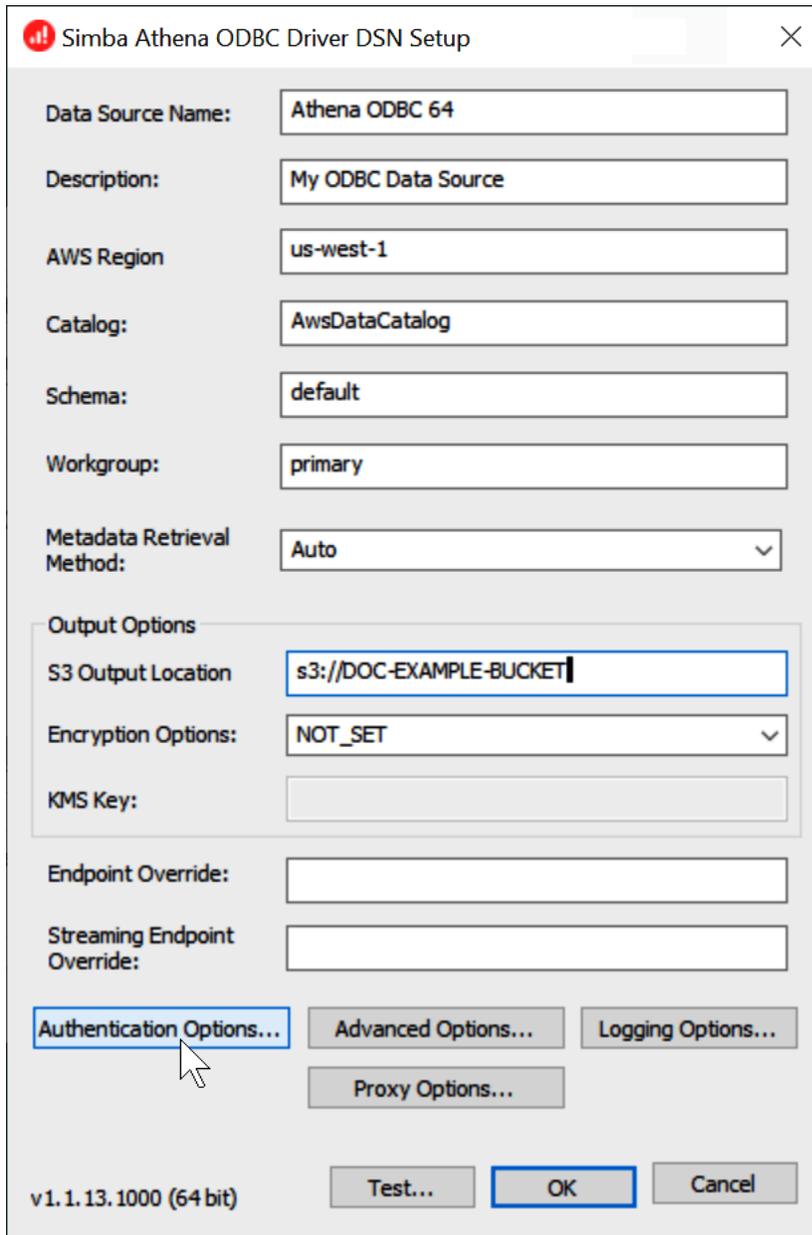
- 在 ODBC Data Source Administrator (ODBC 資料來源管理員) 程式中，選擇 Add (新增)。



- 選擇 Simba Athena ODBC Driver (Simba Athena ODBC 驅動程式)，然後選擇 Finish (完成)。



- 在 Simba Athena ODBC Driver DSN Setup (Simba Athena ODBC 驅動程式 DSN 設定) 對話方塊中，輸入所述的值。
  - 在 Data Source Name (資料來源名稱)，為您的資料來源輸入名稱，例如 **Athena ODBC 64**。
  - 在 Description (描述) 輸入對資料來源的描述。
  - 針對 AWS 區域，輸入您 AWS 區域正在使用的項目 (例如 **us-west-1**)。
  - 在 S3 Output Location (S3 輸出位置) 輸入您要存放輸出內容的 Amazon S3 路徑。



Simba Athena ODBC Driver DSN Setup

Data Source Name: Athena ODBC 64

Description: My ODBC Data Source

AWS Region: us-west-1

Catalog: AwsDataCatalog

Schema: default

Workgroup: primary

Metadata Retrieval Method: Auto

Output Options

S3 Output Location: s3://DOC-EXAMPLE-BUCKET

Encryption Options: NOT\_SET

KMS Key:

Endpoint Override:

Streaming Endpoint Override:

Authentication Options... Advanced Options... Logging Options...

Proxy Options...

v1.1.13.1000 (64 bit) Test... OK Cancel

- 選擇 Authentication Options (身分驗證選項)。
- 在 Authentication Options (身分驗證選項) 對話方塊中，選擇或輸入以下值。

- 在 Authentication Type (身分驗證類型) 選擇 Okta。
- 在 User (使用者) 輸入您的 Okta 使用者名稱。
- 在 Password (密碼) 輸入您的 Okta 密碼。
- 在 IdP Host (IdP 主機) 輸入您先前記下的值，例如 **trial-1234567.okta.com**。
- 在 IdP Port (IdP 連接埠) 輸入 **443**。
- 在 App ID (應用程式 ID) 輸入您先前記下的值 (Okta 內嵌連結的最後兩個區段)。
- 在 Okta App Name (Okta 應用程式名稱) 輸入 **amazon\_aws\_redshift**。

Authentication Options

Authentication Type: Okta

User: test@amazon.com

Password: ●●●●●●●●

Session Token:

Preferred Role:

Session Duration:

IdP Host: trial-...okta.com

IdP Port: 443

App ID:

Okta App Name: amazon\_aws\_redshift

Okta MFA wait time:

Okta MFA Type:

Okta MFA Phone No:

Use HTTP Proxy For IdP Host       SSL Insecure

OK      Cancel

7. 選擇 OK (確定)。
8. 選擇 Test (測試) 來測試連線，或選擇 OK (確定) 完成操作。

## 使用 ODBC、SAML 2.0 和 Okta 身分提供者設定單一登入

若要連線到資料來源，您可以使用 Amazon Athena 搭配身分供應商 (IdPs) PingOne，例如 Okta 等。OneLogin 從 Athena ODBC 驅動程式 1.1.13 版和 Athena JDBC 驅動程式 2.0.25 版開始，包含瀏覽器 SAML 外掛程式，您可以將其設定為與任何 SAML 2.0 提供者一起使用。此主題說明如何設定 Amazon Athena ODBC 驅動程式和以瀏覽器為基礎的 SAML 外掛程式，以使用 Okta 身分提供者新增單一登入 (SSO) 功能。

### 必要條件

完成本教學課程中的步驟需要以下內容：

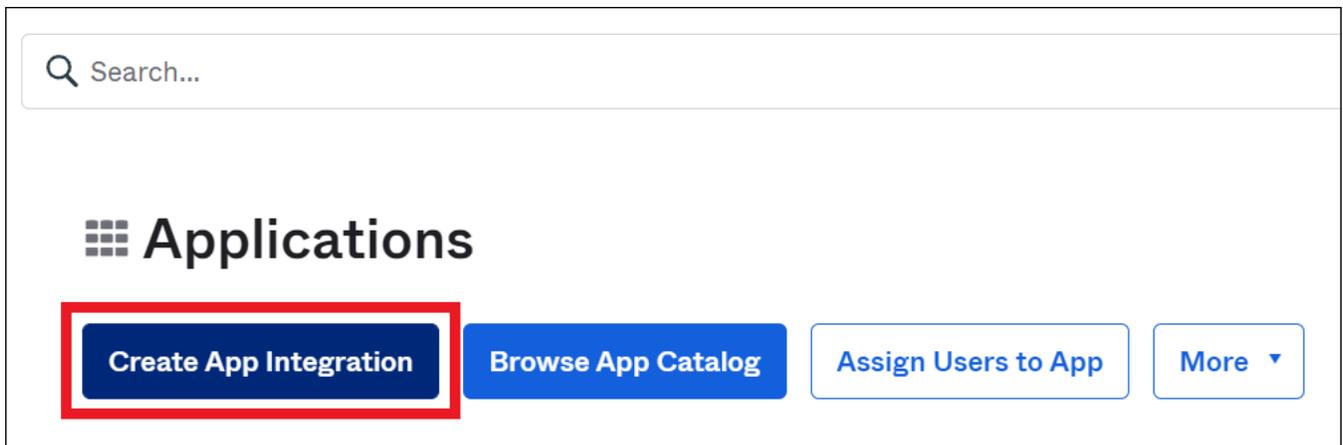
- Athena ODBC 驅動程式 1.1.13 版或更高版本。1.1.13 版及更新版本包括瀏覽器 SAML 支援。如需下載連結，請參閱[使用 ODBC 連接到 Amazon Athena](#)。
- 您要搭配 SAML 使用的 IAM 角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[為 SAML 2.0 聯合身分建立角色](#)。
- Okta 帳戶。如需詳細資訊，請造訪 [okta.com](https://okta.com)。

### 在 Okta 中建立應用程式整合作業

首先，使用 Okta 儀表板建立和設定 SAML 2.0 應用程式，以進行單一登入至 Athena。

使用 Okta 儀表板設定 Athena 的單一登入

1. 登入 Okta 管理頁面，網址為 [okta.com](https://okta.com)。
2. 在導覽窗格中，選擇 Applications (應用程式)、Applications (應用程式)。
3. 在 Applications (應用程式) 頁面，選擇 Create App Integration (建立應用程式整合)。



- 在 Create a new app integration (建立新的應用程式整合) 對話方塊中，為 Sign-in method (登入方法) 選取 SAML 2.0，然後選擇 Next (下一步)。

## Create a new app integration ✕

**Sign-in method**

[Learn More](#)

- OIDC - OpenID Connect**  
Token-based OAuth 2.0 authentication for Single Sign-On (SSO) through API endpoints. Recommended if you intend to build a custom app integration with the Okta Sign-In Widget.
- SAML 2.0**  
XML-based open standard for SSO. Use if the Identity Provider for your application only supports SAML.
- SWA - Secure Web Authentication**  
Okta-specific SSO method. Use if your application doesn't support OIDC or SAML.
- API Services**  
Interact with Okta APIs using the scoped OAuth 2.0 access tokens for machine-to-machine authentication.

[Cancel](#) [Next](#)

- 在 Create SAML Integration (建立 SAML 整合) 頁面的 General Settings (一般設定) 區段中，輸入應用程式的名稱。本教學課程使用名稱 Athena SSO。

### 1 General Settings

App name

App logo (optional)   



App visibility  Do not display application icon to users  
 Do not display application icon in the Okta Mobile app

[Cancel](#) [Next](#)

- 選擇 Next (下一步)。
- 在 Configure SAML (設定 SAML) 頁面的 SAML Settings (SAML 設定) 區段中，輸入以下值：
  - Single sign on URL (URL 上的單一登入)：輸入 **http://localhost:7890/athena**
  - Audience URI (對象 URI)：輸入 **urn:amazon:webservices**

## A SAML Settings

### General

Single sign on URL <sup>?</sup>

Use this for Recipient URL and Destination URL

Allow this app to request other SSO URLs

Audience URI (SP Entity ID) <sup>?</sup>

Default RelayState <sup>?</sup>

If no value is set, a blank RelayState is sent

Name ID format <sup>?</sup>

Application username <sup>?</sup>

[Show Advanced Settings](#)

### Attribute Statements (optional)

[LEARN MORE](#)

8. Attribute Statements (optional) (屬性陳述式 (選用))：輸入以下兩個名稱/值對。這些是必要的映射屬性。

- 於 Name (名稱) 輸入下列 URL：

**`https://aws.amazon.com/SAML/Attributes/Role`**

Value (值)：輸入 IAM 角色名稱。如需 IAM 角色格式的相關資訊，請參閱《IAM 使用者指南》的[為身分驗證回應設定 SAML 聲明](#)。

- 於 NAME (名稱) 輸入下列 URL：

**`https://aws.amazon.com/SAML/Attributes/RoleSessionName`**

於 Value (數值) 輸入 **`user.email`**。

Name	Name format (optional)	Value
https://aws.	Unspecified	YOUR_ROLE
https://aws.	Unspecified	user.email

[LEARN MORE](#)

[Add Another](#)

9. 選擇 Next (下一步)，然後選擇 Finish (完成)。

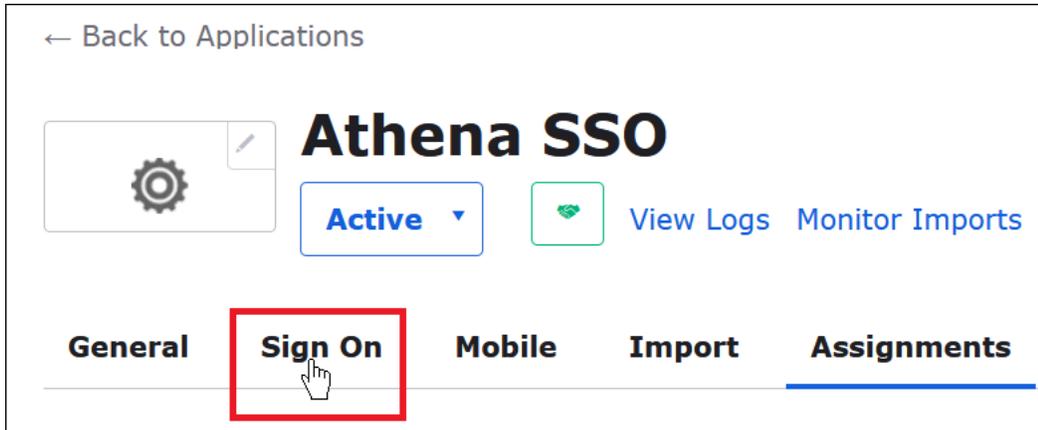
Okta 建立應用程式時還會建立您的登入 URL；接下來會擷取該 URL。

從 Okta 儀表板取得登入 URL

現在您的應用程式已建立完畢，您可以從 Okta 儀表板取得其登入 URL 和其他中繼資料。

## 從 Okta 儀表板取得登入 URL

1. 在 Okta 導覽窗格中，選擇 Applications (應用程式)、Applications (應用程式)。
2. 選擇您要尋找登入 URL 的應用程式，(例如 AthenaSSO)。
3. 在您應用程式的頁面中，選擇 Sign On (登入)。



4. 選擇 View Setup Instructions (檢視設定指示)。

← Back to Applications

# Athena SSO

Active ▾ View Logs Monitor Imports

General **Sign On** Mobile Import Assignments

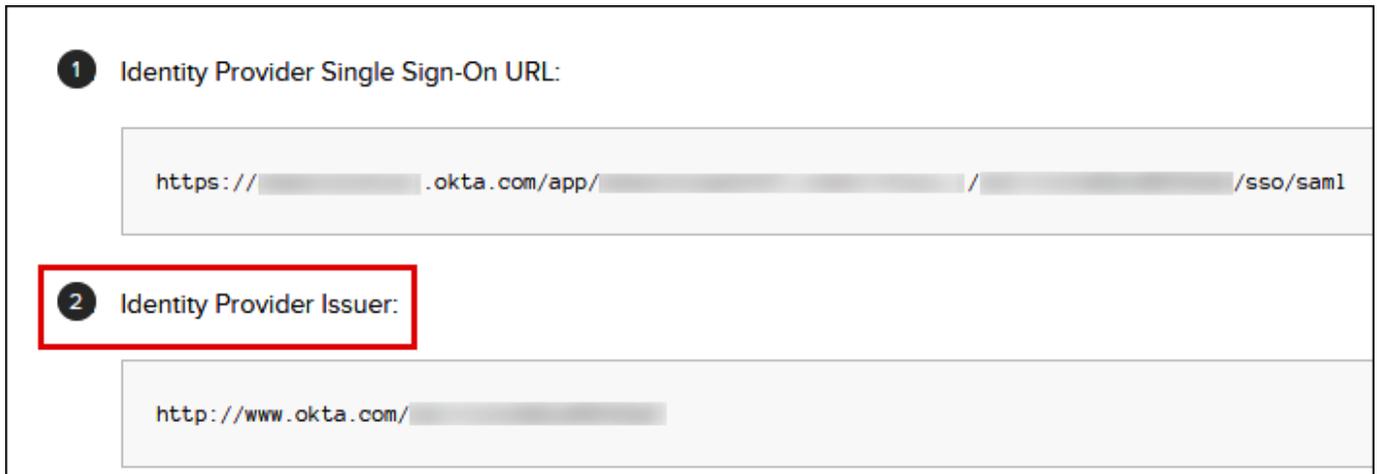
## Settings Edit

**SAML 2.0** is not configured until you complete the setup instructions.

[View Setup Instructions](#)

Identity Provider metadata is available if this application supports dynamic configuration.

5. 在 How to Configure SAML 2.0 for Athena SSO (如何為 Athena SSO 設定 SAML 2.0) 頁面上，尋找身分提供者身分提供者發行者的 URL。Okta 儀表板中的某些位置會將此 URL 視為 SAML 發行者 ID。



1 Identity Provider Single Sign-On URL:

`https://[redacted].okta.com/app/[redacted]/[redacted]/sso/saml`

2 Identity Provider Issuer:

`http://www.okta.com/[redacted]`

6. 複製或存放身分提供者身分提供者單一登入 URL 的值。

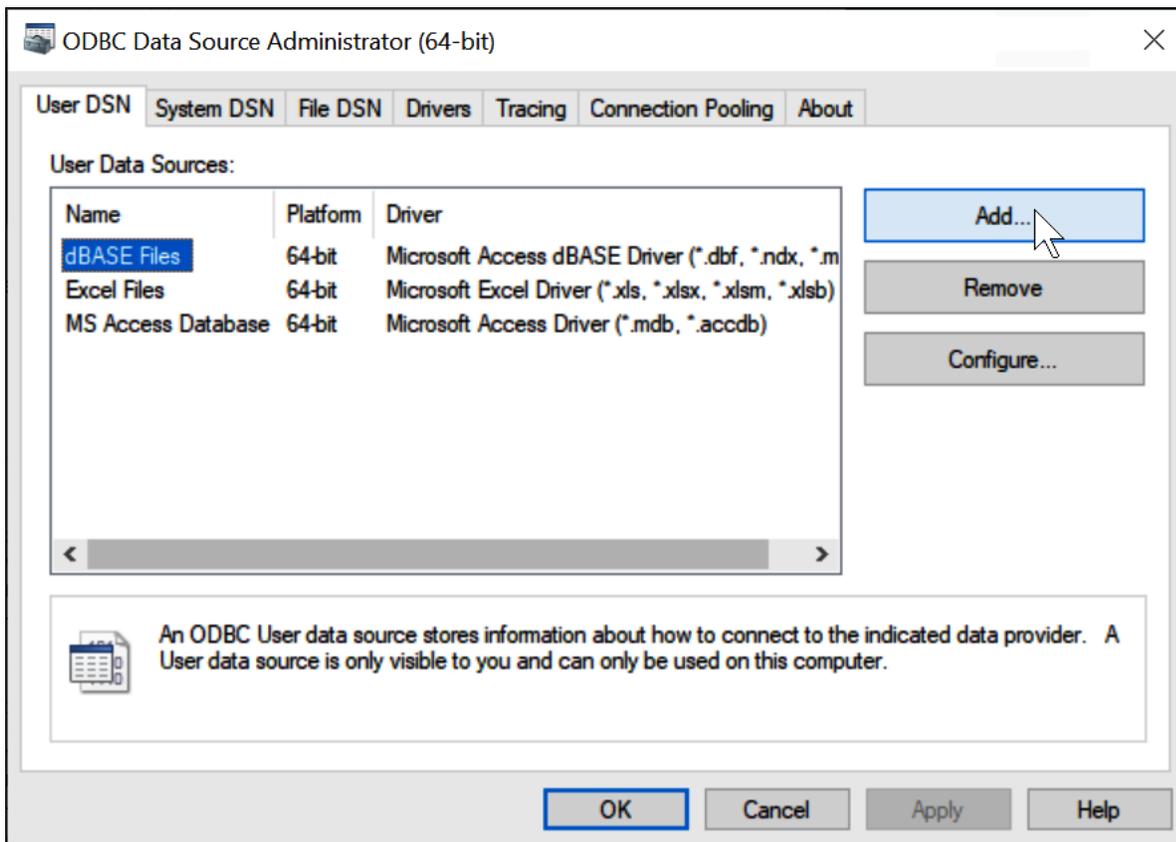
在下一區段中，當您設定 ODBC 連接時，您將提供這個值作為瀏覽器 SAML 外掛程式的登入 URL 連接參數。

#### 設定瀏覽器 SAML ODBC 連接至 Athena

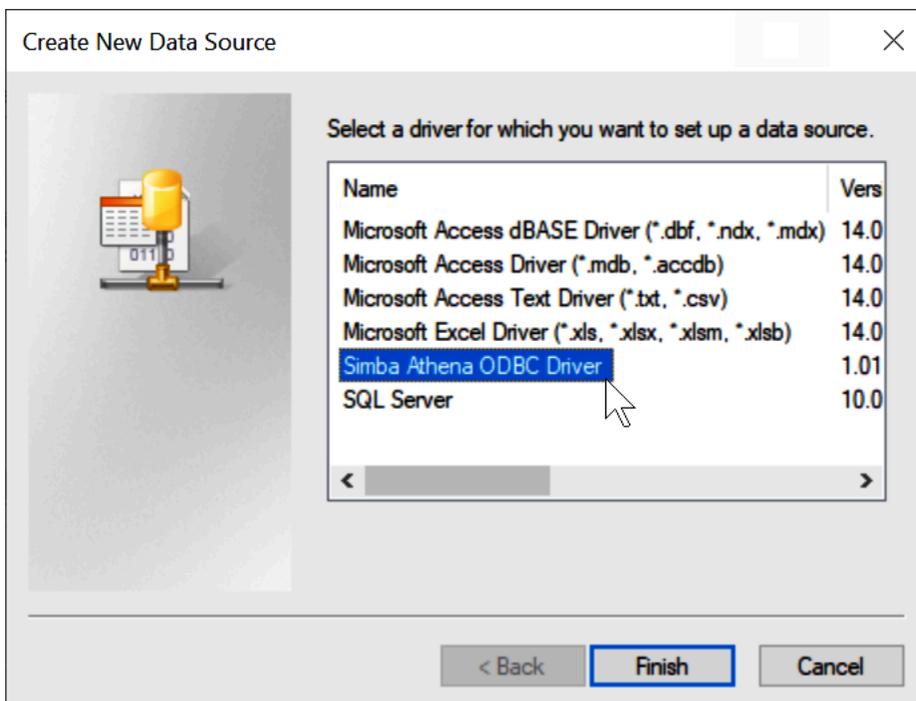
現在，您已經準備好在 Windows 中使用 ODBC 資料來源程式來設定瀏覽器 SAML 與 Athena 的連接。

#### 設定瀏覽器 SAML ODBC 連接至 Athena

1. 在 Windows 中，啟動 ODBC 資料來源程式。
2. 在 ODBC Data Source Administrator (ODBC 資料來源管理員) 程式中，選擇 Add (新增)。



3. 選擇 Simba Athena ODBC Driver (Simba Athena ODBC 驅動程式) , 然後選擇 Finish (完成)。



4. 在 Simba Athena ODBC Driver DSN Setup (Simba Athena ODBC 驅動程式 DSN 設定) 對話方塊中，輸入描述的值。

Simba Athena ODBC Driver DSN Setup

Data Source Name: Athena ODBC 64

Description: My ODBC Data Source

AWS Region: us-west-1

Catalog: AwsDataCatalog

Schema: default

Workgroup: primary

Metadata Retrieval Method: Auto

Output Options

S3 Output Location: s3://DOC-EXAMPLE-BUCKET

Encryption Options: NOT\_SET

KMS Key:

Endpoint Override:

Streaming Endpoint Override:

Authentication Options... Advanced Options... Logging Options... Proxy Options...

v1.1.13.1000 (64 bit) Test... OK Cancel

- Data Source Name (資料來源名稱)：輸入您的資料來源名稱 (例如 Athena ODBC 64)。
  - 於 Description (描述) 輸入您資料來源的描述。
  - 對於 AWS 區域，輸入您 AWS 區域正在使用的 (例如，**us-west-1**)。
  - 在 S3 Output Location (S3 輸出位置) 輸入您要存放輸出內容的 Amazon S3 路徑。
5. 選擇 Authentication Options (身分驗證選項)。
  6. 在 Authentication Options (身分驗證選項) 對話方塊中，選擇或輸入以下值。

### Authentication Options

Authentication Type:

User:

Password:

Session Token:

Preferred Role:

Session Duration:

Login URL:

Listen Port:

Timeout (sec):

Use HTTP Proxy For IdP Host       SSL Insecure

- Authentication Type (身分驗證類型)：選擇 BrowserSAML。
  - 對於登入 URL，輸入您從 Okta 儀表板取得的身分提供者身分提供者單一登入 URL。
  - 於 Listen Port (接聽連接埠) 輸入 7890。
  - Timeout (sec) (逾時 (秒))：輸入連接逾時值 (以秒為單位)。
7. 選擇 OK (確定)，以關閉 Authentication Options (身分驗證選項)。
  8. 選擇 Test (測試) 以測試連接，或 OK (確定) 以完成操作。

## 使用 Amazon Athena Power BI 連接器

您可以在 Windows 作業系統上，使用 Amazon Athena 的 Microsoft Power BI 連接器，以在 Microsoft Power BI 桌面版中分析 Amazon Athena 的資料。如需 Power BI 的相關資訊，請參閱 [Microsoft Power BI](#)。在您將內容發佈至 Power BI 服務後，便可使用 2021 年 7 月或更新版本的 [Power BI 閘道](#)，透過隨需或排程的重新整理，讓內容保持在最新狀態。

### 必要條件

在您開始前，請確定您的環境符合下列要求。需使用 Amazon Athena ODBC 驅動程式。

- [AWS 帳戶](#)
- [使用 Athena 的許可](#)
- [Amazon Athena ODBC 驅動程式](#)
- [Power BI 桌面版](#)

### 支援的功能

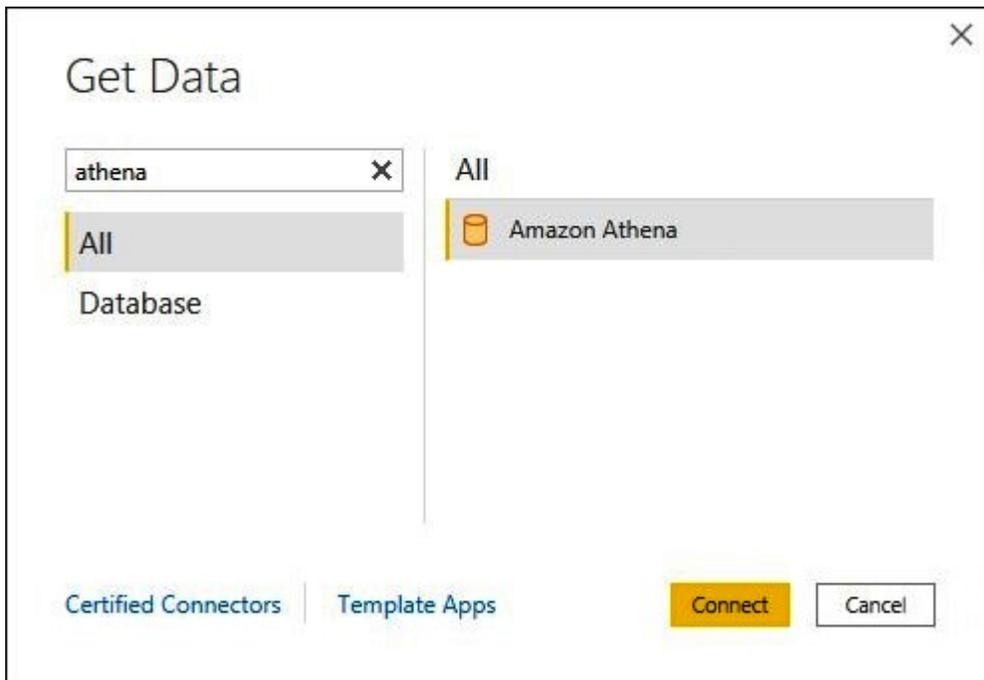
- 匯入 - 將選取的資料表和資料欄匯入 Power BI 桌面版，以進行查詢。
- DirectQuery— 沒有數據導入或複製到 Power BI 桌面。Power BI 桌面版會直接查詢基礎資料來源。
- 電源 BI 閘道 — 在您 AWS 帳戶 的內部部署資料閘道器，就像 Microsoft 電力 BI 服務和 Athena 之間的橋樑。需要有閘道才能查看您在 Microsoft Power BI 服務上的資料。

## 連接至 Amazon Athena

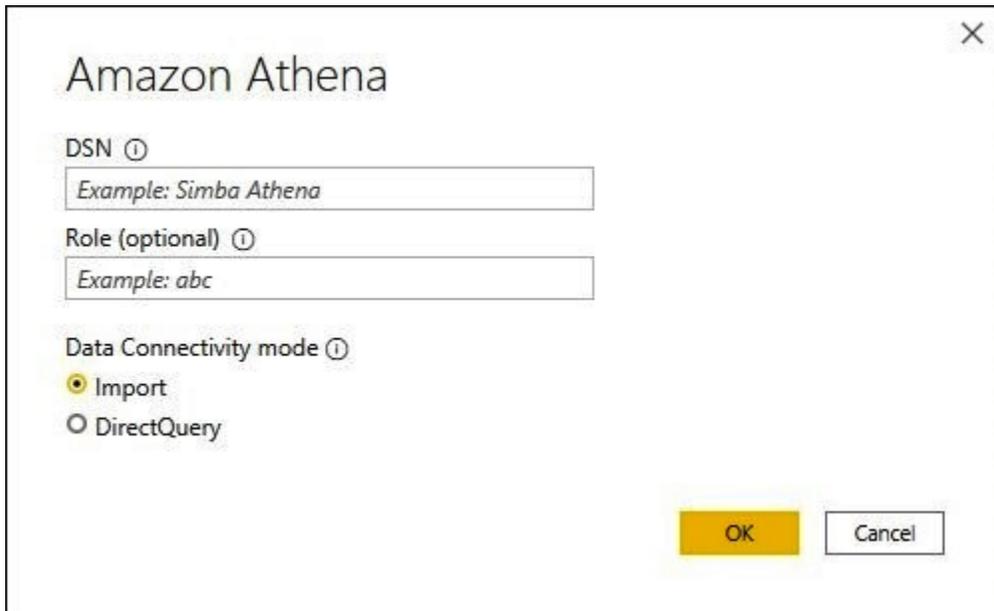
若要將 Power BI 桌面版連接至您的 Amazon Athena 資料，請執行下列步驟。

## 從 Power BI 桌面版連接至 Athena 資料

1. 啟動 Power BI 桌面版。
2. 執行以下任意一項：
  - 選擇 File (檔案)，然後選取 Get Data (取得資料)
  - 從 Home (首頁) 功能區中，選擇 Get Data (取得資料)。
3. 在搜尋方塊中，輸入 Athena。
4. 選取 Amazon Athena，然後選擇 Connect (連接)。

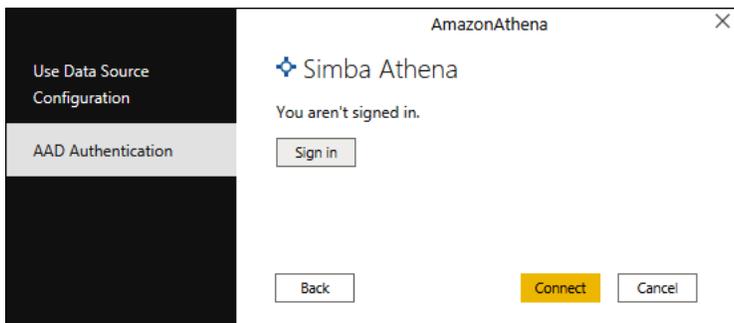


5. 在 Amazon Athena 連接頁面上，輸入以下資訊。
  - 若為 DSN，請輸入您要使用的 ODBC DSN 名稱。如需有關設定 DSN 的指示，請參閱 [ODBC 驅動程式文件](#)。
  - 若為資料連線模式，請依照以下一般準則選擇一個適合您使用案例的模式：
    - 若為較小的資料集，請選擇 Import (匯入)。在使用匯入模式時，Power BI 會與 Athena 搭配運作，以匯入整個資料集的內容，以便在視覺效果中使用。
    - 對於較大的資料集，請選擇 DirectQuery。在 DirectQuery 模式下，不會將任何資料下載到您的工作站。在您建立視覺效果或與其互動時，Microsoft Power BI 會與 Athena 搭配運作，動態查詢基礎資料來源，讓您隨時都能檢視目前資料。如需有關的詳細資訊 DirectQuery，請參閱 [Microsoft 文件 DirectQuery 中的在 BI 桌面](#) 中使用。



The screenshot shows a dialog box titled "Amazon Athena" with a close button (X) in the top right corner. It contains three input fields and two radio buttons. The first field is labeled "DSN" with a help icon (i) and contains the text "Example: Simba Athena". The second field is labeled "Role (optional)" with a help icon (i) and contains the text "Example: abc". The third section is labeled "Data Connectivity mode" with a help icon (i) and has two radio buttons: "Import" (which is selected) and "DirectQuery". At the bottom right, there are two buttons: "OK" (highlighted in yellow) and "Cancel".

6. 選擇確定。
7. 在設定資料來源身分驗證的提示中，選擇 Use Data Source Configuration (使用資料來源組態) 或AAD Authentication (AAD 身分驗證)，然後選擇 Connect (連接)。



The screenshot shows a dialog box titled "AmazonAthena" with a close button (X) in the top right corner. On the left side, there is a vertical sidebar with two options: "Use Data Source Configuration" (highlighted in black) and "AAD Authentication" (highlighted in grey). The main area of the dialog shows a blue diamond icon followed by "Simba Athena" and the text "You aren't signed in." Below this text is a "Sign in" button. At the bottom of the dialog, there are three buttons: "Back", "Connect" (highlighted in yellow), and "Cancel".

您的資料目錄、資料庫和資料表會顯示在 Navigator (導覽器) 對話方塊中。

Navigator

Display Options ▾

- demo-dsn [1]
- AwsDataCatalog [3]
  - default [8]
  - demo-datasets [2]
    - iris
    - demo\_datasets
  - sampledb [5]

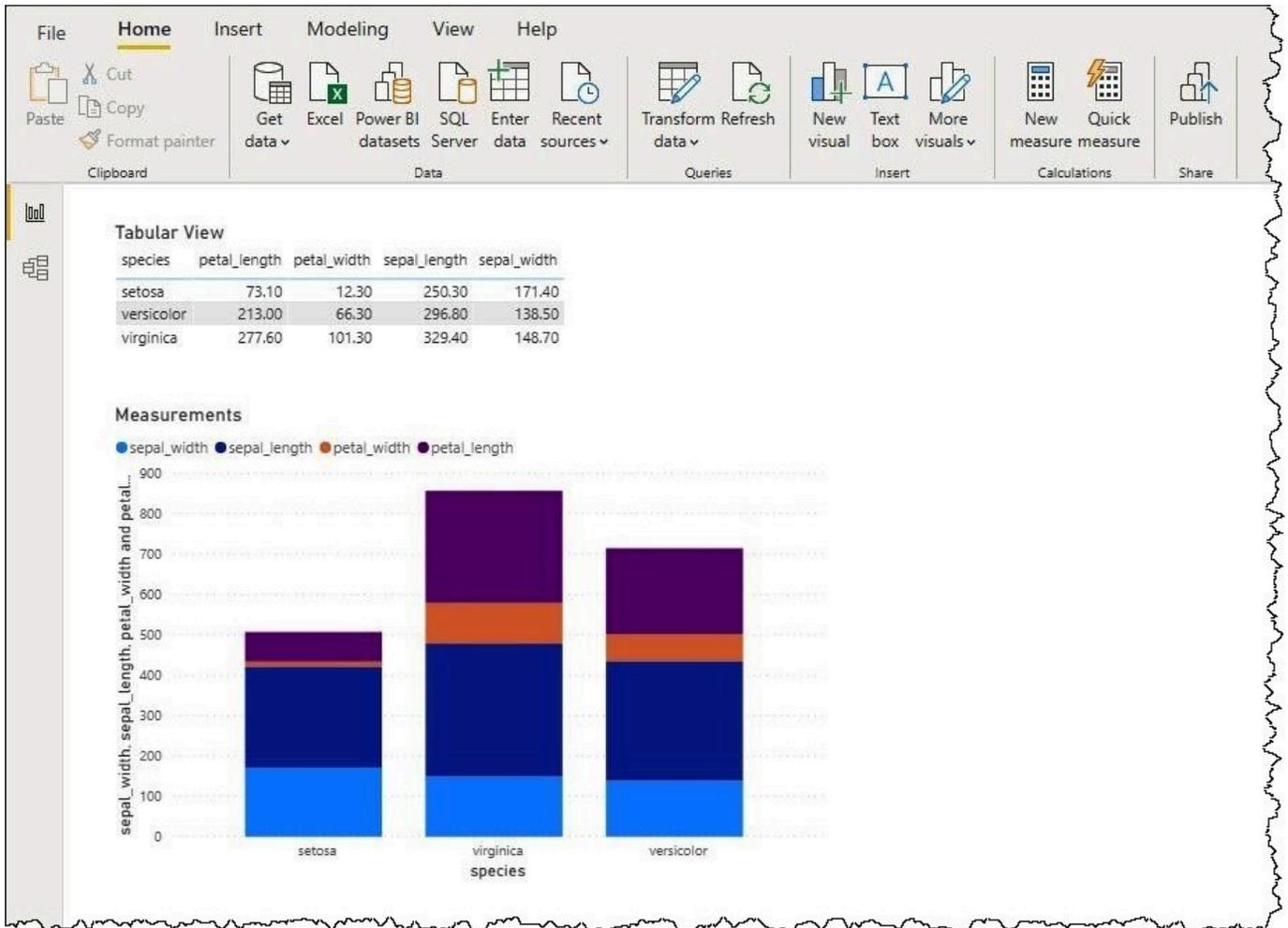
iris

Preview downloaded on Thursday

species	sepal_length	sepal_width	petal_length	petal_width
setosa	5.1	3.5	1.4	0.0
setosa	4.9	3	1.4	0.0
setosa	4.7	3.2	1.3	0.0
setosa	4.6	3.1	1.5	0.0
setosa	5	3.6	1.4	0.0
setosa	5.4	3.9	1.7	0.0
setosa	4.6	3.4	1.4	0.0
setosa	5	3.4	1.5	0.0
setosa	4.4	2.9	1.4	0.0
setosa	4.9	3.1	1.5	0.0
setosa	5.4	3.7	1.5	0.0
setosa	4.8	3.4	1.6	0.0
setosa	4.8	3	1.4	0.0
setosa	4.3	3	1.1	0.0
setosa	5.8	4	1.2	0.0
setosa	5.7	4.4	1.5	0.0
setosa	5.4	3.9	1.3	0.0
setosa	5.1	3.5	1.4	0.0
setosa	5.7	3.8	1.7	0.0
setosa	5.1	3.8	1.5	0.0
setosa	5.4	3.4	1.7	0.0
setosa	5.1	3.7	1.5	0.0

Load Transform Data Cancel

- 在 Display Options (顯示選項) 窗格中，選取您要使用之資料集的核取方塊。
- 如果您要在匯入資料集之前轉換資料集，請前往對話方塊底部，並選擇 Transform Data (轉換資料)。這會開啟 Power Query 編輯器，以便您篩選和精簡要使用的資料集。
- 選擇載入。載入完成後，您便可以建立視覺效果 (如下圖所示)。如果您選取 DirectQuery 為匯入模式，Power BI 會針對您要求的視覺效果發出查詢給 Athena。



## 設定內部部署閘道

您可以將儀表板和資料集發佈至 Power BI 服務，讓其他使用者可以透過 Web、行動裝置和內嵌應用程式與其互動。若要在 Microsoft Power BI 服務中查看您的資料，請在 AWS 帳戶中安裝 Microsoft Power BI 內部部署資料閘道。閘道會如同 Microsoft Power BI 服務和 Athena 之間的橋樑般運作。

### 下載、安裝及測試內部部署資料閘道

1. 請前往 [Microsoft Power BI gateway download](#) (Microsoft Power BI 閘道下載) 頁面，然後選擇個人模式或標準模式。個人模式適用於在本機測試 Athena 連接器。標準模式適用於多名使用者生產設定。
2. 若要安裝內部部署閘道 (個人或標準模式)，請參閱 Microsoft 文件中的 [安裝內部部署資料閘道](#)。
3. 若要測試閘道，請依照 Microsoft 文件中的 [搭配使用自訂資料連接器和內部部署資料閘道](#)。

如需內部部署資料閘道的詳細資訊，請參閱以下 Microsoft 資源。

- [什麼是內部部署資料閘道？](#)
- [部署 Power BI 資料閘道的指引](#)

如需設定 Power BI 閘道以搭配 Athena 使用的範例，請參閱 AWS 大數據部落格文章[使用亞馬遜 Athena 在 Microsoft Power BI 上快速建立儀表板](#)。

## 建立資料庫和資料表

Amazon Athena 支援一部分的資料定義語言 (DDL) 陳述式及 ANSI SQL 函數和運算子，以定義和查詢 Amazon Simple Storage Service 中資料所在的外部資料表。

在 Athena 中建立資料庫和資料表時，您需要描述資料的結構描述和位置，才能讓資料表中的資料可供即時查詢。

為改善查詢效能並降低成本，建議您分割資料，並使用開放原始碼單欄格式以儲存於 Amazon S3 中，例如 [Apache Parquet](#) 或 [ORC](#)。

### 主題

- [在 Athena 中建立資料庫](#)
- [在 Athena 中建立資料表](#)
- [資料表、資料庫和資料欄的名稱](#)
- [保留的關鍵字](#)
- [Amazon S3 中的資料表位置](#)
- [單欄式儲存格式](#)
- [轉換為單欄式格式](#)
- [在 Athena 中分割資料](#)
- [使用 Amazon Athena 進行分割區投影](#)

## 在 Athena 中建立資料庫

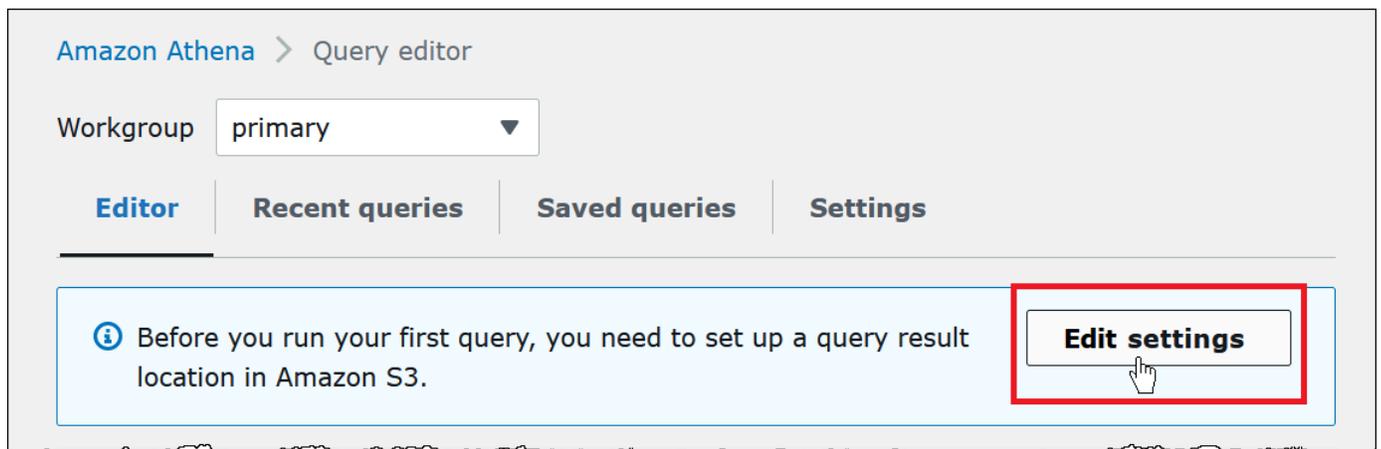
Athena 中的資料庫是您在其中建立的資料表的邏輯群組。

## 先決條件

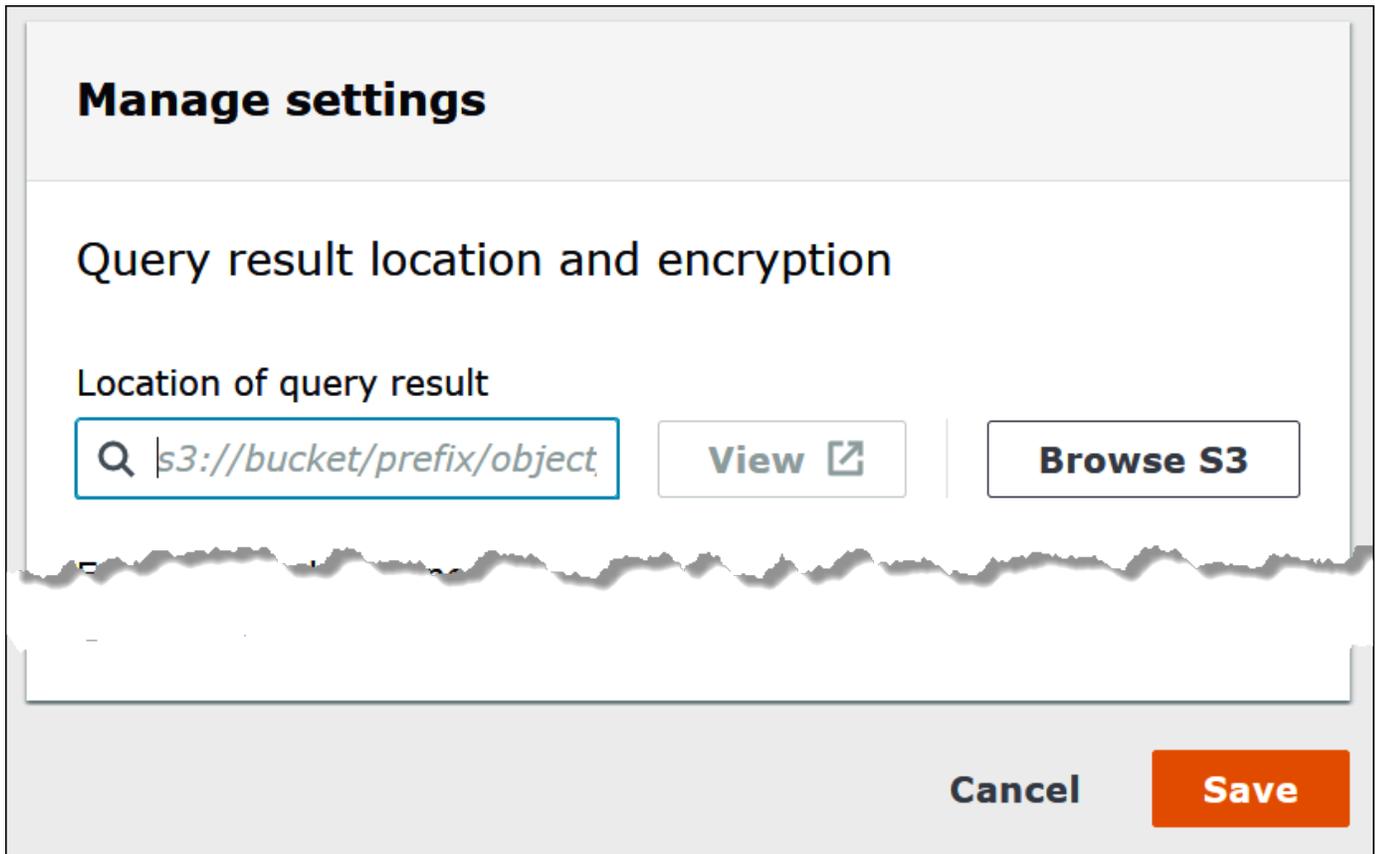
如果您尚未在 Amazon S3 中設定查詢輸出位置，請執行下列先決條件步驟。

### 若要建立查詢輸出位置

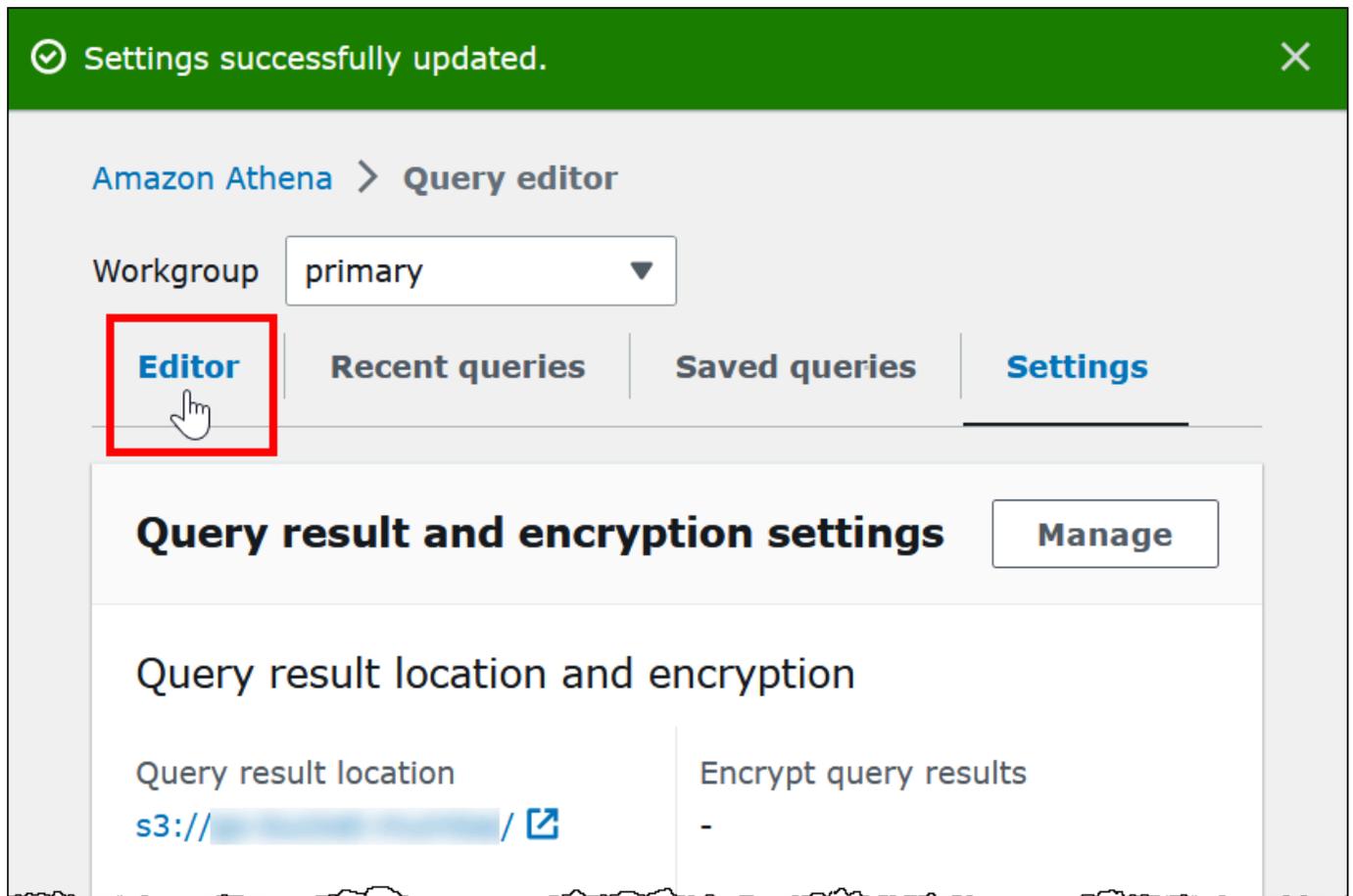
1. 使用您用於 Athena 的相同 AWS 區域和帳戶，請依照步驟 (例如，使用 Amazon S3 主控台) [在 Amazon S3 中建立儲存貯體](#) 來保存您的 Athena 查詢結果。您可將此儲存貯體設定為您的查詢輸出位置。
2. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
3. 如果這是您第一次造訪此 AWS 區域中的 Athena 主控台，請選擇探索查詢編輯器以開啟查詢編輯器。此外，Athena 會於查詢編輯器中開啟。
4. 選擇 Edit Settings (編輯設定) 以在 Amazon S3 中設定查詢結果位置。



5. 對於 Manage settings (管理設定)，請執行下列其中一項動作：
  - 在 Location of query result (查詢結果的位置) 方塊中，輸入您在 Amazon S3 中為查詢結果建立的儲存貯體路徑。在路徑中加上 `s3://` 作為字首。
  - 選擇 Browse S3 (瀏覽 S3)，選擇您在目前區域建立的 Amazon S3 儲存貯體，然後選擇 Choose (選擇)。



6. 選擇 Save (儲存)。
7. 選擇 Editor (編輯器) 以切換至查詢編輯器。



## 建立資料庫

在您設定查詢結果位置之後，在 Athena 主控台查詢編輯器中建立資料庫輕鬆簡單。

若要使用 Athena 查詢編輯器建立資料庫

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在編輯器索引標籤上，輸入 Hive 資料定義語言 (DDL) 命令 `CREATE DATABASE myDataBase`。以您想要的資料庫名稱取代 *myDatabase*。有關資料庫名稱的限制，請參閱 [資料表、資料庫和資料欄的名稱](#)。
3. 選擇 Run (執行) 或按 **Ctrl+ENTER**。
4. 如需使您的資料庫訂為目前的資料庫，請從查詢編輯器左側的 Database (資料庫) 選單中選取它。

如需 Athena 資料庫控制許可的相關資訊，請參閱 [對於中的資料庫和資料表的精細存取 AWS Glue Data Catalog](#)。

## 在 Athena 中建立資料表

您可以使用 JDBC 或 ODBC 驅動程式，或使用 Athena [Create table \(建立資料表\) 表單](#)，在 Athena 主控台中執行 DDL 陳述式。

您在 Athena 中建立新的資料表結構描述時，Athena 會將結構描述存放在資料目錄中，並在您執行查詢時使用它。

Athena 使用稱為的方法 schema-on-read，也就是說，在您執行查詢時，結構描述會投影到您的資料上。如此便不必載入或轉換資料。

Athena 不會修改您在 Amazon S3 中的資料。

Athena 使用 Apache Hive 來定義資料表和建立資料庫，這本質上是資料表的邏輯命名空間。

在 Athena 中建立資料庫和資料表時，您只需描述結構描述，以及位於 Amazon S3 中供讀取時間查詢的資料表資料所在的位置。因此，資料庫和資料表會有與傳統關聯式資料庫系統稍微不同的意義，因為資料不會隨著資料庫和資料表的結構描述定義存放。

查詢時，您會使用標準 SQL 和查詢資料表，並在那時讀取資料。您可以使用 [Apache Hive 文件](#) 來尋找如何建立資料庫和資料表的指導，但以下提供 Athena 的專用指導。

最大查詢字串長度為 256 KB。

蜂巢通過使用序列化程序解串器 ( ) 庫的支持多種數據格式。SerDe 您也可以使用規則表達式來定義複雜的結構描述。如需支援的 SerDe 程式庫清單，請參閱 [支援的 SerDes 和資料格式](#)。

### 考量與限制

以下是 Athena 中針對資料表的一些重要限制和注意事項。

#### Athena 資料表和 Amazon S3 資料的要求

建立資料表時，您需使用 LOCATION 子句來指定基礎資料的 Amazon S3 儲存貯體位置。考慮下列各項：

- Athena 只能查詢版本控制的 Amazon S3 儲存貯體上最新版本的資料，無法查詢之前版本的資料。
- 您必須擁有適當的許可，才可使用 Amazon S3 位置的資料。如需詳細資訊，請參閱 [從 Athena 訪問 Amazon S3](#)。
- Athena 支援查詢的物件可以與多個儲存類別一起存放在 LOCATION 子句指定的相同儲存貯體中。例如，您可以在存放於 Amazon S3 中不同儲存類別 (標準、標準 - IA 和智慧型分層) 的物件中查詢資料。

- Athena 支援[申請者付款儲存貯體](#)。如需有關如何對含有您要在 Athena 中查詢之來源資料的儲存貯體啟用申請者付款的資訊，請參閱[建立工作群組](#)。
- Athena 不支援查詢 [S3 Glacier Flexible Retrieval](#) 中的資料或 S3 Glacier Deep Archive 儲存類別。在 S3 Glacier Flexible Retrieval 或 S3 Glacier Deep Archive 儲存類別中忽略的物件。作為替代方法，可以使用 Amazon S3 Glacier Instant Retrieval 儲存類別，該儲存類別可由 Athena 查詢。如需詳細資訊，請參閱 [Amazon S3 Glacier Instant Retrieval 儲存類別](#)。

如需有關儲存類別的資訊，請參閱[儲存類別](#)、[在 Amazon S3 中變更物件的儲存類別](#)、[轉換為 GLACIER 儲存類別 \(物件封存\)](#)，以及《Amazon Simple Storage Service 使用者指南》中的[申請者付款儲存貯體](#)。

- 如果您對具有大量物件的 Amazon S3 儲存貯體發出查詢，並且未將資料分割，這類查詢可能會影響 Amazon S3 中的 Get 請求率限制，並導致 Amazon S3 例外狀況。若要避免錯誤，請分割您的資料。此外，請考慮調校您的 Amazon S3 請求率。如需詳細資訊，請參閱[請求率和效能考量](#)。
- 如果您使用 AWS Glue [CreateTable](#) API 作業或 AWS CloudFormation [AWS::Glue::Table](#) 範本建立要在 Athena 中使用的資料表，但未指定 `TableType` 屬性，然後執行類似 `SHOW CREATE TABLE` 或的 DDL 查詢 `MSCK REPAIR TABLE`，則可能會收到錯誤訊息 `FileNotFoundException` 名稱為空值。

若要解決錯誤，請指定 [TableInput](#) `TableType` 屬性值做為 AWS Glue `CreateTable` API 呼叫或 [AWS CloudFormation 範本](#) 的一部分。`TableType` 可能的值包括 `EXTERNAL_TABLE` 或 `VIRTUAL_VIEW`。

只有當您使用 AWS Glue `CreateTable` API 作業或 [AWS::Glue::Table](#) 範本建立資料表時，才會套用此需求。如果您使用 DDL 陳述式或 AWS Glue 爬蟲程式建立 Athena 資料表，則系統會為您自動定義 `TableType` 屬性。

## 支援的函數

Athena 查詢中支援的函數對應於 Trino 和 Presto 中的函數。如需有關每個函數的資訊，請參閱 [Trino](#) 或 [Presto](#) 文件中的函數和運算子部分。

## 不支援交易資料轉換

Athena 不支援對資料表資料進行以交易為基礎的操作 (例如可在 Hive 或 Presto 中找到的操作)。如需不支援的關鍵字完整清單，請參閱[不支援的 DDL](#)。

## 變更資料表狀態的操作為 ACID

當您建立、更新或刪除資料表時，這些操作保證與 ACID 相容。例如，如果多個使用者或用戶端同時嘗試建立或修改現有的資料表，只有其中一個會成功。

## 資料表為 EXTERNAL

除了在建立 [Iceberg](#) 資料表時，請始終使用 EXTERNAL 關鍵字。如果您在非 Iceberg 資料表上使用 CREATE TABLE 卻沒有帶 EXTERNAL 關鍵字，則 Athena 會發出錯誤。當您在 Athena 中捨棄資料表時，只會移除資料表中繼資料，資料會保留在 Amazon S3 中。

## 使用 AWS Glue 或 Athena 主控台建立資料表

您可以使用新增表格表單 AWS Glue，或在 Athena 查詢編輯器中執行 DDL 陳述式，在 Athena 中建立資料表。

### 使用 AWS Glue 爬行者程式建立表格

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在查詢編輯器中，選擇位於資料表和檢視旁的建立，然後選擇 AWS Glue 爬蟲程式。
3. 請依照 AWS Glue 主控台 Add crawler (新增爬蟲程式) 頁面上的步驟新增爬蟲程式。

如需詳細資訊，請參閱 [使用 AWS Glue 爬蟲](#)。

### 使用 Athena 新增資料表表單來建立資料表

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在查詢編輯器中，選擇位在 Tables and views (資料表和檢視) 旁的 Create (建立)，然後選擇 S3 bucket data (S3 儲存貯體資料)。
3. 在 Create Table From S3 bucket data (從 S3 儲存貯體資料建立資料表) 表單中，輸入用於建立資料表的資訊，然後選擇 Create table (建立資料表)。如需表單欄位的詳細資訊，請參閱[使用表單新增資料表](#)。

### 使用 Hive DDL 建立資料表

1. 從 Database (資料庫) 選單中，選擇要為其建立資料表的資料庫。如果您未在 CREATE TABLE 陳述式中指定資料庫，則會在查詢編輯器中目前選取的資料庫中建立資料表。
2. 在查詢編輯器中輸入類似下列的陳述式，然後選擇 Run (執行)，或按 **Ctrl+ENTER**。

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (
  `Date` Date,
  Time STRING,
  Location STRING,
  Bytes INT,
  RequestIP STRING,
  Method STRING,
  Host STRING,
  Uri STRING,
  Status INT,
  Referrer STRING,
  OS String,
  Browser String,
  BrowserVersion String
) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  "input.regex" = "^(?!#)([ ]+)\s+([ ]+)\s+([ ]+)\s+([ ]+)\s+([ ]+)\s+
+([ ]+)\s+([ ]+)\s+([ ]+)\s+([ ]+)\s+([ ]+)\s+([ ]+)\s+[\(\)]([\;\;]+).*%20([
\ ]+)\[\\](.*)$"
) LOCATION 's3://athena-examples-MyRegion/cloudfront/plaintext/';
```

## 顯示資料表資訊

在 Athena 中建立資料表之後，其名稱會顯示在左側的 Tables (資料表) 清單中。如需顯示有關資料表的資訊並進行管理，可以選擇 Athena 主控台中資料表名稱旁邊的三個垂直點。

- 預覽資料表 – 透過執行 Athena 查詢編輯器中的 `SELECT * FROM "database_name"."table_name" LIMIT 10` 陳述式，顯示所有資料欄的前 10 列。
- 產生建立資料表 DDL – 您可以透過執行 Athena 查詢編輯器中的 [SHOW CREATE TABLE table\\_name](#) 陳述式，產生可用來重新建立資料表的 DDL 陳述式。
- 載入分割區 – 執行 Athena 查詢編輯器中的 [MSCK REPAIR TABLE table\\_name](#) 陳述式。僅當資料表有分割區時，才可使用此選項。
- 插入編輯器中 - 在目前編輯位置將資料表的名稱插入查詢編輯器中。
- Delete table (刪除資料表) – 顯示確認對話方塊，詢問您是否要刪除資料表。如果您同意，請執行 Athena 查詢編輯器中的 [DROP TABLE table\\_name](#) 陳述式。
- Table properties (資料表屬性) – 顯示資料表名稱、資料庫名稱、建立時間，以及資料表是否有加密資料。

## 資料表、資料庫和資料欄的名稱

使用這些秘訣來命名 Athena 中的資料庫物件。

### 資料庫、資料表和資料欄名稱需求

- 中的資料庫名稱、表格名稱和欄名稱可接受的字元 AWS Glue 必須是 UTF-8 字串。字串長度不得小於 1 或超過 255 個位元組。超過此限制會產生錯誤，例如「name」的值無法滿足限制條件：成員的長度必須小於或等於 255。可以使用的字元包括空格，並由下列單行字串樣式定義：

```
[\\u0020-\\uD7FF\\uE000-\\uFFFF\\uD800\\uDC00-\\uDBFF\\uDFFF\\t]*
```

- 目前，AWS Glue regex 模式允許將前導空格添加到名稱的開頭。由於這些前導空格很難偵測，而且在建立之後可能會造成可用性問題，因此請避免建立具有前導空格的物件名稱。
- 如果您使用 [AWS::Glue::Database](#) AWS CloudFormation 範本來建立資 AWS Glue 料庫，但未指定資料庫名稱，則會 AWS Glue 自動產生與 Athena 不相容的 ##### - random\_string ##### 稱。
- 您可以使用「目 AWS Glue 錄管理員」更名欄，但無法更名表格名稱或資料庫名稱。若要解決此限制，您必須使用舊資料庫的定義，建立具有新名稱的資料庫。然後，您可以使用舊資料庫中資料表的定義，重新建立新資料庫中的資料表。若要這麼做，您可以使用 AWS CLI 或 AWS Glue SDK。如需這些步驟，請參閱 [使用重新 AWS CLI 建立 AWS Glue 資料庫及其表格](#)。

### 對 Athena 中的資料表名稱和資料表資料欄名稱使用小寫

Athena 接受在 DDL 和 DML 查詢中混合使用大小寫，但是當其執行查詢時名稱須為小寫。出於這個原因，請避免對資料表或資料欄名稱混合使用大小寫，並且不要在 Athena 中僅依賴大小寫來區分這些名稱。例如，如果您使用 DDL 陳述式來建立名為 Castle 的資料欄，則建立的資料欄將會是小寫的 castle。如果您接著在 DML 查詢中將資料欄名稱指定為 Castle 或 CASTLE，則 Athena 會將您要執行查詢的名稱變成小寫，但會使用您在查詢中選擇的大小寫來顯示資料欄標題。

資料庫、資料表和資料欄名稱必須小於或等於 255 個字元的長度。

### 以底線開頭的名稱

建立資料表時，使用反引號括住以底線開頭的資料表、檢視或資料欄名稱。例如：

```
CREATE EXTERNAL TABLE IF NOT EXISTS `myunderscoretable`(  
  `_id` string, `_index` string)
```

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

## 以數字開頭的資料表、檢視或資料行名稱

在執行 SELECT、CTAS 或 VIEW 查詢時，在以數字開頭的識別碼 (例如資料表、檢視或資料欄名稱) 周圍加上引號。例如：

```
CREATE OR REPLACE VIEW "123view" AS
SELECT "123columnone", "123columntwo"
FROM "234table"
```

## 資料欄名稱和複雜類型

對於複雜類型，資料欄名稱中僅允許英數字元、底線 (\_) 和句點 (.)。若要為具有限制字元的索引鍵建立資料表和映射，您可以使用自訂 DDL 陳述式。如需詳細資訊，請參閱AWS 大數據部落格[格中的使用 JSON 從巢狀 JSON 和對應SerDe在 Amazon Athena 建立資料表一文](#)。

## 保留字

Athena 中的某些保留字必須逸出。若要逸出 DDL 陳述式中的預留關鍵字，請以反引號 (`) 括住它們。若要逸出[檢視](#)上的 SQL SELECT 陳述式和查詢中的預留關鍵字，請以雙引號 (") 將其括住。

如需詳細資訊，請參閱 [保留的關鍵字](#)。

## 其他資源

如需完整的資料庫和資料表建立語法，請參閱下列頁面。

- [CREATE DATABASE](#)
- [CREATE TABLE](#)

如需中資料庫和表格的詳細資訊 AWS Glue，請參閱《AWS Glue 開發人員指南》中的 < [資料庫和表格](#) >。

## 保留的關鍵字

在 Athena 中執行包含預留關鍵字的查詢時，必須以特殊字元括住關鍵字以將它們逸出。請使用此主題中的清單來檢查 Athena 中的預留關鍵字。

若要逸出 DDL 陳述式中的預留關鍵字，請以反引號 ( ` ) 括住它們。若要逸出[檢視](#)上的 SQL SELECT 陳述式和查詢中的預留關鍵字，請以雙引號 ( " ) 將其括住。

- [DDL 陳述式中的預留關鍵字清單](#)
- [SQL SELECT 陳述式中的預留關鍵字清單](#)
- [含有預留關鍵字的查詢範例](#)

## DDL 陳述式中的預留關鍵字清單

Athena 在其 DDL 陳述式中使用下列預留關鍵字清單。如果您使用它們時沒有逸出，Athena 會發出錯誤。若要逸出它們，請以反引號 ( ` ) 括住。

您在 DDL 陳述式中使用 DDL 預留關鍵字作為識別符名稱時，必須以反引號 ( ` ) 括住。

```
ALL, ALTER, AND, ARRAY, AS, AUTHORIZATION, BETWEEN, BIGINT,
BINARY, BOOLEAN, BOTH, BY, CASE, CASHE, CAST, CHAR, COLUMN,
CONF, CONSTRAINT, COMMIT, CREATE, CROSS, CUBE, CURRENT,
CURRENT_DATE, CURRENT_TIMESTAMP, CURSOR, DATABASE, DATE,
DAYOFWEEK, DECIMAL, DELETE, DESCRIBE, DISTINCT, DOUBLE, DROP,
ELSE, END, EXCHANGE, EXISTS, EXTENDED, EXTERNAL, EXTRACT,
FALSE, FETCH, FLOAT, FLOOR, FOLLOWING, FOR, FOREIGN, FROM,
FULL, FUNCTION, GRANT, GROUP, GROUPING, HAVING, IF, IMPORT,
IN, INNER, INSERT, INT, INTEGER, INTERSECT, INTERVAL, INTO,
IS, JOIN, LATERAL, LEFT, LESS, LIKE, LOCAL, MACRO, MAP, MORE,
NONE, NOT, NULL, NUMERIC, OF, ON, ONLY, OR, ORDER, OUT,
OUTER, OVER, PARTIALSCAN, PARTITION, PERCENT, PRECEDING,
PRECISION, PRESERVE, PRIMARY, PROCEDURE, RANGE, READS,
REDUCE, REGEXP, REFERENCES, REVOKE, RIGHT, RLIKE, ROLLBACK,
ROLLUP, ROW, ROWS, SELECT, SET, SMALLINT, START, TABLE,
TABLESAMPLE, THEN, TIME, TIMESTAMP, TO, TRANSFORM, TRIGGER,
TRUE, TRUNCATE, UNBOUNDED, UNION, UNIQUEJOIN, UPDATE, USER,
USING, UTC_TIMESTAMP, VALUES, VARCHAR, VIEWS, WHEN, WHERE,
WINDOW, WITH
```

## SQL SELECT 陳述式中的預留關鍵字清單

Athena 在檢視上的 SQL SELECT 陳述式和查詢中，使用下列預留關鍵字清單。

如果您使用這些關鍵字作為識別符，您必須在查詢陳述式中以雙引號 ( " ) 括住它們。

```
ALTER, AND, AS, BETWEEN, BY, CASE, CAST, CONSTRAINT, CREATE,
CROSS, CUBE, CURRENT_CATALOG, CURRENT_DATE, CURRENT_PATH,
CURRENT_SCHEMA, CURRENT_TIME, CURRENT_TIMESTAMP, CURRENT_USER,
DEALLOCATE, DELETE, DESCRIBE, DISTINCT, DROP, ELSE, END, ESCAPE,
EXCEPT, EXECUTE, EXISTS, EXTRACT, FALSE, FIRST, FOR, FROM,
FULL, GROUP, GROUPING, HAVING, IN, INNER, INSERT, INTERSECT,
INTO, IS, JOIN, JSON_ARRAY, JSON_EXISTS, JSON_OBJECT,
JSON_QUERY, JSON_TABLE, JSON_VALUE, LAST, LEFT, LIKE,
LISTAGG, LOCALTIME, LOCALTIMESTAMP, NATURAL, NORMALIZE,
NOT, NULL, OF, ON, OR, ORDER, OUTER, PREPARE, RECURSIVE, RIGHT,
ROLLUP, SELECT, SKIP, TABLE, THEN, TRIM, TRUE, UESCAPE, UNION,
UNNEST, USING, VALUES, WHEN, WHERE, WITH
```

## 含有預留關鍵字的查詢範例

下列範例中的查詢使用反引號 (`) ，以逸出作為資料表名稱和其中一個欄名稱的 DDL 相關預留關鍵字 `partition` 和 `date` :

```
CREATE EXTERNAL TABLE `partition` (
  `date` INT,
  col2 STRING
)
PARTITIONED BY (year STRING)
STORED AS TEXTFILE
LOCATION 's3://DOC-EXAMPLE-BUCKET/test_examples/';
```

以下範例查詢在 `ALTER TABLE ADD PARTITION` 和 `ALTER TABLE DROP PARTITION` 陳述式中，有一個欄名稱包含 DDL 相關的預留關鍵字。這些 DDL 預留關鍵字以反引號 (`) 括住：

```
ALTER TABLE test_table
ADD PARTITION (`date` = '2018-05-14')
```

```
ALTER TABLE test_table
DROP PARTITION (`partition` = 'test_partition_value')
```

以下範例查詢在 `SELECT` 陳述式中包含預留關鍵字 (`end`) 作為識別符。此關鍵字以雙引號逸出：

```
SELECT *
FROM TestTable
WHERE "end" != nil;
```

以下範例查詢在 SELECT 陳述式中包含預留關鍵字 (first)。此關鍵字以雙引號逸出：

```
SELECT "itemId"."first"  
FROM testTable  
LIMIT 10;
```

## Amazon S3 中的資料表位置

當您在 Athena 執行 CREATE TABLE 查詢時，Athena 會在資料目錄中註冊您的 AWS Glue 資料表，資料目錄是 Athena 儲存中繼資料的地方。

若要在 Amazon S3 中指定資料的路徑，請使用 LOCATION 屬性，如下列範例所示：

```
CREATE EXTERNAL TABLE `test_table`(  
...  
)  
ROW FORMAT ...  
STORED AS INPUTFORMAT ...  
OUTPUTFORMAT ...  
LOCATION s3://DOC-EXAMPLE-BUCKET/folder/
```

- 如需有關命名儲存貯體的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[儲存貯體限制](#)。
- 如需有關在 Amazon S3 中使用資料夾的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用資料夾](#)。

Amazon S3 中的 LOCATION 會指定代表您資料表的所有檔案。

### Important

Athena 會讀取在您指定的 Amazon S3 資料夾中存放的所有資料。如果您有不希望 Athena 讀取的資料，請勿使用與您希望 Athena 讀取之資料相同的 Amazon S3 資料夾存放該資料。若您使用的是分割，為確保 Athena 掃描分割區中的資料，WHERE 篩選條件必須包含分割區。如需詳細資訊，請參閱 [資料表位置和分割區](#)。

當您在 CREATE TABLE 陳述式中指定 LOCATION 時，請使用下列指導方針：

- 使用後置斜線。

- 您可以使用 Amazon S3 資料夾的路徑或 Amazon S3 存取點別名。如需有關 Amazon S3 存取點別名的資訊，請參閱《Amazon S3 使用者指南》中的[為存取點使用儲存貯體型別名](#)。

使用：

```
s3://DOC-EXAMPLE-BUCKET/folder/
```

```
s3://DOC-EXAMPLE-BUCKET-metadata-s3alias/folder/
```

請勿使用以下任何項目指定您資料的 LOCATION。

- 請勿使用檔案名稱、底線、萬用字元或 glob 模式來指定檔案位置。
- 請勿將完整 HTTP 標記法 (例如 `s3.amazonaws.com`) 新增至 Amazon S3 儲存貯體路徑。
- 請勿在路徑中使用 `//` 等使用空的資料夾，如下所示：`S3://DOC-EXAMPLE-BUCKET/folder//folder/`。雖然這是有效的 Amazon S3 路徑，但 Athena 不允許如此，而且會將其變更為 `s3://DOC-EXAMPLE-BUCKET/folder/folder/`，並移除額外的 `/`。

請勿使用。

```
s3://DOC-EXAMPLE-BUCKET
s3://DOC-EXAMPLE-BUCKET/*
s3://DOC-EXAMPLE-BUCKET/mySpecialFile.dat
s3://DOC-EXAMPLE-BUCKET/prefix/filename.csv
s3://DOC-EXAMPLE-BUCKET.s3.amazonaws.com
S3://DOC-EXAMPLE-BUCKET/prefix//prefix/
arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix
s3://arn:aws:s3:<region>:<account_id>:accesspoint/<accesspointname>
https://<accesspointname>-<number>.s3-accesspoint.<region>.amazonaws.com
```

## 資料表位置和分割區

可以根據一組資料欄，將來源資料分組至名為 `partitions` 的 Amazon S3 資料夾。例如，這些欄可代表建立特定記錄的年、月、日。

當您建立資料表時，您可以選擇為其建立分割區。Athena 針對未分割的資料表執行 SQL 查詢時，它會使用資料表定義的 `LOCATION` 屬性作為要列出的基礎路徑，並掃描所有可用的檔案。但是，在查詢分區資料表之前，您必須使用分區資訊更新「資 AWS Glue 料目錄」。此資訊表示特定分割區中的檔案結構描述，以及分割區在 Amazon S3 中檔案的 `LOCATION`。

- 若要瞭解 AWS Glue 爬行者程式如何新增分割區，請參閱[爬行者程式如何判斷何時建立分割區？](#)在 AWS Glue 開發人員指南中。
- 若要了解如何設定爬蟲程式，使其在現有分割區中為資料建立資料表，請參閱[對爬蟲程式使用多個資料來源](#)。
- 您也可以直接在 Athena 中直接於資料表內建立分割區。如需詳細資訊，請參閱 [在 Athena 中分割資料](#)。

當 Athena 在分割資料表上執行查詢時，它會檢查是否已在查詢的 WHERE 子句中使用任何分割資料欄。如果使用分割資料行，Athena 會要求 AWS Glue 資料目錄傳回符合指定分割區資料行的分割區規格。該分割區規格包含 LOCATION 屬性，其會告知 Athena 在讀取資料時需要使用哪一個 Amazon S3 字首。在這種情況下，「只會」掃描存放在此字首中的資料。如果您未在 WHERE 子句中使用分割的資料欄，Athena 會掃描屬於資料表分割區的所有檔案。

如需使用 Athena 進行分割以改善查詢效能並降低查詢成本的範例，請參閱 [Amazon Athena 的十大效能調整秘訣](#)。

## 單欄式儲存格式

[Apache 實木複合地板](#) 和 [ORC](#) 是針對快速檢索數據進行了優化，並在分析應用中 AWS 使用的柱狀存儲格式。

單欄式儲存格式具有以下特點，使其適用於 Athena：

- 針對資料欄資料類型選擇壓縮演算法來壓縮欄，可以節省 Amazon S3 中的儲存空間，並在查詢處理期間降低磁碟空間和輸入/輸出。
- Parquet 和 ORC 中的述詞下推，可讓 Athena 查詢僅擷取需要的區塊，進而提高查詢效能。當 Athena 查詢從您的資料中取得特定資料欄值時，它會使用資料區塊述詞 (例如上限/下限值) 的統計資料，判斷要讀取或略過該區塊。
- Parquet 和 ORC 中的資料分割，可讓 Athena 將資料的讀取分割給多個讀取器，在其查詢處理期間增加平行處理。

若要將現有的原始資料從其他儲存格式轉換為 Parquet 或 ORC，您可以在 Athena 執行[建立表格 AS SELECT \(CTAS\)](#) 查詢，並將資料儲存格式指定為實木地板或 ORC，或使用爬蟲程式。AWS Glue

## 在 Parquet 與 ORC 之間進行選擇

ORC (優化列單欄式) 和 Parquet 之間的選擇取決於您的特定使用需求。

Apache Parquet 提供高效的資料壓縮和編碼機制，非常適合用於執行複雜查詢和處理大量資料。為與 [Apache Arrow](#) 搭配使用，Parquet 進行了優化，因此如果您使用與 Arrow 相關的工具，Parquet 可能比較有利。

ORC 可以很有效率地存放 Hive 資料。ORC 檔案通常比 Parquet 檔案要小，因此 ORC 索引可以加快查詢速度。此外，ORC 支援複雜類型，例如結構、映射和清單。

在 Parquet 或 ORC 間選擇時，請考慮以下因素：

查詢效能 – 由於 Parquet 能夠支援更廣泛的查詢類型，因此，如果您打算執行複雜的查詢，則 Parquet 可能是更好的選擇。

複雜的資料類型 – 如果您使用的是複雜的資料類型，那麼 ORC 可能是更好的選擇，因為它能支援更廣泛的複雜資料類型。

檔案大小 – 如果需要考慮磁碟空間，ORC 通常可產生較小的檔案，進而降低儲存成本。

壓縮 – Parquet 和 ORC 均可提供良好的壓縮，但最適合您的最佳格式主要取決於您的特定使用案例。

演進 – Parquet 和 ORC 均可支援支持結構描述變化，這意味著您可以隨時間新增、刪除或修改資料欄。

對於大數據應用程式來說，Parquet 和 ORC 都是不錯的選擇，但在選擇之前，請考慮您的案例需求。您可能希望對資料和查詢執行基準測試，以查看哪種格式更適合您的使用案例。

## 轉換為單欄式格式

若您將資料轉換為開放原始碼單欄式格式，例如 [Apache Parquet](#) 或 [ORC](#)，則可改善您的 Amazon Athena 查詢效能。

將 JSON 或 CSV 等來源資料輕鬆轉換為單欄式格式的選項，包括使用 [CREATE TABLE AS](#) 查詢或執行 AWS Glue 中的任務。

- 使用 CREATE TABLE AS (CTAS) 查詢，只需一個步驟即可將資料轉換為 Parquet 或 ORC。如需範例，請參閱 [CTAS 查詢的範例](#) 頁面上的 [範例：將查詢結果寫為不同格式](#)。
- 如需執行 AWS Glue 任務以將 CSV 資料轉換為 Parquet 的詳細資訊，請參閱 AWS 大數據部落格文章 [使用 AWS Glue 和 Amazon S3 建立資料湖基礎中的「將資料從 CSV 轉換為實木複合地板格式」一節](#)。AWS Glue 支持使用相同的技術將 CSV 數據轉換為 ORC，或者將 JSON 數據轉換為實木複合地板或 ORC。

## 在 Athena 中分割資料

您可以分割資料，以限制每個查詢所掃描的資料量，從而提高效能和降低成本。您可透過任何索引鍵來分割您的資料。常見做法是根據時間來分割資料，這通常會產生多層級的分割機制。例如，每小時都有資料傳入的客戶可能決定依年、月、日、小時來分割。另一個客戶的資料來自許多不同來源，且每天只載入一次，則可能依資料來源識別符和日期來分割。

Athena 可以使用 Apache Hive 樣式的分割區，其資料路徑包含由等號連接的鍵值對 (例如 `country=us/...` 或 `year=2021/month=01/day=26/...`)。因此，路徑會包含分割區索引鍵的名稱，以及每個路徑所代表的值。若要將新的 Hive 分割區載入已分割的資料表，您可以使用 [MSCK REPAIR TABLE](#) 命令，其只適用於 Hive 樣式的分割區。

Athena 也可以使用非 Hive 樣式的分割區結構描述。例如，CloudTrail 記錄檔和 Firehose 傳送串流會針對日期部分使用不同的路徑元件，例如 `data/2021/01/26/us/6fc7845e.json`。對於這類非 Hive 樣式的分割區，您可以使用 [ALTER TABLE ADD PARTITION](#) 手動新增分割區。

### 考量與限制

使用分割區時，請謹記以下幾點：

- 若您查詢已分割的資料表並在 WHERE 子句中指定分割區，Athena 便只會掃描該分割區中的資料。如需詳細資訊，請參閱 [資料表位置和分割區](#)。
- 如果您對 Amazon S3 儲存貯體發出具有大量物件的查詢，並且未將資料分割，這類查詢可能會影響 Amazon S3 中的 GET 請求率限制，並導致 Amazon S3 例外狀況。若要避免錯誤，請分割您的資料。此外，請考慮調校您的 Amazon S3 請求率。如需詳細資訊，請參閱 [最佳實務設計模式：最佳化 Amazon S3 效能](#)。
- 要搭配 Athena 使用的分割區位置必須使用 s3 通訊協定 (例如，`s3://DOC-EXAMPLE-BUCKET/folder/`)。在 Athena 中，當在包含的資料表上執行 MSCK REPAIR TABLE 查詢時，使用其他通訊協定的位置 (例如 `s3a://DOC-EXAMPLE-BUCKET/folder/`) 會導致查詢失敗。
- 請確定 Amazon S3 路徑是小寫而不是小駝峰式命名法 (camel case) (例如，`userid` 而非 `userId`)。如果 S3 路徑是小駝峰式命名法 (camel case)，則 MSCK REPAIR TABLE 不會將分割區新增到 AWS Glue Data Catalog。如需詳細資訊，請參閱 [MSCK REPAIR TABLE](#)。
- 由於 MSCK REPAIR TABLE 會同時掃描資料夾及其子資料夾，以尋找相符的分割區配置，請務必將個別資料表的資料留在不同的資料夾階層中。例如，假設您在中有表格 1 的資料 `s3://DOC-EXAMPLE-BUCKET1`，在中有表格 2 的資料 `s3://DOC-EXAMPLE-BUCKET1/table-2-data`。如果兩個資料表都依字串分割，則 MSCK REPAIR TABLE 會將資料表 2 的分割區新增至資料表 1。為了避免這種情況，請 `s3://DOC-EXAMPLE-BUCKET2` 改用單獨的 `s3://DOC-EXAMPLE-BUCKET1` 文件夾結構。請注意，此行為與 Amazon EMR 和 Apache Hive 一致。

- 如果您 AWS Glue Data Catalog 搭配 Athena 使用，請參閱每個帳戶和每個表格分割區的[AWS Glue 端點和配額](#)。
- 雖然 Athena 支援查詢具有 1000 萬個分割區的 AWS Glue 資料表，但是 Athena 在一次掃描中無法讀取超過 100 萬個分割區。在這種情況下，分割區索引可能有利。如需詳細資訊，請參閱 AWS 大數據部落格文章[使用 AWS Glue Data Catalog 分割索引改善 Amazon Athena 查詢效能](#)。
- 若要在使用時要求增加分割區配額 AWS Glue Data Catalog，請造訪的 [Service Quotas 主控台 AWS Glue](#)。

## 建立和載入含有分割資料的資料表

若要建立使用分割區的資料表，請在 [CREATE TABLE](#) 陳述式中使用 PARTITIONED BY 子句。PARTITIONED BY 子句定義對資料進行分割區的索引鍵，如下列範例所示。LOCATION 子句指定已分割的資料的根位置。

```
CREATE EXTERNAL TABLE users (  
  first string,  
  last string,  
  username string  
)  
PARTITIONED BY (id string)  
STORED AS parquet  
LOCATION 's3://DOC-EXAMPLE-BUCKET'
```

建立資料表之後，您可以在分割區中載入要查詢的資料。對於 Hive 樣式的分割區，您可以執行 [MSCK REPAIR TABLE](#)。對於非 Hive 樣式的分割區，您可以使用 [ALTER TABLE ADD PARTITION](#) 手動新增分割區。

## 準備 Hive 樣式和非 Hive 樣式資料進行查詢

下列區段說明如何準備 Hive 樣式和非 Hive 樣式資料，以便在 Athena 中進行查詢。

### 案例 1：以 Hive 格式存放在 Amazon S3 上的資料

在此案例中，分割區存放在 Amazon S3 中的單獨資料夾。例如，以下是 [aws s3 ls](#) 命令輸出的廣告曝光範例的部分清單，其中列出了指定字首下的 S3 物件：

```
aws s3 ls s3://elasticmapreduce/samples/hive-ads/tables/impressions/
```

```
PRE dt=2009-04-12-13-00/  
PRE dt=2009-04-12-13-05/  
PRE dt=2009-04-12-13-10/  
PRE dt=2009-04-12-13-15/  
PRE dt=2009-04-12-13-20/  
PRE dt=2009-04-12-14-00/  
PRE dt=2009-04-12-14-05/  
PRE dt=2009-04-12-14-10/  
PRE dt=2009-04-12-14-15/  
PRE dt=2009-04-12-14-20/  
PRE dt=2009-04-12-15-00/  
PRE dt=2009-04-12-15-05/
```

在此，存放的日誌將欄名稱 (dt) 設為等於日期、小時和分鐘增量。當您在 DDL 中提供父資料夾的位置、結構描述和分割欄的名稱時，Athena 就能在這些子資料夾中查詢資料。

## 建立資料表

若要使用此資料建立資料表，請沿著 'dt' 建立分割區，如下列 Athena DDL 陳述式所示：

```
CREATE EXTERNAL TABLE impressions (  
    requestBeginTime string,  
    adId string,  
    impressionId string,  
    referrer string,  
    userAgent string,  
    userCookie string,  
    ip string,  
    number string,  
    processId string,  
    browserCookie string,  
    requestEndTime string,  
    timers struct<modelLookup:string, requestTime:string>,  
    threadId string,  
    hostname string,  
    sessionId string)  
PARTITIONED BY (dt string)  
ROW FORMAT serde 'org.apache.hive.hcatalog.data.JsonSerDe'  
LOCATION 's3://elasticmapreduce/samples/hive-ads/tables/impressions/' ;
```

此資料表使用 Hive 的原生 JSON 序列化程式-還原序列化程式，讀取存放在 Amazon S3 中的 JSON 資料。如需支援格式的詳細資訊，請參閱[支援的 SerDes 和資料格式](#)。

## 執行 MSCK REPAIR TABLE

在您執行 CREATE TABLE 查詢後，在 Athena 查詢編輯器中執行 MSCK REPAIR TABLE 命令以載入分割區，如下列範例所示。

```
MSCK REPAIR TABLE impressions
```

執行此命令後，即可對資料進行查詢。

### 查詢資料

使用分割區欄位查詢 impressions 資料表的資料。範例如下：

```
SELECT dt,impressionid FROM impressions WHERE dt<'2009-04-12-14-00' and
dt>='2009-04-12-13-00' ORDER BY dt DESC LIMIT 100
```

此查詢應該會顯示類似下列的資料：

```
2009-04-12-13-20    ap3HcVKAWfXtgIPu6WpuUfAfL0DQEc
2009-04-12-13-20    17uchtodoS9kdeQP1x0XThK15IuRsV
2009-04-12-13-20    J0Uf1SCtRwviGw8sVcghqE5h0nkgtp
2009-04-12-13-20    NQ2XP0J0dvVbCXJ0pb4XvqJ5A4QxxH
2009-04-12-13-20    fFAItiBMsgqro9kRdIwbeX60SR0axr
2009-04-12-13-20    V4og4R9W6G3QjHHwF7gI1cSqig5D1G
2009-04-12-13-20    hPEPtBwk45msmwWTxPVVo1kVu4v11b
2009-04-12-13-20    v0SkfxegheD90gp31UCr6Fp1nKpx6i
2009-04-12-13-20    1iD9odVg0Ii4QWkwHMc0hmwTkWDFj
2009-04-12-13-20    b31tJiIA25CK8eDHQrHnbcknfSndUk
```

### 案例 2：資料未分割 (Hive 格式)

在下列範例中，aws s3 ls 命令顯示存放在 Amazon S3 中的 [ELB](#) 日誌。請注意，資料配置如何不使用 key=value 對，因此不屬於 Hive 格式。(aws s3 ls 命令的 --recursive 選項指定列出指定目錄或字首下的所有檔案或物件。)

```
aws s3 ls s3://athena-examples-myregion/elb/plaintext/ --recursive
```

```
2016-11-23 17:54:46    11789573 elb/plaintext/2015/01/01/part-r-00000-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46    8776899  elb/plaintext/2015/01/01/part-r-00001-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
```

```
2016-11-23 17:54:46 9309800 elb/plaintext/2015/01/01/part-r-00002-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 9412570 elb/plaintext/2015/01/01/part-r-00003-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 10725938 elb/plaintext/2015/01/01/part-r-00004-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46 9439710 elb/plaintext/2015/01/01/part-r-00005-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 0 elb/plaintext/2015/01/01_$folder$
2016-11-23 17:54:47 9012723 elb/plaintext/2015/01/02/part-r-00006-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 7571816 elb/plaintext/2015/01/02/part-r-00007-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 9673393 elb/plaintext/2015/01/02/part-r-00008-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 11979218 elb/plaintext/2015/01/02/part-r-00009-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 9546833 elb/plaintext/2015/01/02/part-r-00010-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 10960865 elb/plaintext/2015/01/02/part-r-00011-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 0 elb/plaintext/2015/01/02_$folder$
2016-11-23 17:54:48 11360522 elb/plaintext/2015/01/03/part-r-00012-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 11211291 elb/plaintext/2015/01/03/part-r-00013-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 8633768 elb/plaintext/2015/01/03/part-r-00014-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 11891626 elb/plaintext/2015/01/03/part-r-00015-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 9173813 elb/plaintext/2015/01/03/part-r-00016-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 11899582 elb/plaintext/2015/01/03/part-r-00017-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 0 elb/plaintext/2015/01/03_$folder$
2016-11-23 17:54:50 8612843 elb/plaintext/2015/01/04/part-r-00018-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 10731284 elb/plaintext/2015/01/04/part-r-00019-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 9984735 elb/plaintext/2015/01/04/part-r-00020-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 9290089 elb/plaintext/2015/01/04/part-r-00021-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
```

```
2016-11-23 17:54:50 7896339 elb/plaintext/2015/01/04/part-r-00022-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8321364 elb/plaintext/2015/01/04/part-r-00023-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 0 elb/plaintext/2015/01/04_$$folder$
2016-11-23 17:54:51 7641062 elb/plaintext/2015/01/05/part-r-00024-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 10253377 elb/plaintext/2015/01/05/part-r-00025-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8502765 elb/plaintext/2015/01/05/part-r-00026-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 11518464 elb/plaintext/2015/01/05/part-r-00027-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 7945189 elb/plaintext/2015/01/05/part-r-00028-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 7864475 elb/plaintext/2015/01/05/part-r-00029-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 0 elb/plaintext/2015/01/05_$$folder$
2016-11-23 17:54:51 11342140 elb/plaintext/2015/01/06/part-r-00030-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8063755 elb/plaintext/2015/01/06/part-r-00031-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9387508 elb/plaintext/2015/01/06/part-r-00032-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9732343 elb/plaintext/2015/01/06/part-r-00033-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 11510326 elb/plaintext/2015/01/06/part-r-00034-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9148117 elb/plaintext/2015/01/06/part-r-00035-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 0 elb/plaintext/2015/01/06_$$folder$
2016-11-23 17:54:52 8402024 elb/plaintext/2015/01/07/part-r-00036-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 8282860 elb/plaintext/2015/01/07/part-r-00037-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 11575283 elb/plaintext/2015/01/07/part-r-00038-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 8149059 elb/plaintext/2015/01/07/part-r-00039-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 10037269 elb/plaintext/2015/01/07/part-r-00040-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 10019678 elb/plaintext/2015/01/07/part-r-00041-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 0 elb/plaintext/2015/01/07_$$folder$
```

```
2016-11-23 17:54:53      0 elb/plaintext/2015/01_$$folder$
2016-11-23 17:54:53      0 elb/plaintext/2015_$$folder$
```

## 執行 ALTER TABLE ADD PARTITION

因為資料不是 Hive 格式，您無法在建立資料表後，使用 MSCK REPAIR TABLE 命令將分割區新增至該資料表。相反，您可以使用 [ALTER TABLE ADD PARTITION](#) 命令以手動新增每個分割區。例如，如需載入 `s3://athena-examples-myregion/elb/plaintext/2015/01/01/` 中的資料，您可以執行下列查詢：請注意，每個 Amazon S3 資料夾不需要單獨的分割區資料欄，且分割區鍵值可與 Amazon S3 索引鍵不同。

```
ALTER TABLE elb_logs_raw_native_part ADD PARTITION (dt='2015-01-01') location 's3://athena-examples-us-west-1/elb/plaintext/2015/01/01/'
```

如果已具有分割區，您會收到 Partition already exists (分割區已存在) 錯誤。若要避免發生此錯誤，您可以使用 IF NOT EXISTS 子句。如需詳細資訊，請參閱 [ALTER TABLE ADD PARTITION](#)。如需移除分割區，請使用 [ALTER TABLE DROP PARTITION](#)。

## 分割區投影

若要避免必須管理分割區，您可以使用分割區投影。分割區投影是高度已分割的資料表，其結構是提前已知的選項。在分割區投影中，分割區的值和位置是以您所設定的資料表屬性計算，而不是從中繼資料儲存庫所讀取到的來計算。因為記憶體中的計算是比遠端查詢更快，分割區投影的使用可以大幅減少查詢執行時間。

如需詳細資訊，請參閱 [使用 Amazon Athena 進行分割區投影](#)。

## 其他資源

- 如需 Firehose 資料的分割選項的詳細資訊，請參閱 [Amazon 數據 Firehose 示例](#)。
- 您可以使用 [JDBC 驅動程式](#) 來自動新增分割區。
- 您可以使用 CTAS 和 INSERT INTO 來分割資料集。如需更多詳細資訊，請參閱 [使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析](#)。

## 使用 Amazon Athena 進行分割區投影

您可以使用 Athena 中的分割區投影來加速高度分割資料表的查詢處理，以及自動化分割區管理。

在分割區投影中，Athena 會使用您直接在 AWS Glue 中的資料表設定的資料表屬性來計算分割區的值和位置。資料表屬性可讓 Athena 「投影」或判斷必要的分割區資訊，而不必在 AWS Glue Data Catalog 中執行耗時的中繼資料查詢。因為記憶體內操作通常比遠端操作更快，所以分割區投影可以減少對高度分割資料表的查詢執行期。取決於查詢和基礎資料的特定特性，分割區投影可能大幅減少受限於分割區中繼資料擷取的查詢執行期。

## 重度分割資料表的清除和投影

分割區清除會收集中繼資料，並將它「修剪」至只限適用於查詢的分割區。這通常可加快查詢速度。Athena 會對所有具有分割區資料欄的資料表使用分割區剔除，包括為分割區投影設定的資料表。

一般而言，在處理查詢時，Athena 會 AWS Glue Data Catalog 在執行分割區修剪之前先 GetPartitions 呼叫。如果資料表有大量分割區，使用 GetPartitions 可能會對效能造成負面影響。若要避免這種情況，您可以使用分割區投影。分割區投影讓 Athena 可以避免呼叫 GetPartitions，因為分割區投影組態可提供 Athena 建置分割區本身的全部所需資訊。

## 使用分割區投影

若要使用資料分割投影，您可以在 AWS Glue Data Catalog 或 [外部 Hive 中繼存放區](#) 的表格屬性中，為每個分割區資料行指定分割區值和投影類型的範圍。資料表上的這些自訂屬性可讓 Athena 知道在資料表上執行查詢時預期的分割區模式。在查詢執行期間，Athena 會使用此資訊來投影分割區值，而不是從 AWS Glue Data Catalog 或外部 Hive 中繼存放區擷取它們。這不僅可以減少查詢執行期，還可以自動化分割區管理，因為它不需要在 Athena、AWS Glue 或外部 Hive 中繼存放區中手動建立分割區。

### Important

在資料表上啟用分割區投影會導致 Athena 忽略任何註冊至 AWS Glue Data Catalog 或 Hive 中繼資料表的分割區中繼資料。

## 使用案例

適合使用分割區投影的案例包括下列：

- 針對高度分割資料表的查詢無法如您所願的那麼快完成。
- 隨著在資料中建立新的日期或時間分割區，您定期將分割區新增至資料表。使用分割區投影，您可以設定相對日期範圍，以便在新資料到達時使用。
- 您在 Amazon S3 中有高度分割的資料。在 AWS Glue Data Catalog 或 Hive 中繼存放區中建模資料是不切實際的，而您的查詢只會讀取其中的一小部分。

## 可投影的分割區結構

當您的分割區遵循可預測的模式 (例如，但不限於下列項目) 時，分割區投影最容易設定：

- 整數 – 任何連續的整數序列，例如 [1, 2, 3, 4, ..., 1000] 或 [0500, 0550, 0600, ..., 2500]。
- 日期 – 任何連續的日期或日期時間序列，例如 [20200101, 20200102, ..., 20201231] 或 [1-1-2020 00:00:00, 1-1-2020 01:00:00, ..., 12-31-2020 23:00:00]。
- 列舉值 — 一組有限的列舉值，例如機場代碼或 AWS 區域
- AWS 服務 log — 記 AWS 服務 錄檔通常具有已知的結構，您可以在其中指定其資料分割配置 AWS Glue，因此 Athena 可用於分割區投影。

## 自訂分割區路徑範本

依預設，Athena 會使用格式 `s3://DOC-EXAMPLE-BUCKET/<table-root>/partition-col-1=<partition-col-1-val>/partition-col-2=<partition-col-2-val>/` 建置分割區位置，但如果您的資料以不同的方式組織，Athena 也提供自訂此路徑範本的機制。如需這些步驟，請參閱 [指定自訂 S3 儲存位置](#)。

## 考量與限制

適用下列注意事項：

- 分割區投影無須在 AWS Glue 或外部 Hive 中繼存放區手動指定分割區。
- 當您在資料表上啟用分割區投影時，Athena 會忽略該資料表 AWS Glue Data Catalog 或外部 Hive 中繼資料中繼資料中的任何分區中繼資料。
- 如果投影的分割區不存在於 Amazon S3 中，Athena 仍會投影該分割區。Athena 不會擲回錯誤，但也不會傳回任何資料。但是，如果太多的分割區是空的，相較於傳統的分 AWS Glue 割區，效能可能會較慢。如果超過一半的投影分割區是空的，建議您使用傳統分割區。
- 如果數值超過分割區投影定義的範圍邊界，查詢不會傳回錯誤。查詢反而會執行，但不會傳回任何資料行。例如，如果您擁有自 2020 年起並定義為 `'projection.timestamp.range'='2020/01/01,NOW'` 的時間相關資料，如 `SELECT * FROM table-name WHERE timestamp = '2019/02/02'` 等的查詢將成功完成，但不會傳回任何資料行。
- 分割區投影只適用於透過 Athena 查詢資料表時。如果透過其他服務 (例如 Amazon Redshift Spectrum、Athena for Spark 或 Amazon EMR) 讀取相同的資料表，則會使用標準的分割區中繼資料。

- 由於分割區投影是僅限 DML 的功能，因此 SHOW PARTITIONS 不會列出 Athena 投影但未在 AWS Glue 目錄或外部 Hive 中繼存放區中註冊的分割區。
- Athena 未將檢視的資料表屬性作為分割區投影的組態。若要解決此限制，請在檢視參照之資料表的資料表屬性中設定並啟用分割區投影。
- 在 Athena 引擎第 2 版中，Lake Formation [資料篩選器](#) 無法與分區投影搭配使用。

## 影片

以下影片展示如何使用分割區投影，改善在 Athena 中的查詢效能。

### [使用 Amazon Athena 進行分割區投影](#)

#### 主題

- [設定分割區投影](#)
- [支援的分割區投影類型](#)
- [動態 ID 分割](#)
- [Amazon 數據 Firehose 示例](#)

## 設定分割區投影

在資料表屬性中設定分割區投影共有兩個步驟：

1. 為每個分割區資料欄指定資料範圍和相關模式，或使用自訂範本。
2. 啟用資料表的分割區投影。

#### Note

在您將分割區投影屬性新增至現有的資料表之前，您要設定分割區投影屬性的分割區資料欄必須已經存在於資料表結構描述中。如果分割區資料行尚不存在，您必須手動將分割區資料欄新增至現有的資料表。AWS Glue 不會自動為您執行此步驟。

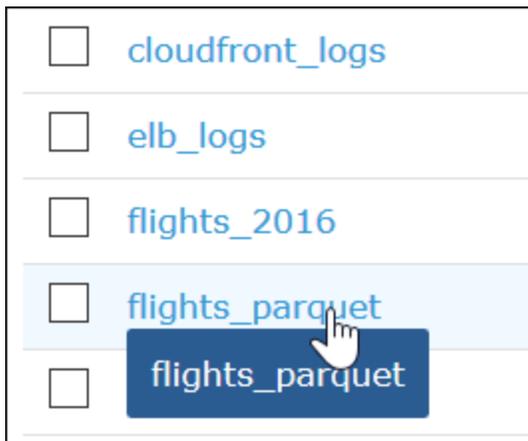
本節說明如何設定的表格屬性 AWS Glue。若要進行設定，您可以使用 AWS Glue 主控台、Athena [CREATE TABLE](#) 查詢或 [AWS Glue API](#) 作業。下列程序顯示如何在 AWS Glue 主控台中設定屬性。

## 使用 AWS Glue 控制台配置和啟用分割區投影

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，網址為 <https://console.aws.amazon.com/glue/>。
2. 選擇 Tables (資料表) 索引標籤。

在 Tables (資料表) 索引標籤上，您可以編輯現有資料表，或選擇 Add tables (新增資料表) 以建立新的資料表。如需有關手動或使用爬蟲程式新增資料表的資訊，請參閱《AWS Glue 開發人員指南》中的 [在 AWS Glue 主控台上使用資料表](#)。

3. 在資料表清單中，選擇您要編輯之資料表的連結。



4. 選擇 Actions (動作)、Edit table (編輯資料表)。
5. 在 Edit table (編輯資料表) 頁面的 Table properties (資料表屬性) 區段中，針對每個分割的資料欄新增下列索引鍵/值組：
  - a. 對於 Key (索引鍵)，新增 `projection.columnName.type`。
  - b. 對於 Value (值)，新增其中一個支援的類型：`enum`、`integer`、`date` 或 `injected`。如需詳細資訊，請參閱 [支援的分割區投影類型](#)。
6. 遵循 [支援的分割區投影類型](#) 中的指導，根據您的組態需求新增額外的索引鍵/值組。

下列範例資料表組態會設定分割區投影的 `year` 資料欄，限制可傳回值的範圍為 2010 年到 2016 年。

## Edit table details

**Table name**  
flights\_parquet

**Input format**  
org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat

**Output format**

**Table properties**

Key	Value	
last_modified_time	1582588443	×
EXTERNAL	TRUE	×
last_modified_by	hadoop	×
projection.year.type	integer	×
projection.year.range	2010,2016	×
		×

7. 新增索引鍵/值組以啟用分割區投影。對於 Key (索引鍵)，輸入 `projection.enabled`，對於其 Value (值)，輸入 `true`。

**Note**

您可以透過將 `projection.enabled` 設定為 `false`，隨時停用此資料表上的分割區投影。

8. 完成時，請選擇 **Save** (儲存)。
9. 在 Athena 查詢編輯器中，測試查詢您為資料表設定的資料欄。

下列範例查詢使用 `SELECT DISTINCT` 從 `year` 資料欄傳回唯一值。資料庫包含 1987 年到 2016 年的資料，但 `projection.year.range` 屬性將傳回的值限制為 2010 年到 2016 年。

 **Query 1**

```
1 SELECT DISTINCT year FROM flights_parquet
2 ORDER BY year ASC
```

SQL Ln 2, Col 18

**Run again** Cancel Save as Clear

 **Completed**  
Time in queue: 0.25 sec Run time: 0.535 sec Data

**Results (7)**  **Copy**

year
2010
2011
2012
2013
2014
2015
2016

**Note**

如果您將 `projection.enabled` 設定為 `true` 但無法設定一或多個分割區資料欄，您會收到如下所示的錯誤訊息：

```
HIVE_METASTORE_ERROR: Table database_name.table_name is configured for partition projection, but the following partition columns are missing projection configuration: [column_name] (table database_name.table_name).
```

## 指定自訂 S3 儲存位置

當您在中編輯表格屬性時 AWS Glue，也可以為投影的分割區指定自訂 Amazon S3 路徑範本。自訂範本可讓 Athena 將分割區值正確地對應到不遵循典型 `.../column=value/...` 模式的自訂 Amazon S3 檔案位置。

使用自訂範本是選擇性的。不過，如果您使用自訂範本，則範本必須包含每個分割區資料欄的預留位置。範本位置必須以正斜線結尾，以便分割的資料檔案存在於每個分割區的「資料夾」中。

## 指定自訂分割區位置範本

1. 按照使用 [AWS Glue 控制台配置和啟用分區投影](#) 的步驟，添加一個額外的鍵值對來指定自定義模板，如下所示：
  - a. 在 Key (索引鍵) 欄位，輸入 `storage.location.template`。
  - b. 對於 Value (值)，指定包含每個分割區資料欄之預留位置的位置。請確定每個預留位置 (及 S3 路徑本身) 是以單一正斜線結尾。

下列範例範本值假設資料表具有分割區資料欄 a、b 和 c。

```
s3://DOC-EXAMPLE-BUCKET/table_root/a=${a}/${b}/some_static_subdirectory/${c}/
```

```
s3://DOC-EXAMPLE-BUCKET/table_root/c=${c}/${b}/some_static_subdirectory/${a}/${b}/${c}/${c}/
```

對於相同的資料表，下列範例範本值無效，因為它不包含資料欄 c 的預留位置。

```
s3://DOC-EXAMPLE-BUCKET/table_root/a=${a}/${b}/some_static_subdirectory/
```

2. 選擇 Apply (套用)。

## 支援的分割區投影類型

資料表可以具有 enum、integer、date，或 injected 分割區資料欄類型的任意組合。

### 列舉類型

對值為列舉集成員的分割區欄使用此enum類型 (例如，機場代碼或 AWS 區域)。

在資料表中定義分割區屬性，如下所示：

屬性名稱	範例值	描述
projection. <i>columnName</i> .type	enum	必要。用於資料欄 <i>columnName</i> 的投影類型。該值必須是 enum (不區分大小寫) 以表示使用列舉類型。允許開頭和結尾空格。
projection. <i>columnName</i> .values	A,B,C,D,E,F,G,Unknown	必要。用於資料欄 <i>columnName</i> 之以逗號分隔的列舉分割區值清單。任何空格都會被視為列舉值的一部分。

### Note

建議的最佳實務是將 enum 型分割區投影的使用限制成幾十個或以下。雖然沒有特定的預enum測限制，但是當 gzip 壓縮時，表格中繼資料的總大小不能超過約 1 MB 的 AWS Glue 限制。請注意，您資料表的主要部分會共用此限制，例如資料欄名稱、位置、儲存格式等。如果您發現自己在 enum 投影中使用幾十個以上的唯一 ID，請考慮使用替代方法，例如歸納為代理欄位中較少數的唯一數值。您可以透過取捨基數的方式，控制 enum 欄位中唯一數值的數量。

## 整數類型

對於其可能值會被解譯為定義範圍內整數的分割區資料欄，請使用整數類型。投影整數資料欄目前限制為帶正負號的 Java 長整數範圍 ( $-2^{63}$  至  $2^{63}-1$ ，含)。

屬性名稱	範例值	描述
<code>projection. <i>columnName</i>.type</code>	integer	必要。用於資料欄 <i>columnName</i> 的投影類型。該值必須是 integer (不區分大小寫) 以表示使用整數類型。允許開頭和結尾空格。
<code>projection. <i>columnName</i>.range</code>	0,10 -1,8675309 0001,9999	必要。以逗號分隔的雙元素清單，提供資料欄 <i>columnName</i> 上的查詢要傳回的最小和最大範圍值。請注意，值必須以逗號分隔，而非連字號。這些值包含在內，可以是負的，也可以以零開頭。允許開頭和結尾空格。
<code>projection. <i>columnName</i>.interval</code>	1 5	選用。正整數，指定用於資料欄 <i>columnName</i> 之連續分割區值之間間隔。例如，range 值 "1,3" 具有 "1" 的 interval 值，會產生 1、2 和 3 的值。range 值相同但 interval 值為 "2" 會產生 1 和 3 的值，略過 2。允許開頭和結尾空格。預設為 1。
<code>projection. <i>columnName</i>.digits</code>	1 5	選用。正整數，指定用於資料欄 <i>columnName</i> 之分割區值最終表示法中要包含的位數。例如，range 值 "1,3" 具有 "1" 的 digits 值，會產生 1、2 和 3 的值。range 值相

屬性名稱	範例值	描述
		同但 <code>digits</code> 值為 "2" 會產生 01、02 和 03 的值。允許開頭和結尾空格。預設值是非靜態位數，且不以零開頭。

## 日期類型

對於其值會被解譯為定義範圍內之日期 (以及選用的時間) 的分割區資料欄，請使用整數類型。

### Important

投影的日期資料欄會在查詢執行期以國際標準時間 (UTC) 產生。

屬性名稱	範例值	描述
<code>projection. <i>columnName</i>.type</code>	<code>date</code>	必要。用於資料欄 <code>columnName</code> 的投影類型。該值必須是 <code>date</code> (不區分大小寫) 以表示使用日期類型。允許開頭和結尾空格。
<code>projection. <i>columnName</i>.range</code>	<code>201701,201812</code> <code>01-01-2010,12-31-2018</code> <code>NOW-3YEARS,NOW</code> <code>201801,NOW+1MONTH</code>	必要。以逗號分隔的雙元素清單，提供資料欄 <code>columnName</code> 的最小和最大 <code>range</code> 值。這些數值包含在內，可以使用與 Java <code>java.time.*</code> 日期類型相容的任何格式。最小值和最大值都必須使用相同的格式。 <code>.format</code> 屬性中指定的格式必須是用於這些值的格式。  此資料欄也可以包含相對日期字串，格式化為此規則運算式模式： <pre>\s*NOW\s*(([\+ -])\s*([0-9]+)\s*(YEARS? MONTHS? WEEKS? DAYS? HOURS? MINUTES? SECONDS?)\s*)?</pre>

屬性名稱	範例值	描述
		允許使用空格，但在日期常值中會被視為日期字串本身的一部分。
projection. <i>columnName</i> .format	yyyyMM dd-MM-yyyy y dd-MM-yyyy y-HH-mm-ss	必要。以 Java 日期格式為基礎的日期格式字串 <a href="#">DateTimeFormatter</a> 。可以是任何支援的 <code>Java.time.*</code> 類型。
projection. <i>columnName</i> .interval	1 5	<p>正整數，指定用於資料欄 <i>columnName</i> 之連續分割區值之間的時間隔。例如，range 值為 2017-01,2018-12 搭配 interval 值為 1 以及 interval.unit 值為 MONTHS，會產生值 2017-01、2017-02、2017-03 等。相同的 range 值搭配 interval 值為 2 且 interval.unit 值為 MONTHS，會產生值 2017-01、2017-03、2017-05 等。允許開頭和結尾空格。</p> <p>如果提供的日期採用單日或單月精確度，則 interval 是選用的，並且分別預設為 1 天或 1 個月。否則，interval 是必要的。</p>

屬性名稱	範例值	描述
projection. <i>columnName</i> .interval.unit	YEARS	代表 a 的序列化形式的時間單位字。 <a href="#">ChronoUnit</a> 可能值為 YEARS、MONTHS、WEEKS、DAYS、HOURS、MINUTES、或 MILLIS。這些值不區分大小寫。  如果提供的日期採用單日或單月精確度，則 interval.unit 是選用的，並且分別預設為 1 天或 1 個月。否則，interval.unit 是必要的。
	MONTHS	
	WEEKS	
	DAYS	
	HOURS	
	MINUTES	
	SECONDS	
	MILLIS	

## 注入類型

注入類型適用於其可能值不能在某個邏輯範圍內依程序產生，但可在查詢的 WHERE 子句中作為單一值提供的分割區資料欄。

請務必牢記以下幾點：

- 如果未為每個注入的資料欄提供篩選條件表達式，則注入資料欄上的查詢會失敗。
- 只有當值為分離值時，才能成功在注入資料欄上針對篩選條件表達式查詢多個值。
- 僅支援 string 類型的資料欄。

屬性名稱	Value	描述
projection. <i>columnName</i> .type	injected	必要。用於資料欄 <i>columnName</i> 的投影類型。僅支援 string 類型。指定的值必須是 injected (不區分大小寫)。允許開頭和結尾空格。

如需詳細資訊，請參閱[使用 injected 投影類型](#)。

## 動態 ID 分割

當您的資料以高基數屬性分割，或者無法事先知道這些值時，您可使用 `injected` 投影類型。此類屬性的範例包括使用者名稱，以及裝置或產品的 ID。當您使用 `injected` 投影類型來設定分割區索引鍵時，Athena 會使用查詢本身的值來計算將要讀取的分割區集。

對於具有使用 `injected` 投影類型設定的分割區索引鍵的資料表，若要讓 Athena 能夠在此類資料表上執行查詢，必須符合下列條件：

- 您的查詢必須包含至少一個分割區索引鍵值。
- 值必須是文字或運算式，可在不讀取任何資料的情況下進行評估。

如果不滿足上述任何條件，您的查詢會失敗，並出現下列錯誤：

約束\_違規：針對插入的投影資料分割資料行 `column_name`，WHERE 子句必須只包含靜態相等條件，且至少必須存在一個此類條件。

### 使用 `injected` 投影類型

假設您有一個資料集，其中包含 IoT 裝置的事件，並根據裝置的 ID 進行分割。此資料集具有下列特性：

- 裝置 ID 隨機產生。
- 新裝置會經常進行佈建。
- 目前有數十萬部裝置，未來將會有數百萬部裝置。

使用傳統的中繼儲存難以管理此資料集。在資料儲存與中繼儲存之間難以保持分割區同步，並且在查詢規劃期間篩選分割區可能較慢。但是，如果您將資料表設定為使用分割區投影，並使用 `injected` 投影類型，則有兩項優勢：您不必管理中繼儲存中的分割區，並且您的查詢也不必查詢分割區中繼資料。

下列 CREATE TABLE 範例會針對剛才描述的裝置事件資料集來建立資料表。該資料表使用注入的投影類型。

```
CREATE EXTERNAL TABLE device_events (  
    event_time TIMESTAMP,  
    data STRING,  
    battery_level INT  
)
```

```
PARTITIONED BY (  
  device_id STRING  
)  
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"  
TBLPROPERTIES (  
  "projection.enabled" = "true",  
  "projection.device_id.type" = "injected",  
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${device_id}"  
)
```

下列範例查詢會查詢 12 小時內，從三個特定裝置接收的事件數目。

```
SELECT device_id, COUNT(*) AS events  
FROM device_events  
WHERE device_id IN (  
  '4a770164-0392-4a41-8565-40ed8cec737e',  
  'f71d12cf-f01f-4877-875d-128c23cbde17',  
  '763421d8-b005-47c3-ba32-cc747ab32f9a'  
)  
AND event_time BETWEEN TIMESTAMP '2023-11-01 20:00' AND TIMESTAMP '2023-11-02 08:00'  
GROUP BY device_id
```

當您執行此查詢時，Athena 會看到 `device_id` 分割區索引鍵的三個值，並使用這些值來計算分割區位置。Athena 會使用 `storage.location.template` 屬性的值來產生下列位置：

- `s3://DOC-EXAMPLE-BUCKET/prefix/4a770164-0392-4a41-8565-40ed8cec737e`
- `s3://DOC-EXAMPLE-BUCKET/prefix/f71d12cf-f01f-4877-875d-128c23cbde17`
- `s3://DOC-EXAMPLE-BUCKET/prefix/763421d8-b005-47c3-ba32-cc747ab32f9a`

如果您從分割區投影組態中省略 `storage.location.template` 屬性，Athena 會根據 `LOCATION` 中的值 (例如 `s3://DOC-EXAMPLE-BUCKET/prefix/device_id=4a770164-0392-4a41-8565-40ed8cec737e`)，使用 HIVE 樣式的分割區來投影分割區位置。

## Amazon 數據 Firehose 示例

當您使用 Firehose 將資料傳送到 Amazon S3 時，預設組態會使用類似下列範例的金鑰寫入物件：

```
s3://DOC-EXAMPLE-BUCKET/prefix/yyyy/MM/dd/HH/file.extension
```

若要建立在查詢時自動尋找分割區的 Athena 資料表，您可以使用分割區投影，而不必在新資料送達時將它們新增至。AWS Glue Data Catalog

下列 CREATE TABLE 範例會使用預設的 Firehose 組態。

```
CREATE EXTERNAL TABLE my_ingested_data (  
  ...  
)  
  ...  
PARTITIONED BY (  
  datehour STRING  
)  
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"  
TBLPROPERTIES (  
  "projection.enabled" = "true",  
  "projection.datehour.type" = "date",  
  "projection.datehour.format" = "yyyy/MM/dd/HH",  
  "projection.datehour.range" = "2021/01/01/00,NOW",  
  "projection.datehour.interval" = "1",  
  "projection.datehour.interval.unit" = "HOURS",  
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${datehour}/"  
)
```

CREATE TABLE 陳述式中的 TBLPROPERTIES 子句告訴 Athena 以下內容：

- 查詢資料表時使用分割區投影
- 分割區索引鍵 datehour 的類型為 date (其中包括可選時間)
- 將日期格式化的方式
- 日期時間的範圍。請注意，值必須以逗號分隔，而非連字號。
- 可尋找 Amazon S3 上資料的位置。

當您查詢資料表時，Athena 會計算 datehour 的值，並使用儲存位置範本來產生分割區位置的清單。

### 使用 **date** 類型

當您將 date 類型用於投影分割區索引鍵時，您必須指定範圍。由於您沒有建立 Firehose 傳送串流之前的日期資料，因此您可以使用建立日期做為開始日期。並且因為您沒有未來日期的資料，您可以使用特殊字符 NOW 作為結束。

在 CREATE TABLE 範例中，開始日期指定為世界協調時間 (UTC) 2021 年 1 月 1 日午夜。

**Note**

設定與您的資料盡可能相符的範圍，以便 Athena 僅查詢現有分割區。

在範例資料表上執行查詢時，Athena 將 datehour 分割區索引鍵上的條件與範圍結合使用來產生值。請考慮下列查詢：

```
SELECT *
FROM my_ingested_data
WHERE datehour >= '2020/12/15/00'
AND datehour < '2021/02/03/15'
```

SELECT 查詢中的第一個條件會使用日期，該日期會在 CREATE TABLE 陳述句指定的日期範圍開始之前。由於分割區投影組態沒有為 2021 年 1 月 1 日之前的日期指定分割區，因此 Athena 僅在以下位置查詢資料，並忽略查詢中較早的日期。

```
s3://DOC-EXAMPLE-BUCKET/prefix/2021/01/01/00/
s3://DOC-EXAMPLE-BUCKET/prefix/2021/01/01/01/
s3://DOC-EXAMPLE-BUCKET/prefix/2021/01/01/02/
...
s3://DOC-EXAMPLE-BUCKET/prefix/2021/02/03/12/
s3://DOC-EXAMPLE-BUCKET/prefix/2021/02/03/13/
s3://DOC-EXAMPLE-BUCKET/prefix/2021/02/03/14/
```

同樣地，如果在 2021 年 2 月 3 日 15:00 之前的日期和時間執行查詢，則最後一個分割區會反映目前的日期和時間，而不是在查詢條件中的日期和時間。

如果要查詢最新資料，您可以利用 Athena 不產生未來日期並僅指定開頭 datehour 的事實，如下列範例所示。

```
SELECT *
FROM my_ingested_data
WHERE datehour >= '2021/11/09/00'
```

### 選擇分割區索引鍵

您可以指定分割區投影將分割區位置映射至分割區索引鍵的方式。在前一部份的 CREATE TABLE 範例中，日期和小時被合併成一個名為 datehour 的分割區索引鍵，但也可以使用其他結構描述。例如，您也可以針對年份、月份、日期及小時，設定個別分割區索引鍵的資料表。

但是，將日期分割為年、月和日，表示無法使用 date 分割區投影類型。另一種方法是將日期與小時分開，即可仍然利用 date 分割區投影類型，但讓指定小時範圍的查詢更易於閱讀。

請記住，以下 CREATE TABLE 範例將日期與小時分開。由於 date 是 SQL 中的保留字詞，因此此範例會用 day 表示日期之分割區索引鍵的名稱。

```
CREATE EXTERNAL TABLE my_ingested_data2 (  
  ...  
)  
  ...  
PARTITIONED BY (  
  day STRING,  
  hour INT  
)  
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"  
TBLPROPERTIES (  
  "projection.enabled" = "true",  
  "projection.day.type" = "date",  
  "projection.day.format" = "yyyy/MM/dd",  
  "projection.day.range" = "2021/01/01,NOW",  
  "projection.day.interval" = "1",  
  "projection.day.interval.unit" = "DAYS",  
  "projection.hour.type" = "integer",  
  "projection.hour.range" = "0,23",  
  "projection.hour.digits" = "2",  
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${day}/${hour}/"  
)
```

在範例 CREATE TABLE 陳述式中，小時是個別的分割區索引鍵並設定為整數。小時分割區索引鍵的組態會指定範圍 0 到 23，而當 Athena 產生分割區位置時，小時應該格式化為兩位數。

對 my\_ingested\_data2 資料表的查詢可能如下所示：

```
SELECT *  
FROM my_ingested_data2  
WHERE day = '2021/11/09'  
AND hour > 3
```

## 分割區索引鍵類型和分割區投影類型

請注意，第一個 CREATE TABLE 範例中的 datehour 索引鍵在分割區投影組態中設定為 date，但是分割區索引鍵的類型為 string。第二個範例中的 day 也是如此。分割區投影組態中的類型只會告訴 Athena 在產生分割區位置時如何將值格式化。您指定的類型不會變更分割區索引鍵的類型；在查詢中，datehour 和 day 為 string 類型。

當查詢包含類似 day = '2021/11/09' 的條件時，Athena 會使用分割區投影組態中指定的日期格式剖析表達式右側的字串。在 Athena 確認日期是否位於所設定範圍後，它會再次使用日期格式將日期作為字串插入儲存位置範本。

同樣，對於像 day > '2021/11/09' 的查詢條件，Athena 會剖析右側並產生設定範圍內所有相符日期的清單。然後，它會使用日期格式將每個日期插入儲存位置範本，以建立分割區位置的清單。

寫入與 day > '2021-11-09' 或 day > DATE '2021-11-09' 相同的條件不起作用。在第一種情況下，日期格式不相符 (請注意連字號而不是正斜線)，而在第二種情況下，資料類型不相符。

### 使用自訂字首和動態分割

Firehose 可以使用自定 [義前綴](#) 和 [動態](#) 分區進行配置。使用這些功能，您可以設定 Amazon S3 索引鍵，並設定能對您使用案例提供更佳支援的分割區結構描述。您還可以將分割區投影與這些分割結構描述搭配使用並相應地設定它們。

例如，您可以使用自訂字首功能來取得具有 ISO 格式化日期的 Amazon S3 索引鍵，而不是預設 yyyy/MM/dd/HH 結構描述。

您也可以將自訂前置詞與動態磁碟分割結合在一起，customer\_id 從 Firehose 訊息擷取屬性，如下列範例所示。

```
prefix/!{timestamp:yyyy}-!{timestamp:MM}-!{timestamp:dd}/!  
{partitionKeyFromQuery:customer_id}/
```

使用該 Amazon S3 前綴，Firehose 交付串流會將物件寫入金鑰，例如 s3://DOC-EXAMPLE-BUCKET/prefix/2021-11-01/customer-1234/file.extension。對於像 customer\_id 的屬性，其中值並未事先知道，您可以使用分割區投影類型 [injected](#) 並使用 CREATE TABLE 陳述式，如下所示：

```
CREATE EXTERNAL TABLE my_ingested_data3 (  
  ...
```

```

)
...
PARTITIONED BY (
  day STRING,
  customer_id STRING
)
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"
TBLPROPERTIES (
  "projection.enabled" = "true",
  "projection.day.type" = "date",
  "projection.day.format" = "yyyy-MM-dd",
  "projection.day.range" = "2021-01-01,NOW",
  "projection.day.interval" = "1",
  "projection.day.interval.unit" = "DAYS",
  "projection.customer_id.type" = "injected",
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${day}/${customer_id}/"
)

```

當您查詢具有類型 `injected` 的分割區索引鍵的資料表時，您的查詢必須包含該分割區索引鍵的值。對 `my_ingested_data3` 資料表的查詢可能如下所示：

```

SELECT *
FROM my_ingested_data3
WHERE day BETWEEN '2021-11-01' AND '2021-11-30'
AND customer_id = 'customer-1234'

```

## ISO 格式化的日期

因為 `day` 分割區索引鍵的值是 ISO 格式化，所以您也可以使用 `DATE` 類型而不是 `STRING` 來做為日期分割區索引鍵，如下列範例所示：

```

PARTITIONED BY (day DATE, customer_id STRING)

```

當您查詢時，此策略可讓您在分割區索引鍵上使用日期函數，而不需要剖析或轉換，如下列範例所示：

```

SELECT *
FROM my_ingested_data3
WHERE day > CURRENT_DATE - INTERVAL '7' DAY
AND customer_id = 'customer-1234'

```

**Note**

指定 DATE 類型的分割區索引鍵，假設您已使用[自訂字首](#)功能建立具有 ISO 格式化日期的 Amazon S3 索引鍵。如果您使用的預設 Firehose 格式 yyyy/MM/dd/HH，即使對應的 table 屬性為類型，您 string 也必須將分割區索引鍵指定為類型 date，如下列範例所示：

```
PARTITIONED BY (  
  `mydate` string)  
TBLPROPERTIES (  
  'projection.enabled'='true',  
  ...  
  'projection.mydate.type'='date',  
  'storage.location.template'='s3://DOC-EXAMPLE-BUCKET/prefix/${mydate}')
```

## 從查詢結果建立資料表 (CTAS)

CREATE TABLE AS SELECT (CTAS) 查詢會根據另一個查詢的 SELECT 陳述式結果，在 Athena 中建立新的資料表。Athena 會將 CTAS 陳述式建立的資料檔案存放在 Simple Storage Service (Amazon S3) 中的指定位置。如需語法，請參閱[CREATE TABLE AS](#)。

CREATE TABLE AS 結合使用 CREATE TABLE DDL 陳述式與 SELECT DML 陳述式，因此技術上同時包含 DDL 和 DML。不過請注意，針對 Service Quotas 而言，Athena 中的 CTAS 查詢會被視為 DML。如需有關 Athena 中的 Service Quotas 的相關資訊，請參閱[Service Quotas](#)。

使用 CTAS 查詢來：

- 以一個步驟從查詢結果建立資料表，而不需重複查詢原始資料集。這可讓您更輕鬆地使用原始資料集。
- 轉換查詢結果並將資料表遷移為其他資料表格式，例如 Apache Iceberg。這可以改善查詢效能並降低 Athena 中的查詢成本。如需相關資訊，請參閱[建立 Iceberg 資料表](#)。
- 將查詢結果轉換為 Parquet 和 ORC 等儲存格式。這可以改善查詢效能並降低 Athena 中的查詢成本。如需相關資訊，請參閱[單欄式儲存格式](#)。
- 建立僅包含您所需資料的現有資料表的副本。

### 主題

- [CTAS 查詢的考量事項與限制](#)

- [在主控台中執行 CTAS 查詢](#)
- [在 Athena 中分割和歸納](#)
- [CTAS 查詢的範例](#)
- [使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析](#)
- [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)

## CTAS 查詢的考量事項與限制

下列各節會描述在 Athena 中使用 CREATE TABLE AS SELECT (CTAS) 查詢時要記住的注意事項和限制。

### CTAS 查詢語法

CTAS 查詢語法與用於建立資料表的 CREATE [EXTERNAL] TABLE 語法不同。請參閱[CREATE TABLE AS](#)。

### CTAS 查詢與檢視

CTAS 查詢會將新資料寫入到 Simple Storage Service (Amazon S3) 中的指定位置，但檢視不會寫入任何資料。

### CTAS 查詢結果的位置

如果您的工作群組為查詢結果位置[覆寫用戶端設定](#)，則 Athena 會在位置 `s3://DOC-EXAMPLE-BUCKET/tables/<query-id>/` 中建立資料表。若要查看為工作群組指定的查詢結果位置，請[檢視工作群組的詳細資訊](#)。

如果您的工作群組未覆寫查詢結果位置，您可以在 CTAS 查詢中使用語法 WITH (`external_location = 's3://DOC-EXAMPLE-BUCKET/'`) 來指定 CTAS 查詢結果的儲存位置。

#### Note

`external_location` 屬性必須指定空白的位置。CTAS 查詢會檢查儲存貯體中的路徑位置 (前綴) 是空的，並且如果該位置中已有資料，則不會加以覆寫。若要再次使用相同的位置，請刪除儲存貯體的金鑰前綴位置中的資料。

如果您省略 `external_location` 語法而且未使用工作群組設定，則 Athena 會使用查詢結果位置的 [用戶端設定](#)，並在位置 `s3://DOC-EXAMPLE-BUCKET/<Unsaved-or-query-name>/<year>/<month>/<date>/tables/<query-id>/` 中建立資料表。

## 尋找孤立檔案

如果 CTAS 或 `INSERT INTO` 陳述式失敗，該資料位置可能遺留孤立的資料。由於 Athena 在某些案例中不會從儲存貯體中刪除資料或部分資料，因此您可以在後續查詢中讀取此部分資料。若要尋找孤立檔案以進行檢測或刪除，您可以使用 Athena 提供的資料資訊清單檔案來追蹤要寫入的檔案清單。如需詳細資訊，請參閱 [識別查詢輸出檔案](#) 及 [DataManifestLocation](#)。

## 忽略 ORDER BY 子句

在 CTAS 查詢中，Athena 會忽略查詢 `SELECT` 部分中的 `ORDER BY` 子句。

根據 SQL 規格 (ISO 9075 第 2 部分)，只有立即包含 `ORDER BY` 子句的查詢表達式，才能保證查詢表達式所指定之資料表資料列的順序。在任何情況下，SQL 中的資料表本身是無順序的，並且在子查詢子句中實作 `ORDER BY` 都會導致查詢執行不彰，並且將會導致無序輸出。因此，在 Athena CTAS 查詢中，無法保證在寫入資料時會保留 `ORDER BY` 子句指定的順序。

## 用於存放查詢結果的格式

在預設情況下，如果您不指定資料儲存體格式，CTAS 查詢的結果會存放在 Parquet 中。您可以將 CTAS 結果存放在 PARQUET、ORC、AVRO、JSON 和 TEXTFILE 中。CTAS TEXTFILE 格式不支援多字元分隔符號。CTA 查詢不需要指定 a SerDe 來解譯格式轉換。請參閱 [Example: Writing query results to a different format](#)。

## 壓縮格式

GZIP 壓縮用於 JSON 和 TEXTFILE 格式的 CTAS 查詢結果。對於 Parquet，您可以使用 GZIP 或 SNAPPY，預設值為 GZIP。對於 ORC，您可以使用 LZ4、SNAPPY、ZLIB 或 ZSTD，預設值為 ZLIB。如需指定壓縮的 CTAS 範例，請參閱 [Example: Specifying data storage and compression formats](#)。如需有關 Athena 中壓縮的資訊，請參閱 [Athena 壓縮支援](#)。

## 分割區和儲存貯體限制

您可以分割和歸納 CTAS 查詢的結果資料。如需詳細資訊，請參閱 [在 Athena 中分割和歸納](#)。使用 CTAS 建立分區資料表時，Athena 的寫入限制為 100 個分區。

在指定目標資料表屬性之 `WITH` 子句的結尾包含分割和歸納述詞。如需詳細資訊，請參閱 [Example: Creating bucketed and partitioned tables](#) 及 [在 Athena 中分割和歸納](#)。

如需解決 100 個分割區限制的相關資訊，請參閱 [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)。

## 加密

您可以在 Simple Storage Service (Amazon S3) 中加密 CTAS 查詢結果，類似於您在 Athena 中加密其他查詢結果的方式。如需詳細資訊，請參閱 [加密 Amazon S3 中存放的 Athena 查詢結果](#)。

## 預期的儲存貯體擁有者

對於 CTAS 陳述式，預期的儲存貯體擁有者設定不適用於 Simple Storage Service (Amazon S3) 中的目的地資料表位置。預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Simple Storage Service (Amazon S3) 輸出位置。如需詳細資訊，請參閱 [使用 Athena 主控台指定查詢結果位置](#)。

## 資料類型

CTAS 查詢的資料欄資料類型與為原始查詢所指定的相同。

## 在主控台中執行 CTAS 查詢

在 Athena 主控台中，您可以從另一個查詢建立一個 CTAS 查詢。

若要從另一個查詢建立 CTAS 查詢

1. 在 Athena 主控台查詢編輯器中執行查詢。
2. 選擇查詢編輯器最下方的 Create (建立) 選項，然後選擇 Table from query (查詢中的資料表)。
3. 在 Create table as select (建立所選資料表) 表單中，按如下方式填寫欄位：
  - a. 在 Table name (資料表名稱) 中，輸入新資料表的名稱。僅使用小寫和底線，例如 `my_select_query_parquet`。
  - b. 在 Database configuration (資料庫組態) 中，使用選項來選擇現有資料庫或建立資料庫。
  - c. (選用) 在 Result configuration (結果組態) 的 Location of CTAS query results (CTAS 查詢結果的位置) 中，如果您的工作群組查詢結果位置設定未覆寫此選項，請執行下列其中一項作業：
    - 在搜尋方塊中輸入現有 S3 位置的路徑，或選擇 Browse S3 (瀏覽 S3) 從清單中選擇位置。
    - 選擇 View (檢視)，以開啟 Amazon S3 主控台的 Buckets (儲存貯體) 頁面，您可以在其中檢視現有儲存貯體的詳細資訊，並選擇儲存貯體，或使用自己的設定建立儲存貯體。

您應該在 Amazon S3 中指定將進行資料輸出的空白位置。如果您指定的位置中已存有資料，則查詢失敗並出現錯誤。

如果您的工作群組查詢結果位置設定覆寫此位置的設定，則 Athena 會在位置 `s3://DOC-EXAMPLE-BUCKET/tables/query_id/` 中建立資料表。

- d. 在 Data format (資料格式) 中，指定資料所使用的格式。
  - Table type (資料表類型) – Athena 中的預設資料表類型為 Apache Hive。
  - File format (檔案格式) – 選擇 CSV、TSV、JSON、Parquet 或 ORC 等選項。如需 Parquet 和 ORC 格式的相關資訊，請參閱 [單欄式儲存格式](#)。
  - Write compression (寫入壓縮) – (選用) 選擇壓縮格式。Athena 支援各種壓縮格式來讀取和寫入資料，包括從使用多種壓縮格式的資料表讀取。例如，當某些 Parquet 檔案使用 Snappy 壓縮而其他 Parquet 檔案使用 GZIP 壓縮時，Athena 可以成功讀取使用 Parquet 檔案格式的資料表中的資料。相同的原則適用於 ORC、文字檔案和 JSON 儲存格式。如需詳細資訊，請參閱 [Athena 壓縮支援](#)。
  - Partitions (分割區) – (選用) 選取要分割的資料欄。分割您的資料會限制每個查詢所掃描的資料量，從而可以提高效能和降低成本。您可透過任何索引鍵來分割您的資料。如需詳細資訊，請參閱 [在 Athena 中分割資料](#)。
  - Buckets (儲存貯體) – (選用) 選取您要歸納的資料欄。歸納是一種依照特定資料欄將資料分組到單個分割區的技術。這些資料欄稱為儲存貯體金鑰。將相關資料分組至單一儲存貯體 (分割區內的檔案)，可大幅減少 Athena 掃描的資料量，進而改善查詢效能並降低成本。如需詳細資訊，請參閱 [在 Athena 中分割和歸納](#)。
- e. 使用 Preview table query (預覽資料表查詢) 可請檢閱您的查詢。如需查詢語法，請參閱 [CREATE TABLE AS](#)。
- f. 選擇 建立資料表。

## 使用 SQL 範本建立 CTAS 查詢

使用 CREATE TABLE AS SELECT 範本在查詢編輯器中建立 CTAS 查詢。

1. 在 Athena 主控台中，選擇位在 Tables and views (資料表和檢視) 旁的 Create table (建立資料表)，然後選擇 CREATE TABLE AS SELECT (建立資料表作為選擇)。這會使用帶有預留位置值的 CTAS 查詢填入查詢編輯器。
2. 在查詢編輯器中，視所需編輯查詢。如需查詢語法，請參閱 [CREATE TABLE AS](#)。

### 3. 選擇執行。

如需範例，請參閱 [CTAS 查詢的範例](#)。

## 在 Athena 中分割和歸納

分割和歸納是減少執行查詢時 Athena 必須掃描的資料量的兩種方法。分割和歸納互為補充，且可搭配使用。減少掃描的資料量可改善查詢效能並降低成本。如需有關 Athena 查詢效能的一般指導方針，請參閱 [Amazon Athena 的十大效能調校秘訣](#)。

### 什麼是分割？

分割表示根據資料的特定屬性，將資料整理到 Amazon S3 上的目錄 (或「字首」)。這類屬性稱為分割區索引鍵。常見的分割區索引鍵是日期或一些其他時間單位，例如年份或月份。不過，資料集可以依多個索引鍵進行分割。例如，有關產品銷售的資料可能依日期、產品類別和市場進行分割。

### 決定如何分割

分割區索引鍵的理想候選者是始終或經常在查詢中使用且具有低基數的屬性。分割區過多和分割區過少之間需要取得權衡。由於分割區過多，檔案數增加會產生負荷。篩選分割區本身也有一些負荷。如果分割區太少，查詢通常需要掃描更多資料。

### 建立分割的資料表

分割資料集後，您可以在 Athena 建立分割的資料表。分割的資料表是具有分割區索引鍵的資料表。使用 CREATE TABLE 時，您可以將分割區新增至資料表中。使用 CREATE TABLE AS 時，在 Amazon S3 上建立的分割區會自動新增至資料表。

在 CREATE TABLE 陳述式中，您可以在 PARTITIONED BY (*column\_name data\_type*) 子句中指定分割區索引鍵。在 CREATE TABLE AS 陳述式中，您可以在 WITH (partitioned\_by = ARRAY['*partition\_key*']) 子句或 Iceberg 資料表的 WITH (partitioning = ARRAY['*partition\_key*']) 中指定分割區索引鍵。出於效能考量，分割區索引鍵應始終是類型 STRING。如需詳細資訊，請參閱 [使用字串做為分割區索引鍵的資料類型](#)。

如需其他 CREATE TABLE 和 CREATE TABLE AS 語法詳細資訊，請參閱 [CREATE TABLE](#) 和 [CTAS 資料表屬性](#)。

## 查詢分割的資料表

當您查詢分割區資料表時，Athena 會使用查詢中的述詞來篩選分割區的清單。然後，其會使用相符分割區的位置來處理找到的檔案。Athena 只要不讀取與查詢述詞不相符的分割區中的資料，就能有效減少掃描的資料量。

### 範例

假設您有一個依 `sales_date` 和 `product_category` 分割的資料表，並且想知道特定類別中一周的總收入。您可以在 `sales_date` 和 `product_category` 資料欄上加入述詞，以確保 Athena 只掃描最少量的資料，如下列範例所示。

```
SELECT SUM(amount) AS total_revenue
FROM sales
WHERE sales_date BETWEEN '2023-02-27' AND '2023-03-05'
AND product_category = 'Toys'
```

假設您有依日期分割的資料集，但也有精細的時間戳記。

使用 Iceberg 資料表，您可以宣告分割區索引鍵與資料欄之間的關係，但是使用 Hive 資料表時，查詢引擎對資料欄和分割區索引鍵之間的關係一無所知。因此，您必須在查詢中包含資料欄和分割區索引鍵的述詞，以確保查詢掃描的資料不會超過必要數目。

例如，假設上一個範例中的 `sales` 資料表也有 `TIMESTAMP` 資料類型的 `sold_at` 資料欄。如果您只想在特定時間範圍內獲得收入，則可以像這樣寫入查詢：

```
SELECT SUM(amount) AS total_revenue
FROM sales
WHERE sales_date = '2023-02-28'
AND sold_at BETWEEN TIMESTAMP '2023-02-28 10:00:00' AND TIMESTAMP '2023-02-28
  12:00:00'
AND product_category = 'Toys'
```

如需有關查詢 Hive 和 Iceberg 資料之間的差異的詳細資訊，請參閱 [如何寫入同樣按時間分割的時間戳記欄位的查詢](#)。

## 什麼是歸納？

歸納是一種將資料集的記錄整理至稱為儲存貯體的類別的方法。

儲存貯體和歸納的含義與 Amazon S3 儲存貯體不同，且不應與 Amazon S3 儲存貯體混淆。在資料歸納中，具有相同屬性值的記錄會進入同一個儲存貯體。記錄會盡可能在儲存貯體之間平均分配，因此每個儲存貯體的資料量大致相同。

實際上，儲存貯體是檔案，而雜湊函數可確定記錄進入的儲存貯體。歸納的資料集在每個分割區的每個儲存貯體中會有一或多個檔案。檔案所屬的儲存貯體會以檔案名稱編碼。

### 歸納益處

當資料集依特定屬性進行歸納，而且您想要擷取該屬性具有特定值的記錄時，歸納功能非常有用。由於資料是歸納性質的，Athena 可以使用此值來判斷要查看的檔案。例如，假設資料集依 `customer_id` 進行歸納，而您想要尋找特定客戶的所有記錄。Athena 會判斷包含這些記錄的儲存貯體，而且只會讀取該儲存貯體中的檔案。

當您的資料欄具有高基數 (也就是，有許多不同的值)、平均分散，以及您經常查詢特定值的資料欄時，就會發生歸納的適當候選項。

#### Note

Athena 不支援使用 `INSERT INTO` 將新記錄新增至歸納的資料表。

### 支援依據已歸納資料欄進行篩選的資料類型

您可以在具有某些資料類型的歸納資料欄上新增篩選條件。Athena 支援對具有下列資料類型的已歸納資料欄進行篩選：

- BOOLEAN
- BYTE
- DATE
- DOUBLE
- FLOAT
- INT
- LONG
- SHORT
- STRING
- VARCHAR

## Hive 和 Spark 支援

Athena 引擎版本 2 支援使用 Hive 儲存貯體演算法歸納的資料集，而 Athena 引擎版本 3 也支援 Apache Spark 歸納演算法。預設為 Hive 歸納。如果您的資料集是使用 Spark 演算法歸納，請使用 TBLPROPERTIES 子句將 bucketing\_format 屬性值設定為 spark。

### Note

Athena 在每個 CREATE TABLE AS SELECT (CTAS) 查詢的分割區限制為 100 個。同樣地，您僅可以使用 INSERT INTO 陳述式將最多 100 個分割區新增至目的地資料表。100 個的限制只有在歸納和分割資料表時才會套用。

如果您超出此限制，您可能會收到錯誤訊息 HIVE\_TOO\_MANY\_OPEN\_PARTITIONS:  
Exceeded limit of 100 open writers for partitions/buckets

(HIVE\_TOO\_MANY\_OPEN\_PARTITIONS：超過分割區 / 儲存貯體 100 個開啟寫入器的限制)。若要避開此限制，您可以使用 CTAS 陳述式和一系列的 INSERT INTO 陳述式，每個陳述式可建立或插入最多 100 個分割區。如需詳細資訊，請參閱 [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)。

### 歸納 CREATE TABLE 範例

若要為現有歸納的資料集建立資料表，請使用 CLUSTERED BY (*column*) 子句，後面接著 INTO *N* BUCKETS 子句。INTO *N* BUCKETS 子句指定了資料要歸納到的儲存貯體數量。

在下列 CREATE TABLE 範例中，使用 Spark 演算法，依 customer\_id 將 sales 資料集歸納成 8 個儲存貯體。CREATE TABLE 陳述式會使用 CLUSTERED BY 和 TBLPROPERTIES 子句來設定相應的屬性。

```
CREATE EXTERNAL TABLE sales (...)  
...  
CLUSTERED BY (`customer_id`) INTO 8 BUCKETS  
...  
TBLPROPERTIES (  
  'bucketing_format' = 'spark'  
)
```

### 歸納 CREATE TABLE AS (CTAS) 範例

若要使用 CREATE TABLE AS 指定歸納，請使用 bucketed\_by 和 bucket\_count 參數，如下列範例所示。

```
CREATE TABLE sales
WITH (
  ...
  bucketed_by = ARRAY['customer_id'],
  bucket_count = 8
)
AS SELECT ...
```

## 歸納查詢範例

下列範例查詢會尋找特定客戶在一週內購買的產品名稱。

```
SELECT DISTINCT product_name
FROM sales
WHERE sales_date BETWEEN '2023-02-27' AND '2023-03-05'
AND customer_id = 'c123'
```

如果此資料表依 `sales_date` 進行分割並依 `customer_id` 歸納，則 Athena 可以計算客戶記錄所在的儲存貯體。Athena 每個分割區最多只能讀取一個檔案。

## 其他資源

- 有關建立歸納和分割資料表的 CREATE TABLE AS 範例，請參閱[範例：建立歸納和分割的資料表](#)。
- 如需在資 AWS 料湖上實作儲存貯體的相關資訊，包括使用 Athena CTAS 陳述式 (適用 AWS Glue 於 Apache Spark)，以及 Apache Iceberg 資料表分段的相關資訊，請參閱 AWS 大數據部落格文章[透過分組與 Amazon Athena 進行分組來優化資料配置，並加速下游查詢](#)。AWS Glue

## CTAS 查詢的範例

使用下列範例來建立 CTAS 查詢。如需 CTAS 語法的相關資訊，請參閱[CREATE TABLE AS](#)。

在本節中：

- [Example: Duplicating a table by selecting all columns](#)
- [Example: Selecting specific columns from one or more tables](#)
- [Example: Creating an empty copy of an existing table](#)
- [Example: Specifying data storage and compression formats](#)
- [Example: Writing query results to a different format](#)

- [Example: Creating unpartitioned tables](#)
- [Example: Creating partitioned tables](#)
- [Example: Creating bucketed and partitioned tables](#)
- [Example: Creating an Iceberg table with Parquet data](#)
- [Example: Creating an Iceberg table with Avro data](#)

Example 範例：選取所有資料欄以複製資料表

以下範例會透過複製資料表的所有資料欄來建立資料表：

```
CREATE TABLE new_table AS
SELECT *
FROM old_table;
```

在相同範例的以下變化中，您的 SELECT 陳述式也包含 WHERE 子句。在這種情況下，查詢只會從資料表中選取滿足 WHERE 子句的那些資料列：

```
CREATE TABLE new_table AS
SELECT *
FROM old_table
WHERE condition;
```

Example 範例：從一或多個資料表選取特定資料欄

以下範例會建立對來自另一個資料表的一組資料欄執行的新查詢：

```
CREATE TABLE new_table AS
SELECT column_1, column_2, ... column_n
FROM old_table;
```

相同範例的這個變化會來自多個資料表的特定資料欄建立新的資料表：

```
CREATE TABLE new_table AS
SELECT column_1, column_2, ... column_n
FROM old_table_1, old_table_2, ... old_table_n;
```

Example 範例：建立現有資料表的空白副本

以下範例使用 WITH NO DATA 來建立空白且與原始資料表具有相同結構描述的新資料表：

```
CREATE TABLE new_table
AS SELECT *
FROM old_table
WITH NO DATA;
```

### Example 範例：指定資料儲存和壓縮格式

利用 CTAS，您可以使用一種儲存格式的來源資料表，建立不同儲存格式的另一個資料表。

使用 `format` 屬性來指定 ORC、PARQUET、AVRO、JSON 或 TEXTFILE 作為新資料表的儲存格式。

對於 PARQUET、ORC、TEXTFILE，和 JSON 儲存格式，請使用 `write_compression` 屬性指定新資料表資料的壓縮格式。如需每個檔案格式支援之壓縮格式的詳細資訊，請參閱 [Athena 壓縮支援](#)。

下列範例指定資料表 `new_table` 中的資料以 Parquet 格式儲存並使用 Snappy 壓縮。Parquet 的預設壓縮為 GZIP。

```
CREATE TABLE new_table
WITH (
    format = 'Parquet',
    write_compression = 'SNAPPY')
AS SELECT *
FROM old_table;
```

下列範例指定使用 Snappy 壓縮以 ORC 格式儲存資料表 `new_table` 中的資料。ORC 的預設壓縮為 ZLIB。

```
CREATE TABLE new_table
WITH (format = 'ORC',
    write_compression = 'SNAPPY')
AS SELECT *
FROM old_table ;
```

下列範例指定使用 Snappy 壓縮以文字檔案格式儲存資料表 `new_table` 中的資料。Textfile 和 JSON 格式的預設壓縮皆為 GZIP。

```
CREATE TABLE new_table
WITH (format = 'TEXTFILE',
    write_compression = 'SNAPPY')
AS SELECT *
```

```
FROM old_table ;
```

Example 範例：將查詢結果寫入不同的格式

以下 CTAS 查詢會從 `old_table` 選取所有記錄 (可用 CSV 或其他格式儲存)，並使用以 ORC 格式儲存至 Simple Storage Service (Amazon S3) 的基礎資料建立新的資料表：

```
CREATE TABLE my_orc_ctas_table
WITH (
    external_location = 's3://DOC-EXAMPLE-BUCKET/my_orc_stas_table/',
    format = 'ORC')
AS SELECT *
FROM old_table;
```

Example 範例：建立未分割的資料表

以下範例會建立未分割的資料表。資料表資料會以不同格式存放。其中的一些範例會指定外部位置。

以下範例會建立可將結果存放為文字檔案的 CTAS 查詢：

```
CREATE TABLE ctas_csv_unpartitioned
WITH (
    format = 'TEXTFILE',
    external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_csv_unpartitioned/')
AS SELECT key1, name1, address1, comment1
FROM table1;
```

在下列範例中，結果會存放在 Parquet 中，並且會使用預設結果位置：

```
CREATE TABLE ctas_parquet_unpartitioned
WITH (format = 'PARQUET')
AS SELECT key1, name1, comment1
FROM table1;
```

在以下查詢中，資料表會存放在 JSON 中，並且從原始資料表的結果選取特定資料欄：

```
CREATE TABLE ctas_json_unpartitioned
WITH (
    format = 'JSON',
    external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_json_unpartitioned/')
AS SELECT key1, name1, address1, comment1
```

```
FROM table1;
```

在下列範例中，格式為 ORC：

```
CREATE TABLE ctas_orc_unpartitioned
WITH (
    format = 'ORC')
AS SELECT key1, name1, comment1
FROM table1;
```

在下列範例中，格式為 Avro：

```
CREATE TABLE ctas_avro_unpartitioned
WITH (
    format = 'AVRO',
    external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_avro_unpartitioned/')
AS SELECT key1, name1, comment1
FROM table1;
```

Example 範例：建立分割的資料表

以下範例顯示採用不同儲存格式的分割資料表的 CREATE TABLE AS SELECT 查詢，使用 `partitioned_by`，以及 WITH 子句中的其他屬性。如需語法，請參閱 [CTAS 資料表屬性](#)。如需選擇用於分割的資料欄的詳細資訊，請參閱 [在 Athena 中分割和歸納](#)。

#### Note

在 SELECT 陳述式中資料欄清單結尾列出分割區資料欄。您可以依多於一個資料欄分割，且最多可以有 100 個唯一分割區和儲存貯體組合。例如，如果未指定儲存貯體，則可有 100 個分割區。

```
CREATE TABLE ctas_csv_partitioned
WITH (
    format = 'TEXTFILE',
    external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_csv_partitioned/',
    partitioned_by = ARRAY['key1'])
AS SELECT name1, address1, comment1, key1
FROM tables1;
```

```
CREATE TABLE ctas_json_partitioned
WITH (
  format = 'JSON',
  external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_json_partitioned/',
  partitioned_by = ARRAY['key1'])
AS select name1, address1, comment1, key1
FROM table1;
```

### Example 範例：建立歸納和分割的資料表

以下範例顯示一個 CREATE TABLE AS SELECT 查詢，同時使用歸納和分割來在 Simple Storage Service (Amazon S3) 中存放查詢結果。資料表結果會依不同的資料欄分割和歸納。Athena 支援最多 100 個唯一儲存貯體和分割區組合。例如，如果您建立包含 5 個儲存貯體的資料表，則支援 20 個分割區 (每個都有 5 個儲存貯體)。如需語法，請參閱 [CTAS 資料表屬性](#)。

如需選擇用於歸納的資料欄的詳細資訊，請參閱 [在 Athena 中分割和歸納](#)。

```
CREATE TABLE ctas_avro_bucketed
WITH (
  format = 'AVRO',
  external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_avro_bucketed/',
  partitioned_by = ARRAY['nationkey'],
  bucketed_by = ARRAY['mktsegment'],
  bucket_count = 3)
AS SELECT key1, name1, address1, phone1, acctbal, mktsegment, comment1, nationkey
FROM table1;
```

### Example 範例：建立一個包含 Parquet 資料的 Iceberg 資料表。

下列範例會建立包含 Parquet 資料檔案的 Iceberg 資料表。使用 table1 中的 dt 資料欄按月份分割檔案。此範例會更新資料表上的保留屬性，以便在資料表中的每個分支上保留 10 個快照。另外，還會保留過去 7 天內的快照。如需有關 Athena 中的 Iceberg 資料表屬性的詳細資訊，請參閱 [資料表屬性](#)。

```
CREATE TABLE ctas_iceberg_parquet
WITH (table_type = 'ICEBERG',
  format = 'PARQUET',
  location = 's3://DOC-EXAMPLE-BUCKET/ctas_iceberg_parquet/',
  is_external = false,
  partitioning = ARRAY['month(dt)'],
  vacuum_min_snapshots_to_keep = 10,
  vacuum_max_snapshot_age_seconds = 604800)
```

```
)  
AS SELECT key1, name1, dt FROM table1;
```

Example 範例：建立一個包含 Avro 資料的 Iceberg 資料表。

下列範例會建立包含 Avro 資料檔案的 Iceberg 資料表，其中資料表會依 key1 進行分割。

```
CREATE TABLE ctas_iceberg_avro  
WITH ( format = 'AVRO',  
      location = 's3://DOC-EXAMPLE-BUCKET/ctas_iceberg_avro/',  
      is_external = false,  
      table_type = 'ICEBERG',  
      partitioning = ARRAY['key1'])  
AS SELECT key1, name1, date FROM table1;
```

## 使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析

您可以在 Athena 中使用 Create Table as Select ([CTAS](#)) 與 [INSERT INTO](#) 陳述式，以擷取、轉換和載入 (ETL) 資料至 Amazon S3 中進行資料處理。本主題說明如何使用這些陳述式以將資料集分割及轉換成單欄式資料格式，以最佳化資料分析。

CTAS 陳述式使用標準 [SELECT](#) 查詢來建立新的資料表。您可以使用 CTAS 陳述式來建立資料的子集以供分析。在一個 CTAS 陳述式中，您可以分割資料、指定壓縮，並將資料轉換成一個單欄格式，如 Apache Parquet 或 Apache ORC。當您執行 CTAS 查詢時，它所建立的資料表和分割會自動加入到 [AWS Glue Data Catalog](#)。這會讓它建立的新資料表和分割區立即可供後續查詢使用。

INSERT INTO 陳述式會根據在來源資料表上執行的 SELECT 查詢陳述式，將新的資料列插入目的地資料表。您可以使用 INSERT INTO 陳述式將 CSV 格式的來源資料表資料轉換並載入目的地資料表資料 (此資料使用 CTAS 支援的所有轉換)。

### 概觀

在 Athena 中，使用 CTAS 陳述式執行資料的初始批次轉換。然後使用多個 INSERT INTO 陳述式，對 CTAS 陳述式所建立的資料表進行累加式更新。

### 步驟

- [步驟 1：建立以原始資料集為基礎的資料表](#)
- [步驟 2：使用 CTAS 來分割、轉換和壓縮資料](#)

- [步驟 3：使用 INSERT INTO 以新增資料](#)
- [步驟 4：測量效能與成本差異](#)

## 步驟 1：建立以原始資料集為基礎的資料表

本主題中的範例使用公開提供之 [NOAA Global Historical Climatology Network Daily \(GHCN-D\)](#) 資料集的 Amazon S3 可讀取子集。Amazon S3 上的資料具有下列特性。

```
Location: s3://aws-bigdata-blog/artifacts/athena-ctas-insert-into-blog/  
Total objects: 41727  
Size of CSV dataset: 11.3 GB  
Region: us-east-1
```

原始資料存放在 Amazon S3 中，無任何分割區。這些資料是 CSV 格式的文件，如下所示。

```
2019-10-31 13:06:57 413.1 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0000  
2019-10-31 13:06:57 412.0 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0001  
2019-10-31 13:06:57 34.4 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0002  
2019-10-31 13:06:57 412.2 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0100  
2019-10-31 13:06:57 412.7 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0101
```

此範例中的檔案大小相對較小。藉由將檔案合併至較大的檔案中，您可以減少檔案總數，以提高查詢效能。您可以使用 CTAS 和 INSERT INTO 陳述式來增強查詢效能。

根據範例資料集建立資料庫和資料表

1. 在 Athena 主控台中，選擇美國東部 (維吉尼亞北部) AWS 區域。請務必在 us-east-1 的本教學課程中執行所有查詢。
2. 在 Athena 查詢編輯器中，執行 [CREATE DATABASE](#) 命令以建立資料庫。

```
CREATE DATABASE blogdb
```

3. 執行下列陳述式來 [建立資料表](#)。

```
CREATE EXTERNAL TABLE `blogdb`.`original_csv` (  
  `id` string,  
  `date` string,  
  `element` string,  
  `datavalue` bigint,  
  `mflag` string,
```

```

`qflag` string,
`sflag` string,
`obstime` bigint)
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://aws-bigdata-blog/artifacts/athena-ctas-insert-into-blog/'

```

## 步驟 2：使用 CTAS 來分割、轉換和壓縮資料

建立資料表之後，您可以使用單一 [CTAS](#) 陳述式將資料轉換為具有 Snappy 壓縮的 Parquet 格式與，並按年份分割資料。

您在步驟 1 中建立的表格有一個 date 欄位，其日期格式為 YYYYMMDD (例如，20100104)。因為新的資料表會在 year 被分割，下列程序中的範例陳述式會使用 Presto 函式 substr("date",1,4) 以從 date 欄位擷取 year 值。

將資料轉換為 Snappy 壓縮的 Parquet 格式，按年份分割

- 執行以下 CTAS 陳述式，將 *your-bucket* 替換為您的 Amazon S3 儲存貯體位置。

```

CREATE table new_parquet
WITH (format='PARQUET',
      parquet_compression='SNAPPY',
      partitioned_by=array['year'],
      external_location = 's3://DOC-EXAMPLE-BUCKET/optimized-data/')
AS
SELECT id,
       date,
       element,
       datavalue,
       mflag,
       qflag,
       sflag,
       obstime,
       substr("date",1,4) AS year
FROM original_csv
WHERE cast(substr("date",1,4) AS bigint) >= 2015

```

```
AND cast(substr("date",1,4) AS bigint) <= 2019
```

### Note

在本範例中，您建立的資料表會包含從 2015 年到 2019 年的資料。在步驟 3 中，您可以使用 INSERT INTO 命令將新的資料加入到此資料表。

查詢完成時，請使用下列程序來確認您在 CTAS 陳述式中指定的 Amazon S3 位置中的輸出。

查看 CTAS 陳述式所建立的分割區和 parquet 檔案

1. 若要顯示建立的分割區，請執行下列 AWS CLI 命令。請務必包含最後的正斜線 (/)。

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/
```

輸出會顯示分割區。

```
PRE year=2015/  
PRE year=2016/  
PRE year=2017/  
PRE year=2018/  
PRE year=2019/
```

2. 若要查看 Parquet 檔案，請執行以下命令。請注意，在 Windows 上不可使用將輸出限制為前五個結果的 | head -5 選項。

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/ --recursive --human-readable |  
head -5
```

輸出結果與以下內容相似。

```
2019-10-31 14:51:05    7.3 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_1be48df2-3154-438b-b61d-8fb23809679d  
2019-10-31 14:51:05    7.0 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_2a57f4e2-ffa0-4be3-9c3f-28b16d86ed5a  
2019-10-31 14:51:05    9.9 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_34381db1-00ca-4092-bd65-ab04e06dc799  
2019-10-31 14:51:05    7.5 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_354a2bc1-345f-4996-9073-096cb863308d
```

```
2019-10-31 14:51:05    6.9 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_42da4cfd-6e21-40a1-8152-0b902da385a1
```

### 步驟 3：使用 INSERT INTO 以新增資料

在步驟 2 中，您已使用 CTAS 來建立 2015 年到 2019 年的分割區資料表。不過，原始資料集也包含 2010 年至 2014 年的資料。現在您使用 [INSERT INTO](#) 陳述式新增該資料。

使用一或多個 INSERT INTO 陳述式將資料新增至資料表

1. 執行下列 INSERT INTO 指令，在 WHERE 子句中指定 2015 年之前的年份。

```
INSERT INTO new_parquet  
SELECT id,  
       date,  
       element,  
       datavalue,  
       mflag,  
       qflag,  
       sflag,  
       obstime,  
       substr("date",1,4) AS year  
FROM original_csv  
WHERE cast(substr("date",1,4) AS bigint) < 2015
```

2. 使用下列語法再次執行 `aws s3 ls` 命令。

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/
```

該輸出會顯示新的分割區。

```
PRE year=2010/  
PRE year=2011/  
PRE year=2012/  
PRE year=2013/  
PRE year=2014/  
PRE year=2015/  
PRE year=2016/  
PRE year=2017/  
PRE year=2018/  
PRE year=2019/
```

- 若要查看在 Parquet 格式中使用壓縮和單欄儲存所獲得的資料集大小縮減，請執行以下命令。

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/ --recursive --human-readable --summarize
```

以下結果顯示，用 Snappy 壓縮 parquet 後的資料集大小為 1.2 GB。

```
...
2020-01-22 18:12:02 2.8 MiB optimized-data/
year=2019/20200122_181132_00003_nja5r_f0182e6c-38f4-4245-afa2-9f5bfa8d6d8f
2020-01-22 18:11:59 3.7 MiB optimized-data/
year=2019/20200122_181132_00003_nja5r_fd9906b7-06cf-4055-a05b-f050e139946e
Total Objects: 300
Total Size: 1.2 GiB
```

- 如果有更多 CSV 資料新增至原始資料表，則您可以使用 INSERT INTO 陳述式將該資料加入至 parquet 資料表。例如，如果您有 2020 年的新資料，則您可以執行下列 INSERT INTO 陳述式。該陳述式會將資料和相關的磁碟分割新增至 new\_parquet 資料表。

```
INSERT INTO new_parquet
SELECT id,
       date,
       element,
       datavalue,
       mflag,
       qflag,
       sflag,
       obstime,
       substr("date",1,4) AS year
FROM original_csv
WHERE cast(substr("date",1,4) AS bigint) = 2020
```

#### Note

INSERT INTO 陳述式支援最多將 100 個分割區寫入目標資料表。不過，若要新增超過 100 個分割區，您可以執行多個 INSERT INTO 陳述式。如需詳細資訊，請參閱[使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)。

## 步驟 4：測量效能與成本差異

在您轉換資料之後，您可以在新舊資料表上執行相同的查詢，並比較結果來衡量效能提升和成本節省。

### Note

如需 Athena 每個查詢成本的資訊，請參閱 [Amazon Athena 定價](#)。

### 衡量效能提升與成本差異

1. 在原始資料表上執行下列查詢。該查詢會尋找一年中每個值的不同 ID 數目。

```
SELECT substr("date",1,4) as year,
       COUNT(DISTINCT id)
FROM original_csv
GROUP BY 1 ORDER BY 1 DESC
```

2. 請注意查詢執行的時間，以及掃描的資料量。
3. 在新資料表上執行相同的查詢，並記下查詢執行時間和掃描的資料量。

```
SELECT year,
       COUNT(DISTINCT id)
FROM new_parquet
GROUP BY 1 ORDER BY 1 DESC
```

4. 比較結果並計算效能和成本差異。下列範例結果顯示，在新資料表上的測試查詢比舊資料表上的查詢更快且更便宜。

資料表	執行期	已掃描的資料
原始的	16.88 秒	11.35 GB
新增	3.79 秒	428.05 MB

5. 在原始資料表上執行下列範例查詢。該查詢會計算 2018 年地球的平均最高溫度 (攝氏)、平均最低溫度 (攝氏) 和平均降雨量 (mm)。

```
SELECT element, round(avg(CAST(datavalue AS real)/10),2) AS value
FROM original_csv
```

```
WHERE element IN ('TMIN', 'TMAX', 'PRCP') AND substr("date",1,4) = '2018'
GROUP BY 1
```

- 請注意查詢執行的時間，以及掃描的資料量。
- 在新資料表上執行相同的查詢，並記下查詢執行時間和掃描的資料量。

```
SELECT element, round(avg(CAST(datavalue AS real)/10),2) AS value
FROM new_parquet
WHERE element IN ('TMIN', 'TMAX', 'PRCP') and year = '2018'
GROUP BY 1
```

- 比較結果並計算效能和成本差異。下列範例結果顯示，在新資料表上的測試查詢比舊資料表上的查詢更快且更便宜。

資料表	執行期	已掃描的資料
原始的	18.65 秒	11.35 GB
新增	1.92 秒	68 MB

## Summary

本主題說明如何使用 Athena 中的 CTAS 和 INSERT INTO 陳述式來執行 ETL 操作。您會使用 CTAS 陳述式將資料轉換為具有 Snappy 壓縮的 Parquet 格式，以執行第一組轉換。CTAS 陳述式也會將資料集從非分割轉換為已分割。這會減少其大小並降低執行查詢的成本。當新資料可用時，您可以使用 INSERT INTO 陳述式來轉換並將資料載入您使用 CTAS 陳述式建立的資料表。

## 使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制

Athena 每個 CREATE TABLE AS SELECT ([CTAS](#)) 查詢的分割區限制為 100 個。同樣地，您可以使用 [INSERT INTO](#) 陳述式將最多 100 個分割區新增至目的地資料表。

如果您超出此限制，您可能會收到錯誤訊息 HIVE\_TOO\_MANY\_OPEN\_PARTITIONS: Exceeded limit of 100 open writers for partitions/buckets (HIVE\_TOO\_MANY\_OPEN\_PARTITIONS：超過分割區 / 儲存貯體 100 個開啟寫入器的限制)。若要避開此限制，您可以使用 CTAS 陳述式和一系列的 INSERT INTO 陳述式，每個陳述式可建立或插入最多 100 個分割區。

本主題中的範例使用名為 tpch100 其資料位於 Amazon S3 儲存貯體位置的資料庫 s3://DOC-EXAMPLE-BUCKET/。

## 使用 CTAS 和 INSERT INTO 來建立超過 100 個分割區的資料表

1. 使用 CREATE EXTERNAL TABLE 陳述式在您要的欄位上建立分割的資料表。

下列範例陳述式會依資料欄 l\_shipdate 來分割資料。該資料表有 2525 個分割區。

```
CREATE EXTERNAL TABLE `tpch100.lineitem_parq_partitioned`(  
  `l_orderkey` int,  
  `l_partkey` int,  
  `l_suppkey` int,  
  `l_linenum` int,  
  `l_quantity` double,  
  `l_extendedprice` double,  
  `l_discount` double,  
  `l_tax` double,  
  `l_returnflag` string,  
  `l_linestatus` string,  
  `l_commitdate` string,  
  `l_receiptdate` string,  
  `l_shipinstruct` string,  
  `l_comment` string)  
PARTITIONED BY (  
  `l_shipdate` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe' STORED AS  
INPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat' LOCATION  
  's3://DOC-EXAMPLE-BUCKET/lineitem/'
```

2. 執行如下所示的 SHOW PARTITIONS *<table\_name>* 命令以列出分割區。

```
SHOW PARTITIONS lineitem_parq_partitioned
```

以下是部分樣本結果。

```
/*  
l_shipdate=1992-01-02  
l_shipdate=1992-01-03  
l_shipdate=1992-01-04  
l_shipdate=1992-01-05  
l_shipdate=1992-01-06
```

```
...  
  
l_shipdate=1998-11-24  
l_shipdate=1998-11-25  
l_shipdate=1998-11-26  
l_shipdate=1998-11-27  
l_shipdate=1998-11-28  
l_shipdate=1998-11-29  
l_shipdate=1998-11-30  
l_shipdate=1998-12-01  
*/
```

### 3. 執行 CTAS 查詢以建立分割的資料表。

下列範例會建立名為 `my_lineitem_parq_partitioned` 的資料表，並使用 `WHERE` 子句來限制 `DATE` 早於 `1992-02-01`。因為範例資料集開始於 1992 年 1 月，所以只會建立 1992 年 1 月的分割區。

```
CREATE table my_lineitem_parq_partitioned  
WITH (partitioned_by = ARRAY['l_shipdate']) AS  
SELECT l_orderkey,  
       l_partkey,  
       l_suppkey,  
       l_linenumber,  
       l_quantity,  
       l_extendedprice,  
       l_discount,  
       l_tax,  
       l_returnflag,  
       l_linestatus,  
       l_commitdate,  
       l_receiptdate,  
       l_shipinstruct,  
       l_comment,  
       l_shipdate  
FROM tpch100.lineitem_parq_partitioned  
WHERE cast(l_shipdate as timestamp) < DATE ('1992-02-01');
```

### 4. 執行 `SHOW PARTITIONS` 命令，以確認資料表包含您想要的分割區。

```
SHOW PARTITIONS my_lineitem_parq_partitioned;
```

範例中的分割區是從 1992 年 1 月開始。

```
/*  
l_shipdate=1992-01-02  
l_shipdate=1992-01-03  
l_shipdate=1992-01-04  
l_shipdate=1992-01-05  
l_shipdate=1992-01-06  
l_shipdate=1992-01-07  
l_shipdate=1992-01-08  
l_shipdate=1992-01-09  
l_shipdate=1992-01-10  
l_shipdate=1992-01-11  
l_shipdate=1992-01-12  
l_shipdate=1992-01-13  
l_shipdate=1992-01-14  
l_shipdate=1992-01-15  
l_shipdate=1992-01-16  
l_shipdate=1992-01-17  
l_shipdate=1992-01-18  
l_shipdate=1992-01-19  
l_shipdate=1992-01-20  
l_shipdate=1992-01-21  
l_shipdate=1992-01-22  
l_shipdate=1992-01-23  
l_shipdate=1992-01-24  
l_shipdate=1992-01-25  
l_shipdate=1992-01-26  
l_shipdate=1992-01-27  
l_shipdate=1992-01-28  
l_shipdate=1992-01-29  
l_shipdate=1992-01-30  
l_shipdate=1992-01-31  
*/
```

##### 5. 使用 INSERT INTO 陳述式將分割區新增至資料表。

下列範例會針對 1992 年 2 月份的日期新增分割區。

```
INSERT INTO my_lineitem_parq_partitioned  
SELECT l_orderkey,  
       l_partkey,
```

```
    l_suppkey,  
    l_linenumber,  
    l_quantity,  
    l_extendedprice,  
    l_discount,  
    l_tax,  
    l_returnflag,  
    l_linestatus,  
    l_commitdate,  
    l_receiptdate,  
    l_shipinstruct,  
    l_comment,  
    l_shipdate  
FROM tpch100.lineitem_parq_partitioned  
WHERE cast(l_shipdate as timestamp) >= DATE ('1992-02-01')  
AND cast(l_shipdate as timestamp) < DATE ('1992-03-01');
```

## 6. 再次執行 SHOW PARTITIONS。

```
SHOW PARTITIONS my_lineitem_parq_partitioned;
```

範例資料表現在有來自 1992 年 1 月和 2 月的分割區。

```
/*  
l_shipdate=1992-01-02  
l_shipdate=1992-01-03  
l_shipdate=1992-01-04  
l_shipdate=1992-01-05  
l_shipdate=1992-01-06  
  
...  
  
l_shipdate=1992-02-20  
l_shipdate=1992-02-21  
l_shipdate=1992-02-22  
l_shipdate=1992-02-23  
l_shipdate=1992-02-24  
l_shipdate=1992-02-25  
l_shipdate=1992-02-26  
l_shipdate=1992-02-27  
l_shipdate=1992-02-28  
l_shipdate=1992-02-29
```

```
*/
```

- 繼續使用每個讀取和新增不超過 100 個分割區的 INSERT INTO 陳述式。繼續進行，直到達到您需要的分割區數目為止。

### Important

設定 WHERE 條件時，請確定查詢不會重疊。否則，某些分割區可能會有重複的資料。

## SerDe 參考

Athena 支援多個 SerDe 程式庫，以剖析不同資料格式的資料，例如 CSV、JSON、Parquet 和 ORC。Athena 不支援自訂 SerDes。

### 主題

- [使用 SerDe](#)
- [支援的 SerDes 和資料格式](#)

## 使用 SerDe

A SerDe (序列化程式/還原序列化程式) 是 Athena 與各種格式的資料互動的一種方式。

定義資料表結構定義的是 SerDe 您指定的，而不是 DDL。換句話說，當您建立資料表時，SerDe 可以覆寫您在 Athena 中指定的 DDL 組態。

## SerDe 在查詢中使用

若要在 Athena 中建立資料表 SerDe 時使用，請使用下列其中一種方法：

- 指定 ROW FORMAT DELIMITED，然後使用 DDL 陳述式來指定欄位分隔符號，如以下範例所示。當您指定時 ROW FORMAT DELIMITED，LazySimpleSerDe 依預設，Athena 會使用。

```
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
ESCAPED BY '\\\'
COLLECTION ITEMS TERMINATED BY '|'
MAP KEYS TERMINATED BY ':'
```

如需 ROW FORMAT DELIMITED 的範例，請參閱以下主題：

[LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案](#)

[查詢 Amazon CloudFront 日誌](#)

[查詢 Amazon EMR 日誌](#)

[查詢 Amazon VPC 流程日誌](#)

[使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析](#)

- 用 ROW FORMAT SERDE 於明確指定 Athena 在 SerDe 將資料讀取和寫入資料表時應使用的類型。下列範例會指定 LazySimpleSerDe。若要指定分隔符號，請使用 WITH SERDEPROPERTIES。由 WITH SERDEPROPERTIES 指定的屬性會對應至 ROW FORMAT DELIMITED 範例中的單獨陳述式 (例如 FIELDS TERMINATED BY)。

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'  
WITH SERDEPROPERTIES (  
  'serialization.format' = ',',  
  'field.delim' = ',',  
  'collection.delim' = '|',  
  'mapkey.delim' = ':',  
  'escape.delim' = '\\'  
)
```

如需 ROW FORMAT SERDE 的範例，請參閱以下主題：

[Avro SerDe](#)

[GROK SerDe](#)

[JSON SerDe 程式庫](#)

[SerDe 用於處理 CSV 的開啟 CSV](#)

[正則表 SerDe](#)

## 支援的 SerDes 和資料格式

Athena 支援建立資料表和查詢 CSV、TSV、自訂分隔和 JSON 格式的資料；Hadoop 相關格式的資料：ORC、Apache Avro 和 Parquet；來自 Logstash 的日誌、AWS CloudTrail 日誌和 Apache WebServer 日誌。

### Note

本節所列的格式由 Athena 用於讀取資料。如需有關 Athena 執行 CTAS 查詢時用於寫入資料的格式的資訊，請參閱 [從查詢結果建立資料表 \(CTAS\)](#)。

若要在 Athena 中建立資料表和查詢這些格式的資料，請指定序列化程式-還原序列化程式類別 (SerDe)，讓 Athena 知道使用何種格式及如何剖析資料。

下表列出 Athena 中支援的資料格式及其對應的 SerDe 程式庫。

SerDe 是自訂程式庫，可指示 Athena 所用的資料目錄如何處理資料。若要指定 SerDe 類型，請在 Athena 中，在 CREATE TABLE 陳述式的 ROW FORMAT 部分中明確列出 SerDe 類型。在某些情況下，您可以省略 SerDe 名稱，因為對於某些類型的資料格式，Athena 預設會使用一些 SerDe 類型。

### 支援的資料格式和 SerDes

資料格式	說明	Athena 中支援的 SerDe 類型
Amazon Ion	Amazon Ion 是由 Amazon 開發及開放原始碼的豐富類型、自行描述的資料格式，是 JSON 的超集。	使用 <a href="#">Amazon Ion Hive SerDe</a> 。
Apache Avro	Hadoop 中用於存放資料的格式，使用以 JSON 為基礎的結構描述來存放記錄值。	使用 <a href="#">Avro SerDe</a> 。
Apache Parquet	Hadoop 中的資料的單欄式儲存格式。	使用 <a href="#">木地板 SerDe</a> 和 SNAPPY 壓縮。
Apache WebServer 日誌	Apache WebServer 中存放日誌的格式。	使用 <a href="#">GROK SerDe</a> 或 <a href="#">正則表 SerDe</a> 。

資料格式	說明	Athena 中支援的 SerDe 類型
CloudTrail 日誌	CloudTrail 中存放日誌的格式。	<ul style="list-style-type: none"> <li>使用 <a href="#">蜂巢 SerDe</a>。如需更多詳細資訊，請參閱 <a href="#">查詢 AWS CloudTrail 記錄檔</a>。</li> </ul>
CSV (逗號分隔值)	對於 CSV 的資料，每一行代表一筆資料記錄，而每一筆記錄由一或多個欄位組成 (以逗號分隔)。	<ul style="list-style-type: none"> <li>如果您的資料不包含以引號括住的值或其使用 <code>java.sql.Timestamp</code> 格式，則請使用 <a href="#">LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案</a>。</li> <li>在您的資料值中包含引號或為 <code>TIMESTAMP</code> 使用 UNIX 數字格式時 (例如 <code>1564610311</code>)，請使用 <a href="#">SerDe 用於處理 CSV 的開啟 CSV</a>。</li> </ul>
自訂分隔	對於此格式的資料，每一行代表一筆資料記錄，而記錄以自訂單一字元分隔符號隔開。	使用 <a href="#">LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案</a> ，並指定自訂單一字元分隔符號。
JSON (JavaScript 物件標記法)	對於 JSON 資料，每一行代表一筆資料記錄，而每一筆記錄由屬性/值對組成 (以逗號分隔)。	<ul style="list-style-type: none"> <li>使用 <a href="#">蜂巢 SerDe</a>。</li> <li>使用 <a href="#">OpenX SerDe</a>。</li> </ul>
Logstash 日誌	Logstash 中存放日誌的格式。	使用 <a href="#">GROK SerDe</a> 。
ORC (Optimized Row Columnar, 最佳化列單欄式)	Hive 資料的最佳化單欄式儲存格式。	使用 <a href="#">獸人 SerDe</a> 和 ZLIB 壓縮。

資料格式	說明	Athena 中支援的 SerDe 類型
TSV (定位字元分隔值)	對於 TSV 的資料，每一行代表一筆資料記錄，而每一筆記錄由一或多個欄位組成 (以定位字元分隔)。	使用 <a href="#">LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案</a> ，並將分隔符號字元指定為 <code>FIELDS TERMINATED BY '\t'</code> 。

## 主題

- [Amazon Ion Hive SerDe](#)
- [Avro SerDe](#)
- [GROK SerDe](#)
- [JSON SerDe 程式庫](#)
- [LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案](#)
- [SerDe 用於處理 CSV 的開啟 CSV](#)
- [獸人 SerDe](#)
- [木地板 SerDe](#)
- [正則表 SerDe](#)

## Amazon Ion Hive SerDe

您可以使用 Amazon Ion Hive SerDe 來查詢以 [Amazon Ion](#) 格式存放的資料。Amazon Ion 是一種豐富類型、自行描述、開放原始碼資料格式。Amazon Ion 格式由 [Amazon Quantum Ledger Database \(Amazon QLDB\)](#) 等服務和開放原始碼 SQL 查詢語言 [PartiQL](#) 使用。

Amazon Ion 具有可互換的二進位格式和文字格式。此功能結合了文字的易用性和二進位編碼的效率。

要從 Athena 查詢 Amazon Ion 資料，您可以使用 [Amazon Ion Hive SerDe](#)，且其可將 Amazon Ion 資料序列化和還原序列化。還原序列化允許您對 Amazon Ion 資料執行查詢，或讀取資料以便寫出至 Parquet 或 ORC 等不同格式。藉助序列化，您可以透過使用 `CREATE TABLE AS SELECT (CTAS)` 或 `INSERT INTO` 查詢從現有資料表中複製資料，以產生 Amazon Ion 格式的資料。

**Note**

由於 Amazon Ion 是 JSON 的超集，因此您可以使用 Amazon Ion Hive SerDe 來查詢非 Amazon Ion JSON 資料集。與其他 [JSON SerDe 程式庫](#) 不同，Amazon Ion SerDe 不期望每一資料列都位於單獨一行上。如果您想查詢使用「美化顯示」格式的 JSON 資料集，或以其他方式用新行字元分解資料列中的欄位，則此功能非常有用。

如需使用 Athena 查詢 Amazon Ion 的其他資訊和範例，請參閱 [使用 Amazon Athena 分析 Amazon Ion 資料集](#)。

**SerDe 名稱**

- [com.amazon.ionhiveserde.IonHiveSerDe](#)

**考量與限制**

- 重複的欄位 – Amazon Ion 結構井然有序並支援重複的欄位，而 Hive 的 STRUCT<> 和 MAP<> 並非如此。因此，當您從 Amazon Ion 結構中將重複的欄位還原序列化時，將非確定性地選擇一個值，而忽略其他值。
- 不支援外部符號資料表 – 目前，Athena 不支援外部符號資料表或以下 Amazon Ion Hive SerDe 屬性：
  - `ion.catalog.class`
  - `ion.catalog.file`
  - `ion.catalog.url`
  - `ion.symbol_table_imports`
- 檔案副檔名 – Amazon Ion 使用檔案副檔名來確定用於還原序列化 Amazon Ion 檔案的壓縮轉碼器。因此，壓縮檔案必須具有與所使用的壓縮演算法相對應的檔案副檔名。例如，如果使用 ZSTD，則相應的檔案應該具有副檔名 `.zst`。
- 同質資料 – Amazon Ion 對可用於特定欄位中的值的資料類型沒有限制。例如，兩個不同的 Amazon Ion 文件可能包含一個具有不同資料類型及相同名稱的欄位。但是，由於 Hive 使用結構描述，所以擷取到單一 Hive 資料欄的所有值都必須具有相同的資料類型。
- 映射金鑰類型限制 – 將其他格式的資料序列化為 Amazon Ion 時，請確認映射金鑰類型是 STRING、VARCHAR 或 CHAR。雖然 Hive 允許您使用任何基本資料類型作為映射金鑰，但 [Amazon Ion 符號](#) 必須是字串類型。
- Union 類型 – Athena 目前不支援 Hive [Union 類型](#)。

- **雙精確度資料類型** – Amazon Ion 目前不支援 `double` 資料類型。

## 主題

- [使用 CREATE TABLE 建立 Amazon Ion 資料表](#)
- [使用 CTAS 和 INSERT INTO 來建立 Amazon Ion 資料表](#)
- [使用 Amazon Ion SerDe 屬性](#)
- [使用路徑擷取器](#)

## 使用 CREATE TABLE 建立 Amazon Ion 資料表

若要在 Athena 中，從以 Amazon Ion 格式存放的資料中建立資料表，您可以在 CREATE TABLE 陳述式中使用下列其中一項技術：

- 指定 `STORED AS ION`。在這種用法中，您不必 SerDe 明確指定 Amazon 離子蜂巢。這種選擇是更為直接的選項。
- 在 `ROW FORMAT SERDE`、`INPUTFORMAT` 和 `OUTPUTFORMAT` 欄位中指定 Amazon Ion 類別路徑。

您還可以使用 CREATE TABLE AS SELECT (CTAS) 陳述式在 Athena 中建立 Amazon Ion 資料表。如需相關資訊，請參閱[使用 CTAS 和 INSERT INTO 來建立 Amazon Ion 資料表](#)。

## 指定 STORED AS ION

以下 CREATE TABLE 陳述式範例在 LOCATION 子句前使用 `STORED AS ION`，以基於 Amazon Ion 格式的航班資料建立資料表。LOCATION 子句指定了 Ion 格式輸入文件所在的儲存貯體或資料夾。掃描指定位置中的所有檔案。

```
CREATE EXTERNAL TABLE flights_ion (  
  yr INT,  
  quarter INT,  
  month INT,  
  dayofmonth INT,  
  dayofweek INT,  
  flightdate STRING,  
  uniquecarrier STRING,  
  airlineid INT,  
)  
STORED AS ION
```

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

## 指定 Amazon Ion 類別路徑

您可以明確指定 ROW FORMAT SERDE、INPUTFORMAT 和 OUTPUTFORMAT 子句的 Ion 類別路徑值，但不要使用 STORED AS ION 語法，如下所示。

參數	Ion 類別路徑
ROW FORMAT SERDE	'com.amazon.ionhiveserde.IonHiveSerDe'
STORED AS INPUTFORMAT	'com.amazon.ionhiveserde.formats.IonInputFormat'
OUTPUTFORMAT	'com.amazon.ionhiveserde.formats.IonOutputFormat'

以下 DDL 查詢使用此技術建立與上一個範例相同的外部資料表。

```
CREATE EXTERNAL TABLE flights_ion (
  yr INT,
  quarter INT,
  month INT,
  dayofmonth INT,
  dayofweek INT,
  flightdate STRING,
  uniquecarrier STRING,
  airlineid INT,
)
ROW FORMAT SERDE
  'com.amazon.ionhiveserde.IonHiveSerDe'
STORED AS INPUTFORMAT
  'com.amazon.ionhiveserde.formats.IonInputFormat'
OUTPUTFORMAT
  'com.amazon.ionhiveserde.formats.IonOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

如需 Athena 中 CREATE TABLE 陳述式 SerDe 屬性的相關資訊，請參閱 [使用 Amazon Ion SerDe 屬性](#)。

## 使用 CTAS 和 INSERT INTO 來建立 Amazon Ion 資料表

您可以在 Athena 中，使用 CREATE TABLE AS SELECT (CTAS) 和 INSERT INTO 陳述式將資料表中的資料複製或插入到 Amazon Ion 格式的新資料表中。

在 CTAS 查詢中，以 WITH 子句指定 format='ION'，如下列範例所示。

```
CREATE TABLE new_table
WITH (format='ION')
AS SELECT * from existing_table
```

依預設，Athena 將以 [Ion 二進位格式](#) 序列化 Amazon Ion 結果，但您也可以使用文字格式。要使用文字格式，請以 CTAS WITH 子句指定 ion\_encoding = 'TEXT'，如下列範例所示。

```
CREATE TABLE new_table
WITH (format='ION', ion_encoding = 'TEXT')
AS SELECT * from existing_table
```

如需有關 CTAS WITH 子句中的 Amazon Ion 特定屬性的詳細資訊，請參閱下節。

### CTAS WITH 子句 Amazon Ion 屬性

在 CTAS 查詢中，您可以使用 WITH 子句來指定 Amazon Ion 格式，並可選擇指定要使用的 Amazon Ion 編碼及/或寫入壓縮演算法。

#### format

您可以將 ION 關鍵字指定為 CTAS 查詢的 WITH 子句中的格式選項。在執行這項操作時，您建立的資料表將使用您為 IonInputFormat 指定的格式進行讀取，並以您為 IonOutputFormat 指定的格式序列化資料。

以下範例指定 CTAS 查詢使用 Amazon Ion 格式。

```
WITH (format='ION')
```

#### ion\_encoding

選用

預設：BINARY

值：BINARY、TEXT

指定資料是以 Amazon Ion 二進位格式還是以 Amazon Ion 文字格式序列化。以下範例指定了 Amazon Ion 文字格式。

```
WITH (format='ION', ion_encoding='TEXT')
```

## write\_compression

選用

預設：GZIP

值：GZIP、ZSTD、BZIP2、SNAPPY、NONE

指定用於壓縮輸出檔案的壓縮演算法。

以下範例指定了 CTAS 查詢使用 [Zstandard](#) 壓縮演算法以 Amazon Ion 格式寫入其輸出。

```
WITH (format='ION', write_compression = 'ZSTD')
```

如需有關在 Athena 中使用壓縮的資訊，請參閱[Athena 壓縮支援](#)。

如需有關在 Athena 中的其他 CTAS 屬性，請參閱 [CTAS 資料表屬性](#)。

## 使用 Amazon Ion SerDe 屬性

本主題包含 Athena 中 CREATE TABLE 陳述式的 SerDe 屬性的相關資訊。如需 Amazon Ion SerDe 屬性使用的詳細資訊和範例，請參閱 [GitHub](#) 上的 Amazon Ion Hive SerDe 文件中的 [SerDe properties](#) (SerDe 屬性)。

## 指定 Amazon Ion SerDe 屬性

若要在 CREATE TABLE 陳述式中指定 Amazon Ion Hive SerDe 的屬性，請使用 WITH SERDEPROPERTIES 子句。由於 WITH SERDEPROPERTIES 是 ROW FORMAT SERDE 子句的子欄位，您必須先指定 ROW FORMAT SERDE 和 Amazon Ion Hive SerDe 類別路徑，如下列語法所示。

```
...  
ROW FORMAT SERDE  
  'com.amazon.ionhiveserde.IonHiveSerDe'
```

```
WITH SERDEPROPERTIES (  
  'property' = 'value',  
  'property' = 'value',  
  ...  
)
```

請注意，雖然如果您要使用 WITH SERDEPROPERTIES，則 ROW FORMAT SERDE 子句為必要條件，您可以使用 STORED AS ION 或者較長 INPUTFORMAT 和 OUTPUTFORMAT 語法來指定 Amazon Ion 格式。

## Amazon Ion SerDe 屬性

以下是 Amazon Ion SerDe 屬性，可用於 Athena 中的 CREATE TABLE 陳述式。

### ion.encoding

選用

預設：BINARY

值：BINARY、TEXT

此屬性會宣告新增的值是否序列化為 [Amazon Ion 二進位](#) 或 Amazon Ion 文字格式。

以下 SerDe 屬性範例指定了 Amazon Ion 文字格式。

```
'ion.encoding' = 'TEXT'
```

### ion.fail\_on\_overflow

選用

預設：true

值：true、false

Amazon Ion 允許任意大型數值類型，而 Hive 則不允許。依預設，如果 Amazon Ion 值不適合 Hive 資料欄，則 SerDe 將失敗，但您可以使用 fail\_on\_overflow 組態選項來使值溢位而不是失敗。

可以在資料表或資料欄層級設定此屬性。要在資料表層級進行指定，請按如下範例所示指定 ion.fail\_on\_overflow。這樣一來，即會為所有資料欄設定預設行為。

```
'ion.fail_on_overflow' = 'true'
```

要控制特定資料欄，請指定 `ion` 與 `fail_on_overflow` 之間的資料欄名稱，其中由句點分隔，如下列範例所示。

```
'ion.<column>.fail_on_overflow' = 'false'
```

## ion.path\_extractor.case\_sensitive

選用

預設：false

值：true、false

確定是否將 Amazon Ion 欄位名稱視為區分大小寫。當為 false 時，SerDe 將忽略剖析 Amazon Ion 欄位名稱的大小寫。

例如，假設您有 Hive 資料表結構描述，它以小寫形式定義了欄位 `alias`，而 Amazon Ion 文件則包含 `alias` 欄位和 `ALIAS` 欄位，如下列範例所示。

```
-- Hive Table Schema
alias: STRING

-- Amazon Ion Document
{ 'ALIAS': 'value1' }
{ 'alias': 'value2' }
```

以下範例說明了當區分大小寫設定為 false 時的 SerDe 屬性以及產生的擷取資料表：

```
-- Serde properties
'ion.alias.path_extractor' = '(alias)'
'ion.path_extractor.case_sensitive' = 'false'

--Extracted Table
| alias      |
|-----|
| "value1"  |
| "value2"  |
```

以下範例說明了當區分大小寫設定為 true 時的 SerDe 屬性以及產生的擷取資料表：

```
-- Serde properties
'ion.alias.path_extractor' = '(alias)'
'ion.path_extractor.case_sensitive' = 'true'

--Extracted Table
| alias      |
|-----|
| "value2" |
```

在第二種情況下，當區分大小寫設定為 true 且路徑擷取器指定為 alias 時，會忽略 ALIAS 欄位的 value1。

ion.<*column*>.path\_extractor

選用

預設：不適用

值：含搜尋路徑的字串

使用特定資料欄的指定搜尋路徑建立路徑擷取器。路徑擷取器將 Amazon Ion 欄位映射到 Hive 資料欄。如果未指定路徑擷取器，Athena 會根據資料欄名稱在執行時間動態建立路徑擷取器。

以下路徑擷取器範例將 example\_ion\_field 映射至 example\_hive\_column。

```
'ion.example_hive_column.path_extractor' = '(example_ion_field)'
```

如需有關路徑擷取器和搜尋路徑的詳細資訊，請參閱[使用路徑擷取器](#)。

ion.timestamp.serialization\_offset

選用

預設：'Z'

值：OFFSET，其中 OFFSET 表示為 <*signal*>hh:mm。範例值：01:00、+01:00、-09:30、Z (UTC，與 00:00 相同)

與 Apache Hive [時間戳記](#)不同，它們沒有內建時區，並且存放為從 UNIX epoch 的偏移，Amazon Ion 時間戳記確實具有偏移。在序列化到 Amazon Ion 時，使用此屬性指定偏移。

以下範例新增了一小時的偏移。

```
'ion.timestamp.serialization_offset' = '+01:00'
```

### ion.serialize\_null

選用

預設：OMIT

值：OMIT、UNTYPED、TYPED

可以將 Amazon Ion SerDe 設定為序列化或省略具有 null 值的資料欄。您可以選擇寫出強類型的 null 值 (TYPED) 或無類型的 null 值 (UNTYPED)。強類型 null 值是根據預設的 Amazon Ion 到 Hive 類型映射來決定的。

以下範例指定了強類型的 null 值。

```
'ion.serialize_null'='TYPED'
```

### ion.ignore\_malformed

選用

預設：false

值：true、false

當為 true 時，如果 SerDe 無法讀取，則會忽略格式錯誤的項目或整個檔案。如需詳細資訊，請參閱 GitHub 上文件中的[忽略格式錯誤](#)。

### ion.<column>.serialize\_as

選用

預設：資料欄的預設類型。

值：包含 Amazon Ion 類型的字串

確定在其中序列化值的 Amazon Ion 資料類型。由於 Amazon Ion 和 Hive 類型並不總是具有直接映射，因此少數 Hive 類型具有多種用於序列化的有效資料類型。要將資料序列化為非預設資

料類型，請使用此屬性。如需有關類型映射的詳細資訊，請參閱 GitHub 上的 Amazon Ion [Type mapping](#) (類型映射) 頁面。

依預設，二進位 Hive 資料欄序列化為 Amazon Ion blob，但它們也可以序列化為 [Amazon Ion clob](#) (字元大型物件)。以下範例將資料欄 `example_hive_binary_column` 序列化為 `clob`。

```
'ion.example_hive_binary_column.serialize_as' = 'clob'
```

## 使用路徑擷取器

Amazon Ion 是文件樣式檔案格式，而 Apache Hive 是平面直欄式格式。您可以使用稱為在兩種格式之間 `path extractors` 進行映射的特殊 Amazon Ion SerDe 屬性。路徑擷取器會展平階層式 Amazon Ion 格式，將 Amazon Ion 值映射到 Hive 資料欄，並可用於重新命名欄位。

Athena 可以為您產生擷取器，但如有必要，您也可以定義自己的擷取器。

## 產生的路徑擷取器

依預設，Athena 會搜尋與 Hive 資料欄名稱相符的頂層 Amazon Ion 值，並根據這些相符的值在執行時間建立路徑擷取器。如果您的 Amazon Ion 資料格式與 Hive 資料表結構描述相符，Athena 會為您動態產生擷取器，並且您無需新增任何其他路徑擷取器。這些預設路徑擷取器不會存放在資料表中繼資料中。

下列範例說明了 Athena 如何根據資料欄名稱產生擷取器。

```
-- Example Amazon Ion Document
{
  identification: {
    name: "John Smith",
    driver_license: "XXXX"
  },

  alias: "Johnny"
}

-- Example DDL
CREATE EXTERNAL TABLE example_schema2 (
  identification MAP<STRING, STRING>,
  alias STRING
)
STORED AS ION
```

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_extraction1/'
```

下列範例擷取器由 Athena 產生。第一個會將 `identification` 欄位擷取為 `identification` 資料欄，第二個會將 `alias` 欄位擷取為 `alias` 資料欄。

```
'ion.identification.path_extractor' = '(identification)'
'ion.alias.path_extractor' = '(alias)'
```

下列範例說明了擷取的資料表。

```
|          identification          | alias |
|-----|-----|
| [{"name", "driver_license"}, ["John Smith", "XXXX"]] | "Johnny" |
```

### 指定自己的路徑擷取器

如果您的 Amazon Ion 欄位未能整齊地映射到 Hive 資料欄，您可以指定自己的路徑擷取器。在 CREATE TABLE 陳述式的 WITH SERDEPROPERTIES 子句中，使用以下語法。

```
WITH SERDEPROPERTIES (
  "ion.path_extractor.case_sensitive" = "<Boolean>",
  "ion.<column_name>.path_extractor" = "<path_extractor_expression>"
)
```

#### Note

依預設，路徑擷取器不區分大小寫。若要覆寫此設定，請將 [ion.path\\_extractor.case\\_sensitive](#) SerDe 屬性設定為 `true`。

### 在路徑擷取器中使用搜尋路徑

<path\_extractor\_expression> 路徑提取器的 SerDe 屬性語法包含：

```
"ion.<column_name>.path_extractor" = "<path_extractor_expression>"
```

您可以使用 `<path_extractor_expression>` 來指定用於剖析 Amazon Ion 文件並尋找相符資料的搜尋路徑。搜尋路徑含括在括號內，並且可以包含以下一個或多個由空格分隔的元件。

- 萬用字元 – 符合所有值。
- 索引 – 符合指定數字索引位置的值。索引以零為起始。
- 文字 – 符合欄位名稱與指定文字相符的所有值。
- 註釋 – 符合具有指定註釋的包裝路徑元件指定的值。

下列範例說明了 Amazon Ion 文件和一些搜尋路徑範例。

```
-- Amazon Ion document
{
  foo: ["foo1", "foo2"] ,
  bar: "myBarValue",
  bar: A:"annotatedValue"
}

-- Example search paths
(foo 0)      # matches "foo1"
(1)         # matches "myBarValue"
(*)         # matches ["foo1", "foo2"], "myBarValue" and A:"annotatedValue"
()          # matches {foo: ["foo1", "foo2"] , bar: "myBarValue", bar:
A:"annotatedValue"}
(bar)       # matches "myBarValue" and A:"annotatedValue"
(A::bar)    # matches A:"annotatedValue"
```

## 擷取器範例

### 展平和重新命名欄位

下列範例說明了一組展平和重新命名欄位的搜尋路徑。此範例使用搜尋路徑來執行下列動作：

- 將 nickname 資料欄映射至 alias 欄位
- 將 name 資料欄映射至位於 identification 結構中的 name 子欄位。

以下是 Amazon Ion 文件範例。

```
-- Example Amazon Ion Document
{
  identification: {
    name: "John Smith",
    driver_license: "XXXX"
```

```

    },

    alias: "Johnny"
}

```

以下是定義路徑擷取器的範例 CREATE TABLE 陳述式。

```

-- Example DDL Query
CREATE EXTERNAL TABLE example_schema2 (
    name STRING,
    nickname STRING
)
ROW FORMAT SERDE
'com.amazon.ionhiveserde.IonHiveSerDe'
WITH SERDEPROPERTIES (
  'ion.nickname.path_extractor' = '(alias)',
  'ion.name.path_extractor' = '(identification name)'
)
STORED AS ION
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_extraction2/'

```

下列範例說明了擷取的資料。

```

-- Extracted Table
| name          | nickname      |
|-----|-----|
| "John Smith" | "Johnny"     |

```

如需有關搜尋路徑和其他搜尋路徑範例的詳細資訊，請參閱上的 [Ion Java 路徑擷取](#) 頁面 GitHub。

將航班資料擷取為文字格式

以下 CREATE TABLE 查詢範例使用 WITH SERDEPROPERTIES 來新增路徑擷取器，以擷取航班資料並將輸出編碼指定為 Amazon Ion 文字。此範例使用 STORED AS ION 語法。

```

CREATE EXTERNAL TABLE flights_ion (
    yr INT,
    quarter INT,
    month INT,
    dayofmonth INT,
    dayofweek INT,

```

```
    flightdate STRING,
    uniquecarrier STRING,
    airlineid INT,
)
ROW FORMAT SERDE
  'com.amazon.ionhiveserde.IonHiveSerDe'
WITH SERDEPROPERTIES (
  'ion.encoding' = 'TEXT',
  'ion.yr.path_extractor'='(year)',
  'ion.quarter.path_extractor'='(results quarter)',
  'ion.month.path_extractor'='(date month)')
STORED AS ION
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

## Avro SerDe

SerDe 名稱

[Avro SerDe](#)

程式庫名稱

[org.apache.hadoop.hive.serde2.avro.AvroSerDe](http://org.apache.hadoop.hive.serde2.avro.AvroSerDe)

範例

基於安全理由，Athena 不支援使用 `avro.schema.url` 來指定資料表結構描述。請使用 `avro.schema.literal`。要從 Avro 格式的資料中擷取結構描述，請使用 Apache `avro-tools-<version>.jar` 搭配 `getschema` 參數。這會傳回您可以用於 WITH SERDEPROPERTIES 陳述式中的結構描述。例如：

```
java -jar avro-tools-1.8.2.jar getschema my_data.avro
```

`avro-tools-<version>.jar` 檔案位於您安裝之 Avro 版本的 `java` 子目錄中。若要下載 Avro，請參閱 [Apache Avro 版本](#)。若要直接下載 Apache Avro 工具，請參閱 [Apache Avro 工具 Maven 儲存庫](#)。

取得結構描述後，使用 CREATE TABLE 陳述式，根據存放在 Amazon S3 中的基礎 Avro 資料建立 Athena 資料表。若要指定 Avro SerDe，請使用 ROW FORMAT SERDE `'org.apache.hadoop.hive.serde2.avro.AvroSerDe'`。如下列範例所示，除了指定資料欄名稱和資料表對應的資料類型之外，您還須使用 WITH SERDEPROPERTIES 子句來指定結構描述。

**Note**

在 `s3://athena-examples-myregion/path/to/data/` 中，以您執行 Athena 之處的區域識別符 (例如 `s3://athena-examples-us-west-1/path/to/data/`) 取代 *myregion*。

```
CREATE EXTERNAL TABLE flights_avro_example (  
  yr INT,  
  flightdate STRING,  
  uniquecarrier STRING,  
  airlineid INT,  
  carrier STRING,  
  flightnum STRING,  
  origin STRING,  
  dest STRING,  
  depdelay INT,  
  carrierdelay INT,  
  weatherdelay INT  
)  
PARTITIONED BY (year STRING)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'  
WITH SERDEPROPERTIES ('avro.schema.literal'='  
{  
  "type" : "record",  
  "name" : "flights_avro_subset",  
  "namespace" : "default",  
  "fields" : [ {  
    "name" : "yr",  
    "type" : [ "null", "int" ],  
    "default" : null  
  }, {  
    "name" : "flightdate",  
    "type" : [ "null", "string" ],  
    "default" : null  
  }, {  
    "name" : "uniquecarrier",  
    "type" : [ "null", "string" ],  
    "default" : null  
  }, {  
    "name" : "airlineid",  
    "type" : [ "null", "int" ],
```

```
    "default" : null
  }, {
    "name" : "carrier",
    "type" : [ "null", "string" ],
    "default" : null
  }, {
    "name" : "flightnum",
    "type" : [ "null", "string" ],
    "default" : null
  }, {
    "name" : "origin",
    "type" : [ "null", "string" ],
    "default" : null
  }, {
    "name" : "dest",
    "type" : [ "null", "string" ],
    "default" : null
  }, {
    "name" : "depdelay",
    "type" : [ "null", "int" ],
    "default" : null
  }, {
    "name" : "carrierdelay",
    "type" : [ "null", "int" ],
    "default" : null
  }, {
    "name" : "weatherdelay",
    "type" : [ "null", "int" ],
    "default" : null
  } ]
}
')
STORED AS AVRO
LOCATION 's3://athena-examples-myregion/flight/avro/';
```

對資料表執行 `MSCK REPAIR TABLE` 陳述式，以重新整理分割區中繼資料。

```
MSCK REPAIR TABLE flights_avro_example;
```

透過出境總數查詢 10 大出境城市。

```
SELECT origin, count(*) AS total_departures
FROM flights_avro_example
```

```
WHERE year >= '2000'  
GROUP BY origin  
ORDER BY total_departures DESC  
LIMIT 10;
```

### Note

航班資料表資料來自美國運輸部[交通統計局](#)提供的[航班](#)。從原本資料去掉飽和度。

## GROK SerDe

Logstash Grok SerDe 是一個庫，具有一組用於反序列化非結構化文本數據的專門模式，通常是日誌。每個 Grok 模式是具名的規則表達式。您可以視需要識別和重複使用這些還原序列化模式。相較於使用規則表達式，這可讓您更輕鬆地使用 Grok。Grok 提供一組[預先定義的模式](#)。您也可以建立自訂模式。

若要在 Athena 中建立資料表 SerDe 時指定 Grok，請使用 ROW FORMAT SERDE

'com.amazonaws.glue.serde.GrokSerDe' 子句，後面接著指定要在資料中比對之模式的 WITH SERDEPROPERTIES 子句，其中：

- `input.format` 表達式定義資料中要符合的模式。這是必要的。
- `input.grokCustomPatterns` 表達式定義具名的自訂模式，供您後續在 `input.format` 表達式內使用。這是選用的。若要將多個模式項目併入 `input.grokCustomPatterns` 表達式中，請使用新行逸出字元 (`\n`) 來分隔，如下所示：'`input.grokCustomPatterns`'=`'INSIDE_QS ([^\"]*)\nINSIDE_BRACKETS ([^\[]*)'`。
- `STORED AS INPUTFORMAT` 和 `OUTPUTFORMAT` 子句是必要的。
- `LOCATION` 子句會指定 Amazon S3 儲存貯體，其可以包含多個資料物件。儲存貯體中的所有資料物件都還原序列化來建立資料表。

### 範例

這些範例倚賴預先定義的 Grok 模式清單。請參閱[預先定義的模式](#)。

#### 範例 1

此範例使用 `s3://DOC-EXAMPLE-BUCKET/groksample/` 中儲存的 Postfix maillog 項目內的來源資料。

```
Feb  9 07:15:00 m4eastmail postfix/smtpd[19305]: B88C4120838: connect from
unknown[192.168.55.4]
Feb  9 07:15:00 m4eastmail postfix/smtpd[20444]: B58C4330038:
client=unknown[192.168.55.4]
Feb  9 07:15:03 m4eastmail postfix/cleanup[22835]: BDC22A77854: message-
id=<31221401257553.5004389LCBF@m4eastmail.example.com>
```

以下陳述式使用自訂模式和您指定的預先定義模式，在 Athena 中從來源資料建立名為 `mygroktable` 的資料表：

```
CREATE EXTERNAL TABLE `mygroktable` (
  syslogbase string,
  queue_id string,
  syslog_message string
)
ROW FORMAT SERDE
  'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.grokCustomPatterns' = 'POSTFIX_QUEUEID [0-9A-F]{7,12}',
  'input.format' = '%{SYSLOGBASE} %{POSTFIX_QUEUEID:queue_id}:
%{GREEDYDATA:syslog_message}'
)
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/groksample/';
```

首先使用簡單模式，例如 `%{NOTSPACE:column}`，先映射欄，再視需要將欄特殊化。

## 範例 2

在下列範例中，您為 Log4j 日誌建立查詢。範例日誌中的項目採用此格式：

```
2017-09-12 12:10:34,972 INFO - processType=AZ, processId=ABCDEF614B6F5E49,
status=RUN,
threadId=123:amqListenerContainerPool23P:AJ|ABCDE9614B6F5E49||
2017-09-12T12:10:11.172-0700],
executionTime=7290, tenantId=12456, userId=123123f8535f8d76015374e7a1d87c3c,
shard=testapp1,
jobId=12312345e5e7df0015e777fb2e03f3c, messageType=REAL_TIME_SYNC,
```

```
action=receive, hostname=1.abc.def.com
```

查詢此日誌資料：

- 針對每個欄，將 Grok 模式新增到 `input.format`。例如，對於 `timestamp`，新增 `%{TIMESTAMP_ISO8601:timestamp}`。對於 `loglevel`，新增 `%{LOGLEVEL:loglevel}`。
- 映射在日誌格式中用於分隔項目的虛線 (-) 和逗號，以確定 `input.format` 中的模式完全符合日誌的格式。

```
CREATE EXTERNAL TABLE bltest (  
  timestamp STRING,  
  loglevel STRING,  
  processtype STRING,  
  processid STRING,  
  status STRING,  
  threadid STRING,  
  executiontime INT,  
  tenantid INT,  
  userid STRING,  
  shard STRING,  
  jobid STRING,  
  messagetype STRING,  
  action STRING,  
  hostname STRING  
)  
ROW FORMAT SERDE 'com.amazonaws.glue.serde.GrokSerDe'  
WITH SERDEPROPERTIES (  
  "input.grokCustomPatterns" = 'C_ACTION receive|send',  
  "input.format" = "%{TIMESTAMP_ISO8601:timestamp} %{LOGLEVEL:loglevel} - processType=  
  %{NOTSPACE:processtype}, processId=%{NOTSPACE:processid}, status=%{NOTSPACE:status},  
  threadId=%{NOTSPACE:threadid}, executionTime=%{POSINT:executiontime}, tenantId=  
  %{POSINT:tenantid}, userId=%{NOTSPACE:userid}, shard=%{NOTSPACE:shard}, jobId=  
  %{NOTSPACE:jobid}, messageType=%{NOTSPACE:messagetype}, action=%{C_ACTION:action},  
  hostname=%{HOST:hostname}"  
) STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'  
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/samples/';
```

## 範例 3

以下查詢 Amazon S3 日誌的範例顯示 'input.grokCustomPatterns' 表達式，其中包含兩個以新行逸出字元 (\n) 來分隔的模式項目，如查詢範例中的這個片段所示：'input.grokCustomPatterns'='INSIDE\_QS ([^\"]\*)\nINSIDE\_BRACKETS ([^\]]\*)')。

```
CREATE EXTERNAL TABLE `s3_access_auto_raw_02` (
  `bucket_owner` string COMMENT 'from deserializer',
  `bucket` string COMMENT 'from deserializer',
  `time` string COMMENT 'from deserializer',
  `remote_ip` string COMMENT 'from deserializer',
  `requester` string COMMENT 'from deserializer',
  `request_id` string COMMENT 'from deserializer',
  `operation` string COMMENT 'from deserializer',
  `key` string COMMENT 'from deserializer',
  `request_uri` string COMMENT 'from deserializer',
  `http_status` string COMMENT 'from deserializer',
  `error_code` string COMMENT 'from deserializer',
  `bytes_sent` string COMMENT 'from deserializer',
  `object_size` string COMMENT 'from deserializer',
  `total_time` string COMMENT 'from deserializer',
  `turnaround_time` string COMMENT 'from deserializer',
  `referrer` string COMMENT 'from deserializer',
  `user_agent` string COMMENT 'from deserializer',
  `version_id` string COMMENT 'from deserializer')
ROW FORMAT SERDE
  'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.format'='%{NOTSPACE:bucket_owner} %{NOTSPACE:bucket} \
\[%{INSIDE_BRACKETS:time}\\] %{NOTSPACE:remote_ip} %{NOTSPACE:requester}
%{NOTSPACE:request_id} %{NOTSPACE:operation} %{NOTSPACE:key} \"?
%{INSIDE_QS:request_uri}\\"? %{NOTSPACE:http_status} %{NOTSPACE:error_code}
%{NOTSPACE:bytes_sent} %{NOTSPACE:object_size} %{NOTSPACE:total_time}
%{NOTSPACE:turnaround_time} \"?%{INSIDE_QS:referrer}\\"? \"?%{INSIDE_QS:user_agent}\\"?
%{NOTSPACE:version_id}',
  'input.grokCustomPatterns'='INSIDE_QS ([^\"]*)\nINSIDE_BRACKETS ([^\]]*)')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET'
```

## JSON SerDe 程式庫

在 Athena，您可以使用 SerDe 程式庫來還原序列化 JSON 資料。還原序列化會轉換 JSON 資料，讓這些資料能夠序列化 (寫出) 至 Parquet 或 ORC 等不同格式。

- 原生 [蜂巢 SerDe](#)
- [OpenX SerDe](#)
- [Amazon Ion Hive SerDe](#)

### Note

Hive 和 OpenX 程式庫預期 JSON 資料位於單行上 (未格式化)，且記錄由換行字元分隔。Amazon 離子蜂巢 SerDe 沒有該要求，因為離子數據格式是 JSON 的超集合，因此可以用作替代方案。

### 程式庫名稱

請使用下列其中一個：

[或者 .apache.H 目錄. JsonSerDe](#)

[組織. 開發. 數據. JsonSerDe](#)

[am. 亞馬遜. 離子組織. IonHiveSerDe](#)

### 蜂巢 SerDe

蜂巢 JSON 通 SerDe 常用於處理 JSON 數據，如事件。這些事件以 JSON 編碼文字 (以新行分隔) 的單行字串表示。蜂巢 JSON SerDe 不允許在map或struct密鑰名稱中重複的密鑰。

### Note

Serde 預期每個 JSON 文件都位於單行文字上，且記錄中欄位之間沒有行終止字元。如果 JSON 文本是漂亮的打印格式，您可能會收到一條錯誤消息，如 HIVE\_CURSOR\_ERROR：行不是有效的 JSON 對象或 HIVE\_CURSOR\_ERROR:: 意外 JsonParseException end-of-input：當您嘗試查詢表後創建表時，對象的預期關閉標記。如需詳細資訊，請參閱上 GitHub 的 [OpenX SerDe 文件中的 JSON 資料](#) 檔案。

下列範例 DDL 陳述式會使用 Hive JSON SerDe，根據範例線上廣告資料建立資料表。在 LOCATION 子句中，請以您執行 Athena 所在位置的區域識別符 (例如 s3://us-west-2.elasticmapreduce/samples/hive-ads/tables/impressions) 來取代 s3://DOC-EXAMPLE-BUCKET.elasticmapreduce/samples/hive-ads/tables/impressions 中的 *myregion*。

```
CREATE EXTERNAL TABLE impressions (  
    requestbegintime string,  
    adid string,  
    impressionid string,  
    referrer string,  
    useragent string,  
    usercookie string,  
    ip string,  
    number string,  
    processid string,  
    browsercookie string,  
    requestendtime string,  
    timers struct  
        <  
            modellookup:string,  
            requesttime:string  
        >,  
    threadid string,  
    hostname string,  
    sessionid string  
)  
PARTITIONED BY (dt string)  
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'  
LOCATION 's3://DOC-EXAMPLE-BUCKET.elasticmapreduce/samples/hive-ads/tables/  
impressions';
```

## 使用蜂巢 JSON 指定時間戳記格式 SerDe

若要從字串中剖析時間戳記值，您可以將 WITH SERDEPROPERTIES 子欄位新增至 ROW FORMAT SERDE 子句並使用它來指定 timestamp.formats 參數。在參數中，指定一個或多個時間戳記模式的逗號分隔清單，如下範例所示：

```
...  
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'  
WITH SERDEPROPERTIES ("timestamp.formats"="yyyy-MM-dd'T'HH:mm:ss.SSS'Z',yyyy-MM-  
dd'T'HH:mm:ss")
```

...

如需詳細資訊，請參閱 Apache Hive 文件中[時間戳記](#)。

載入資料表以進行查詢

建立資料表後，請執行 [MSCK REPAIR TABLE](#) 以載入資料表，並讓其可從 Athena 查詢：

```
MSCK REPAIR TABLE impressions
```

查詢 CloudTrail 記錄

您可以使用蜂巢 JSON SerDe 查詢 CloudTrail 記錄檔。如需詳細資訊和範例 CREATE TABLE 陳述式，則請參閱 [查詢 AWS CloudTrail 記錄檔](#)。

OpenX SerDe

像蜂巢 JSON 一樣 SerDe，您可以使用 OpenX 的 JSON 來處理 JSON 資料。這些資料也可表示為 JSON 編碼文字 (以新行分隔) 的單行字串。像蜂巢 JSON 一樣 SerDe，OpenX 的 JSON SerDe 不允許重複的鍵map或struct密鑰名稱。

#### Note

Serde 預期每個 JSON 文件都位於單行文字上，且記錄中欄位之間沒有行終止字元。如果 JSON 文本是漂亮的打印格式，您可能會收到一條錯誤消息，如 HIVE\_CURSOR\_ERROR：行不是有效的 JSON 對象或 HIVE\_CURSOR\_ERROR:: 意外 JsonParseException end-of-input：當您嘗試在創建表後查詢對象的預期關閉標記。如需詳細資訊，請參閱上 GitHub 的 OpenX SerDe 文件中的 [JSON 資料](#) 檔案。

可選屬性

與蜂巢 JSON 不同 SerDe，OpenX JSON SerDe 也具有下列選用 SerDe 屬性，可用於解決資料中的不一致性。

ignore.malformed.json

選用。設定為 TRUE 時，可讓您略過格式不正確的 JSON 語法。預設值為 FALSE。

## dots.in.keys

選用。預設值為 FALSE。設定為時 TRUE，允許 SerDe 用底線取代關鍵字名稱中的點。例如，如果 JSON 資料集包含名為 "a.b" 的索引鍵，您可以使用此屬性在 Athena 中將資料欄名稱定義為 "a\_b"。依預設 (如果沒有此選項 SerDe)，Athena 不允許在欄名稱中使用點。

## case.insensitive

選用。預設值為 TRUE。設定為時 TRUE，會 SerDe 將所有大寫欄轉換為小寫。

若要在資料中使用區分大小寫的索引鍵名稱，請使用 WITH SERDEPROPERTIES ("case.insensitive" = FALSE;)。然後，對於不是全部小寫的每個索引鍵，使用下列語法提供從資料欄名稱到屬性名稱的映射：

```
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
WITH SERDEPROPERTIES ("case.insensitive" = "FALSE", "mapping.userid" = "userId")
```

如果您有兩個在小寫時相同的索引鍵，如 url 和 URL，則可能會發生類似以下的錯誤：

HIVE\_CURSOR\_ERROR：列不是有效的 JSON 物件 - JSONException：重複索引鍵 "url"

若要解決這個問題，請將 case.insensitive 屬性設定為 FALSE，並將索引鍵對應至不同的名稱，如下列範例所示：

```
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
WITH SERDEPROPERTIES ("case.insensitive" = "FALSE", "mapping.url1" = "URL",  
"mapping.url2" = "url")
```

## 對應

選用。將欄名稱映射到與欄名稱不完全相同的 JSON 索引鍵。當 JSON 資料包含為 [關鍵字](#) 的索引鍵時，mapping 參數將十分實用。例如，如果您有一個名為 timestamp 的 JSON 索引鍵，請使用下列語法將索引鍵映射至名為 ts 的欄：

```
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
WITH SERDEPROPERTIES ("mapping.ts" = "timestamp")
```

將帶有冒號的巢狀欄位名稱映射到 Hive 相容名稱

如果 struct 內的欄位名稱中有冒號，則可以使用 mapping 屬性將該欄位映射至 Hive 相容名稱。例如，如果您的資料欄類型定義包含 my:struct:field:string，則可透過將下列項目包含在 WITH SERDEPROPERTIES 中，將定義映射至 my\_struct\_field:string：

```
("mapping.my_struct_field" = "my:struct:field")
```

下列範例會顯示對應的 CREATE TABLE 陳述式。

```
CREATE EXTERNAL TABLE colon_nested_field (  
  item struct<my_struct_field:string>  
  ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
  WITH SERDEPROPERTIES ("mapping.my_struct_field" = "my:struct:field")
```

### 範例：廣告資料

下列範例 DDL 陳述式會使用 OpenX JSON，根據範例中用於 Hive JSON 的相同範例線上廣告資料 SerDe 來建立資料表。SerDe 在 LOCATION 子句中，請以您執行 Athena 所在之處的區域識別符取代 *myregion*。

```
CREATE EXTERNAL TABLE impressions (  
  requestbegintime string,  
  adid string,  
  impressionId string,  
  referrer string,  
  useragent string,  
  usercookie string,  
  ip string,  
  number string,  
  processid string,  
  browsercookie string,  
  requestendtime string,  
  timers struct<  
    modellookup:string,  
    requesttime:string>,  
  threadid string,  
  hostname string,  
  sessionid string  
) PARTITIONED BY (dt string)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
LOCATION 's3://DOC-EXAMPLE-BUCKET.elasticmapreduce/samples/hive-ads/tables/  
impressions';
```

## 範例：還原序列化巢狀 JSON

您可以使用 JSON SerDes 來剖析更複雜的 JSON 編碼資料。這需要使用 CREATE TABLE 陳述式，它會使用 struct 和 array 元素來表示巢狀結構。

下列範例會根據具有巢狀結構的 JSON 資料建立 Athena 資料表。該示例具有以下結構：

```
{
  "DocId": "AWS",
  "User": {
    "Id": 1234,
    "Username": "carlos_salazar",
    "Name": "Carlos",
    "ShippingAddress": {
      "Address1": "123 Main St.",
      "Address2": null,
      "City": "Anytown",
      "State": "CA"
    },
    "Orders": [
      {
        "ItemId": 6789,
        "OrderDate": "11/11/2022"
      },
      {
        "ItemId": 4352,
        "OrderDate": "12/12/2022"
      }
    ]
  }
}
```

請記住，OpenX SerDe 希望每個 JSON 記錄都位於單行文本上。存放在 Amazon S3 時，上述範例中的所有資料都應放在單行上，如下所示：

```
{"DocId":"AWS","User":
{"Id":1234,"Username":"carlos_salazar","Name":"Carlos","ShippingAddress" ...
```

下列 CREATE TABLE 陳述式會使用 [Openx-JsonSerDe](#) 搭配 struct 和 array 集合資料類型，為範例資料建立物件群組。

```
CREATE external TABLE complex_json (
```

```

docid string,
`user` struct<
    id:INT,
    username:string,
    name:string,
    shippingaddress:struct<
        address1:string,
        address2:string,
        city:string,
        state:string
    >,
    orders:array<
        struct<
            itemid:INT,
            orderdate:string
        >
    >
>
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/myjsondata/';

```

若要查詢資料表，請使用如下所示的SELECT陳述式。

```

SELECT
    user.name as Name,
    user.shippingaddress.address1 as Address,
    user.shippingaddress.city as City,
    o.itemid as Item_ID, o.orderdate as Order_date
FROM complex_json, UNNEST(user.orders) as temp_table (o)

```

若要存取結構內的資料欄位，範例查詢會使用點標記法 (例如，`user.name`)。要訪問結構數組內的數據 (與 `orders` 字段一樣)，可以使用該 `UNNEST` 函數。該 `UNNEST` 函數將數組扁平化為臨時表 (在這種情況下稱為 `o`)。這可讓您像使用結構一樣使用點符號來存取非巢狀陣列元素 (例如)。 `o.itemid` 範例中為了說明目的而使用的名稱 `temp_table` 通常會縮寫為 `t`。

下表顯示了查詢結果。

#	名稱	Address	City	項目識別碼	訂單日期
1	Carlos	123 主要 聖彼得堡	任何城 鎮	6789	11/11/2022
2	Carlos	123 主要 聖彼得堡	任何城 鎮	4352	12/12/2022

## 其他資源

如需有關在 Athena 中使用 JSON 和巢狀 JSON 的詳細資訊，請參閱下列資源：

- 使用 JSONSerDe ( AWS 大數據博客 ) [從嵌套 JSON 和映射在 Amazon Athena 中創建表](#)
- [我嘗試在 Amazon Athena 讀取 JSON 資料時收到錯誤](#) 訊息 (AWS 知識中心文章)
- [hive-json-schema](#)(GitHub) — 使用 Java 編寫的工具，可從範例 JSON 文件產生 CREATE TABLE 陳述式。產生的 CREATE TABLE 陳述式會使用 OpenX JSON Serde。

## LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案

指定這 SerDe 是選擇性的。這是 Athena SerDe 預設使用的 CSV、TSV 和自訂分隔格式的資料。如果您不指定任何 SerDe 並且僅指定，則使用此 SerDe 選項 ROW FORMAT DELIMITED。SerDe 如果您的數據沒有用引號括起來的值，請使用此選項。

如需有關的參考文件 LazySimpleSerDe，請參閱 Apache [Hive](#) 開發人員指南的 Hive — SerDe 節。

## 程式庫名稱

是的類別程式庫名 LazySimpleSerDe

稱 org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe。GitHub 如需有關 LazySimpleSerDe 類別的資訊，請參閱 [LazySimpleSerDe.java](#) 上的 .java。

## 忽略標頭

若要在定義資料表時忽略資料中的標題，您可以使用 skip.header.line.count 資料表屬性，如下範例所示。

```
TBLPROPERTIES ("skip.header.line.count"="1")
```

如需相關範例，請參閱 [查詢 Amazon VPC 流程日誌](#) 和 [查詢 Amazon CloudFront 日誌](#) 中的 CREATE TABLE 陳述式。

## CSV 範例

以下範例說明如何使用 LazySimpleSerDe 以在 Athena 中根據 CSV 資料建立資料表。要使用它還原序列化自定義分隔的文件 SerDe，請按照示例中的模式進行操作，但使用子 FIELDS TERMINATED BY 句來指定不同的單字符分隔符。LazySimpleSerDe 不支援多字元分隔符號。

### Note

在 `s3://athena-examples-myregion/path/to/data/` 中，以您執行 Athena 之處的區域識別符 (例如 `s3://athena-examples-us-west-1/path/to/data/`) 取代 *myregion*。

使用 CREATE TABLE 陳述式，根據存放於 Amazon S3 的 CSV 基礎資料建立 Athena 資料表。

```
CREATE EXTERNAL TABLE flight_delays_csv (  
  yr INT,  
  quarter INT,  
  month INT,  
  dayofmonth INT,  
  dayofweek INT,  
  flightdate STRING,  
  uniquecarrier STRING,  
  airlineid INT,  
  carrier STRING,  
  tailnum STRING,  
  flightnum STRING,  
  originairportid INT,  
  originairportseqid INT,  
  origincitymarketid INT,  
  origin STRING,  
  origincityname STRING,  
  originstate STRING,  
  originstatefips STRING,  
  originstatename STRING,  
  originwac INT,  
  destairportid INT,  
  destairportseqid INT,  
  destcitymarketid INT,
```

```
dest STRING,  
destcityname STRING,  
deststate STRING,  
deststatefips STRING,  
deststatename STRING,  
destwac INT,  
crsdeptime STRING,  
deptime STRING,  
depdelay INT,  
depdelayminutes INT,  
depdel15 INT,  
departuredelaygroups INT,  
deptimeblk STRING,  
taxiout INT,  
wheelsoff STRING,  
wheelson STRING,  
taxiin INT,  
crsarrrtime INT,  
arrtime STRING,  
arrdelay INT,  
arrdelayminutes INT,  
arrdel15 INT,  
arrivaldelaygroups INT,  
arrtimeblk STRING,  
cancelled INT,  
cancellationcode STRING,  
diverted INT,  
crselapsedtime INT,  
actualelapsedtime INT,  
airtime INT,  
flights INT,  
distance INT,  
distancegroup INT,  
carrierdelay INT,  
weatherdelay INT,  
nasdelay INT,  
securitydelay INT,  
lateaircraftdelay INT,  
firstdeptime STRING,  
totaladdgtime INT,  
longestaddgtime INT,  
divairportlandings INT,  
divreacheddest INT,  
divactualelapsedtime INT,
```

```
divarrdelay INT,  
divdistance INT,  
div1airport STRING,  
div1airportid INT,  
div1airportseqid INT,  
div1wheelson STRING,  
div1totalgtime INT,  
div1longestgtime INT,  
div1wheelsoff STRING,  
div1tailnum STRING,  
div2airport STRING,  
div2airportid INT,  
div2airportseqid INT,  
div2wheelson STRING,  
div2totalgtime INT,  
div2longestgtime INT,  
div2wheelsoff STRING,  
div2tailnum STRING,  
div3airport STRING,  
div3airportid INT,  
div3airportseqid INT,  
div3wheelson STRING,  
div3totalgtime INT,  
div3longestgtime INT,  
div3wheelsoff STRING,  
div3tailnum STRING,  
div4airport STRING,  
div4airportid INT,  
div4airportseqid INT,  
div4wheelson STRING,  
div4totalgtime INT,  
div4longestgtime INT,  
div4wheelsoff STRING,  
div4tailnum STRING,  
div5airport STRING,  
div5airportid INT,  
div5airportseqid INT,  
div5wheelson STRING,  
div5totalgtime INT,  
div5longestgtime INT,  
div5wheelsoff STRING,  
div5tailnum STRING  
)  
PARTITIONED BY (year STRING)
```

```
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  ESCAPED BY '\\\'
  LINES TERMINATED BY '\n'
LOCATION 's3://athena-examples-myregion/flight/csv/';
```

每次有新的分割區新增到這個資料表時，請執行 `MSCK REPAIR TABLE` 來重新整理分割區中繼資料：

```
MSCK REPAIR TABLE flight_delays_csv;
```

查詢延誤超過 1 小時的前 10 個航線：

```
SELECT origin, dest, count(*) as delays
FROM flight_delays_csv
WHERE depdelayminutes > 60
GROUP BY origin, dest
ORDER BY 3 DESC
LIMIT 10;
```

#### Note

航班資料表資料來自美國運輸部[交通統計局](#)提供的[航班](#)。從原本資料去掉飽和度。

## TSV 範例

若要根據存放於 Amazon S3 的 TSV 資料建立 Athena 資料表，請使用 `ROW FORMAT DELIMITED`，然後將 `\t` 指定為標籤欄位分隔符號、將 `\n` 指定為行分隔符號，以及將 `\` 指定為逸出字元。此語法如下列範例所示。athena-examples 位置沒有可用的範例 TSV 航班資料，但是如同 CSV 資料表一樣，每次新增分割區時，您都會執行 `MSCK REPAIR TABLE` 重新整理分割區中繼資料。

```
...
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
ESCAPED BY '\\\'
LINES TERMINATED BY '\n'
...
```

## SerDe 用於處理 CSV 的開啟 CSV

當您為 CSV 資料建立 Athena 表格時，請根據資料包含的值類型決定 SerDe 要使用的資料：

- 如果您的資料包含以雙引號 (") 括住的值，您可以使用 [OpenCSV SerDe](#) 來還原序列化 Athena 中的值。如果您的資料不包含以雙引號 (") 括住的值，您可以省略指定 any SerDe。在這種情況下，Athena 會使用預設 LazySimpleSerDe。如需相關資訊，請參閱[LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案](#)。
- 如果您的資料具有 UNIX 數TIMESTAMP值 (例如，1579059880000)，請使用 OpenCSV SerDe。如果您的資料使用該java.sql.Timestamp格式，請使用 LazySimpleSerDe。

## CSV SerDe (OpenCSVSerDe)

[OpenCSV SerDe](#) 具有字串資料的下列特性：

- 使用雙引號 (") 做為預設的引號字元，並且可讓您指定分隔符號、引號和逸出字元，例如：

```
WITH SERDEPROPERTIES ("separatorChar" = ",", "quoteChar" = "`", "escapeChar" = "\\")
```

- 無法直接逸出 \t 或 \n。若要將它們逸出，請使用 "escapeChar" = "\\\"。請參閱本主題中的範例。
- 不支援在 CSV 檔案中使用內嵌的換行符號。

對於以外的資料類型STRING，OpenCSV 的SerDe 行為如下：

- 識別 BOOLEAN、BIGINT、INT，以及 DOUBLE 資料類型。
- 無法識別資料欄中定義為數值資料類型的空值或 null 值，而會將其保留為 string。一種解決方法是建立帶有 null 值的資料欄作為 string，然後使用 CAST 將查詢中的欄位轉換為數字資料類型，並為 null 提供 0 預設值。如需詳細資訊，請參閱[當我在 Athena 查詢 CSV 資料時，出現錯誤 HIVE\\_BAD\\_DATA：在知識中心剖析欄位值時發生錯誤](#)。AWS
- 對於使用 CREATE TABLE 陳述式中 timestamp 資料類型指定的資料欄，如果其是以 UNIX 數字格式指定 (以毫秒為單位)，例如 1579059880000，請識別 TIMESTAMP 資料。
  - OpenCSV SerDe 不支援與 JDB 相容TIMESTAMP的java.sql.Timestamp格式，例如 "YYYY-MM-DD HH:MM:SS.ffffffff" (9 位小數位數)。
- 對於使用 CREATE TABLE 陳述式中 DATE 資料類型指定的資料欄，如果這些值代表 1970 年 1 月 1 日以來經過的天數，則請將這些值識別為日期。例如，資料欄中具有 date 資料類型的值 18276 會在查詢時呈現為 2020-01-15。在此 UNIX 格式中，每一天都會被認為有 86,400 秒。
  - OpenCSV SerDe 不直接支援DATE任何其他格式。若要處理其他格式的時間戳記資料，您可以將資料欄定義為 string，然後使用時間轉換函數在 SELECT 查詢中傳回所需結果。如需詳細資

訊，請參閱 [AWS 知識中心](#) 中的 [當我在 Amazon Athena 中查詢資料表時，TIMESTAMP 結果為空](#)。

- 若要進一步將資料欄轉換為資料表中所需的類型，您可以對資料表 [建立檢視](#)，並使用 CAST 來轉換為所需的類型。

Example 範例：使用以 UNIX 數字格式指定的 TIMESTAMP (時間戳記) 類型和 DATE (日期) 類型。

請考慮以下以逗號分隔資料的三個資料欄。每個資料欄中的值皆以雙引號括住。

```
"unixvalue creationdate 18276 creationdatetime 1579059880000","18276","1579059880000"
```

以下陳述式會從指定的 Amazon S3 儲存貯體位置在 Athena 中建立資料表。

```
CREATE EXTERNAL TABLE IF NOT EXISTS testtimestamp1(
  `profile_id` string,
  `creationdate` date,
  `creationdatetime` timestamp
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
LOCATION 's3://DOC-EXAMPLE-BUCKET'
```

下一步，執行下列查詢：

```
SELECT * FROM testtimestamp1
```

查詢會傳回下列結果，顯示日期和時間資料：

profile_id	creationdate
creationdatetime	
unixvalue creationdate 18276 creationdatetime 1579146280000	2020-01-15
2020-01-15 03:44:40.000	

Example 範例：逸出 \t 或 \n

考量下列測試資料：

```
" \\t\\t\\n 123 \\t\\t\\n ",abc
" 456 ",xyz
```

下列陳述式會在 Athena 中建立資料表，並指定 "escapeChar" = "\\\"。

```
CREATE EXTERNAL TABLE test1 (
  f1 string,
  s2 string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ("separatorChar" = ",", "escapeChar" = "\\")
LOCATION 's3://DOC-EXAMPLE-BUCKET/dataset/test1/'
```

下一步，執行下列查詢：

```
SELECT * FROM test1;
```

它會傳回此結果，正確地逸出 \t 或 \n：

```
f1          s2
\t\t\n 123 \t\t\n      abc
456                      xyz
```

SerDe 名稱

### [CSV 文件 SerDe](#)

程式庫名稱

要使用它 SerDe，請在後面指定其完全限定的類名 ROW FORMAT SERDE。此外，在 SERDEPROPERTIES 內指定分隔符號，如下所示：

```
...
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  "separatorChar" = ",",
  "quoteChar"     = "\"",
  "escapeChar"   = "\\")
)
```

忽略標頭

若要在定義資料表時忽略資料中的標題，您可以使用 `skip.header.line.count` 資料表屬性，如下範例所示。

```
TBLPROPERTIES ("skip.header.line.count"="1")
```

如需相關範例，請參閱 [查詢 Amazon VPC 流程日誌](#) 和 [查詢 Amazon CloudFront 日誌](#) 中的 CREATE TABLE 陳述式。

## 範例

此範例假設 CSV 中的資料隨著下列內容儲存在 `s3://DOC-EXAMPLE-BUCKET/mycsv/`：

```
"a1","a2","a3","a4"
"1","2","abc","def"
"a","a1","abc3","ab4"
```

使用 CREATE TABLE 陳述式以根據資料建立 Athena 資料表。在之後參照 OpenCSV SerDe 類別，ROW FORMAT SERDE 並在中指定字元分隔符號、引號字元和逸出字元 WITH SERDEPROPERTIES，如下列範例所示。

```
CREATE EXTERNAL TABLE myopencsvtable (
  col1 string,
  col2 string,
  col3 string,
  col4 string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  'separatorChar' = ',',
  'quoteChar' = '"',
  'escapeChar' = '\\\
'
)
STORED AS TEXTFILE
LOCATION 's3://DOC-EXAMPLE-BUCKET/mycsv/';
```

查詢資料表中的所有值：

```
SELECT * FROM myopencsvtable;
```

此查詢會傳回下列值：

col1	col2	col3	col4
a1	a2	a3	a4
1	2	abc	def
a	a1	abc3	ab4

## 獸人 SerDe

SerDe 名稱

OrcSerDe

程式庫名稱

這個庫使用阿帕奇蜂巢 [OrcSerde.java](#) 類在 ORC 格式的數據。它會將物件從 ORC 傳遞至讀取器，也會從 ORC 傳遞至寫入器。

範例

### Note

在 `s3://athena-examples-myregion/path/to/data/` 中，以您執行 Athena 之處的區域識別符 (例如 `s3://athena-examples-us-west-1/path/to/data/`) 取代 *myregion*。

以下範例以 ORC 為班機延誤資料建立資料表。資料表包含分割區：

```
DROP TABLE flight_delays_orc;
CREATE EXTERNAL TABLE flight_delays_orc (
  yr INT,
  quarter INT,
  month INT,
  dayofmonth INT,
  dayofweek INT,
  flightdate STRING,
  uniquecarrier STRING,
  airlineid INT,
  carrier STRING,
  tailnum STRING,
  flightnum STRING,
  originairportid INT,
  originairportseqid INT,
  origincitymarketid INT,
  origin STRING,
  origincityname STRING,
  originstate STRING,
  originstatefips STRING,
  originstatename STRING,
```

```
originwac INT,  
destairportid INT,  
destairportseqid INT,  
destcitymarketid INT,  
dest STRING,  
destcityname STRING,  
deststate STRING,  
deststatefips STRING,  
deststatename STRING,  
destwac INT,  
crsdeptime STRING,  
deptime STRING,  
depdelay INT,  
depdelayminutes INT,  
depdel15 INT,  
departuredelaygroups INT,  
deptimeblk STRING,  
taxiout INT,  
wheelsoff STRING,  
wheelson STRING,  
taxiin INT,  
crsarrrtime INT,  
arrtime STRING,  
arrdelay INT,  
arrdelayminutes INT,  
arrdel15 INT,  
arrivaldelaygroups INT,  
arrtimeblk STRING,  
cancelled INT,  
cancellationcode STRING,  
diverted INT,  
crselapsedtime INT,  
actualelapsedtime INT,  
airtime INT,  
flights INT,  
distance INT,  
distancegroup INT,  
carrierdelay INT,  
weatherdelay INT,  
nasdelay INT,  
securitydelay INT,  
lateaircraftdelay INT,  
firstdeptime STRING,  
totaladdgtime INT,
```

```
longestaddgtime INT,  
divairportlandings INT,  
divreacheddest INT,  
divactualelapsedtime INT,  
divarrdelay INT,  
divdistance INT,  
div1airport STRING,  
div1airportid INT,  
div1airportseqid INT,  
div1wheelson STRING,  
div1totalgtime INT,  
div1longestgtime INT,  
div1wheelsoff STRING,  
div1tailnum STRING,  
div2airport STRING,  
div2airportid INT,  
div2airportseqid INT,  
div2wheelson STRING,  
div2totalgtime INT,  
div2longestgtime INT,  
div2wheelsoff STRING,  
div2tailnum STRING,  
div3airport STRING,  
div3airportid INT,  
div3airportseqid INT,  
div3wheelson STRING,  
div3totalgtime INT,  
div3longestgtime INT,  
div3wheelsoff STRING,  
div3tailnum STRING,  
div4airport STRING,  
div4airportid INT,  
div4airportseqid INT,  
div4wheelson STRING,  
div4totalgtime INT,  
div4longestgtime INT,  
div4wheelsoff STRING,  
div4tailnum STRING,  
div5airport STRING,  
div5airportid INT,  
div5airportseqid INT,  
div5wheelson STRING,  
div5totalgtime INT,  
div5longestgtime INT,
```

```
div5wheelsoff STRING,  
div5tailnum STRING  
)  
PARTITIONED BY (year String)  
STORED AS ORC  
LOCATION 's3://athena-examples-myregion/flight/orc/'  
tblproperties ("orc.compress"="ZLIB");
```

在資料表上執行 `MSCK REPAIR TABLE` 陳述式，以重新整理分割區中繼資料：

```
MSCK REPAIR TABLE flight_delays_orc;
```

使用此查詢以取得延誤超過 1 小時的前 10 個航線：

```
SELECT origin, dest, count(*) as delays  
FROM flight_delays_orc  
WHERE depdelayminutes > 60  
GROUP BY origin, dest  
ORDER BY 3 DESC  
LIMIT 10;
```

## 木地板 SerDe

Serde 名稱

ParquetHiveSerDe 用於存儲在[鑲木地板格式](#)的數據。

### Note

要將資料轉換為 Parquet 格式，您可以使用 [CREATE TABLE AS SELECT \(CTAS\)](#) 查詢。如需詳細資訊，請參閱[從查詢結果建立資料表 \(CTAS\)](#)、[CTAS 查詢的範例](#)及[使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析](#)。

程式庫名稱

Athena 在需要還原序列化以 Parquet 存放的資料時會使用以下類

別：`org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe`

## 範例：查詢以 Parquet 存放的檔案

### Note

在 `s3://athena-examples-myregion/path/to/data/` 中，以您執行 Athena 之處的區域識別符 (例如 `s3://athena-examples-us-west-1/path/to/data/`) 取代 *myregion*。

使用下列 CREATE TABLE 陳述式，從 Amazon S3 中以實木複合格式儲存的基礎資料建立 Athena 資料表：

```
CREATE EXTERNAL TABLE flight_delays_pq (  
  yr INT,  
  quarter INT,  
  month INT,  
  dayofmonth INT,  
  dayofweek INT,  
  flightdate STRING,  
  uniquecarrier STRING,  
  airlineid INT,  
  carrier STRING,  
  tailnum STRING,  
  flightnum STRING,  
  originairportid INT,  
  originairportseqid INT,  
  origincitymarketid INT,  
  origin STRING,  
  origincityname STRING,  
  originstate STRING,  
  originstatefips STRING,  
  originstatename STRING,  
  originwac INT,  
  destairportid INT,  
  destairportseqid INT,  
  destcitymarketid INT,  
  dest STRING,  
  destcityname STRING,  
  deststate STRING,  
  deststatefips STRING,  
  deststatename STRING,  
  destwac INT,
```

```
crsdeptime STRING,  
deptime STRING,  
depdelay INT,  
depdelayminutes INT,  
depdel15 INT,  
departuredelaygroups INT,  
deptimeblk STRING,  
taxiout INT,  
wheelsoff STRING,  
wheelson STRING,  
taxiin INT,  
crsarrrtime INT,  
arrtime STRING,  
arrdelay INT,  
arrdelayminutes INT,  
arrdel15 INT,  
arrivaldelaygroups INT,  
arrtimeblk STRING,  
cancelled INT,  
cancellationcode STRING,  
diverted INT,  
crselapsedtime INT,  
actualelapsedtime INT,  
airtime INT,  
flights INT,  
distance INT,  
distancegroup INT,  
carrierdelay INT,  
weatherdelay INT,  
nasdelay INT,  
securitydelay INT,  
lateaircraftdelay INT,  
firstdeptime STRING,  
totaladdgtime INT,  
longestaddgtime INT,  
divairportlandings INT,  
divreacheddest INT,  
divactualelapsedtime INT,  
divarrdelay INT,  
divdistance INT,  
div1airport STRING,  
div1airportid INT,  
div1airportseqid INT,  
div1wheelson STRING,
```

```
div1totalgtime INT,  
div1longestgtime INT,  
div1wheelsoff STRING,  
div1tailnum STRING,  
div2airport STRING,  
div2airportid INT,  
div2airportseqid INT,  
div2wheelson STRING,  
div2totalgtime INT,  
div2longestgtime INT,  
div2wheelsoff STRING,  
div2tailnum STRING,  
div3airport STRING,  
div3airportid INT,  
div3airportseqid INT,  
div3wheelson STRING,  
div3totalgtime INT,  
div3longestgtime INT,  
div3wheelsoff STRING,  
div3tailnum STRING,  
div4airport STRING,  
div4airportid INT,  
div4airportseqid INT,  
div4wheelson STRING,  
div4totalgtime INT,  
div4longestgtime INT,  
div4wheelsoff STRING,  
div4tailnum STRING,  
div5airport STRING,  
div5airportid INT,  
div5airportseqid INT,  
div5wheelson STRING,  
div5totalgtime INT,  
div5longestgtime INT,  
div5wheelsoff STRING,  
div5tailnum STRING  
)  
PARTITIONED BY (year STRING)  
STORED AS PARQUET  
LOCATION 's3://athena-examples-myregion/flight/parquet/'  
tblproperties ("parquet.compression"="SNAPPY");
```

在資料表上執行 `MSCK REPAIR TABLE` 陳述式，以重新整理分割區中繼資料：

```
MSCK REPAIR TABLE flight_delays_pq;
```

查詢延誤超過 1 小時的前 10 個航線：

```
SELECT origin, dest, count(*) as delays
FROM flight_delays_pq
WHERE depdelayminutes > 60
GROUP BY origin, dest
ORDER BY 3 DESC
LIMIT 10;
```

### Note

航班資料表資料來自美國運輸部[交通統計局](#)提供的[航班](#)。從原本資料去掉飽和度。

忽略 Parquet 統計資料

當您讀取 Parquet 資料時，您可能會收到下列錯誤訊息：

```
HIVE_CANNOT_OPEN_SPLIT: Index x out of bounds for length y
HIVE_CURSOR_ERROR: Failed to read x bytes
HIVE_CURSOR_ERROR: FailureException at Malformed input: offset=x
HIVE_CURSOR_ERROR: FailureException at java.io.IOException:
can not read class org.apache.parquet.format.PageHeader: Socket is closed by peer.
```

若要解決此問題，請使用[CREATE TABLE](#)或[ALTER TABLE SET TBLPROPERTIES](#)陳述式將 Parquet SerDe `parquet.ignore.statistics` 屬性設定為 `true`，如下列範例所示。

CREATE TABLE 範例

```
...
ROW FORMAT SERDE
'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'
WITH SERDEPROPERTIES (
'parquet.ignore.statistics'='true')
STORED AS PARQUET
...
```

ALTER TABLE 範例

```
ALTER TABLE ... SET TBLPROPERTIES ('parquet.ignore.statistics'='true')
```

## 正則表 SerDe

正則表達式 SerDe 使用正則表達式 ( regex ) 通過將正則表達式組提取到表列中來反序列化數據。

如果資料中的某一列不符合 regex，則該列中的所有欄都會以 NULL 的形式傳回。如果某一列符合 regex 但擁有的群組數量低於預期，則遺失的群組為 NULL。如果資料中的某一列符合 regex 但具有比 regex 中的群組更多的欄，則會忽略額外的欄。

如需詳細資訊，請參閱 Apache Hive 文件 [RegexSerDe](#) 中的 [類別](#)。

### SerDe 名稱

### RegexSerDe

### 程式庫名稱

### RegexSerDe

### 範例

下列範例會使用 CloudFront 記錄檔建立資料表 RegExSerDe。在 `s3://athena-examples-myregion/cloudfront/plaintext/` 中，以您執行 Athena 之處的區域識別符 (例如 `s3://athena-examples-us-west-1/cloudfront/plaintext/`) 取代 *myregion*。

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (  
  `Date` DATE,  
  Time STRING,  
  Location STRING,  
  Bytes INT,  
  RequestIP STRING,  
  Method STRING,  
  Host STRING,  
  Uri STRING,  
  Status INT,  
  Referrer STRING,  
  os STRING,  
  Browser STRING,  
  BrowserVersion STRING  
)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
```



- [查詢陣列](#)
- [查詢地理空間資料](#)
- [查詢 JSON](#)
- [搭配 Amazon Athena 使用機器學習 \(ML\)](#)
- [利用使用者定義函數來查詢](#)
- [跨區域查詢](#)
- [查詢 AWS Glue Data Catalog](#)
- [查詢 AWS 服務 記錄檔](#)
- [查詢 Amazon S3 中存放的 Web 伺服器日誌](#)

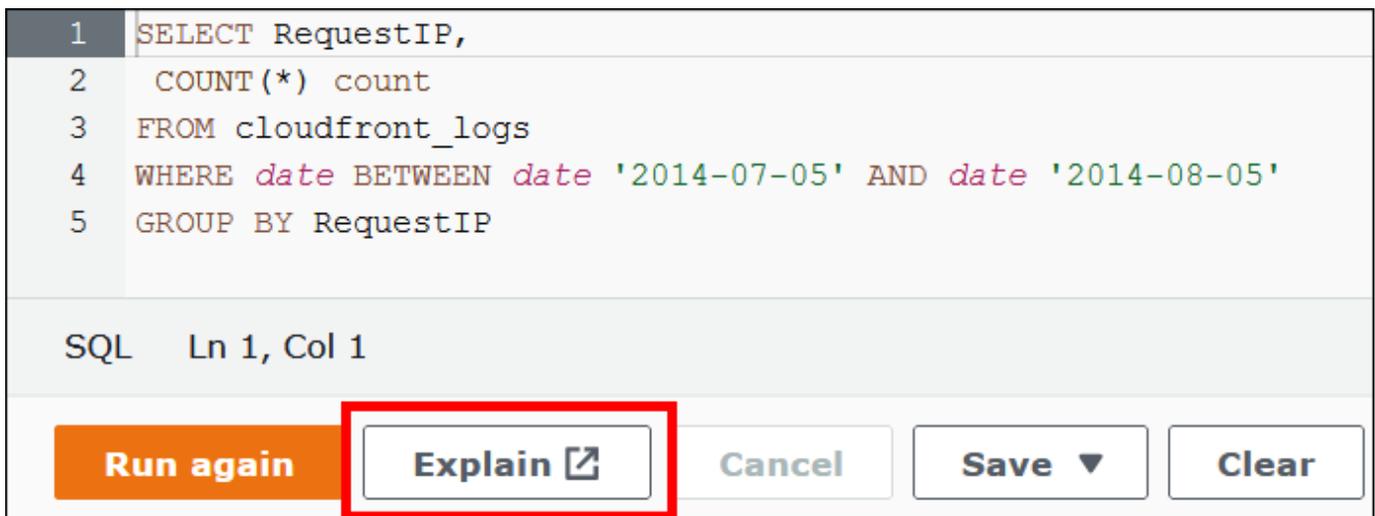
如需注意事項和限制，請參閱 [在 Amazon Athena 中進行 SQL 查詢的考量事項與限制](#)。

## 檢視 SQL 查詢的執行計劃

您可以使用 Athena 查詢編輯器來查看查詢執行方式的圖形呈現。當您在編輯器中輸入查詢並選擇 Explain (解釋) 選項時，Athena 會在您的查詢上使用 [EXPLAIN](#) SQL 陳述式來建立兩個相應的圖形：分散式執行計劃和邏輯執行計劃。您可以使用這些圖形來分析、疑難排解並提高查詢的效率。

若要檢視查詢的執行計劃

1. 在 Athena 查詢編輯器中輸入查詢，然後選擇 Explain (解釋)。



Distributed plan (分散式計劃) 索引標籤會顯示查詢在分散式環境中的執行計劃。分散式計劃具有處理片段或階段。每個階段都有從零開始的索引編號，而且會由一或多個節點處理。資料可以在節點之間進行交換。

Amazon Athena > Query editor > Explain

# Explain

**Distributed plan** | Logical plan

---

## Distributed plan

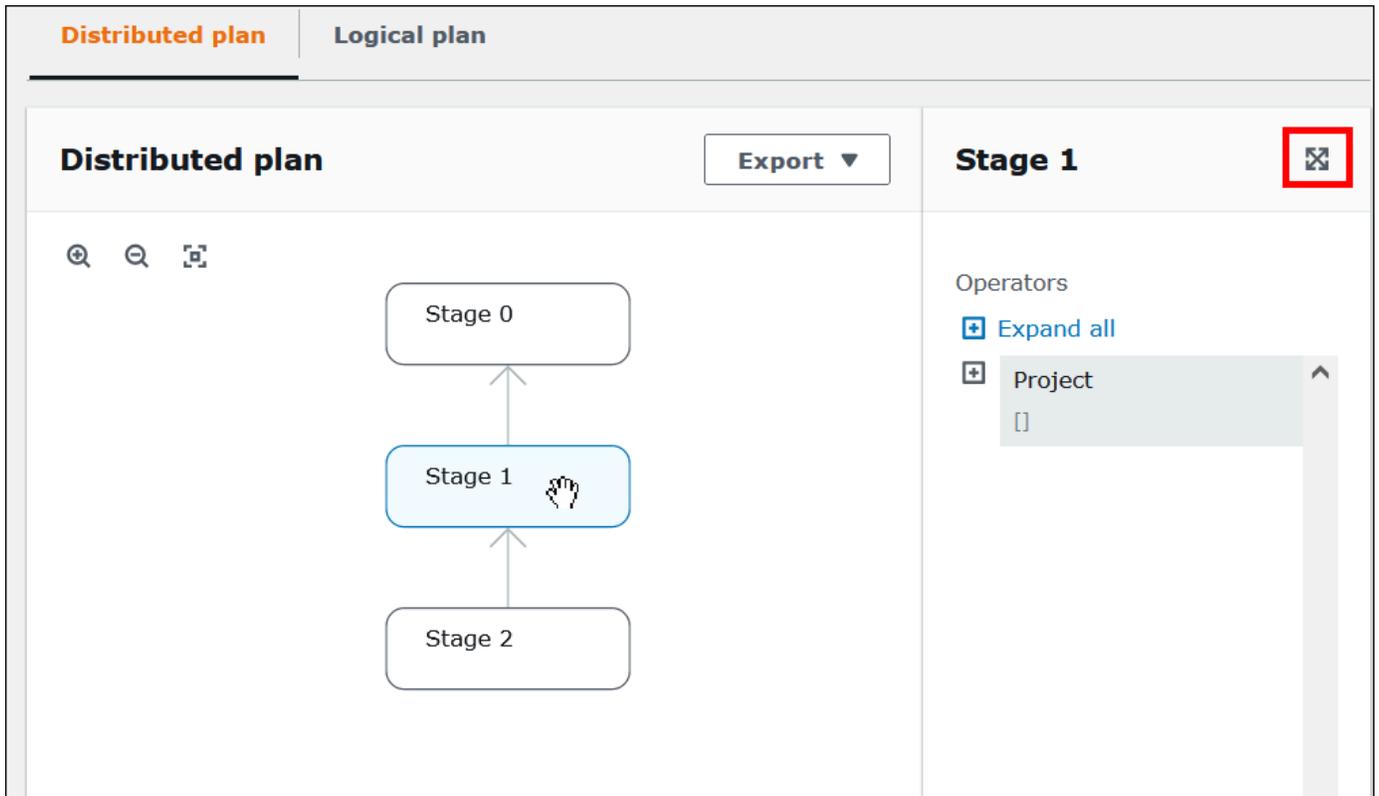
Export ▼

🔍 🔍 🖨

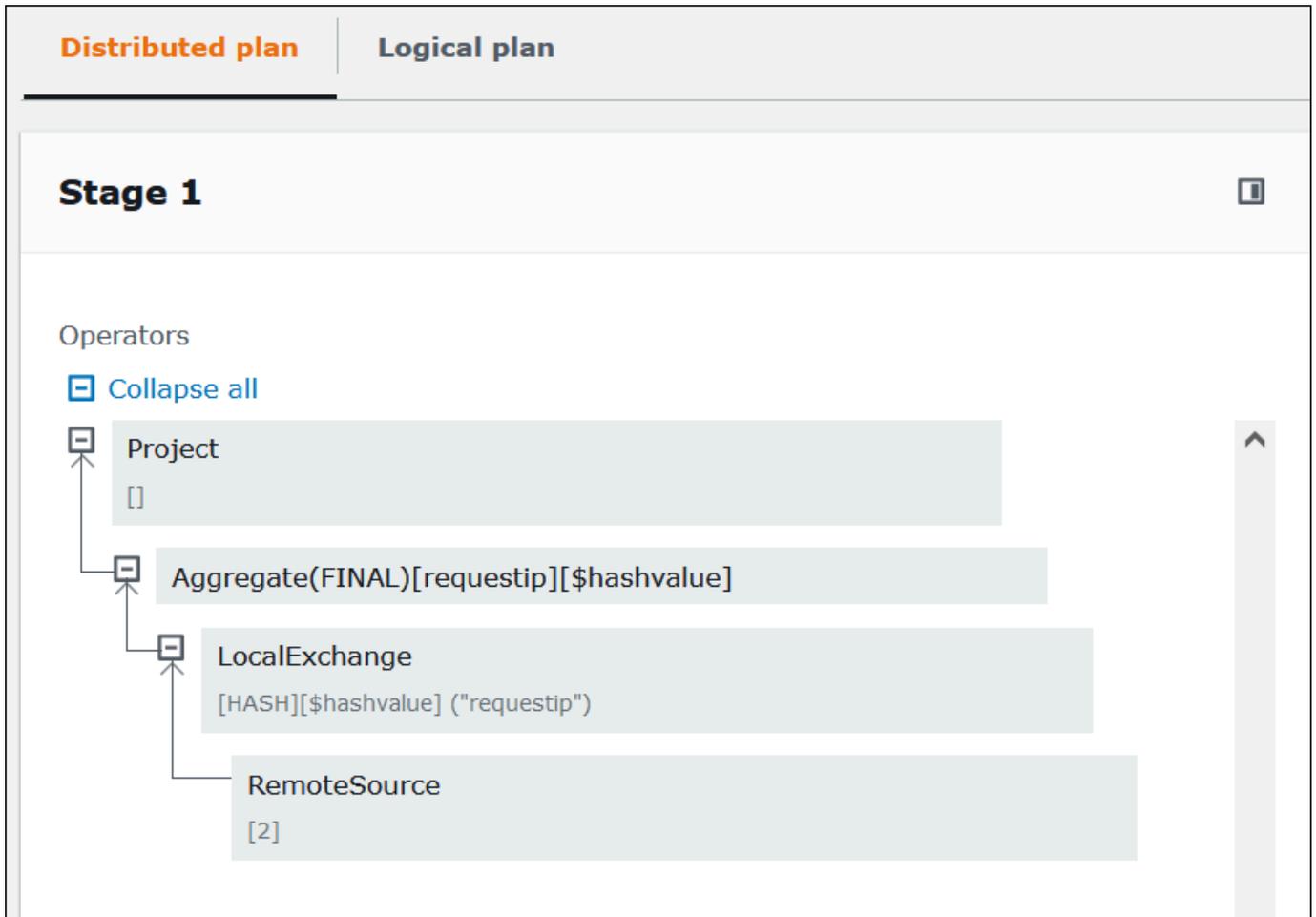
```
graph BT; S2[Stage 2] --> S1[Stage 1]; S1 --> S0[Stage 0];
```

**No stage selected**  
Select a stage to view execution details

- 若要導覽圖形，請使用以下選項：
  - 若要進行縮放，請使用滑鼠滾輪，或使用放大圖示。
  - 若要調整圖形以符合螢幕，請選擇 Zoom to fit (縮放至佈滿) 圖示。
  - 若要移動圖形，請用滑鼠游標拖曳。
- 若要查看階段的詳細資訊，請選擇階段。



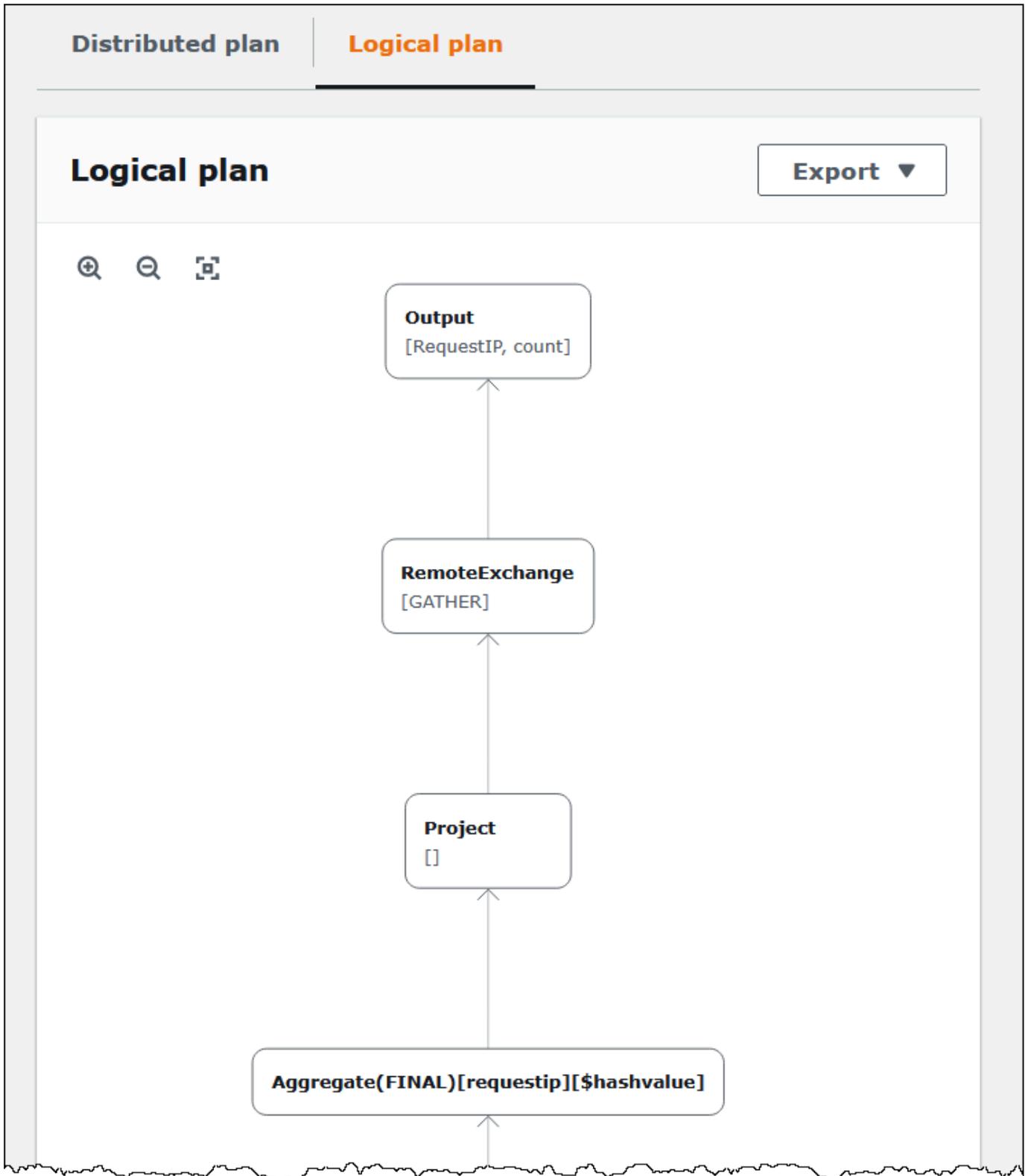
4. 若要以全幅檢視階段詳細資訊，請選擇詳細資訊窗格右上角的展開圖示。
5. 若要查看更多詳細資訊，請展開運算子樹狀結構中的一或多個項目。如需有關分散式計劃片段的資訊，請參閱 [EXPLAIN 陳述式輸出類型](#)。



**⚠ Important**

目前，即使 Athena 確實將這些篩選條件套用至您的查詢，有些分割區篩選條件可能不會顯示在巢狀運算子樹狀結構圖形中。若要驗證此等篩選條件的效果，請在查詢上執行 [EXPLAIN](#) 或 [EXPLAIN ANALYZE](#) 並查看結果。

6. 選擇 Logical plan (邏輯計劃) 索引標籤。此圖形顯示執行查詢的邏輯計劃。如需操作用語的詳細資訊，請參閱 [了解 Athena EXPLAIN 陳述式結果](#)。



7. 若要將計劃匯出為 SVG 或 PNG 圖像，或匯出為 JSON 文字，請選擇 Export (匯出)。

## 其他資源

如需詳細資訊，請參閱下列資源。

[在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)

[了解 Athena EXPLAIN 陳述式結果](#)

[檢視已完成查詢的統計資料和執行詳細資訊](#)

## 使用查詢結果、近期查詢和輸出檔案

Amazon Athena 會自動將執行的每個查詢的查詢結果和中繼資料資訊存放在查詢結果位置，您可以在 Amazon S3 中指定該位置。如有必要，您可以存取此位置中的檔案來使用它們。您也可以直接從 Athena 主控台下載查詢結果檔案。

若是第一次設定 Amazon S3 查詢結果位置，請參閱[使用 Athena 主控台指定查詢結果位置](#)。

每個執行的查詢都會自動儲存輸出檔案。若要使用 Athena 主控台存取和檢視查詢輸出檔案，IAM 主體 (使用者和角色) 需要查詢結果位置的 Amazon S3 [GetObject](#) 動作的權限，以及 Athena [GetQueryResults](#) 動作的許可。查詢結果位置可以加密。如果位置已加密，則使用者必須擁有適當的金鑰許可，才能加密和解密查詢結果位置。

### Important

具有查詢結果位置之 Amazon S3 GetObject 動作許可的 IAM 委託人，能夠從 Amazon S3 擷取查詢結果，即使 Athena GetQueryResults 動作的許可遭拒也一樣。

## 指定查詢結果位置

Athena 使用的查詢結果位置取決於工作群組設定與用戶端設定的組合。用戶端設定是以您執行查詢的方式為基礎。

- 如果您使用 Athena 主控台執行查詢，在導覽列中 Settings (設定) 下輸入的 Query result location (查詢結果位置) 會決定用戶端設定。
- 如果您使用 Athena API 執行查詢，[StartQueryExecution](#) 動作的 OutputLocation 參數會決定用戶端設定。
- 如果您使用 ODBC 或 JDBC 驅動程式執行查詢，連線 URL 中指定的 S3OutputLocation 屬性會決定用戶端設定。

**⚠ Important**

當您使用 API 或使用 ODBC 或 JDBC 驅動程式執行查詢時，主控台設定並不適用。

每個工作群組組態都有可以啟用的 [Override client-side settings \(覆寫用戶端設定\)](#) 選項。啟用此選項時，若與該工作群組相關聯的 IAM 委託人執行查詢，工作群組設定會優先於適用的用戶端設定。

### 使用 Athena 主控台指定查詢結果位置

您必須指定 Amazon S3 中的查詢結果儲存貯體位置，然後才能執行查詢，否則您必須使用已指定儲存貯體，且其組態可覆寫用戶端設定的工作群組。

### 若要使用 Athena 主控台指定用戶端設定查詢結果位置

1. [切換](#)至您要為其指定查詢結果位置的工作群組。預設工作群組的名稱為 primary (主要)。
2. 從導覽列中，選擇 Settings (設定)。
3. 請在導覽列中選擇 Manage (管理)。
4. 對於 Manage settings (管理設定)，請執行下列其中一項動作：
  - 在 Location of query result (查詢結果的位置) 方塊中，輸入您在 Amazon S3 中為查詢結果建立的儲存貯體路徑。在路徑中加上 s3:// 作為字首。
  - 選擇 Browse S3 (瀏覽 S3)，選擇您在目前區域建立的 Amazon S3 儲存貯體，然後選擇 Choose (選擇)。

**Note**

如果您使用為工作群組中的所有使用者指定查詢結果位置的工作群組，則變更查詢結果位置的選項將無法使用。

5. (選用) 選擇 View lifecycle configuration (檢視生命週期組態)，以檢視和設定查詢結果儲存貯體上的 [Amazon S3 生命週期規則](#)。您建立的 Amazon S3 生命週期規則可以是到期規則或轉換規則。到期規則會在一段時間之後自動刪除查詢結果。轉換規則會將其移至另一個 Amazon S3 儲存層。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [設定儲存貯體的生命週期組態](#)。
6. (選擇性) 針對「預期值區擁有者」AWS 帳戶，輸入您希望成為輸出位置值區擁有者的 ID。這是一項附加的安全措施。如果儲存貯體擁有者的帳戶 ID 與您在此處指定的 ID 不相符，則嘗試輸出

到儲存貯體的動作會失敗。如需詳細資訊，請參閱 Amazon S3 使用者指南中的[使用儲存貯體擁有者條件驗證儲存貯體擁有權](#)。

#### Note

預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Amazon S3 輸出位置。它不適用於其他 Amazon S3 位置，例如外部 Amazon S3 儲存貯體、CTAS 和 INSERT INTO 目的地資料表位置、UNLOAD 陳述式輸出位置等資料來源位置，也不適用於溢出聯合查詢儲存貯體的操作，或針對另一個帳戶中的資料表執行的 SELECT 查詢。

7. (選用) 如果要將存放在 Simple Storage Service (Amazon S3) 中的查詢結果加密，選擇 Encrypt query results (將查詢結果加密)。如需在 Athena 中加密的詳細資訊，請參閱[靜態加密](#)。
8. (選用) 對於查詢結果儲存貯體，選擇 Assign bucket owner full control over query results (指派儲存貯體擁有者對查詢結果的完整控制)，以便在[ACL 已啟用](#)時，授予儲存貯體擁有者對查詢結果的完整存取。例如，如果您的查詢結果位置係由另一個帳戶所擁有，則可以授予另一個帳戶對查詢結果的所有權和完整控制。如需詳細資訊，請參閱《Simple Storage Service (Amazon S3) 使用者指南》中的[控制物件的所有權並停用儲存貯體的 ACL](#)。
9. 選擇儲存。

### 先前建立的預設位置

先前，如果您在 Athena 中執行查詢而未指定 Query result location (查詢結果位置) 的值，且查詢結果位置設定未被工作群組覆寫，則 Athena 會為您建立預設位置。預設位置是 `aws-athena-query-results-MyAcctID-MyRegion`，其中 `MyAcctID` 是執行查詢之 IAM 主體的 Amazon Web Services 帳戶 ID，而且 `MyRegion` 是執行查詢的區域 (例如 `us-west-1`。)

現在，您必須指定查詢結果位置，或使用可覆寫查詢結果位置設定的工作群組，才能在帳戶之前並未使用過 Athena 的區域中執行 Athena 查詢。雖然 Athena 不再為您建立預設查詢結果位置，但先前建立的預設 `aws-athena-query-results-MyAcctID-MyRegion` 位置仍然有效，您可以繼續使用。

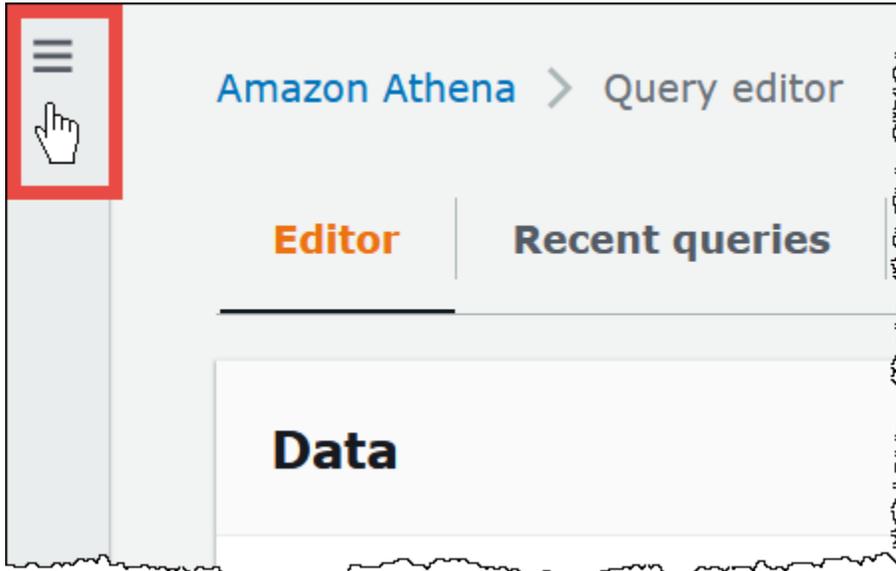
### 使用工作群組指定查詢結果位置

您可以使用 AWS Management Console、AWS CLI 或 Athena API，在工作群組組態中指定查詢結果位置。

使用時 AWS CLI，請在執行 `aws athena create-work-group` 或 `aws athena update-work-group` 命令時，使用 `--configuration` 選項的 `OutputLocation` 參數指定查詢結果位置。

## 若要使用 Athena 主控台指定工作群組的查詢結果位置

1. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



2. 在導覽窗格中，選擇 Workgroups (工作群組)。
3. 在工作群組清單中，選擇您要編輯的工作群組之連結。
4. 選擇編輯。
5. 對於 Query result location and encryption (查詢結果位置和加密)，執行下列其中一項：
  - 在 Location of query result (查詢結果的位置) 文字方塊中，輸入您要在 Amazon S3 中存放查詢結果的儲存貯體路徑。在路徑中加上 s3:// 作為字首。
  - 選擇 Browse S3 (瀏覽 S3)，選擇您在目前區域中想要使用的 Amazon S3 儲存貯體，然後選擇 Choose (選擇)。
6. (選擇性) 針對「預期值區擁有者」AWS 帳戶，輸入您希望成為輸出位置值區擁有者的 ID。這是一項附加的安全措施。如果儲存貯體擁有者的帳戶 ID 與您在此處指定的 ID 不相符，則嘗試輸出到儲存貯體的動作會失敗。如需詳細資訊，請參閱 Amazon S3 使用者指南中的[使用儲存貯體擁有者條件驗證儲存貯體擁有權](#)。

### Note

預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Amazon S3 輸出位置。它不適用於其他 Amazon S3 位置，例如外部 Amazon S3 儲存貯體、CTAS 和 INSERT INTO 目的地資料表位置、UNLOAD 陳述式輸出位置等資料來源位置，也不適用於溢出聯合查詢儲存貯體的操作，或針對另一個帳戶中的資料表執行的 SELECT 查詢。

7. (選用) 如果要將存放在 Simple Storage Service (Amazon S3) 中的查詢結果加密，選擇 Encrypt query results (將查詢結果加密)。如需在 Athena 中加密的詳細資訊，請參閱 [靜態加密](#)。
8. (選用) 對於查詢結果儲存貯體，選擇 Assign bucket owner full control over query results (指派儲存貯體擁有者對查詢結果的完整控制)，以便在 [ACL 已啟用](#) 時，授予儲存貯體擁有者對查詢結果的完整存取。例如，如果您的查詢結果位置係由另一個帳戶所擁有，則可以授予另一個帳戶對查詢結果的所有權和完整控制。

如果儲存貯體的 S3 物件所有權設定為 Bucket owner preferred (首選的儲存貯體擁有者) 時，則儲存貯體擁有者亦擁有從此工作群組寫入的所有查詢結果物件。例如，如果外部帳戶的工作群組啟用此選項，並將其查詢結果位置設定為您帳戶的 Simple Storage Service (Amazon S3) 儲存貯體，且該儲存貯體的 S3 物件所有權設定為 Bucket owner preferred (首選的儲存貯體擁有者) 時，則您擁有外部工作群組的查詢結果並具有完整的控制存取。

當查詢結果儲存貯體的 S3 物件所有權設定為 Bucket owner enforced (強制的儲存貯體擁有者) 時，則選取此選項不會有任何作用。如需詳細資訊，請參閱《Simple Storage Service (Amazon S3) 使用者指南》中的 [控制物件的所有權並停用儲存貯體的 ACL](#)。

9. 如果要讓工作群組的所有使用者使用您指定的查詢結果位置，請往下捲動到 Settings (設定) 部分，然後選擇 Override client-side settings (覆寫用戶端設定)。
10. 選擇儲存變更。

## 使用 Athena 主控台下載查詢結果檔案

您可以在執行查詢之後，立即從查詢窗格下載查詢結果 CSV 檔案檔案。您也可以從 Recent queries (近期查詢) 索引標籤中，下載近期查詢的查詢結果。

### Note

Athena 查詢結果檔案是資料檔案，內含個別使用者可設定的資訊。讀取與分析此資料的某些程式很可能會將某些資料解譯為命令 (CSV injection)。因此，將查詢結果 CSV 資料匯入至試算表程式時，該程式可能會提醒您關於安全方面的考量。為確保系統的安全，您應該始終選擇停用下載的查詢結果中的連結或巨集。

## 執行查詢並下載查詢結果

1. 在查詢編輯器中輸入查詢，然後選擇 Run (執行)。

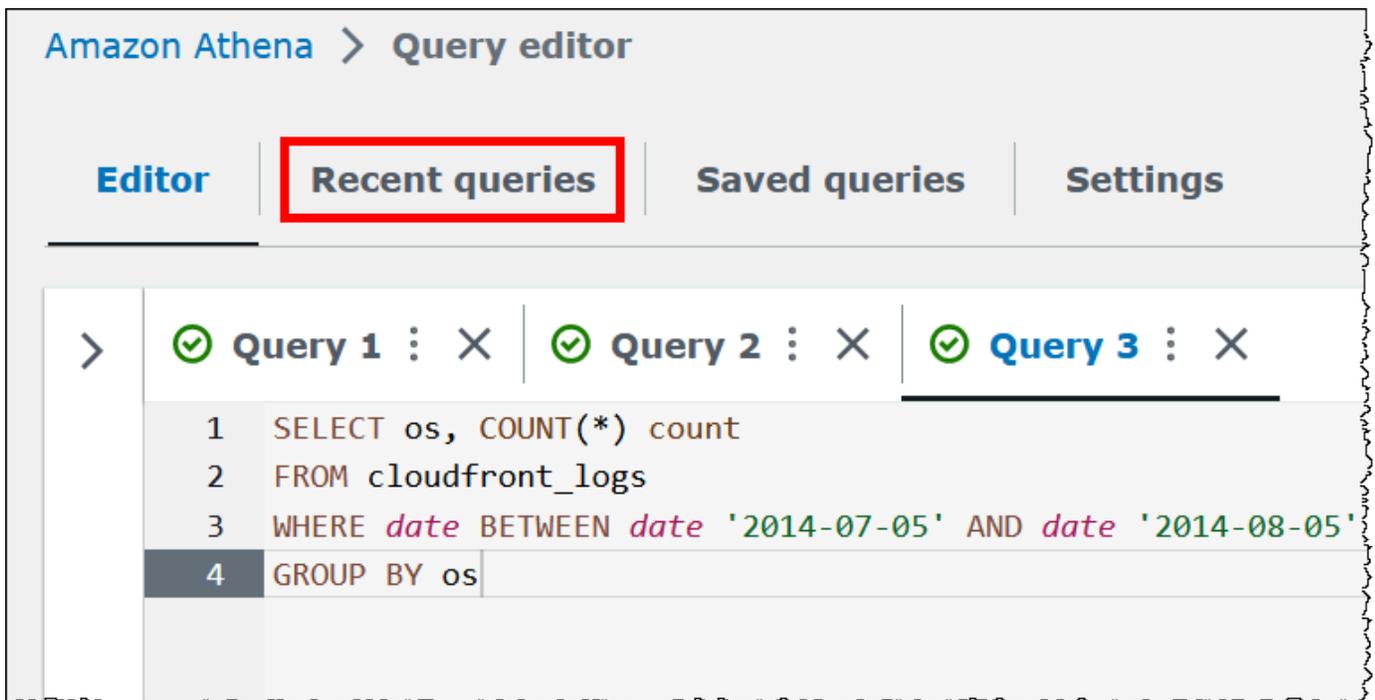
當查詢完成執行時，Results (結果) 窗格會顯示查詢結果。

- 若要下載查詢結果的 CSV 檔案，請選擇查詢結果窗格上方的 Download results (下載結果)。根據您的瀏覽器和瀏覽器組態，您可能需要確認下載。



下載先前查詢的查詢結果檔案

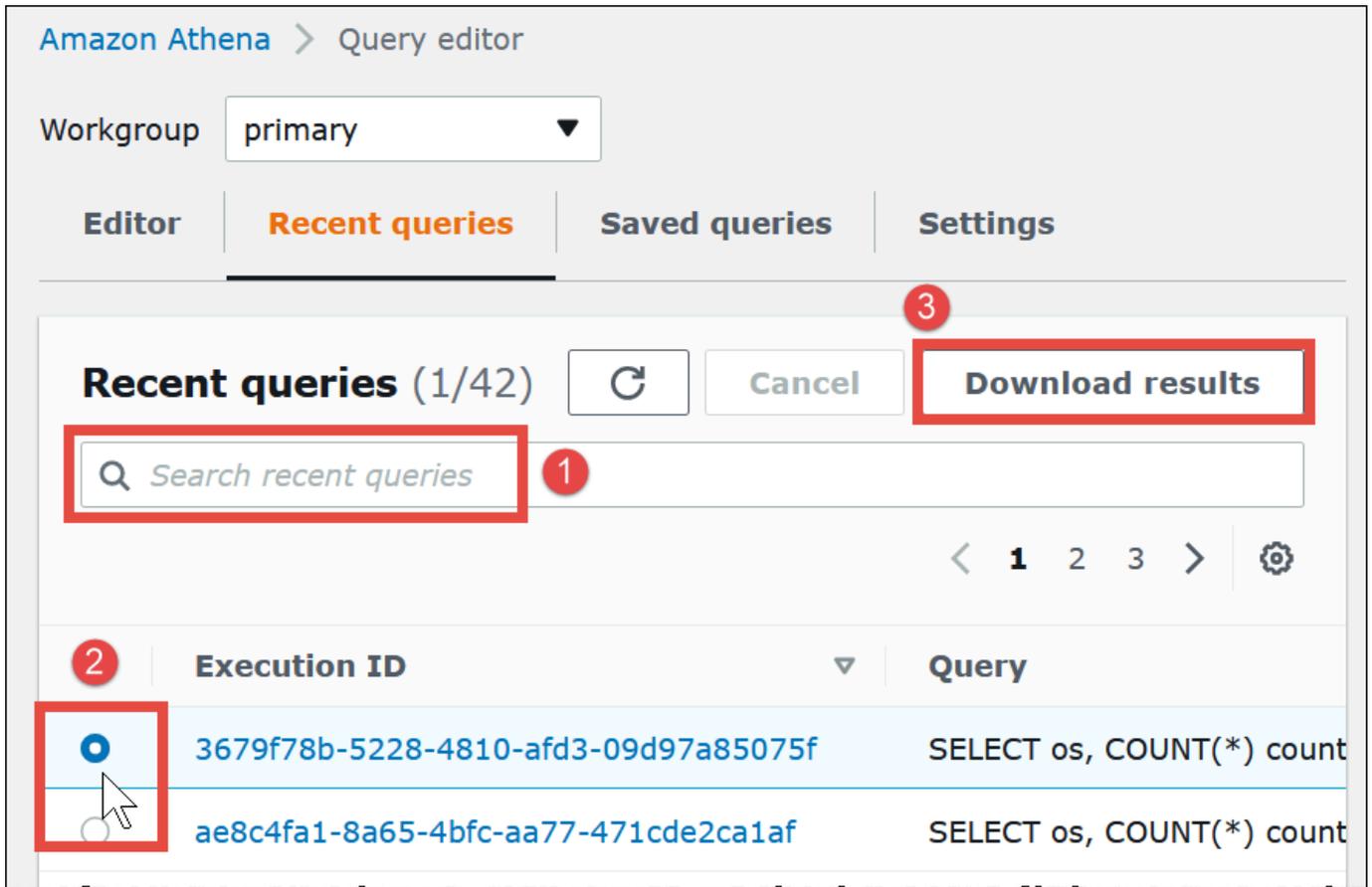
- 選擇 Recent queries (近期查詢)。



- 使用搜尋方塊尋找查詢、選擇查詢，然後選擇 Download results (下載結果)。

#### Note

您無法使用 Download results (下載結果) 選項來擷取已手動刪除的查詢結果，或擷取已由 Amazon S3 [生命週期規則](#) 刪除或移至其他位置的查詢結果。

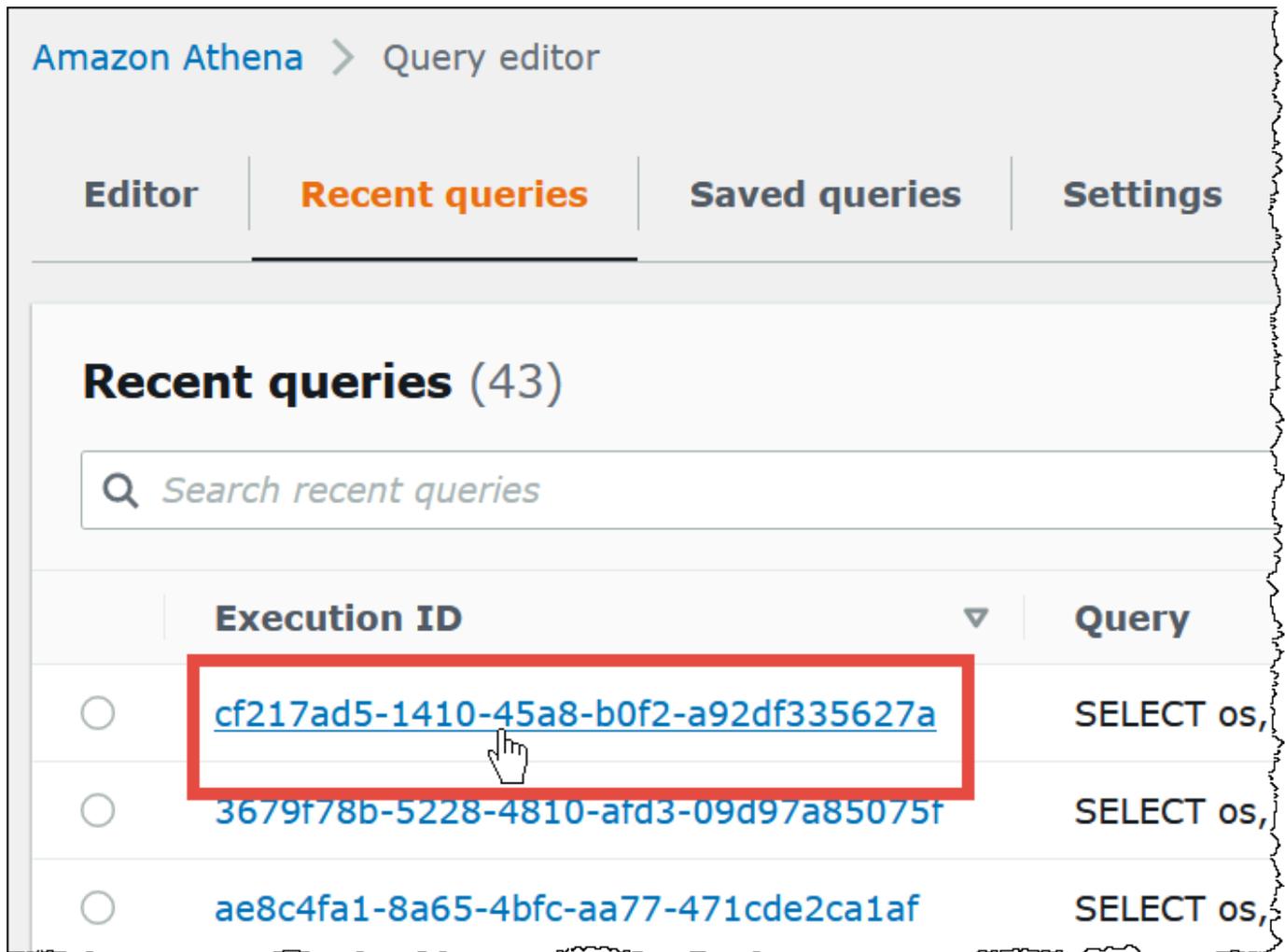


## 檢視近期查詢

使用 Athena 主控台可查看哪些查詢成功或失敗，並檢視失敗查詢的錯誤詳細資訊。Athena 會將查詢歷史記錄保留 45 天。

在 Athena 主控台中檢視近期查詢。

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 選擇 Recent queries (近期查詢)。Recent queries (近期查詢) 索引標籤顯示每個已執行查詢的相關資訊。
3. 若要在查詢編輯器中開啟查詢陳述式，請選擇查詢的執行 ID。



Amazon Athena > Query editor

Editor | **Recent queries** | Saved queries | Settings

### Recent queries (43)

🔍 Search recent queries

	Execution ID	Query
<input type="radio"/>	<a href="#">cf217ad5-1410-45a8-b0f2-a92df335627a</a>	SELECT os,
<input type="radio"/>	<a href="#">3679f78b-5228-4810-afd3-09d97a85075f</a>	SELECT os,
<input type="radio"/>	<a href="#">ae8c4fa1-8a65-4bfc-aa77-471cde2ca1af</a>	SELECT os,

- 若要查看失敗查詢的詳細資訊，請選擇查詢的 Failed (失敗) 連結。

Start time	Status	Run time
	Failed	0.229 sec
	Failed	0.203 sec
	Completed	3.484 sec
	Completed	3.143 sec
	Completed	3.517 sec
	Completed	3.398 sec
	Completed	3.412 sec

**Error** ✕

Query ID  
6a242b5c-226b-4a51-aec6-e9667c5bcd66 📄

Error details  
SYNTAX\_ERROR: line 1:18: Table  
awsdatacatalog.mydatabase.mytable does not exist

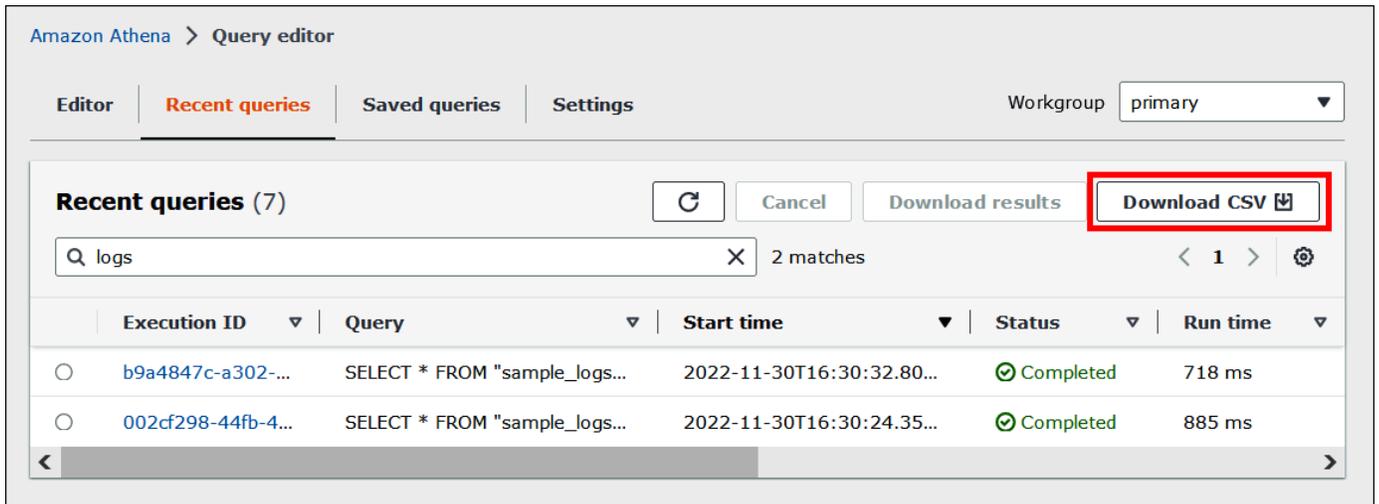
This query ran against the "mydatabase" database, unless qualified by the query. Please post the error message on our [forum](#) or contact [customer support](#) with query id.

## 將多個近期查詢下載到 CSV 檔案

您可以使用 Athena 主控台的 Recent queries (近期查詢) 標籤，將一或多個近期查詢匯出至 CSV 檔案，以便以表格格式檢視這些查詢。下載的檔案不包含查詢結果，而是 SQL 查詢字串本身以及有關查詢的其他資訊。匯出的欄位包括執行 ID、查詢字串內容、查詢開始時間、狀態、執行時間、掃描的資料量、使用的查詢引擎版本以及加密方法。您可以匯出最多 500 個近期查詢，或使用在搜尋方塊中輸入的條件，匯出最多 500 個篩選的查詢。

若要將一或多個近期查詢匯出至 CSV 檔案

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 選擇 Recent queries (近期查詢)。
3. (選用) 使用搜尋方塊，篩選您要下載的近期查詢。
4. 選擇下載 CSV。



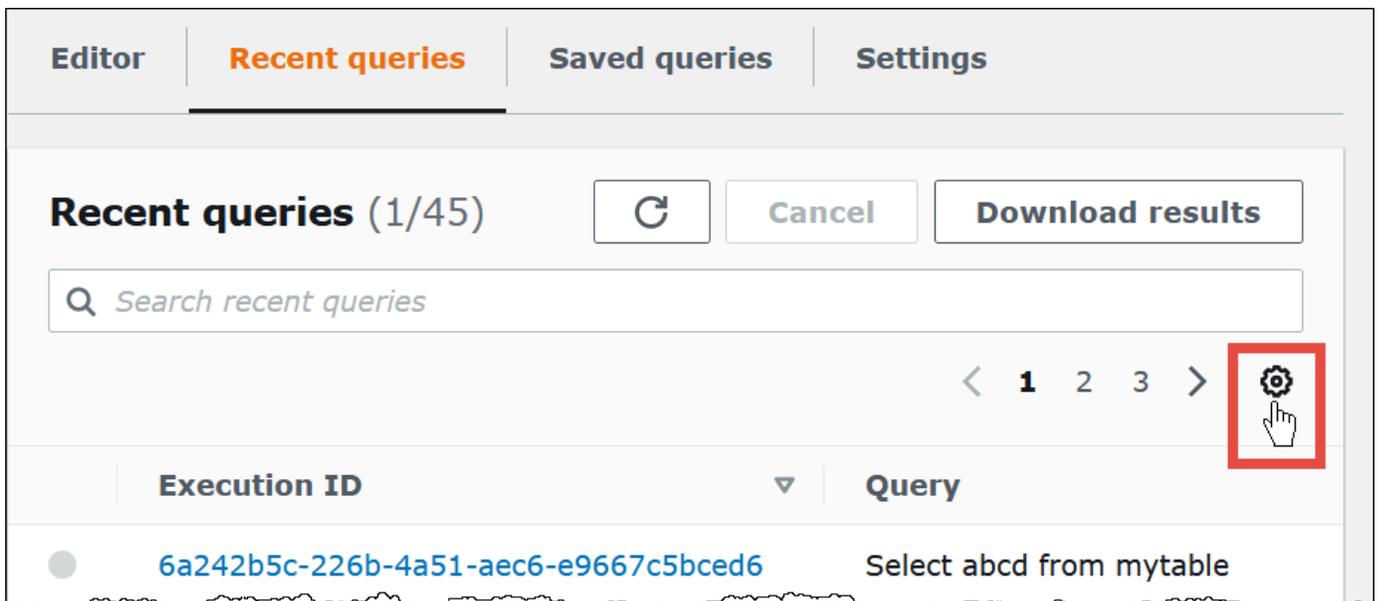
5. 出現檔案儲存提示時，選擇 Save (儲存)。預設檔案名稱 Recent Queries 後面接著一個時間戳記 (例如，Recent Queries 2022-12-05T16 04 27.352-08 00.csv)

### 設定近期查詢顯示選項

您可以設定 Recent queries (近期查詢) 標籤的選項 (例如要顯示的資料欄和文字換行)。

若要設定 Recent queries (近期查詢) 標籤的選項

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 選擇 Recent queries (近期查詢)。
3. 選擇選項按鈕 (齒輪圖示)。



4. 在 Preferences (偏好設定) 對話方塊中，選擇每頁的列數、換行反應和要顯示的資料欄。

## Preferences ✕

Select rows per page

10 queries

20 queries

Wrap lines  
Wraps long lines to show all the text

Select visible content

Properties

Execution ID	<input checked="" type="checkbox"/>
Query	<input checked="" type="checkbox"/>
Start time	<input checked="" type="checkbox"/>
Run time	<input checked="" type="checkbox"/>
Status	<input checked="" type="checkbox"/>
Data scanned	<input checked="" type="checkbox"/>
Query engine version used	<input checked="" type="checkbox"/>
Encryption	<input checked="" type="checkbox"/>

## 5. 選擇確認。

### 保留超過 45 天的查詢歷史記錄

若要將查詢歷史記錄保留 45 天以上，您可以擷取查詢歷史記錄，並將其儲存至資料存放區，例如 Amazon S3。若要自動執行此程序，您可以使用 Athena 和 Amazon S3 API 動作以及 CLI 命令。下列程序概述這些步驟。

### 使用程式擷取和儲存查詢歷史記錄的步驟

1. 使用 Athena [ListQueryExecutions](#) API 動作或 [list-query-executions](#) CLI 命令擷取查詢識別碼。
2. 使用 Athena [GetQueryExecution](#) API 動作或 [get-query-execution](#) CLI 命令，根據查詢的 ID 擷取有關每個查詢的資訊。
3. 使用 Amazon S3 [PutObject](#) API 動作或置入 [物件](#) CLI 命令，將資訊儲存在 Amazon S3 中。

### 在 Amazon S3 中查找查詢輸出檔案

除非查詢是發生在其組態會覆寫用戶端設定的工作群組中，否則查詢輸出檔案會以下列路徑模式存放在 Amazon S3 的子資料夾中。當工作群組組態取代用戶端設定時，查詢會使用工作群組指定的結果路徑。

```
QueryResultsLocationInS3/[QueryName | Unsaved/yyyy/mm/dd/]
```

- *QueryResultsLocationInS3* 是工作群組設定或用戶端設定所指定的查詢結果位置。如需詳細資訊，請參閱本文件稍後的 [the section called “指定查詢結果位置”](#)。
- 只有從主控台執行且其結果路徑尚未被工作群組組態覆寫的查詢，會建立下列子資料夾。從 AWS CLI 或使用 Athena API 執行的查詢會直接儲存到 *QueryResultsLocationInS3*。
  - *QueryName* 是儲存結果的查詢名稱。如果查詢已執行但未儲存，則會使用 *Unsaved*。
  - *yyyy/mm/dd* 是查詢執行的日期。

與 CREATE TABLE AS SELECT 查詢相關聯的檔案會存放在上述模式的 *tables* 子資料夾中。

### 識別查詢輸出檔案

系統會根據查詢的名稱、查詢 ID 和查詢執行的日期，將檔案儲存至 Amazon S3 中的查詢結果位置。每個查詢的檔案都是使用 *QueryID* 命名，此 QueryID 是每個查詢執行時 Athena 指派給查詢的唯一識別符。

系統會儲存下列檔案類型：

檔案類型	檔案命名模式	描述
查詢結果檔案	<i>QueryID</i> .csv <i>QueryID</i> .txt	<p>DML 查詢結果檔案會以逗號分隔值 (CSV) 格式儲存。</p> <p>DDL 查詢結果會儲存為純文字檔案。</p> <p>使用主控台時，您可以從主控台的 Results (結果) 窗格下載結果檔案，或從查詢 History (歷史記錄) 進行下載。如需詳細資訊，請參閱<a href="#">使用 Athena 主控台下載查詢結果檔案</a>。</p>
查詢中繼資料檔案	<i>QueryID</i> .csv.metadata <i>QueryID</i> .txt.metadata	<p>DML 和 DDL 查詢中繼資料檔案以二進位格式儲存，而且人類不可讀取。副檔名會對應到相關的查詢結果檔案。Athena 在使用 GetQueryResults 動作讀取查詢結果時，會使用中繼資料。雖然可以刪除這些檔案，但我們不建議這麼做，因為有關查詢的重要資訊會遺失。</p>
資料資訊清單檔案	<i>QueryID</i> -manifest.csv	<p>產生資料資訊清單檔案，以追蹤在執行 <a href="#">INSERT INTO</a> 查詢時，Athena 在 Amazon S3 資料來源位置中建立的檔案。如果查詢失敗，資訊清單也會追蹤查詢要寫入的檔案。資訊清單可用於識別失敗查詢所產生的遺棄檔案。</p>

## 使用 AWS CLI 來識別查詢輸出位置和檔案

若要使用 AWS CLI 來識別查詢輸出位置和結果檔案，請執行 `aws athena get-query-execution` 命令，如下列範例所示。將 `abc1234d-5efg-67hi-jklm-89n0op12qr34` 取代為查詢 ID。

```
aws athena get-query-execution --query-execution-id abc1234d-5efg-67hi-jklm-89n0op12qr34
```

此命令會傳回如下輸出：如需每個輸出參數的描述，請參閱《AWS CLI 指令參考》[get-query-execution](#) 中的 `<`。

```
{
  "QueryExecution": {
    "Status": {
      "SubmissionDateTime": 1565649050.175,
      "State": "SUCCEEDED",
      "CompletionDateTime": 1565649056.6229999
    },
    "Statistics": {
      "DataScannedInBytes": 5944497,
      "DataManifestLocation": "s3://DOC-EXAMPLE-BUCKET/athena-query-results-123456789012-us-west-1/MyInsertQuery/2019/08/12/abc1234d-5efg-67hi-jklm-89n0op12qr34-manifest.csv",
      "EngineExecutionTimeInMillis": 5209
    },
    "ResultConfiguration": {
      "EncryptionConfiguration": {
        "EncryptionOption": "SSE_S3"
      },
      "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/athena-query-results-123456789012-us-west-1/MyInsertQuery/2019/08/12/abc1234d-5efg-67hi-jklm-89n0op12qr34"
    },
    "QueryExecutionId": "abc1234d-5efg-67hi-jklm-89n0op12qr34",
    "QueryExecutionContext": {},
    "Query": "INSERT INTO mydb.elb_log_backup SELECT * FROM mydb.elb_logs LIMIT 100",
    "StatementType": "DML",
    "WorkGroup": "primary"
  }
}
```

## 重複使用查詢結果

當您在 Athena 中重新執行查詢時，您可以選擇性地選擇重複使用最近儲存的查詢結果。此選項可以提高效能並降低已掃描的位元組數的成本。例如，如果您知道結果在指定時間範圍內不會變更，則重複使用查詢結果就很有用。您可以指定重複使用查詢結果的最長期限。只要 Athena 的結果不超過您指定的期限，就會使用儲存的結果。如需詳細資訊，請參閱 AWS 大數據部落格中的[使用 Amazon Athena 降低成本和提升查詢效能](#)。

### Note

查詢結果重複使用功能需要 Athena 引擎第 3 版。如需有關變更引擎版本的資訊，請參閱[變更 Athena 引擎版本](#)。

## 主要功能

- 重複使用查詢結果是每次查詢的選擇加入功能。您可以針對每次查詢啟用查詢結果重複使用。
- 您可以以分鐘、小時或天數來指定重複使用查詢結果的最長期限。無論使用的時間單位為何，可指定的最長期限相等於 7 天。預設值為 60 分鐘。
- 當您啟用查詢的結果重複使用時，Athena 會在相同的工作群組中尋找先前執行的查詢。如果 Athena 找到對應的已儲存查詢結果，就不會重新執行查詢，而是指向先前的結果位置或從中擷取資料。
- 對於任何啟用結果重複使用選項的查詢，Athena 只會在下列所有條件皆為 true 時，才會重複使用上次儲存至工作群組資料夾的查詢結果：
  - 查詢字串是完全相符的字串。
  - 資料庫與目錄名稱相符。
  - 先前的結果不會超過指定的最長期限，或者如果尚未指定最長期限，則不超過 60 分鐘。
  - Athena 只會重複使用與目前執行具有完全相同的[結果組態](#)的執行。
  - 您可以存取查詢中參考的所有資料表。
  - 您可以存取儲存先前結果的 S3 檔案位置。

如果不符合上述任何條件，則 Athena 會在不使用快取結果的情況下執行查詢。

## 考量與限制

使用查詢結果重複使用功能時，請謹記以下幾點：

- Athena 只會在相同的工作群組中重複使用查詢結果。
- 重複使用查詢結果功能遵循工作群組組態。如果您覆寫查詢的結果組態，則會停用該功能。
- 支援使用 AWS Glue 註冊的阿帕奇蜂巢、阿帕奇胡迪、阿帕奇冰山和 Linux 基金會三角洲湖表。不支援外部 Hive 中繼存放區。
- 不支援參考聯合目錄或外部 Hive 中繼存放區的查詢。
- Lake Formation 受管理的資料表不支援查詢結果重複使用。
- 當資料表來源的 Amazon S3 位置註冊為 Lake Formation 中的資料位置時，不支援查詢結果重複使用。
- 不支援具有資料列和資料欄許可的資料表。
- 不支援具有精細的存取控制 (例如資料欄或資料列篩選) 的資料表。
- 任何參考不支援資料表的查詢都不符合查詢結果重複使用的資格。
- Athena 要求您擁有 Amazon S3 讀取許可，如此才能重複使用之前產生的輸出檔案。
- 重複使用查詢結果功能會假設先前結果的內容尚未修改。Athena 在使用之前不會檢查先前結果的完整性。
- 如果先前執行的查詢結果已刪除或移至 Amazon S3 中的其他位置，後續執行相同查詢將不會重複使用查詢結果。
- 可能會傳回過時結果。在達到您指定的重複使用最長期限之前，Athena 不會檢查來源資料是否有變更。
- 如果有多個結果可供重複使用，則 Athena 會使用最新的結果。
- 使用類似 `rand()` 或 `shuffle()` 的非確定性運算子或函數的查詢，並未使用快取結果。例如，不含 `ORDER BY` 的 `LIMIT` 是非確定性的，且並非快取，但含有 `ORDER BY` 的 `LIMIT` 是確定性的其已快取。
- Athena 主控台、Athena API 和 JDBC 驅動程式支援查詢結果重複使用。目前，查詢結果重複使用的 ODBC 驅動程式支援僅適用於 Windows。
- 若要搭配 JDBC 使用查詢結果重複使用功能，所需的最低驅動程式版本為 2.0.34.1000。對於 ODBC，所需的最低驅動程式版本為 1.1.19.1002。如需驅動程序下載資訊，請參閱 [使用 ODBC 和 JDBC 驅動程式連接到 Amazon Athena](#)。
- 使用多個資料目錄的查詢不支援查詢結果重複使用。
- 查詢包含超過 20 個資料表的查詢不支援重複使用查詢結果。

## 在 Athena 主控台中重複使用查詢結果

若要使用此功能，請在 Athena 查詢編輯器中啟用 Reuse query results (重複使用查詢結果)。

Query 1

```
1 SELECT * FROM mytable
```

SQL Ln 1, Col 22

Run Explain Cancel Save Clear Create

Reuse query results  
up to 60 minutes ago

Query results | Query stats

Results (0) Copy Download results

Search rows < 1 >

No results  
Run a query to view results

### 若要設定重複使用查詢結果功能

1. 在 Athena 查詢編輯器的 Reuse query results (重複使用查詢結果) 選項下，選擇 up to 60 minutes ago (最多 60 分鐘前) 旁邊的編輯圖示。
2. 在 Edit reuse time (編輯重複使用時間) 對話方塊中，從右側的方塊中選擇時間單位 (分鐘、小時或天)。
3. 在左側的方塊中，輸入或選擇您要指定的時間單位數目。無論選擇的時間單位為何，您可以輸入的最長時間相當於七天。

## Edit reuse time ✕

**Maximum age of reused query results**  
Athena will return the most recent results available within this time frame.

↕  ▼

Minimum: 1 minute, Maximum: 10080 minutes.

**Cancel** **Confirm**

下列範例將重複使用的最長期限指定為兩天。

## Edit reuse time ✕

**Maximum age of reused query results**  
Athena will return the most recent results available within this time frame.

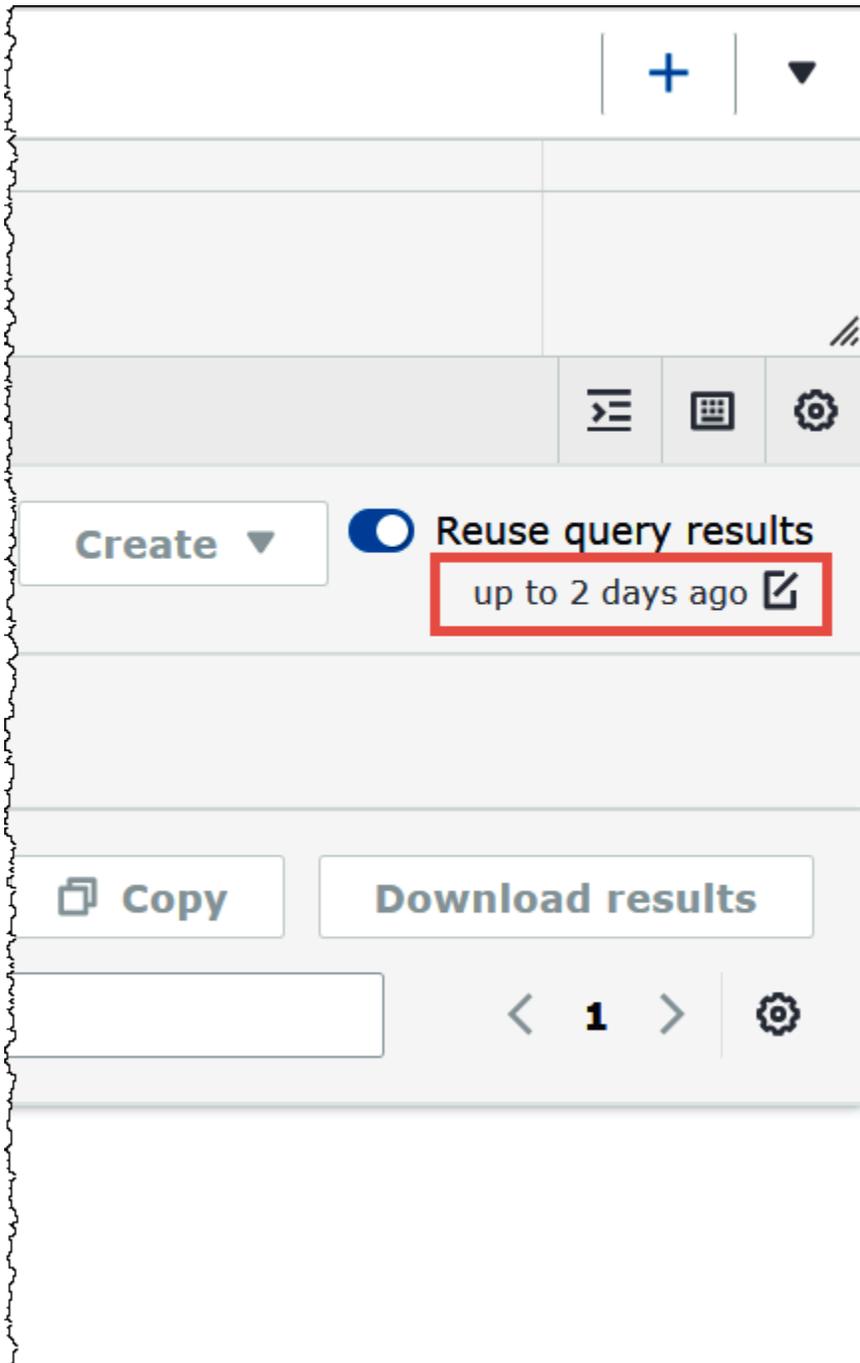
↕  ▼

Minimum: 1 day, Maximum: 7 days.

**Cancel** **Confirm**

4. 選擇確認。

橫幅會確認您的組態變更，而 Reuse query results (重新使用查詢結果) 選項會顯示您的新設定。



## 檢視已完成查詢的統計資料和執行詳細資訊

執行查詢之後，您可以取得所處理之輸入和輸出資料的統計資料、查看查詢每個階段所花費時間的圖形呈現，並以互動方式探索執行詳細資訊。

## 若要檢視已完成查詢的查詢統計資料

1. 在 Athena 查詢編輯器中執行查詢之後，請選擇 Query stats (查詢統計資料) 索引標籤。

1 `SELECT * FROM "sampledb"."elb_logs" limit 10;`

SQL Ln 1, Col 46

[Run again](#) [Explain](#) [Cancel](#) [Save](#) [Clear](#) [Create](#)

Query results **Query stats**

**Data processed**

Input rows	Input bytes	Output rows	Output bytes
26.43 K	9.00 MB	10	3.41 KB

**Total runtime - 1.4 seconds** [Execution details](#)

0 0.2 0.4 0.6 0.8 1 1.2 1.4 seconds

■ Queuing 17% ■ Planning 19% ■ Execution 58% ■ Service processing 6%

Query stats (查詢統計資料) 索引標籤提供以下資訊：

- Data processed (處理的資料) – 顯示已處理的輸入資料列數和位元組數，以及輸出的資料列數和位元組數。
- The Total runtime (總執行時間) – 顯示執行查詢所花費的總時間，以及花費在佇列、規劃、執行和服務處理上的時間的圖形呈現。

**Note**

當查詢具有 Lake Formation 中定義的資料列層級篩選條件時，不會顯示階段層級輸入和輸出資料列計數和資料大小資訊。

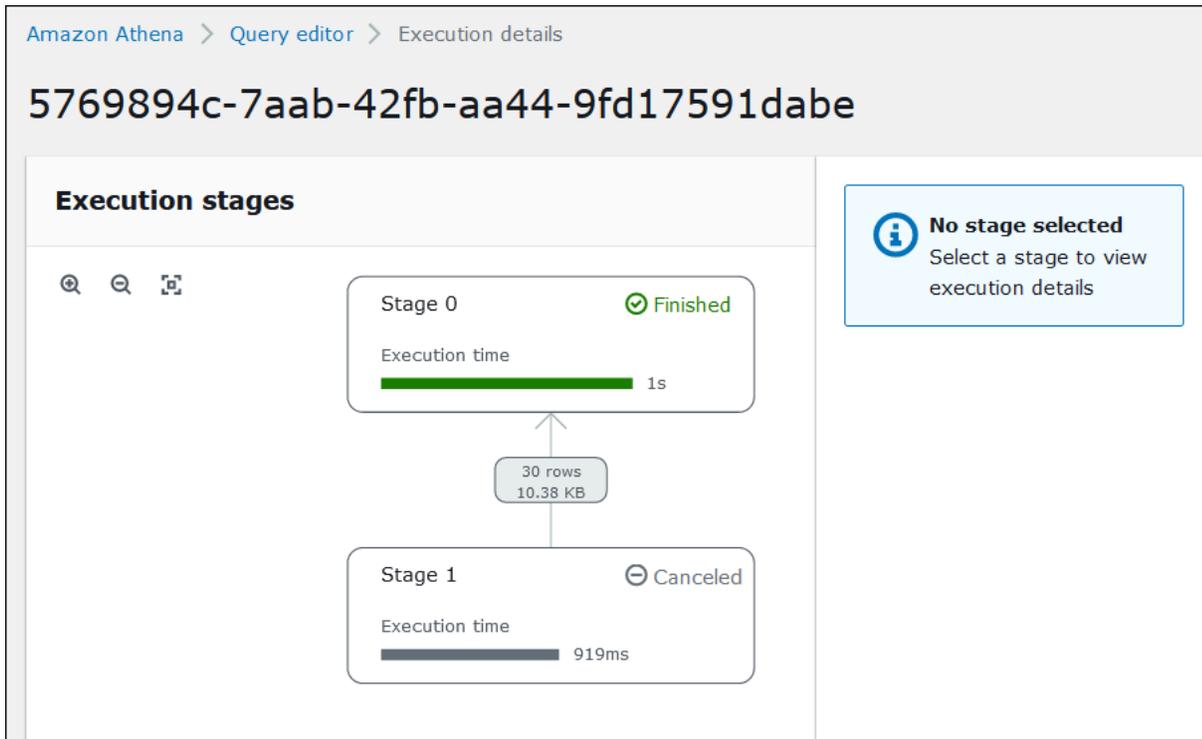
2. 若要以互動方式探索查詢執行方式的資訊，請選擇 Execution details (執行詳細資訊)。



Execution details (執行詳細資訊) 頁面顯示查詢的執行 ID，以及查詢中從零開始之階段的圖形。階段從下到上按開始到結束的順序排列。每個階段的標籤都會顯示階段執行所花費的時間量。

**Note**

查詢的執行階段和執行階段時間總計通常會有很大差異。例如，查詢的執行時間總計 (以分鐘為單位) 可以顯示階段的執行時間 (以小時為單位)。因為階段是跨許多工作 parallel 執行的運算邏輯單元，因此階段的執行時間就是其所有工作的彙總執行時間。儘管有這種差異，階段執行時間可以作為查詢中運算最密集的階段的相對指標很有用。



若要導覽圖形，請使用以下選項：

- 若要進行縮放，請使用滑鼠滾輪，或使用放大圖示。
  - 若要調整圖形以符合螢幕，請選擇 Zoom to fit (縮放至佈滿) 圖示。
  - 若要移動圖形，請用滑鼠游標拖曳。
3. 若要查看階段的更多詳細資訊，請選擇階段。右側的階段詳細資訊窗格顯示輸入和輸出的資料列數和位元組數，以及運算子樹狀結構。

**Execution stages**

Stage 0 ✔ Finished  
Execution time: 1s

30 rows  
10.38 KB

Stage 1 ⊖ Canceled  
Execution time: 919ms

**Stage 0** ⊞

Status: ✔ Finished

Input rows	Input bytes
10	3.31 KB
Output rows	Output bytes
10	3.31 KB

Execution time: 1.3 sec

Operators: [Expand all](#)

**Output**  
[request\_timestamp, elb\_name, backend\_port, request\_processing\_time, client\_response\_time, elb\_response\_time, received\_bytes, sent\_bytes, received\_bytes\_sent\_ratio, ssl\_cipher, ssl\_protocol]

4. 若要以全幅檢視階段詳細資訊，請選擇詳細資訊窗格右上角的展開圖示。
5. 若要取得有關階段各部分的資訊，請展開運算子樹狀結構中的一或多個項目。

### Stage 0

Status  
✔ Finished

Input rows	Input bytes
10	3.31 KB
Output rows	Output bytes
10	3.31 KB

Execution time  
1.3 sec

Operators  
[Collapse all](#)

```
graph BT; RemoteSource[RemoteSource [1]] --> LocalExchange[LocalExchange [SINGLE] ()]; LocalExchange --> Limit[Limit [10]]; Limit --> Output[Output [request_timestamp, elb_name, request_ip, request_port, backend_ip, backend_port, request_processing_time, backend_processing_time, client_response_time, elb_response_code, backend_response_code, received_bytes, sent_bytes, request_verb, url, protocol, user_agent, ssl_cipher, ssl_protocol]];
```

如需執行詳細資訊的更多資訊，請參閱 [了解 Athena EXPLAIN 陳述式結果](#)。

## 其他資源

如需詳細資訊，請參閱下列資源。

[檢視 SQL 查詢的執行計劃](#)

[在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)

## 使用檢視

Amazon Athena 中的檢視是邏輯資料表，而非實體資料表。每次在查詢中參考檢視時，定義該檢視的查詢就會執行。

您可以從 SELECT 查詢建立檢視，然後在未來的查詢中參考此檢視。如需詳細資訊，請參閱[CREATE VIEW](#)。

## 主題

- [何時使用檢視？](#)
- [Athena 中對檢視支援的動作](#)
- [檢視的考量](#)
- [檢視限制](#)
- [在主控台使用檢視](#)
- [建立檢視](#)
- [檢視的範例](#)
- [使用 AWS Glue Data Catalog 視圖](#)

## 何時使用檢視？

您建立檢視可能是為了：

- 查詢資料子集。例如，您可以根據原始資料表建立包含資料欄子集的檢視，以簡化查詢資料。
- 在一個查詢中合併多個資料表。當您有多個資料表，且想要使用 UNION ALL 合併它們時，您可以使用該表達式建立檢視，以簡化查詢已合併的資料表。
- 隱藏現有基礎查詢的複雜性並簡化使用者執行的查詢。基礎查詢通常包含資料表之間的聯結、欄清單中的表達式，以及其他 SQL 語法，導致難以了解和除錯。您可以建立檢視以隱藏複雜性並簡化查詢。
- 實驗最佳化技巧並建立最佳化查詢。例如，如果您發現有一組 WHERE 條件、JOIN 順序或其他表達式可展現最佳效能，您可以使用這些子句和表達式來建立檢視。然後，應用程式可以對此檢視執行相當簡單的查詢。如果您後來發現有更好的方式可最佳化原始查詢，當您重新建立檢視時，所有應用程式就可立即利用最佳化基礎查詢。
- 隱藏基礎資料表和欄名稱，將維護問題減到最少 (如果這些名稱變更)。在這種情況下，您可以使用新的名稱重新建立檢視。所有使用檢視 (而不是基礎資料表) 的查詢可繼續執行，而不需要變更。

## Athena 中對檢視支援的動作

Athena 對檢視支援下列動作。您可以在查詢編輯器中執行這些命令。

陳述式	描述
<a href="#">CREATE VIEW</a>	從指定的 SELECT 查詢建立新的檢視。如需詳細資訊，請參閱 <a href="#">建立檢視</a> 。  選用的 OR REPLACE 子句可讓您透過取代來更新現有的檢視。
<a href="#">DESCRIBE VIEW</a>	顯示指定檢視的資料欄清單。這可讓您檢查複雜檢視的屬性。
<a href="#">DROP VIEW</a>	刪除現有的檢視。如果檢視不存在，選用的 IF EXISTS 子句可以阻止錯誤發生。
<a href="#">SHOW CREATE VIEW</a>	顯示用於建立指定檢視的 SQL 陳述式。
<a href="#">SHOW VIEWS</a>	列出指定的資料庫或目前資料庫 (如果省略資料庫名稱) 中的檢視。使用選用的 LIKE 子句搭配規則表達式，以限制檢視名稱的清單。您也可以在主控台的左側窗格中看到檢視清單。
<a href="#">SHOW COLUMNS</a>	列出檢視的結構描述中的欄。

## 檢視的考量

在 Athena 中建立和使用檢視時有以下考量事項：

- 在 Athena 中，您可以預覽並使用在 Athena 主控台中建立的檢視，如果您已遷移到使用檢視 AWS Glue Data Catalog，或者在連線至相同目錄的 Amazon EMR 叢集上執行 Presto，則可以在中進行預覽和使用。您無法預覽或在 Athena 中新增以其他方式建立的檢視。
- 如果您已在資料目錄中建立 Athena 檢視，則資料目錄會將檢視視為資料表。您可以在資料目錄中使用資料表層級的精細存取控制，對這些檢視[限制存取](#)。
- Athena 會阻止您執行遞迴檢視，且在這種情況下會顯示錯誤訊息。遞迴檢視是自我參考的檢視查詢。
- Athena 會在偵測到過時的檢視時顯示錯誤訊息。當發生下列其中一種情況時，便會報告過時的檢視：
  - 檢視參考資料表或資料庫不存在。
  - 在參考的資料表中進行了結構描述或中繼資料變更。
  - 參考的資料表遭捨棄，並使用不同的結構描述或組態重新建立。
- 只要巢狀檢視背後的查詢有效，而且資料表和資料庫存在，您就可以建立和執行巢狀檢視。

## 檢視限制

- Athena 檢視名稱不可包含除了底線 (\_) 以外的特殊字元。如需詳細資訊，請參閱[資料表、資料庫和資料欄的名稱](#)。
- 避免使用預留關鍵字來命名檢視。如果您使用預留關鍵字，請在檢視的查詢中以雙引號括住預留關鍵字。請參閱[保留的關鍵字](#)。
- 您無法將在 Athena 中建立的檢視與外部 Hive 中繼存放區或 UDF 搭配使用。如需使用在 Hive 外部建立的檢視的相關資訊，請參閱[使用 Hive 檢視](#)。
- 您不能使用具有地理空間函數的檢視。
- 對於 Amazon S3 中的資料，您無法使用檢視來管理存取控制。若要查詢檢視，您需要許可來存取 Amazon S3 中存放的資料。如需詳細資訊，請參閱[從 Athena 訪問 Amazon S3](#)。
- 雖然 Athena 引擎版本 2 和 Athena 引擎版本 3 皆支援跨帳戶查詢檢視，但您無法建立包含跨帳戶 AWS Glue Data Catalog 的檢視。如需有關跨帳戶資料目錄存取的資訊，請參閱[跨帳戶存取 AWS Glue 資料目錄](#)。
- Athena 中的檢視不支援 Hive 或 Iceberg 隱藏的中繼資料欄 `$bucket`、`$file_modified_time`、`$file_size` 和 `$partition`。如需有關在 Athena 中使用 `$path` 中繼資料欄的資訊，請參閱[在 Amazon S3 中取得來源資料的檔案位置](#)。

## 在主控台使用檢視

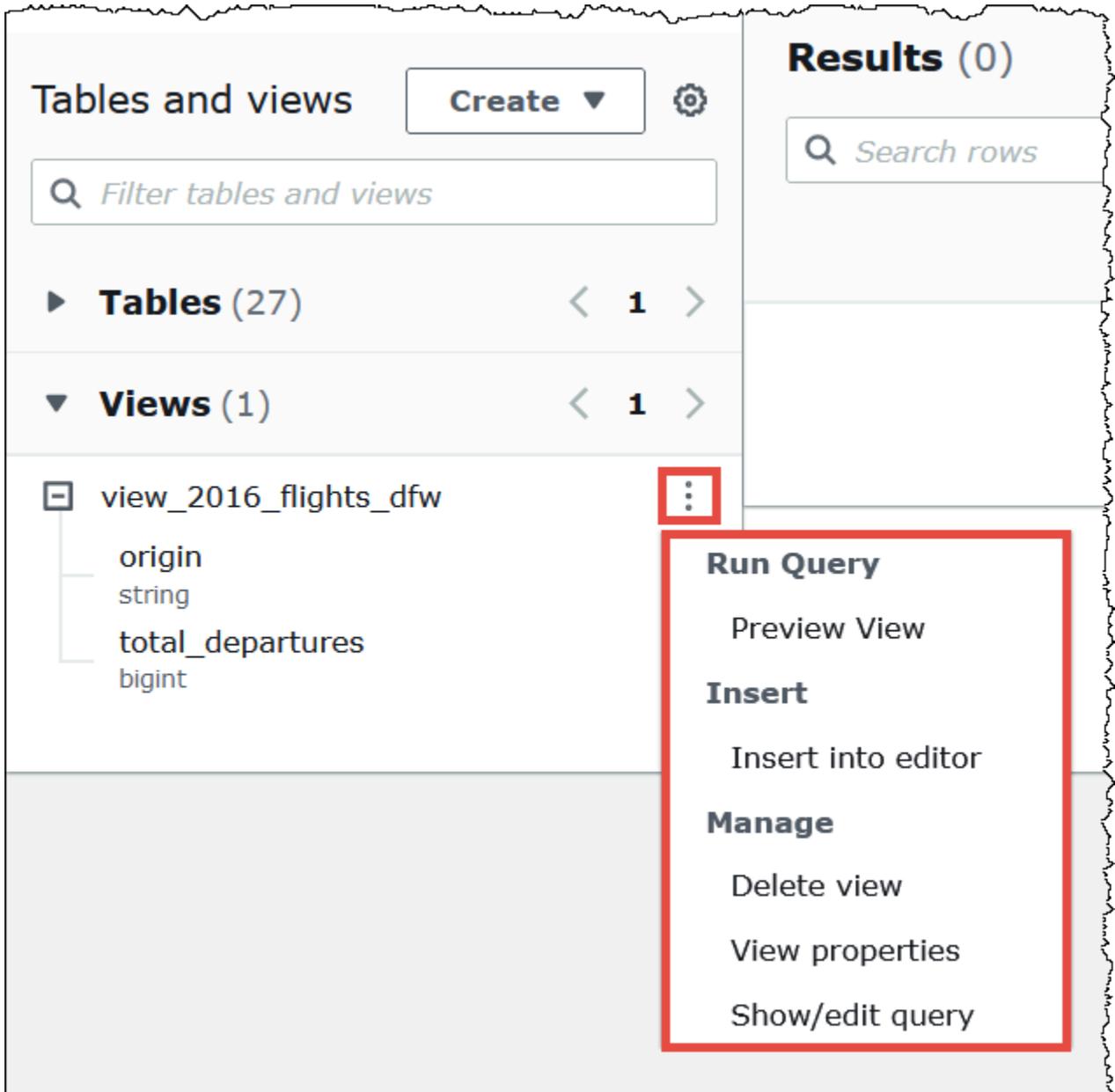
在 Athena 主控台中，您可以：

- 在列出資料表的左側窗格中，找到所有檢視。
- 篩選檢視。
- 預覽檢視、顯示其屬性、編輯它或刪除它。

### 顯示檢視畫面的動作

必須先建立檢視，檢視才會出現在主控台中。

1. 在 Athena 主控台中，選擇 Views (檢視)，然後選擇一個檢視以將其展開並在檢視中顯示資料欄。
2. 選擇檢視旁的三個垂直點可顯示檢視的動作清單。



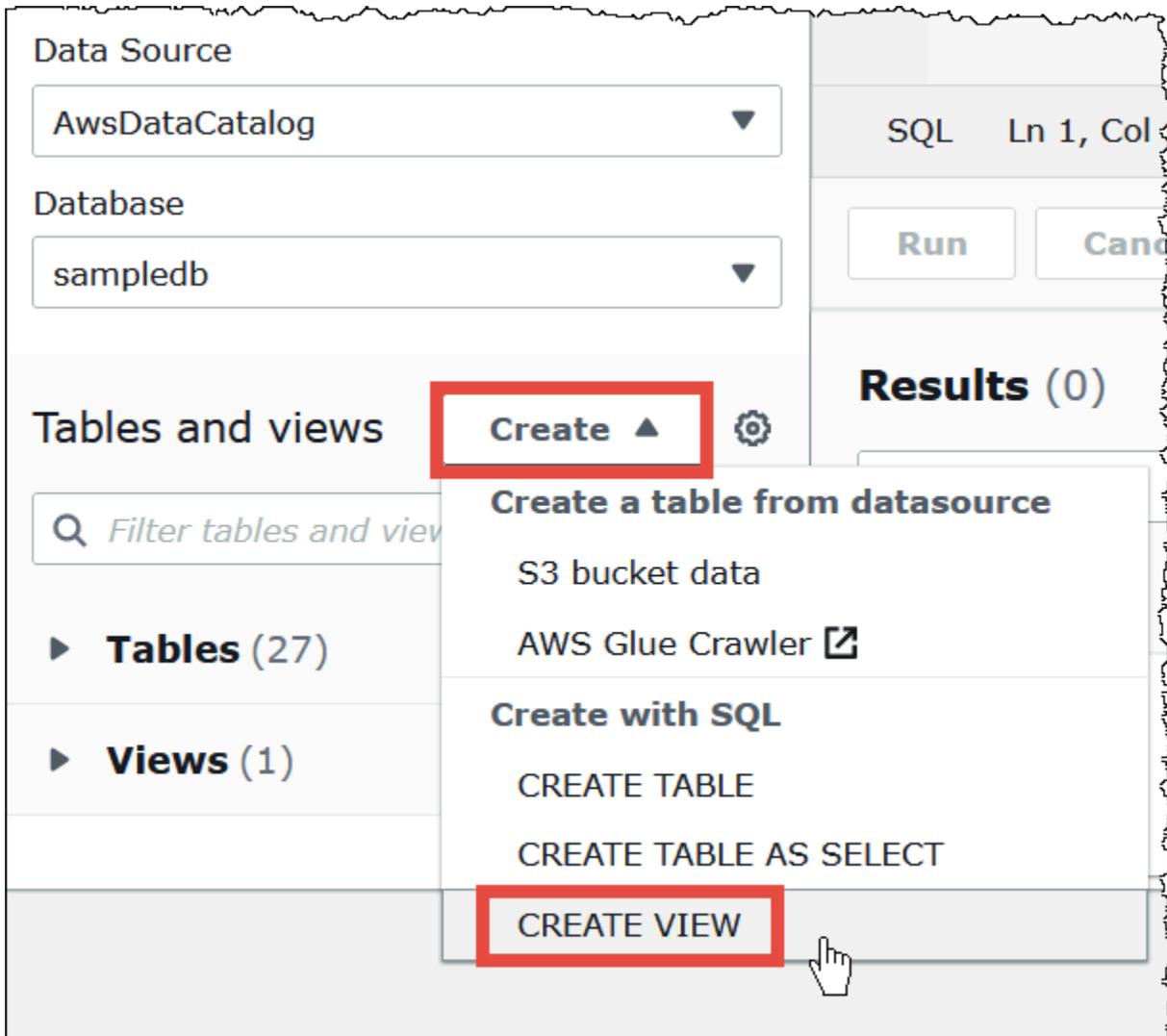
3. 選擇動作可預覽檢視、將檢視名稱插入查詢編輯器中、刪除檢視、查看檢視的屬性，或在查詢編輯器中顯示和編輯檢視。

## 建立檢視

使用範本或執行現有查詢可在 Athena 主控台中建立檢視。

## 使用範本建立檢視

1. 在 Athena 主控台中的 Tables and views (資料表和檢視) 旁，選擇 Create (建立)，然後選擇 Create view (建立檢視)。



此動作會將可編輯的檢視範本放入查詢編輯器中。

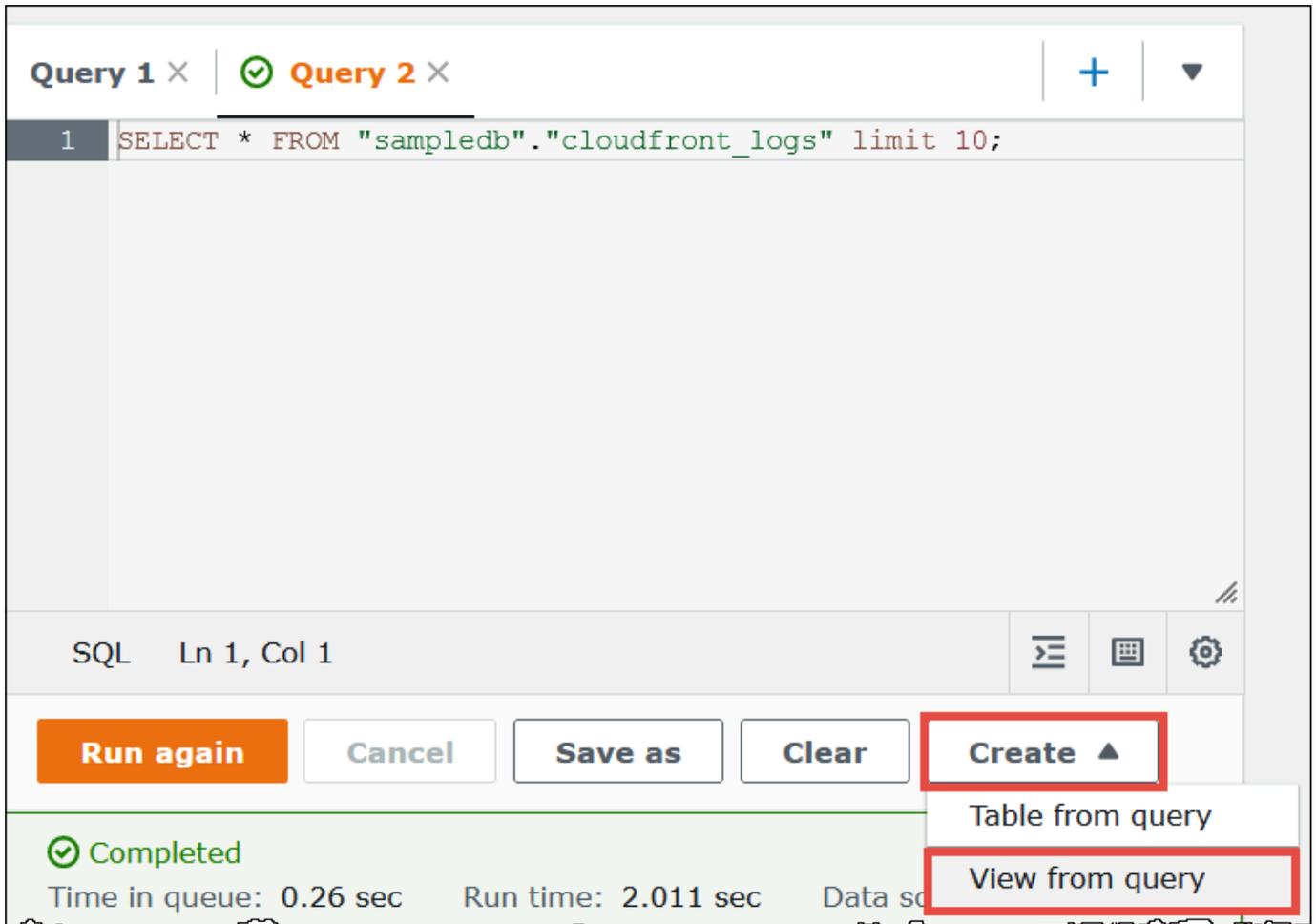
2. 根據您的需求編輯檢視範本。在陳述式中輸入檢視的名稱時，請謹記，視圖名稱不能包含底線 (\_) 以外的特殊字元。請參閱[資料表、資料庫和資料欄的名稱](#)。避免使用[保留的關鍵字](#)來命名檢視。

如需建立檢視的相關詳細資訊，請參閱[CREATE VIEW](#)和[檢視的範例](#)。

3. 選擇 Run (執行) 以建立檢視。檢視隨即出現在 Athena 主控台的檢視清單中。

## 根據現有查詢建立檢視

1. 使用 Athena 查詢編輯器執行現有查詢。
2. 在查詢編輯器視窗底下，選擇 Create (建立)，然後選擇 View from query (從查詢建立檢視)。



3. 在 Create View (建立檢視) 對話方塊中輸入檢視的名稱，然後選擇 Create (建立)。檢視名稱不可包含除了底線 (\_) 以外的特殊字元。請參閱[資料表、資料庫和資料欄的名稱](#)。避免使用[保留的關鍵字](#)來命名檢視。

Athena 會將檢視新增至主控台的檢視清單中，並在查詢編輯器中顯示檢視的 CREATE VIEW 陳述式。

### 備註

- 如果您刪除用以建立資料表的資料表，然後嘗試執行檢視，Athena 會顯示錯誤訊息。
- 您可以建立巢狀檢視，這是位於現有檢視上方的檢視。Athena 不會讓您執行自我參照的遞迴檢視。

## 檢視的範例

若要顯示檢視查詢的語法，請使用 [SHOW CREATE VIEW](#)。

### Example 範例 1

假設有以下兩個資料表：資料表 `employees` 有兩欄：`id` 和 `name`；資料表 `salaries` 有兩欄：`id` 和 `salary`。

在這個範例中，我們以 `SELECT` 查詢建立名為 `name_salary` 的檢視，從資料表 `employees` 和 `salaries` 取得映射到 `salaries` 的 ID 清單：

```
CREATE VIEW name_salary AS
SELECT
  employees.name,
  salaries.salary
FROM employees, salaries
WHERE employees.id = salaries.id
```

### Example 範例 2

在下列範例中，我們建立名為 `view1` 的檢視，讓您隱藏較複雜的查詢語法。

此檢視在兩個資料表上執行：`table1` 和 `table2`，其中每個資料表是不同的 `SELECT` 查詢。此檢視從 `table1` 選取資料欄，然後將結果與 `table2` 聯結。聯結是根據兩個資料表中都存在的 `a` 資料欄。

```
CREATE VIEW view1 AS
WITH
  table1 AS (
    SELECT a,
    MAX(b) AS the_max
    FROM x
    GROUP BY a
  ),
  table2 AS (
    SELECT a,
    AVG(d) AS the_avg
    FROM y
    GROUP BY a)
SELECT table1.a, table1.the_max, table2.the_avg
FROM table1
JOIN table2
ON table1.a = table2.a;
```

如需有關查詢聯合檢視的資訊，請參閱 [查詢聯合檢視](#)。

## 使用 AWS Glue Data Catalog 視圖

這項功能目前在預覽版本中，並可能會有所變更。如需詳細資訊，請參閱 [AWS 服務條款](#) 文件中的「測試版和預覽版」一節。

當您想要橫跨亞馬遜雅典娜和亞馬 Amazon Redshift AWS 服務 等單一通用 AWS Glue Data Catalog 視圖時，請使用視圖。在 Data Catalog 檢視中，存取許可由建立檢視的使用者定義，而不是由查詢檢視的使用者定義。授予許可的這種方法被稱為定義程式語義。

下列使用案例顯示如何使用 Data Catalog 檢視。

- **更大的存取權** – 您建立的檢視會根據使用者需要的許可層級，來限制資料存取權。例如，您可使用 Data Catalog 檢視，來阻止不在人力資源 (HR) 部門工作的員工查看可識別的個人身分資訊。
- **確保完整記錄** – 透過將某些篩選條件套用至 Data Catalog 檢視，您可確保 Data Catalog 檢視中的資料記錄一律是完整的。
- **增強的安全性** – 在 Data Catalog 檢視中，建立檢視的查詢定義必須完整，才能建立檢視。這使得 Data Catalog 檢視不易受到惡意執行者的 SQL 命令的影響。
- **防止存取基礎資料表** – 定義程式語意可讓使用者存取檢視，而無需讓基礎資料表可供其使用。只有定義檢視的使用者才需要存取該資料表。

Data Catalog 檢視定義存放在 AWS Glue Data Catalog 中。這意味著您可使用 AWS Lake Formation，以透過資源授權、資料欄授權或以標籤為基礎的存取控制來授予存取權。如需有關在 Lake Formation 中授予和撤銷存取權的詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的 [授與和撤銷存取 Data Catalog 資源的許可](#)。

### 許可

Data Catalog 檢視需要三個角色：Lake Formation Admin、Definer 和 Invoker。

- **Lake Formation Admin** – 具有設定所有 Lake Formation 許可的存取權。
- **Definer** – 建立 Data Catalog 檢視。對於檢視定義參考的所有基礎資料表，Definer 角色必須具有完整的可授予 SELECT 許可。
- **Invoker** – 可查詢 Data Catalog 檢視或檢查其中繼資料。若要顯示查詢的呼叫程式，您可以使用 `invoker_principal()` DML 函式。如需詳細資訊，請參閱 [發票主體 \(\)](#)。

Definer角色的信任關係必須允許 AWS Glue 和 Lake Formation 服務主體的sts:AssumeRole動作，如下列範例所示。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "glue.amazonaws.com",
          "lakeformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

還需要 Athena 進行存取的 IAM 許可。如需詳細資訊，請參閱 [AWS Amazon Athena 的受管政策](#)。

## 限制

- Data Catalog 檢視無法參考其他檢視。
- 檢視定義中可參考最多 10 個資料表。
- 基礎資料表必須在 Lake Formation 註冊。
- DEFINER 主體只能是 IAM 角色。
- DEFINER 角色必須具有基礎資料表的完整 SELECT (可授予) 許可。
- UNPROTECTED Data Catalog 檢視不受支援。
- 檢視定義中不支援使用者定義函數 (UDF)。
- Athena 聯合資料來源無法用於 Data Catalog 檢視。
- 外部 Hive 中繼儲存不支援 Data Catalog 檢視。
- Athena 會在偵測到過時的檢視時顯示錯誤訊息。當發生下列其中一種情況時，便會報告過時的檢視：
  - 檢視參考資料表或資料庫不存在。
  - 在參考的資料表中進行了結構描述或中繼資料變更。
  - 參考的資料表遭捨棄，並使用不同的結構描述或組態重新建立。

## 建立 Data Catalog 檢視

下列範例語法顯示具有 Definer 角色的使用者如何建立 orders\_by\_date Data Catalog 檢視。此範例假設 Definer 角色對 default 資料庫中的 orders 資料表具有完整的 SELECT 許可。

```
CREATE PROTECTED MULTI DIALECT VIEW orders_by_date
SECURITY DEFINER
AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
WHERE order_city = 'SEATTLE'
GROUP BY orderdate
```

## 查詢 Data Catalog 檢視

建立檢視之後，Lake Formation Admin 可將 Data Catalog 檢視的 SELECT 許可授予 Invoker 主體。然後，Invoker 主體可查詢檢視，而無需存取檢視參考的基礎基本資料表。以下是範例 Invoker 查詢。

```
SELECT * from orders_by_date where price > 5000
```

## 更新 Data Catalog 檢視

Lake Formation Admin 或 Definer 可使用 ALTER VIEW UPDATE DIALECT 語法來更新檢視定義。下列範例會修改檢視定義，以從 returns 資料表，而非 orders 資料表中選取資料欄。

```
ALTER VIEW orders_by_date UPDATE DIALECT
AS
SELECT return_date, sum(totalprice) AS price
FROM returns
WHERE order_city = 'SEATTLE'
GROUP BY orderdate
```

如需有關建立和管理 Data Catalog 檢視的語法的詳細資訊，請參閱 [Glue Data Catalog 檢視語法](#)。

## Glue Data Catalog 檢視語法

這項功能目前在預覽版本中，並可能會有所變更。如需詳細資訊，請參閱 [AWS 服務條款](#) 文件中的「測試版和預覽版」一節。

本節說明用於建立和管理 AWS Glue Data Catalog 檢視表的資料定義語言 (DDL) 命令。

## 變更檢視方言

您可透過新增引擎方言，或者更新或捨棄現有引擎方言來更新 Data Catalog 檢視。只有 Lake FormationAdmin 和 Definer (建立檢視的使用者) 具有在 Data Catalog 檢視上使用 ALTER VIEW DIALECT 陳述式的許可。

## 語法

```
ALTER VIEW name [ FORCE ] [ ADD|UPDATE ] DIALECT AS query
```

```
ALTER VIEW name [ DROP ] DIALECT
```

## FORCE

FORCE 關鍵字會導致檢視中的衝突性引擎方言資訊被新的定義覆寫。當 Data Catalog 檢視更新導致現有引擎方言具有衝突性檢視定義時，FORCE 關鍵字非常有用。假設 Data Catalog 檢視同時具有 Athena 和 Amazon Redshift 方言，而且更新會導致檢視定義中與 Amazon Redshift 方言發生衝突。在此情況下，您可使用 FORCE 關鍵字允許更新完成，並將 Amazon Redshift 方言標記為過時。當引擎標記為過時查詢檢視時，查詢會失敗。引擎擲回例外狀況，以禁止過時的結果。若要更正此問題，請更新檢視中過時的方言。

## ADD

將新引擎方言新增至 Data Catalog 檢視。指定的引擎不得存在於 Data Catalog 檢視中。

## UPDATE

更新 Data Catalog 檢視中已存在的引擎方言。

## DROP

從 Data Catalog 檢視中捨棄現有的引擎方言。從 Data Catalog 檢視中捨棄引擎後，捨棄的引擎無法查詢 Data Catalog 檢視。檢視中的其他引擎方言仍可查詢檢視。

## DIALECT AS

引入引擎特定的 SQL 查詢。

## 範例

```
ALTER VIEW orders_by_date FORCE ADD DIALECT
```

```
AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
GROUP BY orderdate
```

```
ALTER VIEW orders_by_date FORCE UPDATE DIALECT
AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
GROUP BY orderdate
```

```
ALTER VIEW orders_by_date DROP DIALECT
```

## CREATE PROTECTED MULTI DIALECT VIEW

在中建立資料目錄檢視 AWS Glue Data Catalog。Data Catalog 檢視是一種單一檢視結構描述，可在 Athena 和其他 SQL 引擎 (例如 Amazon Redshift 和 Amazon EMR) 中順暢運作。

### 語法

```
CREATE [ OR REPLACE ] PROTECTED MULTI DIALECT VIEW view_name
[ SECURITY DEFINER ]
AS query
```

### 受保護

必要的關鍵字。指定檢視受到保護，以防止資料洩漏。Data Catalog 檢視只能做為 PROTECTED 檢視來建立。

### 多方言

指定檢視支援不同查詢引擎的 SQL 方言，因此這些引擎可讀取。

### 安全定義程式

指定對此檢視有效的定義程式語意。定義程式語意意味著基礎資料表上的有效讀取許可屬於定義檢視的主體或角色，而不是執行實際讀取的主體。

### OR REPLACE

如果檢視中存在其他引擎的 SQL 方言，則無法取代 Data Catalog 檢視。如果呼叫引擎只存取 SQL 方言，則可取代該檢視。

## 範例

下列範例會根據 `orders` 資料表上的查詢，來建立 `orders_by_date` Data Catalog 檢視。

```
CREATE PROTECTED MULTI DIALECT VIEW orders_by_date
SECURITY DEFINER
AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
WHERE order_city = 'SEATTLE'
GROUP BY orderdate
```

## DESCRIBE

顯示指定 Data Catalog 檢視的資料欄清單。DESCRIBE 陳述式類似於 Athena 檢視的 DESCRIBE 陳述式。與 Athena 檢視不同，陳述式的輸出透過 Lake Formation 存取控制來進行控制。因此，此查詢的輸出並非檢視的所有資料欄，而只是呼叫者可存取的資料欄。

## 語法

```
DESCRIBE [db_name.]view_name
```

## 範例

```
DESCRIBE orders
```

## DROP VIEW

只有當 Data Catalog 檢視存在呼叫引擎方言時，才會捨棄 Data Catalog 檢視。例如，如果使用者從 Athena 呼叫 DROP VIEW，則只有在檢視中存在 Athena 的方言時，才會捨棄該檢視。否則，操作會失敗。只有 Lake Formation 管理員和檢視定義程式，才具有在 Data Catalog 檢視上使用 DROP VIEW 陳述式的許可。

## 語法

```
DROP VIEW [ IF EXISTS ] view_name
```

## 範例

```
DROP VIEW orders_by_date
```

```
DROP FORCE VIEW IF EXISTS orders_by_date
```

如果檢視不存在，則選用的 IF EXISTS 子句會造成錯誤隱藏。

## SHOW COLUMNS

僅顯示單一指定 Data Catalog 檢視的資料欄名稱。SHOW COLUMNS 陳述式類似於 Athena 檢視的 SHOW COLUMNS 陳述式。與 Athena 檢視不同，陳述式的輸出透過 Lake Formation 存取控制來進行控制。因此，此查詢的輸出並非檢視的所有資料欄，而只是呼叫者可存取的資料欄。

### 語法

```
SHOW COLUMNS {FROM|IN} database_name.view_name
```

```
SHOW COLUMNS {FROM|IN} view_name [{FROM|IN} database_name]
```

## SHOW CREATE VIEW

顯示建立 Data Catalog 檢視的 SQL 語法。傳回的 SQL 會顯示 Athena 使用的建立檢視語法。只有 Lake Formation 管理員和檢視定義程式主體，才有權呼叫 SHOW CREATE VIEW Data Catalog 檢視上的。

### 語法

```
SHOW CREATE VIEW view_name
```

### 範例

```
SHOW CREATE VIEW orders_by_date
```

## SHOW VIEWS

列出資料庫中所有檢視的名稱。這會列出資料庫中所有具有 Athena 引擎 SQL 方言的 Data Catalog 檢視。檢視中不存在 Athena 引擎方言的其他 Data Catalog 檢視則會被篩選掉。

### 語法

```
SHOW VIEWS [IN database_name] [LIKE 'regular_expression']
```

## 範例

```
SHOW VIEWS
```

```
SHOW VIEWS IN marketing_analytics LIKE 'orders*'
```

## 使用已儲存的查詢

您可以使用 Athena 主控台保存、編輯、執行、重新命名和刪除您在 Athena 查詢編輯器中建立的查詢。

### 考量與限制

- 您可以更新儲存的查詢的名稱、描述和查詢文字。
- 您只能在您自己的帳戶中更新查詢。
- 您無法變更查詢所屬的工作群組或資料庫。
- Athena 不會保留查詢修改的歷史記錄。如果您要保留查詢的特定版本，請使用其他名稱進行儲存。

## 在 Athena 主控台中使用儲存的查詢

### 若要儲存查詢並為其命名

1. 在 Athena 主控台查詢編輯器中，輸入或執行查詢。
2. 在查詢編輯器視窗上方的查詢標籤上，選擇三個垂直點，然後選擇 Save as (另存新檔)。
3. 在 Save query (儲存查詢) 對話方塊中，輸入查詢的名稱和選擇性描述。您可以使用可展開的 Preview SQL query (預覽 SQL 查詢) 視窗先確認查詢的內容，然後再儲存查詢。
4. 選擇 Save query (儲存查詢)。

在查詢編輯器中，查詢的索引標籤顯示您指定的名稱。

### 執行已儲存的查詢

1. 在 Athena 主控台中，選擇 Saved queries (已儲存的查詢) 索引標籤。
2. 在 Saved queries (已儲存的查詢) 清單中，選擇您要執行的查詢 ID。

查詢編輯器會顯示您選擇的查詢。

3. 選擇執行。

## 若要編輯已儲存的查詢

1. 在 Athena 主控台中，選擇 Saved queries (已儲存的查詢) 索引標籤。
2. 在 Saved queries (已儲存的查詢) 清單中，選擇您要編輯的查詢 ID。
3. 在查詢編輯器中編輯查詢。
4. 執行下列步驟之一：
  - 若要執行查詢，選擇 Run (執行)。
  - 若要儲存查詢，請選擇查詢標籤上的三個垂直點，然後選擇 Save (儲存)。
  - 若要以不同的名稱儲存查詢，請選擇查詢標籤上的三個垂直點，然後選擇 Save as (另存新檔)。

## 重新命名或刪除已顯示在查詢編輯器中的已儲存的查詢

1. 選擇查詢標籤上的三個垂直點，然後選擇 Rename (重新命名) 或 Delete (刪除)。
2. 依照提示重新命名或刪除查詢。

## 若要重新命名已顯示在查詢編輯器中的已儲存的查詢

1. 在 Athena 主控台中，選擇 Saved queries (已儲存的查詢) 索引標籤。
2. 選取您要重新命名的查詢的核取方塊。
3. 選擇 Rename (重新命名)。
4. 在 Rename query (重新命名查詢) 對話方塊中，編輯查詢的名稱和查詢的描述。您可以使用可展開的 Preview SQL query (預覽 SQL 查詢) 視窗先確認查詢的內容，然後再重新命名。
5. 選擇 Rename query (重新命名查詢)。

重新命名的查詢會出現在 Saved queries (已儲存的查詢) 清單中。

## 若要刪除已顯示在查詢編輯器中的已儲存的查詢

1. 在 Athena 主控台中，選擇 Saved queries (已儲存的查詢) 索引標籤。
2. 選取一個或多個您要刪除的查詢的核取方塊。
3. 選擇 Delete (刪除)。
4. 在確認提示中，選擇 Delete (刪除)。

將從 Saved queries (已儲存的查詢) 清單中刪除一個或多個查詢。

## 使用 Athena API 更新已儲存的查詢

如需有關使用 Athena API 更新已儲存的查詢的詳細資訊，請參閱 Athena API 參考中的 [UpdateNamedQuery](#) 動作。

## 使用參數化查詢

您可以使用 Athena 參數化查詢，在執行時間使用不同的參數值重新執行相同的查詢，並協助防範 SQL 隱碼攻擊。在 Athena 中，參數化查詢可以採用任何 DML 查詢或 SQL 預備陳述式中的執行參數形式。

- 具有執行參數的查詢可以在一個步驟中完成，而且並非針對特定工作群組。對於您想要參數化的值，可以在任何 DML 查詢中放置問號。當您執行查詢時，依序宣告執行參數值。參數的宣告和參數值的指派可以在同一個查詢中完成，但是以解耦的方式進行。與預備陳述式不同，您可以在提交具有執行參數的查詢時選取工作群組。
- 預備陳述式需要兩個獨立的 SQL 陳述式：PREPARE 和 EXECUTE。首先，您在 PREPARE 陳述式中定義參數。接下來，執行一個為您定義的參數提供值的 EXECUTE 陳述式。預備陳述式針對特定工作群組；您無法在其所屬的工作群組情境之外執行這些陳述式。

## 考量與限制

- Athena 引擎版本 2 和更新版本才支援參數化查詢。如需有關 Athena 引擎版本的資訊，請參閱 [Athena 引擎版本控制](#)。
- 目前，參數化查詢僅支援 SELECT、INSERT INTO、CTAS 以及 UNLOAD 陳述式。
- 在參數化查詢中，參數具有位置性，以 ? 表示。依參數在查詢中的順序指派參數值。不支援具名參數。
- 目前，? 參數只能放置在 WHERE 子句。不支援類似 SELECT ? FROM table 的語法。
- 問號參數不能放在雙引號或單引號中 (也就是說，'?' 和 "?" 不是有效的語法)。
- 若要將 SQL 執行參數視為字串，它們必須使用單引號而非雙引號。
- 如有必要，您可以在輸入參數化術語的值時使用 CAST 函數。例如，如果您有一個已在查詢中參數化的 date 類型資料欄，並且想要查詢日期 2014-07-05，則輸入 CAST('2014-07-05' AS DATE) 參數值將會傳回結果。
- 預備陳述式針對特定工作群組，預備陳述式的名稱在工作群組中必須是唯一的。
- 需要預備陳述式的 IAM 權限。如需詳細資訊，請參閱 [允許存取預備陳述式](#)。
- 在 Athena 主控台中具有執行參數的查詢限制為最多 25 個問號。

## 使用執行參數進行查詢

您可以在任何 DML 查詢中使用問號預留位置來建立參數化查詢，無需建立預備陳述式。若要執行這些查詢，您可以使用 Athena 主控台，或使用 AWS CLI 或 AWS 開發套件，並在 `execution-parameters` 引數中宣告變數。

在 Athena 主控台中執行具有執行參數的查詢

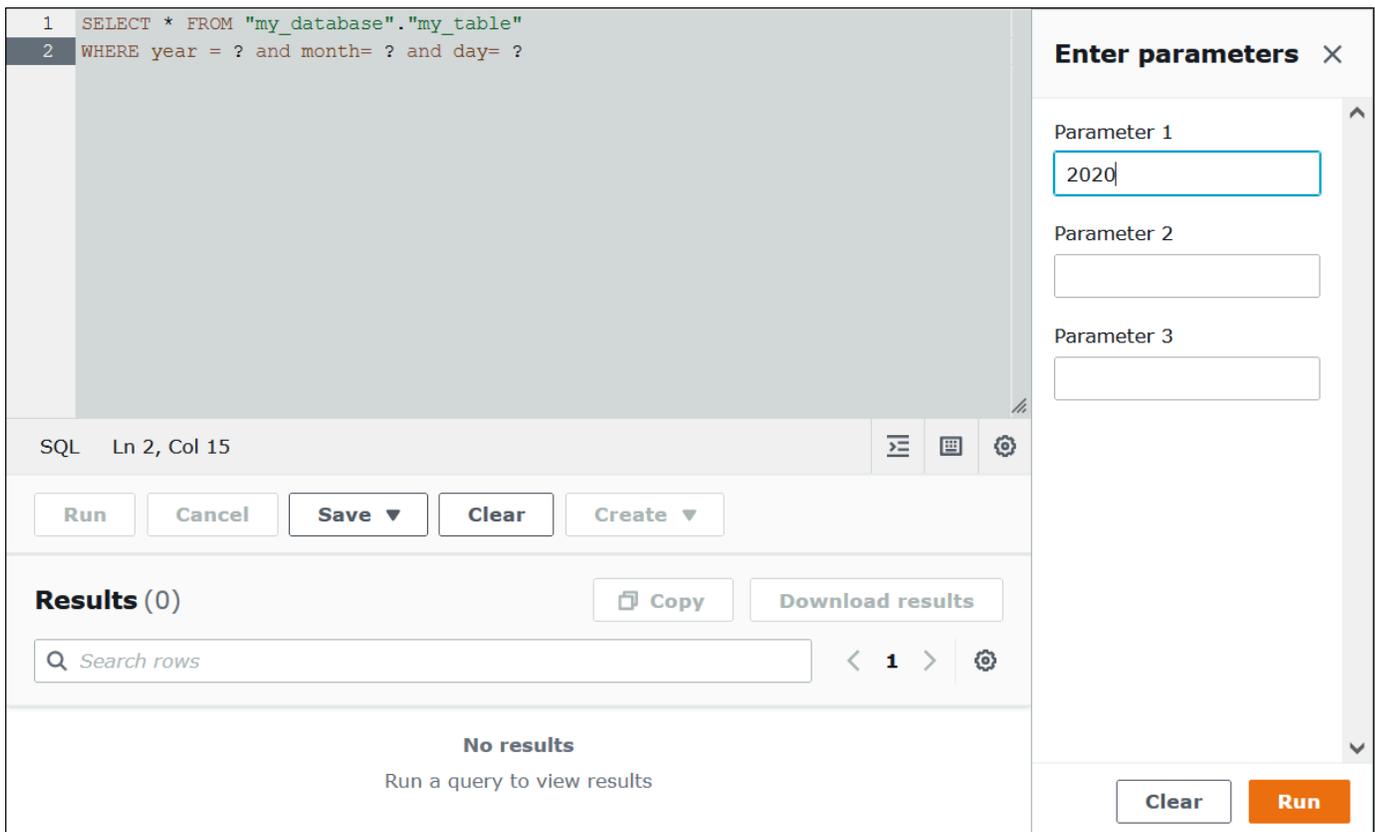
當您在 Athena 主控台中執行具有執行參數 (問號) 的參數化查詢時，系統會依照問號在查詢中出現的順序提示您輸入值。

若要執行具有執行參數的查詢

1. 在 Athena 編輯器中輸入帶有問號預留位置的查詢，如以下範例所示。

```
SELECT * FROM "my_database"."my_table"  
WHERE year = ? and month= ? and day= ?
```

2. 選擇執行。
3. 在 Enter parameters (輸入參數) 對話方塊中，依序輸入查詢中每個問號的值。



The screenshot displays the Athena console interface. On the left, the SQL editor shows a query with three question marks as placeholders: `SELECT * FROM "my_database"."my_table" WHERE year = ? and month= ? and day= ?`. The cursor is positioned at the end of the second line. Below the editor are buttons for `Run`, `Cancel`, `Save`, `Clear`, and `Create`. The `Results (0)` section shows a search bar and navigation controls, with a message indicating `No results` and a prompt to `Run a query to view results`. On the right, the `Enter parameters` dialog box is open, showing three input fields for `Parameter 1`, `Parameter 2`, and `Parameter 3`. The first field contains the value `2020`. At the bottom right of the dialog are `Clear` and `Run` buttons.

4. 完成參數輸入後，請選擇 `Run` (執行)。編輯器會顯示您輸入之參數值的查詢結果。

在此時，您可以執行以下其中一項：

- 為相同的查詢輸入不同的參數值，然後選擇 Run again (再次執行)。
- 若要一次清除輸入的所有值，請選擇 Clear (清除)。
- 若要直接編輯查詢 (例如，若要新增或移除問號)，請先關閉 Enter parameters (輸入參數) 對話方塊。
- 若要儲存參數化查詢以供日後使用，請選擇 Save (儲存) 或 Save as (另存為)，然後為查詢命名。如需使用已儲存查詢的詳細資訊，請參閱 [使用已儲存的查詢](#)。

為了方便起見，只要您在查詢編輯器中使用相同的索引標籤，Enter parameters (輸入參數) 對話方塊就會記住您先前為查詢輸入的值。

### 使用執行參數執行查詢 AWS CLI

若要使用 AWS CLI 執行參數執行查詢，請使用命 start-query-execution 令並在 query-string 引數中提供參數化查詢。然後，在 execution-parameters 引數中，提供執行參數的值。以下範例說明此技術。

```
aws athena start-query-execution
--query-string "SELECT * FROM table WHERE x = ? AND y = ?"
--query-execution-context "Database"="default"
--result-configuration "OutputLocation"="s3://DOC-EXAMPLE-BUCKET;/..."
--execution-parameters "1" "2"
```

### 使用預備陳述式查詢

您可以使用預備陳述式來重複執行具有不同查詢參數的相同查詢。預備陳述式包含參數預留位置，其值在執行時提供。

#### Note

工作群組中預備陳述式的數量上限為 1000。

### SQL 陳述式

您可以使用 PREPARE、EXECUTE 和 DEALLOCATE PREPARE SQL 陳述式，在 Athena 主控台查詢編輯器中執行參數化查詢。

- 若要指定您通常會使用常值的參數，請使用 PREPARE 陳述式中的問號。
- 若要在執行查詢時以值取代參數，請使用 EXECUTE 陳述式中的 USING 子句。
- 若要從工作群組的預備陳述式中移除預備陳述式，請使用 DEALLOCATE PREPARE 陳述式。

下列各節提供有關各個陳述式的其他詳細資訊。

## PREPARE

預備好要在稍後執行的陳述式。預備陳述式會以您指定的名稱儲存在目前的工作群組中。該陳述式可以包含可代替文字的參數，以在查詢執行時替換文字。要由值取代的參數以問號表示。

### 語法

```
PREPARE statement_name FROM statement
```

下表描述了這些參數。

參數	描述
<i>statement_name</i>	要預備之陳述式的名稱。該名稱在工作群組內必須是唯一的。
###	SELECT、CTAS 或 INSERT INTO 查詢。

## PREPARE 範例

下列範例顯示如何使用 PREPARE 陳述式。問號表示執行查詢時由 EXECUTE 陳述式提供的值。

```
PREPARE my_select1 FROM
SELECT * FROM nation
```

```
PREPARE my_select2 FROM
SELECT * FROM "my_database"."my_table" WHERE year = ?
```

```
PREPARE my_select3 FROM
```

```
SELECT order FROM orders WHERE productid = ? and quantity < ?
```

```
PREPARE my_insert FROM  
INSERT INTO cities_usa (city, state)  
SELECT city, state  
FROM cities_world  
WHERE country = ?
```

```
PREPARE my_unload FROM  
UNLOAD (SELECT * FROM table1 WHERE productid < ?)  
TO 's3://DOC-EXAMPLE-BUCKET/'  
WITH (format='PARQUET')
```

## EXECUTE

執行預備陳述式。參數的值在 USING 子句中指定。

### 語法

```
EXECUTE statement_name [USING value1 [ ,value2, ... ] ]
```

*statement\_name* 是預備陳述式的名稱。*value1* 和 *value2* 是要為陳述式中的參數指定的值。

### EXECUTE 範例

下列範例會執行 my\_select1 預備陳述式，其中不包含任何參數。

```
EXECUTE my_select1
```

下列範例會執行 my\_select2 預備陳述式，其中包含一個參數。

```
EXECUTE my_select2 USING 2012
```

下列範例會執行 my\_select3 預備陳述式，其中有兩個參數。

```
EXECUTE my_select3 USING 346078, 12
```

下列範例為預備陳述式 my\_insert 中的參數提供字串值。

```
EXECUTE my_insert USING 'usa'
```

下列範例為預備陳述式 `my_unload` 中的 `productid` 參數提供數值。

```
EXECUTE my_unload USING 12
```

## DEALLOCATE PREPARE

從目前工作群組的預備陳述式清單中移除具有指定名稱的預備陳述式。

### 語法

```
DEALLOCATE PREPARE statement_name
```

*statement\_name* 是要移除的預備陳述式名稱。

### 範例

下列範例會從目前的工作群組中移除 `my_select1` 預備陳述式。

```
DEALLOCATE PREPARE my_select1
```

在 Athena 主控台中執行沒有 USING 子句的預備陳述式

如果您在查詢編輯器中使用語法 `EXECUTE prepared_statement` 執行現有的預備陳述式，則 Athena 會開啟 Enter parameters (輸入參數) 對話方塊，以便您輸入通常會進入 `EXECUTE ... USING` 陳述式 USING 子句的值。

若要使用 Enter parameters (輸入參數) 對話方塊執行預備陳述式

1. 在查詢編輯器中，使用語法 `EXECUTE prepared_statement`，而不是使用語法 `EXECUTE prepared_statement USING value1, value2 ...`。
2. 選擇執行。出現 Enter parameters (輸入參數) 對話方塊。

3. 在 Execution parameters (執行參數) 對話方塊中依序輸入值。由於看不到查詢的原始文字，因此您必須記住每個位置參數的意義，或有預備陳述式可供參考。
4. 選擇執行。

### 使用建立準備好的陳述式 AWS CLI

若要使用 AWS CLI 建立準備好的陳述式，您可以使用下列其中一個 athena 命令：

- 使用 `create-prepared-statement` 命令並提供具有執行參數的查詢陳述式。
- 使用 `start-query-execution` 指令並提供使用 PREPARE 語法的查詢字串。

### 使用 create-prepared-statement

在 `create-prepared-statement` 命令中，在 `query-statement` 引數中定義查詢文字，如以下範例所示。

```
aws athena create-prepared-statement
--statement-name PreparedStatement1
--query-statement "SELECT * FROM table WHERE x = ?"
```

```
--work-group athena-engine-v2
```

## 使用 start-query-execution 和準備語法

使用 start-query-execution 命令。將 PREPARE 陳述式放置在 query-string 引數中，如以下範例所示：

```
aws athena start-query-execution
--query-string "PREPARE PreparedStatement1 FROM SELECT * FROM table WHERE x = ?"
--query-execution-context '{"Database": "default"}'
--result-configuration '{"OutputLocation": "s3://DOC-EXAMPLE-BUCKET/...}"'
```

## 使用執行準備好的陳述式 AWS CLI

若要使用執行已準備好的陳述式 AWS CLI，您可以使用下列其中一種方法來提供參數的值：

- 使用 execution-parameters 引數。
- 在 query-string 引數中使用 EXECUTE ... USING SQL 語法。

## 使用執行參數引數

在此方式中，您可以使用 start-query-execution 命令並在 query-string 引數中提供現有預備陳述式的名稱。然後，在 execution-parameters 引數中，提供執行參數的值。此方法如以下範例所示。

```
aws athena start-query-execution
--query-string "Execute PreparedStatement1"
--query-execution-context "Database="default"
--result-configuration "OutputLocation"="s3://DOC-EXAMPLE-BUCKET/..."
--execution-parameters "1" "2"
```

## 使用 EXECUTE ... 使用 SQL 語法

若要使用 EXECUTE ... USING 語法執行現有的預備陳述式，您可以使用 start-query-execution 命令並將預備陳述式的名稱和參數值都放在 query-string 引數中，如以下範例所示：

```
aws athena start-query-execution
--query-string "EXECUTE PreparedStatement1 USING 1"
--query-execution-context '{"Database": "default"}'
--result-configuration '{"OutputLocation": "s3://DOC-EXAMPLE-BUCKET/...}"'
```

## 列出準備陳述式

若要列出特定工作群組準備好的陳述式，您可以使用 Athena [list-prepared-statements](#) AWS CLI 命令或 [ListPreparedStatements](#) Athena API 動作。--work-group 參數是必要參數。

```
aws athena list-prepared-statements --work-group primary
```

## 其他資源

請參閱 AWS 大數據部落格中的下列相關文章。

- [Improve reusability and security using Amazon Athena parameterized queries](#) (使用 Amazon Athena 參數化查詢改善可再用性和安全性)
- [Use Amazon Athena parameterized queries to provide data as a service](#) (使用 Amazon Athena 參數化查詢提供資料即服務)

## 使用成本型最佳化工具

您可以使用 Athena SQL 中的成本型最佳化工具 (CBO) 功能，來最佳化您的查詢。您可以選擇請求 Athena 為 AWS Glue 的其中一個資料表收集資料表，或資料欄層級的統計資訊。如果查詢中的所有資料表都有統計資訊，Athena 會使用這些統計資訊建立其判斷為效能最佳的執行計畫。查詢最佳化工具會依據統計模型計算備選計畫，然後選取執行查詢最快的計畫。

資料 AWS Glue 表的統計資料會收集並儲存在中，AWS Glue Data Catalog 並提供給 Athena，以改善查詢規劃和執行。這些統計資訊是有關檔案類型 (例如 Parquet、ORC、JSON、ION、CSV 和 XML) 的資料欄層級的統計資訊，例如不同值的數目、空值數目、最大值和最小值。Amazon Athena 會使用這些統計資訊，儘早在查詢處理中套用限制最嚴格的篩選條件，以最佳化查詢。這項篩選會限制記憶體用量，以及為傳送查詢結果必須讀取的記錄數目。

當與 CBO 結合時，Athena 會使用稱為規則型最佳化工具 (RBO) 的功能。RBO 以機械方式套用預期可以改善查詢效能的規則。RBO 通常是有益的，因為它的轉換旨在簡化查詢計畫。但是，由於 RBO 不執行成本計算或計畫比較，因此更複雜的查詢會使 RBO 難以建立最佳計畫。

因此，Athena 同時使用 RBO 和 CBO 來最佳化您的查詢。Athena 在識別到改善查詢執行的機會之後，就會建立最佳計畫。如需有關執行計畫的詳細資訊，請參閱 [檢視 SQL 查詢的執行計畫](#)。如需 CBO 運作方式的詳細討論，請參閱 AWS 大數據部落格中的 [Amazon Athena 使用以成本為基礎的優化器加快查詢速度](#)。

若要產生 AWS Glue 目錄資料表的統計資料，您可以使用 Athena AWS Glue 主控台、主控台或 AWS Glue API。由於 Athena 已與 AWS Glue 型錄整合，因此當您從 Amazon Athena 執行查詢時，您會自動獲得對應的查詢效能改善。

## 考量與限制

- 資料表類型 – 目前，Athena 的 CBO 功能只支援 AWS Glue Data Catalog 中的 Hive 資料表。
- Athena for Spark – CBO 功能在 Athena for Spark 中不可用。
- 定價 – 如需定價資訊，請造訪 [AWS Glue 定價頁面](#)。

## 使用 Athena 主控台產生資料表統計資訊

本節說明如何使用 Athena 主控台在 AWS Glue 中產生資料表或資料表資料欄層級的統計資訊。如需使用 AWS Glue 產生資料表統計資料的相關資訊，請參閱 [AWS Glue 發人員指南中的使用欄統計資料](#)。

若要使用 Athena 主控台產生資料表統計資訊

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在 Athena 查詢編輯器的資料表清單中，為您想要的資料表選擇三個垂直點，然後選擇產生統計資訊。

The screenshot shows the Amazon Athena Query Editor interface. The 'Data' panel is open, displaying a list of tables and views. The 'customer\_address' table is selected, and a context menu is open over it. The 'Generate statistics - new' option is highlighted in red. The query editor shows the query 'select dt.d\_year'.

3. 在產生統計資訊對話方塊中，選擇所有資料欄以產生資料表中所有資料欄的統計資訊，或選擇選取的資料欄來選取特定的資料欄。預設值是所有資料欄。

## Generate statistics for customer\_address ✕

If statistics already exist, they will be updated.

Choose columns

All columns

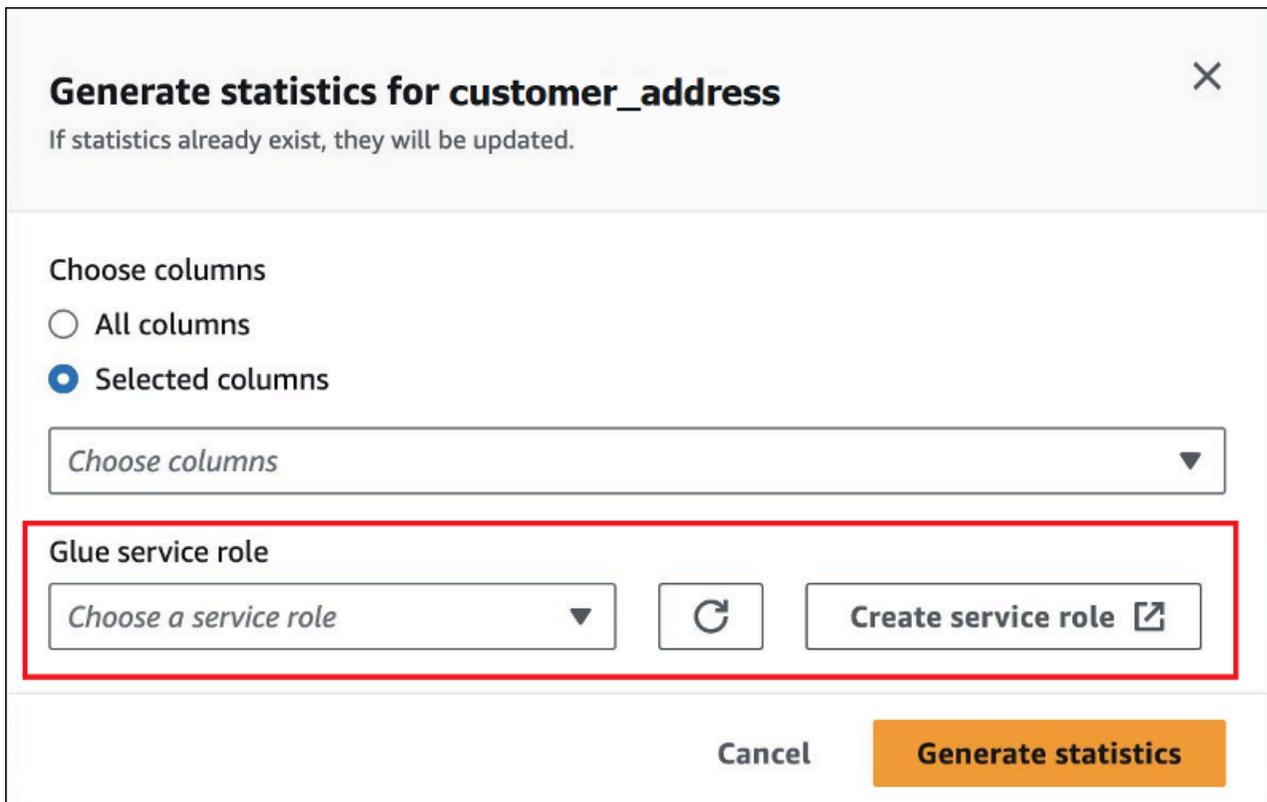
Selected columns

Choose one or more columns ▲

Q

<input checked="" type="checkbox"/>	address_id	string
<input checked="" type="checkbox"/>	address	string
<input type="checkbox"/>	address2	string
<input checked="" type="checkbox"/>	city_id	string
<input type="checkbox"/>	location	string
<input type="checkbox"/>	phone	int

- 對於AWS Glue 服務角色，請建立或選取現有的服務角色，以授與產生統計資料的權限。AWS Glue 服務角色還需要對包含資料表資料的 Amazon S3 儲存貯體的 [S3:GetObject](#) 許可。



**Generate statistics for customer\_address** ✕

If statistics already exist, they will be updated.

Choose columns

All columns

Selected columns

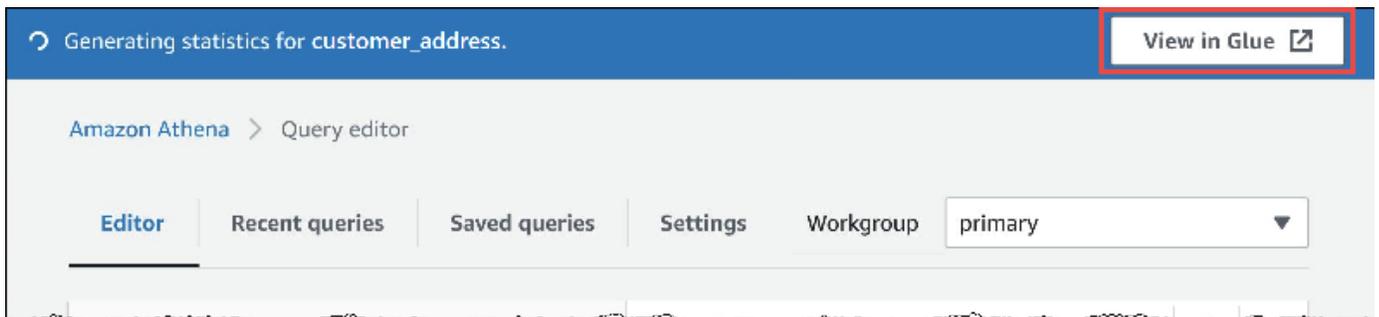
Choose columns ▼

**Glue service role**

Choose a service role ▼ ↻ Create service role ↗

Cancel Generate statistics

5. 選擇產生統計資料。正在產生 *table\_name* 的統計資訊通知橫幅會顯示任務狀態。



6. 若要在 AWS Glue 主控台中檢視詳細資料，請選擇在 Glue 中檢視。

如需在 AWS Glue 主控台中檢視統計資料的詳細資訊，請參閱AWS Glue 開發人員指南中的[檢視欄統計資料](#)

7. 產生統計資訊之後，具有統計資訊的資料表和資料欄會在括弧中顯示統計資訊一詞，如下圖所示。

▼ <b>Tables</b> (16)		< 1 >
 iris-json	<u>(Statistics)</u>	⋮
 iris-json-2.0	<u>(Statistics)</u>	⋮
 iris-json-3.0	<u>(Statistics)</u>	⋮
 iris-json-v2		⋮
 iris-json-v3		⋮
 iris-json-v4	<u>(Statistics)</u>	⋮
 iris-json-v5	<u>(Statistics)</u>	⋮
 iris-json-v6	<u>(Statistics)</u>	⋮

現在，當您執行查詢時，Athena 會針對產生統計資訊的資料表和資料欄，執行成本型最佳化。

## 其他資源

如需其他資訊，請參閱下列資源。

## 查詢 S3 Express 單區域資料

Amazon S3 Express 單區域儲存類別是一種高效能 Amazon S3 儲存類別，可提供延遲不到 10 毫秒的回應時間。因此，對於經常以每秒數十萬個請求存取資料的應用程式而言非常有用。

S3 Express 單區域會在相同的可用區域內複寫和存放資料，以最佳化速度和成本。這與 Amazon S3 區域儲存類別不同，它們會自動跨至少三個 AWS 可用區域複寫資料 AWS 區域。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[什麼是 S3 Express 單區域？](#)。

### 必要條件

請確認滿足下列條件再開始：

- Athena 引擎第 3 版 – 若要搭配 Athena SQL 使用 S3 Express 單區域，您的工作群組必須設定為使用 Athena 引擎第 3 版。
- S3 Express 單區域許可 – 當 S3 Express 單區域在 Amazon S3 物件上呼叫 GET、LIST 或 PUT 等動作，儲存類別會代表您呼叫 CreateSession。因此，您的 IAM 政策必須允許 s3express:CreateSession 動作，這樣 Athena 才能調用對應的 API 作業。

### 考量與限制

當您使用 Athena 查詢 S3 Express 單區域時，請考慮下列幾點。

- S3 Express 單區域儲存貯體僅支援 SSE\_S3 加密。無論您在工作群組設定中選擇哪個選項來加密查詢結果，都會使用 SSE\_S3 加密來編寫 Athena 查詢結果。此限制包括 Athena 將資料寫入 S3 Express 單區域儲存貯體的所有案例，包括 CREATE TABLE AS (CTA) 和 INSERT INTO 陳述式。
- 在 S3 Express 單區資料上建立表格時，不支援 AWS Glue 爬行者程式。
- MSCK REPAIR TABLE 陳述式不受支援。解決方法是使用 [ALTER TABLE ADD PARTITION](#)。
- ALTER TABLE ADD PARTITION、ALTER TABLE DROP PARTITION、和 ALTER TABLE RENAME PARTITION 不支援 S3 快速單區中的冰山表格。
- 下列檔案和資料表格式不受支援或支援有限。如果格式未列出，但 Athena 支援這些格式 (例如 Parquet、ORC 和 JSON)，則使用 S3 Express 單區域時也支援這些格式。

檔案或資料表格式	限制
Apache Avro	不支援

檔案或資料表格式	限制
CloudTrail 日誌	不支援
Apache Hudi	不支援
Amazon Ion	不支援
Logstash 日誌	不支援
阿帕奇 WebServer 日誌	不支援
Delta Lake	DDL 不受支援。如需使用虛擬結構描述建立 Delta Lake 資料表的詳細資訊，請參閱 <a href="#">同步處理 Delta Lake 中繼資料</a> 。支援針對資料表進行的 SELECT 查詢。

## 開始使用

使用 Athena 查詢 S3 Express 單區域資料非常簡單。若要開始使用，請使用下列程序。

### 使用 Athena SQL 查詢 S3 Express 單區域資料

1. 將您的資料轉移到 S3 Express 單區域儲存。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[設定物件的儲存類別](#)。
2. 使用 Athena 中的 [CREATE TABLE](#) 陳述式來編目 AWS Glue Data Catalog 中的資料。如需在 Athena 中建立資料表的詳細資訊，請參閱 [在 Athena 中建立資料表](#) 和 [CREATE TABLE](#) 陳述式。
3. (選用) 設定 Athena 工作群組的查詢結果位置，以使用 Amazon S3 目錄儲存貯體。Amazon S3 目錄儲存貯體的效能比一般儲存貯體更佳，並且專為需要延遲不到 10 毫秒持續效能的工作負載或效能關鍵型應用程式而設計。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[目錄儲存貯體概觀](#)。

## 查詢還原的 Amazon S3 Glacier 物件

您可以使用 Athena 查詢從 [S3 Glacier Flexible Retrieval \(舊稱 Glacier\)](#) 和 [S3 Glacier Deep Archive Amazon S3 儲存類別](#) 中還原的物件。您必須基於每個資料表啟用此功能。如果您在執行查詢之前未在資料表上啟用此功能，Athena 會在查詢執行期略過資料表的所有 S3 Glacier Flexible Retrieval 和 S3 Glacier Deep Archive 物件。

## 考量事項與限制

- 只有 Athena 引擎版本 3 才支援查詢還原的 Amazon S3 Glacier 物件。
- 此功能僅支援 Apache Hive 資料表。
- 在查詢資料之前，您必須先還原物件；Athena 不會為您還原物件。

## 設定資料表，以使用還原的物件

若要將 Athena 資料表設定為在查詢中包含還原的物件，您必須將其 `read_restored_glacier_objects` 資料表屬性設定為 `true`。若要這麼做，您可以使用 Athena 查詢編輯器或主 AWS Glue 控制台。您也可以使用 [AWS Glue CLI](#)、[AWS Glue API](#) 或 [AWS Glue SDK](#)。

### 使用 Athena 查詢編輯器

在 Athena 中，您可以使用 [ALTER TABLE SET TBLPROPERTIES](#) 命令來設定資料表屬性，如下列範例所示。

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'true')
```

### 使用 AWS Glue 主控台

在主 AWS Glue 控制台中，執行下列步驟以新增 `read_restored_glacier_objects` table 屬性。

若要在 AWS Glue 主控台中設定資料表屬性

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，[網址為 https://console.aws.amazon.com/glue/](https://console.aws.amazon.com/glue/)。
2. 執行以下任意一項：
  - 選擇移至資料目錄。
  - 在導覽窗格中，選擇資料目錄資料表。
3. 在資料表頁面的資料表清單中，選擇您要編輯之資料表的連結。
4. 選擇 Actions (動作)、Edit table (編輯資料表)。
5. 在編輯資料表頁面的資料表屬性區段中，新增下列鍵值對。
  - 對於 Key (索引鍵)，新增 `read_restored_glacier_objects`。
  - 針對數值，輸入 `true`。

## 6. 選擇儲存。

### 使用 AWS CLI

在中 AWS CLI，您可以使用 AWS Glue [update-table](#) 命令及其 `--table-input` 引數來重新定義資料表，並在此新增屬性。`read_restored_glacier_objects` 在 `--table-input` 引數中，使用 `Parameters` 結構來指定 `read_restored_glacier_objects` 屬性和 `true` 的值。請注意，`--table-input` 的引數不得有空格，且必須使用反斜線來逸出雙引號。在以下範例中，使用您資料庫和資料表的名稱取代 `my_database` 和 `my_table`。

```
aws glue update-table \  
  --database-name my_database \  
  --table-input={"Name\":\"my_table\",\"Parameters\":{\"read_restored_glacier_objects\":"true\"}}
```

#### Important

該 AWS Glue `update-table` 命令在覆蓋模式下工作，這意味著它將用 `table-input` 參數指定的新定義替換現有的表定義。因此，當您新增屬性時，請務必同時指定 `table-input` 參數中要放在資料表中的所有欄 `read_restored_glacier_objects` 位。

## 處理結構描述更新

本節指導處理各種資料格式的結構描述更新。Athena 是 `schema-on-read` 查詢引擎。這表示當您在 Athena 中建立資料表時，將於讀取資料時套用結構描述。並不會變更或重寫基礎資料。

如果您預期資料表結構描述會變更，請考慮以符合您需要的資料格式建立結構描述。您的目標是隨著不斷演變的結構描述而重複使用現有的 Athena 查詢，並避免在查詢含有分割區的資料表時，發生結構描述不符的錯誤。

為了達到這些目標，請根據下列主題中的表格來選擇資料表的資料格式。

### 主題

- [摘要：Athena 中的更新和資料格式](#)
- [ORC 和 parquet 的索引存取](#)
- [更新類型](#)
- [資料表和分割區中的更新](#)

## 摘要：Athena 中的更新和資料格式

下表彙總資料儲存格式及其支援的結構描述操作。使用此資料表協助您選擇格式，即使結構描述隨著時間而變更，此格式也能讓您繼續使用 Athena 查詢。

在此表格中，請注意 Parquet 和 ORC 是單欄式格式，具有不同的預設欄存取方法。在預設情況下，Parquet 依名稱存取欄，而 ORC 依索引 (序數值) 存取欄。因此，Athena 會在建立資料表時提供定義的 SerDe 屬性，以切換預設資料欄存取方法，以提供結構描述演進的更大彈性。

對於 Parquet，`parquet.column.index.access` 屬性可以設為 `true`，以設定資料欄存取方法來使用資料欄的序號。將此屬性設為 `false` 會變更欄存取方法來使用欄名稱。同樣地，對於 ORC，請使用 `orc.column.index.access` 屬性來控制欄存取方法。如需詳細資訊，請參閱 [ORC 和 parquet 的索引存取](#)。

CSV 和 TSV 可讓您執行所有結構描述操作，但不包括重新排序欄，或在資料表開頭新增欄。例如，如果您的結構描述演變只需要重新命名欄，而不需要移除欄，則您可以選擇以 CSV 或 TSV 建立資料表。如果您需要移除欄，請勿使用 CSV 或 TSV，而應該使用任何其他支援的格式，最好是單欄格式，例如 Parquet 或 ORC。

### Athena 中的結構描述更新和資料格式

預期的結構描述更新類型	Summary	CSV (具有和沒有標頭) 和 TSV	JSON	AVRO	PARQUET 依名稱讀取 (預設)	PARQUET 依索引讀取	ORC：依索引讀取 (預設)	ORC：依名稱讀取
<a href="#">重新命名欄</a>	以 CSV 和 TSV 存放資料，或如果是依索引讀取資料，請以 ORC 和 Parquet 存放資料。	Y	N	N	N	Y	Y	N
<a href="#">在資料表開頭或中間新增欄</a>	以 JSON、AVRO 存放資料，或如果是依名稱讀取資料，請以 Parquet	N	Y	Y	Y	N	N	Y

預期的結構描述更新類型	Summary	CSV (具有和沒有標頭) 和 TSV	JSON	AVRO	PARQUE 依名稱讀取 (預設)	PARQUE 依索引讀取	ORC : 依索引讀取 (預設)	ORC : 依名稱讀取
	和 ORC 存放資料。請勿使用 CSV 和 TSV。							
<a href="#">在資料表尾端新增欄</a>	以 CSV 或 TSV、JSON、AVRO、ORC 或 Parquet 存放您的資料。	Y	Y	Y	Y	Y	Y	Y
<a href="#">移除欄</a>	如果是依名稱讀取資料，請以 JSON、AVRO 或 Parquet 和 ORC 存放資料。請勿使用 CSV 和 TSV。	N	Y	Y	Y	N	N	Y
<a href="#">重新排序欄</a>	如果是依名稱讀取資料，請以 AVRO、JSON 或 ORC 和 Parquet 存放資料。	N	Y	Y	Y	N	N	Y
<a href="#">變更欄的資料類型</a>	以任何格式存放資料，但在 Athena 中測試您的查詢，以確保資料類型相容。對於 Parquet 和 ORC，變更資料類型僅適用於分割資料表。	Y	Y	Y	Y	Y	Y	Y

## ORC 和 parquet 的索引存取

PARQUET 和 ORC 是可依索引或依名稱讀取的單欄資料儲存格式。以這些格式存放資料可讓您對結構描述執行所有操作和執行 Athena 查詢，而不會發生結構描述不符的錯誤。

- Athena 根據預設依索引讀取 ORC，如 `SERDEPROPERTIES ('orc.column.index.access'='true')` 中所定義。如需詳細資訊，請參閱 [ORC：依索引讀取](#)。
- Athena 根據預設依名稱讀取 Parquet，如 `SERDEPROPERTIES ('parquet.column.index.access'='false')` 中所定義。如需詳細資訊，請參閱 [Parquet：依名稱讀取](#)。

由於這些屬性是預設值，因此在 `CREATE TABLE` 查詢中指定這些 SerDe 屬性是可選的，因此會以隱含方式使用它們。使用時可讓您執行一些結構描述更新操作，同時會防止其他這類操作。若要啟用這些作業，請執行其他 `CREATE TABLE` 查詢並變更 SerDe 設定。

### Note

Serde 屬性不會自動傳播到每個分區。使用 `ALTER TABLE ADD PARTITION` 陳述式來設定每個分割區的 SerDe 屬性。若要讓這個程序自動化，請編寫可執行 `ALTER TABLE ADD PARTITION` 陳述式的指令碼。

下列各節詳細說明這些案例。

### ORC：依索引讀取

預設會依索引讀取 ORC 資料表。這是由下列語法所定義：

```
WITH SERDEPROPERTIES (  
    'orc.column.index.access'='true')
```

依索引讀取可讓您重新命名欄。但之後您就無法在資料表中間移除欄或新增欄。

若要讓 ORC 依名稱讀取，這可讓您在資料表中間新增資料欄或移除 ORC 中的資料行，請在陳述式 `false` 中將 SerDe 屬性設定 `orc.column.index.access` 為 `CREATE TABLE` 在此組態中，您將無法重新命名欄。

**Note**

在 Athena 引擎版本 2 中，當 ORC 資料表設定為依名稱讀取時，Athena 會要求 ORC 檔案中的所有資料欄名稱都為小寫字母。Apache Spark 在產生 ORC 檔案時不會以小寫字母顯示欄位名稱，所以 Athena 可能無法讀取所產生的資料。解決方法是重新以小寫字母命名資料欄，或使用 Athena 引擎版本 3。

以下範例說明如何將 ORC 變更為依名稱讀取：

```
CREATE EXTERNAL TABLE orders_orc_read_by_name (  
  `o_comment` string,  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderpriority` string,  
  `o_orderstatus` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_orderdate` string  
)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.orc.OrcSerde'  
WITH SERDEPROPERTIES (  
  'orc.column.index.access'='false')  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_orc/';
```

Parquet：依名稱讀取

預設會依名稱讀取 Parquet 資料表。這是由下列語法所定義：

```
WITH SERDEPROPERTIES (  
  'parquet.column.index.access'='false')
```

依名稱讀取可讓您在資料表中間新增欄及移除欄。但之後您就無法重新命欄。

若要讓 Parquet 透過索引讀取，這可讓您重新命名欄，您必須建立 `parquet.column.index.access` SerDe 屬性設定為的表格 `true`。

## 更新類型

本主題說明您可以在不實際變更資料的情況下對 CREATE TABLE 陳述式中的結構描述進行的一些變更。我們檢閱每一種結構描述更新，並指定哪些資料格式可在 Athena 中實現這些更新。要更新結構描述，您可以在某些情況下使用 ALTER TABLE 命令，但在其他情況下，您實際上不會修改現有的資料表。相反，您可以使用新名稱建立資料表，該資料表會修改您在原始 CREATE TABLE 陳述式中使用的結構描述。

- [在資料表開頭或中間新增欄](#)
- [在資料表尾端新增欄](#)
- [移除資料欄](#)
- [重新命名資料欄](#)
- [重新排序資料欄](#)
- [變更資料欄的資料類型](#)

根據您預期結構描述如何演進，若要繼續使用 Athena 查詢，請選擇相容的資料格式。

假設有一個應用程式會讀取訂單資訊，而此資訊來自於兩種格式的 orders 資料表：CSV 和 Parquet。

以下範例以 Parquet 建立資料表：

```
CREATE EXTERNAL TABLE orders_parquet (  
  `orderkey` int,  
  `orderstatus` string,  
  `totalprice` double,  
  `orderdate` string,  
  `orderpriority` string,  
  `clerk` string,  
  `shippriority` int  
) STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_ parquet/';
```

以下範例以 CSV 建立同一個資料表：

```
CREATE EXTERNAL TABLE orders_csv (  
  `orderkey` int,  
  `orderstatus` string,  
  `totalprice` double,
```

```
`orderdate` string,  
`orderpriority` string,  
`clerk` string,  
`shippriority` int  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_csv/';
```

在以下各節，我們檢閱這些資料表的更新如何影響 Athena 查詢。

### 在資料表開頭或中間新增欄

新增欄是最常見的其中一個結構描述變更。例如，您可以新增欄以新資料來富集資料表。或者，如果現有欄的來源已變更，您可以新增欄，並保留此欄的前一版，以調整依賴這些欄的應用程式。

若要在資料表的開頭或中間新增資料行，並繼續針對現有資料表執行查詢，請使用 AVRO、JSON 和 Parquet 和 ORC (如果其 SerDe 屬性設定為依名稱讀取)。如需相關資訊，請參閱[ORC 和 parquet 的索引存取](#)。

請勿在 CSV 和 TSV 表格的開頭或中間新增欄，因為這些格式取決於排序。如果在這種情況下新增欄，當分割區的結構描述變更時，將會導致結構描述不符錯誤。

下列範例會建立新資料表，該資料表會根據 JSON 資料在資料表中間新增 o\_comment 資料欄。

```
CREATE EXTERNAL TABLE orders_json_column_addition (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_comment` string,  
  `o_totalprice` double,  
  `o_orderdate` string,  
  `o_orderpriority` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_json/';
```

### 在資料表尾端新增欄

如果您以 Athena 支援的任何格式 (如 Parquet、ORC、Avro、JSON、CSV 和 TSV) 建立資料表，您可以使用 ALTER TABLE ADD COLUMNS 陳述式將資料欄新增在現有資料欄之後，但在分割區資料欄之前。

下列範例會在任何分割區資料欄之前的 `comment` 資料表結尾新增 `orders_parquet` 資料欄：

```
ALTER TABLE orders_parquet ADD COLUMNS (comment string)
```

### Note

若要在執行 `ALTER TABLE ADD COLUMNS` 之後查看 Athena 查詢編輯器中的新資料表資料欄，請手動重新整理編輯器中的資料表清單，然後再次展開資料表。

## 移除資料欄

您可能需要從資料表中移除已不再包含資料的欄，或是限制存取這些欄的資料。

- 您可以從 JSON、Avro 及 Parquet 和 ORC (如果依名稱讀取) 的資料表中移除欄。如需相關資訊，請參閱[ORC 和 parquet 的索引存取](#)。
- 如果想要保留您在 Athena 中已建立的資料表，則不建議從 CSV 和 TSV 格式的資料表中移除資料欄。移除資料欄會破壞結構描述，而且您需要重新建立不含已移除的資料欄的資料表。

在這個範例中，從 Parquet 的資料表中移除 `totalprice` 資料欄，並執行查詢。在 Athena 中，預設會依名稱讀取 Parquet，因此我們省略 `SERDEPROPERTIES` 組態 (指定依名稱讀取)。請注意，即使您變更結構描述，以下查詢也會成功：

```
CREATE EXTERNAL TABLE orders_parquet_column_removed (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_orderdate` string,  
  `o_orderpriority` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_comment` string  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_parquet/';
```

## 重新命名資料欄

您可能需要重新命名資料表以更正拼字、讓資料欄名稱更具描述性，或重複使用現有的資料欄以避免資料欄重新排序。

如果您以 CSV 和 TSV 存放資料，或以設定為依索引讀取的 Parquet 和 ORC 存放資料，您可以重新命名資料欄。如需相關資訊，請參閱[ORC 和 parquet 的索引存取](#)。

Athena 依結構描述中的資料欄順序讀取 CSV 和 TSV 資料，並以相同順序傳回資料。它不使用資料欄名稱將資料映射到資料欄，因此您可以重新命名 CSV 或 TSV 格式的資料欄，而不會破壞 Athena 查詢。

重新命名資料欄的一種策略是根據相同的基礎資料建立一個新資料表，但使用新的資料欄名稱。下列範例建立新的 `orders_parquet` 資料表，名為 `orders_parquet_column_renamed`。此範例會變更資料欄 ``o_totalprice`` 名為 ``o_total_price``，然後在 Athena 中執行查詢：

```
CREATE EXTERNAL TABLE orders_parquet_column_renamed (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_total_price` double,  
  `o_orderdate` string,  
  `o_orderpriority` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_comment` string  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_parquet/';
```

在 Parquet 資料表案例中，以下查詢執行，但重新命名的資料欄不會顯示資料，因為資料欄是依名稱存取 (Parquet 的預設值)，而不是依索引：

```
SELECT *  
FROM orders_parquet_column_renamed;
```

CSV 資料表的查詢也類似：

```
CREATE EXTERNAL TABLE orders_csv_column_renamed (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_total_price` double,  
  `o_orderdate` string,  
  `o_orderpriority` string,  
  `o_clerk` string,
```

```
`o_shippriority` int,  
`o_comment` string  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_csv/';
```

在 CSV 資料表案例中，以下查詢執行，且資料會顯示在所有資料欄中，包括已重新命名的資料欄：

```
SELECT *  
FROM orders_csv_column_renamed;
```

## 重新排序資料欄

只有資料格式依名稱讀取的資料表，才能重新排序資料欄，例如 JSON 或 Parquet (預設為依名稱讀取)。如有需要，您也可以將 ORC 變成依名稱讀取。如需相關資訊，請參閱[ORC 和 parquet 的索引存取](#)。

下列範例會建立新的資料表，其中資料欄的順序各不相同：

```
CREATE EXTERNAL TABLE orders_parquet_columns_reordered (  
  `o_comment` string,  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderpriority` string,  
  `o_orderstatus` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_orderdate` string  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_parquet/';
```

## 變更資料欄的資料類型

當現有類型無法再保留所需資訊量時，您可能想要使用不同的資料欄類型。例如，ID 資料欄的值可能超過 INT 資料類型的大小，而且需要使用 BIGINT 資料類型。

當您規劃使用不同的資料類型做為資料欄時，請考慮下列幾點：

- 在大多數的案例中，您無法直接變更資料欄的資料類型。相反，您要重新建立 Athena 資料表，並使用新的資料類型定義資料欄。
- 只有特定資料類型才可讀取為其他資料類型。有關可如此處理的資料類型，請參閱本節中的資料表。

- 對於 Parquet 和 ORC 的資料，如果資料表未分割，則您無法對一個資料欄使用不同的資料類型。
- 對於 Parquet 和 ORC 的分割資料表，分割區的資料欄類型可能不同於另一個分割區的資料欄類型，可能的話，Athena 會 CAST 到所需的類型。如需相關資訊，請參閱[避免含有分割區的資料表發生結構描述不符的錯誤](#)。
- 對於使用 [LazySimpleSerDe](#) only 建立的資料表，可以使用 ALTER TABLE REPLACE COLUMNS 陳述式將現有的資料行取代為不同的資料類型，但是您想要保留的所有現有資料行也必須在陳述式中重新定義，否則將會刪除這些資料行。如需詳細資訊，請參閱 [ALTER TABLE REPLACE COLUMNS](#)。
- 僅對於 Apache Iceberg 資料表，您可以使用 [ALTER TABLE CHANGE COLUMN](#) 陳述式來變更資料欄的資料類型。ALTER TABLE REPLACE COLUMNS 不支援 Iceberg 資料表。如需詳細資訊，請參閱 [逐步發展的 Iceberg 資料表結構描述](#)。

### ⚠ Important

在執行資料類型轉換之前，強烈建議您測試和驗證您的查詢。如果 Athena 無法使用目標資料類型，CREATE TABLE 查詢可能會失敗。

下表列出了會被視為其他資料類型的資料類型：

#### 相容的資料類型

原始資料類型	可用的目標資料類型
STRING	BYTE, TINYINT, SMALLINT, INT, BIGINT
BYTE	TINYINT, SMALLINT, INT, BIGINT
TINYINT	SMALLINT, INT, BIGINT
SMALLINT	INT, BIGINT
INT	BIGINT
FLOAT	DOUBLE

下列範例會使用原始 orders\_json 資料表的 CREATE TABLE 陳述式來建立名為 orders\_json\_bigint 的新資料表。新資料表使用 BIGINT (而不是 INT) 做為 `o\_shippriority` 資料欄的資料類型。

```
CREATE EXTERNAL TABLE orders_json_bigint (
  `o_orderkey` int,
  `o_custkey` int,
  `o_orderstatus` string,
  `o_totalprice` double,
  `o_orderdate` string,
  `o_orderpriority` string,
  `o_clerk` string,
  `o_shippriority` BIGINT
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_json';
```

在資料類型變更之前，以下查詢執行成功，類似於原始 SELECT 查詢：

```
Select * from orders_json
LIMIT 10;
```

## 資料表和分割區中的更新

在 Athena 中，資料表及其分割區必須使用相同的資料格式，但其結構描述可能不同。當您建立新的分割區時，該分區通常會繼承資料表的結構描述。隨著時間經過，結構描述可能開始不同。原因包括：

- 如果資料表的結構描述變更，分割區的結構描述不會更新，而無法與資料表的結構描述保持同步。
- AWS Glue 爬行者程式可讓您探索具有不同綱要之分割區中的資料。這表示，如果您在 Athena 中建立資料表 AWS Glue，在爬行者程式完成處理之後，資料表及其分割區的結構定義可能會有所不同。
- 如果您直接使用 AWS API 新增分割區。

如果含有分割區的資料表符合下列限制條件，Athena 會成功處理資料表。如果不符合這些限制條件，Athena 會發出 HIVE\_PARTITION\_SCHEMA\_MISMATCH 錯誤。

- 每個分割區的結構描述與資料表的結構描述相容。
- 資料表的資料格式允許您想要執行的更新類型：新增、刪除、重新排序欄，或變更欄的資料類型。

例如，對於 CSV 和 TSV 格式，您可以重新命名欄、在資料表尾端新增欄，以及變更欄的資料類型 (如果類型相容)，但您無法移除欄。對於其他格式，您可以新增或移除欄，或將欄變更為另一種資料類型 (如果類型相容)。如需相關資訊，請參閱[摘要：Athena 中的更新和資料格式](#)。

## 避免含有分割區的資料表發生結構描述不符的錯誤

在開始執行查詢時，Athena 會檢查資料表與分割區之間的每個資料欄資料類型是否相容，以驗證資料表的結構描述。

- 對於 Parquet 和 ORC 資料儲存類型，Athena 倚賴資料欄名稱，並使用它們來進行以資料欄名稱為基礎的結構描述驗證。這樣可避免含有分割區的 Parquet 和 ORC 資料表發生 HIVE\_PARTITION\_SCHEMA\_MISMATCH 錯誤。(如果 SerDe 屬性設置為按名稱訪問索引，則對 ORC 來說是如此：`orc.column.index.access=FALSE`。實木複合地板默認情況下按名稱讀取索引)。
- 對於 CSV、JSON 和 Avro，Athena 使用以索引為基礎的結構描述驗證。這表示如果您遇到結構描述不符的錯誤，則應該捨棄導致結構描述不符的分割區，並重新建立分割區，讓 Athena 在查詢它時不會失敗。

Athena 會比較資料表的結構描述與分割區結構描述。如果您使用爬行者程式在 Athena 中以 CSV、JSON 和 AVRO 建立資料表，則在 AWS Glue 爬行者程式完成處理之後，表格及其分割區的結構定義可能會有所不同。如果資料表的結構描述與分割區結構描述不相符，則您在 Athena 中的查詢會失敗，因為發生類似如下的結構描述驗證錯誤：`'crawler_test.click_avro' is declared as type 'string', but partition 'partition_0=2017-01-17' declared column 'col68' as type 'double'.` ('crawler\_test.click\_avro' 宣告為類型 'string'，但分割區 'partition\_0=2017-01-17' 將資料欄 'col68' 宣告為類型 'double'。)

這類錯誤的典型解決方法是捨棄造成錯誤的分割區，然後重新建立。如需詳細資訊，請參閱 [ALTER TABLE DROP PARTITION](#) 及 [ALTER TABLE ADD PARTITION](#)。

## 查詢陣列

Amazon Athena 可讓您建立陣列、串連陣列、將陣列轉換為不同的資料類型，然後篩選、展開和排序陣列。

### 主題

- [建立陣列](#)
- [串連字串和陣列](#)
- [轉換陣列資料類型](#)
- [尋找長度](#)
- [存取陣列元素](#)
- [將巢狀陣列扁平化](#)

- [從子查詢建立陣列](#)
- [篩選陣列](#)
- [排序陣列](#)
- [搭配陣列使用彙總函數](#)
- [將陣列轉換為字串](#)
- [使用陣列建立映射](#)
- [查詢具有複雜類型和巢狀結構的陣列](#)

## 建立陣列

若要在 Athena 中建置陣列常值，請使用 ARRAY 關鍵字，後面接著方括弧 [ ] 並包含陣列元素 (以逗號分隔)。

### 範例

此查詢會建立具有四個元素的陣列。

```
SELECT ARRAY [1,2,3,4] AS items
```

它會傳回：

```
+-----+
| items  |
+-----+
| [1,2,3,4] |
+-----+
```

此查詢會建立兩個陣列。

```
SELECT ARRAY[ ARRAY[1,2], ARRAY[3,4] ] AS items
```

它會傳回：

```
+-----+
| items          |
+-----+
| [[1, 2], [3, 4]] |
+-----+
```

若要從選取的相容類型資料欄建立陣列，請使用查詢，如此範例所示：

```
WITH
dataset AS (
  SELECT 1 AS x, 2 AS y, 3 AS z
)
SELECT ARRAY [x,y,z] AS items FROM dataset
```

此查詢會傳回：

```
+-----+
| items  |
+-----+
| [1,2,3] |
+-----+
```

在以下範例中，選取了兩個陣列，並以歡迎訊息傳回。

```
WITH
dataset AS (
  SELECT
    ARRAY ['hello', 'amazon', 'athena'] AS words,
    ARRAY ['hi', 'alexa'] AS alexa
)
SELECT ARRAY[words, alexa] AS welcome_msg
FROM dataset
```

此查詢會傳回：

```
+-----+
| welcome_msg          |
+-----+
| [[hello, amazon, athena], [hi, alexa]] |
+-----+
```

若要建立一系列的金鑰值組，請使用 MAP 運算子，其會使用一系列的金鑰，接著是一系列的值，如此範例所示：

```
SELECT ARRAY[
  MAP(ARRAY['first', 'last', 'age'],ARRAY['Bob', 'Smith', '40']),
  MAP(ARRAY['first', 'last', 'age'],ARRAY['Jane', 'Doe', '30']),
```

```
MAP(ARRAY['first', 'last', 'age'],ARRAY['Billy', 'Smith', '8'])
] AS people
```

此查詢會傳回：

```
+-----+
+
| people
|
+-----+
+
| [{last=Smith, first=Bob, age=40}, {last=Doe, first=Jane, age=30}, {last=Smith,
| first=Billy, age=8}] |
+-----+
+
```

## 串連字串和陣列

### 串連字串

若要串連兩個字串，您可以使用雙縱線 `||` 運算子，如下列範例所示。

```
SELECT 'This' || ' is' || ' a' || ' test.' AS Concatenated_String
```

此查詢會傳回：

#	Concatenated_String
1	This is a test.

您可以使用 `concat()` 函數達到相同的效果。

```
SELECT concat('This', ' is', ' a', ' test.') AS Concatenated_String
```

此查詢會傳回：

#	Concatenated_String
1	This is a test.

您可以使用 `concat_ws()` 函式將字串與第一個引數中指定的分隔符號串連。

```
SELECT concat_ws(' ', 'This', 'is', 'a', 'test.') as Concatenated_String
```

此查詢會傳回：

#	Concatenated_String
1	This is a test.

若要使用點串連字串資料類型的兩個資料欄，請使用雙引號引用這兩個資料欄，並以單引號括住該點做為硬式編碼字串。如果資料欄不是字串資料類型，您可以先使用 `CAST("column_name" as VARCHAR)` 轉換資料欄類型。

```
SELECT "col1" || '.' || "col2" as Concatenated_String
FROM my_table
```

此查詢會傳回：

#	Concatenated_String
1	<i>col1_string_value</i> . <i>col2_string_value</i>

## 串連陣列

您可以使用相同的技巧串連陣列。

若要串連多個陣列，請使用雙縱線 `||` 運算子。

```
SELECT ARRAY [4,5] || ARRAY[ ARRAY[1,2], ARRAY[3,4] ] AS items
```

此查詢會傳回：

#	items
1	[[4, 5], [1, 2], [3, 4]]

若要將多個陣列合併到單一陣列，請使用雙縱線運算子或 `concat()` 函數。

```
WITH
dataset AS (
  SELECT
    ARRAY ['Hello', 'Amazon', 'Athena'] AS words,
    ARRAY ['Hi', 'Alexa'] AS alexa
)
SELECT concat(words, alexa) AS welcome_msg
FROM dataset
```

此查詢會傳回：

#	welcome_msg
1	[Hello, Amazon, Athena, Hi, Alexa]

如需有關 `concat()` 其他字串函數的詳細資訊，請參閱 Trino 文件中的 [String functions and operators](#) (字串函數和運算子)。

## 轉換陣列資料類型

若要將陣列中的資料轉換為支援的資料類型，請以 `CAST(value AS type)` 的格式使用 `CAST` 運算子。Athena 支援所有原生 Presto 資料類型。

```
SELECT
  ARRAY [CAST(4 AS VARCHAR), CAST(5 AS VARCHAR)]
AS items
```

此查詢會傳回：

```
+-----+
| items |
+-----+
| [4,5] |
+-----+
```

以鍵值組元素建立兩個陣列，將它們轉換為 JSON 並串連在一起，如這個範例所示：

```
SELECT
```

```

ARRAY[CAST(MAP(ARRAY['a1', 'a2', 'a3'], ARRAY[1, 2, 3]) AS JSON)] ||
ARRAY[CAST(MAP(ARRAY['b1', 'b2', 'b3'], ARRAY[4, 5, 6]) AS JSON)]
AS items

```

此查詢會傳回：

```

+-----+
| items |
+-----+
| [{"a1":1,"a2":2,"a3":3}, {"b1":4,"b2":5,"b3":6}] |
+-----+

```

## 尋找長度

`cardinality` 函數會傳回陣列的長度，如此範例所示：

```

SELECT cardinality(ARRAY[1,2,3,4]) AS item_count

```

此查詢會傳回：

```

+-----+
| item_count |
+-----+
| 4          |
+-----+

```

## 存取陣列元素

若要存取陣列元素，使用 `[]` 運算子，並以 1 指定第一個元素，2 指定第二個元素，依此類推，如此範例所示：

```

WITH dataset AS (
SELECT
  ARRAY[CAST(MAP(ARRAY['a1', 'a2', 'a3'], ARRAY[1, 2, 3]) AS JSON)] ||
  ARRAY[CAST(MAP(ARRAY['b1', 'b2', 'b3'], ARRAY[4, 5, 6]) AS JSON)]
AS items )
SELECT items[1] AS item FROM dataset

```

此查詢會傳回：

```

+-----+

```

```
| item |
+-----+
| {"a1":1,"a2":2,"a3":3} |
+-----+
```

若要存取指定位置的陣列元素 (稱為索引位置)，請使用 `element_at()` 函數，並指定陣列名稱和索引位置：

- 如果索引大於 0，`element_at()` 會傳回您指定的元素，從陣列的開頭開始往結尾算。它的行為和 `[]` 運算子一樣。
- 如果索引小於 0，`element_at()` 傳回的元素是從陣列的結尾開始往開頭算。

以下查詢會建立陣列 `words`，並從中選取第一個元素 `hello` 做為 `first_word`，選取第二個元素 `amazon` (從陣列結尾往前算) 做為 `middle_word`，選取第三個元素 `athena` 做為 `last_word`。

```
WITH dataset AS (
  SELECT ARRAY ['hello', 'amazon', 'athena'] AS words
)
SELECT
  element_at(words, 1) AS first_word,
  element_at(words, -2) AS middle_word,
  element_at(words, cardinality(words)) AS last_word
FROM dataset
```

此查詢會傳回：

```
+-----+
| first_word | middle_word | last_word |
+-----+
| hello      | amazon      | athena    |
+-----+
```

## 將巢狀陣列扁平化

使用巢狀陣列時，您經常需要將巢狀陣列元素展開至單一陣列，或將陣列展開至多個資料列。

### 範例

若要將巢狀陣列的元素扁平化至值的單一陣列，請使用 `flatten` 函數。此查詢會為陣列中的每個元素傳回一個資料列。

```
SELECT flatten(ARRAY[ ARRAY[1,2], ARRAY[3,4] ]) AS items
```

此查詢會傳回：

```
+-----+
| items  |
+-----+
| [1,2,3,4] |
+-----+
```

若要將陣列扁平化至多個資料列，請使用 CROSS JOIN 結合 UNNEST 運算子，如此範例所示：

```
WITH dataset AS (
  SELECT
    'engineering' as department,
    ARRAY['Sharon', 'John', 'Bob', 'Sally'] as users
)
SELECT department, names FROM dataset
CROSS JOIN UNNEST(users) as t(names)
```

此查詢會傳回：

```
+-----+
| department | names |
+-----+
| engineering | Sharon |
+-----+
| engineering | John  |
+-----+
| engineering | Bob   |
+-----+
| engineering | Sally |
+-----+
```

若要將一系列的金鑰值組扁平化，請將選取的金鑰調換至資料欄，如此範例所示：

```
WITH
dataset AS (
  SELECT
    'engineering' as department,
    ARRAY[
```

```

    MAP(ARRAY['first', 'last', 'age'],ARRAY['Bob', 'Smith', '40']),
    MAP(ARRAY['first', 'last', 'age'],ARRAY['Jane', 'Doe', '30']),
    MAP(ARRAY['first', 'last', 'age'],ARRAY['Billy', 'Smith', '8'])
  ] AS people
)
SELECT names['first'] AS
  first_name,
  names['last'] AS last_name,
  department FROM dataset
CROSS JOIN UNNEST(people) AS t(names)

```

此查詢會傳回：

```

+-----+
| first_name | last_name | department |
+-----+
| Bob        | Smith    | engineering |
| Jane       | Doe      | engineering |
| Billy      | Smith    | engineering |
+-----+

```

從員工清單中選擇具有最高組合評分的員工。UNNEST 可用於 FROM 子句，而不需先有 CROSS JOIN，因為它是預設的聯結運算子，因此已隱含具備。

```

WITH
dataset AS (
  SELECT ARRAY[
    CAST(ROW('Sally', 'engineering', ARRAY[1,2,3,4]) AS ROW(name VARCHAR, department
  VARCHAR, scores ARRAY(INTEGER))),
    CAST(ROW('John', 'finance', ARRAY[7,8,9]) AS ROW(name VARCHAR, department VARCHAR,
  scores ARRAY(INTEGER))),
    CAST(ROW('Amy', 'devops', ARRAY[12,13,14,15]) AS ROW(name VARCHAR, department
  VARCHAR, scores ARRAY(INTEGER)))
  ] AS users
),
users AS (
  SELECT person, score
  FROM
    dataset,
    UNNEST(dataset.users) AS t(person),
    UNNEST(person.scores) AS t(score)
)
SELECT person.name, person.department, SUM(score) AS total_score FROM users

```

```
GROUP BY (person.name, person.department)
ORDER BY (total_score) DESC
LIMIT 1
```

此查詢會傳回：

```
+-----+
| name | department | total_score |
+-----+
| Amy  | devops     | 54          |
+-----+
```

從員工清單中選擇具有最高個別評分的員工。

```
WITH
dataset AS (
  SELECT ARRAY[
    CAST(ROW('Sally', 'engineering', ARRAY[1,2,3,4]) AS ROW(name VARCHAR, department
  VARCHAR, scores ARRAY(INTEGER))),
    CAST(ROW('John', 'finance', ARRAY[7,8,9]) AS ROW(name VARCHAR, department VARCHAR,
  scores ARRAY(INTEGER))),
    CAST(ROW('Amy', 'devops', ARRAY[12,13,14,15]) AS ROW(name VARCHAR, department
  VARCHAR, scores ARRAY(INTEGER)))
  ] AS users
),
users AS (
  SELECT person, score
  FROM
    dataset,
    UNNEST(dataset.users) AS t(person),
    UNNEST(person.scores) AS t(score)
)
SELECT person.name, score FROM users
ORDER BY (score) DESC
LIMIT 1
```

此查詢會傳回：

```
+-----+
| name | score |
+-----+
| Amy  | 15    |
```

```
+-----+
```

## 考量與限制

如果 UNNEST 在查詢中的一個或多個陣列上使用，且其中一個陣列是 NULL，則查詢不會傳回任何資料列。如果 UNNEST 在一個空字串陣列上使用，則會傳回空字串。

例如，在下列查詢中，由於第二個陣列為 Null，因此查詢不會傳回任何資料列。

```
SELECT
  col1,
  col2
FROM UNNEST (ARRAY ['apples','oranges','lemons']) AS t(col1)
CROSS JOIN UNNEST (ARRAY []) AS t(col2)
```

在下面的範例中，第二個數組修改為包含一個空字串。針對每次資料列，查詢會傳回 col1 中的值，以及 col2 中的值的空字串。第二個陣列中的空字串為必要，以便傳回第一個陣列中的值。

```
SELECT
  col1,
  col2
FROM UNNEST (ARRAY ['apples','oranges','lemons']) AS t(col1)
CROSS JOIN UNNEST (ARRAY ['']) AS t(col2)
```

## 從子查詢建立陣列

從資料列的集合建立陣列。

```
WITH
dataset AS (
  SELECT ARRAY[1,2,3,4,5] AS items
)
SELECT array_agg(i) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
```

此查詢會傳回：

```
+-----+
| array_items |
+-----+
```

```
| [1, 2, 3, 4, 5] |
+-----+
```

若要從一組資料列建立一系列的多個唯一值，請使用 `distinct` 關鍵字。

```
WITH
dataset AS (
  SELECT ARRAY [1,2,2,3,3,4,5] AS items
)
SELECT array_agg(distinct i) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
```

此查詢會傳回下列結果。請注意，無法保證順序。

```
+-----+
| array_items |
+-----+
| [1, 2, 3, 4, 5] |
+-----+
```

如需有關使用 `array_agg` 函數的詳細資訊，請參閱 Trino 文件中的 [Aggregate functions](#) (彙總函數)。

## 篩選陣列

如果資料列的集合符合篩選條件，則從資料列的集合建立陣列。

```
WITH
dataset AS (
  SELECT ARRAY[1,2,3,4,5] AS items
)
SELECT array_agg(i) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
WHERE i > 3
```

此查詢會傳回：

```
+-----+
| array_items |
+-----+
```

```
| [4, 5] |
+-----+
```

根據其元素是否包含特定的值 (例如 2) 來篩選陣列，如此範例所示：

```
WITH
dataset AS (
  SELECT ARRAY
  [
    ARRAY[1,2,3,4],
    ARRAY[5,6,7,8],
    ARRAY[9,0]
  ] AS items
)
SELECT i AS array_items FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
WHERE contains(i, 2)
```

此查詢會傳回：

```
+-----+
| array_items |
+-----+
| [1, 2, 3, 4] |
+-----+
```

## filter 函數

```
filter(ARRAY [list_of_values], boolean_function)
```

您可以在 ARRAY 表達式上使用 filter 函數建立一個新的陣列，該陣列是 *list\_of\_values* 中項目的子集，其 *boolean\_function* 為 true。filter 函數在您無法使用 *UNNEST* 函數的情況下十分有用。

以下範例會篩選陣列 [1,0,5,-1] 中大於零的值。

```
SELECT filter(ARRAY [1,0,5,-1], x -> x>0)
```

結果

```
[1,5]
```

以下範例會篩選陣列 [-1, NULL, 10, NULL] 中的非零值。

```
SELECT filter(ARRAY [-1, NULL, 10, NULL], q -> q IS NOT NULL)
```

結果

[-1,10]

排序陣列

若要從一組資料行建立已排序的唯一值陣列，您可以使用 [array\\_sort](#) 函數，如下列範例所示。

```
WITH
dataset AS (
  SELECT ARRAY[3,1,2,5,2,3,6,3,4,5] AS items
)
SELECT array_sort(array_agg(distinct i)) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
```

此查詢會傳回：

```
+-----+
| array_items |
+-----+
| [1, 2, 3, 4, 5, 6] |
+-----+
```

如需將陣列展開成多個資料列的相關資訊，請參閱[將巢狀陣列扁平化](#)。

搭配陣列使用彙總函數

- 若要在陣列中新增值，請使用 SUM，如下列範例所示。
- 若要彙總陣列中的多個資料列，請使用 array\_agg。如需相關資訊，請參閱[從子查詢建立陣列](#)。

#### Note

自 Athena 引擎版本 2 開始，彙總函數支援 ORDER BY。

```

WITH
dataset AS (
  SELECT ARRAY
  [
    ARRAY[1,2,3,4],
    ARRAY[5,6,7,8],
    ARRAY[9,0]
  ] AS items
),
item AS (
  SELECT i AS array_items
  FROM dataset, UNNEST(items) AS t(i)
)
SELECT array_items, sum(val) AS total
FROM item, UNNEST(array_items) AS t(val)
GROUP BY array_items;

```

在最後一個 SELECT 陳述式中，與其使用 `sum()` 和 `UNNEST`，您可以使用 `reduce()` 來減少處理時間和資料傳輸，如下列範例所示。

```

WITH
dataset AS (
  SELECT ARRAY
  [
    ARRAY[1,2,3,4],
    ARRAY[5,6,7,8],
    ARRAY[9,0]
  ] AS items
),
item AS (
  SELECT i AS array_items
  FROM dataset, UNNEST(items) AS t(i)
)
SELECT array_items, reduce(array_items, 0, (s, x) -> s + x, s -> s) AS total
FROM item;

```

此查詢會傳回下列結果。無法保證傳回結果的順序。

```

+-----+
| array_items | total |
+-----+
| [1, 2, 3, 4] | 10    |

```

```
| [5, 6, 7, 8] | 26 |
| [9, 0]       | 9  |
+-----+
```

## 將陣列轉換為字串

若要將陣列為單一字串，請使用 `array_join` 函數。以下個別範例會建立名為 `dataset` 的資料表，其中包含名為 `words` 的別名陣列。查詢會使用 `array_join` 讓陣列元素加入 `words`、以空格分隔元素，然後在名為 `welcome_msg` 的別名資料欄中傳回產生的字串。

```
WITH
dataset AS (
  SELECT ARRAY ['hello', 'amazon', 'athena'] AS words
)
SELECT array_join(words, ' ') AS welcome_msg
FROM dataset
```

此查詢會傳回：

```
+-----+
| welcome_msg |
+-----+
| hello amazon athena |
+-----+
```

## 使用陣列建立映射

映射是由 Athena 中可用的資料類型組成的鍵值組。若要建立映射，請使用 `MAP` 運算子並傳入兩個陣列：第一個是欄 (金鑰) 名稱，第二個是值。陣列的所有值必須是相同類型。如果任何映射值陣列元素必須是不同類型，您稍後可以轉換它們。

### 範例

此範例從資料集選取使用者。它使用 `MAP` 運算子並傳入兩個陣列。第一個陣列包含欄名稱的值，例如 "first"、"last" 和 "age"。第二個陣列包含每個欄的值，例如 "Bob"、"Smith"、"35"。

```
WITH dataset AS (
  SELECT MAP(
    ARRAY['first', 'last', 'age'],
    ARRAY['Bob', 'Smith', '35']
  ) AS user
)
```

```
SELECT user FROM dataset
```

此查詢會傳回：

```
+-----+
| user          |
+-----+
| {last=Smith, first=Bob, age=35} |
+-----+
```

您可以選取欄位名稱，後面接著 [key\_name]，以擷取 Map 值，如以下範例所示：

```
WITH dataset AS (
  SELECT MAP(
    ARRAY['first', 'last', 'age'],
    ARRAY['Bob', 'Smith', '35']
  ) AS user
)
SELECT user['first'] AS first_name FROM dataset
```

此查詢會傳回：

```
+-----+
| first_name |
+-----+
| Bob        |
+-----+
```

## 查詢具有複雜類型和巢狀結構的陣列

您的來源資料通常包含具有複雜資料類型和巢狀結構的陣列。本節中的範例展示如何使用 Athena 查詢來變更元素的資料類型、在陣列內尋找元素，以及尋找關鍵字。

- [建立 ROW](#)
- [使用 CAST 變更陣列中的欄位名稱](#)
- [使用 . 標記法篩選陣列](#)
- [篩選含有巢狀值的陣列](#)
- [使用 UNNEST 篩選陣列](#)
- [使用 regexp\\_like 在陣列中尋找關鍵字](#)

## 建立 ROW

### Note

本節中的範例使用 ROW 當為方法來建立要使用的範例資料。當您在 Athena 內查詢資料表時，您不需要建立 ROW 資料類型，因為已經從您的資料來源建立它們。當您使用 CREATE\_TABLE 時，Athena 會在其中定義 STRUCT，在其中填入資料，然後替您為資料集的每個資料行建立 ROW 資料類型。基礎 ROW 資料類型由任何受支援 SQL 資料類型的具名欄位組成。

```
WITH dataset AS (
  SELECT
    ROW('Bob', 38) AS users
)
SELECT * FROM dataset
```

此查詢會傳回：

```
+-----+
| users          |
+-----+
| {field0=Bob, field1=38} |
+-----+
```

使用 **CAST** 變更陣列中的欄位名稱

若要變更陣列中含有 ROW 值的欄位名稱，您可以 CAST ROW 宣告：

```
WITH dataset AS (
  SELECT
    CAST(
      ROW('Bob', 38) AS ROW(name VARCHAR, age INTEGER)
    ) AS users
)
SELECT * FROM dataset
```

此查詢會傳回：

```
+-----+
| users          |
+-----+
```

```
| {NAME=Bob, AGE=38} |
+-----+
```

### Note

在上述範例中，您將 name 宣告為 VARCHAR，因為這是它在 Presto 中的類型。如果您在 CREATE TABLE 陳述式內宣告這個 STRUCT，請使用 String 類型，因為 Hive 將此資料類型定義為 String。

## 使用 . 標記法篩選陣列

在下列範例中，使用點 . 標記法，從 AWS CloudTrail 日誌表格的 userIdentity 欄選取 accountId 欄位。如需詳細資訊，請參閱[查詢 AWS CloudTrail 日誌](#)。

```
SELECT
  CAST(useridentity.accountid AS bigint) as newid
FROM cloudtrail_logs
LIMIT 2;
```

此查詢會傳回：

```
+-----+
| newid      |
+-----+
| 112233445566 |
+-----+
| 998877665544 |
+-----+
```

若要查詢值陣列，請發出此查詢：

```
WITH dataset AS (
  SELECT ARRAY[
    CAST(ROW('Bob', 38) AS ROW(name VARCHAR, age INTEGER)),
    CAST(ROW('Alice', 35) AS ROW(name VARCHAR, age INTEGER)),
    CAST(ROW('Jane', 27) AS ROW(name VARCHAR, age INTEGER))
  ] AS users
)
SELECT * FROM dataset
```

它會傳回此結果：

```
+-----+
| users                                     |
+-----+
| [{NAME=Bob, AGE=38}, {NAME=Alice, AGE=35}, {NAME=Jane, AGE=27}] |
+-----+
```

篩選含有巢狀值的陣列

大型陣列通常包含巢狀結構，您必須能夠篩選或搜尋其中的值。

若要為包含巢狀 BOOLEAN 值的值陣列定義資料集，請發出此查詢：

```
WITH dataset AS (
  SELECT
    CAST(
      ROW('aws.amazon.com', ROW(true)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ) AS sites
)
SELECT * FROM dataset
```

它會傳回此結果：

```
+-----+
| sites                                     |
+-----+
| {HOSTNAME=aws.amazon.com, FLAGGEDACTIVITY={ISNEW=true}} |
+-----+
```

接著，若要篩選並存取該元素的 BOOLEAN 值，請繼續使用點 . 標記法。

```
WITH dataset AS (
  SELECT
    CAST(
      ROW('aws.amazon.com', ROW(true)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ) AS sites
)
SELECT sites.hostname, sites.flaggedactivity.isnew
FROM dataset
```

此查詢會選取巢狀欄位，並傳回此結果：

```
+-----+
| hostname      | isnew |
+-----+
| aws.amazon.com | true  |
+-----+
```

### 使用 UNNEST 篩選陣列

若要依其中一個子元素來篩選含有巢狀結構的陣列，請以 UNNEST 運算子發出查詢。如需 UNNEST 的詳細資訊，請參閱[展開巢狀陣列](#)。

例如，此查詢會在資料集尋找網站的主機名稱。

```
WITH dataset AS (
  SELECT ARRAY[
    CAST(
      ROW('aws.amazon.com', ROW(true)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ),
    CAST(
      ROW('news.cnn.com', ROW(false)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ),
    CAST(
      ROW('netflix.com', ROW(false)) AS ROW(hostname VARCHAR, flaggedActivity ROW(isNew
BOOLEAN))
    )
  ] as items
)
SELECT sites.hostname, sites.flaggedActivity.isNew
FROM dataset, UNNEST(items) t(sites)
WHERE sites.flaggedActivity.isNew = true
```

它會傳回：

```
+-----+
| hostname      | isnew |
+-----+
| aws.amazon.com | true  |
+-----+
```

## 使用 `regexp_like` 在陣列中尋找關鍵字

以下範例說明如何使用 `regexp_like` 函數，在資料集搜尋陣列內某個元素內的關鍵字。它接受規則表達式模式或以管線 (|) 分隔的詞彙清單作為輸入來評估，評估模式，然後判斷指定的字串是否包含它。

規則表達式模式必須存在於字串內，但不一定要符合它。若要比對整個字串，請在模式開頭加上 `^`，在尾端加上 `&`，以括住模式，例如 `^pattern&`。

假設有一個包含主機名稱的網站陣列，還有一個 `flaggedActivity` 元素。這個元素包含 ARRAY，其中包含幾個 MAP 元素，各列出不同的熱門關鍵字及熱門度計數。假設您想在此陣列中的 MAP 內尋找特定的關鍵字。

為了依特定關鍵字在此資料集搜尋網站，我們使用 `regexp_like` 而不是類似的 SQL LIKE 運算子，因為以 `regexp_like` 搜尋大量關鍵字更有效率。

### Example 範例 1：使用 `regexp_like`

這個範例中的查詢使用 `regexp_like` 函數來搜尋詞彙 `'politics|bigdata'`，並於陣列內的值中找到：

```
WITH dataset AS (
  SELECT ARRAY[
    CAST(
      ROW('aws.amazon.com', ROW(ARRAY[
        MAP(ARRAY['term', 'count'], ARRAY['bigdata', '10']),
        MAP(ARRAY['term', 'count'], ARRAY['serverless', '50']),
        MAP(ARRAY['term', 'count'], ARRAY['analytics', '82']),
        MAP(ARRAY['term', 'count'], ARRAY['iot', '74'])
      ])
    ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
  VARCHAR))) )
),
  CAST(
    ROW('news.cnn.com', ROW(ARRAY[
      MAP(ARRAY['term', 'count'], ARRAY['politics', '241']),
      MAP(ARRAY['term', 'count'], ARRAY['technology', '211']),
      MAP(ARRAY['term', 'count'], ARRAY['serverless', '25']),
      MAP(ARRAY['term', 'count'], ARRAY['iot', '170'])
    ])
    ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
  VARCHAR))) )
),
  CAST(
```

```

ROW('netflix.com', ROW(ARRAY[
  MAP(ARRAY['term', 'count'], ARRAY['cartoons', '1020']),
  MAP(ARRAY['term', 'count'], ARRAY['house of cards', '112042']),
  MAP(ARRAY['term', 'count'], ARRAY['orange is the new black', '342']),
  MAP(ARRAY['term', 'count'], ARRAY['iot', '4'])
]))
) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR))) ))
)
] AS items
),
sites AS (
  SELECT sites.hostname, sites.flaggedactivity
  FROM dataset, UNNEST(items) t(sites)
)
SELECT hostname
FROM sites, UNNEST(sites.flaggedActivity.flags) t(flags)
WHERE regexp_like(flags['term'], 'politics|bigdata')
GROUP BY (hostname)

```

此查詢會傳回兩個網站：

```

+-----+
| hostname      |
+-----+
| aws.amazon.com |
+-----+
| news.cnn.com  |
+-----+

```

## Example 範例 2：使用 **regexp\_like**

下列範例中的查詢對於 **regexp\_like** 函數中符合搜尋詞彙的網站，合計其熱門度總分，然後依最高到最低排序。

```

WITH dataset AS (
  SELECT ARRAY[
    CAST(
      ROW('aws.amazon.com', ROW(ARRAY[
        MAP(ARRAY['term', 'count'], ARRAY['bigdata', '10']),
        MAP(ARRAY['term', 'count'], ARRAY['serverless', '50']),
        MAP(ARRAY['term', 'count'], ARRAY['analytics', '82']),
        MAP(ARRAY['term', 'count'], ARRAY['iot', '74'])
      ]))
    ]

```

```

    ])
  ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR)) ))
),
CAST(
  ROW('news.cnn.com', ROW(ARRAY[
    MAP(ARRAY['term', 'count'], ARRAY['politics', '241']),
    MAP(ARRAY['term', 'count'], ARRAY['technology', '211']),
    MAP(ARRAY['term', 'count'], ARRAY['serverless', '25']),
    MAP(ARRAY['term', 'count'], ARRAY['iot', '170'])
  ])
  ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR)) ))
),
CAST(
  ROW('netflix.com', ROW(ARRAY[
    MAP(ARRAY['term', 'count'], ARRAY['cartoons', '1020']),
    MAP(ARRAY['term', 'count'], ARRAY['house of cards', '112042']),
    MAP(ARRAY['term', 'count'], ARRAY['orange is the new black', '342']),
    MAP(ARRAY['term', 'count'], ARRAY['iot', '4'])
  ])
  ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR)) ))
)
] AS items
),
sites AS (
  SELECT sites.hostname, sites.flaggedactivity
  FROM dataset, UNNEST(items) t(sites)
)
SELECT hostname, array_agg(flags['term']) AS terms, SUM(CAST(flags['count'] AS
INTEGER)) AS total
FROM sites, UNNEST(sites.flaggedActivity.flags) t(flags)
WHERE regexp_like(flags['term'], 'politics|bigdata')
GROUP BY (hostname)
ORDER BY total DESC

```

此查詢會傳回兩個網站：

```

+-----+
| hostname      | terms    | total  |
+-----+-----+
| news.cnn.com  | politics | 241    |

```

```
+-----+-----+
| aws.amazon.com | bigdata | 10 |
+-----+-----+
```

## 查詢地理空間資料

地理空間資料包含識別符來指定物體的地理位置。這類資料的例子包括天氣報告、地圖方向、含地理位置的推文、商店位置，以及飛機航線。地理空間資料在商業分析、報告和預測方面扮演重要角色。

地理空間識別符 (例如經緯度) 可讓您將任何郵寄地址轉換為一組地理座標。

### 主題

- [什麼是地理空間查詢？](#)
- [輸入資料格式和幾何資料類型](#)
- [支援的地理空間函數](#)
- [範例：地理空間查詢](#)

## 什麼是地理空間查詢？

地理空間查詢為 Athena 中支援的特殊 SQL 查詢類型。它們與非空間 SQL 查詢的差別在於下列各方面：

- 使用以下特殊化幾何資料類型：point、line、multiline、polygon 和 multipolygon。
- 表示幾何資料類型之間的關係，例如 distance、equals、crosses、touches、overlaps、disjoint 等等。

您可以在 Athena 中使用地理空間查詢，執行這些和其他類似操作：

- 尋找兩點之間的距離。
- 檢查是否一個區域 (多邊形) 包含其他區域。
- 檢查一個線條是否與其他線條或多邊形交錯或碰觸。

例如，若要在 Athena 中從瑞尼爾山之地理座標的類型 double 值中取得 point 幾何資料類型，請如下列範例所示，使用 ST\_Point (longitude, latitude) 地理空間函數。

```
ST_Point(-121.7602, 46.8527)
```

## 輸入資料格式和幾何資料類型

若要在 Athena 使用地理空間函數，請以 WKT 格式輸入資料，或使用 Hive JSON SerDe。您也可以使用 Athena 中支援的幾何資料類型。

### 輸入資料格式

為處理地理空間查詢，Athena 支援以下資料格式的輸入資料：

- WKT (已知文字)。在 Athena 中，WKT 會以 `varchar(x)` 或 `string` 資料類型表示。
- JSON 編碼的地理空間資料。為了剖析 JSON 檔案與地理空間資料，並為它們建立資料表，Athena 使用 [Hive JSON SerDe](#)。如需 SerDe 在 Athena 使用此功能的詳細資訊，請參閱[JSON SerDe 程式庫](#)。

### 幾何資料類型

為處理地理空間查詢，Athena 支援以下特殊幾何資料類型：

- `point`
- `line`
- `polygon`
- `multiline`
- `multipolygon`

## 支援的地理空間函數

Athena 中可用的地理空間函數取決於您使用的引擎版本。

- 如需有關 Athena 引擎版本 3 中地理空間函數的資訊，請參閱 Trino 文件中的 [Geospatial functions](#) (地理空間函數)。
- 如需從 Athena 引擎版本 2 開始採用的函數名稱變更和新函數之清單，請參閱[Athena 引擎版本 2 中地理空間函數名稱變更和新函數](#)。

如需有關 Athena 引擎版本控制的資訊，請參閱[Athena 引擎版本控制](#)。

### 主題

- [Athena 引擎版本 3 中的地理空間函數](#)

- [Athena 引擎版本 2 中的地理空間函數](#)

### Athena 引擎版本 3 中的地理空間函數

如需有關 Athena 引擎版本 3 中地理空間函數的資訊，請參閱 Trino 文件中的 [Geospatial functions](#) (地理空間函數)。

### Athena 引擎版本 2 中的地理空間函數

此主題列出 Athena 引擎版本 2 開始支援之 ESRI 地理空間函數。如需有關 Athena 引擎版本的資訊，請參閱 [Athena 引擎版本控制](#)。

### Athena 引擎版本 2 中的變更

- 某些函數的輸入和輸出類型已經變更。最值得注意的是，VARBINARY 類型不再直接支援輸入。如需詳細資訊，請參閱 [對空間函數的變更](#)。
- 部分地理空間函數的名稱已有所變動。如需詳細資訊，請參閱 [Athena 引擎版本 2 中地理空間函數名稱變更](#)。
- 已新增新函數。如需詳細資訊，請參閱 [Athena 引擎版本 2 中的新地理空間函數](#)。

Athena 支援以下類型的地理空間函數：

- [建構子函數](#)
- [地理空間關係函數](#)
- [操作函數](#)
- [存取子函數](#)
- [彙總函數](#)
- [Bing tile 函數](#)

### 建構子函數

使用建構子函數來取得 point、line 或 polygon 幾何資料類型的二進位表示法。對於以已知文字 (WKT) 表達的幾何資料，您也可以使用這些函數將二進位資料轉換為文字並取得二進位值。

### ST\_AsBinary(geometry)

傳回包含指定幾何之 WKB 表示法的 varbinary 資料類型。範例：

```
SELECT ST_AsBinary(ST_Point(-158.54, 61.56))
```

### ST\_AsText(geometry)

將每個指定的[幾何資料類型](#)轉換成文字。傳回 varchar 資料類型中的值，它是幾何資料類型的 WKT 表示法。範例：

```
SELECT ST_AsText(ST_Point(-158.54, 61.56))
```

### ST\_GeomAsLegacyBinary(geometry)

透過指定的幾何傳回舊版 varbinary。範例：

```
SELECT ST_GeomAsLegacyBinary(ST_Point(-158.54, 61.56))
```

### ST\_GeometryFromText(varchar)

將 WKT 格式的文字轉換為幾何資料類型。傳回幾何資料類型中的值。範例：

```
SELECT ST_GeometryFromText(ST_AsText(ST_Point(1, 2)))
```

### ST\_GeomFromBinary(varbinary)

透過 WKB 表示法傳回幾何類型物件。範例：

```
SELECT ST_GeomFromBinary(ST_AsBinary(ST_Point(-158.54, 61.56)))
```

### ST\_GeomFromLegacyBinary(varbinary)

透過舊版 varbinary 類型傳回幾何類型物件。範例：

```
SELECT ST_GeomFromLegacyBinary(ST_GeomAsLegacyBinary(ST_Point(-158.54, 61.56)))
```

### ST\_LineFromText(varchar)

傳回[幾何資料類型](#)為 line 的值。範例：

```
SELECT ST_Line('linestring(1 1, 2 2, 3 3)')
```

## ST\_LineString(array(point))

傳回根據點幾何類型的陣列形成的 LineString 幾何類型。如果指定的陣列中有少於兩個非空點，則會傳回空的 LineString。如果陣列中的任何元素為 null、空或與前一個元素相同，則擲出例外狀況。傳回的幾何可能並不簡單。根據指定的輸入，傳回的幾何可能會自我相交或包含重複的頂點。範例：

```
SELECT ST_LineString(ARRAY[ST_Point(-158.54, 61.56), ST_Point(-158.55, 61.56)])
```

## ST\_MultiPoint(array(point))

傳回根據指定點形成的 MultiPoint 幾何物件。如果指定的陣列為空，則傳回 null。如果陣列中的任何元素為 null 或空，則擲出例外狀況。傳回的幾何可能並不簡單，如果指定的陣列有重複的項目，則可能包含重複的點。範例：

```
SELECT ST_MultiPoint(ARRAY[ST_Point(-158.54, 61.56), ST_Point(-158.55, 61.56)])
```

## ST\_Point(double, double)

傳回幾何類型 point 物件。對於此函數的輸入資料值，請使用幾何值，例如世界橫麥卡托投影 (UTM) 笛卡兒座標系統，或地理圖單位 (經度和緯度) (十進制度數)。經度和緯度值使用全球測量系統，也稱為 WGS 1984 或 EPSG:4326。WGS 1984 是全球定位系統 (GPS) 採用的座標系統。

例如，在以下標記法中，地圖座標以經度和緯度指定，而值 .072284 (緩衝距離) 是以角度單位 (十進制度數) 指定：

```
SELECT ST_Buffer(ST_Point(-74.006801, 40.705220), .072284)
```

語法：

```
SELECT ST_Point(longitude, latitude) FROM earthquakes LIMIT 1
```

以下範例使用特定的經度和緯度座標：

```
SELECT ST_Point(-158.54, 61.56)
FROM earthquakes
LIMIT 1
```

下一個範例使用特定的經度和緯度座標：

```
SELECT ST_Point(-74.006801, 40.705220)
```

下列範例使用 `ST_AsText` 函數從 WKT 取得幾何：

```
SELECT ST_AsText(ST_Point(-74.006801, 40.705220)) AS WKT
```

### **ST\_Polygon(varchar)**

順時針使用提供的座標序列 (從左到右)，傳回[幾何資料類型](#) polygon。從 Athena 引擎版本 2 開始，僅接受多邊形作為輸入。範例：

```
SELECT ST_Polygon('polygon ((1 1, 1 4, 4 4, 4 1))')
```

### **to\_geometry(sphericalGeography)**

從指定的球形地理物件傳回幾何物件。範例：

```
SELECT to_geometry(to_spherical_geography(ST_Point(-158.54, 61.56)))
```

### **to\_spherical\_geography(geometry)**

從指定的幾何傳回球形地理物件。使用此函數，可將幾何物件轉換為地球半徑球體上的球形地理物件。此函數只能用於在 2D 空間或這類幾何的 GEOMETRYCOLLECTION 中定義的 POINT、MULTIPOINT、LINESTRING、MULTILINESTRING、POLYGON 以及 MULTIPOLYGON 幾何。對於指定幾何的每個點，此函數會確認 `point.x` 在 `[-180.0, 180.0]` 內部，而 `point.y` 在 `[-90.0, 90.0]` 內部。此函數使用這些點作為經度和緯度來構建 sphericalGeography 結果的形狀。

範例：

```
SELECT to_spherical_geography(ST_Point(-158.54, 61.56))
```

### 地理空間關係函數

以下函數會表達您指定作為輸入的兩個不同幾何之間的關係，並傳回類型 `boolean` 的結果。您指定幾何對組的順序很重要：第一個幾何值稱為左幾何、第二個幾何值稱為右幾何。

這些函數會傳回：

- TRUE，如果且只有在滿足函數所描述的關係時。
- FALSE，如果且只有在不滿足函數所描述的關係時。

### **ST\_Contains(geometry, geometry)**

傳回 TRUE，如果且只有在左幾何包含右幾何時。範例：

```
SELECT ST_Contains('POLYGON((0 2,1 1,0 -1,0 2))', 'POLYGON((-1 3,2 1,0 -3,-1 3))')
```

```
SELECT ST_Contains('POLYGON((0 2,1 1,0 -1,0 2))', ST_Point(0, 0))
```

```
SELECT ST_Contains(ST_GeometryFromText('POLYGON((0 2,1 1,0 -1,0 2))'),  
ST_GeometryFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'))
```

### **ST\_Crosses(geometry, geometry)**

傳回 TRUE，如果且只有在左幾何與右幾何交叉時。範例：

```
SELECT ST_Crosses(ST_Line('linestring(1 1, 2 2)'), ST_Line('linestring(0 1, 2 2)'))
```

### **ST\_Disjoint(geometry, geometry)**

傳回 TRUE，如果且只有在左幾何和右幾何的交集是空的時。範例：

```
SELECT ST_Disjoint(ST_Line('linestring(0 0, 0 1)'), ST_Line('linestring(1 1, 1 0)'))
```

### **ST\_Equals(geometry, geometry)**

傳回 TRUE，如果且只有在左幾何等於右幾何時。範例：

```
SELECT ST_Equals(ST_Line('linestring( 0 0, 1 1)'), ST_Line('linestring(1 3, 2 2)'))
```

### **ST\_Intersects(geometry, geometry)**

傳回 TRUE，如果且只有在左幾何與右幾何交集時。範例：

```
SELECT ST_Intersects(ST_Line('linestring(8 7, 7 8)'), ST_Polygon('polygon((1 1, 4 1, 4  
4, 1 4))'))
```

## ST\_Overlaps(geometry, geometry)

傳回 TRUE，如果且只有在左幾何與右幾何重疊時。範例：

```
SELECT ST_Overlaps(ST_Polygon('polygon((2 0, 2 1, 3 1))'), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## ST\_Relate(geometry, geometry, varchar)

傳回 TRUE，如果且只有在左幾何與右幾何具有指定的維度延伸九交模型 ([DE-9IM](#)) 關係時。第三個 (varchar) 輸入採用此關係。範例：

```
SELECT ST_Relate(ST_Line('linestring(0 0, 3 3)'), ST_Line('linestring(1 1, 4 4)'), 'T*****')
```

## ST\_Touches(geometry, geometry)

傳回 TRUE，如果且只有在左幾何接觸右幾何時。

範例：

```
SELECT ST_Touches(ST_Point(8, 8), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## ST\_Within(geometry, geometry)

傳回 TRUE，如果且只有在左幾何在右幾何內時。

範例：

```
SELECT ST_Within(ST_Point(8, 8), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## 操作函數

使用操作函數來對幾何資料類型值執行操作。例如，您可以取得單一幾何資料類型的邊界；兩個幾何資料類型之間差異的交集；左右幾何之間的差異，其中的每個有相同幾何資料類型；或是環繞特定幾何資料類型的外部緩衝區或環。

## geometry\_union(array(geometry))

傳回代表指定幾何之點集合聯集的幾何。範例：

```
SELECT geometry_union(ARRAY[ST_Point(-158.54, 61.56), ST_Point(-158.55, 61.56)])
```

## ST\_Boundary(geometry)

作為其中一個幾何資料類型的輸入，並傳回 boundary 幾何資料類型。

範例：

```
SELECT ST_Boundary(ST_Line('linestring(0 1, 1 0)'))
```

```
SELECT ST_Boundary(ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## ST\_Buffer(geometry, double)

做為其中一個幾何資料類型的輸入，例如點、線條、多邊形、多線條或多個多邊形，以及距離做為類型 double。傳回由指定距離 (或半徑) 緩衝的幾何資料類型。範例：

```
SELECT ST_Buffer(ST_Point(1, 2), 2.0)
```

在下列範例中，地圖座標以經度和緯度指定，而值 .072284 (緩衝距離) 是以角度單位 (十進制度數) 指定：

```
SELECT ST_Buffer(ST_Point(-74.006801, 40.705220), .072284)
```

## ST\_Difference(geometry, geometry)

傳回左幾何和右幾何之間差異的幾何。範例：

```
SELECT ST_AsText(ST_Difference(ST_Polygon('polygon((0 0, 0 10, 10 10, 10 0))'),  
ST_Polygon('polygon((0 0, 0 5, 5 5, 5 0))')))
```

## ST\_Envelope(geometry)

接受 line、polygon、multiline 和 multipolygon 幾何資料類型做為輸入。不支援 point 幾何資料類型。以幾何形式傳回信封，其中的信封為環繞指定幾何資料類型的矩形。範例：

```
SELECT ST_Envelope(ST_Line('linestring(0 1, 1 0)'))
```

```
SELECT ST_Envelope(ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## ST\_EnvelopeAsPts(geometry)

傳回兩個點的陣列，這兩個點代表幾何邊界矩形多邊形的左下角和右上角。在指定的幾何為空白時，傳回 null。範例：

```
SELECT ST_EnvelopeAsPts(ST_Point(-158.54, 61.56))
```

## ST\_ExteriorRing(geometry)

傳回輸入類型 polygon 的外部環幾何。從 Athena 引擎版本 2 開始，多邊形是唯一可接受的輸入幾何形狀。範例：

```
SELECT ST_ExteriorRing(ST_Polygon(1,1, 1,4, 4,1))
```

```
SELECT ST_ExteriorRing(ST_Polygon('polygon ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))'))
```

## ST\_Intersection(geometry, geometry)

傳回左幾何和右幾何之交集的幾何。範例：

```
SELECT ST_Intersection(ST_Point(1,1), ST_Point(1,1))
```

```
SELECT ST_Intersection(ST_Line('linestring(0 1, 1 0)'), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

```
SELECT ST_AsText(ST_Intersection(ST_Polygon('polygon((2 0, 2 3, 3 0))'), ST_Polygon('polygon((1 1, 4 1, 4 4, 1 4))')))
```

## ST\_SymDifference(geometry, geometry)

傳回左幾何和右幾何之間幾何對稱差異的幾何。範例：

```
SELECT ST_AsText(ST_SymDifference(ST_Line('linestring(0 2, 2 2)'), ST_Line('linestring(1 2, 3 2)')))
```

## ST\_Union(geometry, geometry)

傳回代表指定幾何之點集合聯集的幾何資料類型。範例：

```
SELECT ST_Union(ST_Point(-158.54, 61.56),ST_LineString(array[ST_Point(1,2),
  ST_Point(3,4)]))
```

## 存取子函數

存取子函數對於從不同的 geometry 資料類型取得類型 varchar、bigint 或 double 中的值很有用，其中的 geometry 是 Athena 中支援的任何幾何資料類型：point、line、polygon、multiline 和 multipolygon。例如，您可以取得 polygon 幾何資料類型的一個區域，指定幾何資料類型的最大和最小 X 和 Y 值，取得 line 的長度，或是接收指定幾何資料類型中的點數量。

### **geometry\_invalid\_reason(geometry)**

以 varchar 資料類型傳回指定幾何無效或不簡單的原因。如果指定的幾何無效或不簡單，則會傳回無效的原因。如果指定的幾何有效且簡單，則傳回 null。範例：

```
SELECT geometry_invalid_reason(ST_Point(-158.54, 61.56))
```

### **great\_circle\_distance(latitude1, longitude1, latitude2, longitude2)**

以雙精度浮點數的形式傳回地球表面上兩點間的大圓距離 (以公里為單位)。範例：

```
SELECT great_circle_distance(36.12, -86.67, 33.94, -118.40)
```

### **line\_locate\_point(lineString, point)**

傳回介於 0 和 1 的雙精度浮點數，其表示指定單行字串上離指定點最近點的位置，作為 2D 單行總長度的一小部分。

如果指定的單行字串或點為空或 null，則傳回 null。範例：

```
SELECT line_locate_point(ST_GeometryFromText('LINESTRING (0 0, 0 1)'), ST_Point(0,
  0.2))
```

### **simplify\_geometry(geometry, double)**

使用 [Ramer-douglas-peucker 演算法](#) 傳回幾何圖形資料類型，此類型為指定幾何的簡化版本。避免建立無效的衍生幾何 (特別是多邊形)。範例：

```
SELECT simplify_geometry(ST_GeometryFromText('POLYGON ((1 0, 2 1, 3 1, 3 1, 4 1, 1 0))'), 1.5)
```

### ST\_Area(geometry)

做為幾何資料類型的輸入，並傳回類型中的區域double。範例：

```
SELECT ST_Area(ST_Polygon('polygon((1 1, 4 1, 4 4, 1 4))'))
```

### ST\_Centroid(geometry)

作為[幾何資料類型](#) polygon 的輸入，並傳回 point 幾何資料類型，其是多邊形信封的中心。範例：

```
SELECT ST_Centroid(ST_GeometryFromText('polygon ((0 0, 3 6, 6 0, 0 0))'))
```

```
SELECT ST_AsText(ST_Centroid(ST_Envelope(ST_GeometryFromText('POINT (53 27)'))))
```

### ST\_ConvexHull(geometry)

傳回幾何資料類型，其是用指定輸入括住所有幾何的最小凸面幾何。範例：

```
SELECT ST_ConvexHull(ST_Point(-158.54, 61.56))
```

### ST\_CoordDim(geometry)

作為其中一個支援的[幾何資料類型](#)的輸入，並傳回 tinyint 類型中座標元件的計數。範例：

```
SELECT ST_CoordDim(ST_Point(1.5,2.5))
```

### ST\_Dimension(geometry)

做為其中一個支援的[幾何資料類型](#)輸入，並傳回 tinyint 類型中幾何的空間維度。範例：

```
SELECT ST_Dimension(ST_Polygon('polygon((1 1, 4 1, 4 4, 1 4))'))
```

### ST\_Distance(geometry, geometry)

根據空間參考，以預計單位傳回雙精度浮點數，其包含兩個幾何之間的 2D 最短笛卡兒距離。從 Athena 引擎版本 2 開始，如果其中一個輸入為空幾何形狀，則傳回 null。範例：

```
SELECT ST_Distance(ST_Point(0.0,0.0), ST_Point(3.0,4.0))
```

## ST\_Distance(sphericalGeography, sphericalGeography)

以雙精度浮點數的形式傳回兩個球形地理點之間的大圓距離 (以公尺為單位)。範例：

```
SELECT ST_Distance(to_spherical_geography(ST_Point(61.56,
-86.67)),to_spherical_geography(ST_Point(61.56, -86.68)))
```

## ST\_EndPoint(geometry)

傳回 point 幾何資料類型中 line 幾何資料類型的最後一個點。範例：

```
SELECT ST_EndPoint(ST_Line('linestring(0 2, 2 2)'))
```

## ST\_Geometries(geometry)

傳回指定集合中的幾何陣列。如果指定的幾何不是多幾何，則傳回單一元素陣列。如果指定的幾何為空，則傳回 null。

例如，考量到 MultiLineString 物件，ST\_Geometries 會建立 LineString 物件的陣列。考量到 GeometryCollection 物件，ST\_Geometries 會傳回其組成的未平面化陣列。範例：

```
SELECT ST_Geometries(GEOMETRYCOLLECTION(MULTIPOINT(0 0, 1 1),
GEOMETRYCOLLECTION(MULTILINESTRING((2 2, 3 3)))))
```

結果：

```
array[MULTIPOINT(0 0, 1 1),GEOMETRYCOLLECTION(MULTILINESTRING((2 2, 3 3)))]
```

## ST\_GeometryN(geometry, index)

以幾何資料類型的形式傳回位於指定整數索引處的幾何元素。索引從 1 開始。如果指定的幾何是幾何的集合 (例如 GEOMETRYCOLLECTION 或 MULTI\* 物件)，則會傳回指定索引處的幾何。如果指定的索引小於 1 或大於集合中元素的總數，則傳回 null。若要尋找元素的總數，請使用 [ST\\_NumGeometries](#)。單數幾何 (例如，POINT、LINESTRING 或 POLYGON) 被視為一個元素的集合。空幾何會被視為空集合。範例：

```
SELECT ST_GeometryN(ST_Point(-158.54, 61.56),1)
```

## ST\_GeometryType(geometry)

以 varchar 的形式傳回幾何的類型。範例：

```
SELECT ST_GeometryType(ST_Point(-158.54, 61.56))
```

## ST\_InteriorRingN(geometry, index)

傳回指定索引處的內部環元素 (索引從 1 開始)。如果給定的索引小於 1 或大於指定幾何中內部環總數，則傳回 null。如果指定的幾何不是多邊形，則會擲出錯誤。若要尋找元素的總數，請使用 [ST\\_NumInteriorRing](#)。範例：

```
SELECT ST_InteriorRingN(st_polygon('polygon ((0 0, 1 0, 1 1, 0 1, 0 0))'),1)
```

## ST\_InteriorRings(geometry)

傳回指定幾何中找到之所有內部環的幾何陣列，或如果多邊形沒有內部環，則傳回空陣列。如果指定的幾何為空，則傳回 null。如果指定的幾何不是多邊形，則會擲出錯誤。範例：

```
SELECT ST_InteriorRings(st_polygon('polygon ((0 0, 1 0, 1 1, 0 1, 0 0))'))
```

## ST\_IsClosed(geometry)

做為僅輸入用的 line 和 multiline [幾何資料類型](#)。傳回 TRUE (類型 boolean)，如果且只有在線條為關閉時。範例：

```
SELECT ST_IsClosed(ST_Line('linestring(0 2, 2 2)'))
```

## ST\_IsEmpty(geometry)

做為僅輸入用的 line 和 multiline [幾何資料類型](#)。如果且只有在指定的幾何為空時，才會傳回 TRUE (類型 boolean)，換句話說，即 line 的開始與結束值共同存在時。範例：

```
SELECT ST_IsEmpty(ST_Point(1.5, 2.5))
```

## ST\_IsRing(geometry)

傳回 TRUE (類型 boolean)，如果且只有在 line 類型為關閉且簡單時。範例：

```
SELECT ST_IsRing(ST_Line('linestring(0 2, 2 2)'))
```

### ST\_IsSimple(geometry)

如果指定的幾何沒有異常幾何點 (例如，自我相交或自我相切)，則傳回 true。若要判斷幾何不簡單的原因，請使用 [geometry\\_invalid\\_reason\(\)](#)。範例：

```
SELECT ST_IsSimple(ST_LineString(array[ST_Point(1,2), ST_Point(3,4)]))
```

### ST\_IsValid(geometry)

如果且只有在指定幾何格式正確時傳回 true。若要判斷幾何格式不正確的原因，請使用 [geometry\\_invalid\\_reason\(\)](#)。範例：

```
SELECT ST_IsValid(ST_Point(61.56, -86.68))
```

### ST\_Length(geometry)

傳回類型 double 中 line 的長度。範例：

```
SELECT ST_Length(ST_Line('linestring(0 2, 2 2)'))
```

### ST\_NumGeometries(geometry)

以整數的形式傳回集合中幾何的數量。如果幾何是幾何的集合 (例如 GEOMETRYCOLLECTION 或 MULTI\* 物件)，則會傳回幾何的數量。單一幾何傳回 1；空幾何則傳回 0。GEOMETRYCOLLECTION 物件中的空幾何會計為一個幾何。例如，下列范例會評估為 1：

```
ST_NumGeometries(ST_GeometryFromText('GEOMETRYCOLLECTION(MULTIPOINT EMPTY)'))
```

### ST\_NumInteriorRing(geometry)

傳回類型 bigint 中 polygon 幾何中的內部環數量。範例：

```
SELECT ST_NumInteriorRing(ST_Polygon('polygon ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))'))
```

### ST\_NumPoints(geometry)

傳回在類型 bigint 中幾何中的點數量。範例：

```
SELECT ST_NumPoints(ST_Point(1.5, 2.5))
```

### **ST\_PointN(lineString, index)**

以點幾何資料類型傳回指定整數索引處指定單行字串的頂點。索引從 1 開始。如果給定的索引小於 1 或大於集合中元素的總數，則傳回 null。若要尋找元素的總數，請使用 [ST\\_NumPoints](#)。範例：

```
SELECT ST_PointN(ST_LineString(array[ST_Point(1,2), ST_Point(3,4)]),1)
```

### **ST\_Points(geometry)**

從指定的單行字串幾何物件傳回點的陣列。範例：

```
SELECT ST_Points(ST_LineString(array[ST_Point(1,2), ST_Point(3,4)]))
```

### **ST\_StartPoint(geometry)**

傳回 point 幾何資料類型中 line 幾何資料類型的第一個點。範例：

```
SELECT ST_StartPoint(ST_Line('linestring(0 2, 2 2)'))
```

### **ST\_X(point)**

傳回類型 double 中點的 X 座標。範例：

```
SELECT ST_X(ST_Point(1.5, 2.5))
```

### **ST\_XMax(geometry)**

傳回類型 double 中幾何的最大 X 座標。範例：

```
SELECT ST_XMax(ST_Line('linestring(0 2, 2 2)'))
```

### **ST\_XMin(geometry)**

傳回類型 double 中幾何的最小 X 座標。範例：

```
SELECT ST_XMin(ST_Line('linestring(0 2, 2 2)'))
```

## ST\_Y(point)

傳回類型 `double` 中點的 Y 座標。範例：

```
SELECT ST_Y(ST_Point(1.5, 2.5))
```

## ST\_YMax(geometry)

傳回類型 `double` 中幾何的最大 Y 座標。範例：

```
SELECT ST_YMax(ST_Line('linestring(0 2, 2 2)'))
```

## ST\_YMin(geometry)

傳回類型 `double` 中幾何的最小 Y 座標。範例：

```
SELECT ST_YMin(ST_Line('linestring(0 2, 2 2)'))
```

## 彙總函數

### convex\_hull\_agg(geometry)

傳回最小凸面幾何，所有以輸入形式傳遞的幾何會以此最小凸面幾何括起。

### geometry\_union\_agg(geometry)

傳回幾何，該幾何表示以輸入形式傳遞之所有幾何的點集合聯集。

## Bing tile 函數

下列函數在 Microsoft [Bing maps tile system](#) 中的幾何和圖標之間進行轉換。

### bing\_tile(x, y, zoom\_level)

從整數座標 `x`、`y` 和指定縮放比例傳回 Bing 圖標物件。縮放比例必須是介於 1 到 23 之間的整數。範例：

```
SELECT bing_tile(10, 20, 12)
```

### bing\_tile(quadKey)

從 `quadkey` 傳回 Bing 圖標物件。範例：

```
SELECT bing_tile(bing_tile_quadkey(bing_tile(10, 20, 12)))
```

### **bing\_tile\_at(latitude, longitude, zoom\_level)**

以指定的緯度、經度和縮放比例傳回 Bing 圖標物件。緯度必須介於 -85.05112878 和 85.05112878 之間。經度必須介於 -180 和 180 之間。latitude 和 longitude 值必須為 double，而 zoom\_level 必須為整數。範例：

```
SELECT bing_tile_at(37.431944, -122.166111, 12)
```

### **bing\_tiles\_around(latitude, longitude, zoom\_level)**

以指定縮放比例傳回環繞指定緯度和經度點的 Bing 圖標陣列。範例：

```
SELECT bing_tiles_around(47.265511, -122.465691, 14)
```

### **bing\_tiles\_around(latitude, longitude, zoom\_level, radius\_in\_km)**

以指定縮放比例傳回 Bing 圖標陣列。該陣列包含最小 Bing 圖標組，其涵蓋圍繞指定緯度和經度指定半徑的圓圈 (以公里為單位)。latitude、longitude 以及 radius\_in\_km 值為 double；縮放比例為 integer。範例：

```
SELECT bing_tiles_around(37.8475, 112.596667, 10, .5)
```

### **bing\_tile\_coordinates(tile)**

傳回指定 Bing 圖標的 x 和 y 座標。範例：

```
SELECT bing_tile_coordinates(bing_tile_at(37.431944, -122.166111, 12))
```

### **bing\_tile\_polygon(tile)**

傳回指定 Bing 圖標的多邊形表示法。範例：

```
SELECT bing_tile_polygon(bing_tile_at(47.265511, -122.465691, 4))
```

### **bing\_tile\_quadkey(tile)**

傳回指定 Bing 圖標的 quadkey。範例：

```
SELECT bing_tile_quadkey(bing_tile(52, 143, 10))
```

### bing\_tile\_zoom\_level(tile)

以整數形式傳回指定 Bing 圖標的縮放比例。範例：

```
SELECT bing_tile_zoom_level(bing_tile(52, 143, 10))
```

### geometry\_to\_bing\_tiles(geometry, zoom\_level)

以指定縮放比例傳回完整涵蓋指定幾何的最小 Bing 圖標組。支援從 1 到 23 的縮放比例。範例：

```
SELECT geometry_to_bing_tiles(ST_Point(61.56, 58.54), 10)
```

## Athena 引擎版本 2 中地理空間函數名稱變更和新函數

本節列出了在 Athena 引擎版本 2 中，新增的地理空間函數名稱和地理空間函數的變更。如需有關從 Athena 引擎版本 2 開始採用的其他變更的資訊，請參閱[Athena 引擎版本 2](#)。

如需有關 Athena 引擎版本控制的資訊，請參閱[Athena 引擎版本控制](#)。

### Athena 引擎版本 2 中地理空間函數名稱變更

下列函數的名稱已變更。在某些情況下，輸入和輸出類型也發生了變更。如需詳細資訊，請造訪相應的連結。

舊函數名稱	從 Athena 引擎版本 2 開始採用的函數名稱
st_coordinate_dimension	<a href="#">ST_CoordDim</a>
st_end_point	<a href="#">ST_EndPoint</a>
st_exterior_ring	<a href="#">ST_ExteriorRing</a>
st_interior_ring_number	<a href="#">ST_NumInteriorRing</a>
st_geometry_from_text	<a href="#">ST_GeometryFromText</a>
st_is_closed	<a href="#">ST_IsClosed</a>

舊函數名稱	從 Athena 引擎版本 2 開始採用的函數名稱
st_is_empty	<a href="#">ST_IsEmpty</a>
st_is_ring	<a href="#">ST_IsRing</a>
st_max_x	<a href="#">ST_XMax</a>
st_max_y	<a href="#">ST_YMax</a>
st_min_x	<a href="#">ST_XMin</a>
st_min_y	<a href="#">ST_YMin</a>
st_point_number	<a href="#">ST_NumPoints</a>
st_start_point	<a href="#">ST_StartPoint</a>
st_symmetric_difference	<a href="#">ST_SymDifference</a>

## Athena 引擎版本 2 中的新地理空間函數

下列地理空間函數是從 Athena 引擎版本 2 開始採用的新函數。如需詳細資訊，請造訪相應的連結。

### 建構子函數

- [ST\\_AsBinary](#)
- [ST\\_GeomAsLegacyBinary](#)
- [ST\\_GeomFromBinary](#)
- [ST\\_GeomFromLegacyBinary](#)
- [ST\\_LineString](#)
- [ST\\_MultiPoint](#)
- [to\\_geometry](#)
- [to\\_spherical\\_geography](#)

### 操作函數

- [geometry\\_union](#)

- [ST\\_EnvelopeAsPts](#)
- [ST\\_Union](#)

### 存取子函數

- [geometry\\_invalid\\_reason](#)
- [great\\_circle\\_distance](#)
- [line\\_locate\\_point](#)
- [simplify\\_geometry](#)
- [ST\\_ConvexHull](#)
- [ST\\_Distance \(球形地理位置\)](#)
- [ST\\_Geometries](#)
- [ST\\_GeometryN](#)
- [ST\\_GeometryType](#)
- [ST\\_N InteriorRing](#)
- [ST\\_InteriorRings](#)
- [ST\\_IsSimple](#)
- [ST\\_IsValid](#)
- [ST\\_NumGeometries](#)
- [ST\\_PointN](#)
- [ST\\_Points](#)

### 彙總函數

- [convex\\_hull\\_agg](#)
- [geometry\\_union\\_agg](#)

### Bing tile 函數

- [bing\\_tile](#)
- [bing\\_tile \(quadkey\)](#)

- [bing\\_tile\\_at](#)
- [bing\\_tiles\\_around](#)
- [bing\\_tiles\\_around \(半徑\)](#)
- [bing\\_tile\\_coordinates](#)
- [bing\\_tile\\_polygon](#)
- [bing\\_tile\\_quadkey](#)
- [bing\\_tile\\_zoom\\_level](#)
- [geometry\\_to\\_bing\\_tiles](#)

## 範例：地理空間查詢

本主題中的範例會根據可用的範例資料建立兩個資料表，GitHub 並根據資料查詢資料表。範例資料位於下列檔案中，僅供說明之用且不保證準確無誤：

- [earthquakes.csv](#) – 列出在加州發生的地震。範例 earthquakes 資料表使用此資料的欄位。
- [california-counties.json](#) – 以符合 ESRI 規範的 GeoJSON 格式列出加州的郡資料。該資料包含許多欄位，例如 AREA、PERIMETER、STATE、COUNTY 和 NAME，但範例 counties 資料表僅使用兩個欄位：Name (字串) 和 BoundaryShape (二進位)。

### Note

Athena 使用 `com.esri.json.hadoop.EnclosedEsriJsonInputFormat`，將 JSON 資料轉換為地理空間二進位格式。

以下程式碼範例會建立名為 earthquakes 的資料表：

```
CREATE external TABLE earthquakes
(
  earthquake_date string,
  latitude double,
  longitude double,
  depth double,
  magnitude double,
  magtype string,
  mbstations string,
  gap string,
```

```

distance string,
rms string,
source string,
eventid string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION 's3://DOC-EXAMPLE-BUCKET/my-query-log/csv/';

```

以下程式碼範例會建立名為 `counties` 的資料表：

```

CREATE external TABLE IF NOT EXISTS counties
(
  Name string,
  BoundaryShape binary
)
ROW FORMAT SERDE 'com.esri.hadoop.hive.serde.EsriJsonSerDe'
STORED AS INPUTFORMAT 'com.esri.json.hadoop.EnclosedEsriJsonInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/my-query-log/json/';

```

下列範例查詢使用 `counties` 和 `earthquake` 資料表上的 `CROSS JOIN` 函數。該範例使用 `ST_CONTAINS` 查詢其界限包括地震位置的郡，該地震位置是使用 `ST_POINT` 指定。該查詢會將這類郡依名稱分組、依計數排序，並依遞減順序傳回。

#### Note

從 Athena 引擎版本 2 中開始，`ST_CONTAINS` 這類函數不再支援 `VARBINARY` 類型作為輸入。基於這個原因，該範例使用 [ST\\_GeomFromLegacyBinary\(varbinary\)](#) 函數，將 `boundaryshape` 二進位值轉換為幾何。如需詳細資訊，請參閱 [Athena 引擎版本 2 參考中的對空間函數的變更](#)。

```

SELECT counties.name,
       COUNT(*) cnt
FROM counties
CROSS JOIN earthquakes
WHERE ST_CONTAINS (ST_GeomFromLegacyBinary(counties.boundaryshape),
                  ST_POINT(earthquakes.longitude, earthquakes.latitude))
GROUP BY counties.name
ORDER BY cnt DESC

```

此查詢會傳回：

```
+-----+
| name          | cnt |
+-----+
| Kern          | 36  |
+-----+
| San Bernardino | 35  |
+-----+
| Imperial      | 28  |
+-----+
| Inyo          | 20  |
+-----+
| Los Angeles   | 18  |
+-----+
| Riverside     | 14  |
+-----+
| Monterey      | 14  |
+-----+
| Santa Clara   | 12  |
+-----+
| San Benito    | 11  |
+-----+
| Fresno        | 11  |
+-----+
| San Diego     | 7   |
+-----+
| Santa Cruz    | 5   |
+-----+
| Ventura       | 3   |
+-----+
| San Luis Obispo | 3   |
+-----+
| Orange        | 2   |
+-----+
| San Mateo     | 1   |
+-----+
```

## 其他資源

如需地理空間查詢的其他範例，請參閱下列部落格文章：

- [使用 UDF 和擴展亞 Amazon Athena 的地理空間查詢 AWS Lambda](#)

- [使用 Amazon 雅典娜和亞馬遜將 200 多年的全球氣候資料視覺化 QuickSight。](#)
- [查詢 OpenStreetMap Amazon Athena](#)

## 查詢 JSON

Amazon Athena 可讓您查詢 JSON 編碼的資料、從巢狀 JSON 擷取資料、搜尋值，以及尋找 JSON 陣列的長度和大小。若要瞭解在 Athena 查詢 JSON 資料的基本概念，請參考下列行星資料範例：

```
{name:"Mercury",distanceFromSun:0.39,orbitalPeriod:0.24,dayLength:58.65}
{name:"Venus",distanceFromSun:0.72,orbitalPeriod:0.62,dayLength:243.02}
{name:"Earth",distanceFromSun:1.00,orbitalPeriod:1.00,dayLength:1.00}
{name:"Mars",distanceFromSun:1.52,orbitalPeriod:1.88,dayLength:1.03}
```

請注意每個記錄（本質上是表中的每一行）是如何在單獨的行上。若要查詢此 JSON 資料，您可以使用如下所示的 CREATE TABLE 陳述式：

```
CREATE EXTERNAL TABLE `planets_json` (
  `name` string,
  `distancefromsun` double,
  `orbitalperiod` double,
  `daylength` double)
ROW FORMAT SERDE
  'org.openx.data.jsonserde.JsonSerDe'
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/json/'
```

若要查詢資料，請使用類似下列範例的簡單 SELECT 陳述式。

```
SELECT * FROM planets_json
```

查詢結果如下所示。

#	name	距離太陽的距離	奧爾巴時期	日長
1	水星	0.39	0.24	58.65

#	name	距離太陽的距離	奧爾巴時期	日長
2	金星	0.72	0.62	243.02
3	地球	1.0	1.0	1.0
4	火星	1.52	1.88	1.03

請注意CREATE TABLE陳述式如何使用[OpenX SerDe](#)，這會要求每個JSON記錄都在不同的行上。如果JSON採用漂亮的打印格式，或者所有記錄都在一行上，則數據將無法正確讀取。

要查詢JSON數據是在漂亮的打印格式，你可以使用[Amazon Ion Hive SerDe](#)而不是OpenX JSON SerDe。考慮以前存儲在漂亮的打印格式的數據：

```
{
  name:"Mercury",
  distanceFromSun:0.39,
  orbitalPeriod:0.24,
  dayLength:58.65
}
{
  name:"Venus",
  distanceFromSun:0.72,
  orbitalPeriod:0.62,
  dayLength:243.02
}
{
  name:"Earth",
  distanceFromSun:1.00,
  orbitalPeriod:1.00,
  dayLength:1.00
}
{
  name:"Mars",
  distanceFromSun:1.52,
  orbitalPeriod:1.88,
  dayLength:1.03
}
```

若要在不重新格式化的情況下查詢此資料，您可以使用類似下列的CREATE TABLE陳述式。請注意，陳述式會指定SerDe，而不是指定STORED AS ION OpenX JSON。

```
CREATE EXTERNAL TABLE `planets_ion`(  
  `name` string,  
  `distancefromsun` DECIMAL(10, 2),  
  `orbitalperiod` DECIMAL(10, 2),  
  `daylength` DECIMAL(10, 2))  
STORED AS ION  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/json-ion/'
```

查詢 `SELECT * FROM planets_ion` 會產生與以前相同的結果。如需有關使用 Amazon Ion Hive 以這種方式建立資料表的詳細資訊 SerDe，請參閱[使用 CREATE TABLE 建立 Amazon Ion 資料表](#)。

上述範例 JSON 資料不包含複雜的資料類型，例如巢狀陣列或結構。如需有關查詢巢狀 JSON 資料的詳細資訊，請參閱[範例：還原序列化巢狀 JSON](#)。

## 主題

- [讀取 JSON 資料的最佳實務](#)
- [從字符串中提取 JSON 數據](#)
- [搜尋 JSON 陣列中的值](#)
- [取得 JSON 陣列的長度和大小](#)
- [對 JSON 查詢進行故障診斷](#)

## 讀取 JSON 資料的最佳實務

JavaScript 對象符號 ( JSON ) 是將數據結構編碼為文本的常用方法。許多應用程式和工具輸出的資料都是 JSON 編碼。

在 Amazon Athena 中，您可以從外部資料建立資料表，並將 JSON 編碼的資料加入資料表。對於此類型的來源資料，請將 Athena 搭配 [JSON SerDe 程式庫](#) 一起使用。

使用以下秘訣讀取 JSON 編碼的資料：

- 選擇正確的 SerDe、原生 JSON SerDe 或 OpenX SerDe、`org.openx.data.jsonserde.JsonSerDe`、`org.apache.hive.hcatalog.data.JsonSerDe`。需詳細資訊，請參閱 [JSON SerDe 程式庫](#)。
- 確保每一筆 JSON 編碼記錄都自成一列，並沒有呈現漂亮列印的狀態。

**Note**

SerDe 預期每個 JSON 文件都位於單行文字上，且記錄中欄位之間沒有行終止字元。如果 JSON 文本是漂亮的打印格式，您可能會收到一條錯誤消息，如 HIVE\_CURSOR\_ERROR：行不是有效的 JSON 對象或 HIVE\_CURSOR\_ERROR:: 意外 JsonParseException end-of-input：當您嘗試查詢表後創建表時，對象的預期關閉標記。如需詳細資訊，請參閱上 GitHub 的 OpenX SerDe 文件中的 [JSON 資料](#) 檔案。

- 在區分大小寫的欄中產生 JSON 編碼的資料。
- 提供選項以忽略格式不正確的記錄，如這個範例所示。

```
CREATE EXTERNAL TABLE json_table (  
  column_a string,  
  column_b int  
)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
WITH SERDEPROPERTIES ('ignore.malformed.json' = 'true')  
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/';
```

- 在 Athena 中，將來源資料中結構描述不明的欄位，轉換為 JSON 編碼字串。

Athena 在建立受 JSON 資料支援的資料表時，會根據現有和預先定義的結構描述來剖析資料。不過，並非所有資料都有預先定義的結構描述。在這種情況下，為了簡化結構描述管理，通常建議在 Athena 中將來源資料中結構描述不明的欄位，轉換為 JSON 字串，然後使用 [JSON SerDe 程式庫](#)。

例如，假設 IoT 應用程式從不同的感應器發佈含有常見欄位的事件。其中一個欄位必須存放傳送事件的感應器所獨有的自訂承載。在這種情況下，因為您不知道結構描述，建議您以 JSON 編碼字串來存放資訊。若要這樣做，請將 Athena 資料表中的資料轉換為 JSON，如下列範例所示。您也可以將 JSON 編碼的資料轉換為 Athena 資料類型。

- [將 Athena 資料類型轉換為 JSON](#)
- [將 JSON 轉換為 Athena 資料類型](#)

將 Athena 資料類型轉換為 JSON

若要將 Athena 資料類型轉換為 JSON，請使用 CAST。

```
WITH dataset AS (  

```

```

SELECT
  CAST('HELLO ATHENA' AS JSON) AS hello_msg,
  CAST(12345 AS JSON) AS some_int,
  CAST(MAP(ARRAY['a', 'b'], ARRAY[1,2]) AS JSON) AS some_map
)
SELECT * FROM dataset

```

此查詢會傳回：

```

+-----+
| hello_msg      | some_int | some_map      |
+-----+
| "HELLO ATHENA" | 12345    | {"a":1,"b":2} |
+-----+

```

將 JSON 轉換為 Athena 資料類型

若要將 JSON 資料轉換為 Athena 資料類型，請使用 CAST。

#### Note

在這個範例中，若要以 JSON 編碼表示字串，請以 JSON 關鍵字為開頭，並使用單引號，例如 JSON '12345'

```

WITH dataset AS (
  SELECT
    CAST(JSON '"HELLO ATHENA"' AS VARCHAR) AS hello_msg,
    CAST(JSON '12345' AS INTEGER) AS some_int,
    CAST(JSON '{"a":1,"b":2}' AS MAP(VARCHAR, INTEGER)) AS some_map
)
SELECT * FROM dataset

```

此查詢會傳回：

```

+-----+
| hello_msg      | some_int | some_map      |
+-----+

```

```
| HELLO ATHENA | 12345 | {a:1,b:2} |  
+-----+
```

## 從字符串中提取 JSON 數據

您可能會有來源資料，其中包含您不想要還原序列化到 Athena 中的資料表、以 JSON 編碼的字串。在這種情況下，您仍然可以使用 Presto 中提供的 JSON 函數來對此資料執行 SQL 操作。

將此 JSON 字串做為範例資料集。

```
{"name": "Susan Smith",  
 "org": "engineering",  
 "projects":  
   [  
     {"name":"project1", "completed":false},  
     {"name":"project2", "completed":true}  
   ]  
}
```

### 範例：擷取屬性

若要從 JSON 字串擷取 name 和 projects 屬性，請使用 `json_extract` 函數，如以下範例所示。`json_extract` 函數會取得包含 JSON 字串的資料欄，並使用具有句點 . 表示法、類似 JSONPath 的表達式來搜尋它。

#### Note

JSONPath 會執行簡單的樹狀目錄周遊。它使用 \$ 符號來表示 JSON 文件的根目錄，接著是一個句點和一個元素，直接在根目錄形成巢狀，例如 `$.name`。

```
WITH dataset AS (  
  SELECT '{"name": "Susan Smith",  
         "org": "engineering",  
         "projects": [{"name":"project1", "completed":false},  
                      {"name":"project2", "completed":true}]}'  
        AS myblob  
)  
SELECT  
  json_extract(myblob, '$.name') AS name,
```

```
json_extract(myblob, '$.projects') AS projects
FROM dataset
```

傳回的值是一個 JSON 編碼字串，而不是原生 Athena 資料類型。

```
+-----+
+
| name          | projects
|
+-----+
+
| "Susan Smith" | [{"name":"project1","completed":false},
{"name":"project2","completed":true}] |
+-----+
+
```

若要從 JSON 字串擷取純量值，請使用 `json_extract_scalar` 函數。它類似於 `json_extract`，但只會傳回純量值 (布林值、數字或字串)。

#### Note

請勿在陣列、地圖或結構上使用 `json_extract_scalar` 函數。

```
WITH dataset AS (
  SELECT '{"name": "Susan Smith",
         "org": "engineering",
         "projects": [{"name": "project1", "completed": false}, {"name": "project2",
         "completed": true}]}'
         AS myblob
)
SELECT
  json_extract_scalar(myblob, '$.name') AS name,
  json_extract_scalar(myblob, '$.projects') AS projects
FROM dataset
```

此查詢會傳回：

```
+-----+
| name          | projects |
```

```
+-----+
| Susan Smith |
+-----+
```

若要取得範例陣列中 `projects` 屬性的第一個元素，請使用 `json_array_get` 函數，並指定索引位置。

```
WITH dataset AS (
  SELECT '{"name": "Bob Smith",
         "org": "engineering",
         "projects": [{"name": "project1", "completed": false}, {"name": "project2",
         "completed": true}]}'
        AS myblob
)
SELECT json_array_get(json_extract(myblob, '$.projects'), 0) AS item
FROM dataset
```

它會傳回在以 JSON 編碼的陣列中指定索引位置的值。

```
+-----+
| item |
+-----+
| {"name": "project1", "completed": false} |
+-----+
```

若要傳回 Athena 字串類型，請在 JSONPath 表達式內使用 `[]` 運算子，然後使用 `json_extract_scalar` 函數。如需 `[]` 的相關資訊，請參閱 [存取陣列元素](#)。

```
WITH dataset AS (
  SELECT '{"name": "Bob Smith",
         "org": "engineering",
         "projects": [{"name": "project1", "completed": false}, {"name": "project2",
         "completed": true}]}'
        AS myblob
)
SELECT json_extract_scalar(myblob, '$.projects[0].name') AS project_name
FROM dataset
```

它會傳回此結果：

```
+-----+
```

```
| project_name |
+-----+
| project1    |
+-----+
```

## 搜尋 JSON 陣列中的值

若要判斷特定的值是否存在 JSON 編碼陣列內，請使用 `json_array_contains` 函數。

以下查詢列出參與 "project2" 的使用者名稱。

```
WITH dataset AS (
  SELECT * FROM (VALUES
    (JSON '{"name": "Bob Smith", "org": "legal", "projects": ["project1"]}' ),
    (JSON '{"name": "Susan Smith", "org": "engineering", "projects": ["project1",
"project2", "project3"]}' ),
    (JSON '{"name": "Jane Smith", "org": "finance", "projects": ["project1",
"project2"]}' )
  ) AS t (users)
)
SELECT json_extract_scalar(users, '$.name') AS user
FROM dataset
WHERE json_array_contains(json_extract(users, '$.projects'), 'project2')
```

此查詢會傳回使用者清單。

```
+-----+
| user    |
+-----+
| Susan Smith |
+-----+
| Jane Smith |
+-----+
```

以下查詢範例列出已完成專案的使用者名稱，以及完成的專案總數。它會執行下列動作：

- 使用巢狀 SELECT 陳述式以清楚顯示。
- 擷取專案陣列。
- 使用 CAST 將陣列轉換為鍵值對的原生陣列。
- 使用 UNNEST 運算子擷取每一個陣列元素。

- 依完成的專案篩選獲得的值並計算總數。

### Note

對 MAP 使用 CAST 時，您可以將索引鍵元素指定為 VARCHAR (Presto 中的原生 String)，但值要保持為 JSON，因為 MAP 中的值是不同類型：第一個鍵值對使用 String，第二個使用 Boolean。

```
WITH dataset AS (
  SELECT * FROM (VALUES
    (JSON '{"name": "Bob Smith",
          "org": "legal",
          "projects": [{"name":"project1", "completed":false}]}' ),
    (JSON '{"name": "Susan Smith",
          "org": "engineering",
          "projects": [{"name":"project2", "completed":true},
                      {"name":"project3", "completed":true}]}' ),
    (JSON '{"name": "Jane Smith",
          "org": "finance",
          "projects": [{"name":"project2", "completed":true}]}' )
  ) AS t (users)
),
employees AS (
  SELECT users, CAST(json_extract(users, '$.projects') AS
    ARRAY(MAP(VARCHAR, JSON))) AS projects_array
  FROM dataset
),
names AS (
  SELECT json_extract_scalar(users, '$.name') AS name, projects
  FROM employees, UNNEST (projects_array) AS t(projects)
)
SELECT name, count(projects) AS completed_projects FROM names
WHERE cast(element_at(projects, 'completed') AS BOOLEAN) = true
GROUP BY name
```

此查詢會傳回下列結果：

```
+-----+
| name          | completed_projects |
+-----+
```

```

| Susan Smith | 2      |
+-----+
| Jane Smith  | 1      |
+-----+

```

## 取得 JSON 陣列的長度和大小

### 範例：json\_array\_length

若要取得 JSON 編碼陣列的長度，請使用 `json_array_length` 函數。

```

WITH dataset AS (
  SELECT * FROM (VALUES
    (JSON '{"name":
      "Bob Smith",
      "org":
      "legal",
      "projects": [{"name":"project1", "completed":false}]') ,
    (JSON '{"name": "Susan Smith",
      "org": "engineering",
      "projects": [{"name":"project2", "completed":true},
        {"name":"project3", "completed":true}]') ,
    (JSON '{"name": "Jane Smith",
      "org": "finance",
      "projects": [{"name":"project2", "completed":true}]')
  ) AS t (users)
)
SELECT
  json_extract_scalar(users, '$.name') as name,
  json_array_length(json_extract(users, '$.projects')) as count
FROM dataset
ORDER BY count DESC

```

此查詢會傳回以下結果：

```

+-----+
| name      | count |
+-----+
| Susan Smith | 2    |
+-----+
| Bob Smith  | 1    |
+-----+

```

```
| Jane Smith | 1 |
+-----+
```

### 範例：json\_size

若要取得 JSON 編碼陣列或物件的大小，請使用 `json_size` 函數，並指定包含 JSON 字串的欄，以及陣列或物件的 JSONPath 表達式。

```
WITH dataset AS (
  SELECT * FROM (VALUES
    (JSON '{"name": "Bob Smith", "org": "legal", "projects": [{"name":"project1",
"completed":false}]}' ),
    (JSON '{"name": "Susan Smith", "org": "engineering", "projects":
[{"name":"project2", "completed":true},{ "name":"project3", "completed":true}]}' ),
    (JSON '{"name": "Jane Smith", "org": "finance", "projects": [{"name":"project2",
"completed":true}]}' )
  ) AS t (users)
)
SELECT
  json_extract_scalar(users, '$.name') as name,
  json_size(users, '$.projects') as count
FROM dataset
ORDER BY count DESC
```

此查詢會傳回以下結果：

```
+-----+
| name          | count |
+-----+
| Susan Smith  | 2     |
+-----+
| Bob Smith    | 1     |
+-----+
| Jane Smith   | 1     |
+-----+
```

### 對 JSON 查詢進行故障診斷

如需對 JSON 相關查詢之問題進行疑難排解的說明，請參閱 [JSON 相關錯誤](#) 或諮詢下列資源：

- [當我嘗試在 Amazon Athena 中讀取 JSON 資料時出現錯誤](#)

- [從 Athena 讀取文件時，如何解決「HIVE\\_CURSOR\\_ERROR：行不是有效的 JSON 對象-JSON 例外：重複密鑰」？ AWS Config](#)
- [輸入 JSON 文件有多筆記錄，但 Amazon Athena 中的 SELECT COUNT 查詢只傳回一筆記錄](#)
- [如何查看 Athena 資料表中某資料列的 Amazon S3 來源檔案？](#)

另請參閱[在 Amazon Athena 中進行 SQL 查詢的考量事項與限制](#)。

## 搭配 Amazon Athena 使用機器學習 (ML)

搭配 Amazon Athena 的 Machine Learning (ML) 可讓您使用 Athena 撰寫使用 Amazon SageMaker 執行 Machine Learning (ML) 推論的 SQL 陳述式。此功能讓您輕鬆存取 ML 模型以分析資料，無需使用複雜的程式設計方法來執行推論。

若要搭配 Athena 使用機器學習 (ML)，您可以使用 USING EXTERNAL FUNCTION 子句定義帶有 Athena 函數的機器學習 (ML)。函數指向要使用的 SageMaker 模型端點，並指定要傳遞給模型的變數名稱和資料類型。查詢中的後續子句參考此函數，將值傳遞給模型。模型根據查詢傳遞的值執行推論，然後傳回推論結果。如需 SageMaker 端點運作方式 SageMaker 及其運作方式的詳細資訊，請參閱 [Amazon SageMaker 開發人員指南](#)。

如需使用 ML 搭配 Athena 和 SageMaker 推論來偵測結果集中的異常值的範例，請參閱 AWS 大數據部落格文章 [叫用 Amazon Athena 機器學習推論函數來偵測異常值](#)。

### 考量與限制

- 可用區域 — Athena ML 功能是支援 Athena AWS 區域引擎第 2 版或更新版本的功能。
- SageMaker 模型端點必須接受並傳回 **text/csv** — 如需有關資料格式的詳細資訊，請參閱 Amazon SageMaker 開發人員指南中的 [推論常見資料格式](#)。
- Athena 不會傳送 CSV 標頭 — 如果您的 SageMaker 端點是 text/csv，則輸入處理常式不應假設輸入的第一行是 CSV 標頭。由於 Athena 不會傳送 CSV 標頭，因此傳回至 Athena 的輸出將包含比 Athena 預期的少一個資料列，並導致錯誤。
- SageMaker 端點擴展 — 確保參考的 SageMaker 模型端點已充分擴展，以供 Athena 呼叫端點。如需詳細資訊，請參閱 [Amazon SageMaker 開發人員指南](#)和 [Amazon SageMaker API 參考>CreateEndpointConfig](#)中的 [自動擴展 SageMaker 模型](#)。
- IAM 許可 — 若要執行指定具有 Athena 函數之 ML 的查詢，必須允許執行查詢的 IAM 主體對參考的 SageMaker 模型端點執行 sagemaker:InvokeEndpoint 動作。如需詳細資訊，請參閱 [允許使用 Athena 存取機器學習 \(ML\)](#)。
- 不能直接在 **GROUP BY** 子句中使用帶有 Athena 函數的機器學習 (ML)

## 帶有 Athena 的機器學習 (ML) 語法

USING EXTERNAL FUNCTION 子句會指定帶有一或多個 Athena 函數的機器學習 (ML)，供查詢中的後續 SELECT 陳述句參考。您定義函數名稱、變數名稱，以及變數和傳回值的資料類型。

### 概要

下面的語法顯示了 USING EXTERNAL FUNCTION 子句，其會指定帶有 Athena 函數的機器學習 (ML)。

```
USING EXTERNAL FUNCTION ml_function_name (variable1 data_type [, variable2 data_type]
[,...])
RETURNS data_type
SAGEMAKER 'sagemaker_endpoint'
SELECT ml_function_name()
```

### 參數

USING EXTERNAL FUNCTION *ml\_function\_name* (*variable1 data\_type* [, *variable2 data\_type*] [,...])

*ml\_function\_name* 定義函數名稱，供後續查詢子句中使用。每個## *data\_type* 指定一個具名的變量及其對應的數據類型，該 SageMaker 模型接受作為輸入。指定的資料類型必須是支援的 Athena 資料類型。

RETURNS *data\_type*

數###指定的 SQL 數據類型 *ML\_function #####* 為從模型的輸出。SageMaker

SAGEMAKER '*sagemaker\_endpoint*'

*#####*點。SageMaker

SELECT [...] *ml\_function\_name*(*expression*) [...]

將值傳遞給函數變量和 SageMaker 模型返回結果的 SELECT 查詢。*ml\_function\_name* 會指定先前在查詢中定義的函數，後面接著評估為傳遞值的###。傳遞和傳回的值必須符合 USING EXTERNAL FUNCTION 子句中為函數指定的相應資料類型。

### 範例

下例範例示範一個搭配 Athena 使用機器學習 (ML) 的查詢。

## Example

```
USING EXTERNAL FUNCTION predict_customer_registration(age INTEGER)
  RETURNS DOUBLE
  SAGEMAKER 'xgboost-2019-09-20-04-49-29-303'
SELECT predict_customer_registration(age) AS probability_of_enrolling, customer_id
  FROM "sampledb"."ml_test_dataset"
  WHERE predict_customer_registration(age) < 0.5;
```

## 客戶使用範例

下列影片將 Machine Learning (ML) 預覽版與 Amazon Athena 搭配使用，展示您可以使用 Athena SageMaker 的方式。

### 預測客戶流失率

以下影片說明如何將 Athena 與 Amazon 的機器學習功能相結合，SageMaker 以預測客戶流失率。

[使用 Amazon 雅典娜和亞馬遜預測客戶流失 SageMaker](#)

### 偵測殭屍網路

下面的視頻顯示了一家公司如何使用 Amazon Athena 和 Amazon SageMaker 來檢測殭屍網路。

[使用 Amazon 雅典娜和亞馬遜偵測殭屍網 SageMaker](#)

## 利用使用者定義函數來查詢

Amazon Athena 中的使用者定義函數 (UDF) 可讓您建立自訂函數，以處理記錄或記錄群組。UDF 接受參數、執行工作，然後傳回結果。

若要在 Athena 中使用 UDF，請在 SQL 查詢中的 SELECT 陳述式之前撰寫 USING EXTERNAL FUNCTION 子句。SELECT 陳述式參考 UDF，並定義在執行查詢時傳遞至 UDF 的變數。SQL 查詢呼叫 UDF 時使用 Java 執行時間叫用 Lambda 函數。UDF 在 Lambda 函數內被定義為 Java 部署套件中的方法。在同一個 Java 部署套件中，一個 Lambda 函數可以定義多個 UDF。您也可以在使用 USING EXTERNAL FUNCTION 子句中指定 Lambda 函數的名稱。

部署 Athena UDF 的 Lambda 函數時有兩種選項。您可以直接使用 Lambda，也可以使用 AWS Serverless Application Repository 部署函數。若要尋找 UDF 的現有 Lambda 函數，您可以搜尋公用 AWS Serverless Application Repository 或私有存放庫，然後部署到 Lambda。您也可以建立或修改 Java 原始程式碼、將其封裝成 JAR 檔案，然後使用 Lambda 或 AWS Serverless Application

Repository 部署此檔案。如需 Java 原始程式碼和套件範例來協助您開始使用，請參閱 [使用 Lambda 建立和部署 UDF](#)。如需有關 Lambda 的詳細資訊，請參閱《[AWS Lambda 開發人員指南](#)》。如需詳細資訊 AWS Serverless Application Repository，請參閱 [AWS Serverless Application Repository 開發人員指南](#)。

如需使用 UDF 搭配 Athena Translate 和分析文字的範例，請參閱 Machine Learning AWS earning 部落格文章 [使用 SQL 函數與 Amazon Athena、Amazon Translate 和 Amazon Comprehend 搭配使用 SQL 函數翻譯和分析文字](#)，或觀看 [video](#)。

如需使用 UDF 在 Amazon Athena 擴充地理空間查詢的範例，請參閱使用 AWS 大數據部落格中的 [使用 UDF 和 AWS Lambda 在 Amazon Athena 中擴充地理空間查詢](#)。

## 考量與限制

- 內建 Athena 函數 – Athena 中內建的函數專為高效能而設計。建議您盡可能使用內建函數，而不要使用 UDF。如需內建函數的詳細資訊，請參閱 [Amazon Athena 中的函數](#)。
- 僅限純量 UDF – Athena 僅支援純量 UDF，一次只處理一個資料行，並傳回單一資料欄值。Athena 每次叫用 Lambda 時，可能會平行地將一整批的資料行傳遞給 UDF。在設計 UDF 和查詢時，請注意此處理可能對網路流量造成的潛在影響。
- UDF 處理常式函數使用縮寫格式 – 針對 UDF 函數使用縮寫格式 (非完整格式) (例如 `package.Class` 而不是 `package.Class::method`)。
- UDF 方法必須為小寫 – UDF 方法必須為小寫；不允許使用駱駝大小寫。
- UDF 方法需要參數 – UDF 方法至少必須有一個輸入參數。嘗試調用不帶輸入參數定義的 UDF 會導致執行期例外。UDF 旨在針對資料記錄執行函數，但是不帶引數的 UDF 不接受任何資料，因此發生例外。
- Java 執行時間支援 – 目前 Athena UDF 支援 Lambda 的 Java 8 和 Java 11 執行時間。如需詳細資訊，請參閱《[AWS Lambda 開發人員指南](#)》中的 [使用 Java 建立 Lambda 函數](#)。
- IAM 許可 – 若要在 Athena 中執行和建立 UDF 查詢陳述式，執行查詢的 IAM 委託人必須能夠執行 Athena 函數以外的動作。如需詳細資訊，請參閱 [允許 Amazon Athena 使用者定義函數 \(UDF\) 的 IAM 許可政策範例](#)。
- Lambda 配額 – Lambda 配額適用於 UDF。如需詳細資訊，請參閱《[AWS Lambda 開發人員指南](#)》中的 [Lambda 配額](#)。
- 資料列層級篩選 – UDF 不支援 Lake Formation 資料列層級篩選。
- 檢視 – 您無法搭配使用檢視與 UDF。
- 已知問題 — 如需已知問題的大部分 up-to-date 清單，請參閱的 [awslabs/ aws-athena-query-federation](#) 一節中的 [限制和問題](#)。GitHub

## 影片

請觀看以下影片，進一步了解如何在 Athena 中使用 UDF。

影片：Amazon Athena 中使用者定義函數 (UDF) 的簡介

下面的影片展示如何在 Amazon Athena 中使用 UDF 來修訂敏感資訊。

### Note

本影片中的語法是發行前版本，但概念相同。在沒有 AmazonAthenaPreviewFunctionality 工作群組的情況下使用 Athena。

## [Amazon Athena 中使用者定義函數 \(UDF\) 的簡介](#)

影片：使用 Amazon Athena 中的 SQL 查詢，翻譯、分析和修訂文字欄位

下面的影片示範如何在 Amazon Athena 中使用 UDF 與其他 AWS 服務來翻譯和分析文字。

### Note

本影片中的語法是發行前版本，但概念相同。如需正確的語法，請參閱在 AWS Machine Learning 部落格中的相關部落格文章：[搭配 Amazon Athena、Amazon Translate 和 Amazon Comprehend 使用 SQL 函數，翻譯、修訂和分析文字](#)。

## [使用 Amazon Athena 中的 SQL 查詢翻譯、分析和修訂文字欄位](#)

## UDF 查詢語法

USING EXTERNAL FUNCTION 子句指定一個 UDF 或多個 UDF，供查詢中的後續 SELECT 陳述式參考。您需要 UDF 的方法名稱及託管 UDF 之 Lambda 函數的名稱。若要取代 Lambda 函數名稱，您可以使用 Lambda ARN。在跨帳戶案例中，需要使用 Lambda ARN。

### 概要

```
USING EXTERNAL FUNCTION UDF_name(variable1 data_type [, variable2 data_type] [, ...])  
RETURNS data_type  
LAMBDA 'lambda_function_name_or_ARN'  
[, EXTERNAL FUNCTION UDF_name2(variable1 data_type [, variable2 data_type] [, ...])
```

```

RETURNS data_type
LAMBDA 'lambda_function_name_or_ARN'[,...]
SELECT [...] UDF_name(expression) [, UDF_name2(expression)] [...]

```

## 參數

USING EXTERNAL FUNCTION ***UDF\_name(variable1 data\_type*** [, ***variable2 data\_type***][,...])

***UDF\_name*** 指定 UDF 的名稱，其必須對應於所參考 Lambda 函數中的 Java 方法。每個 ***variable data\_type*** 會指定具名變數及其相應的資料類型，供 UDF 接受作為輸入。此 ***data\_type*** 必須是下表列出的其中一種支援的 Athena 資料類型，並對應至相應的 Java 資料類型。

Athena 資料類型	Java 資料類型
TIMESTAMP	爪哇時間。LocalDateTime (UTC)
DATE	爪哇時間。LocalDate (UTC)
TINYINT	java.lang.Byte
SMALLINT	java.lang.Short
REAL	java.lang.Float
DOUBLE	java.lang.Double
DECIMAL (請參閱 RETURNS 備註)	爪哇數學。BigDecimal
BIGINT	java.lang.Long
INTEGER	java.lang.Int
VARCHAR	java.lang.String
VARBINARY	byte[]
BOOLEAN	java.lang.Boolean

Athena 資料類型	Java 資料類型
ARRAY	java.util.List
ROW	java.util.Map<String, Object>

## RETURNS *data\_type*

*data\_type* 會指定 UDF 傳回作為輸出的 SQL 資料類型。支援上表列出的 Athena 資料類型。對於 DECIMAL 資料類型，請使用語法 RETURNS DECIMAL(*precision*, *scale*)，其中###和##是整數。

## LAMBDA '*lambda\_function*'

*lambda\_function* 指定在執行 UDF 時要叫用的 Lambda 函數之名稱。

## SELECT [...] *UDF\_name*(*expression*) [...]

將值傳遞給 UDF 並傳回結果的 SELECT 查詢。*UDF\_name* 指定要使用的 UDF，後接評估為傳遞值的 *expression*。傳遞和傳回的值必須符合 USING EXTERNAL FUNCTION 子句中為 UDF 指定的相應資料類型。

## 範例

如需以 [AthenaUDFHandler.java](#) 程式碼為基礎的查詢範例 GitHub，請參閱 GitHub [Amazon Athena UDF 連接器](#) 頁面。

## 使用 Lambda 建立和部署 UDF

若要建立自訂 UDF，請延伸 UserDefinedFunctionHandler 類別來建立新的 Java 類別。SDK 中 [UserDefinedFunctionHandler.java](#) 的源代碼可在 [aws-labs/aws-athena-query-federation-s/athena-federation-sdk](#) [存儲庫 GitHub](#) 中找到，以及 [示例 UDF 實現](#)，您可以檢查和修改以創建自定義 UDF。

本節的步驟示範使用 [Apache Maven](#)，從命令列和部署來撰寫和建置自訂 UDF Jar 檔案。

使用 Maven 為 Athena 建立自訂 UDF 的步驟

- [複製開發套件並準備開發環境](#)
- [建立 Maven 專案](#)
- [將相依性和外掛程式新增至 Maven 專案](#)

- [撰寫 UDF 的 Java 程式碼](#)
- [建置 JAR 檔案](#)
- [將 JAR 部署到 AWS Lambda](#)

## 複製開發套件並準備開發環境

開始之前，務必使用 `sudo yum install git -y` 將 git 安裝在您的系統上。

## 若要安裝查 AWS 詢同盟 SDK

- 在命令列輸入以下命令，以複製開發套件儲存庫。此儲存庫包含開發套件、範例和一套資料來源連接器。如需資料來源連接器的詳細資訊，請參閱[使用 Amazon Athena 聯合查詢](#)。

```
git clone https://github.com/awslabs/aws-athena-query-federation.git
```

## 安裝此程序的先決條件

如果您正在使用已安裝 Apache Maven AWS CLI、和 AWS Serverless Application Model 建置工具的開發機器，則可以略過此步驟。

1. 從您複製時所建立的 `aws-athena-query-federation` 目錄的根目錄，執行 [prepare\\_dev\\_env.sh](#) 指令碼以準備開發環境。
2. 更新 shell 以獲取安裝程序所建立的新變數，或重新啟動終端機工作階段。

```
source ~/.profile
```

### Important

如果跳過此步驟，稍後會收到有關 AWS CLI 或 AWS SAM 建置工具無法發佈 Lambda 函數的錯誤訊息。

## 建立 Maven 專案

執行以下命令來建立 Maven 專案。將 `groupId` 替換為組織的唯一 ID，並替換 `my-athena-udf` 為應用程式的名稱如需更多信息，請參閱[如何創建我的第一個 Maven 項目](#)？在阿帕奇 Maven 的文檔。

```
mvn -B archetype:generate \
```

```
-DarchetypeGroupId=org.apache.maven.archetypes \  
-DgroupId=groupId \  
-DartifactId=my-athena-udfs
```

將相依性和外掛程式新增至 Maven 專案

將以下組態新增至 Maven 專案 pom.xml 檔檔案。如需範例，請參閱中的 [pom.xml](#) 檔案 GitHub。

```
<properties>  
  <aws-athena-federation-sdk.version>2022.47.1</aws-athena-federation-sdk.version>  
</properties>  
  
<dependencies>  
  <dependency>  
    <groupId>com.amazonaws</groupId>  
    <artifactId>aws-athena-federation-sdk</artifactId>  
    <version>${aws-athena-federation-sdk.version}</version>  
  </dependency>  
</dependencies>  
  
<build>  
  <plugins>  
    <plugin>  
      <groupId>org.apache.maven.plugins</groupId>  
      <artifactId>maven-shade-plugin</artifactId>  
      <version>3.2.1</version>  
      <configuration>  
        <createDependencyReducedPom>>false</createDependencyReducedPom>  
        <filters>  
          <filter>  
            <artifact>*:*</artifact>  
            <excludes>  
              <exclude>META-INF/*.SF</exclude>  
              <exclude>META-INF/*.DSA</exclude>  
              <exclude>META-INF/*.RSA</exclude>  
            </excludes>  
          </filter>  
        </filters>  
      </configuration>  
      <executions>  
        <execution>  
          <phase>package</phase>  
          <goals>  
            <goal>shade</goal>
```

```
        </goals>
    </execution>
</executions>
</plugin>
</plugins>
</build>
```

## 撰寫 UDF 的 Java 程式碼

通過擴展 [UserDefinedFunctionHandler.java](#) 創建一個新的類。在類別中撰寫 UDF。

在下列範例中，類別 `MyUserDefinedFunctions` 內建立 UDF 的兩個 Java 方法：`compress()` 和 `decompress()`。

```
*package *com.mycompany.athena.udfs;

public class MyUserDefinedFunctions
    extends UserDefinedFunctionHandler
{
    private static final String SOURCE_TYPE = "MyCompany";

    public MyUserDefinedFunctions()
    {
        super(SOURCE_TYPE);
    }

    /**
     * Compresses a valid UTF-8 String using the zlib compression library.
     * Encodes bytes with Base64 encoding scheme.
     *
     * @param input the String to be compressed
     * @return the compressed String
     */
    public String compress(String input)
    {
        byte[] inputBytes = input.getBytes(StandardCharsets.UTF_8);

        // create compressor
        Deflater compressor = new Deflater();
        compressor.setInput(inputBytes);
        compressor.finish();

        // compress bytes to output stream
```

```
byte[] buffer = new byte[4096];
ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(inputBytes.length);
while (!compressor.finished()) {
    int bytes = compressor.deflate(buffer);
    byteArrayOutputStream.write(buffer, 0, bytes);
}

try {
    byteArrayOutputStream.close();
}
catch (IOException e) {
    throw new RuntimeException("Failed to close ByteArrayOutputStream", e);
}

// return encoded string
byte[] compressedBytes = byteArrayOutputStream.toByteArray();
return Base64.getEncoder().encodeToString(compressedBytes);
}

/**
 * Decompresses a valid String that has been compressed using the zlib compression
library.
 * Decodes bytes with Base64 decoding scheme.
 *
 * @param input the String to be decompressed
 * @return the decompressed String
 */
public String decompress(String input)
{
    byte[] inputBytes = Base64.getDecoder().decode((input));

    // create decompressor
    Inflater decompressor = new Inflater();
    decompressor.setInput(inputBytes, 0, inputBytes.length);

    // decompress bytes to output stream
    byte[] buffer = new byte[4096];
    ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(inputBytes.length);
    try {
        while (!decompressor.finished()) {
            int bytes = decompressor.inflate(buffer);
            if (bytes == 0 && decompressor.needsInput()) {
```

```
        throw new DataFormatException("Input is truncated");
    }
    byteArrayOutputStream.write(buffer, 0, bytes);
}
}
catch (DataFormatException e) {
    throw new RuntimeException("Failed to decompress string", e);
}

try {
    byteArrayOutputStream.close();
}
catch (IOException e) {
    throw new RuntimeException("Failed to close ByteArrayOutputStream", e);
}

// return decoded string
byte[] decompressedBytes = byteArrayOutputStream.toByteArray();
return new String(decompressedBytes, StandardCharsets.UTF_8);
}
}
```

## 建置 JAR 檔案

執行 `mvn clean install` 來建置專案。成功建置後，專案的 `target` 資料夾中會建立名為 `artifactId-version.jar` 的 JAR 檔案，其中 `artifactId` 是您在 Maven 專案中提供的名稱，例如 `my-athena-udfs`。

## 將 JAR 部署到 AWS Lambda

將程式碼部署到 Lambda 時有兩種選項：

- 部署方式 AWS Serverless Application Repository (建議)
- 從 JAR 檔案建立 Lambda 函數

### 選項 1：部署到 AWS Serverless Application Repository

當您將 JAR 檔案部署到時 AWS Serverless Application Repository，您會建立代表應用程式架構的 AWS SAM 範本 YAML 檔案。然後，您需指定此 YAML 檔案和 Simple Storage Service (Amazon S3) 儲存貯體，應用程式的成品會上傳到此處，並提供給 AWS Serverless Application Repository。下列程序使用 [publish.sh](#) 指令碼 (位於您先前複製的 Athena Query Federation 軟體開發套件的 `athena-query-federation/tools` 目錄中)。

如需詳細資訊和需求，請參閱AWS Serverless Application Repository 開發人員指南中的發佈應用程式、開發AWS Serverless Application Model 人員指南中的[AWS SAM 範本概念](#)，以及使用 [AWS SAM CLI 發佈無伺服器應用程式](#)。

下列範例示範 YAML 檔案中的參數。將類似的參數新增至 YAML 檔案，並儲存在專案目錄中。有關完整示例，請參閱中的[雅典娜-udf.yaml](#)。GitHub

```
Transform: 'AWS::Serverless-2016-10-31'
Metadata:
  'AWS::ServerlessRepo::Application':
    Name: MyApplicationName
    Description: 'The description I write for my application'
    Author: 'Author Name'
    Labels:
      - athena-federation
    SemanticVersion: 1.0.0
Parameters:
  LambdaFunctionName:
    Description: 'The name of the Lambda function that will contain your UDFs.'
    Type: String
  LambdaTimeout:
    Description: 'Maximum Lambda invocation runtime in seconds. (min 1 - 900 max)'
    Default: 900
    Type: Number
  LambdaMemory:
    Description: 'Lambda memory in MB (min 128 - 3008 max).'
    Default: 3008
    Type: Number
Resources:
  ConnectorConfig:
    Type: 'AWS::Serverless::Function'
    Properties:
      FunctionName: !Ref LambdaFunctionName
      Handler: "full.path.to.your.handler. For example, com.amazonaws.athena.connectors.udfs.MyUDFHandler"
      CodeUri: "Relative path to your JAR file. For example, ./target/athena-udfs-1.0.jar"
      Description: "My description of the UDFs that this Lambda function enables."
      Runtime: java8
      Timeout: !Ref LambdaTimeout
      MemorySize: !Ref LambdaMemory
```

將 `publish.sh` 指令碼複製到您儲存 YAML 檔案的專案目錄，然後執行以下命令：

```
./publish.sh MyS3Location MyYamlFile
```

例如，如果儲存貯體位置是 `s3://DOC-EXAMPLE-BUCKET/mysarapps/athenaudf`，而且 YAML 檔案儲存為 `my-athena-udfs.yaml`：

```
./publish.sh DOC-EXAMPLE-BUCKET/mysarapps/athenaudf my-athena-udfs
```

若要建立 Lambda 函數

1. 前往以下位置開啟 Lambda 主控台：<https://console.aws.amazon.com/lambda/>，選擇 Create function (建立函數)，然後選擇 Browse serverless app repository (瀏覽無伺服器應用程式儲存庫)
2. 選擇 Private applications (私有應用程式)，在清單中尋找您的應用程式，或使用關鍵字搜尋應用程式，然後選取應用程式。
3. 檢閱並提供應用程式詳細資料，然後選擇 Deploy (部署)。

您現在可以使用 Lambda 函數 JAR 檔案中定義的方法名稱，作為 Athena 中的 UDF。

選項 2：直接建立 Lambda 函數

您也可以直接使用主控台或建立 Lambda 函數 AWS CLI。下列範例示範如何使用 Lambda create-function CLI 命令。

```
aws lambda create-function \  
  --function-name MyLambdaFunctionName \  
  --runtime java8 \  
  --role arn:aws:iam::1234567890123:role/my_lambda_role \  
  --handler com.mycompany.athena.udfs.MyUserDefinedFunctions \  
  --timeout 900 \  
  --zip-file fileb://./target/my-athena-udfs-1.0-SNAPSHOT.jar
```

## 跨區域查詢

Athena 支援在不同於您使用 Athena AWS 區域的區域查詢 Amazon S3 資料的功能。如果移動資料不切實際或未獲許可，或者您想要跨多個區域查詢資料，則可以選擇跨區域查詢。即使 Athena 在特定區域中不可用，也可以從 Athena 可用的另一個區域查詢來自該區域的資料。

若要查詢某個區域中的資料，您的帳戶必須在該區域中啟用，即使 Amazon S3 資料不屬於您的帳戶亦是如此。對於某些區域，如美國東部 (俄亥俄)，您在建立帳戶時會自動啟用對該區域的存取權限。其

他區域要求您的帳戶在使用該區域之前「選擇加入」該區域。如需需要選擇加入的區域清單，請參閱 Amazon EC2 使用者指南中的 [可用區域](#)。如需有關選擇加入區域的特定指示，請參閱中的 [管理 AWS 區域](#)。Amazon Web Services 一般參考

## 考量與限制

- 資料存取許可 – 要從 Athena 跨地區成功查詢 Amazon S3 資料，您的帳戶必須具有讀取資料的許可。如果要查詢的資料屬於另一個帳戶，則另一個帳戶必須授予您對包含資料的 Amazon S3 位置的存取權限。
- 資料傳輸費 – 跨區域查詢要支付 Amazon S3 資料傳輸費。執行查詢可能會導致傳輸的資料量超過資料集大小。建議您先測試資料子集的查詢，並檢閱 [AWS Cost Explorer](#) 中的成本。
- AWS Glue— 您可以 AWS Glue 跨地區使用。跨區域 AWS Glue 流量可能需要支付額外費用。如需詳細資訊，請參閱 AWS 大數據部落格中的 [建立跨帳戶和跨區域 AWS Glue 連線](#)。
- Amazon S3 加密選項 – 跨區域查詢支援 SSE-S3 和 SSE-KMS 加密選項；但不支援 CSE-KMS。如需詳細資訊，請參閱 [支援的 Amazon S3 加密選項](#)。
- 同盟查詢 — 不支援跨 AWS 區域 使用聯合查詢。
- 中國地區 — 中國地區不支援跨區域查詢。

只要符合上述條件，您就可以建立一個指向您指定的 LOCATION 值的 Athena 資料表，並以透明方式查詢資料。並不需要特別語法。如需有關建立 Athena 資料表的資訊，請參閱 [在 Athena 中建立資料表](#)。

## 查詢 AWS Glue Data Catalog

由於許多 AWS 服務 都會使用 AWS Glue Data Catalog 作為其中央中繼資料儲存庫，您可能需要查詢資料目錄中繼資料。若要執行此操作，您可以在 Athena 中使用 SQL 查詢。您可以使用 Athena 查詢 AWS Glue 目錄中繼資料，例如資料庫、資料表、分割區和資料欄。

若要取得 AWS Glue 目錄中繼資料，請在 Athena 後端查詢 information\_schema 資料庫。本主題中的查詢範例會顯示如何使用 Athena 來查詢常見使用案例的 AWS Glue 目錄中繼資料。

### 主題

- [考量與限制](#)
- [列出資料庫並搜尋指定的資料庫](#)
- [列出指定資料庫中的資料表，並依名稱搜尋資料表](#)

- [列出特定資料表的分割區](#)
- [列出所有資料表的所有資料欄](#)
- [列出特定表格共同的資料欄](#)
- [列示或搜尋指定的資料表或檢視的資料欄](#)

## 考量與限制

- 不是查詢 `information_schema` 資料庫，可使用個別 Apache Hive [DDL 命令](#)，擷取特定資料庫、資料表、檢視、分割區和資料欄的中繼資料資訊。然而，輸出採用非表格格式。
- 如果您擁有小到中等數量的 AWS Glue 中繼資料，則查詢 `information_schema` 具有最高效能。如果您有大量中繼資料，則可能會出現錯誤。
- 您無法使用 `CREATE VIEW` 在 `information_schema` 資料庫上建立檢視。

## 列出資料庫並搜尋指定的資料庫

本節中的範例顯示如何依結構描述名稱列出中繼資料中的資料庫。

### Example – 列出資料庫

以下範例查詢會列出 `information_schema.schemata` 資料表中的數據庫。

```
SELECT schema_name
FROM   information_schema.schemata
LIMIT 10;
```

下列資料表會顯示範例結果。

6	alb-databas1
7	alb_original_cust
8	alblogsdatabase
9	athena_db_test
10	athena_ddl_db

## Example – 搜尋指定資料庫

在下列範例查詢中，`rdspostgresql` 是範例資料庫。

```
SELECT schema_name
FROM   information_schema.schemata
WHERE  schema_name = 'rdspostgresql'
```

下列資料表會顯示範例結果。

	schema_name
1	rdspostgresql

## 列出指定資料庫中的資料表，並依名稱搜尋資料表

若要列出資料表的中繼資料，您可以透過資料表模式或資料表名稱查詢。

### Example – 依結構描述列出資料表

下列查詢會列出使用 `rdspostgresql` 資料表模式的資料表。

```
SELECT table_schema,
       table_name,
       table_type
FROM   information_schema.tables
WHERE  table_schema = 'rdspostgresql'
```

下列資料表會顯示範例結果。

	table_schema	table_name	table_type
1	rdspostgresql	rdspostgresqldb1_public_account	BASE TABLE

### Example – 依名稱搜尋資料表

下列查詢包含 `athena1` 資料表的中繼資料資訊。

```
SELECT table_schema,
       table_name,
       table_type
FROM   information_schema.tables
WHERE  table_name = 'athena1'
```

下列資料表會顯示範例結果。

	table_schema	table_name	table_type
1	預設	athena1	BASE TABLE

## 列出特定資料表的分割區

您可以如下列範例所示，使用 `SHOW PARTITIONS table_name` 列出指定資料表的分割區。

```
SHOW PARTITIONS cloudtrail_logs_test2
```

您可以使用 `$partitions` 中繼資料查詢，列出特定資料表的分割區編號和分割區值。

Example – 使用 `$partitions` 語法查詢資料表的分割區

下列範例查詢會使用 `$partitions` 語法，列出資料表 `cloudtrail_logs_test2` 的分割區。

```
SELECT * FROM default."cloudtrail_logs_test2$partitions" ORDER BY partition_number
```

下列資料表會顯示範例結果。

	table_cat alog	table_sch ema	table_name	年	月	天
1	awsdataca talog	預設	cloudtrail_logs_te st2	2020	08	10
2	awsdataca talog	預設	cloudtrail_logs_te st2	2020	08	11
3	awsdataca talog	預設	cloudtrail_logs_te st2	2020	08	12

## 列出所有資料表的所有資料欄

您可以列出 AwsDataCatalog 中所有資料表的所有資料欄，或者 AwsDataCatalog 中特定資料庫的所有資料表。

- 若要列出 AwsDataCatalog 中所有資料庫的所有資料欄，請使用查詢 `SELECT * FROM information_schema.columns`。
- 若要將結果限制在特定資料庫，請在 WHERE 子句中使用 `table_schema='database_name'`。

### Example - 列出特定資料庫中所有資料表的所有資料欄

下面的範例查詢會列出資料庫 `webdata` 中所有資料表的所有資料欄。

```
SELECT * FROM information_schema.columns WHERE table_schema = 'webdata'
```

## 列出特定表格共同的資料欄

可列出資料庫中的特定資料表的共同資料欄。

- 使用語法 `SELECT column_name FROM information_schema.columns`。
- 對於 WHERE 子句，請使用語法 `WHERE table_name IN ('table1', 'table2')`。

### Example – 列出同一資料庫中兩個表的共同資料欄

下列範例查詢會列出資料表 `table1` 和 `table2` 的共同資料欄。

```
SELECT column_name
FROM information_schema.columns
WHERE table_name IN ('table1', 'table2')
GROUP BY column_name
HAVING COUNT(*) > 1;
```

## 列示或搜尋指定的資料表或檢視的資料欄

您可以列出資料表的所有資料欄、檢視的所有資料欄，或在指定的資料庫和資料表中依名稱搜尋欄。

若要列出資料欄，請使用 `SELECT *` 查詢。在 FROM 子句中，指定 `information_schema.columns`。在 WHERE 子句中，使用 `table_schema='database_name'`

來指定資料庫，使用 `table_name = 'table_name'` 以指定具有您想要列出之資料欄的資料表或檢視。

#### Example – 列出指定資料表的所有資料欄

下面的範例查詢會列出資料表 `rdspostgresqldb1_public_account` 的所有資料欄。

```
SELECT *
FROM   information_schema.columns
WHERE  table_schema = 'rdspostgresqldb1'
       AND table_name = 'rdspostgresqldb1_public_account'
```

下列資料表會顯示範例結果。

	table_catalog	table_schema	table_name	column_name	ordinal_position	column_default	is_nullable	data_type	comment	extra_info
1	awsdatacatalog	rdspostgresqldb1_public_account	password	1			是	varchar		
2	awsdatacatalog	rdspostgresqldb1_public_account	user_id	2			是	integer		
3	awsdatacatalog	rdspostgresqldb1_public_account	created_time	3			是	timestamp		
4	awsdatacatalog	rdspostgresqldb1_public_account	last_login	4			是	timestamp		
5	awsdatacatalog	rdspostgresqldb1_public_account	email	5			是	varchar		

	table_cat alog	table_sch ema	table_nam e	column_n me	ordinal_p osition	column_d efault	is_null able	data_t ype	comm ent	extra_inf o
			public_acc ount							
6	awsdataca talog	rdspostg resql	rdspostgr esqldb1_p ublic_acc ount	用戶名	6		是	varchar		

### Example – 列出指定檢視的資料欄

下列範例查詢會針對檢視 default 列出 arrayview 資料庫中的所有資料欄。

```
SELECT *
FROM   information_schema.columns
WHERE  table_schema = 'default'
       AND table_name = 'arrayview'
```

下列資料表會顯示範例結果。

	table_cat alog	table_sch ema	table_n ame	column_n ame	ordinal_p osition	column_d efault	is_null able	data_t ype	comm ent	extra_inf o
1	awsdataca talog	預設	arrayvie w	searchdat e	1		是	varchar		
2	awsdataca talog	預設	arrayvie w	sid	2		是	varchar		
3	awsdataca talog	預設	arrayvie w	btid	3		是	varchar		
4	awsdataca talog	預設	arrayvie w	p	4		是	varchar		
5	awsdataca talog	預設	arrayvie w	infantpri ce	5		是	varchar		

	table_cat alog	table_sch ema	table_n e	column_n me	ordinal_p osition	column_d efault	is_null le	data_t yp	comm	extra_inf o
6	awsdataca talog	預設	arrayvie	sump	6		是	varchar		
7	awsdataca talog	預設	arrayvie	journeym parray	7		是	array(va char)		

Example – 在指定的資料庫與資料表中依名稱搜尋資料欄

下列範例查詢會在 sid 資料庫的 arrayview 檢視中搜尋 default 資料欄的中繼資料。

```
SELECT *
FROM information_schema.columns
WHERE table_schema = 'default'
      AND table_name = 'arrayview'
      AND column_name='sid'
```

下列資料表會顯示範例結果。

	table_cat alog	table_sch ema	table_n e	column_n me	ordinal_p osition	column_d efault	is_null le	data_t yp	comm	extra_inf o
1	awsdataca talog	預設	arrayview	sid	2		是	varcha		

## 查詢 AWS 服務 記錄檔

本節包括使用 Amazon Athena 查詢常用資料集的幾個程序，例如 AWS CloudTrail 日誌、Amazon CloudFront 日誌、Classic Load Balancer 日誌、Application Load Balancer 日誌、Amazon VPC 流程日誌和 Network Load Balancer 日誌。

本節中的任務使用 Athena 主控台，但您也可以使用其他工具，例如 [Athena JDBC 驅動程式](#)、[AWS CLI](#) 或 [Amazon Athena API 參考](#)。

如需在 Athena 中使 AWS CloudFormation 用自動建立日 AWS 服務 誌表、分區和範例查詢的相關資訊，請參閱 AWS 大數據部落格中的 [自動化 AWS 服務 日誌表格建立和使用 Amazon Athena 進行查](#)

[詢](#)。如需使用 Python 程式庫 AWS Glue 來建立用於處理 AWS 服務日誌並在 Athena 中查詢日誌的通用架構的詳細資訊，請參閱[使用 Amazon Athena 輕鬆查詢 AWS 服務日誌](#)。

本節中的主題假設您已設定適當許可來存取 Athena，以及應包含要查詢的資料的 Amazon S3 儲存貯體。如需詳細資訊，請參閱[設定](#) 及 [開始使用](#)。

## 主題

- [查詢 Application Load Balancer 日誌](#)
- [查詢 Classic Load Balancer 日誌](#)
- [查詢 Amazon CloudFront 日誌](#)
- [查詢 AWS CloudTrail 記錄檔](#)
- [查詢 Amazon EMR 日誌](#)
- [查詢 AWS Global Accelerator 流程記錄](#)
- [查詢 Amazon GuardDuty 發現](#)
- [查詢 AWS Network Firewall 記錄檔](#)
- [查詢 Network Load Balancer 日誌](#)
- [查詢 Amazon Route 53 Resolver 查詢日誌](#)
- [查詢 Amazon SES 事件日誌](#)
- [查詢 Amazon VPC 流程日誌](#)
- [查詢 AWS WAF 記錄檔](#)

## 查詢 Application Load Balancer 日誌

Application Load Balancer 是 Elastic Load Balancing 的一個負載平衡選項，可實施使用容器的微服務部署中的流量分配。查詢 Application Load Balancer 日誌可讓您查看流量來源、延遲，以及往返於 Elastic Load Balancing 執行個體與後端應用程式傳輸的位元組。如需詳細資訊，請參閱[Application Load Balancer 使用者指南中的 Application Load Balancer 的存取記錄和應用程式負載平衡器的連線記錄](#)。

## 主題

- [必要條件](#)
- [建立 ALB 存取記錄的資料表](#)
- [使用分區投影在 Athena 中建立 ALB 存取記錄的資料表](#)
- [ALB 存取記錄的查詢範例](#)

- [建立 ALB 連線記錄的資料表](#)
- [使用分割區投影在 Athena 中建立 ALB 連線記錄的資料表](#)
- [ALB 連線記錄的查詢範例](#)
- [其他資源](#)

## 必要條件

- 啟用[存取記錄](#)或[連線記錄](#)，以便將 Application Load Balancer 日誌儲存到 Amazon S3 儲存貯體。
- 用於保存您將為 Athena 建立的資料表的資料庫。若要建立資料庫，您可以使用 Athena 或 AWS Glue 主控台。如需詳細資訊，請參閱本指南中的 [在 Athena 中建立資料庫](#)，或《AWS 開發人員指南》<https://docs.aws.amazon.com/glue/latest/dg/console-databases.html>中的在AWS Glue Glue 主控台上使用資料庫。

## 建立 ALB 存取記錄的資料表

1. 將下列CREATE TABLE陳述式複製並貼到 Athena 主控台的查詢編輯器中，然後根據您自己的記錄項目需求進行修改。如需有關 Athena 主控台入門的資訊，請參閱[開始使用](#)。將LOCATION條款中的路徑取代為您的 Amazon S3 存取日誌資料夾位置。如需有關存取記錄檔位置的詳細資訊，請參閱《[應用程式負載平衡器使用者指南](#)》中的[存取記錄檔](#)。

如需每個記錄檔欄位的相關資訊，請參閱《[應用程式負載平衡器使用者指南](#)》中的[存取記錄項目](#)。

### Note

下列範例CREATE TABLE陳述式包含最近新增的classificationreason、和 conn\_trace\_id (「可追蹤性 ID」或 TID) 欄。若要為不包含這些項目的「應 Application Load Balancer」存取日誌建立表格，請從CREATE TABLE陳述式中移除對應的資料欄，然後視情況修改一般表示式。

```
CREATE EXTERNAL TABLE IF NOT EXISTS alb_access_logs (  
    type string,  
    time string,  
    elb string,  
    client_ip string,  
    client_port int,  
    target_ip string,
```

```

target_port int,
request_processing_time double,
target_processing_time double,
response_processing_time double,
elb_status_code int,
target_status_code string,
received_bytes bigint,
sent_bytes bigint,
request_verb string,
request_url string,
request_proto string,
user_agent string,
ssl_cipher string,
ssl_protocol string,
target_group_arn string,
trace_id string,
domain_name string,
chosen_cert_arn string,
matched_rule_priority string,
request_creation_time string,
actions_executed string,
redirect_url string,
lambda_error_reason string,
target_port_list string,
target_status_code_list string,
classification string,
classification_reason string,
conn_trace_id string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
    '([\ ]*) ([\ ]*) ([\ ]*) ([\ ]*):([0-9]*) ([\ ]*)[:-]([0-9]*) ([-.\ ]*)
    ([-.\ ]*) ([-.\ ]*) (|[-0-9]*) (-|[-0-9]*) ([-0-9]*) ([-0-9]*) \"([\ ]*) (.*) (-
    |[\ ]*)\" \"([\ ]*)\" ([A-Z0-9-_\ ]*) ([A-Za-z0-9-.\ ]*) ([\ ]*) \"([\ ]*)\" \"([\ ]
    \ ]*)\" \"([\ ]*)\" ([-.\ ]*) ([\ ]*) \"([\ ]*)\" \"([\ ]*)\" \"([\ ]*)\" \"([\ ]
    \s]+?)\" \"([\ ]*)\" \"([\ ]*)\" \"([\ ]*)\" ?([\ ]*)?(. *)?'
  LOCATION 's3://DOC-EXAMPLE-BUCKET/access-log-folder-path'

```

2. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 alb\_access\_logs 資料表，讓其中的資料可供您發出查詢。

## 使用分區投影在 Athena 中建立 ALB 存取記錄的資料表

由於 ALB 存取記錄具有已知的結構，您可以事先指定其分割配置，因此您可以使用 Athena 分割區投影功能，縮短查詢執行時間並自動化分割區管理。分割區投影會在新增資料時自動新增分割區。因此您無需使用 ALTER TABLE ADD PARTITION 手動新增分割區。

下列範例 CREATE TABLE 陳述式會自動在指定日期的 ALB 存取記錄上使用資料分割投影，直到單一 AWS 區域為止。該陳述式基於上一節中的範例，但新增了 PARTITIONED BY 和 TBLPROPERTIES 子句以啟用分割區投影。在 LOCATION 和 storage.location.template 子句中，將預留位置取代為識別 ALB 存取日誌之 Amazon S3 儲存貯體位置的值。如需有關存取記錄檔位置的詳細資訊，請參閱 [《應用程式負載平衡器使用者指南》中的存取記錄檔](#)。對於 projection.day.range，請用您要使用的開始日期取代 `2022/01/01`。成功執行查詢之後，您可以查詢資料表。您無須執行 ALTER TABLE ADD PARTITION 就能載入分割區。如需有關每個記錄檔欄位的資訊，請參閱 [存取記錄項目](#)。

```
CREATE EXTERNAL TABLE IF NOT EXISTS alb_access_logs (  
    type string,  
    time string,  
    elb string,  
    client_ip string,  
    client_port int,  
    target_ip string,  
    target_port int,  
    request_processing_time double,  
    target_processing_time double,  
    response_processing_time double,  
    elb_status_code int,  
    target_status_code string,  
    received_bytes bigint,  
    sent_bytes bigint,  
    request_verb string,  
    request_url string,  
    request_proto string,  
    user_agent string,  
    ssl_cipher string,  
    ssl_protocol string,  
    target_group_arn string,  
    trace_id string,  
    domain_name string,  
    chosen_cert_arn string,  
    matched_rule_priority string,  
    request_creation_time string,  
    actions_executed string,
```

```

        redirect_url string,
        lambda_error_reason string,
        target_port_list string,
        target_status_code_list string,
        classification string,
        classification_reason string,
        conn_trace_id string
    )
PARTITIONED BY
(
    day STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
    'serialization.format' = '1',
    'input.regex' =
        '^([ ]*)([ ]*)([ ]*)([ ]*):([0-9]*)([ ]*)[:-]([0-9]*)([-0-9]*)
([-0-9]*)([-0-9]*)|[0-9]*(-|[ ]*)\("([^\"])*\)"([A-Z0-9-_]+)([A-Za-z0-9-]*)
([ ]*)\("([^\"])*\)"\("([^\"])*\)"\("([^\"])*\)"\("([ ]*)\)"\("([^\s]+?)\)"
\("([^\s]+)\)"\("([ ]*)\)"\("([ ]*)\)" ?([ ]*)?(.*)?')
        LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-NUMBER>/
elasticloadbalancing/<REGION>/'
TBLPROPERTIES
(
    "projection.enabled" = "true",
    "projection.day.type" = "date",
    "projection.day.range" = "2022/01/01,NOW",
    "projection.day.format" = "yyyy/MM/dd",
    "projection.day.interval" = "1",
    "projection.day.interval.unit" = "DAYS",
    "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-
NUMBER>/elasticloadbalancing/<REGION>/${day}"
)

```

如需有關分割區投影的詳細資訊，請參閱[使用 Amazon Athena 進行分割區投影](#)。

### ALB 存取記錄的查詢範例

以下查詢會計算負載平衡器收到的 HTTP GET 請求，並依用戶端 IP 地址分組：

```

SELECT COUNT(request_verb) AS
    count,
    request_verb,

```

```
client_ip
FROM alb_access_logs
GROUP BY request_verb, client_ip
LIMIT 100;
```

另一個查詢顯示 Safari 瀏覽器使用者造訪的 URL：

```
SELECT request_url
FROM alb_access_logs
WHERE user_agent LIKE '%Safari%'
LIMIT 10;
```

下列查詢顯示具有 ELB 狀態碼值大於或等於 500 的記錄。

```
SELECT * FROM alb_access_logs
WHERE elb_status_code >= 500
```

下列範例說明如何以 datetime 剖析日誌：

```
SELECT client_ip, sum(received_bytes)
FROM alb_access_logs
WHERE parse_datetime(time, 'yyyy-MM-dd' 'T' 'HH:mm:ss.SSSSSS' 'Z')
      BETWEEN parse_datetime('2018-05-30-12:00:00', 'yyyy-MM-dd-HH:mm:ss')
      AND parse_datetime('2018-05-31-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
GROUP BY client_ip;
```

下列查詢會查詢針對指定日期的所有 ALB 存取記錄使用資料分割投影的資料表。

```
SELECT *
FROM alb_access_logs
WHERE day = '2022/02/12'
```

建立 ALB 連線記錄的資料表

1. 將下列範例 CREATE TABLE 陳述式複製並貼到 Athena 主控台的查詢編輯器中，然後根據您自己的記錄項目需求進行修改。如需有關 Athena 主控台入門的資訊，請參閱[開始使用](#)。將 LOCATION 條款中的路徑取代為 Amazon S3 連線日誌資料夾位置。如需有關連線記錄檔位置的詳細資訊，請參閱《應用程式負載平衡器使用者指南》中的[連線記錄檔](#)。如需有關每個記錄檔欄位的資訊，請參閱[連線記錄項目](#)。

```
CREATE EXTERNAL TABLE IF NOT EXISTS alb_connection_logs (
```

```

time string,
client_ip string,
client_port int,
listener_port int,
tls_protocol string,
tls_cipher string,
tls_handshake_latency double,
leaf_client_cert_subject string,
leaf_client_cert_validity string,
leaf_client_cert_serial_number string,
tls_verify_status string,
conn_trace_id string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
  '^\"([^\"]*)\" ([^ ]*) ([^ ]*) ([^ ]*) ?([^ ]*)?( (.*))?'
  LOCATION 's3://DOC-EXAMPLE-BUCKET/connection-log-folder-path/'

```

2. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 alb\_connection\_logs 資料表，讓其中的資料可供您發出查詢。

### 使用分割區投影在 Athena 中建立 ALB 連線記錄的資料表

由於 ALB 連線記錄具有已知的結構，您可以事先指定其資料分割配置，因此您可以使用 Athena 分割區投影功能，縮短查詢執行階段並自動化磁碟分割管理。分割區投影會在新增資料時自動新增分割區。因此您無需使用 ALTER TABLE ADD PARTITION 手動新增分割區。

下列範例 CREATE TABLE 陳述式會自動在指定日期的 ALB 連線記錄上使用資料分割投影，直到單一 AWS 區域為止。該陳述式基於上一節中的範例，但新增了 PARTITIONED BY 和 TBLPROPERTIES 子句以啟用分割區投影。在 LOCATION 和 storage.location.template 子句中，將預留位置取代為識別 ALB 連線日誌之 Amazon S3 儲存貯體位置的值。如需有關連線記錄檔位置的詳細資訊，請參閱《應用程式負載平衡器使用者指南》中的 [連線記錄檔](#)。對於 projection.day.range，請將 **2023/01/01** 取代為您要使用的開始日期。成功執行查詢之後，您可以查詢資料表。您無須執行 ALTER TABLE ADD PARTITION 就能載入分割區。如需有關每個記錄檔欄位的資訊，請參閱 [連線記錄項目](#)。

```

CREATE EXTERNAL TABLE IF NOT EXISTS alb_connection_logs (
  time string,
  client_ip string,

```

```

client_port int,
listener_port int,
tls_protocol string,
tls_cipher string,
tls_handshake_latency double,
leaf_client_cert_subject string,
leaf_client_cert_validity string,
leaf_client_cert_serial_number string,
tls_verify_status string,
conn_trace_id string
)
PARTITIONED BY
(
  day STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
  '\("[^\"]*"\) ([^ ]*) ([^ ]*) ([0-9]*) ([0-9]*) ([A-Za-z0-9.-]*) ([^ ]*) ([-\.0-9]*)'
  LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-NUMBER>/
elasticloadbalancing/<REGION>/'
TBLPROPERTIES
(
  "projection.enabled" = "true",
  "projection.day.type" = "date",
  "projection.day.range" = "2023/01/01,NOW",
  "projection.day.format" = "yyyy/MM/dd",
  "projection.day.interval" = "1",
  "projection.day.interval.unit" = "DAYS",
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-
NUMBER>/elasticloadbalancing/<REGION>/${day}"
)

```

如需有關分割區投影的詳細資訊，請參閱[使用 Amazon Athena 進行分割區投影](#)。

### ALB 連線記錄的查詢範例

下列查詢會計數未發生的tls\_verify\_status值'Success'，依用戶端 IP 位址分組：

```

SELECT DISTINCT client_ip, count() AS count FROM alb_connection_logs
WHERE tls_verify_status != 'Success'
GROUP BY client_ip

```

```
ORDER BY count() DESC;
```

下列查詢會搜尋在指定時間範圍內超過 2 秒的值：tls\_handshake\_latency

```
SELECT * FROM alb_connection_logs
WHERE
  (
    parse_datetime(time, 'yyyy-MM-dd'T'HH:mm:ss.SSSSSS'Z')
    BETWEEN
    parse_datetime('2024-01-01-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
    AND
    parse_datetime('2024-03-20-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
  )
AND
  (tls_handshake_latency >= 2.0);
```

## 其他資源

- AWS 知識中心中的[如何使用 Amazon Athena 分析 Application Load Balancer 存取日誌？](#)。
- 如需有關 Elastic Load Balancing 中的 HTTP 狀態碼的資訊，請參閱《Application Load Balancers 使用者指南》中的[對 Application Load Balancer 進行疑難排解](#)。
- 透過大數據部落格中的 AWS Glue 自訂分類器和 Amazon Athena，更有效率地編目和分析 Application Load AWS B [alancer 日誌](#)。

## 查詢 Classic Load Balancer 日誌

使用 Classic Load Balancer 日誌來分析和了解往返於 Elastic Load Balancing 執行個體與後端應用程式的流量模式。您可以查看流量的來源、延遲和已傳輸的位元組。

分析 Elastic Load Balancing 日誌之前，請對其進行設定，以在目的地 Amazon S3 儲存貯體中儲存。如需詳細資訊，請參閱[為您的 Classic Load Balancer 啟用存取日誌](#)。

- [為 Elastic Load Balancing 日誌建立資料表](#)
- [Elastic Load Balancing 查詢範例](#)

若要為 Elastic Load Balancing 日誌建立資料表

1. 複製下列 DDL 陳述式，並將其貼到 Athena 主控台。檢查 Elastic Load Balancing 日誌記錄的[語法](#)。您可能需要更新以下查詢，為記錄的最新版本包含欄和 Regex 語法。

```

CREATE EXTERNAL TABLE IF NOT EXISTS elb_logs (

    timestamp string,
    elb_name string,
    request_ip string,
    request_port int,
    backend_ip string,
    backend_port int,
    request_processing_time double,
    backend_processing_time double,
    client_response_time double,
    elb_response_code string,
    backend_response_code string,
    received_bytes bigint,
    sent_bytes bigint,
    request_verb string,
    url string,
    protocol string,
    user_agent string,
    ssl_cipher string,
    ssl_protocol string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
    'serialization.format' = '1',
    'input.regex' = '([^ ]*) ([^ ]*) ([^ ]*):([0-9]*) ([^ ]*)[:-]([0-9]*) ([-\.0-9]*)
([-\.0-9]*) ([-\.0-9]*) (|[-0-9]*) (-|[-0-9]*) ([-0-9]*) ([-0-9]*) \\\"([^ ]*)
([^ ]*) (- |[^ ]*)\\\" (\\"[^\"]*"*) ([A-Z0-9-]+) ([A-Za-z0-9-\.]*)$'
)
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/AWS_account_ID/elasticloadbalancing/';

```

2. 修改 LOCATION Amazon S3 儲存貯體，以指定您的 Elastic Load Balancing 日誌的目的地。
3. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 elb\_logs 資料表，讓其中的資料可供您查詢。如需詳細資訊，請參閱 [Elastic Load Balancing 查詢範例](#)。

### Elastic Load Balancing 查詢範例

使用與以下範例類似的查詢。它會列出傳回 4XX 或 5XX 錯誤回應碼的後端應用程式伺服器。使用 LIMIT 運算子來限制一次查詢的日誌數量。

```
SELECT
```

```
timestamp,  
elb_name,  
backend_ip,  
backend_response_code  
FROM elb_logs  
WHERE backend_response_code LIKE '4%' OR  
       backend_response_code LIKE '5%'  
LIMIT 100;
```

使用後續的查詢來將依後端 IP 地址和 Elastic Load Balancing 執行個體名稱分組的所有交易的回應時間加總。

```
SELECT sum(backend_processing_time) AS  
total_ms,  
elb_name,  
backend_ip  
FROM elb_logs WHERE backend_ip <> ''  
GROUP BY backend_ip, elb_name  
LIMIT 100;
```

如需詳細資訊，請參閱[使用 Athena 在 S3 中分析資料](#)。

## 查詢 Amazon CloudFront 日誌

您可以設定 Amazon CloudFront CDN，將網路分發存取日誌匯出至 Amazon 簡單儲存服務。您可以使用這些記錄來探索使用者所提供服務的 Web 屬性中的瀏覽模式 CloudFront。

在您開始查詢記錄檔之前，請在您偏好的分發上啟用 Web CloudFront 發行版存取記錄檔。如需詳細資訊，請參閱[Amazon CloudFront 開發人員指南中的存取日誌](#)。記下您儲存這些日誌的 Amazon S3 儲存貯體。

- [建立 CloudFront 標準記錄的資料表](#)
- [建立即 CloudFront 時記錄的資料表](#)
- [標準 CloudFront 記錄檔的查詢範例](#)

### 建立 CloudFront 標準記錄的資料表

#### Note

此程序適用於中的 Web 分發存取記錄檔 CloudFront。不適用於來自 RTMP 分佈的串流日誌。

## 為 CloudFront 標準記錄檔欄位建立表格的步驟

1. 複製下列範例 DDL 陳述式，並將其貼到 Athena 主控台查詢編輯器。範例陳述式使用 Amazon CloudFront 開發人員指南「[標準記錄檔欄位](#)」區段中所述的記錄檔欄位。為存放日誌的 Amazon S3 儲存貯體修改 LOCATION。如需有關使用查詢編輯器的資訊，請參閱[開始使用](#)。

此查詢會指 FIELDS TERMINATED BY '\t' 定 ROW FORMAT DELIMITED 並指出欄位由定位字元分隔。對於 ROW FORMAT DELIMITED，Athena 預設 [LazySimpleSerDe](#) 會使用。因為在 Athena 中它是保留字，所以使用反引號 (') 逸出 date 資料欄。如需相關資訊，請參閱[保留的關鍵字](#)。

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_standard_logs (  
  `date` DATE,  
  time STRING,  
  x_edge_location STRING,  
  sc_bytes BIGINT,  
  c_ip STRING,  
  cs_method STRING,  
  cs_host STRING,  
  cs_uri_stem STRING,  
  sc_status INT,  
  cs_referrer STRING,  
  cs_user_agent STRING,  
  cs_uri_query STRING,  
  cs_cookie STRING,  
  x_edge_result_type STRING,  
  x_edge_request_id STRING,  
  x_host_header STRING,  
  cs_protocol STRING,  
  cs_bytes BIGINT,  
  time_taken FLOAT,  
  x_forwarded_for STRING,  
  ssl_protocol STRING,  
  ssl_cipher STRING,  
  x_edge_response_result_type STRING,  
  cs_protocol_version STRING,  
  fle_status STRING,  
  fle_encrypted_fields INT,  
  c_port INT,  
  time_to_first_byte FLOAT,  
  x_edge_detailed_result_type STRING,  
  sc_content_type STRING,  
  sc_content_len BIGINT,  
  sc_range_start BIGINT,
```

```

    sc_range_end BIGINT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
TBLPROPERTIES ( 'skip.header.line.count'='2' )

```

2. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 `cloudfront_standard_logs` 資料表，讓其中的資料可供您發出查詢。

## 建立即 CloudFront 時記錄的資料表

### 建立 CloudFront 即時記錄檔欄位的表格

1. 複製下列範例 DDL 陳述式，並將其貼到 Athena 主控台查詢編輯器。範例陳述式使用 Amazon CloudFront 開發人員指南中「[即時日誌](#)」一節中所記錄的日誌檔欄位。為存放日誌的 Amazon S3 儲存貯體修改 LOCATION。如需有關使用查詢編輯器的資訊，請參閱[開始使用](#)。

此查詢會指 FIELDS TERMINATED BY '\t' 定 ROW FORMAT DELIMITED 並指出欄位由定位字元分隔。對於 ROW FORMAT DELIMITED，Athena 預設 [LazySimpleSerDe](#) 會使用。因為在 Athena 中它是保留字，所以使用反引號 (') 逸出 timestamp 資料欄。如需相關資訊，請參閱[保留的關鍵字](#)。

下列範例包含所有可用欄位。您可以註解掉或移除不需要的欄位。

```

CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_real_time_logs (
  `timestamp` STRING,
  c_ip STRING,
  time_to_first_byte BIGINT,
  sc_status BIGINT,
  sc_bytes BIGINT,
  cs_method STRING,
  cs_protocol STRING,
  cs_host STRING,
  cs_uri_stem STRING,
  cs_bytes BIGINT,
  x_edge_location STRING,
  x_edge_request_id STRING,
  x_host_header STRING,
  time_taken BIGINT,
  cs_protocol_version STRING,
  c_ip_version STRING,
  cs_user_agent STRING,

```

```
cs_referer STRING,  
cs_cookie STRING,  
cs_uri_query STRING,  
x_edge_response_result_type STRING,  
x_forwarded_for STRING,  
ssl_protocol STRING,  
ssl_cipher STRING,  
x_edge_result_type STRING,  
fle_encrypted_fields STRING,  
fle_status STRING,  
sc_content_type STRING,  
sc_content_len BIGINT,  
sc_range_start STRING,  
sc_range_end STRING,  
c_port BIGINT,  
x_edge_detailed_result_type STRING,  
c_country STRING,  
cs_accept_encoding STRING,  
cs_accept STRING,  
cache_behavior_path_pattern STRING,  
cs_headers STRING,  
cs_header_names STRING,  
cs_headers_count BIGINT,  
primary_distribution_id STRING,  
primary_distribution_dns_name STRING,  
origin_fbl STRING,  
origin_lbl STRING,  
asn STRING  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'  
TBLPROPERTIES ( 'skip.header.line.count'='2' )
```

2. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 `cloudfront_real_time_logs` 資料表，讓其中的資料可供您發出查詢。

### 標準 CloudFront 記錄檔的查詢範例

下列查詢會加總 2018 年 6 月 9 日至 6 月 11 日 CloudFront 之間所服務的位元組數目。以雙引號圍住日期欄名稱，因為它是保留字。

```
SELECT SUM(bytes) AS total_bytes
```

```
FROM cloudfront_standard_logs
WHERE "date" BETWEEN DATE '2018-06-09' AND DATE '2018-06-11'
LIMIT 100;
```

若要從查詢結果中除去重複的資料列 (例如，重複的空白資料列)，您可以使用 SELECT DISTINCT 陳述式，如下列範例所示。

```
SELECT DISTINCT *
FROM cloudfront_standard_logs
LIMIT 10;
```

## 其他資源

如需有關使用 Athena 查詢 CloudFront 記錄檔的詳細資訊，請參閱[AWS 大數據部落格](#)中的下列文章。

[使用 Amazon Athena 輕鬆查詢 AWS 服務 日誌](#) (2019 年 5 月 29 日)。

[大規模分析您的 Amazon CloudFront 存取日誌](#) (2018 年 12 月 21 日)。

[建置無伺服器架構 AWS Lambda，使用 Amazon 雅典娜和 Amazon 管理服務 Apache Flink 分析亞馬遜 CloudFront 存取日誌](#) (2017 年 5 月 26 日)。

## 查詢 AWS CloudTrail 記錄檔

AWS CloudTrail 是一種記錄 Amazon Web Services 帳戶的 AWS API 調用和事件的服務。

CloudTrail 日誌包括對您進行的任何 API 調用的詳細信息 AWS 服務，包括控制台。CloudTrail 產生加密的日誌檔案，並將其存放在 Amazon S3 中。如需詳細資訊，請參閱 [AWS CloudTrail 使用者指南](#)。

### Note

如果您想要對跨帳戶、區域和日期的 CloudTrail 事件資訊執行 SQL 查詢，請考慮使用 CloudTrail Lake。CloudTrail Lake 是建立追蹤的 AWS 替代方法，可將企業中的資訊彙總到單一、可搜尋的事件資料存放區。其不使用 Amazon S3 儲存貯體儲存，而是將事件儲存在資料湖中，進而允許更豐富、更快的查詢。您可以使用它來建立 SQL 查詢，以便跨組織、區域並在自訂的時間範圍內搜尋事件。由於您在 CloudTrail 主控台本身內執行 CloudTrail Lake 查詢，因此使用 CloudTrail Lake 不需要 Athena。如需詳細資訊，請參閱 [CloudTrail Lake](#) 文件。

搭配 CloudTrail 日誌使用 Athena 是強化 AWS 服務 活動分析的強大方法。例如，您可以使用查詢來識別趨勢，並依屬性 (例如來源 IP 地址或使用者) 進一步隔離活動。

常見的應用是使用 CloudTrail 日誌來分析操作活動，以確保安全性和合規性。如需詳細範例的詳細資訊，請參閱 AWS 大數據部落格文章：[使用和 Amazon Athena 分析安全性、合規性 AWS CloudTrail 和操作活動](#)。

您可以使用 Athena，指定日誌檔案的 LOCATION，直接從 Amazon S3 查詢這些日誌檔案。有以下兩種做法：

- 通過直接從 CloudTrail 控制台創建 CloudTrail 日誌文件表。
- 在 Athena 主控台中手動建立 CloudTrail 記錄檔的資料表。

## 主題

- [了解 CloudTrail 日誌和 Athena 表](#)
- [使用主 CloudTrail 控台為 CloudTrail 記錄建立 Athena 資料表](#)
- [使用手動磁碟分割在 Athena 中建立 CloudTrail 記錄資料表](#)
- [使用手動分割為整個組織的線索建立資料表](#)
- [使用分割區投影在 Athena 中建立 CloudTrail 日誌資料表](#)
- [查詢巢狀欄位](#)
- [查詢範例](#)
- [查詢 CloudTrail 記錄檔的提示](#)

## 了解 CloudTrail 日誌和 Athena 表

在開始創建表之前，您應該了解更多有關以 CloudTrail 及它如何存儲數據。無論您是從 CloudTrail 主控台還是從 Athena 建立資料表，這都能協助您建立所需的表格。

CloudTrail 以壓縮的 gzip 格式 (\*.json.gzip) 將記錄檔儲存為 JSON 文字檔案。記錄檔的位置取決於您設定追蹤的方式、要記錄的 AWS 區域 或區域以及其他因素。

如需有關日誌存放位置、JSON 結構、日誌檔案內容的詳細資訊，請參閱《[AWS CloudTrail 使用者指南](#)》中的下列主題：

- [尋找您的 CloudTrail 記錄檔](#)
- [CloudTrail 記錄檔範例](#)
- [CloudTrail 記錄內容](#)

- [CloudTrail 事件參考](#)

若要收集日誌並將其儲存到 Amazon S3，請 CloudTrail 從 AWS Management Console。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的[建立追蹤](#)。

請記下您儲存日誌的目的地 Amazon S3 儲存貯體。將LOCATION子句取代為 CloudTrail 記錄檔位置的路徑，以及要使用的物件集。範例中使用特定帳戶日誌的 LOCATION 值，但您可以使用適合您應用程式的具體程度。

例如：

- 若要分析多個帳戶的資料，您可以復原 LOCATION 指標，使用 LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/' 來指定所有 AWSLogs。
- 若要分析特定日期、帳戶和區域的資料，請使用 LOCATION 's3://DOC-EXAMPLE-BUCKET/123456789012/CloudTrail/us-east-1/2016/03/14/'。

當您使用 Athena 進行查詢時，使用物件階層的最高層級可給您最大的靈活性。

使用主 CloudTrail 控制台為 CloudTrail 記錄建立 Athena 資料表

您可以建立非分割的 Athena 資料表，以直接從 CloudTrail 主控台查詢 CloudTrail 記錄檔。您必須使用具有足夠權限在 Athena 中建立資料表的角色登入，才能從 CloudTrail 主控台建立 Athena 表格。

 Note

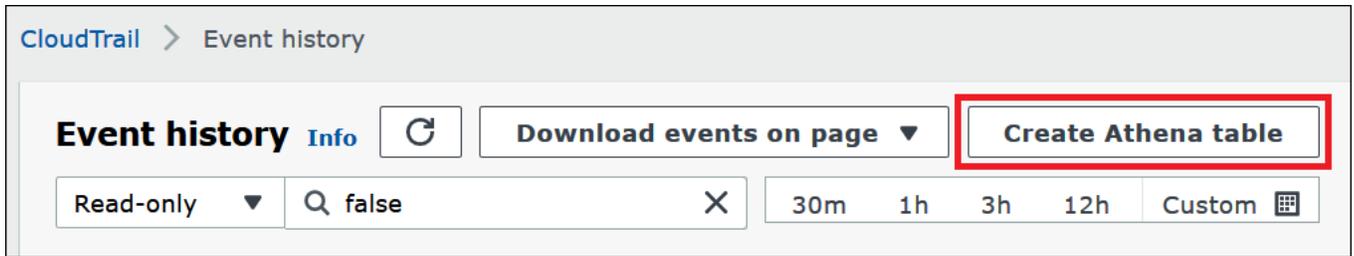
您無法使用主 CloudTrail 控制台為組織追蹤記錄建立 Athena 表格。而是要使用 Athena 主控台手動建立資料表，以便指定正確的儲存位置。如需有關組織線索的資訊，請參閱《AWS CloudTrail 使用者指南》中的[為組織建立追蹤](#)。

- 如需有關設定 Athena 許可的資訊，請參閱 [設定](#)。
- 如需建立包含分割區之資料表的相關資訊，請參閱[使用手動磁碟分割在 Athena 中建立 CloudTrail 記錄資料表](#)。

使用 CloudTrail 主控台為 CloudTrail 追蹤建立 Athena 表格

1. [請在以下位置開啟 CloudTrail 主控台](https://console.aws.amazon.com/cloudtrail/)。 <https://console.aws.amazon.com/cloudtrail/>
2. 在導覽窗格中，選擇 Event history (事件歷史記錄)。

### 3. 選擇 Create Athena table (建立 Athena 資料表)。



### 4. 對於 Storage location (儲存位置), 請使用向下箭頭選取存放日誌檔案的 Amazon S3 儲存貯體, 以供查詢線索。

#### Note

若要尋找與追蹤相關聯的儲存貯體名稱, 請在 CloudTrail 導覽窗格中選擇 Trail, 然後檢視追蹤的 S3 儲存貯體欄。若要查看儲存貯體的 Amazon S3 位置, 請在 S3 bucket (S3 儲存貯體) 資料欄中選擇儲存貯體的連結。這會開啟 Amazon S3 主控台到儲存 CloudTrail 貯體位置。

### 5. 選擇 Create Table (建立資料表)。建立的資料表會使用包含 Amazon S3 儲存貯體名稱的預設名稱。

使用手動磁碟分割在 Athena 中建立 CloudTrail 記錄資料表

您可以在 Athena 主控台中手動建立 CloudTrail 記錄檔的表格, 然後在 Athena 中執行查詢。

使用 Athena 主控台為 CloudTrail 追蹤建立 Athena 桌子

#### 1. 將下列 DDL 陳述式複製並貼上至 Athena 主控台查詢編輯器。

```
CREATE EXTERNAL TABLE cloudtrail_logs (  
  eventversion STRING,  
  useridentity STRUCT<  
    type:STRING,  
    principalid:STRING,  
    arn:STRING,  
    accountid:STRING,  
    invokedby:STRING,  
    accesskeyid:STRING,  
    userName:STRING,  
  sessioncontext:STRUCT<  
    attributes:STRUCT<
```

```
                mfaauthenticated:STRING,
                creationdate:STRING>,
    sessionissuer:STRUCT<
        type:STRING,
        principalId:STRING,
        arn:STRING,
        accountId:STRING,
        userName:STRING>,
    ec2RoleDelivery:string,
    webIdFederationData: STRUCT<
        federatedProvider: STRING,
        attributes: map<string,string>
    >
>
>,
eventtime STRING,
eventsourcesource STRING,
eventname STRING,
awsregion STRING,
sourceipaddress STRING,
useragent STRING,
errorcode STRING,
errormessage STRING,
requestparameters STRING,
responseelements STRING,
additionaleventdata STRING,
requestid STRING,
eventid STRING,
resources ARRAY<STRUCT<
    arn:STRING,
    accountid:STRING,
    type:STRING>>,
eventtype STRING,
apiversion STRING,
readonly STRING,
recipientaccountid STRING,
serviceeventdetails STRING,
shareeventid STRING,
vpcendpointid STRING,
eventCategory STRING,
tlsDetails struct<
    tlsVersion:string,
    cipherSuite:string,
    clientProvidedHostHeader:string>
```

```

)
PARTITIONED BY (region string, year string, month string, day string)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/Account_ID/CloudTrail/';

```

### Note

我們建議使用範例中 `org.apache.hive.hcatalog.data.JsonSerDe` 顯示的。雖然 `com.amazon.emr.hive.serde.CloudTrailSerde` 存在，但它目前不處理某些較新的 CloudTrail 字段。

- (選用) 移除資料表不再需要的任何欄位。如果您只需要讀取特定的資料欄集，則您的資料表定義可排除其他資料欄。
- 修改 `s3://DOC-EXAMPLE-BUCKET/AWSLogs/Account_ID/CloudTrail/` 來指向包含日誌資料的 Amazon S3 儲存貯體。
- 確認列出的欄位正確。如需有關 CloudTrail 記錄中欄位完整清單的詳細資訊，請參閱 [CloudTrail 記錄內容](#)。

步驟 1 中的範例 CREATE TABLE 陳述式會使用 [蜂巢 SerDe](#)。在此範例中，欄位 `requestparametersresponseelements`、和會列 `additional eventdata` 為查詢 STRING 中的類型，但是在 JSON 中使用的 STRUCT 資料類型。因此，若要取得這些欄位的資料，請使用 JSON\_EXTRACT 函數。如需詳細資訊，請參閱 [the section called “從字符串中提取 JSON 數據”](#)。為了改善效能，此範例會依據 AWS 區域、年、月和日來分割資料。

- 在 Athena 主控台執行 CREATE TABLE 陳述式。
- 使用 [ALTER TABLE ADD PARTITION](#) 命令載入分割區，以便進行查詢，如下列範例所示。

```

ALTER TABLE table_name ADD
  PARTITION (region='us-east-1',
            year='2019',
            month='02',
            day='01')
  LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/Account_ID/CloudTrail/us-east-1/2019/02/01/'

```

## 使用手動分割為整個組織的線索建立資料表

若要在 Athena 中為整個組織的 CloudTrail 記錄檔建立表格，請遵循中的步驟[使用手動磁碟分割在 Athena 中建立 CloudTrail 記錄資料表](#)，但要進行下列程序中所述的修改。

若要為整個組織的 CloudTrail 日誌建立 Athena 表格

1. 在 CREATE TABLE 陳述式中，修改 LOCATION 子句以加入組織 ID，如下列範例所示：

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/Account_ID/CloudTrail/'
```

2. 在 PARTITIONED BY 子句中，以字串形式新增帳戶 ID 項目，如下列範例所示：

```
PARTITIONED BY (account string, region string, year string, month string, day string)
```

上述步驟的合併結果如以下範例所示：

```
...  
  
PARTITIONED BY (account string, region string, year string, month string, day string)  
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'  
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'  
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/Account_ID/CloudTrail/'
```

3. 在 ALTER TABLE 陳述式中的 ADD PARTITION 子句內加入帳戶 ID，如下列範例所示：

```
ALTER TABLE table_name ADD  
PARTITION (account='111122223333',  
region='us-east-1',  
year='2022',  
month='08',  
day='08')
```

4. 在 ALTER TABLE 陳述式中的 LOCATION 子句內加入組織 ID、帳戶 ID，以及您要新增的分割區，如下列範例所示：

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/Account_ID/CloudTrail/us-east-1/2022/08/08/'
```

上述步驟的合併結果如以下 ALTER TABLE 陳述式範例所示：

```
ALTER TABLE table_name ADD
PARTITION (account='111122223333',
region='us-east-1',
year='2022',
month='08',
day='08')
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/111122223333/CloudTrail/
us-east-1/2022/08/08/'
```

使用分割區投影在 Athena 中建立 CloudTrail 日誌資料表

因為 CloudTrail 記錄檔具有已知的結構，您可以事先指定其資料分割配置，因此您可以使用 Athena 分割區投影功能，縮短查詢執行階段並自動化分割區管理。分割區投影會在新增資料時自動新增分割區。因此您無需使用 ALTER TABLE ADD PARTITION 手動新增分割區。

下列範例 CREATE TABLE 陳述式會自動在指定日期的 CloudTrail 記錄上使用分割區投影，直到目前為止 AWS 區域。在 LOCATION 和 storage.location.template 子句中，請以對應的相同值取代 **##**、**account-id** 和 **aws-region** 預留位置。對於 projection.timestamp.range，請用您要使用的開始日期取代 **2020/01/01**。成功執行查詢之後，您可以查詢資料表。您無須執行 ALTER TABLE ADD PARTITION 就能載入分割區。

```
CREATE EXTERNAL TABLE cloudtrail_logs_pp(
  eventVersion STRING,
  userIdentity STRUCT<
    type: STRING,
    principalId: STRING,
    arn: STRING,
    accountId: STRING,
    invokedBy: STRING,
    accessKeyId: STRING,
    userName: STRING,
    sessionContext: STRUCT<
      attributes: STRUCT<
        mfaAuthenticated: STRING,
        creationDate: STRING>,
      sessionIssuer: STRUCT<
        type: STRING,
        principalId: STRING,
```

```

        arn: STRING,
        accountId: STRING,
        userName: STRING>,
        ec2RoleDelivery:string,
        webIdFederationData: STRUCT<
            federatedProvider: STRING,
            attributes: map<string,string>
        >
    >
>,
eventTime STRING,
eventSource STRING,
eventName STRING,
awsRegion STRING,
sourceIpAddress STRING,
userAgent STRING,
errorCode STRING,
errorMessage STRING,
requestparameters STRING,
responseelements STRING,
additionaleventdata STRING,
requestId STRING,
eventId STRING,
readOnly STRING,
resources ARRAY<STRUCT<
    arn: STRING,
    accountId: STRING,
    type: STRING>>,
eventType STRING,
apiVersion STRING,
recipientAccountId STRING,
serviceEventDetails STRING,
sharedEventID STRING,
vpcendpointid STRING,
eventCategory STRING,
tlsDetails struct<
    tlsVersion:string,
    cipherSuite:string,
    clientProvidedHostHeader:string>
)
PARTITIONED BY (
    `timestamp` string)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'

```

```

OUTPUTFORMAT 'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/AWSLogs/account-id/CloudTrail/aws-region'
TBLPROPERTIES (
  'projection.enabled'='true',
  'projection.timestamp.format'='yyyy/MM/dd',
  'projection.timestamp.interval'='1',
  'projection.timestamp.interval.unit'='DAYS',
  'projection.timestamp.range'='2020/01/01,NOW',
  'projection.timestamp.type'='date',
  'storage.location.template'='s3://DOC-EXAMPLE-BUCKET/AWSLogs/account-id/
CloudTrail/aws-region/${timestamp}')

```

如需有關分割區投影的詳細資訊，請參閱[使用 Amazon Athena 進行分割區投影](#)。

### 查詢巢狀欄位

由於 `userIdentity` 和 `resources` 欄位是巢狀資料類型，查詢它們需要特殊的處理。

`userIdentity` 物件由巢狀 STRUCT 類型組成。如以下範例所示，可以使用點來分隔欄位，以查詢這些類型：

```

SELECT
  eventsource,
  eventname,
  useridentity.sessioncontext.attributes.creationdate,
  useridentity.sessioncontext.sessionissuer.arn
FROM cloudtrail_logs
WHERE useridentity.sessioncontext.sessionissuer.arn IS NOT NULL
ORDER BY eventsource, eventname
LIMIT 10

```

`resources` 欄位是 STRUCT 物件的陣列。請對這些陣列使用 `CROSS JOIN UNNEST`，解除陣列巢狀，以便於查詢其物件。

以下範例會傳回資源 ARN 以 `example/datafile.txt` 結尾的所有資料行。為了可讀性，`replace` 函數會從 ARN 移除初始 `arn:aws:s3:::` 子字串。

```

SELECT
  awsregion,
  replace(unnested.resources_entry.ARN, 'arn:aws:s3:::') as s3_resource,

```

```
    eventname,  
    eventtime,  
    useragent  
FROM cloudtrail_logs t  
CROSS JOIN UNNEST(t.resources) unnested (resources_entry)  
WHERE unnested.resources_entry.ARN LIKE '%example/datafile.txt'  
ORDER BY eventtime
```

下列範例為 DeleteBucket 事件的查詢範例。該查詢會從 resources 物件擷取儲存貯體的名稱及儲存貯體隸屬的帳戶 ID。

```
SELECT  
    awsregion,  
    replace(unnested.resources_entry.ARN, 'arn:aws:s3:::') as deleted_bucket,  
    eventtime AS time_deleted,  
    useridentity.username,  
    unnested.resources_entry.accountid as bucket_acct_id  
FROM cloudtrail_logs t  
CROSS JOIN UNNEST(t.resources) unnested (resources_entry)  
WHERE eventname = 'DeleteBucket'  
ORDER BY eventtime
```

如需有關解除巢狀的詳細資訊，請參閱[篩選陣列](#)。

## 查詢範例

下列範例顯示查詢的一部分，此部分會傳回針對 CloudTrail 事件記錄檔建立的資料表中的所有匿名 (未簽署) 要求。此查詢選擇 useridentity.accountid 是匿名且未指定 useridentity.arn 的那些請求：

```
SELECT *  
FROM cloudtrail_logs  
WHERE  
    eventsource = 's3.amazonaws.com' AND  
    eventname in ('GetObject') AND  
    useridentity.accountid = 'anonymous' AND  
    useridentity.arn IS NULL AND  
    requestparameters LIKE '%[your bucket name ]%';
```

如需詳細資訊，請參閱 AWS 大數據部落格文章[使用和 Amazon Athena 分析安全性、合規性 AWS CloudTrail 和操作活動](#)。

## 查詢 CloudTrail 記錄檔的提示

若要瀏覽記 CloudTrail 錄檔資料，請使用下列秘訣：

- 查詢日誌之前，確認您的日誌資料表看起來和 [the section called “使用手動磁碟分割在 Athena 中建立 CloudTrail 記錄資料表”](#) 中的相同。如果它不是第一個資料表，使用下列命令刪除現有的資料表：DROP TABLE cloudtrail\_logs。
- 捨棄現有資料表後，再重新建立。如需詳細資訊，請參閱 [使用手動磁碟分割在 Athena 中建立 CloudTrail 記錄資料表](#)。

確認 Athena 查詢中列出的欄位正確。如需有關 CloudTrail 記錄中欄位完整清單的資訊，請參閱 [CloudTrail 記錄內容](#)。

如果您的查詢包含 JSON 格式的欄位，例如 STRUCT，請從 JSON 擷取資料。如需詳細資訊，請參閱 [從字符串中提取 JSON 數據](#)。

針對您的 CloudTrail 表格發出查詢的一些建議：

- 首先，查看哪些使用者呼叫了哪些 API 操作以及從哪些來源 IP 地址進行呼叫。
- 以下列基本 SQL 查詢當做您的範本。將查詢貼到 Athena 主控台然後執行。

```
SELECT
  useridentity.arn,
  eventname,
  sourceipaddress,
  eventtime
FROM cloudtrail_logs
LIMIT 100;
```

- 修改查詢，進一步探索您的資料。
- 為了改善效能，加入 LIMIT 子句來傳回資料列的指定部分。

## 查詢 Amazon EMR 日誌

在 Amazon EMR 上執行的 Amazon EMR 和大數據應用程式會產生日誌檔案。日誌檔案會寫入 [主節點](#)，您也可以將 Amazon EMR 設定為自動將日誌檔存檔到 Amazon S3。您可以使用 Amazon Athena 來查詢這些日誌，以識別應用程式和叢集的事件和趨勢。如需有關 Amazon EMR 中的日誌檔案類型和將其儲存至 Amazon S3 的詳細資訊，請參閱《Amazon EMR 管理指南》中的 [檢視日誌檔案](#)。

## 根據 Amazon EMR 日誌檔案建立和查詢基本資料表

以下範例會根據儲存至 `s3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/elasticmapreduce/` 的日誌檔建立基本資料表 `myemrlogs`。在以下範例中使用的 Amazon S3 位置會反映由 Amazon Web Services 帳戶 `123456789012` 在區域 `us-west-2` 中建立之 EMR 叢集的預設日誌位置的模式。如果您使用自訂位置，則模式為 `s3://DOC-EXAMPLE-BUCKET/ClusterID #`。

如需建立分割的資料表以盡可能提升查詢效能並減少資料傳輸的詳細資訊，請參閱 [根據 Amazon EMR 日誌建立和查詢分割的資料表](#)。

```
CREATE EXTERNAL TABLE `myemrlogs` (
  `data` string COMMENT 'from deserializer')
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LINES TERMINATED BY '\n'
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6'
```

以下範例查詢可在上一個範例建立的 `myemrlogs` 資料表上執行。

Example – 查詢 ERROR、WARN、INFO、EXCEPTION、FATAL 或 DEBUG 發生次數的步驟日誌

```
SELECT data,
  "$PATH"
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 's-86URH188Z6B1')
  AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;
```

Example – 查詢 ERROR、WARN、INFO、EXCEPTION、FATAL 或 DEBUG 的特定執行個體日誌 `i-00b3c0a839ece0a9c`

```
SELECT "data",
  "$PATH" AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 'i-00b3c0a839ece0a9c')
  AND regexp_like("$PATH", 'state')
```

```
AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;
```

Example – 查詢 ERROR、WARN、INFO、EXCEPTION、FATAL 或 DEBUG 的 Presto 應用程式日誌

```
SELECT "data",
       "$PATH" AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 'presto')
      AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;
```

Example – 查詢 ERROR、WARN、INFO、EXCEPTION、FATAL 或 DEBUG 的 Namenode 應用程式日誌

```
SELECT "data",
       "$PATH" AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 'namenode')
      AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;
```

Example – 依日期和小時查詢 ERROR、WARN、INFO、EXCEPTION、FATAL 或 DEBUG 的所有日誌

```
SELECT distinct("$PATH") AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", '2019-07-23-10')
      AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;
```

根據 Amazon EMR 日誌建立和查詢分割的資料表

這些範例使用相同的日誌位置來建立 Athena 資料表，但資料表會進行分割，然後為每個日誌位置建立分割區。如需詳細資訊，請參閱[在 Athena 中分割資料](#)。

以下查詢會建立名為 `mypartitionedemrlogs` 的分割資料表：

```
CREATE EXTERNAL TABLE `mypartitionedemrlogs` (
  `data` string COMMENT 'from deserializer')
  partitioned by (logtype string)
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY '|'
  LINES TERMINATED BY '\n'
```

```
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6'
```

以下查詢陳述式接著會根據 Amazon EMR 在 Amazon S3 中所建立之不同日誌類型的子目錄，建立資料表分割區：

```
ALTER TABLE mypartitionedemrlogs ADD
  PARTITION (logtype='containers')
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/containers/'
```

```
ALTER TABLE mypartitionedemrlogs ADD
  PARTITION (logtype='hadoop-mapreduce')
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/hadoop-mapreduce/'
```

```
ALTER TABLE mypartitionedemrlogs ADD
  PARTITION (logtype='hadoop-state-pusher')
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/hadoop-state-pusher/'
```

```
ALTER TABLE mypartitionedemrlogs ADD
  PARTITION (logtype='node')
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/node/'
```

```
ALTER TABLE mypartitionedemrlogs ADD
  PARTITION (logtype='steps')
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/steps/'
```

建立分割區後，您可以在資料表上執行 SHOW PARTITIONS 查詢，以便確認：

```
SHOW PARTITIONS mypartitionedemrlogs;
```

以下範例示範特定日誌項目的查詢會使用上述範例所建立的資料表和分割區。

Example – 查詢 ERROR 或 WARN 的容器分割區中的應用程式 application\_1561661818238\_0002 日誌

```
SELECT data,
       "$PATH"
FROM "default"."mypartitionedemrlogs"
WHERE logtype='containers'
      AND regexp_like("$PATH",'application_1561661818238_0002')
      AND regexp_like(data, 'ERROR|WARN') limit 100;
```

Example – 查詢任務 job\_1561661818238\_0004 的 hadoop-Mapreduce 分割區和失敗的 Reduce 次數

```
SELECT data,
       "$PATH"
FROM "default"."mypartitionedemrlogs"
WHERE logtype='hadoop-mapreduce'
      AND regexp_like(data,'job_1561661818238_0004|Failed Reduces') limit 100;
```

Example – 在查詢 ID 056e0609-33e1-4611-956c-7a31b42d2663 的節點分割區中查詢 Hive 日誌

```
SELECT data,
       "$PATH"
FROM "default"."mypartitionedemrlogs"
WHERE logtype='node'
      AND regexp_like("$PATH",'hive')
      AND regexp_like(data,'056e0609-33e1-4611-956c-7a31b42d2663') limit 100;
```

Example – 在應用程式 1567660019320\_0001\_01\_000001 的節點分割區中查詢 resourcemanager 日誌

```
SELECT data,
       "$PATH"
FROM "default"."mypartitionedemrlogs"
WHERE logtype='node'
      AND regexp_like(data,'resourcemanager')
      AND regexp_like(data,'1567660019320_0001_01_000001') limit 100
```

## 查詢 AWS Global Accelerator 流程記錄

您可以使用建立 AWS Global Accelerator 加速器，將網路流量引導至 AWS 全域網路的最佳端點。如需有關全域加速器的詳細資訊，請參閱[什麼是 AWS Global Accelerator](#)。

Global Accelerator 流程日誌可讓您擷取加速器中進出網路界面的 IP 地址流量相關資訊。流程日誌資料會發佈至 Amazon S3，您可以在其中擷取和檢視資料。如需詳細資訊，請參閱 [AWS Global Accelerator 中的流程日誌](#)。

您可以使用 Athena 建立在 Amazon S3 中指定其位置的資料表來查詢 Global Accelerator 流程日誌。

若要建立 Global Accelerator 流程日誌的資料表

1. 複製下列 DDL 陳述式，並將其貼到 Athena 主控台。此查詢會指定 ROW FORMAT 分隔，並省略指定 a [SerDe](#)，這表示查詢會使用 [LazySimpleSerDe](#)。在這個查詢中，欄位以空格結尾。

```
CREATE EXTERNAL TABLE IF NOT EXISTS aga_flow_logs (  
  version string,  
  account string,  
  acceleratorid string,  
  clientip string,  
  clientport int,  
  gip string,  
  gipport int,  
  endpointip string,  
  endpointport int,  
  protocol string,  
  ipaddresstype string,  
  numpackets bigint,  
  numbytes int,  
  starttime int,  
  endtime int,  
  action string,  
  logstatus string,  
  agasourceip string,  
  agasourceport int,  
  endpointregion string,  
  agaregion string,  
  direction string  
)  
PARTITIONED BY (dt string)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ' '  
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/account_id/globalaccelerator/  
region/'  
TBLPROPERTIES ("skip.header.line.count"="1");
```

2. 修改 LOCATION 值來指向包含日誌資料的 Amazon S3 儲存貯體。

```
's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/account_id/globalaccelerator/region_code/'
```

3. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 `aga_flow_logs` 資料表，讓其中的資料可供查詢。
4. 建立分割區以讀取資料，如下列查詢範例所示。該查詢會針對指定日期建立單一分割區。將預留位置換成日期和位置。

```
ALTER TABLE aga_flow_logs  
ADD PARTITION (dt='YYYY-MM-dd')  
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/account_id/  
globalaccelerator/region_code/YYYY/MM/dd';
```

## AWS Global Accelerator 流程記錄查詢範例

### Example – 列出透過特定節點傳遞的請求

下列範例查詢列出了透過 LHR 節點傳遞的要求。使用 LIMIT 運算子來一次限制查詢的日誌數量。

```
SELECT  
  clientip,  
  agaregion,  
  protocol,  
  action  
FROM  
  aga_flow_logs  
WHERE  
  agaregion LIKE 'LHR%'  
LIMIT  
  100;
```

### Example – 列出了接收最多 HTTPS 請求的端點 IP 地址

若要查看哪些端點 IP 位址正在接收最多的 HTTPS 要求數目，請使用下列查詢。此查詢會計算 HTTPS 連接埠 443 所接收的封包數、依目的地 IP 地址分組，然後傳回前 10 個 IP 地址。

```
SELECT  
  SUM(numpackets) AS packetcount,  
  endpointip  
FROM  
  aga_flow_logs
```

```
WHERE
  endpointport = 443
GROUP BY
  endpointip
ORDER BY
  packetcount DESC
LIMIT
  10;
```

## 查詢 Amazon GuardDuty 發現

[Amazon GuardDuty](#) 是一種安全監控服務，可協助您識別 AWS 環境中未預期且可能的未經授權或惡意活動。當它偵測到非預期和潛在惡意活動時，GuardDuty 會產生安全 [發現結果](#)，您可以匯出到 Amazon S3 進行儲存和分析。將問題清單匯出至 Amazon S3 之後，您可以使用 Athena 來查詢它們。本文說明如何在 Athena 中為您的 GuardDuty 發現項目建立資料表並進行查詢。

有關 Amazon 的更多信息 GuardDuty，請參閱 [Amazon GuardDuty 用戶指南](#)。

### 必要條件

- 啟用將發現項目匯出到 Amazon S3 的 GuardDuty 功能。如需步驟，請參閱 Amazon GuardDuty 使用者指南中的 [匯出發現項目](#)。

在 Athena 建立用於 GuardDuty 發現項目的表格

若要從 Athena 查詢您的 GuardDuty 發現項目，您必須為這些資料表建立資料表。

若要在 Athena 建立用於 GuardDuty 發現項目的表格

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 將下列 DDL 陳述式複製到 Athena 主控台。修改中的值 LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/*account-id*/GuardDuty/' 以指向 Amazon S3 中的 GuardDuty 發現項目。

```
CREATE EXTERNAL TABLE `gd_logs` (  
  `schemaversion` string,  
  `accountid` string,  
  `region` string,  
  `partition` string,  
  `id` string,  
  `arn` string,
```

```

`type` string,
`resource` string,
`service` string,
`severity` string,
`createdat` string,
`updatedat` string,
`title` string,
`description` string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/account-id/GuardDuty/'
TBLPROPERTIES ('has_encrypted_data'='true')

```

### Note

SerDe 預期每個 JSON 文件都位於單行文字上，且記錄中欄位之間沒有行終止字元。如果 JSON 文本是漂亮的打印格式，您可能會收到一條錯誤消息，如 HIVE\_CURSOR\_ERROR：行不是有效的 JSON 對象或 HIVE\_CURSOR\_ERROR:: 意外 JsonParseException end-of-input：當您嘗試在創建表後查詢對象的預期關閉標記。如需詳細資訊，請參閱上 GitHub 的 OpenX SerDe 文件中的 [JSON 資料檔案](#)。

3. 在 Athena 主控台中執行查詢來註冊 gd\_logs 資料表。查詢完成時，您就可以從 Athena 查詢問題清單。

## 查詢範例

下列範例顯示如何查詢 Athena 的 GuardDuty 發現項目。

### Example – DNS 資料外洩

下列查詢會傳回可能透過 DNS 查詢外洩資料之 Amazon EC2 執行個體的相關資訊。

```

SELECT
  title,
  severity,
  type,
  id AS FindingID,
  accountid,
  region,
  createdat,
  updatedat,
  json_extract_scalar(service, '$.count') AS Count,

```

```
json_extract_scalar(resource, '$.instancedetails.instanceid') AS InstanceID,  
json_extract_scalar(service, '$.action.actiontype') AS DNS_ActionType,  
json_extract_scalar(service, '$.action.dnsrequestaction.domain') AS DomainName,  
json_extract_scalar(service, '$.action.dnsrequestaction.protocol') AS protocol,  
json_extract_scalar(service, '$.action.dnsrequestaction.blocked') AS blocked  
FROM gd_logs  
WHERE type = 'Trojan:EC2/DNSDataExfiltration'  
ORDER BY severity DESC
```

## Example – 未經授權的 IAM 使用者存取

下列查詢會傳回所有區域的 IAM 委託人的所有 UnauthorizedAccess:IAMUser 問題清單類型。

```
SELECT title,  
       severity,  
       type,  
       id,  
       accountid,  
       region,  
       createdat,  
       updatedat,  
       json_extract_scalar(service, '$.count') AS Count,  
       json_extract_scalar(resource, '$.accesskeydetails.username') AS IAMPrincipal,  
       json_extract_scalar(service, '$.action.awsapicallaction.api') AS  
APIActionCalled  
FROM gd_logs  
WHERE type LIKE '%UnauthorizedAccess:IAMUser%'  
ORDER BY severity desc;
```

## 查詢發 GuardDuty 現項目的提示

當您建立查詢時，請記住以下幾點。

- 若要從巢狀 JSON 欄位中提取資料，請使用 Presto `json_extract` 或 `json_extract_scalar` 函數。如需詳細資訊，請參閱[從字符串中提取 JSON 數據](#)。
- 確保 JSON 欄位中的所有字元都是小寫。
- 如需下載查詢結果的相關資訊，請參閱[使用 Athena 主控台下載查詢結果檔案](#)。

## 查詢 AWS Network Firewall 記錄檔

AWS Network Firewall 這是一項受管服務，可用來為 Amazon Virtual Private Cloud 執行個體部署基本網路保護。AWS Network Firewall 與共同運作，以 AWS Firewall Manager 使您可以根據 AWS Network Firewall 規則建立原則，然後將這些原則集中套用至 VPC 和帳戶。如需有關的更多資訊 AWS Network Firewall，請參閱[AWS Network Firewall](#)。

您可以為轉寄至防火牆的可設定狀態規則引擎的流量設定 AWS Network Firewall 記錄。記錄功能提供詳細的網路流量資訊，包括有狀態引擎接收封包的時間、詳細的封包資訊，以及對封包採取的任何有狀態規則動作。日誌會發佈至您已設定的日誌目的地，其中您可以擷取並檢視它們。如需詳細資訊，請參閱《AWS Network Firewall 開發人員指南》中的[記錄來自 AWS Network Firewall 的網路流量](#)。

### 建立警示記錄的表格

1. 修改下列範例 DDL 陳述式，以符合警示日誌的結構。您可能需要更新陳述式，以加入最新版本日誌的資料欄。如需詳細資訊，請參閱《AWS Network Firewall 開發人員指南》中的[防火牆日誌內容](#)。

```
CREATE EXTERNAL TABLE network_firewall_alert_logs (  
  firewall_name string,  
  availability_zone string,  
  event_timestamp string,  
  event struct<  
    timestamp:string,  
    flow_id:bigint,  
    event_type:string,  
    src_ip:string,  
    src_port:int,  
    dest_ip:string,  
    dest_port:int,  
    proto:string,  
    app_proto:string,  
    tls_inspected:boolean,  
    alert:struct<  
      alert_id:string,  
      alert_type:string,  
      action:string,  
      signature_id:int,  
      rev:int,  
      signature:string,  
      category:string,  
      severity:int,
```

```

    rule_name:string,
    alert_name:string,
    alert_severity:string,
    alert_description:string,
    file_name:string,
    file_hash:string,
    packet_capture:string,
    reference_links:array<string>
  >,
  src_country:string,
  dest_country:string,
  src_hostname:string,
  dest_hostname:string,
  user_agent:string,
  url:string
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_to_alert_logs_folder/';

```

2. 修改子LOCATION句，為 Amazon S3 中的日誌指定資料夾。
3. 在 Athena 查CREATE TABLE詢編輯器中執行查詢。查詢完成後，Athena 會註冊資料network\_firewall\_alert\_logs表，讓它指向的資料準備好進行查詢。

## 警示記錄查詢範例

本節中的警示記錄查詢範例會篩選其中執行 TLS 檢查之警示嚴重性等級為 2 或更高的事件。

查詢會使用別名來建立顯示資料行所屬的struct輸出資料行標題。例如，欄event.alert.category位的欄標題event\_alert\_category不只是category。若要進一步自訂欄名稱，您可以修改別名以符合您的偏好設定。例如，您可以使用底線或其他分隔符號來分隔struct名稱和欄位名稱。

請記得根據struct資料表定義以及查詢結果中要的欄位來修改資料行名稱和參照。

```

SELECT
  firewall_name,
  availability_zone,
  event_timestamp,
  event.timestamp AS event_timestamp,
  event.flow_id AS event_flow_id,
  event.event_type AS event_type,

```

```
event.src_ip AS event_src_ip,  
event.src_port AS event_src_port,  
event.dest_ip AS event_dest_ip,  
event.dest_port AS event_dest_port,  
event.proto AS event_protol,  
event.app_proto AS event_app_proto,  
event.tls_inspected AS event_tls_inspected,  
event.alert.alert_id AS event_alert_alert_id,  
event.alert.alert_type AS event_alert_alert_type,  
event.alert.action AS event_alert_action,  
event.alert.signature_id AS event_alert_signature_id,  
event.alert.rev AS event_alert_rev,  
event.alert.signature AS event_alert_signature,  
event.alert.category AS event_alert_category,  
event.alert.severity AS event_alert_severity,  
event.alert.rule_name AS event_alert_rule_name,  
event.alert.alert_name AS event_alert_alert_name,  
event.alert.alert_severity AS event_alert_alert_severity,  
event.alert.alert_description AS event_alert_alert_description,  
event.alert.file_name AS event_alert_file_name,  
event.alert.file_hash AS event_alert_file_hash,  
event.alert.packet_capture AS event_alert_packet_capture,  
event.alert.reference_links AS event_alert_reference_links,  
event.src_country AS event_src_country,  
event.dest_country AS event_dest_country,  
event.src_hostname AS event_src_hostname,  
event.dest_hostname AS event_dest_hostname,  
event.user_agent AS event_user_agent,  
event.url AS event_url  
FROM  
  network_firewall_alert_logs  
WHERE  
  event.alert.severity >= 2  
  AND event.tls_inspected = true  
LIMIT 10;
```

## 建立網路流程記錄的資料表

1. 修改下列範例 DDL 陳述式，以符合網路流量記錄檔的結構。您可能需要更新陳述式，以加入最新版本日誌的資料欄。如需詳細資訊，請參閱《AWS Network Firewall 開發人員指南》中的[防火牆日誌內容](#)。

```
CREATE EXTERNAL TABLE network_firewall_netflow_logs (
```

```

firewall_name string,
availability_zone string,
event_timestamp string,
event struct<
  timestamp:string,
  flow_id:bigint,
  event_type:string,
  src_ip:string,
  src_port:int,
  dest_ip:string,
  dest_port:int,
  proto:string,
  app_proto:string,
  netflow:struct<
    pkts:int,
    bytes:bigint,
    start:string,
    `end`:string,
    age:int,
    min_ttl:int,
    max_ttl:int,
    tcp_flags:struct<
      syn:boolean,
      fin:boolean,
      rst:boolean,
      psh:boolean,
      ack:boolean,
      urg:boolean
    >,
    tls_inspected:boolean
  >
>
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_to_netflow_logs_folder/';

```

2. 修改子LOCATION句，為 Amazon S3 中的日誌指定資料夾。
3. 在 Athena CREATE TABLE 查詢編輯器中執行查詢。查詢完成後，Athena 會註冊資料network\_firewall\_netflow\_logs表，讓它指向的資料準備好進行查詢。

## 網路流量記錄檔範例查詢

本節中的範例 netflow 記錄查詢會篩選執行 TLS 檢查的事件。

查詢會使用別名來建立顯示資料行所屬的struct輸出資料行標題。例如，欄event.netflow.bytes位的欄標題event\_netflow\_bytes不只是bytes。若要進一步自訂欄名稱，您可以修改別名以符合您的偏好設定。例如，您可以使用底線或其他分隔符號來分隔struct名稱和欄位名稱。

請記得根據struct資料表定義以及查詢結果中要的欄位來修改資料行名稱和參照。

```
SELECT
  event.src_ip AS event_src_ip,
  event.dest_ip AS event_dest_ip,
  event.proto AS event_proto,
  event.app_proto AS event_app_proto,
  event.netflow.pkts AS event_netflow_pkts,
  event.netflow.bytes AS event_netflow_bytes,
  event.netflow.tcp_flags.syn AS event_netflow_tcp_flags_syn,
  event.netflow.tls_inspected AS event_netflow_tls_inspected
FROM network_firewall_netflow_logs
WHERE event.netflow.tls_inspected = true
```

## 查詢 Network Load Balancer 日誌

使用 Athena 分析和處理 Network Load Balancer 的日誌。這些日誌會接收傳送至 Network Load Balancer 之 Transport Layer Security (TLS) 的詳細資訊。您可以使用這些存取日誌來分析流量模式和排除問題。

分析 Network Load Balancer 存取日誌之前，請對其啟用和設定，以在目的地 Amazon S3 儲存貯體中儲存。如需詳細資訊，以及有關每個 Network Load Balancer 存取日誌項目的資訊，請參閱[存取您的 Network Load Balancer 日誌](#)。

- [為 Network Load Balancer 日誌建立資料表](#)
- [Network Load Balancer 查詢範例](#)

若要為 Network Load Balancer 日誌建立資料表

1. 複製下列 DDL 陳述式，並將其貼到 Athena 主控台。檢查 Network Load Balancer 日誌記錄的[語法](#)。您可能需要更新以下查詢，為記錄的最新版本包含欄和 Regex 語法。

```
CREATE EXTERNAL TABLE IF NOT EXISTS nlb_tls_logs (
  type string,
  version string,
```

```

time string,
elb string,
listener_id string,
client_ip string,
client_port int,
target_ip string,
target_port int,
tcp_connection_time_ms double,
tls_handshake_time_ms double,
received_bytes bigint,
sent_bytes bigint,
incoming_tls_alert int,
cert_arn string,
certificate_serial string,
tls_cipher_suite string,
tls_protocol_version string,
tls_named_group string,
domain_name string,
alpn_fe_protocol string,
alpn_be_protocol string,
alpn_client_preference_list string,
tls_connection_creation_time string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
    '([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*):([0-9]*) ([^\ ]*):([0-9]*)
  ([-.\0-9]*) ([-.\0-9]*) ([-0-9]*) ([-0-9]*) ([-0-9]*) ([^\ ]*) ([^\ ]*) ([^\ ]*)
  ([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*)$')
  LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/AWS_account_ID/
  elasticloadbalancing/region';

```

2. 修改 LOCATION Amazon S3 儲存貯體，以指定您的 Network Load Balancer 日誌的目的地。
3. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 nlb\_tls\_logs 資料表，讓其中的資料可供您查詢。

## Network Load Balancer 查詢範例

若要查看憑證使用的次數，請使用類似此範例的查詢：

```
SELECT count(*) AS
```

```
    ct,  
    cert_arn  
FROM "nlb_tls_logs"  
GROUP BY cert_arn;
```

下列查詢顯示使用早於 TLS 1.3 版本的使用者數量：

```
SELECT tls_protocol_version,  
       COUNT(tls_protocol_version) AS  
       num_connections,  
       client_ip  
FROM "nlb_tls_logs"  
WHERE tls_protocol_version < 'tlsv13'  
GROUP BY tls_protocol_version, client_ip;
```

使用下列查詢，來識別需要很長 TLS 交握時間的連接：

```
SELECT *  
FROM "nlb_tls_logs"  
ORDER BY tls_handshake_time_ms DESC  
LIMIT 10;
```

使用下列查詢可以識別和計算在過去 30 天內交涉的 TLS 通訊協定版本和密碼套件。

```
SELECT tls_cipher_suite,  
       tls_protocol_version,  
       COUNT(*) AS ct  
FROM "nlb_tls_logs"  
WHERE from_iso8601_timestamp(time) > current_timestamp - interval '30' day  
      AND NOT tls_protocol_version = '-'  
GROUP BY tls_cipher_suite, tls_protocol_version  
ORDER BY ct DESC;
```

## 查詢 Amazon Route 53 Resolver 查詢日誌

您可以為 Amazon Route 53 Resolver 查詢日誌建立 Athena 資料表，並從 Athena 查詢它們。

Route 53 Resolver 查詢記錄功能用於記錄資源在 VPC 內所做的 DNS 查詢、使用傳入解析程式端點的內部部署資源、使用遞迴 DNS 解析之傳出解析程式端點的查詢，以及使用 Route 53 Resolver DNS 防火牆規則封鎖、允許或監控網域清單的查詢。如需有關 Resolver 查詢記錄的詳細資訊，請參閱

《Amazon Route 53 開發人員指南》中的 [Resolver 查詢記錄](#)。如需有關日誌中各欄位的資訊，請參閱《Amazon Route 53 開發人員指南》中的 [在 Resolver 查詢日誌中出現的值](#)。

## 建立 Resolver 查詢日誌的資料表

您可以使用 Athena 主控台內的查詢編輯器，建立並查詢 Route 53 Resolver 查詢日誌的資料表。

若要建立並查詢 Route 53 Resolver 查詢日誌的 Athena 資料表

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在 Athena 查詢編輯器中，輸入以下 CREATE TABLE 陳述式。使用對應於 Amazon S3 中 Resolver 日誌位置的值取代 LOCATION 子句值。

```
CREATE EXTERNAL TABLE r53_rlogs (  
  version string,  
  account_id string,  
  region string,  
  vpc_id string,  
  query_timestamp string,  
  query_name string,  
  query_type string,  
  query_class  
    string,  
  rcode string,  
  answers array<  
    struct<  
      Rdata: string,  
      Type: string,  
      Class: string>  
    >,  
  srcaddr string,  
  srcport int,  
  transport string,  
  srcids struct<  
    instance: string,  
    resolver_endpoint: string  
  >,  
  firewall_rule_action string,  
  firewall_rule_group_id string,  
  firewall_domain_list_id string  
)  
  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
```

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/aws_account_id/vpcdnsquerylogs/{vpc-id}/'
```

因為解析器查詢記錄檔資料是 JSON 格式，因此 CREATE TABLE 陳述式會使用 [JSON SerDe 程式庫](#) 來分析資料。

#### Note

SerDe 預期每個 JSON 文件都位於單行文字上，且記錄中欄位之間沒有行終止字元。如果 JSON 文本是漂亮的打印格式，您可能會收到一條錯誤消息，如 HIVE\_CURSOR\_ERROR：行不是有效的 JSON 對象或 HIVE\_CURSOR\_ERROR:: 意外 JsonParseException end-of-input：當您嘗試查詢表後創建表時，對象的預期關閉標記。如需詳細資訊，請參閱上 GitHub 的 OpenX SerDe 文件中的 [JSON 資料檔案](#)。

3. 選擇 Run query (執行查詢)。該陳述式會建立名為 r53\_rlogs 的 Athena 資料表，其資料欄表示 Resolver 日誌資料中的各欄位。
4. 在 Athena 主控台查詢編輯器中，執行以下查詢，以確認您的資料表已建立。

```
SELECT * FROM "r53_rlogs" LIMIT 10
```

## 分割範例

下列範例顯示使用分割區投影並依據 vpc 和日期分割的 Resolver 查詢日誌 CREATE TABLE 陳述式。如需有關分割區投影的詳細資訊，請參閱 [使用 Amazon Athena 進行分割區投影](#)。

```
CREATE EXTERNAL TABLE r53_rlogs (  
  version string,  
  account_id string,  
  region string,  
  vpc_id string,  
  query_timestamp string,  
  query_name string,  
  query_type string,  
  query_class string,  
  rcode string,  
  answers array<  
    struct<  
      Rdata: string,  
      Type: string,  
      Class: string>  
  >
```

```

    >,
    srcaddr string,
    srcport int,
    transport string,
    srcids struct<
      instance: string,
      resolver_endpoint: string
    >,
    firewall_rule_action string,
    firewall_rule_group_id string,
    firewall_domain_list_id string
  )
PARTITIONED BY (
  `date` string,
  `vpc` string
)
ROW FORMAT SERDE      'org.openx.data.jsonserde.JsonSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT          'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION                's3://DOC-EXAMPLE-BUCKET/route53-query-logging/
AWSLogs/aws_account_id/vpcdnsquerylogs/'
TBLPROPERTIES(
  'projection.enabled' = 'true',
  'projection.vpc.type' = 'enum',
  'projection.vpc.values' = 'vpc-6446ae02',
  'projection.date.type' = 'date',
  'projection.date.range' = '2023/06/26,NOW',
  'projection.date.format' = 'yyyy/MM/dd',
  'projection.date.interval' = '1',
  'projection.date.interval.unit' = 'DAYS',
  'storage.location.template' = 's3://DOC-EXAMPLE-BUCKET/route53-query-logging/
AWSLogs/aws_account_id/vpcdnsquerylogs/${vpc}/${date}/'
)

```

## 查詢範例

以下範例顯示您可以從 Athena 對 Resolver 查詢日誌執行的部分查詢。

### 範例 1 – 以遞減的 query\_timestamp 順序查詢日誌

以下查詢以遞減的 query\_timestamp 順序顯示日誌結果。

```
SELECT * FROM "r53_rlogs"
```

```
ORDER BY query_timestamp DESC
```

## 範例 2 – 在指定的開始和結束時間內查詢日誌

以下查詢會查詢 2020 年 9 月 24 日午夜到早上 8 點之間的日誌。根據您的需求替換開始和結束時間。

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode
FROM "r53_rlogs"
WHERE (parse_datetime(query_timestamp, 'yyyy-MM-dd''T''HH:mm:ss''Z')
      BETWEEN parse_datetime('2020-09-24-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
      AND parse_datetime('2020-09-24-00:08:00', 'yyyy-MM-dd-HH:mm:ss'))
ORDER BY query_timestamp DESC
```

## 範例 3 – 根據指定的 DNS 查詢名稱模式查詢日誌

以下查詢會選取其查詢名稱包括 "example.com" 字串的記錄。

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode, answers
FROM "r53_rlogs"
WHERE query_name LIKE '%example.com%'
ORDER BY query_timestamp DESC
```

## 範例 4 – 查詢無答案的日誌請求

以下查詢會選取請求未收到答案的日誌項目。

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode, answers
FROM "r53_rlogs"
WHERE cardinality(answers) = 0
```

## 範例 5 – 查詢有特定答案的日誌

以下查詢會顯示其中的 answer.Rdata 值有指定 IP 地址的日誌。

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode,
       answer.Rdata
FROM "r53_rlogs"
CROSS JOIN UNNEST(r53_rlogs.answers) as st(answer)
WHERE answer.Rdata='203.0.113.16';
```

## 查詢 Amazon SES 事件日誌

您可以使用 Amazon Athena 來查詢 [Amazon Simple Email Service](#) (Amazon SES) 事件日誌。

Amazon SES 是一種電子郵件平台，提供便利且符合經濟效益的方式，讓您能使用自有的電子郵件地址和網域傳送及接收電子郵件。您可以使用事件、指標和統計資料，以細微的程度監控 Amazon SES 傳送活動。

根據您定義的特性，您可以將 Amazon SES 事件發佈到 [Amazon CloudWatch](#)、[Amazon 資料 Firehose](#) 或 [Amazon 簡單通知服務](#)。將資訊存放在 Amazon S3 之後，您就能從 Amazon Athena 進行查詢。

如需 Amazon SES 日誌的 Athena CREATE TABLE 陳述式範例，包括如何在 Amazon SES 事件日誌資料中建立檢視和扁平化巢狀陣列的步驟，請參閱使用分析 [服務分析 Amazon SES 事件資料的 AWS 部落格文章](#) 中的「[步驟 3：使用 Amazon Athena 查詢 SES 事件日誌](#)」。AWS

## 查詢 Amazon VPC 流程日誌

Amazon Virtual Private Cloud 流量日誌擷取傳入和傳出 VPC 中網路界面之 IP 流量的相關資訊。使用日誌來調查網路流量模式，並識別整個 VPC 網路所遭受的威脅和風險。

若要查詢 Amazon VPC 流程日誌，您有下列兩種選擇：

- Amazon VPC 主控台 — 使用 Amazon VPC 主控台內的 Athena 整合功能產生 AWS CloudFormation 範本，以建立 Athena 資料庫、工作群組和流程日誌表，並為您進行分割。此範本也會建立一組 [預先定義流程日誌查詢](#)，您可以使用其取得關於流經 VPC 流量的可行見解。

如需有關此方法的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [使用 Amazon Athena 查詢流程日誌](#)。

- Amazon Athena 主控台 – 直接在 Athena 主控台中建立資料表和查詢。如需詳細資訊，請繼續閱讀此頁內容。

### 為自訂 VPC 流程日誌建立和查詢資料表

在 Athena 中開始查詢日誌之前，[請啟用 VPC 流程日誌](#)，並將它們設定為儲存到您的 Amazon S3 儲存貯體。建立日誌之後，讓它們執行幾分鐘以收集一些資料。日誌是以 Athena 可讓您直接查詢的 GZIP 壓縮格式建立。

建立 VPC 流程日誌時，若您想要指定要在流程日誌中傳回的欄位以及欄位出現的順序，請使用自訂格式。如需有關流程日誌記錄的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 [流程日誌記錄](#)。

## 一般考量事項

在 Athena 中為 Amazon VPC 流程日誌建立資料表時，請記住以下幾點：

- 預設情況下，在 Athena 中，Parquet 將會依名稱存取欄。如需詳細資訊，請參閱[處理結構描述更新](#)。
- 使用流程日誌記錄中的名稱作為 Athena 中的資料欄名稱。Athena 結構描述中的資料欄名稱應與 Amazon VPC 流程日誌中的欄位名稱完全相符，但有以下差異：
  - 在 Athena 資料欄名稱中，將 Amazon VPC 日誌欄位名稱中的連字號替換為底線。在 Athena 中，資料庫名稱、資料表名稱和資料欄名稱唯一接受的字元為小寫字母、數字和底線字元。如需詳細資訊，請參閱[資料庫、資料表和資料欄名稱](#)。
  - 透過反引號將流程日誌記錄名稱逸出為 Athena 中的[保留關鍵字](#)。
- VPC 流程記錄是 AWS 帳戶 特定的。當您將日誌檔案發布到 Amazon S3 時，Amazon VPC 在 Amazon S3 中建立的路徑會包含用來建立流量日誌的 AWS 帳戶 的 ID。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[發布流量日誌至 Amazon S3](#)。

## Amazon VPC 流程日誌的 CREATE TABLE 陳述式

以下程序會建立 Amazon VPC 流程日誌的 Amazon VPC 資料表。當您使用自訂格式建立流程日誌時，需要建立資料表，其欄位符合您在建立流程日誌時指定的欄位，而順序與您指定它們的順序相同。

### 為 Amazon VPC 流量日誌建立 Athena 資料表

1. 在 Athena 主控台查詢編輯器中輸入類似下列的 DDL 陳述式，遵循 [一般考量事項](#) 區段中的準則。陳述式範例會建立擁有 Amazon VPC 流程日誌版本 2 至 5 (如[流程日誌記錄](#)所記載) 之資料欄的資料表。如果您使用不同組的資料欄或資料欄順序，請據此修改陳述式。

```
CREATE EXTERNAL TABLE IF NOT EXISTS `vpc_flow_logs` (  
  version int,  
  account_id string,  
  interface_id string,  
  srcaddr string,  
  dstaddr string,  
  srcport int,  
  dstport int,  
  protocol bigint,  
  packets bigint,  
  bytes bigint,  
  start bigint,
```

```
`end` bigint,  
action string,  
log_status string,  
vpc_id string,  
subnet_id string,  
instance_id string,  
tcp_flags int,  
type string,  
pkt_srcaddr string,  
pkt_dstaddr string,  
region string,  
az_id string,  
sublocation_type string,  
sublocation_id string,  
pkt_src_aws_service string,  
pkt_dst_aws_service string,  
flow_direction string,  
traffic_path int  
)  
PARTITIONED BY (`date` date)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ' '  
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/{account_id}/  
vpcflowlogs/{region_code}/'  
TBLPROPERTIES ("skip.header.line.count"="1");
```

請注意以下重點：

- 查詢會指定ROW FORMAT DELIMITED並省略指定 SerDe。這表示該查詢會使用 [LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案](#)。在這個查詢中，欄位以空格結尾。
- PARTITIONED BY 子句使用 date 類型。這樣就可以在查詢中使用數學運算子來選取特定日期之前或之後的項目。

 Note

因為 date 是 DDL 陳述式中保留的關鍵字，它會溢出反引號字元。如需詳細資訊，請參閱 [保留的關鍵字](#)。

- 針對具有不同自訂格式的 VPC 流程日誌，請修改欄位，使其符合您建立流程日誌時指定的欄位。

2. 修改 LOCATION 's3://DOC-EXAMPLE-BUCKET/*prefix*/AWSLogs/{*account\_id*}/vpcflowlogs/{*region\_code*}/' 來指向包含日誌資料的 Amazon S3 儲存貯體。
3. 在 Athena 主控台中執行查詢。查詢完成之後，Athena 會註冊 vpc\_flow\_logs 資料表，讓其中的資料可供您發出查詢。
4. 建立能夠讀取資料的分割區，如下列查詢範例所示。這個查詢針對指定的日期建立單一分割區。視需要將預留位置換成日期和位置。

#### Note

這個查詢針對您指定的日期，只建立單一分割區。若要自動執行程序，請使用指令碼以執行此查詢和依此針對 year/month/day 建立分割區，或使用指定[分割區投影](#)的 CREATE TABLE 陳述式。

```
ALTER TABLE vpc_flow_logs
ADD PARTITION (`date`='YYYY-MM-dd')
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/{account_id}/
vpcflowlogs/{region_code}/YYYY/MM/dd';
```

#### vpc\_flow\_logs 資料表的查詢範例

使用 Athena 主控台內的查詢編輯器在您建立的資料表上執行 SQL 陳述式。您可以儲存查詢、檢視先前的查詢，或以 CSV 格式下載查詢結果。在下列範例中，將 vpc\_flow\_logs 替換為您的資料表名稱。根據自己的需求修改欄值和其他變數。

下列範例查詢會列出指定日期的最多 100 個流程日誌。

```
SELECT *
FROM vpc_flow_logs
WHERE date = DATE('2020-05-04')
LIMIT 100;
```

以下查詢列出所有拒絕的 TCP 連線，並使用新建立的日期新分割欄 date，以從中擷取發生這些事件是星期幾。

```
SELECT day_of_week(date) AS
day,
date,
```

```
interface_id,  
srcaddr,  
action,  
protocol  
FROM vpc_flow_logs  
WHERE action = 'REJECT' AND protocol = 6  
LIMIT 100;
```

若要查看您的哪一個伺服器接收最多的 HTTPS 請求，請使用下列查詢。它計算 HTTPS 連接埠 443 所接收的封包數、依目的地 IP 地址分組，然後傳回上週的前 10 個。

```
SELECT SUM(packets) AS  
packetcount,  
dstaddr  
FROM vpc_flow_logs  
WHERE dstport = 443 AND date > current_date - interval '7' day  
GROUP BY dstaddr  
ORDER BY packetcount DESC  
LIMIT 10;
```

以 Parquet 格式為流程日誌建立資料表

以下程序會以 Parquet 格式建立 Amazon VPC 流程日誌的 Amazon VPC 資料表。

為 Amazon VPC 流量日誌以 Parquet 格式建立 Athena 資料表

1. 在 Athena 主控台查詢編輯器中輸入類似下列的 DDL 陳述式，遵循 [一般考量事項](#) 區段中的準則。陳述式範例會建立擁有 Amazon VPC 流程日誌版本 2 至 5 (如 Parquet 格式、Hive 每小時已分割的 [流程日誌記錄](#) 所記載) 之資料欄的資料表。如果您沒有小時分割區，請從 PARTITIONED BY 子句中移除 hour。

```
CREATE EXTERNAL TABLE IF NOT EXISTS vpc_flow_logs_parquet (  
version int,  
account_id string,  
interface_id string,  
srcaddr string,  
dstaddr string,  
srcport int,  
dstport int,  
protocol bigint,  
packets bigint,  
bytes bigint,
```

```
start bigint,
`end` bigint,
action string,
log_status string,
vpc_id string,
subnet_id string,
instance_id string,
tcp_flags int,
type string,
pkt_srcaddr string,
pkt_dstaddr string,
region string,
az_id string,
sublocation_type string,
sublocation_id string,
pkt_src_aws_service string,
pkt_dst_aws_service string,
flow_direction string,
traffic_path int
)
PARTITIONED BY (
`aws-account-id` string,
`aws-service` string,
`aws-region` string,
`year` string,
`month` string,
`day` string,
`hour` string
)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/'
TBLPROPERTIES (
'EXTERNAL'='true',
'skip.header.line.count'='1'
)
```

2. 修改 LOCATION 's3://DOC-EXAMPLE-BUCKET/*prefix*/AWSLogs/' 來指向包含日誌資料的 Amazon S3 路徑。

3. 在 Athena 主控台中執行查詢。
4. 如果您的資料採用 Hive 相容的格式，請在 Athena 主控台中執行以下命令來更新和載入中繼存放區中的 Hive 分割區。查詢完成後，您可以在 `vpc_flow_logs_parquet` 資料表中查詢資料。

```
MSCK REPAIR TABLE vpc_flow_logs_parquet
```

如果您沒有使用 Hive 相容的資料，請執行 [ALTER TABLE ADD PARTITION](#) 來載入分割區。

如需有關使用 Athena 查詢 Parquet 格式的 Amazon VPC 流程日誌的詳細資訊，請參閱 AWS 大數據部落格中的 [使用 Apache Parquet 格式的 VPC 流程日誌最佳化效能並降低網路分析成本](#)。

使用分割區投影為 Amazon VPC 流程日誌建立和查詢資料表

使用如下 CREATE TABLE 陳述句，以建立資料表、分割資料表，並使用 [分割區投影](#) 自動填入分割區。將範例中的資料表名稱 `test_table_vpclogs` 替換為您的資料表名稱。編輯 LOCATION 子句以指定包含 Amazon VPC 日誌資料的 Amazon S3 儲存貯體。

以下 CREATE TABLE 陳述式適用於以非 Hive 樣式分割格式交付的 VPC 流程日誌。此範例允許多帳戶彙總。如果您要將多個帳戶的 VPC 流程日誌集中到一個 Amazon S3 儲存貯體，則必須在 Amazon S3 路徑中輸入帳戶 ID。

```
CREATE EXTERNAL TABLE IF NOT EXISTS test_table_vpclogs (  
  version int,  
  account_id string,  
  interface_id string,  
  srcaddr string,  
  dstaddr string,  
  srcport int,  
  dstport int,  
  protocol bigint,  
  packets bigint,  
  bytes bigint,  
  start bigint,  
  `end` bigint,  
  action string,  
  log_status string,  
  vpc_id string,  
  subnet_id string,  
  instance_id string,  
  tcp_flags int,
```

```
type string,
pkt_srcaddr string,
pkt_dstaddr string,
az_id string,
sublocation_type string,
sublocation_id string,
pkt_src_aws_service string,
pkt_dst_aws_service string,
flow_direction string,
traffic_path int
)
PARTITIONED BY (accid string, region string, day string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
LOCATION '$LOCATION_OF_LOGS'
TBLPROPERTIES
(
"skip.header.line.count"="1",
"projection.enabled" = "true",
"projection.accid.type" = "enum",
"projection.accid.values" = "$ACCID_1,$ACCID_2",
"projection.region.type" = "enum",
"projection.region.values" = "$REGION_1,$REGION_2,$REGION_3",
"projection.day.type" = "date",
"projection.day.range" = "$START_RANGE,NOW",
"projection.day.format" = "yyyy/MM/dd",
"storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/AWSLogs/${accid}/vpcflowlogs/
${region}/${day}"
)
```

### test\_table\_vpclogs 的查詢範例

下列查詢範例會查詢由前面 CREATE TABLE 陳述式所建立的 test\_table\_vpclogs。將查詢中的 test\_table\_vpclogs 替換為您自己的資料表名稱。根據自己的需求修改欄值和其他變數。

若要在指定時段內依時間順序傳回前 100 個存取日誌項目，請執行如下查詢。

```
SELECT *
FROM test_table_vpclogs
WHERE day >= '2021/02/01' AND day < '2021/02/28'
ORDER BY day ASC
LIMIT 100
```

若要查看哪個伺服器在指定時段內接收前十個 HTTP 封包，請執行如下查詢。查詢計算 HTTPS 連接埠 443 所接收的封包數、依目的地 IP 地址分組，然後傳回上週的前 10 個查詢。

```
SELECT SUM(packets) AS packetcount,
       dstaddr
FROM test_table_vpclogs
WHERE dstport = 443
      AND day >= '2021/03/01'
      AND day < '2021/03/31'
GROUP BY dstaddr
ORDER BY packetcount DESC
LIMIT 10
```

若要傳回在指定時段內建立的日誌，請執行如下查詢。

```
SELECT interface_id,
       srcaddr,
       action,
       protocol,
       to_iso8601(from_unixtime(start)) AS start_time,
       to_iso8601(from_unixtime("end")) AS end_time
FROM test_table_vpclogs
WHERE DAY >= '2021/04/01'
      AND DAY < '2021/04/30'
```

若要傳回指定時段之間的來源 IP 地址的存取日誌，請執行如下查詢。

```
SELECT *
FROM test_table_vpclogs
WHERE srcaddr = '10.117.1.22'
      AND day >= '2021/02/01'
      AND day < '2021/02/28'
```

若要列出拒絕的 TCP 連接，請執行如下查詢。

```
SELECT day,
       interface_id,
       srcaddr,
       action,
       protocol
FROM test_table_vpclogs
WHERE action = 'REJECT' AND protocol = 6 AND day >= '2021/02/01' AND day < '2021/02/28'
```

```
LIMIT 10
```

若要傳回以 10.117 開頭的 IP 地址範圍的存取日誌，請執行如下查詢。

```
SELECT *
FROM test_table_vpclogs
WHERE split_part(srcaddr, '.', 1)='10'
      AND split_part(srcaddr, '.', 2) = '117'
```

若要傳回特定時間範圍之間的目的地 IP 地址的存取日誌，請執行如下查詢。

```
SELECT *
FROM test_table_vpclogs
WHERE dstaddr = '10.0.1.14'
      AND day >= '2021/01/01'
      AND day < '2021/01/31'
```

使用分割區投影，以 Apache Parquet 格式為流程日誌建立資料表

下列 VPC 流程日誌的資料分割投影 CREATE TABLE 陳述式採用 Apache Parquet 格式，與 Hive 不相容，並按小時和日期而非按日期進行分割。將範例中的資料表名稱 test\_table\_vpclogs\_parquet 替換為您的資料表名稱。編輯 LOCATION 子句以指定包含 Amazon VPC 日誌資料的 Amazon S3 儲存貯體。

```
CREATE EXTERNAL TABLE IF NOT EXISTS test_table_vpclogs_parquet (
  version int,
  account_id string,
  interface_id string,
  srcaddr string,
  dstaddr string,
  srcport int,
  dstport int,
  protocol bigint,
  packets bigint,
  bytes bigint,
  start bigint,
  `end` bigint,
  action string,
  log_status string,
  vpc_id string,
  subnet_id string,
  instance_id string,
```

```

tcp_flags int,
type string,
pkt_srcaddr string,
pkt_dstaddr string,
az_id string,
sublocation_type string,
sublocation_id string,
pkt_src_aws_service string,
pkt_dst_aws_service string,
flow_direction string,
traffic_path int
)
PARTITIONED BY (region string, date string, hour string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/{account_id}/vpcflowlogs/'
TBLPROPERTIES (
"EXTERNAL"="true",
"skip.header.line.count" = "1",
"projection.enabled" = "true",
"projection.region.type" = "enum",
"projection.region.values" = "us-east-1,us-west-2,ap-south-1,eu-west-1",
"projection.date.type" = "date",
"projection.date.range" = "2021/01/01,NOW",
"projection.date.format" = "yyyy/MM/dd",
"projection.hour.type" = "integer",
"projection.hour.range" = "00,23",
"projection.hour.digits" = "2",
"storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/${account_id}/vpcflowlogs/${region}/${date}/${hour}"
)

```

## 其他資源

如需有關使用 Athena 分析 VPC 流程日誌的詳細資訊，請參閱以下 AWS 大數據部落格文章。

- [利用 point-and-click Amazon Athena 整合分析 VPC 流程日誌](#)
- [使用 Amazon 雅典娜和亞馬遜分析 VPC 流程日誌 QuickSight](#)
- [使用 Apache Parquet 格式的 VPC 流程日誌，最佳化效能並降低網路分析成本](#)

## 查詢 AWS WAF 記錄檔

AWS WAF 是一種 Web 應用程式防火牆，可讓您監視和控制受保護的 Web 應用程式從用戶端接收的 HTTP 和 HTTPS 要求。您可以透過在 Web 存取控制清單 (ACL) 內設定規則來定義如何處理 AWS WAF Web 要求。然後，您可以透過將 Web ACL 與 Web 應用程式建立關聯來保護 Web 應用程式。您可以使用保護的 Web 應用程式資源範例包 AWS WAF 括 Amazon CloudFront 分發、Amazon API Gateway REST API 和應用程式負載平衡器。如需有關的詳細資訊 AWS WAF，請參閱[AWS WAF](#)開發人員指南中的。

AWS WAF 記錄檔包括 Web ACL 分析的流量相關資訊，例如從您的 AWS 資源 AWS WAF 接收要求的時間、有關請求的詳細資訊，以及每個要求符合之規則的動作。

您可以設定 AWS WAF Web ACL，將記錄檔發佈到數個目的地之一，您可以在其中查詢和檢視這些目的地。如需設定 Web ACL 記錄和記錄 AWS WAF 檔內容的詳細資訊，請參閱[AWS WAF 開發人員指南](#)中的[記錄 AWS WAF Web ACL 流量](#)。

如需如何將 AWS WAF 日誌彙總到中央資料湖儲存庫並使用 Athena 進行查詢的範例，請參閱 AWS 大數據部落格文章：[使用 OpenSearch 服務、Amazon Athena 和 Amazon 分析 AWS WAF 日誌 QuickSight](#)。

本主題提供兩個範例 CREATE TABLE 陳述式：一個使用分割區，另一個不使用。

### Note

所以本主題中的 CREATE TABLE 陳述式可同時用於 v1 和 v2 AWS WAF 日誌。在 v1 中，webaclid 欄位含有一個 ID。在 v2 中，webaclid 欄位含有一個完整的 ARN。此 CREATE TABLE 陳述式藉由使用 string 資料類型，以不可知的方式處理此內容。

## 主題

- [使用分割區投影在 Athena 中建立 AWS WAF S3 日誌的資料表](#)
- [為不分割的 AWS WAF 記錄建立資料表](#)
- [AWS WAF 記錄檔查詢範例](#)

使用分割區投影在 Athena 中建立 AWS WAF S3 日誌的資料表

因為 AWS WAF 記錄檔具有已知的結構，您可以事先指定其資料分割配置，因此您可以使用 Athena 分割區[投影功能，縮短查詢執行階段並自動化分割區](#)管理。分割區投影會在新增資料時自動新增分割區。因此您無需使用 ALTER TABLE ADD PARTITION 手動新增分割區。

下列範例CREATE TABLE陳述式會自動在指定日期的 AWS WAF 記錄上使用分割區投影，直到四個不同 AWS 區域為止。本範例中的 PARTITION BY 子句按區域和日期進行分割，但您可以根據自己的要求進行修改。視需要修改欄位，以符合您的日誌輸出。在LOCATION和storage.location.template子句中，將儲存#體和 *accountID* 預留位置取代為識別日誌之 Amazon S3 儲存貯體位置的 AWS WAF 值。對於 projection.day.range，請用您要使用的開始日期取代 *2021/01/01*。成功執行查詢之後，您可以查詢資料表。您無須執行 ALTER TABLE ADD PARTITION 就能載入分割區。

```
CREATE EXTERNAL TABLE `waf_logs` (
  `timestamp` bigint,
  `formatversion` int,
  `webaclid` string,
  `terminatingruleid` string,
  `terminatingruletype` string,
  `action` string,
  `terminatingrulematchdetails` array <
    struct <
      conditiontype: string,
      sensitivitylevel: string,
      location: string,
      matcheddata: array < string >
    >
  >,
  `httpsourcename` string,
  `httpsourceid` string,
  `rulegrouplist` array <
    struct <
      rulegroupid: string,
      terminatingrule: struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
          struct <
            conditiontype:
string,
            sensitivitylevel: string,
            location:
string,
            matcheddata:
array < string >
          >
        >
      >
    >
  >

```

```

>
>,
nonterminatingmatchingrules: array <
    struct <
        ruleid: string,
        action: string,
        overriddenaction:
string,
        rulematchdetails:
array <
    struct <
        conditiontype: string,
        sensitivitylevel: string,
        location: string,
        matcheddata: array < string >
        >
    >,
    challengerresponse:
struct <
    responsecode: string,
    solvetimestamp: string
    >,
    captcharesponse:
struct <
    responsecode: string,
    solvetimestamp: string
    >
    >
    >,
    excludedrules: string
    >
    >,
`ratebasedrulelist` array <
    struct <

```

```

        ratebasedruleid: string,
        limitkey: string,
        maxrateallowed: int
    >
>,
`nonterminatingmatchingrules` array <
    struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
            struct <
                conditiontype: string,
                sensitivitylevel:
string,
                location: string,
                matcheddata: array <
string >
            >
        >,
        challengerresponse: struct <
            responsecode: string,
            solvetimestamp: string
        >,
        captcharesponse: struct <
            responsecode: string,
            solvetimestamp: string
        >
    >
>,
`requestheadersinserted` array <
    struct <
        name: string,
        value: string
    >
>,
`responsecodesent` string,
`httprequest` struct <
    clientip: string,
    country: string,
    headers: array <
        struct <
            name: string,
            value: string
        >
    >

```

```

        >,
        uri: string,
        args: string,
        httpversion: string,
        httpmethod: string,
        requestid: string
    >,
`labels` array <
    struct <
        name: string
    >
>,
`captcharesponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
>,
`challengeresponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
>,
`ja3Fingerprint` string,
`oversizefields` string,
`requestbodysize` int,
`requestbodysizeinspectedbywaf` int
)
PARTITIONED BY (
`region` string,
`date` string)
ROW FORMAT SERDE
    'org.openx.data.jsonserde.JsonSerDe'
STORED AS INPUTFORMAT
    'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
    'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
    's3://DOC-EXAMPLE-BUCKET/AWSLogs/accountID/WAFLogs/region/DOC-EXAMPLE-WEBACL/'
TBLPROPERTIES(
'projection.enabled' = 'true',
'projection.region.type' = 'enum',
'projection.region.values' = 'us-east-1,us-west-2,eu-central-1,eu-west-1',
'projection.date.type' = 'date',
'projection.date.range' = '2021/01/01,NOW',

```

```
'projection.date.format' = 'yyyy/MM/dd',  
'projection.date.interval' = '1',  
'projection.date.interval.unit' = 'DAYS',  
'storage.location.template' = 's3://DOC-EXAMPLE-BUCKET/AWSLogs/accountID/WAFLogs/  
${region}/DOC-EXAMPLE-WEBACL/${date}/')
```

### Note

範例中LOCATION子句中的路徑格式為標準格式，但可能會根據您實作的 AWS WAF 組態而有所不同。例如，下列範例 AWS WAF 記錄檔路徑適用於 CloudFront 散發：

```
s3://DOC-EXAMPLE-BUCKET/AWSLogs/12345678910/WAFLogs/cloudfront/  
cloudfrontyt/2022/08/08/17/55/
```

如果您在建立或查詢 AWS WAF 記錄資料表時遇到問題，請確認記錄資料或[連絡人](#)的位置 AWS Support。

如需有關分割區投影的詳細資訊，請參閱[使用 Amazon Athena 進行分割區投影](#)。

為不分割的 AWS WAF 記錄建立資料表

本節說明如何在不進行資料分割或分割區投影的情況下為 AWS WAF 記錄建立資料表。

### Note

出於性能和成本原因，我們不建議使用非分區的結構描述進行查詢。如需詳細資訊，請參閱[大數據部落格中的 Amazon Athena + AWS 大效能調整秘訣](#)。

建立 AWS WAF 表格的步驟

1. 複製下列 DDL 陳述式，並將其貼到 Athena 主控台。視需要修改欄位，以符合您的日誌輸出。修改 Amazon S3 儲存貯體的 LOCATION 以與存放日誌的儲存貯體相對應。

此查詢會使用 [OpenX SerDe](#)。

### Note

SerDe 預期每個 JSON 文件都位於單行文字上，且記錄中欄位之間沒有行終止字元。如果 JSON 文本是漂亮的打印格式，您可能會收到一條錯誤消息，如 HIVE\_CURSOR\_ERROR：行不是有效的 JSON 對象或 HIVE\_CURSOR\_ERROR:: 意外 JsonParseException end-of-input：當您嘗試查詢表後創建表時，對象的預期關閉標記。如需詳細資訊，請參閱上 GitHub 的 [OpenX SerDe 文件中的 JSON 資料檔案](#)。

```
CREATE EXTERNAL TABLE `waf_logs` (
  `timestamp` bigint,
  `formatversion` int,
  `webaclid` string,
  `terminatingruleid` string,
  `terminatingruletype` string,
  `action` string,
  `terminatingrulematchdetails` array <
    struct <
      conditiontype: string,
      sensitivitylevel: string,
      location: string,
      matcheddata: array < string >
    >
  >,
  `httpsourcename` string,
  `httpsourceid` string,
  `rulegrouplist` array <
    struct <
      rulegroupid: string,
      terminatingrule: struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
          struct <
            conditiontype:
string,
            sensitivitylevel: string,
            location:
string,
```

```

    array < string >
    matcheddata:
    >
    >
    >,
    nonterminatingmatchingrules: array <
    struct <
    ruleid: string,
    action: string,
    overriddenaction:
string,
    rulematchdetails:
array <
    struct <
    conditiontype: string,
    sensitivitylevel: string,
    location: string,
    matcheddata: array < string >
    >
    >,
    challengerresponse:
struct <
responsecode: string,
solvetimestamp: string
    >,
    captcharesponse:
struct <
responsecode: string,
solvetimestamp: string
    >
    >
    >,
    excludedrules: string
    >

```

```

        >,
`ratebasedrulelist` array <
    struct <
        ratebasedruleid: string,
        limitkey: string,
        maxrateallowed: int
    >
    >,
`nonterminatingmatchingrules` array <
    struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
            struct <
                conditiontype:
string,
                sensitivitylevel:
string,
                location: string,
                matcheddata: array <
string >
            >
        >,
        challengerresponse: struct <
            responsecode: string,
            solvetimestamp: string
        >,
        captcharesponse: struct <
            responsecode: string,
            solvetimestamp: string
        >
    >
    >,
`requestheadersinserted` array <
    struct <
        name: string,
        value: string
    >
    >,
`responsecodesent` string,
`httprequest` struct <
    clientip: string,
    country: string,
    headers: array <

```

```

        struct <
            name: string,
            value: string
        >
    >,
    uri: string,
    args: string,
    httpversion: string,
    httpmethod: string,
    requestid: string
    >,
`labels` array <
    struct <
        name: string
    >
    >,
`captcharesponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
    >,
`challengeresponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
    >,
`ja3Fingerprint` string,
`oversizefields` string,
`requestbodysize` int,
`requestbodysizeinspectedbywaf` int
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/'

```

2. 在 Athena 主控台查詢編輯器中執行 CREATE EXTERNAL TABLE 陳述式。這會註冊 waf\_logs 資料表，並讓其中的資料可用於從 Athena 進行查詢。

## AWS WAF 記錄檔查詢範例

以下許多範例查詢使用本文件中先前所建立的分割區投影資料表。請根據您的需求修改範例中的資料表名稱、資料欄值及其他變數。若要改善查詢的效能並降低成本，請在篩選條件中新增分割區資料欄。

- [Count the number of referrers that contain a specified term](#)
- [Count all matched IP addresses in the last 10 days that have matched excluded rules](#)
- [Group all counted managed rules by the number of times matched](#)
- [Group all counted custom rules by number of times matched](#)

### [使用日期和時間](#)

- [Return the timestamp field in human-readable ISO 8601 format](#)
- [Return records from the last 24 hours](#)
- [Return records for a specified date range and IP address](#)
- [For a specified date range, count the number of IP addresses in five minute intervals](#)
- [Count the number of X-Forwarded-For IP in the last 10 days](#)

### [使用遭封鎖的請求和地址](#)

- [Extract the top 100 IP addresses blocked by a specified rule type](#)
- [Count the number of times a request from a specified country has been blocked](#)
- [Count the number of times a request has been blocked, grouping by specific attributes](#)
- [Count the number of times a specific terminating rule ID has been matched](#)
- [Retrieve the top 100 IP addresses blocked during a specified date range](#)

### Example – 計算包含指定字詞的 Referrer 數量

以下查詢會計算在指定的日期範圍內包含 "amazon" 一詞的 Referrer 數量。

```
WITH test_dataset AS
  (SELECT header FROM waf_logs
   CROSS JOIN UNNEST(httprequest.headers) AS t(header) WHERE "date" >= '2021/03/01'
   AND "date" < '2021/03/31')
SELECT COUNT(*) referer_count
```

```
FROM test_dataset
WHERE LOWER(header.name)='referer' AND header.value LIKE '%amazon%'
```

Example – 計算過去 10 天內符合排除規則的所有相符 IP 地址

以下查詢會計算過去 10 天內 IP 地址符合規則群組中排除規則的次數。

```
WITH test_dataset AS
  (SELECT * FROM waf_logs
   CROSS JOIN UNNEST(rulegrouplist) AS t(allrulegroups))
SELECT
  COUNT(*) AS count,
  "httprequest"."clientip",
  "allrulegroups"."excludedrules",
  "allrulegroups"."ruleGroupId"
FROM test_dataset
WHERE allrulegroups.excludedrules IS NOT NULL AND from_unixtime(timestamp/1000) > now()
  - interval '10' day
GROUP BY "httprequest"."clientip", "allrulegroups"."ruleGroupId",
  "allrulegroups"."excludedrules"
ORDER BY count DESC
```

Example – 依相符的次數對所有計數的受管規則進行分組

如果您在 2022 年 10 月 27 日之前在 Web ACL 組態中將規則群組規則動作設定為「計數」，則在 Web ACL JSON 中將覆寫 AWS WAF 儲存為 `excludedRules`。現在，將規則覆寫為計數的 JSON 設定位於 `ruleActionOverrides` 設定中。如需詳細資訊，請參閱《AWS WAF 開發人員指南》中的[規則群組中的動作覆寫](#)。若要從新的日誌結構擷取「計數」模式下的受管規則，請查詢 `ruleGroupList` 區段中的 `nonTerminatingMatchingRules` 而非 `excludedRules` 欄位，如下列範例所示。

```
SELECT
  count(*) AS count,
  httpsourceid,
  httprequest.clientip,
  t.rulegroupid,
  t.nonTerminatingMatchingRules
FROM "waf_logs"
CROSS JOIN UNNEST(rulegrouplist) AS t(t)
WHERE action <> 'BLOCK' AND cardinality(t.nonTerminatingMatchingRules) > 0
GROUP BY t.nonTerminatingMatchingRules, action, httpsourceid, httprequest.clientip,
  t.rulegroupid
```

```
ORDER BY "count" DESC
Limit 50
```

Example – 依相符的次數對所有計數的自訂規則進行分組

下列查詢會依符合的次數，將所有計數的自訂規則分組。

```
SELECT
  count(*) AS count,
  httpsourceid,
  httprequest.clientip,
  t.ruleid,
  t.action
FROM "waf_logs"
CROSS JOIN UNNEST(nonterminatingmatchingrules) AS t(t)
WHERE action <> 'BLOCK' AND cardinality(nonTerminatingMatchingRules) > 0
GROUP BY t.ruleid, t.action, httpsourceid, httprequest.clientip
ORDER BY "count" DESC
Limit 50
```

如需有關自訂規則和受管規則群組的日誌位置的資訊，請參閱《AWS WAF 開發人員指南》中的[監控和調校](#)。

使用日期和時間

Example – 以人類看得懂的 ISO 8601 格式傳回時間戳記欄位

以下查詢會使用 `from_unixtime` 和 `to_iso8601` 函數，以人類看得懂的 ISO 8601 格式傳回 `timestamp` 欄位 (例如 2019-12-13T23:40:12.000Z，而非 1576280412771) 查詢也會傳回 HTTP 來源名稱、來源 ID 和請求。

```
SELECT to_iso8601(from_unixtime(timestamp / 1000)) as time_ISO_8601,
  httpsourcename,
  httpsourceid,
  httprequest
FROM waf_logs
LIMIT 10;
```

Example – 傳回過去 24 小時的記錄

以下查詢會使用 WHERE 子句中的篩選條件，傳回過去 24 小時記錄的 HTTP 來源名稱、HTTP 來源 ID 和 HTTP 請求欄位。

```
SELECT to_iso8601(from_unixtime(timestamp/1000)) AS time_ISO_8601,
       httpsourcename,
       httpsourceid,
       httprequest
FROM waf_logs
WHERE from_unixtime(timestamp/1000) > now() - interval '1' day
LIMIT 10;
```

#### Example – 傳回指定日期範圍和 IP 地址的記錄

以下查詢會列出指定用戶端 IP 地址的指定日期範圍內的記錄。

```
SELECT *
FROM waf_logs
WHERE httprequest.clientip='53.21.198.66' AND "date" >= '2021/03/01' AND "date" <
'2021/03/31'
```

#### Example – 如果是指定的日期範圍，則會計算每隔五分鐘的 IP 地址數量

以下查詢會針對特定日期範圍，計算每隔五分鐘的 IP 地址數量。

```
WITH test_dataset AS
  (SELECT
    format_datetime(from_unixtime((timestamp/1000) -
      ((minute(from_unixtime(timestamp / 1000))%5) * 60)), 'yyyy-MM-dd HH:mm') AS
    five_minutes_ts,
    "httprequest"."clientip"
  FROM waf_logs
  WHERE "date" >= '2021/03/01' AND "date" < '2021/03/31')
SELECT five_minutes_ts,"clientip",count(*) ip_count
FROM test_dataset
GROUP BY five_minutes_ts,"clientip"
```

#### Example – 計算過去 10 天內的 X-Forwarded-For IP 數量

以下查詢會篩選請求標頭，並計算過去 10 天內的 X-Forwarded-For IP 數量。

```
WITH test_dataset AS
  (SELECT header
  FROM waf_logs
  CROSS JOIN UNNEST (httprequest.headers) AS t(header)
```

```
WHERE from_unixtime("timestamp"/1000) > now() - interval '10' DAY)
SELECT header.value AS ip,
       count(*) AS COUNT
FROM test_dataset
WHERE header.name='X-Forwarded-For'
GROUP BY header.value
ORDER BY COUNT DESC
```

如需有關日期和時間函數的詳細資訊，請參閱 Trino 文件中的 [Date and time functions and operators](#) (日期和時間函數和運算子)。

### 使用遭封鎖的請求和地址

Example – 擷取遭到指定規則類型封鎖的前 100 個 IP 地址

以下查詢會擷取並計算在指定日期範圍內，已遭到 RATE\_BASED 終止規則封鎖的前 100 個 IP 地址。

```
SELECT COUNT(httpRequest.clientIp) as count,
       httpRequest.clientIp
FROM waf_logs
WHERE terminatingruletype='RATE_BASED' AND action='BLOCK' and "date" >= '2021/03/01'
AND "date" < '2021/03/31'
GROUP BY httpRequest.clientIp
ORDER BY count DESC
LIMIT 100
```

Example – 計算來自指定國家/地區遭到封鎖的請求次數

以下查詢會計算來自愛爾蘭 (IE) 的 IP 地址，並遭 RATE\_BASED 終止規則封鎖的請求次數。

```
SELECT
  COUNT(httpRequest.country) as count,
  httpRequest.country
FROM waf_logs
WHERE
  terminatingruletype='RATE_BASED' AND
  httpRequest.country='IE'
GROUP BY httpRequest.country
ORDER BY count
LIMIT 100;
```

### Example – 計算遭封鎖的請求次數 (依特定屬性分組)

下列查詢會計算要求遭到封鎖的次數，其結果會依 WebACL、RuleId、用 ClientIP 和 HTTP 要求 URI 分組。

```
SELECT
  COUNT(*) AS count,
  webaclid,
  terminatingruleid,
  httprequest.clientip,
  httprequest.uri
FROM waf_logs
WHERE action='BLOCK'
GROUP BY webaclid, terminatingruleid, httprequest.clientip, httprequest.uri
ORDER BY count DESC
LIMIT 100;
```

### Example – 計算與特定終止規則 ID 相符的次數。

以下查詢會計算與特定終止規則 ID (WHERE terminatingruleid='e9dd190d-7a43-4c06-bcea-409613d9506e') 相符的次數。查詢接著會以 WebACL、Action、ClientIP 和 HTTP Request URI 將結果分組。

```
SELECT
  COUNT(*) AS count,
  webaclid,
  action,
  httprequest.clientip,
  httprequest.uri
FROM waf_logs
WHERE terminatingruleid='e9dd190d-7a43-4c06-bcea-409613d9506e'
GROUP BY webaclid, action, httprequest.clientip, httprequest.uri
ORDER BY count DESC
LIMIT 100;
```

### Example – 擷取指定日期範圍內遭到封鎖的前 100 個 IP 地址

以下查詢會擷取在指定日期範圍內，已遭到封鎖的前 100 個 IP 地址。該查詢也會列出 IP 地址遭到封鎖的次數。

```
SELECT "httprequest"."clientip", "count"(*) "ipcount", "httprequest"."country"
```

```
FROM waf_logs
WHERE "action" = 'BLOCK' and "date" >= '2021/03/01'
AND "date" < '2021/03/31'
GROUP BY "httprequest"."clientip", "httprequest"."country"
ORDER BY "ipcount" DESC limit 100
```

如需有關查詢 Amazon S3 日誌的資訊，請參閱下列主題：

- AWS 知識中心中的[如何使用 Athena 分析 Amazon S3 伺服器存取日誌？](#)
- 《Amazon Simple Storage Service 使用者指南》中的[使用 Amazon Athena 查詢請求的 Amazon S3 存取日誌](#)
- 《Amazon Simple Storage Service 使用者指南》中的[使用 AWS CloudTrail 識別 Amazon S3 請求](#)

## 查詢 Amazon S3 中存放的 Web 伺服器日誌

您可以使用 Athena 查詢存放在 Amazon S3 中的 Web 伺服器日誌。本節主題說明如何在 Athena 中建立資料表，以查詢各種格式的 Web 伺服器日誌。

### 主題

- [查詢存放在 Amazon S3 中的 Apache 日誌](#)
- [查詢存放在 Amazon S3 中的網際網路資訊伺服器 \(IIS\) 日誌](#)

## 查詢存放在 Amazon S3 中的 Apache 日誌

您可以使用 Amazon Athena 查詢存放在 Amazon S3 帳戶中的 [Apache HTTP 伺服器日誌檔](#)。本主題說明如何建立資料表結構描述，以查詢一般日誌格式的 Apache [存取日誌](#) 檔案。

一般日誌格式的欄位包括用戶端 IP 地址、用戶端 ID、使用者 ID、請求接收的時間戳記、用戶端請求的文字、伺服器狀態碼，以及傳回給用戶端之物件的大小。

以下資料範例顯示 Apache 的一般日誌組態。

```
198.51.100.7 - Li [10/Oct/2019:13:55:36 -0700] "GET /logo.gif HTTP/1.0" 200 232
198.51.100.14 - Jorge [24/Nov/2019:10:49:52 -0700] "GET /index.html HTTP/1.1" 200 2165
198.51.100.22 - Mateo [27/Dec/2019:11:38:12 -0700] "GET /about.html HTTP/1.1" 200 1287
198.51.100.9 - Nikki [11/Jan/2020:11:40:11 -0700] "GET /image.png HTTP/1.1" 404 230
198.51.100.2 - Ana [15/Feb/2019:10:12:22 -0700] "GET /favicon.ico HTTP/1.1" 404 30
198.51.100.13 - Saanvi [14/Mar/2019:11:40:33 -0700] "GET /intro.html HTTP/1.1" 200 1608
```

```
198.51.100.11 - Xiulan [22/Apr/2019:10:51:34 -0700] "GET /group/index.html HTTP/1.1"
200 1344
```

## 在 Athena 中為 Apache 日誌建立資料表

查詢 Amazon S3 中存放的 Apache 日誌之前，必須為 Athena 建立資料表結構描述才能讀取日誌資料。若要建立 Apache 記錄的 Athena 資料表，您可以使用 [GROK SerDe](#)。如需有關使用 Grok 的詳細資訊 SerDe，請參閱開發人員指南中的[撰寫 grok 自訂分類器](#)。AWS Glue

## 若要在 Athena 中為 Apache Web 伺服器日誌建立資料表

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 將下列 DDL 陳述式貼上至 Athena 查詢編輯器。修改 LOCATION 's3://DOC-EXAMPLE-BUCKET/*apache-log-folder*/' 中的值來指向 Amazon S3 中的 Apache 日誌。

```
CREATE EXTERNAL TABLE apache_logs (
  client_ip string,
  client_id string,
  user_id string,
  request_received_time string,
  client_request string,
  server_status string,
  returned_obj_size string
)
ROW FORMAT SERDE
  'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.format'='^%{IPV4:client_ip} %{DATA:client_id} %{USERNAME:user_id}
%{GREEDYDATA:request_received_time} %{QUOTEDSTRING:client_request}
%{DATA:server_status} %{DATA: returned_obj_size}$'
)
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/apache-log-folder';
```

3. 在 Athena 主控台中執行查詢來註冊 apache\_logs 資料表。查詢完成時，您就可以從 Athena 查詢日誌了。

## Apache 日誌的選取查詢範例

### Example – 篩選 404 錯誤

下列查詢範例會從 `apache_logs` 資料表選取請求接收的時間、用戶端請求的文字，以及伺服器狀態碼。WHERE 子句會篩選 HTTP 狀態碼 404 (找不到頁面)。

```
SELECT request_received_time, client_request, server_status
FROM apache_logs
WHERE server_status = '404'
```

下圖顯示 Athena 查詢編輯器中查詢的結果。



The screenshot shows the Athena query results interface. At the top right, there are icons for a document and a refresh symbol. The results are displayed in a table with three columns: `request_received_time`, `client_request`, and `server_status`. Each column has a dropdown arrow. The table contains two rows of data, both with a `server_status` of 404.

	<code>request_received_time</code>	<code>client_request</code>	<code>server_status</code>
1	[11/Jan/2020:11:40:11 -0700]	GET /image.png HTTP/1.1	404
2	[15/Feb/2019:10:12:22 -0700]	GET /favicon.ico HTTP/1.1	404

### Example – 篩選成功的請求

下列查詢範例會選取使用者 ID、請求接收時間、用戶端請求的文字，以及 `apache_logs` 資料表中的伺服器狀態碼。HTTP 狀態碼 200 的 WHERE 子句篩選條件 (成功)。

```
SELECT user_id, request_received_time, client_request, server_status
FROM apache_logs
WHERE server_status = '200'
```

下圖顯示 Athena 查詢編輯器中查詢的結果。

Results				
	user_id	request_received_time	client_request	server_status
1	Li	[10/Oct/2019:13:55:36 -0700]	GET /logo.gif HTTP/1.0	200
2	Jorge	[24/Nov/2019:10:49:52 -0700]	GET /index.html HTTP/1.1	200
3	Mateo	[27/Dec/2019:11:38:12 -0700]	GET /about.html HTTP/1.1	200
4	Saanvi	[14/Mar/2019:11:40:33 -0700]	GET /intro.html HTTP/1.1	200
5	Xiulan	[22/Apr/2019:10:51:34 -0700]	GET /group/index.html HTTP/1.1	200

### Example – 按時間戳記過濾

下列範例會查詢要求接收時間大於指定時間戳記的記錄。

```
SELECT * FROM apache_logs WHERE request_received_time > 10/Oct/2023:00:00:00
```

### 查詢存放在 Amazon S3 中的網際網路資訊伺服器 (IIS) 日誌

您可以使用 Amazon Athena 查詢存放在您的 Amazon S3 帳戶中的 Microsoft Internet Information Services (IIS) Web 伺服器日誌。雖然 IIS 使用各種日誌檔案格式，本主題將說明如何建立資料表結構描述，以從 Athena 查詢 W3C 擴充和 IIS 日誌檔案格式。

由於 W3C 延伸檔和 IIS 記錄檔格式使用單一字元分隔符號 (分別是空格和逗號)，而且沒有以引號括住的值，因此您可以使用為這些格式建立 Athena 表格。[LazySimpleSerDe](#)

#### W3C 擴充日誌檔案格式

[W3C 擴充](#)日誌檔案資料格式具有以空格分隔的欄位。W3C 擴充日誌中顯示的欄位是由 Web 伺服器管理員決定，管理員可選擇要包含哪些日誌欄位。下列日誌資料範例具有 date、time、c-ip、s-ip、cs-method、cs-uri-stem、sc-status、sc-bytes、cs-bytes、time-taken 以及 cs-version。

```
2020-01-19 22:48:39 203.0.113.5 198.51.100.2 GET /default.html 200 540 524 157 HTTP/1.0
2020-01-19 22:49:40 203.0.113.10 198.51.100.12 GET /index.html 200 420 324 164 HTTP/1.0
2020-01-19 22:50:12 203.0.113.12 198.51.100.4 GET /image.gif 200 324 320 358 HTTP/1.0
2020-01-19 22:51:44 203.0.113.15 198.51.100.16 GET /faq.html 200 330 324 288 HTTP/1.0
```

## 在 Athena 中為 W3C 擴充日誌建立資料表

在查詢 W3C 擴充日誌之前，您必須建立資料表結構描述，以便讓 Athena 可以讀取日誌資料。

### 若要在 Athena 中為 W3C 擴充日誌建立資料表

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 將如下所示的 DDL 陳述式貼上至 Athena 主控台，並注意下列幾點：
  - a. 新增或移除範例中的資料欄，以對應至您要查詢的日誌欄位。
  - b. W3C 擴充日誌檔案格式中的資料欄名稱包含連字號 (-)。然而，按照 [Athena 命名慣例](#)，範例 CREATE TABLE 陳述式會用底線 (\_) 替換連字號。
  - c. 若要指定空格分隔符號，請使用 FIELDS TERMINATED BY ' '。
  - d. 修改 LOCATION 's3://DOC-EXAMPLE-BUCKET/*w3c-log-folder*/' 中的值，以指向 Amazon S3 中的 W3C 擴充日誌。

```
CREATE EXTERNAL TABLE `iis_w3c_logs`(  
  date_col string,  
  time_col string,  
  c_ip string,  
  s_ip string,  
  cs_method string,  
  cs_uri_stem string,  
  sc_status string,  
  sc_bytes string,  
  cs_bytes string,  
  time_taken string,  
  cs_version string  
)  
ROW FORMAT DELIMITED  
  FIELDS TERMINATED BY ' '  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.mapred.TextInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/w3c-log-folder'
```

3. 在 Athena 主控台中執行查詢來註冊 iis\_w3c\_logs 資料表。查詢完成時，您就可以從 Athena 查詢日誌了。

## W3C 擴充日誌選取查詢範例

下面的查詢範例會從資料表 `iis_w3c_logs` 中選取日期、時間、請求目標以及請求所花費的時間。WHERE 子句篩選 HTTP 方法為 GET 且 HTTP 狀態碼為 200 (成功) 的案例。

```
SELECT date_col, time_col, cs_uri_stem, time_taken
FROM iis_w3c_logs
WHERE cs_method = 'GET' AND sc_status = '200'
```

下圖顯示 Athena 查詢編輯器中查詢的結果。

Results				
	date_col ▼	time_col ▼	cs_uri_stem ▼	time_taken ▼
1	2020-01-19	22:48:39	/default.html	157
2	2020-01-19	22:49:40	/index.html	164
3	2020-01-19	22:50:12	/image.gif	358
4	2020-01-19	22:51:44	/faq.html	288

結合 Date (日期) 和 Time (時間) 欄位

以空格分隔的 date 和 time 欄位是日誌來源資料中的單獨項目，您可以根據需要將它們合併成時間戳記。使用 [SELECT](#) 或 [CREATE TABLE AS SELECT](#) 查詢中的 [concat\(\)](#) 和 [date\\_parse\(\)](#) 函數，串連並轉換日期和時間資料欄為時間戳記格式。下列範例使用 CTAS 查詢來建立包含 `derived_timestamp` 資料欄的新資料表。

```
CREATE TABLE iis_w3c_logs_w_timestamp AS
SELECT
  date_parse(concat(date_col, ' ', time_col), '%Y-%m-%d %H:%i:%s') as derived_timestamp,
  c_ip,
  s_ip,
  cs_method,
  cs_uri_stem,
  sc_status,
  sc_bytes,
  cs_bytes,
  time_taken,
  cs_version
```

```
FROM iis_w3c_logs
```

建立資料表之後，您就可以直接查詢新的時間戳記資料欄，如下例所示。

```
SELECT derived_timestamp, cs_uri_stem, time_taken
FROM iis_w3c_logs_w_timestamp
WHERE cs_method = 'GET' AND sc_status = '200'
```

下圖顯示查詢的結果。

Results			
	▲ derived_timestamp ▼	cs_uri_stem ▼	time_taken ▼
1	2020-01-19 22:48:39.000	/default.html	157
2	2020-01-19 22:49:40.000	/index.html	164
3	2020-01-19 22:50:12.000	/image.gif	358
4	2020-01-19 22:51:44.000	/faq.html	288

## IIS 日誌檔案格式

與 W3C 擴充格式不同，[IIS 日誌檔案格式](#)具有固定的欄位集，並包含逗號作為分隔符號。會 LazySimpleSerDe 將逗號視為分隔符號，逗號後的空格視為下一個欄位的開頭。

以下範例會以 IIS 日誌檔案格式顯示資料範例。

```
203.0.113.15, -, 2020-02-24, 22:48:38, W3SVC2, SERVER5, 198.51.100.4, 254, 501, 488,
200, 0, GET, /index.htm, -,
203.0.113.4, -, 2020-02-24, 22:48:39, W3SVC2, SERVER6, 198.51.100.6, 147, 411, 388,
200, 0, GET, /about.html, -,
203.0.113.11, -, 2020-02-24, 22:48:40, W3SVC2, SERVER7, 198.51.100.18, 170, 531, 468,
200, 0, GET, /image.png, -,
203.0.113.8, -, 2020-02-24, 22:48:41, W3SVC2, SERVER8, 198.51.100.14, 125, 711, 868,
200, 0, GET, /intro.htm, -,
```

## 在 Athena 中為 IIS 日誌檔案建立資料表

若要在 Amazon S3 中查詢 IIS 日誌檔案格式日誌，請先建立資料表結構描述，以便讓 Athena 可以讀取日誌資料。

## 若要在 Athena 中為 IIS 日誌檔案格式日誌建立資料表

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 將下列 DDL 陳述式貼上至 Athena 主控台，並注意下列幾點：
  - a. 若要指定逗號分隔符號，請使用 FIELDS TERMINATED BY ','。
  - b. 修改位置 's3://DOC-EXAMPLE-BUCKET/*iis-log-file-folder*' 中的值，以指向 Amazon S3 中的 IIS 日誌格式日誌檔。

```
CREATE EXTERNAL TABLE `iis_format_logs`(  
  client_ip_address string,  
  user_name string,  
  request_date string,  
  request_time string,  
  service_and_instance string,  
  server_name string,  
  server_ip_address string,  
  time_taken_millisec string,  
  client_bytes_sent string,  
  server_bytes_sent string,  
  service_status_code string,  
  windows_status_code string,  
  request_type string,  
  target_of_operation string,  
  script_parameters string  
)  
ROW FORMAT DELIMITED  
  FIELDS TERMINATED BY ','  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.mapred.TextInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/iis-log-file-folder'
```

3. 在 Athena 主控台中執行查詢來註冊 `iis_format_logs` 資料表。查詢完成時，您就可以從 Athena 查詢日誌了。

## IIS 日誌格式選取查詢範例

下列查詢範例從資料表 `iis_format_logs` 中選取請求日期、請求時間、請求目標和以毫秒為單位的花費時間。WHERE 子句篩選請求類型為 GET 且 HTTP 狀態碼為 200 (成功) 的案例。在查詢中，請注意 ' GET' 和 ' 200' 中需要有前置空格，才能使查詢成功。

```
SELECT request_date, request_time, target_of_operation, time_taken_millisecond
FROM iis_format_logs
WHERE request_type = ' GET' AND service_status_code = ' 200'
```

下圖顯示範例資料的查詢結果。

Results				
	request_date ▼	request_time ▼	target_of_operation ▼	time_taken_millisecond ▼
1	2020-02-24	22:48:38	/index.htm	254
2	2020-02-24	22:48:39	/about.html	147
3	2020-02-24	22:48:40	/image.png	170
4	2020-02-24	22:48:41	/intro.htm	125

## NCSA 日誌檔案格式

IIS 也會使用 [NCSA 記錄](#) 格式，該格式擁有以空格分隔的 ASCII 文字格式的、固定數量的欄位。該結構類似於 Apache 存取日誌所用的一般日誌格式。NCSA 一般日誌資料格式的欄位包括用戶端 IP 地址、用戶端 ID (通常不會使用)、網域\使用者 ID、請求接收的時間戳記、用戶端請求的文字、伺服器狀態碼，以及傳回給用戶端之物件的大小。

下列範例會顯示 NCSA 一般日誌格式的資料，如為 IIS 所記錄之內容。

```
198.51.100.7 - ExampleCorp\Li [10/Oct/2019:13:55:36 -0700] "GET /logo.gif HTTP/1.0" 200
232
198.51.100.14 - AnyCompany\Jorge [24/Nov/2019:10:49:52 -0700] "GET /index.html
HTTP/1.1" 200 2165
198.51.100.22 - ExampleCorp\Mateo [27/Dec/2019:11:38:12 -0700] "GET /about.html
HTTP/1.1" 200 1287
198.51.100.9 - AnyCompany\Nikki [11/Jan/2020:11:40:11 -0700] "GET /image.png HTTP/1.1"
404 230
198.51.100.2 - ExampleCorp\Ana [15/Feb/2019:10:12:22 -0700] "GET /favicon.ico HTTP/1.1"
404 30
```

```
198.51.100.13 - AnyCompany\Saanvi [14/Mar/2019:11:40:33 -0700] "GET /intro.html
HTTP/1.1" 200 1608
198.51.100.11 - ExampleCorp\Xiulan [22/Apr/2019:10:51:34 -0700] "GET /group/index.html
HTTP/1.1" 200 1344
```

## 在 Athena 中為 IIS NCSA 日誌建立資料表

對於 CREATE TABLE 陳述式，您可以使用 [GROK SerDe](#) 和類似於 [Apache Web 伺服器日誌](#) 模式的 grok 模式。與 Apache 日誌不同，grok 模式將 `%{DATA:user_id}` (而不是 `%{USERNAME:user_id}`) 用於第三個欄位，以考慮反斜線在 `domain\user_id` 中的存在。如需有關使用 Grok 的詳細資訊 SerDe，請參閱開發人員指南中的 [撰寫 grok 自訂分類器](#)。AWS Glue

若要在 Athena 中為 IIS NCSA Web 伺服器日誌建立資料表

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 將下列 DDL 陳述式貼上至 Athena 查詢編輯器。修改 LOCATION `'s3://DOC-EXAMPLE-BUCKET/iis-ncsa-logs/'` 中的值，以指向 Amazon S3 中的 IIS NCSA 日誌。

```
CREATE EXTERNAL TABLE iis_ncsa_logs(
  client_ip string,
  client_id string,
  user_id string,
  request_received_time string,
  client_request string,
  server_status string,
  returned_obj_size string
)
ROW FORMAT SERDE
  'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.format'='^%{IPV4:client_ip} %{DATA:client_id} %{DATA:user_id}
%{GREEDYDATA:request_received_time} %{QUOTEDSTRING:client_request}
%{DATA:server_status} %{DATA: returned_obj_size}$'
)
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/iis-ncsa-logs/';
```

3. 在 Athena 主控台中執行查詢來註冊 `iis_ncsa_logs` 資料表。查詢完成時，您就可以從 Athena 查詢日誌了。

### IIS NCSA 日誌的選取查詢範例

#### Example – 篩選 404 錯誤

下列查詢範例會從 `iis_ncsa_logs` 資料表選取請求接收的時間、用戶端請求的文字，以及伺服器狀態碼。WHERE 子句會篩選 HTTP 狀態碼 404 (找不到頁面)。

```
SELECT request_received_time, client_request, server_status
FROM iis_ncsa_logs
WHERE server_status = '404'
```

下圖顯示 Athena 查詢編輯器中查詢的結果。



The screenshot shows the Athena query results interface. At the top right, there are icons for a document and a refresh symbol. The results are displayed in a table with three columns: `request_received_time`, `client_request`, and `server_status`. Each column header has a dropdown arrow. The table contains two rows of data, both with a `server_status` of 404.

	<code>request_received_time</code>	<code>client_request</code>	<code>server_status</code>
1	[11/Jan/2020:11:40:11 -0700]	GET /image.png HTTP/1.1	404
2	[15/Feb/2019:10:12:22 -0700]	GET /favicon.ico HTTP/1.1	404

#### Example – 篩選來自特定網域的成功請求

下列查詢範例會選取使用者 ID、請求接收時間、用戶端請求的文字，以及 `iis_ncsa_logs` 資料表中的伺服器狀態碼。WHERE 子句篩選來自 200 網域中使用者的包含 HTTP 狀態碼 AnyCompany (成功) 的請求。

```
SELECT user_id, request_received_time, client_request, server_status
FROM iis_ncsa_logs
WHERE server_status = '200' AND user_id LIKE 'AnyCompany%'
```

下圖顯示 Athena 查詢編輯器中查詢的結果。

Results			
user_id	request_received_time	client_request	server_status
1	AnyCompany\Jorge	[24/Nov/2019:10:49:52 -0700]	GET /index.html HTTP/1.1 200
2	AnyCompany\Saanvi	[14/Mar/2019:11:40:33 -0700]	GET /intro.html HTTP/1.1 200

## 使用 Athena ACID 交易

「ACID 交易」一詞是指一組可確保資料庫交易中資料完整性的屬性 ([不可分割性](#)、[一致性](#)、[隔離](#)和[持久性](#))。ACID 交易可讓多位使用者以原子方式同時可靠地新增和刪除 Amazon S3 物件，同時透過維持對資料湖查詢的讀取一致性來隔離任何現有查詢。Athena ACID 交易為 Athena SQL 資料處理語言 (DML) 新增對插入、刪除、更新和時間歷程操作的單一資料表支援。您和多個並行使用者可以使用 Athena ACID 交易，對 Amazon S3 資料進行可靠的資料列層級修改。Athena 交易會自動管理鎖定語義和協調，並且不需要自訂記錄鎖定解決方案。

Athena ACID 交易和熟悉的 SQL 語法可簡化您的業務和法規資料的更新。例如，若要回應資料清除請求，您可以執行 SQL DELETE 操作。若要進行手動記錄更正，您可以使用單一 UPDATE 陳述式。若要復原最近刪除的資料，您可以使用 SELECT 陳述式發佈時間歷程查詢。

因為它們是建立在共用資料表格式上，Athena ACID 交易與其他服務和引擎相容，例如 [Amazon EMR](#) 和 [Apache Spark](#) 也支援共用資料表格式。

Athena 交易可透過 Athena 主控台、API 操作以及 ODBC 和 JDBC 驅動程式取得。

### 主題

- [查詢 Linux Foundation Delta Lake 資料表](#)
- [使用 Athena 查詢 Apache Hudi 資料集](#)
- [使用 Apache Iceberg 資料表](#)

## 查詢 Linux Foundation Delta Lake 資料表

Linux Foundation [Delta Lake](#) 是用於大數據分析的資料表格式。您可以使用 Amazon Athena 直接讀取存放在 Amazon S3 中的 Delta Lake 資料表，而無需產生清單檔案或執行 MSCK REPAIR 陳述式。

Delta Lake 格式會儲存每個資料檔案中每個資料欄的最小值和最大值。Athena 實作會利用這項資訊來啟用述詞上的檔案略過功能，從而排除不需要的檔案。

## 考量與限制

Athena 的三角洲湖泊支援有下列考量和限制：

- 僅含 AWS Glue 目錄的表格 — 只有透過使用註冊的表格才支援原生 Delta Lake 支援 AWS Glue。如果有已在其他中繼存放區註冊的 Delta Lake 資料表，則您仍然可以保留並將其視為您的主要中繼存放區。由於 Delta Lake 中繼資料儲存在檔案系統中 (例如，Amazon S3) 而非中繼存放區中，因此 Athena 只需要中的位置屬性即可從 Delta Lake 資料表中 AWS Glue 讀取。
- V3 engine only (僅限 V3 引擎) – 只有 Athena 引擎版本 3 才支援 Delta Lake 查詢。您必須確定您建立的工作群組設定為使用 Athena 引擎版本 3。
- Delta 湖讀取器版本 — 支援最高版本 3 的 Delta 湖讀取器通訊協定。
- 資料欄對應和 TimeStampNTZ — [差異資料欄對應](#)，可讓差異資料表資料行和基礎的 Parquet 檔案資料行使用不同的名稱，並支援不含時區的時間戳記 ([Time StampNTZ](#))。
- No time travel support (不支援時間歷程) – 不支援使用 Delta Lake 時間歷程功能的查詢。
- Read only (唯讀) – 不支援寫入 UPDATE、INSERT 或 DELETE 等 DML 陳述式。
- Lake Formation 支援 – Lake Formation 可用於結構描述與 AWS Glue 同步的 Delta Lake 資料表。如需詳細資訊，請參閱 [AWS Lake Formation 開發人員指南中的搭配 Amazon Athena 使用和設定三角洲湖的許可表格](#)。
- Limited DDL support (有限的 DDL 支援) – 支援下列 DDL 陳述式：CREATE EXTERNAL TABLE、SHOW COLUMNS、SHOW TBLPROPERTIES、SHOW PARTITIONS、SHOW CREATE TABLE 和 DESCRIBE。如需有關使用 CREATE EXTERNAL TABLE 陳述式的詳細資訊，請參閱 [開始使用](#) 一節。
- 略過不支援的 S3 Glacier 物件 – 如果 Linux Foundation Delta Lake 資料表中的物件位於 Amazon S3 Glacier 儲存類別中，則將 `read_restored_glacier_objects` 資料表屬性設定為 `false` 是無效的。

例如，假設您發出下列命令：

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'false')
```

對於 Iceberg 和 Delta Lake 資料表，此命令會產生錯誤不支援的資料表屬性索引鍵：`read_restored_glacier_objects`。對於 Hudi 資料表，ALTER TABLE 命令不會產生錯誤，但是 Amazon S3 Glacier 物件仍不會略過。在 ALTER TABLE 命令繼續傳回所有物件之後執行 SELECT 查詢。

## 支援的非分割區資料欄資料類型

對於非分割區資料欄，支援 Athena 支援的所有資料類型，除了 CHAR 之外 (Delta Lake 通訊協定本身不支援 CHAR)。支援的資料類型包括：

```
boolean
tinyint
smallint
integer
bigint
double
float
decimal
varchar
string
binary
date
timestamp
array
map
struct
```

## 支援的分割區資料欄資料類型

對於分割區資料欄，Athena 支援具有下列資料類型的資料表：

```
boolean
integer
smallint
tinyint
bigint
decimal
float
double
date
timestamp
varchar
```

如需有關 Athena 中資料類型的詳細資訊，請參閱 [Amazon Athena 中的資料類型](#)。

## 開始使用

若要可查詢，您的 Delta 湖資料表必須存在於中 AWS Glue。如果您的資料表位於 Amazon S3 中，但不在中 AWS Glue，請使用下列語法執行 CREATE EXTERNAL TABLE 陳述式。如果您的資料表已存在於中 AWS Glue (例如，因為您正在使用 Apache Spark 或其他引擎 AWS Glue)，您可以略過此步驟。

```
CREATE EXTERNAL TABLE
  [db_name.]table_name
  LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
  TBLPROPERTIES ('table_type' = 'DELTA')
```

請注意遺漏欄定義、資 SerDe 源庫和其他表格屬性。與傳統 Hive 資料表不同，Delta Lake 資料表中繼資料是從 Delta Lake 交易記錄中推斷出來，並直接同步至 AWS Glue。

### Note

對於 Delta Lake 資料表，不允許包含超過 LOCATION 和 table\_type 屬性的 CREATE TABLE 陳述式。

## 讀取 Delta Lake 資料表

若要查詢 Delta Lake 資料表，請使用標準 SQL SELECT 語法：

```
[ WITH with_query [, ...] ] SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ OFFSET count [ ROW | ROWS ] ]
[ LIMIT [ count | ALL ] ]
```

如需有關 SELECT 語法的詳細資訊，請參閱 Athena 文件中的 [SELECT](#)。

Delta Lake 格式會儲存每個資料檔案中每個資料欄的最小值和最大值。Athena 會利用這項資訊來啟用述詞上的檔案略過功能，從而排除不需要的檔案。

## 同步處理 Delta Lake 中繼資料

AWS Glue 如果您使用 Athena 建立 Delta Lake 資料表，Athena 會將資料表中繼資料 (包括結構定義、分割區資料欄和資料表屬性) 同步化。隨著時間流逝，此中繼資料可能會遺失與交易日誌中基礎資料表中繼資料的同步處理。要讓資料表保持最新，您可以選擇下列其中一個選項：

- 使用「三角洲湖」表格的 AWS Glue 爬行者程式。如需詳細資訊，請參閱「[AWS 大數據部落格](#)」中的「[使用 AWS Glue 檢索器介紹原生 Delta Lake 表格支援](#)」和「[AWS Glue 開發人員指南](#)」中的[排程 AWS Glue 爬蟲](#)。
- 在 Athena 捨棄並重新建立資料表。
- 使用 SDK、CLI 或 AWS Glue 主控台手動更新中的結構描述 AWS Glue。

請注意，下列功能需要您的 AWS Glue 結構描述永遠與交易記錄檔具有相同的結構描述：

- Lake Formation
- 檢視
- 資料列與資料欄篩選

如果您的工作流程不需要任何此功能，而且您不想維持此相容性，則可以在 Athena 中使用 CREATE TABLE DDL，然後在中新增 Amazon S3 路徑做為 SerDe 參數。AWS Glue

使用 Athena 和 AWS Glue 控制台創建三角洲湖表

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在 Athena 查詢編輯器中，使用下列 DDL 建立您的 Delta Lake 資料表。請注意，使用此方法時，TBLPROPERTIES 的值必須是 'spark.sql.sources.provider' = 'delta' 而不是 'table\_type' = 'delta'。

請注意，當您使用 Apache Spark (Athena for Apache Spark) 或大多數其他引擎來建立資料表時，將插入此相同的結構描述 (具有名為 col 的類型 array<string> 的單一資料欄)。

```
CREATE EXTERNAL TABLE
  [db_name.]table_name(col array<string>)
  LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
  TBLPROPERTIES ('spark.sql.sources.provider' = 'delta')
```

3. [請在以下位置開啟 AWS Glue 主控台。](https://console.aws.amazon.com/glue/) <https://console.aws.amazon.com/glue/>
4. 在導覽窗格中，選擇資料目錄、資料表。

5. 在資料表清單中，選擇資料表的連結。
6. 在資料表的頁面上，選擇動作、編輯資料表。
7. 在 Serde 參數區段中，新增索引鍵 **path** 與值 **s3://DOC-EXAMPLE-BUCKET/*your-folder*/**。
8. 選擇儲存。

## 其他資源

如需將 Delta Lake 資料表搭配使用 AWS Glue 並向 Athena 查詢的討論，請參閱[使用開放原始碼 Delta 湖和AWS 大數據部 AWS Glue 落格處理 UPSERT 資料作業](#)。

## 使用 Athena 查詢 Apache Hudi 資料集

[Apache Hudi](#) 是一個開源資料管理架構，可簡化增量資料的處理。記錄級的插入、更新、更新插入和刪除動作會得到更細微的處理，從而減少開銷。Upsert 是指將記錄插入現有資料集 (如果記錄尚不存在)，或者更新記錄 (如果記錄存在) 的能力。

Hudi 處理資料插入和更新事件，而不會建立許多可能會導致分析效能問題的小型檔案。Apache Hudi 會自動追蹤變更和合併檔案，以便檔案保持在最佳的大小。如此可避免建置自訂解決方案，這些解決方案用來監控許多小型檔案並將其重新寫入較少的大型檔案。

Hudi 資料集適用於以下使用案例：

- 遵守隱私權規定，例如[一般資料保護規範](#) (歐盟《一般資料保護規範》(GDPR)) 及[加州消費者隱私法](#) (CCPA)，這些法規會強制執行使用者移除個人資訊或變更其資料使用方式的權利。
- 處理來自感應器和其他物聯網 (IoT) 裝置的串流資料，這些資料需要特定的資料插入和更新事件。
- 實作[變更資料擷取 \(CDC\) 系統](#)。

會使用開放儲存格式將 Hudi 管理的資料集存放在 Amazon S3 中。目前，Athena 可以讀取經壓縮的 Hudi 資料集，但不能寫入 Hudi 資料。Athena 支援高達 0.8.0 的胡迪版本與 Athena 引擎第 2 版，胡迪版本 0.14.0 與 Athena 引擎版本 3。可能會變動。Athena 無法保證與使用較新版本 Hudi 建立之資料表的讀取相容性。如需有關 Athena 引擎版本控制的資訊，請參閱[Athena 引擎版本控制](#)。如需有關 Hudi 功能和版本控制的詳細資訊，請參閱 Apache 網站上的 [Hudi 文件](#)。

## Hudi 資料集資料表類型

Hudi 資料集可以是以下類型之一：

- 寫入時複製 (CoW) – 資料會以單欄式格式 (Parquet) 存放，每次更新都會在寫入期間建立新版本的檔案。
- 讀取時合併 (MoR) – 資料的存放是使用單欄式 (Parquet) 和以資料行為基礎 (Avro) 格式的組合。更新會記錄到以資料行為基礎的 delta 檔案，並視需要壓縮以建立新版本的單欄式檔案。

使用 CoW 資料集，每次有記錄進行更新時，包含記錄的檔案就會以更新的值重寫。若使用 MoR 資料集，每次有更新時，Hudi 只會寫入已變更之記錄的資料行。MoR 更適合較少讀取，而寫入或變更較繁重的工作負載。CoW 更適合資料變更較不頻繁，而讀取作業較為繁重的工作負載。

Hudi 提供三個可用於資料存取的查詢類型：

- 快照查詢 – 查詢會查看截至給定遞交或壓縮動作的資料表的最新快照。對於 MoR 資料表，快照查詢會公開資料表的最新狀態，方法是合併查詢時最新檔案切片的基底和 delta 檔案。
- 增量查詢 – 查詢只會看到自給定的遞交/壓縮以來在資料表中寫入的新資料。這會有效地提供變更串流，以啟用增量資料管道。
- 讀取最佳化查詢 – 對於 MoR 資料表，查詢會看到壓縮的最新資料。對於 CoW 資料表，查詢會看到遞交的最新資料。

下表顯示每個資料表類型可能的 Hudi 查詢類型。

資料表類型	可能的 Hudi 查詢類型
寫入時複製	快照、增量
讀取時合併	快照、增量、讀取最佳化

目前，Athena 支援快照查詢和讀取最佳化查詢，但不支援增量查詢。在 MoR 資料表上，所有公開給讀取最佳化查詢的資料均經過壓縮。這可提供良好的效能，但不包含最新的 delta 遞交。快照查詢包含最新資料，但會產生一些運算開銷，這使得這些查詢的效能較為不佳。

如需有關資料表和查詢類型之間取捨的詳細資訊，請參閱 Apache Hudi 文件中的 [資料表和查詢類型](#)。

Hudi 術語變更：檢視現在是查詢

從版本 0.5.1 開始，Apache Hudi 變更了一些術語。以前的檢視在後續版本中稱為查詢。下表摘要列出新舊術語之間的變更。

舊術語	新術語
CoW：讀取最佳化檢視	快照查詢
MoR：即時檢視	
增量檢視	增量查詢
MoR 讀取最佳化檢視	讀取最佳化查詢

## Bootstrap 操作中的資料表

從 Apache Hudi 版本 0.6.0 開始，引導操作功能可為現有 Parquet 資料集提供更好的效能。引導操作不會重寫資料集，而是只能產生中繼資料，使資料集保持在原位。

您可以使用 Athena，透過引導操作對資料表執行查詢，就像以 Amazon S3 中之資料為基礎的其他資料表一樣。在 CREATE TABLE 陳述式中，請在 LOCATION 子句中指定 Hudi 資料表路徑。

如需使用 Amazon EMR 中的啟動程序作業建立 Hudi 資料表的詳細資訊，請參閱大數據部落格 [Amazon EMR 提供的 Apache Hudi 新功能](#) 一文。AWS

## Hudi 中繼資料清單

Apache Hudi 有一個 [中繼資料表](#)，其中包含索引功能，以改善效能，例如檔案列出、使用資料欄統計數字略過資料以及基於 Bloom 篩選條件的索引。

在這些功能中，Athena 目前僅支援檔案列出索引。檔案列出索引可從維護分割區到檔案映射的索引中擷取資訊，進而消除諸如「清單檔案」之類的檔案系統呼叫。這樣就不需要遞迴列出資料表路徑下的每個分割區來查看檔案系統。當您處理大型資料集時，此索引會大幅縮短在寫入和查詢期間取得檔案清單時所產生的延遲。它還可以避免瓶頸，例如 Amazon S3 LIST 呼叫上的請求限制限流。

### Note

Athena 目前不支援資料略過或 Bloom 篩選索引。

## 啟用 Hudi 中繼資料表

依預設，以中繼資料表為基礎的檔案清單已停用。若要啟用 Hudi 中繼資料表和相關檔案清單功能，請將 `hudi.metadata-listing-enabled` 資料表屬性設定為 `TRUE`。

### 範例

下列 `ALTER TABLE SET TBLPROPERTIES` 範例會啟用範例 `partition_cow` 資料表上的中繼資料表。

```
ALTER TABLE partition_cow SET TBLPROPERTIES('hudi.metadata-listing-enabled'='TRUE')
```

## 考量與限制

- Athena 不支援增量查詢。
- Athena 不支援對 Hudi 資料執行 [CTAS](#) 或 [INSERT INTO](#)。如果您希望 Athena 提供寫入 Hudi 資料集的支援，請將意見回饋傳送至 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。

如需有關寫入 Hudi 資料的詳細資訊，請參閱下列資源：

- [《Amazon EMR 版本指南》](#) 中的 [使用 Hudi 資料集](#)。
- Apache Hudi 文件中的 [寫入資料](#)。
- 不支援在 Athena 中對 Hudi 資料表使用 `MSCK REPAIR TABLE`。如果您需要載入未在中建立的 Hudi 表格 AWS Glue，請使用 [ALTER TABLE ADD PARTITION](#)。
- 略過不支援的 S3 Glacier 物件 – 如果 Apache Hudi 資料表中的物件位於 Amazon S3 Glacier 儲存類別中，則將 `read_restored_glacier_objects` 資料表屬性設定為 `false` 是無效的。

例如，假設您發出下列命令：

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'false')
```

對於 Iceberg 和 Delta Lake 資料表，此命令會產生錯誤不支援的資料表屬性索引鍵：`read_restored_glacier_objects`。對於 Hudi 資料表，`ALTER TABLE` 命令不會產生錯誤，但是 Amazon S3 Glacier 物件仍不會略過。在 `ALTER TABLE` 命令繼續傳回所有物件之後執行 `SELECT` 查詢。

## 其他資源

如需搭配 Athena 搭配使用 Apache Hudi 的其他資源，請參閱下列資源。

## 影片

下列影片顯示如何使用 Amazon Athena，查詢基於 Amazon S3 的資料湖中已經過讀取最佳化的 Apache Hudi 資料集。

### [使用 Amazon Athena 查詢 Apache Hudi 資料集](#)

## 部落格文章

下列 AWS 大數據部落格文章包含如何搭配 Athena 使用 Apache Hudi 的說明。

- [使用 AWS Data Exchange 無縫共用 Apache Hudi 資料集](#)
- [使用 Amazon Kinesis AWS DMS、AWS Glue 串流 ETL 和使用亞馬遜的資料視覺化來建立以 Hudi 為基礎的 Apache near-real-time 交易資料湖 QuickSight](#)

## 建立 Hudi 資料表

本節提供 Athena 中 CREATE TABLE 陳述式的範例，用於 Hudi 資料的已分割和未分割的資料表。

如果您已在中建立了 Hudi 資料表 AWS Glue，您可以直接在 Athena 中查詢它們。在 Athena 中建立分割的 Hudi 資料表時，您必須執行 ALTER TABLE ADD PARTITION 以在查詢前載入 Hudi 資料。

### 寫入時複製 (CoW) 建立資料表範例

#### 未分割的 CoW 資料表

以下範例在 Athena 中建立未分割的 CoW 資料表。

```
CREATE EXTERNAL TABLE `non_partition_cow`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,  
  `_hoodie_record_key` string,  
  `_hoodie_partition_path` string,  
  `_hoodie_file_name` string,  
  `event_id` string,  
  `event_time` string,  
  `event_name` string,  
  `event_guests` int,  
  `event_type` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'
```

```

OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/folder/non_partition_cow/'

```

## 分割的 CoW 資料表

以下範例在 Athena 中建立分割的 CoW 資料表。

```

CREATE EXTERNAL TABLE `partition_cow`(
  `_hoodie_commit_time` string,
  `_hoodie_commit_seqno` string,
  `_hoodie_record_key` string,
  `_hoodie_partition_path` string,
  `_hoodie_file_name` string,
  `event_id` string,
  `event_time` string,
  `event_name` string,
  `event_guests` int)
PARTITIONED BY (
  `event_type` string)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/folder/partition_cow/'

```

以下 ALTER TABLE ADD PARTITION 範例會將兩個分割區新增至範例 partition\_cow 資料表。

```

ALTER TABLE partition_cow ADD
  PARTITION (event_type = 'one') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_cow/one/'
  PARTITION (event_type = 'two') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_cow/two/'

```

## 讀取時合併 (MoR) 建立資料表範例

Hudi 在 MoR 的中繼存放區中建立兩個資料表，一個為用於快照查詢的資料表，另一個為用於讀取最佳化查詢的資料表。這兩個資料表都可供查詢。在 0.5.1 之前的 Hudi 版本中，讀取最佳化查詢的資料表

具有您在建立資料表時指定的名稱。從版本 Hudi 0.5.1 開始，資料表名稱的字尾預設為 `_ro`。快照查詢的資料表名稱是您指定的附有 `_rt` 的名稱。

### 讀取時的未分割合併 (MoR) 資料表

以下範例在 Athena 中建立未分割的 MoR 資料表，用於讀取最佳化查詢。請注意，讀取最佳化查詢使用輸入格式 `HoodieParquetInputFormat`。

```
CREATE EXTERNAL TABLE `nonpartition_mor`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,  
  `_hoodie_record_key` string,  
  `_hoodie_partition_path` string,  
  `_hoodie_file_name` string,  
  `event_id` string,  
  `event_time` string,  
  `event_name` string,  
  `event_guests` int,  
  `event_type` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/folder/nonpartition_mor/'
```

下列範例會在 Athena 中建立用於快照查詢的未分割 MoR 資料表。對於快照查詢，請使用輸入格式 `HoodieParquetRealtimeInputFormat`。

```
CREATE EXTERNAL TABLE `nonpartition_mor_rt`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,  
  `_hoodie_record_key` string,  
  `_hoodie_partition_path` string,  
  `_hoodie_file_name` string,  
  `event_id` string,  
  `event_time` string,  
  `event_name` string,  
  `event_guests` int,  
  `event_type` string)  
ROW FORMAT SERDE
```

```
'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hudi.hadoop.realtime.HoodieParquetRealtimeInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/folder/nonpartition_mor/'
```

## 讀取時分割合併 (MoR) 資料表

下列範例會在 Athena 中建立用於讀取最佳化查詢的分割 MoR 資料表。

```
CREATE EXTERNAL TABLE `partition_mor`(
  `_hoodie_commit_time` string,
  `_hoodie_commit_seqno` string,
  `_hoodie_record_key` string,
  `_hoodie_partition_path` string,
  `_hoodie_file_name` string,
  `event_id` string,
  `event_time` string,
  `event_name` string,
  `event_guests` int)
PARTITIONED BY (
  `event_type` string)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/folder/partition_mor/'
```

以下 ALTER TABLE ADD PARTITION 範例會將兩個分割區新增至範例 partition\_mor 資料表。

```
ALTER TABLE partition_mor ADD
  PARTITION (event_type = 'one') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_mor/one/'
  PARTITION (event_type = 'two') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_mor/two/'
```

下列範例會在 Athena 中建立用於快照查詢的分割 MoR 資料表。

```
CREATE EXTERNAL TABLE `partition_mor_rt`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,  
  `_hoodie_record_key` string,  
  `_hoodie_partition_path` string,  
  `_hoodie_file_name` string,  
  `event_id` string,  
  `event_time` string,  
  `event_name` string,  
  `event_guests` int)  
PARTITIONED BY (  
  `event_type` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hudi.hadoop.realtime.HoodieParquetRealtimeInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/folder/partition_mor/'
```

同樣地，以下 ALTER TABLE ADD PARTITION 範例會將兩個分割區新增至範例 partition\_mor\_rt 資料表。

```
ALTER TABLE partition_mor_rt ADD  
  PARTITION (event_type = 'one') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/  
partition_mor/one/'  
  PARTITION (event_type = 'two') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/  
partition_mor/two/'
```

## 其他資源

- 如需使用 AWS Glue 自訂連接器和 AWS Glue 2.0 工作建立可以透過 Athena 進行查詢的 Apache Hudi 資料表的相關資訊，請參閱 AWS 大數據部落格中的[使用 AWS Glue 自訂連接器寫入 Apache Hudi 資料表](#)。
- 如需有關使用 Apache Hudi 和 Amazon Athena 建立資料湖的資料處理架構的文章，請參閱 AWS 大數據部落格中的[使用 AWS Glue 和 Apache Hudi 簡化資料湖中的操作資料處理](#)。AWS Glue

## 使用 Apache Iceberg 資料表

Athena 支援 Apache 冰山資料表的讀取、時間旅行、寫入和 DDL 查詢，這些資料表使用 Apache 拼花格式處理資料及其中繼存放區的 AWS Glue 目錄。

[Apache Iceberg](#) 是開放式的資料表格式，專用於非常大型的分析資料集。Iceberg 以管理資料表的方式來管理大量的檔案集合並支援新型的分析資料湖操作，如記錄層級的插入、更新、刪除和時間歷程查詢。Iceberg 規格可讓資料表的結構描述和分割區等變化順暢無礙，並專為最佳化 Amazon S3 的使用所設計。Iceberg 也有助於在並行寫入的情況下保證資料的正確性。

如需 Apache Iceberg 的相關詳細資訊，請參閱 <https://iceberg.apache.org/>。

### 考量與限制

Athena 對冰山資料表的支援有下列考量和限制：

- 支持冰山版本-Athena 支持阿帕奇冰山版本 1.4.2。
- 僅含 AWS Glue 目錄的表格 — Athena 僅支援根據[開放原始碼膠水 AWS Glue 目錄實作所定義的規格，針對目錄](#)建立的 Iceberg 表格。
- AWS Glue 僅支援表格鎖定 — 與支援外掛程式自訂鎖定的開放原始碼 Glue 目錄實作不同，Athena 僅支援 AWS Glue 樂觀鎖定。使用 Athena 修改具有任何其他鎖定實作的 Iceberg 資料表將導致潛在的資料遺失和交易中斷。
- 支援的檔案格式 – Athena 中的 Iceberg 檔案格式支援視 Athena 引擎版本耳釘，如下表所示。

Athena 殷勤版本	Parquet	ORC	Avro
2	是	否	否
3	是	是	是

- Iceberg v2 資料表 – Athena 只能建立及操作 Iceberg v2 資料表。有關 v1 和 v2 資料表之間的區別，請參閱 Apache Iceberg 文件中的[格式版本變更](#)。
- 顯示不含時區的時間類型 – 以世界協調時間 (UTC) 顯示不含時區類型的時間和時間戳記。如果在時間資料欄的篩選條件表達式中未指定時區，則使用世界協調時間 (UTC)。
- 與時間戳記相關的資料精確度 – 雖然 Iceberg 支持時間戳記資料類型的精度達到微秒，但 Athena 僅在讀取和寫入的時間戳記中支持達毫秒的精度。針對在手動壓縮操作過程中重寫的時間相關資料欄中的資料，Athena 僅保留毫秒精度。
- 不支援的操作 – Iceberg 資料表不支援以下 Athena 操作。

## • [ALTER TABLE SET LOCATION](#)

- Views (檢視) – 使用 CREATE VIEW 建立 Athena 檢視，如 [使用檢視](#) 中所述。如果您有興趣使用 [Iceberg 檢視規格](#) 來建立檢視，請聯絡 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。
- 中不支援 TTF 管理命令 AWS Lake Formation — 雖然您可以使用 Lake Formation 來管理 TransactionTable 格式 (TTF) 的讀取存取權限，例如 Apache 冰山、Apache Hudi 和 Linux 基金會 三角洲湖，但您無法使用 Lake Formation 來管理作業的權限，例如 VACUUMMERGE，UPDATE 或 OPTIMIZE 用這些表格格式。如需有關 Lake Formation 與雅典娜整合的詳細資訊，請參閱 AWS Lake Formation 開發人員指南中的 [AWS Lake Formation 與 Amazon Athena 搭配使用](#)。
- 依巢狀欄位分割 – 不支援依巢狀欄位分割。嘗試這樣做會產生訊息 NOT\_SUPPORTED：不支援依巢狀欄位分割：*column\_name.nested\_field\_name*。
- 略過不支援的 S3 Glacier 物件 – 如果 Apache Iceberg 資料表中的物件位於 Amazon S3 Glacier 儲存類別中，則將 read\_restored\_glacier\_objects 資料表屬性設定為 false 是無效的。

例如，假設您發出下列命令：

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'false')
```

對於 Iceberg 和 Delta Lake 資料表，此命令會產生錯誤不支援的資料表屬性索引鍵：`read_restored_glacier_objects`。對於 Hudi 資料表，ALTER TABLE 命令不會產生錯誤，但是 Amazon S3 Glacier 物件仍不會略過。在 ALTER TABLE 命令繼續傳回所有物件之後執行 SELECT 查詢。

如果您希望 Athena 支援特定功能，請將意見回饋傳送至 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。

## 主題

- [建立 Iceberg 資料表](#)
- [管理 Iceberg 資料表](#)
- [查詢 Iceberg 資料表中繼資料](#)
- [逐步發展的 Iceberg 資料表結構描述](#)
- [查詢 Iceberg 資料表資料並執行時間歷程](#)
- [更新 Iceberg 資料表資料](#)
- [最佳化處理 Iceberg 資料表](#)
- [Athena 中 Iceberg 資料表支援的資料類型](#)
- [Iceberg 資料表上的其他 Athena 操作](#)

- [其他資源](#)

## 建立 Iceberg 資料表

若要建立 Iceberg 表格以供 Athena 使用，您可以使用此頁面所述的 CREATE TABLE 陳述式，或使用 AWS Glue 爬行者程式。

### 使用 CREATE TABLE 陳述式

Athena 可建立 Iceberg V2 資料表。有關 v1 和 v2 資料表之間的區別，請參閱 Apache Iceberg 文件中的 [格式版本變更](#)。

Athena CREATE TABLE 建立一個不含資料的 Iceberg 資料表。如果資料表使用 [Iceberg 開放原始碼 glue 目錄](#)，您可以直接從外部系統 (如 Apache Spark) 查詢資料表。您不必建立外部資料表。

#### Warning

執行 CREATE EXTERNAL TABLE 會導致錯誤訊息 External keyword not supported for table type ICEBERG (資料表類型 ICEBERG 不支援外部關鍵字)。

若要在 Athena 中建立 Iceberg 資料表，請在 TBLPROPERTIES 子句中將 'table\_type' 資料表屬性設定成 'ICEBERG'，如下列語法摘要所示。

```
CREATE TABLE
  [db_name.]table_name (col_name data_type [COMMENT col_comment] [, ...] )
  [PARTITIONED BY (col_name | transform, ... )]
  LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
  TBLPROPERTIES ( 'table_type' = 'ICEBERG' [, property_name=property_value] )
```

如需有關您可在 Iceberg 資料表中查詢之資料類型的資訊，請參閱 [Athena 中 Iceberg 資料表支援的資料類型](#)。

### 分割

若要建立含有分割區的 Iceberg 資料表，請使用 PARTITIONED BY 語法。必須先在資料欄聲明中宣告要分割的資料欄。PARTITIONED BY 子句中不能包括資料欄類型。您也可以使用 CREATE TABLE 語法來定義 [分割區轉換](#)。若要指定分割時要使用的多個資料欄，請使用逗號 (,) 字元來分隔資料欄，如下列範例所示。

```
CREATE TABLE iceberg_table (id bigint, data string, category string)
```

```

PARTITIONED BY (category, bucket(16, id))
LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
TBLPROPERTIES ( 'table_type' = 'ICEBERG' )

```

下表顯示可用的分割區轉換函數。

函式	描述	支援的類型
<code>year(ts)</code>	按年分割區	date, timestamp
<code>month(ts)</code>	按月分割區	date, timestamp
<code>day(ts)</code>	按日分割區	date, timestamp
<code>hour(ts)</code>	按小時分割區	timestamp
<code>bucket(<i>N</i>, col)</code>	按雜湊值模組 <i>N</i> 儲存貯體分割區。這與 Hive 資料表的雜湊儲存貯體具有相同的概念。	int, long, decimal, date, timestamp, string, binary
<code>truncate(<i>L</i>, col)</code>	按截斷至 <i>L</i> 的值分割區	int, long, decimal, string

Athena 支援 Iceberg 的隱藏分割。如需詳細資訊，請參閱 Apache Iceberg 文件中的 [Iceberg 隱藏分割](#)。

## 資料表屬性

本節說明可在 CREATE TABLE 陳述式的 TBLPROPERTIES 子句中指定為鍵值對的資料表屬性。Athena 僅允許使用預先定義之資料表屬性清單中的鍵值對來建立或更改 Iceberg 資料表。下表顯示您可指定的資料表屬性。如需有關壓縮選項的詳細資訊，請參閱本文件中的 [最佳化處理 Iceberg 資料表](#)。如果您希望 Athena 支援特定的開放原始碼資料表組態屬性，請將意見回饋傳送至 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。

## format

Description	檔案資料格式
-------------	--------

允許的屬性值	支援的檔案格式和壓縮組合會隨 Athena 引擎版本而異。如需詳細資訊，請參閱 <a href="#">檔案格式支援 Iceberg 資料表壓縮</a> 。
預設值	parquet

## write\_compression

Description	檔案壓縮轉碼器
允許的屬性值	支援的檔案格式和壓縮組合會隨 Athena 引擎版本而異。如需詳細資訊，請參閱 <a href="#">檔案格式支援 Iceberg 資料表壓縮</a> 。
預設值	預設寫入壓縮會隨 Athena 引擎版本而異。如需詳細資訊，請參閱 <a href="#">檔案格式支援 Iceberg 資料表壓縮</a> 。

## optimize\_rewrite\_data\_file\_threshold

Description	資料最佳化的專用組態。如果需要最佳化的資料檔案少於指定的閾值，則不會重寫這些檔案。這允許累積更多的資料檔案，從而產生更接近目標大小的檔案，並略過不必要的運算以節省成本。
允許的屬性值	正數。必須小於 50。
預設值	5

## optimize\_rewrite\_delete\_file\_threshold

Description	資料最佳化的專用組態。如果與資料檔案關聯的刪除檔案少於閾值，則不會重寫資料檔案。這允許為每個資料檔案累積更多的刪除檔案，從而節省成本。
允許的屬性值	正數。必須小於 50。
預設值	2

`vacuum_min_snapshots_to_keep`

Description	要保留在資料表主要分支上的快照數目下限。  此值優先於 <code>vacuum_max_snapshot_age_seconds</code> 屬性。如果最小剩餘快照早於 <code>vacuum_max_snapshot_age_seconds</code> 指定的期限，則會保留快照，且會忽略 <code>vacuum_max_snapshot_age_seconds</code> 的值。
允許的屬性值	正數。
預設值	1

`vacuum_max_snapshot_age_seconds`

Description	要在主分支上保留的快照最長期限。如果由 <code>vacuum_min_snapshots_to_keep</code> 指定的剩餘最小快照早於指定的期限，則會忽略此值。該表的行為屬性對應於 Apache 冰山配置中的 <code>history.expire.max-snapshot-age-ms</code> 屬性。
允許的屬性值	正數。
預設值	432,000 秒 (5 天)

## 真空最大元資料檔案至保留

Description	要保留在資料表主分支上的先前中繼資料檔案數目上限。
允許的屬性值	正數。
預設值	100

## CREATE TABLE 陳述式範例

下列範例會建立具有三個資料欄的 Iceberg 資料表。

```
CREATE TABLE iceberg_table (
```

```
id int,  
data string,  
category string)  
PARTITIONED BY (category, bucket(16,id))  
LOCATION 's3://DOC-EXAMPLE-BUCKET/iceberg-folder'  
TBLPROPERTIES (  
  'table_type'='ICEBERG',  
  'format'='parquet',  
  'write_compression'='snappy',  
  'optimize_rewrite_delete_file_threshold'='10'  
)
```

## CREATE TABLE AS SELECT (CTAS)

如需使用 CREATE TABLE AS 陳述式建立 Iceberg 資料表的相關資訊，請參閱 [CREATE TABLE AS](#)，特別注意 [CTAS 資料表屬性](#) 一節。

## 使用 AWS Glue 爬蟲程式

您可以使用 AWS Glue 爬蟲將您的冰山表自動註冊到 AWS Glue Data Catalog。如果您想要從其他 Iceberg 目錄遷移，可以建立並排程 AWS Glue 爬行程式，並提供 Iceberg 表所在的 Amazon S3 路徑。您可以指定 AWS Glue 爬蟲程式可周遊的 Amazon S3 路徑的深度上限。排定爬行者程式之後，AWS Glue 爬行者程式會擷取結構描述資訊，並在每次執行時 AWS Glue Data Catalog 隨結構描述變更來更新。AWS Glue 爬行者程式支援跨快照合併結構描述，並更新 AWS Glue Data Catalog。如需詳細資訊，請參閱 [中 AWS Glue 的資料目錄和爬行者程式](#)。

## 管理 Iceberg 資料表

Athena 支援 Iceberg 資料表的以下資料表 DDL 操作。

### ALTER TABLE RENAME

重新命名資料表。

由於 Iceberg 資料表的資料表中繼資料存放在 Amazon S3 中，因此您可以更新 Iceberg 受管資料表的資料庫和資料表名稱，而不會影響基礎資料表資訊。

### 概要

```
ALTER TABLE [db_name.]table_name RENAME TO [new_db_name.]new_table_name
```

## 範例

```
ALTER TABLE my_db.my_table RENAME TO my_db2.my_table2
```

## ALTER TABLE SET PROPERTIES

新增 Iceberg 資料表的屬性並設定他們的指定值。

根據 [Iceberg 規格](#)，資料表屬性會儲存在 Iceberg 資料表中繼資料檔案中，而不是儲存在 AWS Glue 中。Athena 不接受自訂資料表屬性。有關允許使用的鍵值對，請參閱 [資料表屬性](#) 部分。如果您希望 Athena 支援特定的開放原始碼資料表組態屬性，請將意見回饋傳送至 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。

## 概要

```
ALTER TABLE [db_name.]table_name SET TBLPROPERTIES ('property_name' =  
'property_value' [ , ... ])
```

## 範例

```
ALTER TABLE iceberg_table SET TBLPROPERTIES (  
  'format'='parquet',  
  'write_compression'='snappy',  
  'optimize_rewrite_delete_file_threshold'='10'  
)
```

## ALTER TABLE UNSET PROPERTIES

捨棄 Iceberg 資料表的現有屬性。

## 概要

```
ALTER TABLE [db_name.]table_name UNSET TBLPROPERTIES ('property_name' [ , ... ])
```

## 範例

```
ALTER TABLE iceberg_table UNSET TBLPROPERTIES ('write_compression')
```

## DESCRIBE TABLE

描述資料表資訊。

## 概要

```
DESCRIBE [FORMATTED] [db_name.]table_name
```

當指定 FORMATTED 選項時，輸出會顯示資料表位置和屬性等其他資訊。

## 範例

```
DESCRIBE iceberg_table
```

## DROP TABLE

捨棄 Iceberg 資料表。

### Warning

因為 Iceberg 資料表在 Athena 中被視為受管資料表，因此捨棄 Iceberg 資料表也會刪除資料表中的所有資料。

## 概要

```
DROP TABLE [IF EXISTS] [db_name.]table_name
```

## 範例

```
DROP TABLE iceberg_table
```

## SHOW CREATE TABLE

顯示可用於在 Athena 中重新建立 Iceberg 資料表的 CREATE TABLE DDL 陳述式。如果 Athena 無法重現資料表結構 (例如，由於在資料表中指定了自訂資料表屬性)，則會擲回 UNSUPPORTED (不受支援的) 錯誤。

## 概要

```
SHOW CREATE TABLE [db_name.]table_name
```

## 範例

```
SHOW CREATE TABLE iceberg_table
```

## SHOW TABLE PROPERTIES

顯示 Iceberg 資料表的一個或多個資料表屬性。僅顯示 Athena 支援的資料表屬性。

## 概要

```
SHOW TBLPROPERTIES [db_name.]table_name [('property_name')]
```

## 範例

```
SHOW TBLPROPERTIES iceberg_table
```

## 查詢 Iceberg 資料表中繼資料

在 SELECT 查詢中，您可以在 *table\_name* 之後使用下列屬性來查詢 Iceberg 資料表中繼資料：

- \$files – 顯示資料表的目前資料檔案。
- \$manifests – 顯示資料表的目前檔案清單檔案。
- \$history – 顯示資料表的歷程記錄。
- \$partitions – 顯示資料表的目前分割區。
- \$snapshots – 顯示資料表的快照。
- \$refs – 顯示資料表的參考。

## 語法

下列陳述式會列出 Iceberg 資料表的檔案。

```
SELECT * FROM "dbname". "tablename$files"
```

下列陳述式會列出 Iceberg 資料表的清單檔案。

```
SELECT * FROM "dbname". "tablename$manifests"
```

下列陳述式會顯示 Iceberg 資料表的歷程記錄。

```
SELECT * FROM "dbname". "tablename$history"
```

下列範例會顯示 Iceberg 資料表的分割區。

```
SELECT * FROM "dbname". "tablename$partitions"
```

下列範例會列出 Iceberg 資料表的快照。

```
SELECT * FROM "dbname". "tablename$snapshots"
```

下列範例會顯示 Iceberg 資料表的參考。

```
SELECT * FROM "dbname". "tablename$refs"
```

## 逐步發展的 Iceberg 資料表結構描述

Iceberg 結構描述更新只會變更中繼資料。執行結構描述更新時，不會更改任何資料檔案。

Iceberg 格式支持以下結構描述的變化：

- Add – 在資料表或巢狀 struct 中新增資料欄。
- Drop – 在資料表或巢狀 struct 中移除現有的資料欄。
- Rename – 重新命名巢狀 struct 中現有的資料欄或欄位。
- Reorder – 變更資料欄的順序。
- Type promotion – 增加可用的資料欄、struct 欄位、map 鍵、map 值或 list 元素類型。目前，Iceberg 資料表會支援以下情況：
  - 整數到大整數
  - float 到 double
  - 增加十進制類型的精度

## ALTER TABLE ADD COLUMNS

在現有資料表中新增一或多個資料欄。

### 概要

```
ALTER TABLE [db_name.]table_name ADD COLUMNS (col_name data_type [,...])
```

## 範例

下列範例會將 string 類型的 comment 資料欄新增至 Iceberg 資料表。

```
ALTER TABLE iceberg_table ADD COLUMNS (comment string)
```

下列範例會將 struct 類型的 point 資料欄新增至 Iceberg 資料表。

```
ALTER TABLE iceberg_table  
ADD COLUMNS (point struct<x: double, y: double>)
```

下列範例會將結構的陣列 points 資料欄新增至 Iceberg 資料表。

```
ALTER TABLE iceberg_table  
ADD COLUMNS (points array<struct<x: double, y: double>>)
```

## ALTER TABLE DROP COLUMN

捨棄 Iceberg 資料表的資料欄。

### 概要

```
ALTER TABLE [db_name.]table_name DROP COLUMN col_name
```

## 範例

```
ALTER TABLE iceberg_table DROP COLUMN userid
```

## ALTER TABLE CHANGE COLUMN

變更資料欄的名稱、類型、順序或註解。

### Note

不支援 ALTER TABLE REPLACE COLUMNS。由於 REPLACE COLUMNS 會刪除所有資料欄，然後新增新的資料欄，則不支援 Iceberg。CHANGE COLUMN 是結構描述變化的偏好語法。

### 概要

```
ALTER TABLE [db_name.]table_name
```

```
CHANGE [COLUMN] col_old_name col_new_name column_type
[COMMENT col_comment] [FIRST|AFTER column_name]
```

## 範例

```
ALTER TABLE iceberg_table CHANGE comment blog_comment string AFTER id
```

## SHOW COLUMNS

顯示資料表中的資料欄。

## 概要

```
SHOW COLUMNS (FROM|IN) [db_name.]table_name
```

## 範例

```
SHOW COLUMNS FROM iceberg_table
```

## 查詢 Iceberg 資料表資料並執行時間歷程

若要查詢 Iceberg 資料集，請使用標準 SELECT 陳述式，如下所示。查詢遵循 Apache 冰山[格式 v2 規範](#)，並執 merge-on-read 行位置和相等刪除。

```
SELECT * FROM [db_name.]table_name [WHERE predicate]
```

為了盡量縮短查詢時間，所有述詞都會被推送到資料所在的位置。

## 時間歷程和版本歷程查詢

每個 Apache Iceberg 資料表都會有一份其所含 Amazon S3 物件的版本資訊清單。使用過往的版本資訊清單可查詢時間歷程和版本歷程。

Athena 中的時間歷程查詢可在 Simple Storage Service (Amazon S3) 中查詢截至某個指定日期和時間的資料，歷程資料會統一顯示成一個快照。Athena 中的版本歷程查詢可在 Simple Storage Service (Amazon S3) 中查詢截至某個指定快照 ID 的歷程資料。

## 時間歷程查詢

若要執行時間歷程查詢，請在 SELECT 陳述式的資料表名稱後加上 FOR TIMESTAMP AS OF *timestamp*，如下列範例所示。

```
SELECT * FROM iceberg_table FOR TIMESTAMP AS OF timestamp
```

進行歷程查詢時所要指定的系統時間是時間戳記或含時區的時間戳記。如果沒有指定時間，則 Athena 會假設該值為世界協調時間 (UTC) 的時間戳記。

以下示例時間旅行查詢選擇指定日期和時間的 CloudTrail 數據。

```
SELECT * FROM iceberg_table FOR TIMESTAMP AS OF TIMESTAMP '2020-01-01 10:00:00 UTC'
```

```
SELECT * FROM iceberg_table FOR TIMESTAMP AS OF (current_timestamp - interval '1' day)
```

### 版本歷程查詢

若要執行版本歷程查詢 (即查看指定版本統一顯示的快照)，請在 SELECT 陳述式中的資料表名稱後加上 FOR VERSION AS OF *version*，如下列範例所示。

```
SELECT * FROM [db_name.]table_name FOR VERSION AS OF version
```

**##**參數是與 Iceberg 資料表版本相關的 bigint 快照 ID。

下列範例的版本歷程查詢會選擇指定版本的資料。

```
SELECT * FROM iceberg_table FOR VERSION AS OF 949530903748831860
```

#### Note

Athena 引擎版本 2 版中的 FOR SYSTEM\_TIME AS OF 和 FOR SYSTEM\_VERSION AS OF 子句已由 Athena 引擎版本 3 中的 FOR TIMESTAMP AS OF and FOR VERSION AS OF 子句取代。

### 擷取快照 ID

您可以使用冰山提供的 Java [SnapshotUtil](#) 類來檢索冰山快照 ID，如以下示例所示。

```
import org.apache.iceberg.Table;  
import org.apache.iceberg.aws.glue.GlueCatalog;  
import org.apache.iceberg.catalog.TableIdentifier;
```

```
import org.apache.iceberg.util.SnapshotUtil;

import java.text.SimpleDateFormat;
import java.util.Date;

Catalog catalog = new GlueCatalog();

Map<String, String> properties = new HashMap<String, String>();
properties.put("warehouse", "s3://DOC-EXAMPLE-BUCKET/my-folder");
catalog.initialize("my_catalog", properties);

Date date = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").parse("2022/01/01 00:00:00");
long millis = date.getTime();

TableIdentifier name = TableIdentifier.of("db", "table");
Table table = catalog.loadTable(name);
long oldestSnapshotIdAfter2022 = SnapshotUtil.oldestAncestorAfter(table, millis);
```

## 結合時間和版本的歷程

您可在同一查詢中使用時間歷程查詢和版本歷程查詢語法來指定不同的時間和版本控制條件，如下列範例所示。

```
SELECT table1.*, table2.* FROM
  [db_name.]table_name FOR TIMESTAMP AS OF (current_timestamp - interval '1' day) AS
table1
FULL JOIN
  [db_name.]table_name FOR VERSION AS OF 5487432386996890161 AS table2
ON table1.ts = table2.ts
WHERE (table1.id IS NULL OR table2.id IS NULL)
```

## 使用 Iceberg 資料表建立和查詢檢視

若要在 Iceberg 資料表上建立和查詢 Athena 檢視，請遵循 [使用檢視](#) 中所述使用 CREATE VIEW 檢視。

範例：

```
CREATE VIEW view1 AS SELECT * FROM iceberg_table
```

```
SELECT * FROM view1
```

如果您有興趣使用 [Iceberg 檢視規格](#) 來建立檢視，請聯絡 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。

## 使用 Lake Formation 精細存取控制

Athena 引擎版本 3 支援 Lake Formation 與 Iceberg 資料表的精細存取控制，包括資料欄層級和資料列層級的安全存取控制。此存取控制適用於時間歷程查詢及已執行結構描述演變的資料表。如需詳細資訊，請參閱 [Lake Formation 精細存取控制和 Athena 工作群組](#)。

如果您在 Athena 外部建立了 Iceberg 資料表，則請使用 [Apache Iceberg SDK](#) 版本 0.13.0 或更高版本，以便將您的 Iceberg 資料表資料欄資訊填入 AWS Glue Data Catalog。如果您的 Iceberg 資料表不包含中的欄資訊 AWS Glue，您可以使用 Athena [ALTER TABLE SET PROPERTIES](#) 陳述式或最新的 Iceberg SDK 來修正資料表並更新中的欄資訊。AWS Glue

## 更新 Iceberg 資料表資料

Athena 上的 Iceberg 資料表資料可以直接使用 INSERT、UPDATE 和 DELETE 查詢加以管理。每次資料管理異動都會產生新的快照，使用時間歷程查詢可查詢此快照。UPDATE 和 DELETE 陳述式遵循 Iceberg v2 格式的資料列級 [位置刪除](#) 規格，並強制執行快照隔離。

使用以下命令對 Iceberg 資料表執行資料管理操作。

### INSERT INTO

將資料插入 Iceberg 資料表。Athena Iceberg INSERT INTO 的收費與目前針對外部 Hive 資料表的 INSERT INTO 查詢相同，即按掃描的資料量收費。若要將資料插入 Iceberg 資料表，請使用以下語法，其中 *##* 可能是 VALUES (val1, val2, ...) 或 SELECT (col1, col2, ...) FROM *[db\_name.]table\_name* WHERE *predicate*。如需有關 SQL 語法和語意的詳細資訊，則請參閱 [INSERT INTO](#)。

```
INSERT INTO [db_name.]table_name [(col1, col2, ...)] query
```

下列範例會將值插入資料表 iceberg\_table 中。

```
INSERT INTO iceberg_table VALUES (1, 'a', 'c1')
```

```
INSERT INTO iceberg_table (col1, col2, ...) VALUES (val1, val2, ...)
```

```
INSERT INTO iceberg_table SELECT * FROM another_table
```

## DELETE

Athena Iceberg DELETE 將 Iceberg 位置刪除檔案寫入資料表。這就是所謂的merge-on-read刪除。與刪除相反，copy-on-write刪除效率更高，因為它不會重寫文件數據。merge-on-read 當 Athena 讀取 Iceberg 資料時，它將 Iceberg 位置刪除檔案與資料檔案合併，以產生資料表的最新檢視。若要刪除這些位置刪除檔案，您可以執行[重寫資料壓縮動作](#)。DELETE 操作將按掃描的資料量計費。如需語法，請參閱[DELETE](#)。

下列範例會刪除 iceberg\_table 中 category 值為 c3 的資料列。

```
DELETE FROM iceberg_table WHERE category='c3'
```

## UPDATE

Athena Iceberg UPDATE 將 Iceberg 位置刪除檔案和最新更新的資料列寫為相同交易中的資料檔案。UPDATE 可以想象為 INSERT INTO 和 DELETE。UPDATE 操作將按掃描的資料量計費。如需語法，請參閱[UPDATE](#)。

下列範例會更新資料表 iceberg\_table 中的指定值。

```
UPDATE iceberg_table SET category='c2' WHERE category='c1'
```

## 合併為

有條件地更新、刪除資料列或將其插入至 Iceberg 資料表。單一陳述式可以組合更新、刪除和插入動作。如需語法，請參閱[合併為](#)。

### Note

MERGE INTO 是交易型的並且僅由 Athena 引擎版本 3 中的 Apache Iceberg 資料表支援。

下列範例會從來源資料表 s 中的資料表 t 刪除所有客戶。

```
MERGE INTO accounts t USING monthly_accounts_update s
ON t.customer = s.customer
WHEN MATCHED
THEN DELETE
```

下列範例會使用來源資料表 *s* 中的客戶資訊更新目標資料表 *t*。對於資料表 *s* 中具有相符客戶資料列之資料表 *t* 中的客戶資料列，此範例會在資料表 *t* 中遞增購買項目。如果資料表 *t* 與資料表 *s* 中的客戶資料列不相符，則該範例會將客戶資料列從資料表 *s* 插入到資料表 *t* 中。

```

MERGE INTO accounts t USING monthly_accounts_update s
  ON (t.customer = s.customer)
  WHEN MATCHED
    THEN UPDATE SET purchases = s.purchases + t.purchases
  WHEN NOT MATCHED
    THEN INSERT (customer, purchases, address)
      VALUES(s.customer, s.purchases, s.address)

```

下列範例會使用來源資料表 *s* 中的資訊偶爾更新目標資料表 *t*。此範例會刪除來源地址為森特維爾的任何相符的目標資料列。對於所有其他相符的資料列，此範例會新增來源購買項目，並將目標地址設定為來源地址。如果目標資料表中沒有相符項目，則範例會從來源資料表插入資料列。

```

MERGE INTO accounts t USING monthly_accounts_update s
  ON (t.customer = s.customer)
  WHEN MATCHED AND s.address = 'Centreville'
    THEN DELETE
  WHEN MATCHED
    THEN UPDATE
      SET purchases = s.purchases + t.purchases, address = s.address
  WHEN NOT MATCHED
    THEN INSERT (customer, purchases, address)
      VALUES(s.customer, s.purchases, s.address)

```

## 最佳化處理 Iceberg 資料表

隨著 Iceberg 資料表中的資料累積，由於開啟檔案時所需的處理時間增加，查詢的效率會逐漸降低。如果資料表內含 [delete 檔案](#)，則運算成本會更高。Iceberg 中的 delete 檔案會儲存資料列級的刪除內容，引擎查詢結果時也必須將已刪除的資料列納入查詢範圍。

為了最佳化 Iceberg 資料表查詢的效能，Athena 支援使用手動壓縮的資料表維護命令。壓縮可在不改變資料表內容的情況下，最佳化資料表的版面配置。

### OPTIMIZE

`OPTIMIZE table REWRITE DATA` 壓縮動作會根據相關 delete 檔案的大小和數量，將資料檔案重寫成更好的版面配置。如需有關語法和資料表屬性的詳細資訊，則請參閱 [OPTIMIZE](#)。

## 範例

下列範例將 delete 檔案合併到資料檔案中，並產生接近目標檔案大小的檔案，其中 category 的值是 c1。

```
OPTIMIZE iceberg_table REWRITE DATA USING BIN_PACK
WHERE category = 'c1'
```

## VACUUM

VACUUM 會執行[快照過期](#)和[孤立檔案移除](#)。這些動作會減少中繼資料大小，並移除不在目前資料表狀態的檔案，這些檔案也早於為資料表指定的保留期間。如需有關語法的詳細資訊，則請參閱 [VACUUM](#)。

## 範例

下列範例會使用資料表屬性來將資料表 iceberg\_table 設定為保留最近三天的資料，然後使用 VACUUM 將舊快照集設為過期，並從資料表中移除孤立檔案。

```
ALTER TABLE iceberg_table SET TBLPROPERTIES (
  'vacuum_max_snapshot_age_seconds'='259200'
)

VACUUM iceberg_table
```

## Athena 中 Iceberg 資料表支援的資料類型

Athena 可以查詢包含下列資料類型的 Iceberg 資料表：

```
binary
boolean
date
decimal
double
float
int
list
long
map
string
struct
timestamp without time zone
```

如需有關 Iceberg 資料表類型的詳細資訊，請參閱 Apache 文件中的 [Iceberg 結構描述頁](#)。

下表顯示 Athena 資料類型與 Iceberg 資料表資料類型之間的關係。

Iceberg 類型	Athena 類型	備註
boolean	boolean	
-	tinyint	Athena 中的 Iceberg 資料表不支援。
-	smallint	Athena 中的 Iceberg 資料表不支援。
int	int	在 Athena DML 陳述式中，此類型是 INTEGER。
long	bigint	
double	double	
float	float	
decimal(P, S)	decimal(P, S)	P 是總位數，S 是小數部分的位數。
-	char	Athena 中的 Iceberg 資料表不支援。
string	string	在 Athena DML 陳述式中，此類型是 VARCHAR。
binary	binary	
date	date	
time	-	Athena Iceberg DDL 陳述式僅支援使用 CREATE TABLE 之類的 Iceberg 時間戳記 (不含時區)，但可透過 Athena 查詢所有時間戳記類型。
timestamp	timestamp	
timestamp tz	timestamp tz	
list<E>	array	

Iceberg 類型	Athena 類型	備註
map<K,V>	map	
struct<.. >	struct	
fixed(L)	-	Athena 目前不支援使用 fixed(L) 類型。

如需有關 Athena 中資料類型的詳細資訊，請參閱[Amazon Athena 中的資料類型](#)。

## Iceberg 資料表上的其他 Athena 操作

### 資料庫層級操作

當您搭配使用 [DROP DATABASE](#) 與 CASCADE 選項時，則任何 Iceberg 資料表資料也會被刪除。以下 DDL 操作對 Iceberg 資料表沒有影響。

- [CREATE DATABASE](#)
- [ALTER DATABASE SET DBPROPERTIES](#)
- [SHOW DATABASES](#)
- [SHOW TABLES](#)
- [SHOW VIEWS](#)

### 分割區相關操作

因為 Iceberg 資料表使用[隱藏的分割](#)，您不必直接使用實體分割區。因此，Athena 中的 Iceberg 資料表不支援以下與分割區相關的 DDL 操作：

- [SHOW PARTITIONS](#)
- [ALTER TABLE ADD PARTITION](#)
- [ALTER TABLE DROP PARTITION](#)
- [ALTER TABLE RENAME PARTITION](#)

如果您希望查看 Athena 中的 Iceberg [分割區演變](#)，請將意見回饋傳送至 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。

## 卸載 Iceberg 資料表

Iceberg 資料表可以卸載至 Amazon S3 上資料夾中的檔案。如需相關資訊，請參閱[UNLOAD](#)。

## MSCK REPAIR

由於 Iceberg 資料表會追蹤資料表配置資訊，因此不必也不支援像使用 Hive 資料表那樣執行 [MSCK REPAIR TABLE](#)。

## 其他資源

以下文章載於《AWS 規範指引》文件中。

- [使用 Amazon Athena SQL 與阿帕奇冰山表工作](#)

如需有關搭配使用 Athena 與 Apache Iceberg 資料表的深度文章，請參閱 AWS 大數據部落格中的下列文章。

- [使用亞馬遜動態 B 和亞馬 Amazon Athena，使用阿帕奇冰山實作無伺服器 CDC 流程](#)
- [搭配使用 Amazon Athena 與 Apache Iceberg，加速交易資料湖上的資料科學特徵工程](#)
- [使用 Amazon 雅典娜、亞馬遜 EMR 和建立 Apache 冰山資料湖 AWS Glue](#)
- [使用 Amazon Athena 和 Apache Iceberg 執行資料湖中的 upsert](#)
- [使用 Apache 冰山建置交易資料湖 AWS Glue，並使用和 Amazon Athena 建立跨帳戶資料共 AWS Lake Formation 用](#)
- [在資料湖中使用 Apache Iceberg 來支援增量資料處理](#)
- [建置即時 GDPR 對齊的 Apache Iceberg 資料湖](#)
- [使用 Apache Iceberg，自動將關聯式來源複寫到交易式資料湖中，AWS Glue](#)
- [使用 Amazon Athena 和跨帳戶細粒度許可與 Apache 冰山表進行交互 AWS Lake Formation](#)
- [使用 Apache Iceberg、Amazon EMR Serverless 和 Amazon Athena 建置無伺服器交易資料湖](#)

## Amazon Athena 安全性

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。

安全是 AWS 與您之間共同承擔的責任。[共同責任模型](#) 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 — AWS 負責保護在雲 AWS 端 AWS 服務 中執行的基礎架構。AWS 還為您提供可以安全使用的服務。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。若要進一步了解適用於 Athena 的合規計劃，請參閱 [合規計劃範圍內的 AWS 服務](#)。
- 雲端中的安全性 — 您的責任取決於您使用的資料。AWS 服務 您也必須對其他因素負責，包括資料的敏感度、您組織的需求和適用的法律及法規。

本文件協助您了解如何在使用 Amazon Athena 時套用共同的責任模型。下列主題說明如何設定 Athena 以滿足您的安全與合規目標。您還將學習如何使用其他可 AWS 服務 以幫助您監控和保護 Athena 資源的方法。

## 主題

- [Athena 中的資料保護](#)
- [Athena 中的 Identity and Access Management](#)
- [Athena 中的記錄和監控](#)
- [Amazon Athena 的合規驗證](#)
- [Athena 中的復原功能](#)
- [Athena 中的基礎設施安全](#)
- [Athena 中的組態與漏洞分析](#)
- [使用 Athena 查詢向 AWS Lake Formation 註冊的資料](#)

## Athena 中的資料保護

AWS [共同責任模型](#) 適用於 Amazon Athena 的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱 [資料隱私權常見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同的責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 AWS 服務。

- 使用進階的受管安全服務 (例如 Amazon Macie) , 協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組, 請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊, 請參閱[聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊, 放在標籤或自由格式的文字欄位中, 例如名稱欄位。這包括當您 AWS 服務使用主控台、API 或 AWS SDK 與 Athena 或其他 AWS CLI 人合作時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL, 我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

作為額外的安全步驟, 您也能使用 [aws:CalledVia](#) 全域條件內容索引鍵來限制僅能有從 Athena 進行的請求。如需詳細資訊, 請參閱[搭配 CalledVia 內容金鑰使用 Athena](#)。

## 保護多種資料類型

使用 Athena 建立資料庫和資料表時, 涉及多種類型的資料。這些資料類型包括存放在 Amazon S3 中的來源資料、執行查詢時建立的資料庫和表格的中繼資料, 或是用於探索資料的 AWS Glue 爬蟲、查詢結果資料和查詢歷史記錄。本節討論每個類型的資料, 並提供如何保護它的相關指導。

- Source data (來源資料) – 您將資料庫和資料表的資料存放在 Simple Storage Service (Amazon S3) 中, 而且 Athena 不會修改它。如需詳細資訊, 請參閱《Amazon Simple Storage Service 使用者指南》中的[Amazon S3 中的資料保護](#)。您可以控制來源資料的存取, 並可在 Simple Storage Service (Amazon S3) 中將其加密。您可以使用 Athena, [根據 Simple Storage Service \(Amazon S3\) 中的加密資料集建立資料表](#)。
- 資料庫和表格中繼資料 (結構描述) — Athena 使用 schema-on-read 技術, 這表示當 Athena 執行查詢時, 您的表格定義會套用至 Amazon S3 中的資料。您定義的所有結構描述都會自動儲存, 除非您確實將其刪除。在 Athena 中, 您可以使用 DDL 陳述式來修改資料目錄中繼資料。您也可以刪除資料表定義和結構描述, 不會影響存放在 Simple Storage Service (Amazon S3) 中的基礎資料。您在 Athena 中使用之資料庫和資料表的中繼資料會存放在 AWS Glue Data Catalog 中。

您可以[針對在 using AWS Identity and Access Management \(IAM\) 中註冊的資料庫和表格定義精細的 AWS Glue Data Catalog 存取政策](#)。您也可以[加密 AWS Glue Data Catalog 的中繼資料](#)。如果您加密中繼資料, 請使用[加密中繼資料的許可](#)進行存取。

- 查詢結果和查詢歷史記錄, 包括儲存的查詢 – 查詢結果存放在 Simple Storage Service (Amazon S3) 中您可以選擇來全域指定, 或針對每個工作群組指定的位置。如果未指定, Athena 會在每個案例中使用預設位置。您可以控制 Simple Storage Service (Amazon S3) 儲存貯體的存取, 而您可在這些儲存貯體中存放查詢結果和已儲存的查詢。此外, 您還可以選擇加密您存放在 Simple Storage Service (Amazon S3) 中的查詢結果。您的使用者必須擁有適當的許可, 才能存取 Simple Storage

Service (Amazon S3) 位置和解密檔案。如需詳細資訊，請參閱本文件的[加密 Amazon S3 中存放的 Athena 查詢結果](#)。

Athena 會將查詢歷史記錄保留 45 天。您可以在主控台中使用 Athena API [檢視查詢歷史記錄](#)，並使用 AWS CLI。若要存放查詢超過 45 天，請將其儲存。若要保護對已儲存查詢的存取權限，請在 Athena 中[使用工作群組](#)，限制僅有權檢視已儲存查詢的使用者可存取這些查詢。

## 主題

- [靜態加密](#)
- [傳輸中加密](#)
- [金鑰管理](#)
- [網際網路流量隱私權](#)

## 靜態加密

您可以在 Amazon Athena 中，針對同一區域和有限數量區域的 Simple Storage Service (Amazon S3) 中的加密資料執行查詢。您也可以加密 Amazon S3 中的查詢結果和資料目錄中的 AWS Glue 資料。

您可以加密 Athena 中的以下資產：

- Simple Storage Service (Amazon S3) 中所有查詢的結果，Athena 會將其存放在稱為 Simple Storage Service (Amazon S3) 結果位置的位置。您可以加密存放在 Simple Storage Service (Amazon S3) 中的查詢結果，而不論 Simple Storage Service (Amazon S3) 中的基礎資料集是否加密。如需相關資訊，請參閱[加密 Amazon S3 中存放的 Athena 查詢結果](#)。
- 資料目錄中的 AWS Glue 資料。如需相關資訊，請參閱[AWS Glue Data Catalog 內加密中繼資料的許可](#)。

### Note

當您使用 Athena 讀取加密資料表時，Athena 會使用為表格資料指定的加密選項，而不是查詢結果的加密選項。如果針對查詢結果和表格資料設定個別的加密方法或金鑰，Athena 會讀取資料表資料，而不使用加密選項和金鑰來加密或解密查詢結果。

不過，如果您使用 Athena 將資料插入已加密資料的資料表，Athena 會使用針對查詢結果指定的加密組態來加密插入的資料。例如，如果您為查詢結果指定 CSE\_KMS 加密，Athena 會使用您用於查詢結果加密的 AWS KMS 金鑰識別碼相同，將插入的資料表資料加密 CSE\_KMS。

## 主題

- [支援的 Amazon S3 加密選項](#)
- [Amazon S3 中加密資料的許可](#)
- [AWS Glue Data Catalog 內加密中繼資料的許可](#)
- [加密 Amazon S3 中存放的 Athena 查詢結果](#)
- [根據 Amazon S3 中加密的資料集建立資料表](#)

## 支援的 Amazon S3 加密選項

Athena 支援下列 Simple Storage Service (Amazon S3) 中資料集和查詢結果的加密選項。

加密類型	描述	跨區域支援
<a href="#">SSE-S3</a> (SSE-S3)	使用 Simple Storage Service (Amazon S3) 受管金鑰的伺服器端加密 (SSE)。	是
<a href="#">SSE-KMS</a>	使用 AWS Key Management Service 客戶管理金鑰的伺服器端加密 (SSE)。 <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b> 使用此加密類型時，Athena 不會要求您在建立資料表時指出資料已加密。</p> </div>	是
<a href="#">CSE-KMS</a>	使用 AWS KMS 客戶管理金鑰的用戶端加密 (CSE)。在 Athena 中，此選項要求您使用 CREATE TABLE 陳述式與指定 'has_encrypted_data'='true' 的 TBLPROPERTIES 子句。如需詳細資訊，請參閱 <a href="#">根據 Amazon S3 中加密的資料集建立資料表</a> 。	否

如需使用 Amazon S3 AWS KMS 加密的詳細資訊，請參閱[AWS Key Management Service 開發人員指南](#)中的 [Amazon Simple Storage Service \(Amazon S3\) 使用 AWS KMS 方式 AWS Key Management Service 和使用方式](#)。如需有關搭配 Athena 使用 SSE-KMS 或 CSE-KMS 的詳細資訊，請參閱 AWS 大數據部落格的[啟動：Amazon Athena 新增對查詢加密資料的支援](#)。

## 不支援的選項

不支援下列加密選項：

- 使用客戶提供金鑰的 SSE (SSE-C)。
- 使用用戶端受管金鑰的用戶端加密。
- 非對稱金鑰。

若要比較 Amazon S3 加密選項，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用加密保護資料](#)。

## 用戶端加密工具

對於用戶端加密，請注意有兩種工具可用：

- [Amazon S3 加密用戶端](#) – 這只會為 Amazon S3 加密資料，並由 Athena 支援。
- [AWS Encryption SDK](#)— SDK 可用於在任何地方加密資料，AWS 但 Athena 不直接支援。

這些工具不相容，而且使用一個工具加密的資料無法由另一個工具解密。Athena 僅直接支援 Simple Storage Service (Amazon S3) 加密用戶端。如果您使用軟體開發套件來加密資料，您可以從 Athena 執行查詢，但資料會以加密文字傳回。

如果您想要使用 Athena 來查詢已使用 AWS Encryption 軟體開發套件加密的資料，您必須先下載並解密資料，然後使用 Simple Storage Service (Amazon S3) 加密用戶端再次加密資料。

## Amazon S3 中加密資料的許可

根據您在 Simple Storage Service (Amazon S3) 中使用的加密類型，您可能需要針對在 Athena 中使用的政策新增許可 (也稱為「允許」動作)：

- SSE-S3 – 如果您使用 SSE-S3 進行加密，Athena 使用者在他們的政策中不需要額外的許可。有適當 Simple Storage Service (Amazon S3) 位置和 Athena 動作的適當 Simple Storage Service (Amazon S3) 許可即已足夠。如需有關允許適當 Athena 和 Simple Storage Service (Amazon S3) 許可的政策詳細資訊，請參閱[AWS Amazon Athena 的受管政策](#)和[從 Athena 訪問 Amazon S3](#)。
- AWS KMS— 如果您用 AWS KMS 於加密，除了 Athena 和 Amazon S3 許可之外，還必須允許 Athena 使用者執行特定 AWS KMS 動作。您可以編輯用於在 Amazon S3 中加密資料之 AWS KMS 客戶受管 CMK 的金鑰政策，以允許這些動作。若要將金鑰使用者新增至適當的 AWS KMS 金鑰原則，您可以使用 AWS KMS 主控台，網址為 <https://console.aws.amazon.com/kms>。如需如何將使

用者新增至 AWS KMS 金鑰原則的詳細資訊，請參閱AWS Key Management Service 開發人員指南中的[允許金鑰使用者使用 CMK](#)。

#### Note

進階金鑰政策管理員可以調整金鑰政策。`kms:Decrypt` 是允許 Athena 使用者使用加密的資料集的最低允許動作。若要使用加密的查詢結果，允許的最低動作為 `kms:GenerateDataKey` 和 `kms:Decrypt`。

使用 Athena 在 Amazon S3 中查詢含有大量加密物件的資料集時 AWS KMS，AWS KMS 可能會限制查詢結果。當存在大量小型物件時，這種情況更有可能發生。Athena 會退回重試請求，但仍可能發生調節錯誤。如果您正在使用大量加密物件且遭遇此問題，一個方法是啟用 Simple Storage Service (Amazon S3) 儲存貯體金鑰以減少對 KMS 的呼叫次數。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用 Amazon S3 儲存貯體金鑰降低 SSE-KMS 的成本](#)。另一種方式是提高您 AWS KMS 的服務配額。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[配額](#)。

如需有關搭配 Athena 使用 Simple Storage Service (Amazon S3) 時許可的疑難排解資訊，請參閱[Athena 中的疑難排解](#) 主題的 [許可](#) 一節。

#### AWS Glue Data Catalog 內加密中繼資料的許可

如果您在中[加密中繼資料 AWS Glue Data Catalog](#)，則必須在存取 Athena 時使用的原則新"`kms:GenerateDataKey`"增"`kms:Decrypt`"、和"`kms:Encrypt`"動作。如需相關資訊，請參閱[從 Athena 存取加密的中繼資料 AWS Glue Data Catalog](#)。

#### 加密 Amazon S3 中存放的 Athena 查詢結果

您使用 Athena 主控台或在使用 JDBC 或 ODBC 時設定查詢結果加密。工作群組可讓您強制加密查詢結果。

在主控台中，您透過兩種方式設定查詢結果的加密設定：

- 用戶端設定 – 當您使用主控台的 Settings (設定) 或 API 操作指出您想要加密查詢結果時，這稱為使用用戶端設定。用戶端設定包括查詢結果的位置和加密。如果您指定這些設定，就會派上用場，除非被群組設定所覆寫。

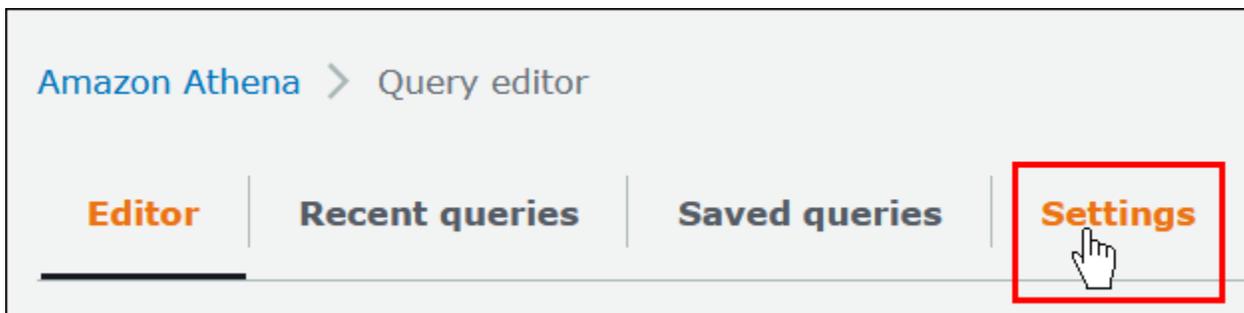
- 工作群組設定 – 當您[建立或編輯工作群組](#)，並選取 Override client-side settings (覆寫用戶端設定) 欄位時，在此工作群組中執行的所有查詢會使用工作群組加密和查詢結果位置設定。如需詳細資訊，請參閱 [工作群組設定覆寫用戶端設定](#)。

若要使用主控台加密存放在 Simple Storage Service (Amazon S3) 中的查詢結果

### ⚠ Important

如果您的工作群組選取 Override client-side settings (覆寫用戶端設定) 欄位，則工作群組中的所有查詢均會使用工作群組設定。不會使用在 Athena 主控台中的 Settings (設定) 索引標籤上指定，或由 API 操作與 JDBC 和 ODBC 驅動程式指定的加密組態和查詢結果位置。如需詳細資訊，請參閱 [工作群組設定覆寫用戶端設定](#)。

1. 在 Athena 主控台中，選擇 Settings (設定)。



2. 選擇管理。
3. 針對 Location of query result (查詢結果的位置)，輸入或選擇 Simple Storage Service (Amazon S3) 路徑。這是儲存查詢結果的 Simple Storage Service (Amazon S3) 位置。
4. 選擇 Encrypt query results (加密查詢結果)。

Amazon Athena > Query editor > Manage settings

## Manage settings

### Query result location and encryption

Location of query result

[View](#) [Browse S3](#)

**Encrypt query results**

#### Encryption type

Choose server-side encryption (SSE) with an S3-managed encryption key (SSE-S3) or a customer master key (CMK) that you provide (SSE-KMS). Or choose client side encryption with a CMK (CSE-KMS).

#### Choose an AWS KMS key

This key will be used to encrypt and decrypt your resources. [Learn more](#)

[Create an AWS KMS key](#)

[Cancel](#) [Save](#)

- 對於 Encryption type (加密類型)，選擇 CSE-KMS、SSE-KMS 或 SSE-S3。在這三個中，CSE-KMS 提供了最高級別的加密，而 SSE-S3 是最低的。
- 如果您選擇 SSE-KMS 或 CSE-KMS，請指定金鑰。AWS KMS

- 對於選擇 AWS KMS 金鑰，如果您的帳戶可存取現有的 AWS KMS 客戶管理金鑰 (CMK)，請選擇其別名或輸入 AWS KMS 金鑰 ARN。
- 如果您的帳戶無法存取現有的客戶管理金鑰 (CMK)，請選擇 [建立 AWS KMS 金鑰]，然後開啟 [AWS KMS 主控台](#)。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [建立金鑰](#)。

 Note

Athena 僅支援用於讀取和寫入資料的對稱金鑰。

7. 返回 Athena 主控台，選擇您透過別名或 ARN 建立的索引鍵。
8. 選擇 Save (儲存)。

### 使用 JDBC 或 ODBC 時加密 Athena 查詢結果

如果您使用 ODBC 或 JDBC 驅動程式連接，您需設定驅動程式選項，以指定要使用的加密類型，以及 Simple Storage Service (Amazon S3) 臨時目錄位置。若要設定 JDBC 或 ODBC 驅動程式，以使用 Athena 支援的任何加密通訊協定來加密您的查詢結果，請參閱 [使用 ODBC 和 JDBC 驅動程式連接到 Amazon Athena](#)。

### 根據 Amazon S3 中加密的資料集建立資料表

建立資料表時，請向 Athena 指出某個資料集已在 Simple Storage Service (Amazon S3) 中加密。使用 SSE-KMS 時，不需要執行此步驟。對於 SSE-S3 和 AWS KMS 加密，Athena 會決定如何解密資料集並建立資料表，因此您不需要提供金鑰資訊。

執行查詢的使用者，包括建立資料表的使用者，必須具有這個主題稍早所述的許可。

 Important

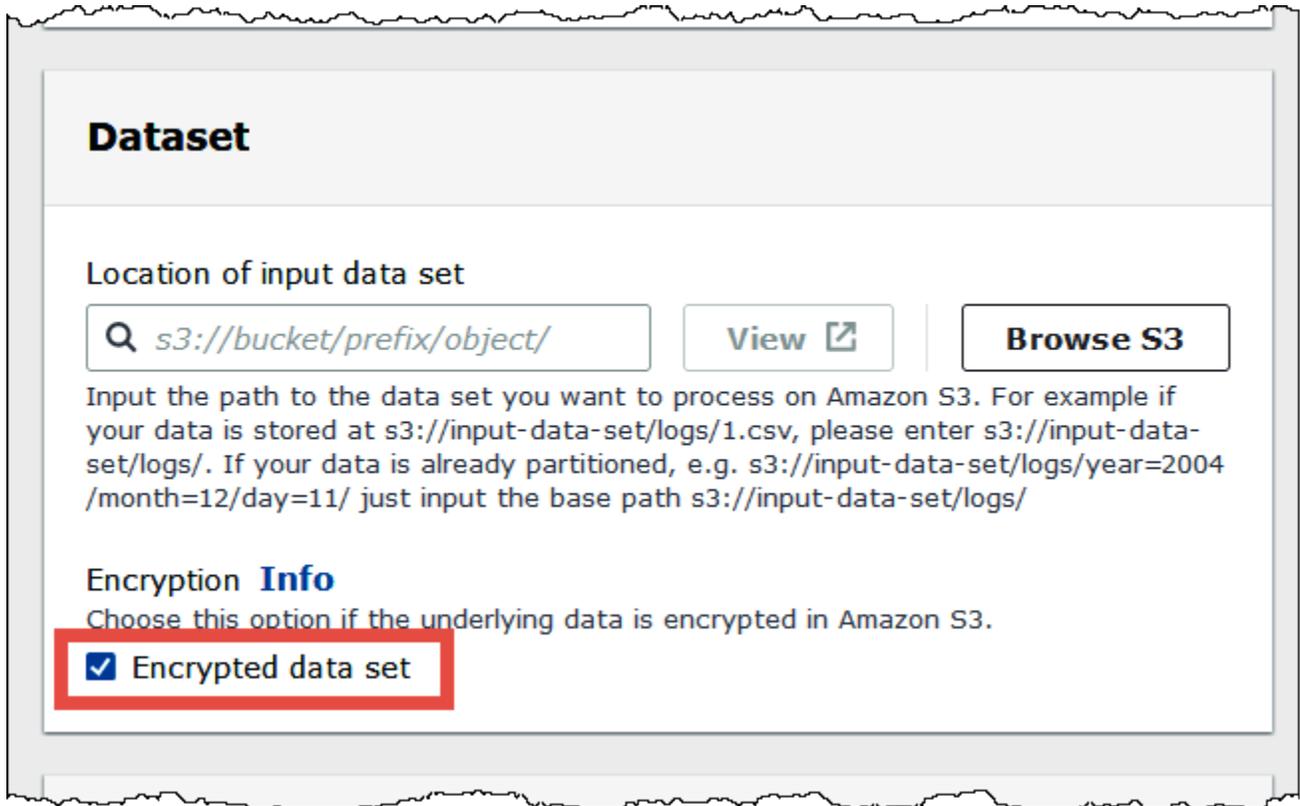
如果您使用 Amazon EMR 搭配 EMRFS 來上傳加密的 Parquet 檔案，您必須將 `fs.s3n.multipart.uploads.enabled` 設定為 `false` 來停用分段上傳功能。如果您不執行此操作，則 Athena 無法判斷 Parquet 檔案長度，並且會發生 `HIVE_CANNOT_OPEN_SPLIT` 錯誤。如需詳細資訊，請參閱《Amazon EMR 管理指南》中的 [設定適用於 Amazon S3 的分段上傳](#)。

若要指出資料集已在 Simple Storage Service (Amazon S3) 中加密，請執行下列其中一個步驟。如果是使用 SSE-KMS，則不需要執行此步驟。

- 在 [CREATE TABLE](#) 陳述式中，請使用指定 'has\_encrypted\_data'='true' 的 TBLPROPERTIES 子句，如以下範例所示。

```
CREATE EXTERNAL TABLE 'my_encrypted_data' (  
  `n_nationkey` int,  
  `n_name` string,  
  `n_regionkey` int,  
  `n_comment` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/folder_with_my_encrypted_data/'  
TBLPROPERTIES (  
  'has_encrypted_data'='true')
```

- 當您使用 `statement.executeQuery()` 執行 [CREATE TABLE](#) 陳述式時，請使用 [JDBC 驅動程式](#)，然後設定 TBLPROPERTIES 值，如上一個範例中所示。
- 使用 Athena 主控台以 [使用表格建立資料表](#) 並指定資料表位置時，請選取 Encrypted data set (加密的資料集) 選項。



## Dataset

Location of input data set

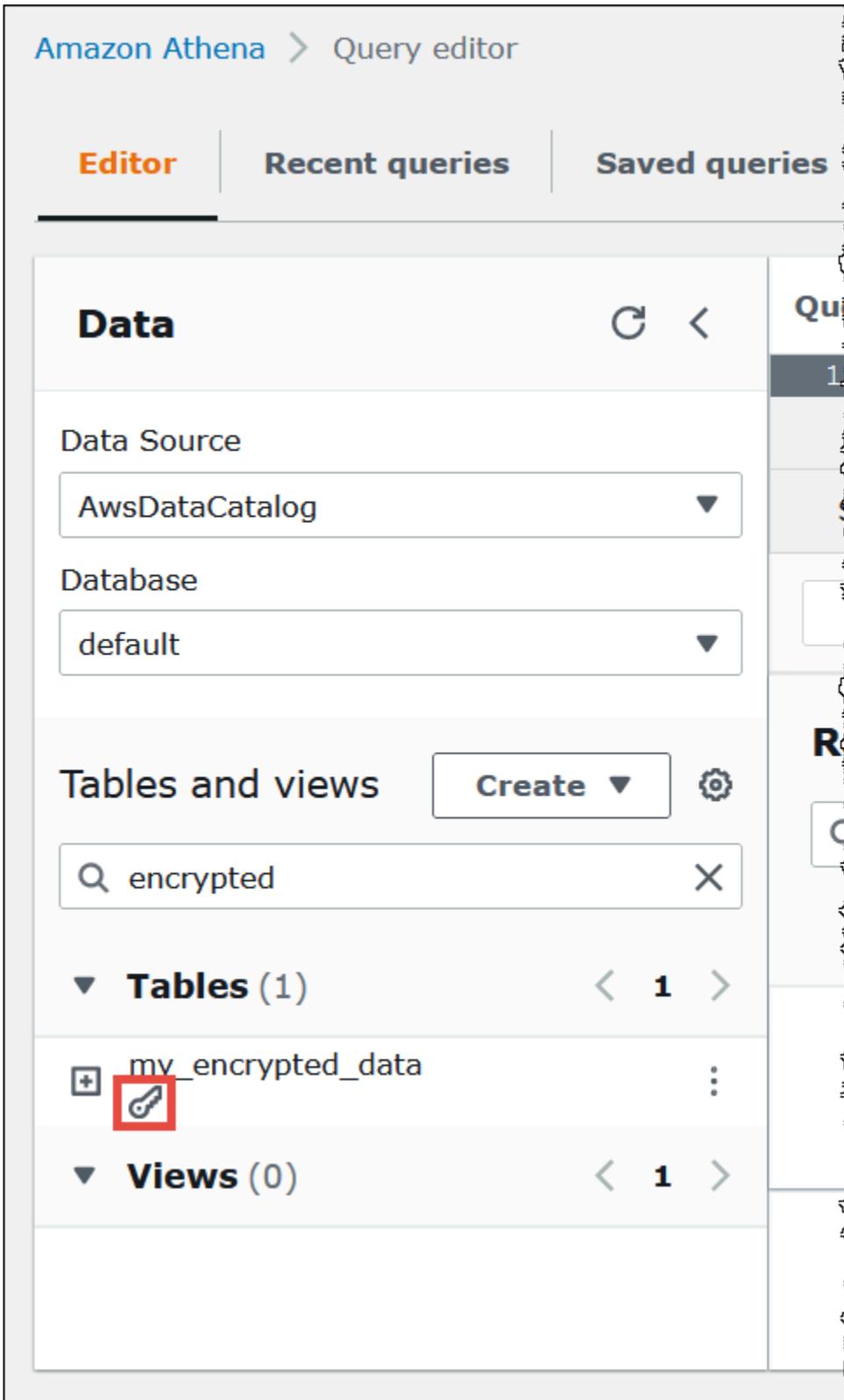
[View](#) [Browse S3](#)

Input the path to the data set you want to process on Amazon S3. For example if your data is stored at `s3://input-data-set/logs/1.csv`, please enter `s3://input-data-set/logs/`. If your data is already partitioned, e.g. `s3://input-data-set/logs/year=2004/month=12/day=11/` just input the base path `s3://input-data-set/logs/`

**Encryption Info**  
Choose this option if the underlying data is encrypted in Amazon S3.

Encrypted data set

在 Athena 主控台資料表清單中，加密資料表會顯示一個鑰匙形圖示。



## 傳輸中加密

除了在 Simple Storage Service (Amazon S3) 中加密靜態資料外，Amazon Athena 還會在 Athena 與 Simple Storage Service (Amazon S3) 之間，以及 Athena 與存取它的客戶應用程式之間，對傳輸中資料使用 Transport Layer Security (TLS) 加密。

您應該使用 Simple Storage Service (Amazon S3) 儲存貯體 IAM 政策上的 [aws:SecureTransport condition](#)，僅允許透過 HTTPS (TLS) 進行加密連線。

串流至 JDBC 或 ODBC 用戶端的查詢結果會使用 TLS 來加密。如需最新版 JDBC 和 ODBC 驅動程式及其文件的詳細資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)和[使用 ODBC 連接到 Amazon Athena](#)。

對於 Athena 聯合資料來源連接器，支援使用 TLS 加密進行傳輸中加密取決於各個連接器。如需相關資訊，請參閱個別[資料來源連接器](#)的文件。

## 金鑰管理

Amazon Athena 支援 AWS Key Management Service (AWS KMS) 來加密 Amazon S3 和 Athena 查詢結果中的資料集。AWS KMS 使用客戶受管金鑰 (CMK) 來加密 Amazon S3 物件，並仰賴[信封加密](#)。

在中 AWS KMS，您可以執行下列動作：

- [建立金鑰](#)
- [為新 CMK 匯入您自己的金鑰材料](#)

### Note

Athena 僅支援用於讀取和寫入資料的對稱金鑰。

如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[什麼是 AWS Key Management Service](#)，以及[Amazon Simple Storage Service 如何使用 AWS KMS](#)。若要檢視帳戶中為您 AWS 建立和管理的金鑰，請在功能窗格中選擇AWS 受管理的金鑰。

如果您要上傳或存取 SSE-KMS 加密的物件，請使用 AWS 簽章版本 4 來增加安全性。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[指定請求身分驗證中的簽章版本](#)。

如果 Athena 工作負載會加密大量資料，您可以使用 Amazon S3 儲存貯體金鑰來降低成本。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用 Amazon S3 儲存貯體金鑰降低 SSE-KMS 的成本](#)。

## 網際網路流量隱私權

同時保護 Athena 與內部部署應用程式之間的流量，以及 Athena 與 Simple Storage Service (Amazon S3) 之間的流量。Athena 與其他服務 (例如 AWS Glue 和 AWS Key Management Service) 之間的流量預設會使用 HTTPS。

- 對於 Athena 與內部部署用戶端和應用程式之間的流量，串流至 JDBC 或 ODBC 用戶端的查詢結果會使用 Transport Layer Security (TLS) 來加密。

您可以在私有網路與 AWS 之間使用其中一個連線選項：

- Site-to-Site VPN AWS VPN 連線。如需詳細資訊，請參閱《AWS VPN 使用者指南》[https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC\\_VPN.html](https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html) 中的什麼是 Site-to-Site VPN AWS Site-to-Site VPN。
- 一個 AWS Direct Connect 連接。如需詳細資訊，請參閱《AWS Direct Connect 使用者指南》中的[什麼是 AWS Direct Connect](#)。
- 對於 Athena 和 Simple Storage Service (Amazon S3) 儲存貯體之間的流量，Transport Layer Security (TLS) 會在 Athena 和 Simple Storage Service (Amazon S3) 之間，以及在 Athena 和存取該物件的客戶應用程式之間加密傳輸中的物件，且您應使用 Simple Storage Service (Amazon S3) 儲存貯體 IAM 政策上的 [aws:SecureTransport condition](#)，僅允許透過 HTTPS (TLS) 的加密連線。儘管 Athena 目前使用公有端點存取 Simple Storage Service (Amazon S3) 儲存貯體中的資料，但這並不意味著資料會周遊公有網際網路。Athena 和 Amazon S3 之間的所有流量都透過 AWS 網路路由，並使用 TLS 加密。
- 合規計劃 — Amazon Athena 符合多種 AWS 合規計劃，包括 SOC、PCI、FedRAMP 等。如需詳細資訊，請參閱[合規計劃的 AWS 服務 範圍](#)。

## Athena 中的 Identity and Access Management

Amazon Athena 使用 [AWS Identity and Access Management \(IAM\)](#) 政策來限制對 Athena 操作的存取。如需 Athena 許可的完整清單，請參閱《服務授權參考》中的[適用於 Amazon Athena 的動作、資源和條件金鑰](#)。

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 中的安全性最佳實務](#)。

執行 Athena 查詢所需的許可包括以下項目：

- Amazon S3 位置，用來存放要查詢的基礎資料。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [Amazon S3 中的 Identity and Access Management](#)。
- 您儲存在中的中繼資料和資源 AWS Glue Data Catalog，例如資料庫和表格，包括加密中繼資料的其他動作。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的 [設定 AWS Glue 的 IAM 許可](#) 以及 [設定 AWS Glue 中的加密](#)。
- Athena API 動作。如需 Athena 中的 API 動作清單，請參閱《Amazon Athena API 參考》中的 [動作](#)。

以下主題會提供有關 Athena 特定區域許可的詳細資訊。

## 主題

- [AWS Amazon Athena 的受管政策](#)
- [透過 JDBC 和 ODBC 連接存取](#)
- [從 Athena 訪問 Amazon S3](#)
- [Athena 中的 Amazon S3 儲存貯體跨帳戶存取](#)
- [對於中的資料庫和資料表的精細存取 AWS Glue Data Catalog](#)
- [跨帳戶存取 AWS Glue 資料目錄](#)
- [從 Athena 存取加密的中繼資料 AWS Glue Data Catalog](#)
- [存取工作群組和標籤](#)
- [允許存取預備陳述式](#)
- [搭配 CalledVia 內容金鑰使用 Athena](#)
- [允許存取外部 Hive 中繼存放區的 Athena 資料連接器](#)
- [允許 Lambda 函數存取外部 Hive 中繼存放區](#)
- [允許 Athena 聯合查詢的 IAM 許可政策範例](#)
- [允許 Amazon Athena 使用者定義函數 \(UDF\) 的 IAM 許可政策範例](#)
- [允許使用 Athena 存取機器學習 \(ML\)](#)
- [啟用對 Athena API 的聯合存取](#)

## AWS Amazon Athena 的受管政策

受 AWS 管理的策略是由建立和管理的獨立策略 AWS。AWS 受管理的策略旨在為許多常見使用案例提供權限，以便您可以開始將權限指派給使用者、群組和角色。

請記住，AWS 受管理的政策可能不會為您的特定使用案例授與最低權限權限，因為這些權限可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的[客戶管理政策](#)，以便進一步減少許可。

您無法變更受 AWS 管理策略中定義的權限。如果 AWS 更新 AWS 受管理原則中定義的權限，則此更新會影響附加原則的所有主體識別 (使用者、群組和角色)。AWS 當新的啟動或新 AWS 服務的 API 操作可用於現有服務時，最有可能更新 AWS 受管理策略。

如需詳細資訊，請參閱《IAM 使用者指南》中的[AWS 受管政策](#)。

### 在搭配 Athena 使用受管政策時的考量事項

受管政策不但易於使用，且會隨著服務的演進，自動更新以具有所需動作。在搭配 Athena 使用受管政策時，請記得以下幾點：

- 若要使用 AWS Identity and Access Management (IAM)，為您自己或其他使用者允許或拒絕 Amazon Athena 服務動作，請將身分型政策連接到委託人，例如使用者或群組。
- 每個身分型政策由陳述式組成，這些陳述式定義了允許或拒絕的動作。如需將政策附加至使用者的詳細資訊和 step-by-step 指示，請參閱 IAM 使用者指南中的[附加受管政策](#)。如需動作的清單，請參閱《[Amazon Athena API 參考](#)》。
- 客戶受管和內嵌的身分型政策，可讓您在政策中指定更詳細的 Athena 動作來微調存取。我們建議您以 AmazonAthenaFullAccess 政策做為起點，然後允許或拒絕 [Amazon Athena API 參考](#) 中列出的特定動作。如需內嵌政策的詳細資訊，請參閱《IAM 使用者指南》中的[受管政策和內嵌政策](#)。
- 如果您也有使用 JDBC 連接的委託人，您必須向您的應用程式提供 JDBC 驅動程式登入資料。如需詳細資訊，請參閱[透過 JDBC 和 ODBC 連接存取](#)。
- 如果您已加密 AWS Glue 資料目錄，則必須在 Athena 的身分型 IAM 政策中指定其他動作。如需詳細資訊，請參閱[從 Athena 存取加密的中繼資料 AWS Glue Data Catalog](#)。
- 如果您建立和使用工作群組，請確保您的政策包含對群組動作的相關存取。如需詳細資訊，請參閱[the section called “用於存取工作群組的 IAM 政策”](#)和[the section called “工作群組政策範例”](#)。

### AWS 受管理的策略：AmazonAthenaFullAccess

AmazonAthenaFullAccess 受管政策會授予 Athena 的完整存取權。

若要提供存取權，請新增許可到您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center :

建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 [建立權限合集](#) 說明進行操作。

- 透過身分提供者在 IAM 中管理的使用者 :

建立聯合身分的角色。請按照 IAM 使用者指南 的 [為第三方身分提供者 \(聯合\) 建立角色](#) 中的指示進行操作。

- IAM 使用者 :

- 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 [為 IAM 使用者建立角色](#) 中的指示進行操作。
- (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的 [新增許可到使用者 \(主控台\)](#) 中的指示。

## 許可群組

AmazonAthenaFullAccess 政策會分組為以下許可集。

- **athena** – 允許委託人存取 Athena 資源。
- **glue**— 允許主參與者存取 AWS Glue 資料庫、表格和分割區。這是必要的，以便主體可以 AWS Glue Data Catalog 與 Athena 一起使用。
- **s3** – 允許委託人寫入和讀取來自 Simple Storage Service (Amazon S3) 的查詢結果，讀取位於 Simple Storage Service (Amazon S3) 中的公開可用 Athena 資料範例，並列出儲存貯體。委託人需要此許可才能使用 Athena 以與 Simple Storage Service (Amazon S3) 搭配運作。
- **sns** – 允許委託人列出 Amazon SNS 主題並取得主題屬性。這可讓委託人將 Amazon SNS 主題搭配 Athena 使用，以用於監控和提醒用途。
- **cloudwatch**— 允許主參與者建立、讀取和刪除 CloudWatch 警報。如需詳細資訊，請參閱 [使用指標和事件控制成本並 CloudWatch 監控查詢](#)。
- **lakeformation** – 允許委託人要求臨時憑證，以存取位於向 Lake Formation 註冊的資料湖位置中的資料。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的 [基礎資料存取控制](#)。
- **datazone**— 允許校長列出 Amazon DataZone 專案、網域和環境。如需 DataZone 在 Athena 使用的相關資訊，請參閱 [在 Athena 中使用 Amazon DataZone](#)。
- **pricing**— 提供存取 AWS Billing and Cost Management. 如需詳細資訊，請參閱 AWS Billing and Cost Management API 參考 [GetProducts](#) 中的。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "BaseAthenaPermissions",
    "Effect": "Allow",
    "Action": [
      "athena:*"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Sid": "BaseGluePermissions",
    "Effect": "Allow",
    "Action": [
      "glue:CreateDatabase",
      "glue>DeleteDatabase",
      "glue:GetDatabase",
      "glue:GetDatabases",
      "glue:UpdateDatabase",
      "glue:CreateTable",
      "glue>DeleteTable",
      "glue:BatchDeleteTable",
      "glue:UpdateTable",
      "glue:GetTable",
      "glue:GetTables",
      "glue:BatchCreatePartition",
      "glue:CreatePartition",
      "glue>DeletePartition",
      "glue:BatchDeletePartition",
      "glue:UpdatePartition",
      "glue:GetPartition",
      "glue:GetPartitions",
      "glue:BatchGetPartition",
      "glue:StartColumnStatisticsTaskRun",
      "glue:GetColumnStatisticsTaskRun",
      "glue:GetColumnStatisticsTaskRuns",
      "glue:GetCatalogImportStatus"
    ],
    "Resource": [
      "*"
    ]
  }
],
```

```

{
  "Sid": "BaseQueryResultsPermissions",
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketLocation",
    "s3:GetObject",
    "s3:ListBucket",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts",
    "s3:AbortMultipartUpload",
    "s3:CreateBucket",
    "s3:PutObject",
    "s3:PutBucketPublicAccessBlock"
  ],
  "Resource": [
    "arn:aws:s3:::aws-athena-query-results-*"
  ]
},
{
  "Sid": "BaseAthenaExamplesPermissions",
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:ListBucket"
  ],
  "Resource": [
    "arn:aws:s3:::athena-examples*"
  ]
},
{
  "Sid": "BaseS3BucketPermissions",
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:GetBucketLocation",
    "s3:ListAllMyBuckets"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Sid": "BaseSNSPermissions",
  "Effect": "Allow",

```

```
    "Action": [
      "sns:ListTopics",
      "sns:GetTopicAttributes"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Sid": "BaseCloudWatchPermissions",
    "Effect": "Allow",
    "Action": [
      "cloudwatch:PutMetricAlarm",
      "cloudwatch:DescribeAlarms",
      "cloudwatch>DeleteAlarms",
      "cloudwatch:GetMetricData"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Sid": "BaseLakeFormationPermissions",
    "Effect": "Allow",
    "Action": [
      "lakeformation:GetDataAccess"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Sid": "BaseDataZonePermissions",
    "Effect": "Allow",
    "Action": [
      "datazone:ListDomains",
      "datazone:ListProjects",
      "datazone:ListAccountEnvironments"
    ],
    "Resource": [
      "*"
    ]
  },
  {
```

```

        "Sid": "BasePricingPermissions",
        "Effect": "Allow",
        "Action": [
            "pricing:GetProducts"
        ],
        "Resource": [
            "*"
        ]
    }
]
}

```

### AWS 受管理的策略：AWSQuicksightAthenaAccess

AWSQuicksightAthenaAccess 授予存取 Amazon 與 Athena 整合所 QuickSight 需的動作。您可將 AWSQuicksightAthenaAccess 政策連接到 IAM 身分。只將此政策附加到 QuickSight 搭配 Athena 使用 Amazon 的主體。此政策包含的 Athena 動作，有些已被取代且未包含在目前的公有 API 中，或是僅用於 JDBC 和 ODBC 驅動程式。

### 許可群組

AWSQuicksightAthenaAccess 政策會分組為以下許可集。

- **athena** – 允許委託人在 Athena 資源上執行查詢。
- **glue**— 允許主參與者存取 AWS Glue 資料庫、表格和分割區。這是必要的，以便主體可以 AWS Glue Data Catalog 與 Athena 一起使用。
- **s3** – 允許委託人從 Simple Storage Service (Amazon S3) 寫入和讀取查詢結果。
- **lakeformation** – 允許委託人要求臨時憑證，以存取位於向 Lake Formation 註冊的資料湖位置中的資料。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[基礎資料存取控制](#)。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:BatchGetQueryExecution",
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",

```

```

        "athena:ListQueryExecutions",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution",
        "athena:ListWorkGroups",
        "athena:ListEngineVersions",
        "athena:GetWorkGroup",
        "athena:GetDataCatalog",
        "athena:GetDatabase",
        "athena:GetTableMetadata",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:ListTableMetadata"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "glue:CreateDatabase",
        "glue>DeleteDatabase",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:UpdateDatabase",
        "glue:CreateTable",
        "glue>DeleteTable",
        "glue:BatchDeleteTable",
        "glue:UpdateTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:BatchCreatePartition",
        "glue:CreatePartition",
        "glue>DeletePartition",
        "glue:BatchDeletePartition",
        "glue:UpdatePartition",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition"
    ],
    "Resource": [
        "*"
    ]
},

```

```

    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket",
        "s3:PutObject",
        "s3:PutBucketPublicAccessBlock"
      ],
      "Resource": [
        "arn:aws:s3:::aws-athena-query-results-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lakeformation:GetDataAccess"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}

```

## Athena AWS 管理政策的更新

檢視 Athena AWS 受管政策更新的詳細資料，因為此服務開始追蹤這些變更。

變更	描述	日期
<a href="#">AmazonAthenaFullAccess</a> – 更新現有政策	讓 Athena 能夠使用公開記錄的 AWS Glue GetCatalogImportStatus API 來擷取目錄匯入狀態。	2024年6月18日
<a href="#">AmazonAthenaFullAccess</a> – 更新現有政策	新增datazone:ListDomains datazone:ListProjects	2024年1月3日

變更	描述	日期
	<p>cts 、和datazone: ListAccountEnvironments 許可，讓 Athena 使用者能夠使用 Amazon DataZone 網域、專案和環境。如需詳細資訊，請參閱 <a href="#">在 Athena 中使用 Amazon DataZone</a>。</p>	
<p><a href="#">AmazonAthenaFullAccess</a> – 更新現有政策</p>	<p>新增glue:StartColumnStatisticsTaskRun glue:GetColumnStatisticsTaskRun 、和glue:GetColumnStatisticsTaskRuns 權限，讓 Athena 有權呼叫以擷取以成本為基礎 AWS Glue 的最佳化程式功能的統計資料。如需詳細資訊，請參閱 <a href="#">使用成本型最佳化工具</a>。</p>	<p>2024 年 1 月 3 日</p>
<p><a href="#">AmazonAthenaFullAccess</a> – 更新現有政策</p>	<p>Athena 已新增 pricing:GetProducts ，以提供對 AWS Billing and Cost Management 的存取。如需詳細資訊，請參閱 AWS Billing and Cost Management API 參考<a href="#">GetProducts</a>中的。</p>	<p>2023 年 1 月 25 日</p>
<p><a href="#">AmazonAthenaFullAccess</a> – 更新現有政策</p>	<p>Athena 新增cloudwatch:GetMetricData 以擷取 CloudWatch 度量值。如需詳細資訊，請參閱 Amazon CloudWatch API 參考資料<a href="#">GetMetricData</a>中的。</p>	<p>2022 年 11 月 14 日</p>

變更	描述	日期
<a href="#">AmazonAthenaFullAccess</a> 和 <a href="#">AWSQuicksightAthenaAccess</a> — 現有政策的更新	Athena 新增了 <code>s3:PutBucketPublicAccessBlock</code> ，以封鎖對 Athena 建立的儲存貯體的公開存取。	2021 年 7 月 7 日
Athena 已開始追蹤變更	Athena 開始追蹤其 AWS 受管理政策的變更。	2021 年 7 月 7 日

## 透過 JDBC 和 ODBC 連接存取

若要取得 Athena AWS 服務 和 Amazon S3 儲存貯體等資源的存取權和資源，請將 JDBC 或 ODBC 驅動程式登入資料提供給您的應用程式。如果您使用的是 JDBC 或 ODBC 驅動程式，請確保 IAM 許可政策包括 [AWS 受管理的策略：AWSQuicksightAthenaAccess](#) 中列出的所有動作。

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

### 身分驗證方法

Athena JDBC 和 ODBC 驅動程式支援 SAML 2.0 型身分驗證，包括以下身分提供者：

- Active Directory Federation Services (AD FS)
- Azure Active Directory (AD)
- Okta
- PingFederate

如需詳細資訊，請參閱各個驅動程式的安裝和組態指南，可從 [JDBC](#) 和 [ODBC](#) 驅動程式頁面下載 PDF 格式的指南。如需其他相關資訊，請參閱以下內容：

- [啟用對 Athena API 的聯合存取](#)
- [使用 Lake Formation 和 Athena JDBC 和 ODBC 驅動程式對 Athena 進行聯合存取](#)
- [使用 ODBC、SAML 2.0 和 Okta 身分提供者設定單一登入](#)

如需最新版 JDBC 和 ODBC 驅動程式及其文件的詳細資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)和[使用 ODBC 連接到 Amazon Athena](#)。

## 從 Athena 訪問 Amazon S3

您可以使用身分型政策、儲存貯體資源政策、存取點政策或上述政策的任何組合，授予對 Amazon S3 位置的存取權。當演員與 Athena 互動時，其許可會傳遞給 Athena，進而確定 Athena 可以存取的內容。這表示使用者必須擁有存取 Amazon S3 儲存貯體的許可，才能向 Athena 查詢這些儲存貯體。

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

請注意，向 Amazon S3 發出的請求來自 Athena 的私有 IPv4 地址，而不是中 `aws:SourceIp` 指定的來源 IP。因此，您無法使用該 `aws:SourceIp` 條件拒絕指定 IAM 政策中對 Amazon S3 動作的存取。您也無法根據或 `aws:SourceVpce` 條件金鑰限制 `aws:SourceVpc` 或允許存取 Amazon S3 資源。

### Note

使用 IAM Identity Center 身分驗證的 Athena 工作群組，需要將 S3 Access Grants 設定為使用受信任的身分傳播身分。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [S3 Access Grants 和目錄身分](#)。

### 主題

- [使用身分型政策控制 Amazon S3 儲存貯體的存取](#)
- [使用儲存貯體資源政策控制 Amazon S3 儲存貯體的存取](#)
- [Amazon S3 存取點、存取點別名和存取點政策](#)
- [使用 CalledVia 上下文鍵](#)
- [其他資源](#)

### 使用身分型政策控制 Amazon S3 儲存貯體的存取

以身分為基礎的政策會連接至 IAM 使用者、群組或角色。這些政策可讓您指定該身分可以執行哪些動作 (其許可)。您可以使用以身分識別為基礎的政策來控制對 Amazon S3 儲存貯體的存取。

下列身分型政策允許 Read 和 Write 存取特定 Amazon S3 儲存貯體中的物件。若要使用此原則，請以您自己的值取代 #####。

```
{
```

```

"Version": "2012-10-17",
"Statement":
  [
    {
      "Sid": "ListObjectsInBucket",
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource":
        ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"]
    },
    {
      "Sid": "AllObjectActions",
      "Effect": "Allow",
      "Action": "s3:*Object",
      "Resource":
        ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"]
    }
  ]
}

```

### 使用儲存貯體資源政策控制 Amazon S3 儲存貯體的存取

您可以使用 Amazon S3 儲存貯體政策來保護儲存貯體中物件的存取，以便只有具有適當許可的使用者才能存取它們。如需建立 Amazon S3 政策的指引，請參閱 Amazon [S3 使用者指南中的使用 Amazon S3 主控台新增儲存貯體政策](#)。

下列範例權限原則會限制使用者讀取具有 `environment: production` 標籤索引鍵和值的物件。範例原則會使用 `s3:ExistingObjectTag` 條件索引鍵來指定標籤鍵和值。

```

{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Principal": {"AWS": "arn:aws:iam::111122223333:role/JohnDoe"},
      "Effect": "Allow",
      "Action": [ "s3:GetObject", "s3:GetObjectVersion" ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
      "Condition":
        {
          "StringEquals": {"s3:ExistingObjectTag/environment": "production"}
        }
    }
  ]
}

```

```
]
}
```

如需更多儲存貯體政策範例，請參閱 [Amazon S3 儲存貯體使用者指南中的 Amazon S3 儲存貯體政策範例](#)。

## Amazon S3 存取點、存取點別名和存取點政策

如果您在 Amazon S3 儲存貯體中具有共用的資料集，則維護負責管理數百個使用案例存取的單一儲存貯體政策可能十分具有挑戰性。

Amazon S3 儲存貯體存取點有助於解決此問題。儲存貯體可有多個存取點，每個存取點都有一個以不同方式控制儲存貯體存取的政策。

Amazon S3 會對您建立的每個存取點產生代表存取點的別名。由於別名採用 Amazon S3 儲存貯體名稱格式，您可以在 Athena 的 CREATE TABLE 陳述式 LOCATION 子句中使用該別名。接著，會由別名所代表的存取點政策來控制 Athena 對儲存貯體的存取。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的 [Amazon S3 中的資料表位置](#) 和 [使用存取點](#)。

## 使用 CalledVia 上下文鍵

為了提升安全性，您可以使用 [aws:CalledVia](#) 全域條件內容索引鍵。aws:CalledVia 索引鍵包含代表主體提出請求的鏈結中，每個服務的排序清單。透過針對 aws:CalledVia 內容索引鍵指定 Athena 服務主體名稱 athena.amazonaws.com，您可以限制僅能有從 Athena 進行的請求。如需詳細資訊，請參閱 [搭配 CalledVia 內容金鑰使用 Athena](#)。

## 其他資源

如需有關如何授予 Amazon S3 存取權的詳細資訊和範例，請參閱以下資源：

- 《Amazon S3 使用者指南》中的 [範例演練：管理存取](#)。
- [如何提供對 Amazon S3 儲存貯體中物件的跨帳戶存取？](#) 在 AWS 知識中心。
- [Athena 中的 Amazon S3 儲存貯體跨帳戶存取](#)。

## Athena 中的 Amazon S3 儲存貯體跨帳戶存取

常見的 Amazon Athena 案例是將存取權授予非儲存貯體擁有者帳戶中的使用者，讓他們可以執行查詢。在這種情況下，使用儲存貯體政策來授予存取權。

**Note**

如需從 Athena 跨帳戶存取資 AWS Glue 料目錄的相關資訊，請參閱[跨帳戶存取 AWS Glue 資料目錄](#)。

以下範例儲存貯體政策由儲存貯體擁有者建立並套用至 s3://DOC-EXAMPLE-BUCKET 儲存貯體，會將存取權授予 123456789123 帳戶 (不同帳戶) 中的所有使用者。

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "MyStatementSid",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789123:root"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ]
    }
  ]
}
```

若要將存取權授予帳戶中的特定使用者，將 Principal 索引鍵取代為指定該使用者而非 root 的索引鍵。例如，針對使用者描述檔 Dave，請使用 arn:aws:iam::123456789123:user/Dave。

### 跨帳戶存取使用自訂 AWS KMS 金鑰加密的儲存貯體

如果您擁有使用自訂 AWS Key Management Service (AWS KMS) 金鑰加密的 Amazon S3 儲存貯體，則可能需要從其他 Amazon Web Services 帳戶將存取權授予使用者。

將帳戶 A 中 AWS KMS 加密儲存貯體的存取權授與帳戶 B 中的使用者需要下列權限：

- 帳戶 A 中的儲存貯體政策必須授與帳戶 B 所擔任的角色的存取權。
- 帳戶 A 中的 AWS KMS 金鑰策略必須授予使用者在帳戶 B 中所扮演角色的存取權。
- 帳戶 B 所承擔的 AWS Identity and Access Management (IAM) 角色必須同時授與存取儲存貯體和帳戶 A 中金鑰的存取權。

下列程序說明如何授與這些許可。

將帳戶 A 中儲存貯體的存取權授與帳戶 B 中的使用者

- 從帳戶 A，[檢閱 S3 儲存貯體政策](#)，並確認有允許透過帳戶 B 的帳戶 ID 存取的陳述式。

例如，下列儲存貯體政策會允許 s3:GetObject 存取帳戶 ID 111122223333：

```
{
  "Id": "ExamplePolicy1",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExampleStmt1",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
      "Principal": {
        "AWS": [
          "111122223333"
        ]
      }
    }
  ]
}
```

從帳戶 a 中的 AWS KMS 金鑰策略授與帳戶 b 中的使用者存取權

1. 在帳戶 A 的 AWS KMS 金鑰策略中，將帳戶 B 所承擔的角色授與下列動作的權限：

- kms:Encrypt

- kms:Decrypt
- kms:ReEncrypt\*
- kms:GenerateDataKey\*
- kms:DescribeKey

下列範例僅會將金鑰存取權授予一個 IAM 角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUseOfTheKey",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<111122223333>:role/role_name"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

2. 在帳戶 A 中，[使用策略檢視檢視金鑰 AWS Management Console 策略](#)。
3. 在金鑰政策中，確認下列陳述式將帳戶 B 列為主體。

```
"Sid": "Allow use of the key"
```

4. 如果 "Sid": "Allow use of the key" 陳述式不存在，請執行下列步驟：
  - a. [使用主控台預設檢視](#) 切換以檢視金鑰政策。
  - b. 將帳戶 B 的帳戶 ID 新增為具有金鑰存取權的外部帳戶。

若要透過帳戶 B 所擔任的 IAM 角色授予帳戶 A 中儲存貯體和金鑰的存取權

1. 從帳戶 B 中，開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在帳戶 B 中開啟與使用者關聯的 IAM 角色。
3. 檢閱套用至 IAM 角色的許可政策清單。
4. 確定已套用會授與儲存貯體存取權的政策。

下列範例陳述式會授予 IAM 角色 DOC-EXAMPLE-BUCKET 儲存貯體上的 s3:GetObject 和 s3:PutObject 操作的存取權：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExampleStmnt2",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    }
  ]
}
```

5. 確定已套用會授予金鑰存取權的政策。

 Note

如果帳戶 B 所擔任的 IAM 角色已經具有[管理員存取權](#)，則您不需要透過使用者 IAM 政策授予金鑰存取權。

下列陳述式範例會授予 IAM 角色使用金鑰 arn:aws:kms:us-west-2:123456789098:key/111aa2bb-333c-4d44-5555-a111bb2c33dd 的存取權。

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

```
"Sid": "ExampleStmt3",
"Action": [
  "kms:Decrypt",
  "kms:DescribeKey",
  "kms:Encrypt",
  "kms:GenerateDataKey",
  "kms:ReEncrypt*"
],
"Effect": "Allow",
"Resource": "arn:aws:kms:us-west-2:123456789098:key/111aa2bb-333c-4d44-5555-
a111bb2c33dd"
}
]
}
```

## 跨帳戶存取儲存貯體物件

由帳戶 (帳戶 C) (而非儲存貯體擁有的帳戶 (帳戶 A)) 上傳的物件，可能需要可授與查詢帳戶 (帳戶 B) 讀取權限的明確物件層級 ACL。為了避免此需求，帳戶 C 應該先負責帳戶 A 中的角色，才能在儲存貯體中放置物件。如需詳細資訊，請參閱[如何提供 Amazon S3 儲存貯體中物件的跨帳戶存取權？](#)。

## 對於中的資料庫和資料表的精細存取 AWS Glue Data Catalog

如果 AWS Glue Data Catalog 搭配 Amazon Athena 使用，您可以為 Athena 中使用的資料庫和表格資料目錄物件定義資源層級政策。

### Note

此處的「精細存取控制」一詞是指數據庫和資料表層級的安全性。如需有關欄、列與儲存格層級安全性的資訊，請參閱[Lake Formation 中的資料篩選和儲存格層級安全性](#)。

您可以定義 IAM 身分型政策中的資源層級許可。

### Important

本節討論 IAM 身分型政策中的資源層級許可。這些不同於資源型政策。如需有關相關差異的詳細資訊，請參閱《IAM 使用者指南》中的[身分型政策和資源型政策](#)。

請參閱以下主題以了解這些任務：

執行此任務	參閱以下主題
建立 IAM 政策來定義資源的精細存取	《IAM 使用者指南》中的 <a href="#">建立 IAM 政策</a> 。
了解中使用的 IAM 身分型政策 AWS Glue	《AWS Glue 開發人員指南》中的 <a href="#">身分型政策 (IAM 政策)</a> 。

## 本節內容

- [限制](#)
- [AWS Glue 存取您的目錄和資料庫 AWS 區域](#)
- [表格分割區和版本 AWS Glue](#)
- [資料表和資料庫的精細分級許可範例](#)

## 限制

透過 AWS Glue Data Catalog 和 Athena 使用精細存取控制時，請考慮下列限制：

- 啟用 IAM Identity Center 的 Athena 工作群組需要將 Lake Formation 設定為使用 IAM Identity Center 身分。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[整合 IAM Identity Center](#)。
- 您可以限制只能存取資料庫和資料表。精細分級的存取控制會套用到資料表層級，您不能限制對資料表中個別分割區的存取。如需詳細資訊，請參閱[表格分割區和版本 AWS Glue](#)。
- 包 AWS Glue Data Catalog 含下列資源：CATALOGDATABASE、TABLE、和FUNCTION。

### Note

從此清單中，說明 Athena 和每個帳號之間常見CATALOG的資源。AWS Glue Data Catalog TABLE DATABASE Function特定於 AWS Glue. 對於 Athena 中的刪除動作，您必須包含 AWS Glue 動作的許可。請參閱[資料表和資料庫的精細分級許可範例](#)。

其階層如下：CATALOG 是每個帳戶中所有 DATABASES 的上階，而每個 DATABASE 是它的所有 TABLES 和 FUNCTIONS 的上階。例如，在帳戶中目錄裡的 db 資料庫中名為 table\_test 的資料表，它的上階是帳戶中的 db 和目錄。對 db 資料庫來說，它的上階是帳戶中的目錄，而它的子代是資料表和函數。如需有關資源階層架構的詳細資訊，請參閱《AWS Glue 開發人員指南》中的[資料目錄中的 ARN 清單](#)。

- 對於資源的任何非刪除 Athena 動作，例如 CREATE DATABASE、CREATE TABLE、SHOW DATABASE、SHOW TABLE 或 ALTER TABLE，您需要許可，才能對資源 (資料表或資料庫) 以及資料目錄中資源的所有上階呼叫此動作。例如，對於表格，其上階為其所屬的資料庫，以及帳戶的目錄。對於資料庫，其上階為帳戶的目錄。請參閱[資料表和資料庫的精細分級許可範例](#)。
- 對於 Athena 中的刪除動作，例如 DROP DATABASE 或 DROP TABLE，您也需要許可，才能對資料目錄中資源的所有上階和下階呼叫刪除動作。例如，若要刪除資料庫，您需要資料庫，目錄 (即其上階)，以及所有表格和使用者定義函數 (即其下階) 的許可。表格沒有下階。若要執行 DROP TABLE，您需要許可，才能對表格、其所屬資料庫，以及目錄執行此動作。請參閱[資料表和資料庫的精細分級許可範例](#)。

## AWS Glue 存取您的目錄和資料庫 AWS 區域

若要使用 Athena AWS Glue，您需要一個原則來授與您的資料庫存取權，以及每個帳戶 AWS 區域的存取權。AWS Glue Data Catalog 若要建立資料庫，還需要 CreateDatabase 許可。在下列範例原則中，將 AWS 區域、AWS 帳戶 ID 和資料庫名稱取代為您自己的名稱。

```
{
  "Sid": "DatabasePermissions",
  "Effect": "Allow",
  "Action": [
    "glue:GetDatabase",
    "glue:GetDatabases",
    "glue:CreateDatabase"
  ],
  "Resource": [
    "arn:aws:glue:us-east-1:123456789012:catalog",
    "arn:aws:glue:us-east-1:123456789012:database/default"
  ]
}
```

## 表格分割區和版本 AWS Glue

在中 AWS Glue，表格可以有分割區和版本。資料表版本和分割區不會被視為中的獨立資源 AWS Glue。授予資料表和資料表上階資源的存取權，便可得到資料表版本和分割區的存取權。

基於精細分級的存取控制之目的，以下存取許可適用：

- 精細分級的存取控制會套用到資料表層級。您可以限制只能存取資料庫和資料表。例如，如果允許存取有分割區的資料表，則此存取權會套用到資料表中的所有分割區。您不能限制存取資料表中個別分割區。

### Important

若要在分割區中 AWS Glue 執行動作，目錄、資料庫和資料表層級需要磁碟分割動作的權限。存取資料表內的分割區並不足夠。例如，若要在資料庫 myDB 中對資料表 myTable 執行 GetPartitions，您必須授予 glue:GetPartitions 目錄、myDB 資料庫和 myTable 資源的許可。

- 精細分級的存取控制不會套用到資料表版本。與分區一樣，對先前版本的表格存取權是透過資料表中 AWS Glue 的表格版本 API 和資料表祖項的存取權授與。

如需有關 AWS Glue 動作權限的詳細資訊，請參閱 AWS Glue 開發人員指南中的 [AWS Glue API 權限：動作和資源參考](#)。

### 資料表和資料庫的精細分級許可範例

下表列出了 IAM 身分型政策範例，這些政策允許對 Athena 中的資料庫和資料表進行精細存取。我們建議您從這些範例開始，並根據您的需要調整它們，來允許或拒絕對特定資料庫和資料表的特定動作。

這些範例包括存取資料庫和目錄，以便 Athena 和 AWS Glue 能夠一起運作。對於多個 AWS 區域，請為每個資料庫和目錄包含類似的政策，每個區域有一行。

在此範例中，將 example\_db 資料庫和 test 資料表名稱取代為您自己的資料庫和資料表名稱。

DDL 陳述式	IAM 存取政策授予資源存取權的範例
ALTER DATABASE	可讓您修改 example_db 資料庫的屬性。 <pre>{   "Effect": "Allow",</pre>

DDL 陳述式	IAM 存取政策授予資源存取權的範例
	<pre>"Action": [   "glue:GetDatabase",   "glue:UpdateDatabase" ], "Resource": [   "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",   "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> " ]</pre>
CREATE DATABASE	<p>可讓您建立名為 <code>example_db</code> 的資料庫。</p> <pre>{   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:CreateDatabase"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> " ]</pre>

DDL 陳述式	IAM 存取政策授予資源存取權的範例
CREATE TABLE	<p>可讓您在 <code>example_db</code> 資料庫中建立名為 <code>test</code> 的資料表。</p> <pre data-bbox="505 296 1507 1528">{   "Sid": "DatabasePermissions",   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetDatabases"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> "   ] }, {   "Sid": "TablePermissions",   "Effect": "Allow",   "Action": [     "glue:GetTables",     "glue:GetTable",     "glue:GetPartitions",     "glue:CreateTable"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_d</i> <i>b</i> /<i>test</i>"   ] }</pre>

DDL 陳述式	IAM 存取政策授予資源存取權的範例
DROP DATABASE	<p>可讓您捨棄 <code>example_db</code> 資料庫，包括其中的所有資料表。</p> <pre data-bbox="505 300 1507 1094">{   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue&gt;DeleteDatabase",     "glue:GetTables",     "glue:GetTable",     "glue&gt;DeleteTable"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database     / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_d</i> <i>b</i> /*",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :userDefi     nedFunction/ <i>example_db</i> /*"   ] }</pre>

DDL 陳述式	IAM 存取政策授予資源存取權的範例
DROP TABLE	<p>可讓您捨棄 <code>example_db</code> 資料庫中名為 <code>test</code> 的分割資料表。如果您的資料表沒有分割區，請勿包含分割動作。</p> <pre data-bbox="505 348 1507 1102">{   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetTable",     "glue&gt;DeleteTable",     "glue:GetPartitions",     "glue:GetPartition",     "glue&gt;DeletePartition"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database     / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_d</i> <i>b</i> /<i>test</i>"   ] }</pre>

DDL 陳述式	IAM 存取政策授予資源存取權的範例
MSCK REPAIR TABLE	<p>允許您在將 Hive 相容分割區新增至 <code>example_db</code> 資料庫中名為 <code>test</code> 的資料表後更新目錄中繼資料。</p> <pre data-bbox="505 348 1507 1102">{   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:CreateDatabase",     "glue:GetTable",     "glue:GetPartitions",     "glue:GetPartition",     "glue:BatchCreatePartition"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database     / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_db</i> /<i>test</i>"   ] }</pre>
SHOW DATABASES	<p>可讓您列出 AWS Glue Data Catalog 中的所有資料庫。</p> <pre data-bbox="505 1213 1507 1690">{   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetDatabases"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database/*"   ] }</pre>

DDL 陳述式	IAM 存取政策授予資源存取權的範例
SHOW TABLES	<p>可讓您列出 <code>example_db</code> 資料庫中的所有資料表。</p> <pre data-bbox="505 300 1507 894"> {   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetTables"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :123456789012 :catalog",     "arn:aws:glue: <i>us-east-1</i> :123456789012 :database / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :123456789012 :table/<i>example_d</i> <i>b</i> /*"   ] } </pre>

## 跨帳戶存取 AWS Glue 資料目錄

您可以使用 Athena 的跨帳戶 AWS Glue 目錄功能，從您自己以外的帳戶註冊 AWS Glue 目錄。設定所需的 IAM 許可 AWS Glue 並將目錄註冊為 Athena [DataCatalog](#) 資源後，您可以使用 Athena 執行跨帳戶查詢。如需使用 Athena 主控台以從其他帳戶註冊目錄的相關資訊，請參閱 [AWS Glue Data Catalog 從另一個帳戶註冊](#)。

如需有關中跨帳戶存取的詳細資訊 AWS Glue，請參閱 AWS Glue 開發人員指南中的 [授予跨帳戶存取權](#)。

### 開始之前

由於此功能會使用現有的 Athena DataCatalog 資源 API 和功能，以啟用跨帳戶存取，因此建議您先閱讀以下資源再開始使用：

- [連線至資料來源](#)-包含使用 Athena 搭配 AWS Glue、蜂巢或 Lambda 資料目錄來源的主題。
- [資料目錄範例政策](#) - 說明如何撰寫控制資料目錄存取的政策。
- [使用 AWS CLI 與蜂巢中繼存儲](#)-顯示如何使用 Hive 中 AWS CLI 繼存放區，但包含適用於其他資料來源的使用案例。

## 考量與限制

目前 Athena 跨帳戶目 AWS Glue 錄存取有下列限制：

- 此功能僅適用於支援 Athena AWS 區域引擎第 2 版或更新版本的地方。如需有關 Athena 引擎版本的資訊，請參閱[Athena 引擎版本控制](#)。若要升級工作群組的引擎版本，請參閱[變更 Athena 引擎版本](#)。
- 當您在帳戶 AWS Glue Data Catalog 中註冊其他帳戶時，您會建立一個區域DataCatalog資源，該資源僅與該特定區域中其他帳戶的資料連結。
- 目前不支援包含跨帳戶 AWS Glue 目錄的 CREATE VIEW 陳述式。
- 使用 AWS 受管理金鑰加密的目錄無法跨帳戶查詢。對於您要跨帳戶查詢的目錄，請改用客戶管理的金鑰 (KMS\_CMK)。如需客戶受管金鑰與 AWS 受控金鑰之間差異的詳細資訊，請參閱AWS Key Management Service 開發人員指南中的[客戶 AWS 金鑰和金鑰](#)。

## 開始使用

在下列案例中，「借款者」帳戶 (666666666666) 會想要執行參照屬於「擁有者」帳戶 (999999999999) 之 AWS Glue 目錄的SELECT查詢，如下列範例所示：

```
SELECT * FROM ownerCatalog.tpch1000.customer
```

在下列程序中，步驟 1a 和 1b 顯示如何讓借款人帳戶存取擁有者帳戶的 AWS Glue 資源，從借款人和擁有者一方。該範例授予對資料庫 tpch1000 和資料表 customer 的存取權。變更這些範例名稱以符合您的需求。

步驟 1a：建立具有存取擁有者資源之 AWS Glue 策略的借款人角色

若要建立具有存取擁有者帳戶 AWS Glue 資源之政策的借用者帳戶角色，您可以使用 AWS Identity and Access Management (IAM) 主控台或 [IAM API](#)。以下為使用 IAM 主控台的程序。

若要建立借款人角色和策略以存取擁有者帳號的 AWS Glue 資源

1. 從借款人帳戶登入 IAM 主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在導覽窗格中，展開存取管理，然後選擇政策。
3. 選擇建立政策。
4. 對於政策編輯器，選擇 JSON。
5. 在策略編輯器中，輸入以下策略，然後根據您的需求進行修改：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "glue:*",
      "Resource": [
        "arn:aws:glue:us-east-1:999999999999:catalog",
        "arn:aws:glue:us-east-1:999999999999:database/tpch1000",
        "arn:aws:glue:us-east-1:999999999999:table/tpch1000/customer"
      ]
    }
  ]
}
```

6. 選擇下一步。
7. 在 [檢閱並建立] 頁面上，對於 [原則名稱]，輸入原則的名稱 (例如，**CrossGluePolicyForBorrowerRole**)。
8. 選擇建立政策。
9. 在導覽窗格中，選擇 Roles (角色)。
10. 選擇 Create Role (建立角色)。
11. 在 [選取信任的實體] 頁面上，選擇 AWS 帳戶，然後選擇 [下一步]。
12. 在 [新增權限] 頁面上，在搜尋方塊中輸入您建立的原則名稱 (例如**CrossGluePolicyForBorrowerRole**)。
13. 選取策略名稱旁的核取方塊，然後選擇 [下一步]。
14. 在 Name, review, and create (名稱，檢閱和建立) 頁面上，針對 Role name (角色名稱) 輸入角色的名稱 (例如 **CrossGlueBorrowerRole**)。
15. 選擇建立角色。

#### 步驟 1b：建立擁有者原則以授予借款人 AWS Glue 存取權

若要 AWS Glue 從擁有者帳戶 (999999999999) 授與借用者角色的存取權，您可以使用 AWS Glue 主控台或 API 作業。AWS Glue [PutResourcePolicy](#) 下列程序會使用主 AWS Glue 控制台。

#### 授予擁有 AWS Glue 者對借款人帳戶的存取權

1. 從擁有者帳戶登入 AWS Glue 主控台 <https://console.aws.amazon.com/glue/>。

2. 在導覽窗格中，展開資料目錄，然後選擇目錄設定。
3. 在 Permissions (許可) 方塊中，輸入如下所示的政策。對於####，請輸入借用者在步驟 1a 中建立的角色 (例如)。**CrossGlueBorrowerRole**如果您想要增加許可範圍，可以將萬用字元 \* 用於資料庫和資料表資源類型。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::666666666666:user/username",
          "arn:aws:iam::666666666666:role/rolename"
        ]
      },
      "Action": "glue:*",
      "Resource": [
        "arn:aws:glue:us-east-1:999999999999:catalog",
        "arn:aws:glue:us-east-1:999999999999:database/tpch1000",
        "arn:aws:glue:us-east-1:999999999999:table/tpch1000/customer"
      ]
    }
  ]
}
```

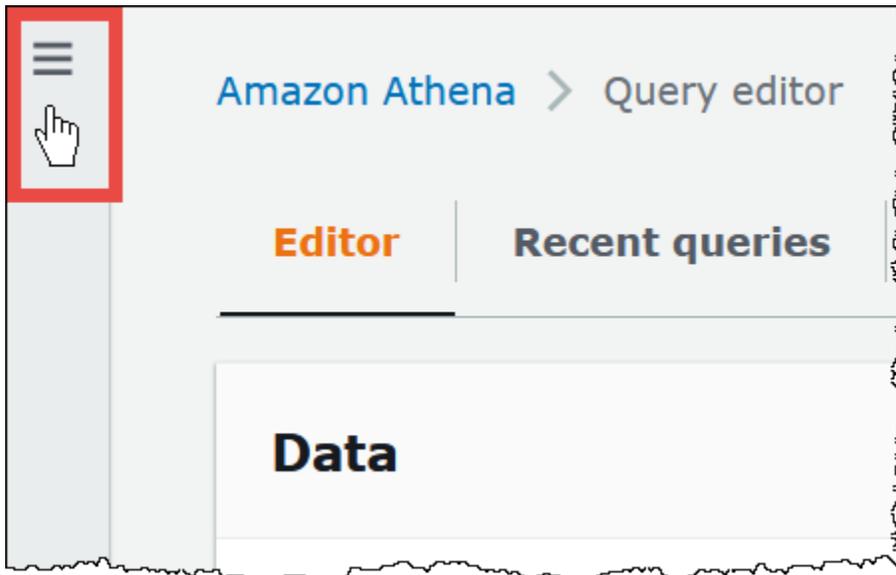
完成後，我們建議您使用 [AWS Glue API](#) 進行一些測試跨帳戶呼叫，以確認權限設定如預期。

## 第 2 步：借款人註冊屬 AWS Glue Data Catalog 於所有者帳戶

下列程序說明如何使用 Athena 主控台，在擁有者 Amazon Web Services 帳戶中設定 AWS Glue Data Catalog 作為資料來源。如需使用 API 操作 (而不是主控台) 註冊型錄的相關資訊，請參閱[使用 API 註冊屬於擁有者帳戶的 Athena 資料型錄](#)。

### 註冊 AWS Glue Data Catalog 屬於另一個帳戶

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 展開管理，然後選擇資料來源。
4. 在右上角，選擇 Create data source (建立資料來源)。
5. 在 [選擇資料來源] 頁面上，針對資料來源選取 S3- AWS Glue Data Catalog，然後選擇 [下一步]。
6. 在 Enter data source details (輸入資料來源詳細資訊) 頁面上的 AWS Glue Data Catalog 區段，針對選擇 AWS Glue Data Catalog，在另一個帳戶選擇 AWS Glue Data Catalog。
7. 在 Data source details (資料來源詳細資訊) 輸入以下資訊：
  - Data source name (資料來源名稱) – 輸入您希望在 SQL 查詢中使用的名稱，以參考其他帳戶中的資料目錄。
  - Description (描述) – (選填) 輸入其他帳戶裡資料目錄的描述。
  - Catalog ID (目錄 ID) – 輸入資料目錄所屬帳戶的 12 位數 Amazon Web Services 帳戶 ID。Amazon Web Services 帳戶 ID 即為目錄 ID。
8. (選用) 展開標籤，然後輸入您希望與資料來源建立關聯的鍵值對。如需標籤的詳細資訊，請參閱[標記 Athena 資源](#)。
9. 選擇下一步。
10. 在 Review and create (檢閱並建立) 頁面上，檢閱您提供的資訊，然後選擇 Create data source (建立資料來源)。Data source details (資料來源詳細資訊) 頁面列出了註冊資料型錄的資料庫和標籤。
11. 選擇 Data Source (資料來源)。您註冊的資料型錄列於 Data Source Name (資料來源名稱) 資料欄。
12. 若要檢視或編輯資料型錄的相關資訊，請選擇型錄，然後選擇 Actions (動作)、Edit (編輯)。
13. 若要刪除新資料型錄，請選擇型錄，然後選擇 Actions (動作)、Delete (刪除)。

### 步驟 3：借用者提交查詢

#####。##語法，如下列範例所示：

```
SELECT * FROM ownerCatalog.tpch1000.customer
```

借用者也可以透過將目錄傳入，而不是使用完整的語法，藉由將目錄傳入。[QueryExecutionContext](#)

#### 其他 Amazon S3 許可

- 如果借用者帳戶使用 Athena 查詢將新資料寫入擁有者帳戶中的資料表，即使該表存在於擁有者帳戶中，擁有者也不會自動存取 Amazon S3 中的此資料。這是因為除非另有設定，否則借款人是 Amazon S3 中資訊的物件擁有者。若要授與擁有者對資料的存取權，請相應地設定物件的權限，做為其他步驟。
- 某些跨帳戶 DDL 操作，如 [MSCK REPAIR TABLE](#) 會需要 Amazon S3 許可。例如，如果借款者帳戶正在針對擁有者帳戶 S3 儲存貯體中具有其資料的擁有者帳戶中的資料表執行跨帳戶 MSCK REPAIR 操作，則該儲存貯體必須授予借款人所假定的角色權限，查詢才能成功。

如需有關授予儲存貯體許可的資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[如何設定 ACL 儲存貯體許可？](#)。

#### 動態使用目錄

在某些情況下，您可能想要在不進行必要步驟註冊的情況下，對跨帳戶 AWS Glue 目錄快速執行測試。如果已如本文件前述，正確設定所需的 IAM 和 Amazon S3 許可，您便可以動態執行跨帳戶查詢，而無需建立 DataCatalog 資源物件。

若要在未註冊的情況下明確參考目錄，請使用以下範例中的語法：

```
SELECT * FROM "glue:arn:aws:glue:us-east-1:999999999999:catalog".tpch1000.customer
```

使用格式「glue:<arn>」，其中 <arn> 為您要使用的 [AWS Glue Data Catalog ARN](#)。在此範例中，Athena 會使用此語法動態指向 999999999999 的 AWS Glue 資料目錄，就如同您已為其個別建立 DataCatalog 物件一樣。

#### 使用動態目錄的注意事項

使用動態目錄時，請記住以下幾點。

- 使用動態目錄會需要您通常用於 Athena 資料目錄 API 操作的 IAM 許可。主要差異在於資料目錄資源名稱會遵循 `glue:*` 命名慣例。
- 目錄 ARN 必須與正在執行的查詢屬於同一區域。
- 在 DML 查詢或檢視中使用動態目錄時，請以逸出的雙引號 (`\`) 括住該目錄。在 DDL 查詢中使用動態目錄時，請使用反引號字元 (```) 括住該目錄。

## 使用 API 註冊屬於擁有者帳戶的 Athena 資料型錄

可以使用 API 操作來註冊屬於擁有者帳戶的資料型錄，而不是使用步驟 2 中所述的 Athena 主控台。

Athena [DataCatalog](#) 資源的建立者必須具備執行 Athena [CreateDataCatalog](#) API 作業的必要權限。依您的需求而定，可能會需要存取其他 API 操作。如需詳細資訊，請參閱 [資料目錄範例政策](#)。

下列 `CreateDataCatalog` 要求主體會註冊跨帳戶存取的 AWS Glue 型錄：

```
# Example CreateDataCatalog request to register a cross-account Glue catalog:
{
  "Description": "Cross-account Glue catalog",
  "Name": "ownerCatalog",
  "Parameters": {"catalog-id" : "999999999999" # Owner's account ID
},
  "Type": "GLUE"
}
```

以下範本程式碼會使用 Java 用戶端來建立 `DataCatalog` 物件。

```
# Sample code to create the DataCatalog through Java client
CreateDataCatalogRequest request = new CreateDataCatalogRequest()
    .withName("ownerCatalog")
    .withType(DataCatalogType.GLUE)
    .withParameters(ImmutableMap.of("catalog-id", "999999999999"));

athenaClient.createDataCatalog(request);
```

完成這些步驟後，借款人應該看到 `ownerCatalog` 何時調用 [ListDataCatalogs](#) API 操作。

## 其他資源

- [AWS Glue Data Catalog 從另一個帳戶註冊](#)

- 在AWS 規範指導模式指南中[AWS Glue Data Catalog 使用 Amazon Athena 設定對共用的跨帳戶存取](#)。
- 在AWS 大數據部落[AWS Glue Data Catalog 格中使用 Amazon Athena 查詢跨帳戶](#)
- 《AWS Glue 開發人員指南》中的[授予跨帳戶存取權](#)

## 從 Athena 存取加密的中繼資料 AWS Glue Data Catalog

如果您 AWS Glue Data Catalog 與 Amazon Athena 搭配使用，則可以 AWS Glue Data Catalog 使用 AWS Glue 控制台或 API 在中啟用加密。如需相關資訊，請參閱《AWS Glue 開發人員指南》中的[加密您的資料目錄](#)。

如果 AWS Glue Data Catalog 已加密，您必須將下列動作新增至用於存取 Athena 的所有原則：

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:GenerateDataKey",
      "kms:Decrypt",
      "kms:Encrypt"
    ],
    "Resource": "(arn of the key used to encrypt the catalog)"
  }
}
```

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 中的安全性最佳實務](#)。

## 存取工作群組和標籤

工作群組是由 Athena 管理的資源。因此，如果您的工作群組政策使用的動作接受 `workgroup` 做為輸入，您必須指定工作群組的 ARN，如下所示，其中 `workgroup-name` 是工作群組的名稱：

```
"Resource": [arn:aws:athena:region:AWSAcctID:workgroup/workgroup-name]
```

例如，在 Amazon Web Services 帳戶 123456789012 的 `us-west-2` 區域中，對於名為 `test_workgroup` 的工作群組，使用以下 ARN 將工作群組指定為資源：

```
"Resource":["arn:aws:athena:us-east-2:123456789012:workgroup/test_workgroup"]
```

若要存取啟用受信任身分傳播 (TIP) 的工作群組，IAM 身分中心使用者必須指派給 Athena [GetWorkGroup](#) API 動作回應所傳回的工作群組。IdentityCenterApplicationArn

- 如需工作群組政策的清單，請參閱 [the section called “工作群組政策範例”](#)。
- 如需工作群組的標籤型政策清單，請參閱 [標籤型 IAM 存取控制政策](#)。
- 如需有關為工作群組建立 IAM 政策的詳細資訊，請參閱 [用於存取工作群組的 IAM 政策](#)。
- 如需 Amazon Athena 動作的完整清單，請參閱 [《Amazon Athena API 參考》](#) 中的 API 動作名稱。
- 如需有關 IAM 政策的詳細資訊，請參閱 [《IAM 使用者指南》](#) 中的 [使用視覺化編輯器來建立政策](#)。

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱 [《IAM 使用者指南》](#) 中的 [IAM 中的安全性最佳實務](#)。

## 允許存取預備陳述式

本主題涵蓋中已準備好陳述式的 IAM 許可 Amazon Athena。每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱 [《IAM 使用者指南》](#) 中的 [IAM 中的安全性最佳實務](#)。

如需有關預備陳述式的詳細資訊，請參閱 [使用參數化查詢](#)。

建立、管理和執行預備陳述式時，需要下列 IAM 許可。

```
athena:CreatePreparedStatement
athena:UpdatePreparedStatement
athena:GetPreparedStatement
athena:ListPreparedStatements
athena>DeletePreparedStatement
```

如以下資料表所示方式使用這些許可。

若要執行此作業	使用這些許可
執行 PREPARE 查詢	athena:StartQueryExecution    athena:CreatePreparedStatement
重新執行 PREPARE 查詢以更新現有的預備陳述式	athena:StartQueryExecution    athena:UpdatePreparedStatement

若要執行此作業	使用這些許可
執行 EXECUTE 查詢	athena:StartQueryExecution athena:GetPreparedStatement
執行 DEALLOCATE PREPARE 查詢	athena:StartQueryExecution athena>DeletePreparedStatement

## 範例

以下 IAM 政策範例授予許可，以在指定的帳戶 ID 和工作群組上管理和執行預備陳述式。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:StartQueryExecution",
        "athena:CreatePreparedStatement",
        "athena:UpdatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena>DeletePreparedStatement",
        "athena:ListPreparedStatements"
      ],
      "Resource": [
        "arn:aws:athena:*:111122223333:workgroup/<workgroup-name>"
      ]
    }
  ]
}
```

## 搭配 CalledVia 內容金鑰使用 Athena

當主體向其提出[要求](#)時 AWS，會將請求資訊 AWS 收集到要求前後關聯中，以評估並授權請求。您可以使用 JSON 政策的 Condition 元素，來比較請求內容中的鍵和您在政策中指定的鍵值。全域條件內容索引鍵為帶有 aws: 字首的條件索引鍵。

## aws:CalledVia 內容索引鍵

您可以使用 [aws:CalledVia](#) 全域條件內容索引鍵，以將政策中的服務與代表 IAM 主體 (使用者或角色) 提出請求的服務進行比較。當主體向某個服務發出要求時 AWS 服務，該服務可能會使用主體的認證向其他服務發出後續要求。aws:CalledVia 鍵包含代表主體提出請求的鏈結中，每個服務的排序清單。

藉由指定aws:CalledVia前後關聯索引鍵的服務主要名稱，您可以將前後關聯索引鍵設為 AWS 服務特定的。例如，您可以使用 aws:CalledVia 條件索引鍵來限制僅能有從 Athena 進行的請求。若要搭配 Athena 使用政策中的 aws:CalledVia 條件索引鍵，您需指定 Athena 服務主體名稱 athena.amazonaws.com，如下列範例所示。

```
...
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:CalledVia": "athena.amazonaws.com"
    }
  }
...

```

您可以使用 aws:CalledVia 內容索引鍵，以確保如果呼叫者從 Athena 呼叫資源，則其僅能存取一項資源 (如 Lambda 函數)。

### Note

aws:CalledVia 內容索引鍵與受信任的身分傳播功能不相容。

新增選用的 CalledVia 內容金鑰，以精細存取 Lambda 函數

Athena 要求呼叫者需具有 lambda:InvokeFunction 許可，以呼叫與查詢相關聯的 Lambda 函數。以下陳述式允許對 Lambda 函數進行精細存取，讓使用者僅能使用 Athena 呼叫 Lambda 函數。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor3",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",

```

```

    "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:OneAthenaLambdaFunction",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "athena.amazonaws.com"
      }
    }
  }
]
}

```

以下範例會說明如何在政策中加入先前的陳述式，以允許使用者執行並讀取聯合查詢。獲准執行這些動作的主體在其執行的查詢中，可以指定與聯合資料來源相關聯的 Athena 目錄。不過，除非透過 Athena 呼叫函數，否則主體無法存取相關聯的 Lambda 函數。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "s3:PutObject",
        "s3:GetObject",
        "athena:StartQueryExecution",
        "s3:AbortMultipartUpload",
        "athena:StopQueryExecution",
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:athena:*:111122223333:workgroup/WorkGroupName",
        "arn:aws:s3:::MyQueryResultsBucket/*",
        "arn:aws:s3:::MyLambdaSpillBucket/MyLambdaSpillPrefix*"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "athena:ListWorkGroups",
      "Resource": "*"
    }
  ]
}

```

```
    },
    {
      "Sid": "VisualEditor2",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::MyLambdaSpillBucket"
    },
    {
      "Sid": "VisualEditor3",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:*:111122223333:function:OneAthenaLambdaFunction",
        "arn:aws:lambda:*:111122223333:function:AnotherAthenaLambdaFunction"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": "athena.amazonaws.com"
        }
      }
    }
  ]
}
```

如需有關 CalledVia 條件索引鍵的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容索引鍵](#)。

## 允許存取外部 Hive 中繼存放區的 Athena 資料連接器

本主題中的許可政策範例示範需要允許的動作，以及允許這些動作的資源。將類似的許可政策連接至 IAM 身分之前，請仔細檢查這些政策，並根據您的需求進行修改。

- [Example Policy to Allow an IAM Principal to Query Data Using Athena Data Connector for External Hive Metastore](#)
- [Example Policy to Allow an IAM Principal to Create an Athena Data Connector for External Hive Metastore](#)

## Example - 允許 IAM 主體使用外部 Hive 中繼存放區的 Athena 資料連接器來查詢資料

除了授權完整存取 Athena 動作的 [AWS 受管理的策略：AmazonAthenaFullAccess](#)，下列政策也會連接至 IAM 委託人。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetLayerVersion",
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:*:111122223333:function:MyAthenaLambdaFunction",
        "arn:aws:lambda:*:111122223333:function:AnotherAthenaLambdaFunction",
        "arn:aws:lambda:*:111122223333:layer:MyAthenaLambdaLayer:*"
      ]
    },
    {
      "Sid": "VisualEditor2",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
      "Resource": "arn:aws:s3:::MyLambdaSpillBucket/MyLambdaSpillLocation"
    }
  ]
}
```

## 許可說明

允許的動作	說明
<pre>"s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket", "s3:PutObject", "s3:ListMultipartUploadParts", "s3:AbortMultipartUpload"</pre>	<p>s3動作允許讀取和寫入指定為的資源 <code>arn:aws:s3::: <i>MyLambdaSpillBucket</i> /<i>MyLambdaSpillLocation</i> </code>，其中 <code><i>MyLambdaSpillLocation</i></code> 標識在正在調用的 Lambda 函數或函數的配置中指定的溢出存儲桶。只有當您使用 <code>Lambda ##### arn:aws:lambda: <i>MyAthenaLambdaLayer</i> *: <i>my AWSacctId</i> : <i>Layer</i>:: * ##</code> 識別碼。最後面的 * 是層版本的萬用字元。</p>
<pre>"lambda:GetFunction", "lambda:GetLayerVersion", "lambda:InvokeFunction"</pre>	<p>允許查詢調用Resource塊中指定的AWS Lambda 函數。例如 <code>arn:aws:lambda:*: <i>MyAWSacctId</i> :function: <i>MyAthenaLambdaFunction</i></code>，其中 <code><i>MyAthenaLambdaFunction</i></code> 指定要調用的 Lambda 函數的名稱。如範例所示，可以指定多個函數。</p>

Example - 允許 IAM 主體建立外部 Hive 中繼存放區的 Athena 資料連接器

除了授權完整存取 Athena 動作的 [AWS 受管理的策略：AmazonAthenaFullAccess](#)，下列政策也會連接至 IAM 委託人。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:ListFunctions",
        "lambda:GetLayerVersion",
        "lambda:InvokeFunction",
```

```

        "lambda:CreateFunction",
        "lambda>DeleteFunction",
        "lambda:PublishLayerVersion",
        "lambda>DeleteLayerVersion",
        "lambda:UpdateFunctionConfiguration",
        "lambda:PutFunctionConcurrency",
        "lambda>DeleteFunctionConcurrency"
    ],
    "Resource": "arn:aws:lambda:*:111122223333:
function: MyAthenaLambdaFunctionsPrefix*"
    }
]
}

```

## 許可說明

允許查詢調用 AWS Lambda Resource 塊中指定的 AWS Lambda 函數的功能。例如 `arn:aws:lambda:*:MyAWSAcctId:function:MyAthenaLambdaFunction`，其中 `MyAthenaLambdaFunction` 指定要調用的 Lambda 函數的名稱。如範例所示，可以指定多個函數。

## 允許 Lambda 函數存取外部 Hive 中繼存放區

若要在帳戶中呼叫 Lambda 函數，您必須建立具有下列許可的角色：

- `AWSLambdaVPCLambdaAccessExecutionRole` - [AWS Lambda 執行角色](#) 許可，用於管理將您的函數連接到 VPC 的彈性網路介面。請確定您有足夠數量的網路界面和 IP 地址可用。
- `AmazonAthenaFullAccess`— [AmazonAthenaFullAccess](#) 受管政策授予對 Athena 的完整存取權限。
- 允許 Lambda 函數寫入 S3 並允許 Athena 從 S3 讀取的 Amazon S3 政策。

例如，下列政策定義溢出位置 `s3://mybucket/spill` 的許可。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",

```

```

        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/spill"
    ]
}
]
}

```

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

## 建立 Lambda 函數

若要在您的帳戶中建立 Lambda 函數，需具備函數開發許可或 AWSLambdaFullAccess 角色。如需詳細資訊，請參閱 [AWS Lambda 身分型 IAM 政策](#)。

由於 Athena 使用建立 Lambda 函數，因此建立 Lambda 函數的超級使用者或管理員也應具有 IAM 政策，[以允許 Athena 聯合查詢](#)。AWS Serverless Application Repository

## 目錄註冊和中繼資料 API 操作

若要存取目錄註冊 API 和中繼資料 API 作業，請使用 [AmazonAthenaFullAccess 受管政策](#)。如果您不使用此政策，請將下列 API 操作新增至您的 Athena 政策：

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListDataCatalogs",
        "athena:GetDataCatalog",
        "athena:CreateDataCatalog",
        "athena:UpdateDataCatalog",
        "athena>DeleteDataCatalog",
        "athena:GetDatabase",
        "athena:ListDatabases",
        "athena:GetTableMetadata",
        "athena:ListTableMetadata"
      ]
    }
  ]
}

```

```
    ],
    "Resource": [
      "*"
    ]
  }
]
```

## 跨區域 Lambda 叫用

若要在執行 Athena 查詢的區域以外的區域叫用 Lambda 函數，請使用 Lambda 函數的完整 ARN。依預設，Athena 會叫用在相同區域中定義的 Lambda 函數。如果您需要在執行 Athena 查詢的區域以外的區域中叫用 Lambda 函數來存取 Hive 中繼存放區，則必須提供 Lambda 函數的完整 ARN。

例如，假設您在歐洲 (法蘭克福) 區域 ehms 上定義目錄 eu-central-1，以便在美國東部 (維吉尼亞北部) 區域中使用下列 Lambda 函數。

```
arn:aws:lambda:us-east-1:111122223333:function:external-hms-service-new
```

當您以這種方式指定完整的 ARN 時，Athena 可以在 us-east-1 呼叫 external-hms-service-new Lambda 函數，從 eu-central-1 擷取 Hive 中繼存放區資料。

### Note

目錄 ehms 應該註冊在您執行 Athena 查詢的相同區域中。

## 跨帳戶 Lambda 叫用

有時您可能需要從不同的帳戶存取 Hive 中繼存放區。例如，若要執行 Hive 中繼存放區，您可以從與用於 Athena 查詢的帳戶不同的帳戶啟動 EMR 叢集。不同的群組或團隊可能會在其 VPC 內使用不同的帳戶來執行 Hive 中繼存放區。或者，您可能想要存取來自不同群組或團隊的不同 Hive 中繼存放區的中繼資料。

Athena 會使用 [AWS Lambda 支援進行跨帳戶存取](#)，以啟用 Hive 中繼存放區的跨帳戶存取。

### Note

請注意，Athena 的跨帳戶存取通常意味著可跨帳戶存取 Amazon S3 中的中繼資料和資料。

想像下列案例：

- 帳戶 111122223333 在 Athena 中的 us-east-1 上設定 Lambda 函數 external-hms-service-new，以存取 EMR 叢集上執行的 Hive 中繼存放區。
- 帳戶 111122223333 想要允許帳戶 444455556666 存取 Hive 中繼存放區資料。

若要授與帳戶對 Lambda 函數的 444455556666 存取權 external-hms-service-new，帳戶請 111122223333 使用下列 AWS CLI add-permission 命令。為了方便閱讀，命令已經過格式化處理。

```
$ aws --profile perf-test lambda add-permission
  --function-name external-hms-service-new
  --region us-east-1
  --statement-id Id-ehms-invocation2
  --action "lambda:InvokeFunction"
  --principal arn:aws:iam::444455556666:user/perf1-test
{
  "Statement": [{"Sid": "Id-ehms-invocation2",
    "Effect": "Allow",
    "Principal": {"AWS": "arn:aws:iam::444455556666:user/perf1-test"}},
    {"Action": "lambda:InvokeFunction",
    "Resource": "arn:aws:lambda:us-east-1:111122223333:function:external-hms-service-new"}]
}
```

若要檢查 Lambda 許可，請使用 get-policy 命令，如下列範例所示。為了方便閱讀，命令已經過格式化處理。

```
$ aws --profile perf-test lambda get-policy
  --function-name arn:aws:lambda:us-east-1:111122223333:function:external-hms-service-new
  --region us-east-1
{
  "RevisionId": "711e93ea-9851-44c8-a09f-5f2a2829d40f",
  "Policy": [{"Version": "2012-10-17",
    "Id": "default",
    "Statement": [{"Sid": "Id-ehms-invocation2",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::444455556666:user/perf1-test"}},
    ]}
}
```

```
        \"Action\": \"lambda:InvokeFunction\",
        \"Resource\": \"arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-new\"}]}"
    }
```

新增許可後，您可以在定義目錄 ehms 時，在 us-east-1 上使用 Lambda 函數的完整 ARN，如下所示：

```
arn:aws:lambda:us-east-1:111122223333:function:external-hms-service-new
```

如需跨區域呼叫的相關資訊，請參閱本主題稍早的[跨區域 Lambda 叫用](#)。

### 授予資料的跨帳戶存取

在執行 Athena 查詢之前，您必須授予對 Amazon S3 中資料的跨帳戶存取。您可採用下列其中一種方式來這麼做：

- 使用[正式使用者 ID](#) 來更新 Amazon S3 儲存貯體的存取控制清單政策。
- 將跨帳戶存取新增至 Amazon S3 儲存貯體政策。

例如，將下列政策新增至帳戶 111122223333 中的 Amazon S3 儲存貯體政策，以允許帳戶 444455556666 從指定的 Amazon S3 位置讀取資料。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567890123",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:user/perf1-test"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::athena-test/lambda/dataset/*"
    }
  ]
}
```

**Note**

您可能不只需要將資料的跨帳戶存取授予 Amazon S3，也需要授予 Amazon S3 溢出位置的跨帳戶存取。當回應物件的大小超過指定閾值時，您的 Lambda 函數會將額外的資料溢出到溢出位置。請參閱本主題開頭的範例政策。

在目前的範例中，將跨帳戶存取授予 444455556666，後，444455556666 可以在自己的 ehms 中使用目錄 account 來查詢帳戶 111122223333 中定義的資料表。

在下列範例中，SQL Workbench 設定檔 perf-test-1 適用於帳戶 444455556666。查詢使用目錄 ehms 來存取 Hive 中繼存放區和帳戶 111122223333 中的 Amazon S3 資料。

The screenshot shows a SQL query in a Workbench interface: `select * from ehms.hms_test_tpch_partitioned.customer limit 100;` The result set contains columns: c\_custkey, c\_name, c\_address, c\_phone, c\_acctbal, c\_mktsegment, and c\_comment. The data includes customer IDs, names, addresses, phone numbers, account balances, market segments, and comments.

c_custkey	c_name	c_address	c_phone	c_acctbal	c_mktsegment	c_comment
375875	Customer#000375875	JvO3Pzge8jZSnokjxEAc7rAVN8IKURVULuRQqRX	16-804-877-2149	7922.59	FURNITURE	final instructions. stealthily regular
375885	Customer#000375885	uNPTa1PIjgEHQoSSoDB	16-255-433-4448	1901.27	AUTOMOBILE	along the blithely bold accounts integrate b
375897	Customer#000375897	vFSYPgsnNPjwLqZkHOSFhnbUCut	16-287-340-3995	3025.81	BUILDING	cording to the quickly even instructio
375904	Customer#000375904	D0okLSAd8MyAO zjuuzmzz2NVYSSPkAFvuc	16-760-202-2511	5746.41	AUTOMOBILE	ees hang unusual accounts. slyly final platelets use slyly. final instructions
375927	Customer#000375927	AaGZcThAlue5THzvAXW	16-913-616-6119	9898.89	HOUSEHOLD	gside of the special, special packages. stealthy th
375936	Customer#000375936	b3bBknxFvP74snCKnV	16-886-740-8768	7741.15	FURNITURE	regular dependencies detect furiously about the blithe

## 允許 Athena 聯合查詢的 IAM 許可政策範例

本主題中的許可政策範例示範需要允許的動作，以及允許這些動作的資源。將這些政策連接至 IAM 身分之前，請仔細檢查政策，並根據您的需求進行修改。

如需有關將政策連接至 IAM 身分的資訊，請參閱 [《IAM 使用者指南》](#) 中的 [新增和移除 IAM 身分許可](#)。

- [Example policy to allow an IAM principal to run and return results using Athena Federated Query](#)
- [Example Policy to Allow an IAM Principal to Create a Data Source Connector](#)

### Example - 允許 IAM 主體使用 Athena 聯合查詢執行並傳回結果

在下列以身分識別為基礎的許可政策中，允許使用者或其他 IAM 委託人需要的動作，有這些動作才能使用 Athena 聯合查詢。獲准執行這些動作的委託人在其執行的查詢中，可以指定與聯合資料來源相關聯的 Athena 目錄。

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

```

{
  "Sid": "Athena",
  "Effect": "Allow",
  "Action": [
    "athena:GetDataCatalog",
    "athena:GetQueryExecution",
    "athena:GetQueryResults",
    "athena:GetWorkGroup",
    "athena:StartQueryExecution",
    "athena:StopQueryExecution"
  ],
  "Resource": [
    "arn:aws:athena:*:111122223333:workgroup/WorkgroupName",
    "arn:aws:athena:aws_region:111122223333:datacatalog/DataCatalogName"
  ]
},
{
  "Sid": "ListAthenaWorkGroups",
  "Effect": "Allow",
  "Action": "athena:ListWorkGroups",
  "Resource": "*"
},
{
  "Sid": "Lambda",
  "Effect": "Allow",
  "Action": "lambda:InvokeFunction",
  "Resource": [
    "arn:aws:lambda:*:111122223333:function:OneAthenaLambdaFunction",
    "arn:aws:lambda:*:111122223333:function:AnotherAthenaLambdaFunction"
  ]
},
{
  "Sid": "S3",
  "Effect": "Allow",
  "Action": [
    "s3:AbortMultipartUpload",
    "s3:GetBucketLocation",
    "s3:GetObject",
    "s3:ListBucket",
    "s3:ListMultipartUploadParts",
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::MyLambdaSpillBucket",

```

```

        "arn:aws:s3:::MyLambdaSpillBucket/*",
        "arn:aws:s3:::MyQueryResultsBucket",
        "arn:aws:s3:::MyQueryResultsBucket/*"
    ]
}

```

## 許可說明

允許的動作	說明
<pre> "athena:GetQueryExecution", "athena:GetQueryResults", "athena:GetWorkGroup", "athena:StartQueryExecution", "athena:StopQueryExecution" </pre>	執行聯合查詢所需的 Athena 許可。
<pre> "athena:GetDataCatalog", "athena:GetQueryExecution", "athena:GetQueryResults", "athena:GetWorkGroup", "athena:StartQueryExecution", "athena:StopQueryExecution" </pre>	執行聯合檢視查詢所需的 Athena 許可。檢視需要 <code>GetDataCatalog</code> 動作。
<pre> "lambda:InvokeFunction" </pre>	允許查詢調用 AWS Lambda Resource 塊中指定的 AWS Lambda 函數的功能。例如 <code>arn:aws:lambda:*:MyAWSacctId:function:MyAthenaLambdaFunction</code> ，其中 <code>MyAthenaLambdaFunction</code> 指定要調用的 Lambda 函數的名稱。如範例所示，可以指定多個函數。
<pre> "s3:AbortMultipartUpload", "s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket", "s3:ListMultipartUploadParts", "s3:PutObject" </pre>	存取執行 <code>StartQueryExecution</code> 行之 IAM 主體的查詢輸出值區需要 <code>s3:ListBucket</code> 和 <code>s3:GetBucketLocation</code> 許可。  <code>s3:PutObject</code> 、 <code>s3:ListMultipartUploadParts</code> ，並 <code>s3:AbortM</code>

允許的動作	說明
	<p>multipartUpload 允許根據arn:aws:s3::: <i>MyQueryResultsBucket</i> /*資源識別碼指定，將查詢結果寫入查詢結果值區的所有子資料夾，其中<i>MyQueryResultsBucket</i> 是 Athena 查詢結果值區。如需詳細資訊，請參閱 <a href="#">使用查詢結果、近期查詢和輸出檔案</a>。</p> <p>s3:GetObject 允許讀取指定為的資源的查詢結果和查詢歷史記錄arn:aws:s3::: <i>MyQueryResultsBucket</i> ，其中<i>MyQueryResultsBucket</i> 是 Athena 查詢結果值區。</p> <p>s3:GetObject 也允許從指定為的資源讀取"arn:aws:s3::: <i>MyLambdaSpillBucket</i> /<i>MyLambdaSpillPrefix</i> *"，其中指定<i>MyLambdaSpillPrefix</i> 於所叫用的 Lambda 函數或函數的組態中。</p>

### Example - 允許 IAM 主體建立資料來源連接器

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:ListVersionsByFunction",
        "iam:CreateRole",
        "lambda:GetFunctionConfiguration",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "lambda:PutFunctionConcurrency",
        "iam:PassRole",
        "iam:DetachRolePolicy",
        "lambda:ListTags",

```

```

        "iam:ListAttachedRolePolicies",
        "iam>DeleteRolePolicy",
        "lambda>DeleteFunction",
        "lambda:GetAlias",
        "iam:ListRolePolicies",
        "iam:GetRole",
        "iam:GetPolicy",
        "lambda:InvokeFunction",
        "lambda:GetFunction",
        "lambda:ListAliases",
        "lambda:UpdateFunctionConfiguration",
        "iam>DeleteRole",
        "lambda:UpdateFunctionCode",
        "s3:GetObject",
        "lambda:AddPermission",
        "iam:UpdateRole",
        "lambda>DeleteFunctionConcurrency",
        "lambda:RemovePermission",
        "iam:GetRolePolicy",
        "lambda:GetPolicy"
    ],
    "Resource": [
        "arn:aws:lambda:*:111122223333:function:MyAthenaLambdaFunctionsPrefix",
        "arn:aws:s3:::awsserverlessrepo-changesets-1iiv3xa62ln3m/*",
        "arn:aws:iam::*:role/RoleName",
        "arn:aws:iam::*:111122223333:policy/*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateUploadBucket",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:ListExports",
        "cloudformation:ListStacks",
        "cloudformation:ListImports",
        "lambda:ListFunctions",
        "iam:ListRoles",
        "lambda:GetAccountSettings",
        "ec2:DescribeSecurityGroups",
        "cloudformation:EstimateTemplateCost",
        "ec2:DescribeVpcs",

```

```

        "lambda:ListEventSourceMappings",
        "cloudformation:DescribeAccountLimits",
        "ec2:DescribeSubnets",
        "cloudformation>CreateStackSet",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
},
{
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": "cloudformation:*",
    "Resource": [
        "arn:aws:cloudformation:*:111122223333:stack/aws-serverless-
repository-MyCFStackPrefix/*",
        "arn:aws:cloudformation:*:111122223333:stack/
serverlessrepo-MyCFStackPrefix/*",
        "arn:aws:cloudformation:*:*:transform/Serverless-*",
        "arn:aws:cloudformation:*:111122223333:stackset/aws-serverless-
repository-MyCFStackPrefix*:*",
        "arn:aws:cloudformation:*:111122223333:stackset/
serverlessrepo-MyCFStackPrefix*:*"
    ]
},
{
    "Sid": "VisualEditor3",
    "Effect": "Allow",
    "Action": "serverlessrepo:*",
    "Resource": "arn:aws:serverlessrepo:*:*:applications/*"
}
]
}

```

## 許可說明

允許的動作	說明
<pre> "lambda&gt;CreateFunction", "lambda:ListVersionsByFunction", "lambda:GetFunctionConfiguration", "lambda:PutFunctionConcurrency", "lambda:ListTags", "lambda&gt;DeleteFunction", </pre>	<p>允許建立和管理列為資源的 Lambda 函數。在此範例中，資源識別碼會使用名稱前置詞 <code>arn:aws:lambda:*: <i>MyAWSAccount</i> :function: <i>MyAthenaLambdaFunctionsPrefix</i> *</code>，其中 <i>MyAthenaL</i></p>

允許的動作	說明
<pre>"lambda:GetAlias", "lambda:InvokeFunction", "lambda:GetFunction", "lambda:ListAliases", "lambda:UpdateFunctionConfigur ation", "lambda:UpdateFunctionCode", "lambda:AddPermission", "lambda&gt;DeleteFunctionConcurrency", "lambda:RemovePermission", "lambda:GetPolicy" "lambda:GetAccountSettings", "lambda:ListFunctions", "lambda:ListEventSourceMappings",</pre>	<p><i>lambdaFunctionsPrefix</i> 是在 Lambda 函數群組名稱中使用的共用前置詞，因此不需要將它們個別指定為資源。您可以指定一或多個 Lambda 函數資源。</p>
<pre>"s3:GetObject"</pre>	<p>允許讀取資源識別碼所指定 AWS Serverless Application Repository 需要的值區 <code>arn:aws:s3:::awsserverlessrepo-changesets-<i>1iiv3xa62ln3m</i>/*</code>。此儲存貯體可能專屬於您的帳戶。</p>
<pre>"cloudformation:*"</pre>	<p>允許建立和管理資源 <i>myCF StackPrefix</i> 指定的 AWS CloudFormation 堆疊。這些堆疊和堆疊集是 AWS Serverless Application Repository 部署連接器和 UDF 的方式。</p>
<pre>"serverlessrepo:*"</pre>	<p>允許在中搜尋、檢視、發行和更新資源識別碼所指定的應用程式 <code>arn:aws:serverlessrepo:*:*:applications/*</code>。AWS Serverless Application Repository</p>

## 允許 Amazon Athena 使用者定義函數 (UDF) 的 IAM 許可政策範例

本主題中的許可政策範例示範需要允許的動作，以及允許這些動作的資源。將類似的許可政策連接至 IAM 身分之前，請仔細檢查這些政策，並根據您的需求進行修改。

- [Example Policy to Allow an IAM Principal to Run and Return Queries that Contain an Athena UDF Statement](#)
- [Example Policy to Allow an IAM Principal to Create an Athena UDF](#)

Example - 允許 IAM 主體執行並傳回包含 Athena UDF 陳述式的查詢

在下列以身分識別為基礎的許可政策中，允許使用者或其他 IAM 委託人需要的動作，有這些動作才能執行使用 Athena UDF 陳述式的查詢。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "athena:StartQueryExecution",
        "lambda:InvokeFunction",
        "athena:GetQueryResults",
        "s3:ListMultipartUploadParts",
        "athena:GetWorkGroup",
        "s3:PutObject",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "athena:StopQueryExecution",
        "athena:GetQueryExecution",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:athena:*:MyAWSAcctId:workgroup/MyAthenaWorkGroup",
        "arn:aws:s3:::MyQueryResultsBucket/*",
        "arn:aws:lambda:*:MyAWSAcctId:function:OneAthenaLambdaFunction",
        "arn:aws:lambda:*:MyAWSAcctId:function:AnotherAthenaLambdaFunction"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "athena:ListWorkGroups",
      "Resource": "*"
    }
  ]
}
```

}

## 許可說明

允許的動作	說明
<pre>"athena:StartQueryExecution", "athena:GetQueryResults", "athena:GetWorkGroup", "athena:StopQueryExecution", "athena:GetQueryExecution",</pre>	<p>在 MyAthenaWorkGroup 工作群組中執行查詢所需的 Athena 許可。</p>
<pre>"s3:PutObject", "s3:GetObject", "s3:AbortMultipartUpload"</pre>	<p>s3:PutObject 並 s3:AbortMultipartUpload 允許將查詢結果寫入查詢結果值區的所有子資料夾，如 <code>arn:aws:s3::: <i>MyQueryResultsBucket</i> /*</code> 資源識別碼所指定，其中 <i>MyQueryResultsBucket</i> 是 Athena 查詢結果值區。如需詳細資訊，請參閱 <a href="#">使用查詢結果、近期查詢和輸出檔案</a>。</p> <p>s3:GetObject 允許讀取指定為的資源的查詢結果和查詢歷史記錄 <code>arn:aws:s3::: <i>MyQueryResultsBucket</i></code>，其中 <i>MyQueryResultsBucket</i> 是 Athena 查詢結果值區。如需詳細資訊，請參閱 <a href="#">使用查詢結果、近期查詢和輸出檔案</a>。</p> <p>s3:GetObject 也允許從指定為的資源讀取 <code>"arn:aws:s3::: <i>MyLambdaSpillBucket</i> /<i>MyLambdaSpillPrefix</i> *"</code>，其 <i>MyLambdaSpillPrefix</i> 中在所叫用的 Lambda 函數或函數的組態中指定。</p>
<pre>"lambda:InvokeFunction"</pre>	<p>允許查詢調用 Resource 塊中指定的 AWS Lambda 函數。例如 <code>arn:aws:lambda:*: <i>MyAWSAcctId</i> :function: <i>MyAthenaLambdaFunction</i></code>，其中 <i>MyAthenaLambdaFunction</i> 指定要調用</p>

允許的動作	說明
	的 Lambda 函數的名稱。如範例所示，可以指定多個函數。

### Example - 允許 IAM 主體建立 Athena UDF

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:ListVersionsByFunction",
        "iam:CreateRole",
        "lambda:GetFunctionConfiguration",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "lambda:PutFunctionConcurrency",
        "iam:PassRole",
        "iam:DetachRolePolicy",
        "lambda:ListTags",
        "iam:ListAttachedRolePolicies",
        "iam>DeleteRolePolicy",
        "lambda>DeleteFunction",
        "lambda:GetAlias",
        "iam:ListRolePolicies",
        "iam:GetRole",
        "iam:GetPolicy",
        "lambda:InvokeFunction",
        "lambda:GetFunction",
        "lambda:ListAliases",
        "lambda:UpdateFunctionConfiguration",
        "iam>DeleteRole",
        "lambda:UpdateFunctionCode",
        "s3:GetObject",
        "lambda:AddPermission",
        "iam:UpdateRole",
        "lambda>DeleteFunctionConcurrency",
        "lambda:RemovePermission",
```

```

        "iam:GetRolePolicy",
        "lambda:GetPolicy"
    ],
    "Resource": [
        "arn:aws:lambda:*:111122223333:function:MyAthenaLambdaFunctionsPrefix*",
        "arn:aws:s3:::awsserverlessrepo-changesets-1iiv3xa62ln3m/*",
        "arn:aws:iam::*:role/RoleName",
        "arn:aws:iam:*:111122223333:policy/*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateUploadBucket",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:ListExports",
        "cloudformation:ListStacks",
        "cloudformation:ListImports",
        "lambda:ListFunctions",
        "iam:ListRoles",
        "lambda:GetAccountSettings",
        "ec2:DescribeSecurityGroups",
        "cloudformation:EstimateTemplateCost",
        "ec2:DescribeVpcs",
        "lambda:ListEventSourceMappings",
        "cloudformation:DescribeAccountLimits",
        "ec2:DescribeSubnets",
        "cloudformation:CreateStackSet",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
},
{
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": "cloudformation:*",
    "Resource": [
        "arn:aws:cloudformation:*:111122223333:stack/aws-serverless-
repository-MyCFStackPrefix*/*",
        "arn:aws:cloudformation:*:111122223333:stack/
serverlessrepo-MyCFStackPrefix*/*",
        "arn:aws:cloudformation::*:transform/Serverless-*",

```

```

        "arn:aws:cloudformation:*:111122223333:stackset/aws-serverless-
repository-MyCFStackPrefix*:*"),
        "arn:aws:cloudformation:*:111122223333:stackset/
serverlessrepo-MyCFStackPrefix*:*"
    ]
  },
  {
    "Sid": "VisualEditor3",
    "Effect": "Allow",
    "Action": "serverlessrepo:*",
    "Resource": "arn:aws:serverlessrepo:*:*:applications/*"
  }
]
}

```

## 許可說明

允許的動作	說明
<pre> "lambda:CreateFunction", "lambda:ListVersionsByFunction", "lambda:GetFunctionConfiguration", "lambda:PutFunctionConcurrency", "lambda:ListTags", "lambda&gt;DeleteFunction", "lambda:GetAlias", "lambda:InvokeFunction", "lambda:GetFunction", "lambda:ListAliases", "lambda:UpdateFunctionConfigur ation", "lambda:UpdateFunctionCode", "lambda:AddPermission", "lambda&gt;DeleteFunctionConcurrency", "lambda:RemovePermission", "lambda:GetPolicy" "lambda:GetAccountSettings", "lambda:ListFunctions", "lambda:ListEventSourceMappings", </pre>	<p>允許建立和管理列為資源的 Lambda 函數。在此範例中，資源識別碼會使用名稱前置詞 <code>arn:aws:lambda:*: <i>MyAWSAcct Id</i> :function: <i>MyAthenaLambdaFunctionsPrefix</i> *</code>，其中 <i>MyAthenaLambdaFunctionsPrefix</i> 是在 Lambda 函數群組名稱中使用的共用前置詞，因此不需要將它們個別指定為資源。您可以指定一或多個 Lambda 函數資源。</p>
<pre> "s3:GetObject" </pre>	<p>允許讀取資源識別碼所指定 AWS Serverless Application Repository 需要的值區 <code>arn:aws:s</code></p>

允許的動作	說明
<code>"cloudformation:*"</code>	<p>3::awsserverlessrepo-changesets- <i>1iiv3xa62ln3m</i> /*。</p> <p>允許建立和管理由資源 <i>myCF StackPrefix</i> 指定的 AWS CloudFormation 堆疊。這些堆疊和堆疊集是 AWS Serverless Application Repository 部署連接器和 UDF 的方式。</p>
<code>"serverlessrepo:*"</code>	<p>允許在中搜尋、檢視、發行和更新資源識別碼所指定的應用程式 <code>arn:aws:serverlessrepo:*:*:applications/*</code>。AWS Serverless Application Repository</p>

## 允許使用 Athena 存取機器學習 (ML)

必須允許執行 Athena 機器學習 (ML) 查詢的 IAM 主體對其使用的 Sagemaker 端點執行 `sagemaker:invokeEndpoint` 動作。在連接至使用者身分的以身分為基礎的許可政策中，加入類似下列的政策陳述式。此外，請連接 [AWS 受管理的策略：AmazonAthenaFullAccess](#)，以授權完整存取 Athena 動作，或連接已修改的內嵌政策，以允許動作子集。

在範例中，以要在查詢中使用的模型端點的一或多個 ARN，替換 `arn:aws:sagemaker:region:AWSacctID:ModelEndpoint`。如需詳細資訊，請參閱服務授權參考 SageMaker 中的動作、資源和條件索引 [鍵](#)。

```
{
  "Effect": "Allow",
  "Action": [
    "sagemaker:invokeEndpoint"
  ],
  "Resource": "arn:aws:sagemaker:us-west-2:123456789012:workteam/public-crowd/default"
}
```

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

## 啟用對 Athena API 的聯合存取

本節討論的聯合存取，可讓組織中的使用者或用戶端應用程式呼叫 Amazon Athena API 操作。在這種情況下，組織的使用者無法直接存取 Athena。相反，您可以管理 Microsoft 活動目錄之外 AWS 的使用者認證。Active Directory 支援 [SAML 2.0](#) (安全聲明標記語言 2.0)。

若要在此案例中驗證使用者，請使用支援 SAML 2.0 的 JDBC 或 ODBC 驅動程式，以存取 Active Directory Federation Services (ADFS) 3.0，並讓用戶端應用程式能夠呼叫 Athena API 操作。

如需有關 SAML 2.0 支援的詳細資訊 AWS，請參閱《IAM 使用者指南》中的[關於 SAML 2.0 聯合](#)。

### Note

特定類型的身分提供者 (IdP)，即 Active Directory Federation Service (ADFS 3.0) (Windows Server 的一部分) 支援對 Athena API 的聯合存取。聯合存取與 IAM Identity Center 信任身分傳播功能不相容。透過支援 SAML 2.0 的 JDBC 或 ODBC 驅動程式版本建立存取權。如需詳細資訊，請參閱 [使用 JDBC 連接到 Amazon Athena](#) 及 [使用 ODBC 連接到 Amazon Athena](#)。

### 主題

- [開始之前](#)
- [架構圖](#)
- [程序：對 Athena API 的 SAML 型聯合存取](#)

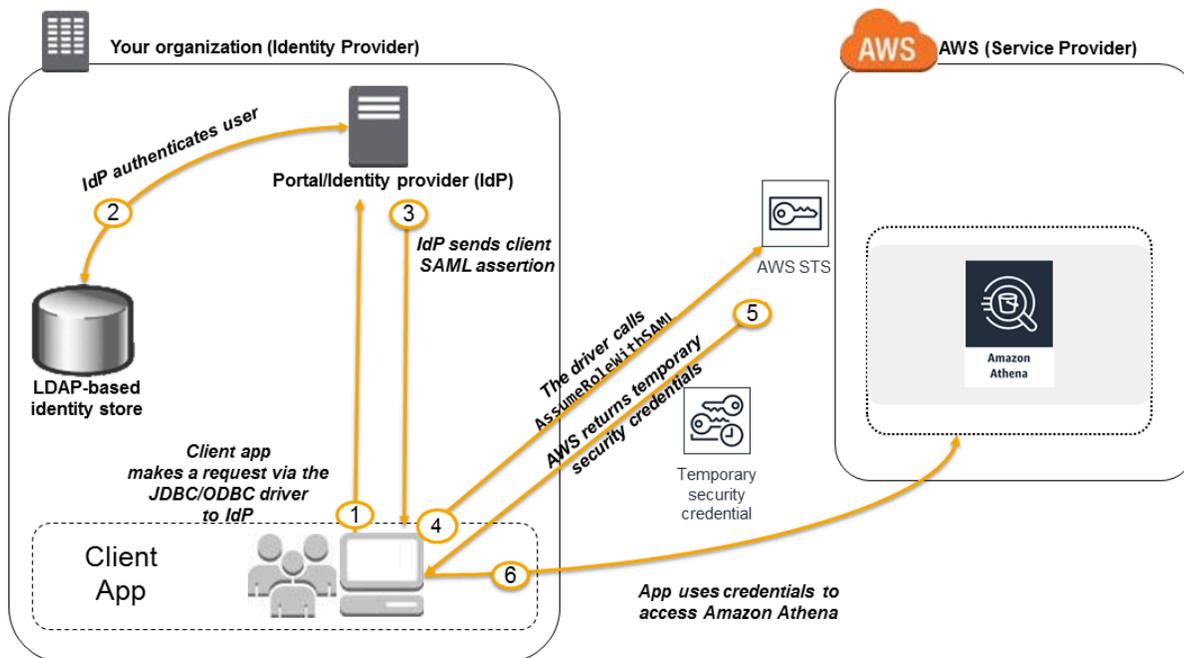
### 開始之前

開始之前，請先完成以下先決條件：

- 在您的組織內，安裝 ADFS 3.0 並將其設定為您的 IdP。
- 在用來存取 Athena 的用戶端上，安裝和設定最新可用版本的 JDBC 或 ODBC 驅動程式。此驅動程式必須支援與 SAML 2.0 相容的聯合身分存取。如需詳細資訊，請參閱 [使用 JDBC 連接到 Amazon Athena](#) 及 [使用 ODBC 連接到 Amazon Athena](#)。

### 架構圖

下圖說明此程序。



1. 組織中的使用者會使用用戶端應用程式搭配 JDBC 或 ODBC 驅動程式，請求獲得組織 IdP 的身分驗證。IdP 是 ADFS 3.0。
2. IdP 根據 Active Directory (組織的身分存放區) 對使用者進行身分驗證。
3. IdP 會建構一個具有使用者相關資訊的 SAML 聲明，並將此聲明經由 JDBC 或 ODBC 驅動程式發送到用戶端應用程式。
4. JDBC 或 ODBC 驅動程式會呼叫 AWS Security Token Service [AssumeRoleWithSAML](#) API 作業，並將下列參數傳遞給它：
  - SAML 供應商的 ARN
  - 要擔任的角色 ARN
  - 來自 IdP 的 SAML 聲明

如需詳細資訊，請參閱 AWS Security Token Service API 參考資料中的 [AssumeRoleWithSAML](#)。
5. API 透過 JDBC 或 ODBC 驅動程式傳給用戶端應用程式的回應，包含臨時安全登入資料。
6. 用戶端應用程式使用臨時安全憑證呼叫 Athena API 操作，讓您的使用者可以存取 Athena API 操作。

## 程序：對 Athena API 的 SAML 型聯合存取

此程序會在組織的 IdP 和您的 AWS 帳戶之間建立信任，以啟用以 SAML 為基礎的聯合身分存取 Amazon Athena API 作業。

若要啟用對 Athena API 的聯合存取：

1. 在您的組織中，在您的 IdP 中註冊 AWS 為服務提供者 (SP)。這個程序稱為依賴方信任。如需詳細資訊，請參閱《IAM 使用者指南》中的[使用依賴方信任設定您的 SAML 2.0 IdP](#)。請執行以下步驟，做為此任務的一部分：
  - a. 從這個 URL 取得範例 SAML 中繼資料文件：<https://signin.aws.amazon.com/static/saml-metadata.xml>。
  - b. 在組織的 IdP (ADFS) 中，產生一個對等的中繼資料 XML 檔案，將您的 IdP 描述為的身分識別提供者。AWS 您的中繼資料檔案必須包含發行者名稱、建立日期、到期日，以及 AWS 用來驗證組織的驗證回應 (宣告) 的金鑰。
2. 在 IAM 主控台中，您可以建立一個 SAML 身分提供者實體。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立 SAML 身分提供者](#)。執行以下操作，做為此步驟的一部分：
  - a. 前往 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
  - b. 上傳在本程序步驟 1 中 IdP (ADFS) 產生的 SAML 中繼資料文件。
3. 在 IAM 主控台中為您的 IdP 建立一或多個 IAM 角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[為第三方身分提供者 \(聯合\) 建立角色](#)。執行以下操作，做為此步驟的一部分：
  - 在該角色的許可政策中，列出允許組織的使用者在 AWS 中執行的動作。
  - 在角色的信任政策中，將您在此程序步驟 2 中建立的 SAML 提供者實體設定為主體。

這會建立組織與之間的信任關係 AWS。

4. 在組織的 IdP (ADFS) 中，定義可將組織中的使用者或群組映射到 IAM 角色的聲明。使用者和群組對 IAM 角色的映射也稱為宣告規則。請注意，組織中不同的使用者和群組可能映射到不同的 IAM 角色。

如需有關在 ADFS 中設定對應的資訊，請參閱部落格文章：[啟用聯合以 AWS 使用 Windows 作用中目錄、ADFS 和 SAML 2.0](#)。

5. 安裝並設定具 SAML 2.0 支援的 JDBC 和 ODBC 驅動程式。如需詳細資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#) 及 [使用 ODBC 連接到 Amazon Athena](#)。

- 指定從您的應用程式到 JDBC 和 ODBC 驅動程式的連接字串。如需有關應用程式應使用之連接字串的資訊，請參閱《JDBC 驅動程式安裝和設定指南》中的「使用 Active Directory 聯合服務 (ADFS) 憑證提供者」，或《ODBC 驅動程式安裝和設定指南》中的類似主題，可透過 PDF 形式從 [使用 JDBC 連接到 Amazon Athena](#) 和 [使用 ODBC 連接到 Amazon Athena](#) 主題中下載。

以下是設定驅動程式連接字串的高階彙總：

- 在 `AwsCredentialsProviderClass` configuration 中，設定 `com.simba.athena.iamsupport.plugin.AdfsCredentialsProvider` 以表示您想要透過 ADFS IdP 使用的 SAML 2.0 身分驗證。
- 在 `idp_host` 提供 ADFS IdP 伺服器的主機名稱。
- 在 `idp_port` 提供 ADFS IdP 監聽 SAML 聲明請求的連接埠號碼。
- 在 `UID` 和 `PWD` 提供 AD 網域使用者登入資料。在 Windows 上使用驅動程式時，若未提供 `UID` 和 `PWD`，驅動程式會嘗試取得登入該 Windows 機器之使用者的使用者登入資料。
- 您可以選擇將 `ssl_insecure` 設定為 `true`。在這種情況下，驅動程式不會檢查 ADFS IdP 伺服器的 SSL 憑證真偽。如果未將 ADFS IdP 的 SSL 憑證設定為受驅動程式信任，則必須設定為 `true`。
- 若要啟用 Active Directory 網域使用者或群組到一個或多個 IAM 角色的映射 (如本程式庫步驟 4 中所述)，在 JDBC 或 ODBC 連接的 `preferred_role` 中，指定驅動程式連接中要擔任的 IAM 角色 (ARN)。指定 `preferred_role` 為選擇性，但如果角色不是宣告規則中列出的第一個角色，則非常有用。

由於此程序，發生下列動作：

- [JDBC 或 ODBC 驅動程式會呼叫 S AWS STSAssumeRoleWithAML API，並將宣告傳遞給它，如架構圖的步驟 4 所示。](#)
- AWS 確保假定角色的請求來自 SAML 提供者實體中參照的 IdP。
- 如果要求成功，AWS STS [AssumeRoleWithSAML](#) API 作業會傳回一組臨時安全登入資料，您的用戶端應用程式會使用這些登入資料向 Athena 發出已簽署的要求。

您的應用程式現在已具有目前使用者相關資訊，且能以程式設計的方式存取 Athena。

## Athena 中的記錄和監控

若要偵測事件，在事件發生時收到提醒，以及回應它們，請搭配 Amazon Athena 使用這些選項：

- 監控 Athena AWS CloudTrail — [AWS CloudTrail](#)提供使用者、角色或 AWS 服務在 Athena 所採取的行動記錄。其會從 Athena 主控台擷取呼叫，並將 Athena API 操作的呼叫編碼為事件。其允許您判斷向 Athena 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 [記錄 Amazon Athena API 呼叫 AWS CloudTrail](#)。

您也可以使用 Athena 查詢不僅適用於 Athena 的 CloudTrail 記錄檔，還可以查詢其他記錄檔 AWS 服務。如需詳細資訊，請參閱 [查詢 AWS CloudTrail 記錄檔](#)。

- 使用 Amazon 監控 Athena 的使用 CloudTrail 情況 QuickSight — [Amazon QuickSight](#) 是全受管、雲端支援的商業智慧服務，可讓您建立組織可從任何裝置存取的互動式儀表板。有關使用 CloudTrail 和 Amazon QuickSight 監控 Athena 使用情況的解決方案示例，請參閱 [AWS 大數據博客文章 Realtor.com 如何監控 Amazon Athena AWS CloudTrail 與 Amazon 的使用情況](#)。QuickSight
- EventBridge 搭配 Athena 使用 — Amazon EventBridge 提供近乎即時的系統事件串流，用於描述 AWS 資源變更。EventBridge 透過傳送訊息以回應環境、啟動功能、進行變更以及擷取狀態資訊，在發生作業變更時瞭解作業變更、回應這些變更，並在必要時採取更正動作。盡可能發出事件。如需詳細資訊，請參閱 [Amazon EventBridge 使用者指南 EventBridge 中的開始使用 Amazon](#)。
- 使用工作群組分隔使用者、團隊、應用程式或工作負載，以及設定查詢限制和控制查詢成本 — 您可以在 Amazon 中檢視與查詢相關的指標 CloudWatch、設定掃描的資料量限制、建立閾值以及觸發動作 (例如 Amazon SNS 警示)，在超出這些閾值時控制查詢成本。如需高階程序，請參閱 [設定工作群組](#)。使用資源層級 IAM 許可來控制特定群組的存取。如需詳細資訊，請參閱 [使用工作群組來執行查詢](#)和 [使用指標和事件控制成本並 CloudWatch 監控查詢](#)。

## 主題

- [記錄 Amazon Athena API 呼叫 AWS CloudTrail](#)

## 記錄 Amazon Athena API 呼叫 AWS CloudTrail

Athena 與服務整合 AWS CloudTrail，可提供使用者、角色或 AWS 服務在 Athena 所採取的動作記錄的服務。

CloudTrail 以事件形式擷取 Athena 的所有 API 呼叫。擷取的呼叫包括從 Athena 主控台進行的呼叫，以及針對 Athena API 操作的程式碼呼叫。如果您建立追蹤，您可以啟用持續交付 CloudTrail 事件到 Amazon S3 儲存貯體，包括 Athena 的事件。如果您未設定追蹤，您仍然可以在 [事件歷程記錄] 中檢視 CloudTrail 主控台中最近的事件。

使用收集的資訊 CloudTrail，您可以判斷向 Athena 提出的要求、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱[AWS CloudTrail 用者指南](#)。

您可以使用 Athena 來查詢 Athena 本身和其他地方的 CloudTrail 記錄檔 AWS 服務。如需詳細資訊 [查詢 AWS CloudTrail 記錄檔蜂巢 SerDe](#)，請參閱和 AWS 大數據部落格文章[與 Amazon Athena 搭配使用 CTAS 陳述式來降低成本並改善效能](#)，以提供 Athena 使 CloudTrail 用情況的深入解析。

## Athena 資訊 CloudTrail

CloudTrail 當您創建帳戶時，您的 Amazon Web Services 帳戶已啟用。在 Athena 發生活動時，該活動會與事件歷史記錄中的其他 AWS 服務 CloudTrail 事件一起記錄在事件中。您可以檢視、搜尋和下載 Amazon Web Services 帳戶中的最近事件。如需詳細資訊，請參閱[使用 CloudTrail 事件歷程記錄檢視事件](#)。

如需 Amazon Web Services 帳戶中正在進行事件的記錄 (包含 Athena 的事件)，請建立線索。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。追蹤記錄來自 AWS 分區中所有區域的事件，並將日誌檔傳送到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他，AWS 服務 以進一步分析 CloudTrail 記錄中收集的事件資料並採取行動。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務與整合](#)
- [設定 Amazon SNS 通知 CloudTrail](#)
- [從多個區域接收 CloudTrail 日誌文件並從多個帳戶接收 CloudTrail 日誌文件](#)

所有的 Athena 動作都由記錄下來，CloudTrail 並記錄在 [Amazon Athena API 參考](#)中。例如，呼叫 [StartQueryExecution](#)和動 [GetQueryResults](#)作會在 CloudTrail 記錄檔中產生項目。

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：

- 要求是使用根使用者登入資料還是 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。
- 該請求是否由另一項 AWS 服務服務提出。

如需詳細資訊，請參閱[CloudTrail 使 userIdentity 元素](#)。

## 了解 Athena 日誌檔案項目

追蹤是一種組態，可讓事件以日誌檔的形式傳遞到您指定的 Amazon S3 儲存貯體。CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求，包括有關請求的操作，動作的日期和時間，請求參數等信息。CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

### Note

為了防止意外揭露敏感資訊，`StartQueryExecution` 和 `CreateNamedQuery` 日誌中的 `queryString` 項目的值都為 `***OMITTED***`。這是設計本身所致。若要存取實際的查詢字串，您可以使用 Athena [GetQueryExecution](#) API 並 `responseElements.queryExecutionId` 從 CloudTrail 記錄傳入的值。

下列範例會示範下列項目的 CloudTrail 記錄項目：

- [StartQueryExecution\(成功\)](#)
- [StartQueryExecution \(失敗\)](#)
- [CreateNamedQuery](#)

### StartQueryExecution (成功)

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "johndoe"
  },
  "eventTime": "2017-05-04T00:23:55Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "StartQueryExecution",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "77.88.999.69",
  "userAgent": "aws-internal/3",
  "requestParameters": {
```

```

    "clientRequestToken":"16bc6e70-f972-4260-b18a-db1b623cb35c",
    "resultConfiguration":{
      "outputLocation":"s3://DOC-EXAMPLE-BUCKET/test/"
    },
    "queryString":"***OMITTED***"
  },
  "responseElements":{
    "queryExecutionId":"b621c254-74e0-48e3-9630-78ed857782f9"
  },
  "requestID":"f5039b01-305f-11e7-b146-c3fc56a7dc7a",
  "eventID":"c97cf8c8-6112-467a-8777-53bb38f83fd5",
  "eventType":"AwsApiCall",
  "recipientAccountId":"123456789012"
}

```

### StartQueryExecution (失敗)

```

{
  "eventVersion":"1.05",
  "userIdentity":{
    "type":"IAMUser",
    "principalId":"EXAMPLE_PRINCIPAL_ID",
    "arn":"arn:aws:iam::123456789012:user/johndoe",
    "accountId":"123456789012",
    "accessKeyId":"EXAMPLE_KEY_ID",
    "userName":"johndoe"
  },
  "eventTime":"2017-05-04T00:21:57Z",
  "eventSource":"athena.amazonaws.com",
  "eventName":"StartQueryExecution",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"77.88.999.69",
  "userAgent":"aws-internal/3",
  "errorCode":"InvalidRequestException",
  "errorMessage":"Invalid result configuration. Should specify either output location or result configuration",
  "requestParameters":{
    "clientRequestToken":"ca0e965f-d6d8-4277-8257-814a57f57446",
    "queryString":"***OMITTED***"
  },
  "responseElements":null,
  "requestID":"aefbc057-305f-11e7-9f39-bbc56d5d161e",
  "eventID":"6e1fc69b-d076-477e-8dec-024ee51488c4",

```

```
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

## CreateNamedQuery

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "johndoe"
  },
  "eventTime": "2017-05-16T22:00:58Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "CreateNamedQuery",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "77.88.999.69",
  "userAgent": "aws-cli/1.11.85 Python/2.7.10 Darwin/16.6.0 botocore/1.5.48",
  "requestParameters": {
    "name": "johndoetest",
    "queryString": "****OMITTED****",
    "database": "default",
    "clientRequestToken": "fc1ad880-69ee-4df0-bb0f-1770d9a539b1"
  },
  "responseElements": {
    "namedQueryId": "cdd0fe29-4787-4263-9188-a9c8db29f2d6"
  },
  "requestID": "2487dd96-3a83-11e7-8f67-c9de5ac76512",
  "eventID": "15e3d3b5-6c3b-4c7c-bc0b-36a8dd95227b",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
},
```

## Amazon Athena 的合規驗證

在多個 AWS 合規計劃中，第三方稽核人員會評估 Amazon Athena 的安全性與合規性。這些計劃包括 SOC、PCI、FedRAMP 等等。

如需特定合規計劃範圍內的 AWS 服務清單，請參閱[合規計劃內的 AWS 服務](#)。如需一般資訊，請參閱[AWS 合規計劃](#)。

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱[在 AWS Artifact 中下載報告](#)。

您在使用 Athena 時的合規責任，取決於資料的敏感性、您公司的合規目標，以及適用的法律和法規。AWS 會提供以下資源協助您處理合規事宜：

- [安全與合規快速入門指南](#) – 這些部署指南討論在 AWS 上部署以安全及合規為重心基準環境的架構考量和步驟。
- [Amazon Web Services 的 HIPAA 安全與合規架構](#) – 本白皮書說明公司可如何運用 AWS 來建立符合 HIPAA 規範的應用程式。
- [AWS 合規資源](#) – 這組手冊和指南可能適用於您的產業和位置。
- [AWS Config](#) – 此 AWS 服務可評定資源組態與內部實務、業界準則和法規的合規狀態。
- [AWS Security Hub](#) – 此 AWS 服務可供您檢視 AWS 中的安全狀態，可助您檢查是否符合安全產業標準和最佳實務。

## Athena 中的復原功能

AWS 全球基礎架構是圍繞 AWS 區域和可用區域建立的。AWS 區域提供多個實體分離和隔離的可用區域，這些區域與低延遲、高輸送量和高冗餘網路相連。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 全域基礎結構](#)。

除了 AWS 全球基礎架構外，Athena 還提供多種功能，協助支援您的資料恢復能力和備份需求。

Athena 為無伺服器，無須設定或管理任何基礎設施。Athena 具有高可用性，並跨多個可用區域使用運算資源來執行查詢，如果無法連線特定可用區域，便會自動適當地路由查詢。Athena 會使用 Amazon S3 作為其基礎資料存放區，讓您的資料具有高可用性和耐用性。Amazon S3 會提供耐用的基礎設施來存放重要資料，其旨在提供 99.999999999% 的物件耐用性。您的資料會以冗餘方式存放在多個設施以及每個設施的多個裝置。

## Athena 中的基礎設施安全

作為一項受管服務，Amazon Athena 受到 AWS 全球網路安全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱[AWS 雲端安全](#)。若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱[安全性支柱架構](#)。良 AWS 好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫透過網路存取 Athena。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。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

使用 IAM 政策來限制對 Athena 操作的存取。每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

Athena [受管政策](#)不但易於使用，且會隨著服務的演進以所需動作自動更新。客戶受管政策和內嵌政策允許您透過在政策內指定更精密的 Athena 動作來微調這些政策。授予對資料的 Amazon S3 位置之適當存取權。如需有關如何授予 Amazon S3 存取權的詳細資訊和案例，請參閱《Amazon Simple Storage Service 開發人員指南》中的 [演練範例：管理存取權](#)。如需詳細資訊以及允許哪些 Amazon S3 動作的範例，請參閱 [跨帳戶存取](#) 中的儲存貯體政策範例。

## 主題

- [使用界面 VPC 端點連接至 Amazon Athena](#)

## 使用界面 VPC 端點連接至 Amazon Athena

在虛擬私有雲端 (VPC) 中使用 [介面 VPC 端點 \(AWS PrivateLink\)](#) 和 [AWS Glue VPC 端點](#)，改善 VPC 的安全狀態。介面 VPC 端點可讓您控制從 VPC 內部到達哪些目的地，從而提高安全性。每個 VPC 端點皆會由一個或多個具私有 IP 地址 [彈性網路介面](#) (ENI) 來表示，而該介面位於 VPC 子網路中。

介面虛擬私人雲端端點可將您的 VPC 直接連接至 Athena，而無需網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址，就能與 Athena API 進行通訊。

若要透過 VPC 來使用 Athena，您必須從 VPC 中的執行個體來連接，或使用 Amazon Virtual Private Network (VPN) 或 AWS Direct Connect 將私有網路連接至 VPC。如需有關 Amazon VPN 的資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 [VPN 連接](#)。若要取得有關資訊 AWS Direct Connect，請參閱《[使用指南](#)》中的 [AWS Direct Connect](#) [〈建立連接〉](#)。

在所有 [Amazon VPC 和 Athena 均可使用的情 AWS 區域](#) 況下，Athena 都支援 VPC 端點。

您可以建立介面 VPC 端點，以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI) 命令連線至 Athena。如需詳細資訊，請參閱[建立介面端點](#)。

在建立介面 VPC 端點之後，如果您啟用端點的[私有 DNS](#) 主機名稱，則預設 Athena 端點 (<https://athena.Region.amazonaws.com>) 會解析為您的 VPC 端點。

如果您尚未啟用私有 DNS 主機名稱，Amazon VPC 會透過以下格式提供一個 DNS 端點名稱，供您使用：

```
VPC_Endpoint_ID.athena.Region.vpce.amazonaws.com
```

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的介面虛擬私人雲端端點 ([AWS PrivateLink](#))。

Athena 會支援在您的 VPC 內呼叫其所有 [API 動作](#)。

### 為 Athena 建立 VPC 端點政策

您可以為 Athena 的 Amazon VPC 端點建立政策，以指定下列限制：

- 主體 – 可執行動作的主體。
- 動作 – 可執行的動作。
- 資源 – 可供執行動作的資源。
- 僅受信任的身分 — 使用此 `aws:PrincipalOrgId` 條件可限制只存取屬於您 AWS 組織中的認證。這有助於防止非預期的主體存取。
- 僅可信資源 – 使用 `aws:ResourceOrgId` 條件，以防止存取非預期的資源。
- 僅可信身分和資源 – 為 VPC 端點建立組合政策，以協助防止存取非預期的主體和資源。

有關詳情，請參閱 Amazon VPC 使用者指南中的[使用 VPC 端點控制服務的存取](#)和 AWS 白皮書中的[附錄 2 — VPC 端點政策範例](#)中的建立資料周邊。AWS

### Example – VPC 端點政策

下列範例允許依組織識別對組織資源的要求，並允許 AWS 服務主體的要求。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
```

```
    "Principal": {
      "AWS": "*"
    },
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:PrincipalOrgID": "my-org-id",
        "aws:ResourceOrgID": "my-org-id"
      }
    }
  },
  {
    "Sid": "AllowRequestsByAWSServicePrincipals",
    "Effect": "Allow",
    "Principal": {
      "AWS": "*"
    },
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "Bool": {
        "aws:PrincipalIsAWSService": "true"
      }
    }
  }
]
}
```

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

### 共用子網路

無法在與您共用的子網路中建立、描述、修改或刪除 VPC 端點。不過，可以在與您共用的子網路中使用 VPC 端點。如需有關 VPC 子網路共用的資訊，請參閱《Amazon VPC 使用者指南》中的 [與其他帳戶共用 VPC](#)。

## Athena 中的組態與漏洞分析

Athena 是無伺服器服務，因此無需設定或管理基礎架構。AWS 處理基本安全性工作，例如客體作業系統 (OS) 和資料庫修補、防火牆組態和嚴重損壞修復。這些程序已由適當的第三方進行檢閱並認證。如需詳細資訊，請參閱下列 AWS 資源：

- [共同的責任模型](#)
- [安全性、身分與合規的最佳實務](#)

## 使用 Athena 查詢向 AWS Lake Formation 註冊的資料

[AWS Lake Formation](#) 可讓您在使用 Athena 查詢來讀取存放在 Simple Storage Service (Amazon S3) 中的資料時，定義並強制執行資料庫、資料表和資料欄層級存取政策。Lake Formation 會針對存放在 Simple Storage Service (Amazon S3) 中的資料提供授權和管理層。您可以使用 Lake Formation 中的許可階層來授予或撤銷讀取資料目錄物件 (例如資料庫、資料表和資料欄) 的許可。Lake Formation 可簡化許可的管理，並讓您實作資料的精細定義存取控制 (FGAC)。

您可以使用 Athena 查詢向 Lake Formation 註冊的資料，以及未向 Lake Formation 註冊的資料。

使用 Athena 查詢已向 Lake Formation 註冊之 Simple Storage Service (Amazon S3) 位置的來源資料時，Lake Formation 許可即適用。當您建立指向已註冊 Simple Storage Service (Amazon S3) 資料位置的資料庫和資料表時，Lake Formation 許可也適用。若要使用 Athena 搭配使用 Lake Formation 註冊的資料，Athena 必須設定為使用 AWS Glue Data Catalog。

寫入物件至時 Simple Storage Service (Amazon S3)，Lake Formation 許可不適用，也不適用於查詢存放在 Simple Storage Service (Amazon S3) 中的資料，或未向 Lake Formation 註冊的中繼資料。對於 Amazon S3 中的來源資料和未向 Lake Formation 註冊的中繼資料，存取權由 Amazon S3 的 IAM 許可政策和 AWS Glue 動作決定。Simple Storage Service (Amazon S3) 中的 Athena 查詢結果位置無法向 Lake Formation 註冊，且 Simple Storage Service (Amazon S3) 的 IAM 許可政策控制存取權限。此外，Lake Formation 許可不適用於 Athena 查詢歷史記錄。您可以使用 Athena 工作群組來控制查詢歷史記錄的存取。

如需有關 Lake Formation 的詳細資訊，請參閱 [Lake Formation 常見問答集](#) 與 [《AWS Lake Formation 開發人員指南》](#)。

### 主題

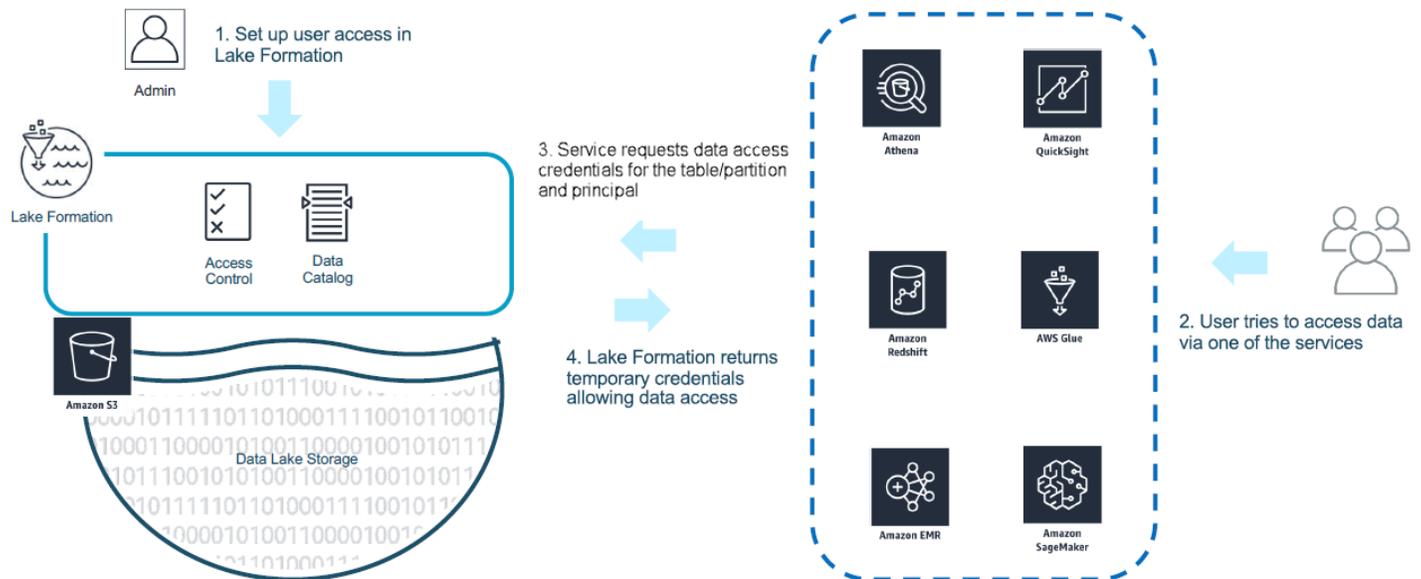
- [Athena 如何存取向 Lake Formation 註冊的資料](#)
- [使用 Athena 查詢向 Lake Formation 註冊之資料的考量事項與限制](#)
- [管理 Lake Formation 和 Athena 使用者許可](#)
- [將 Lake Formation 許可套用至現有資料庫和資料表](#)
- [使用 Lake Formation 和 Athena JDBC 和 ODBC 驅動程式對 Athena 進行聯合存取](#)

## Athena 如何存取向 Lake Formation 註冊的資料

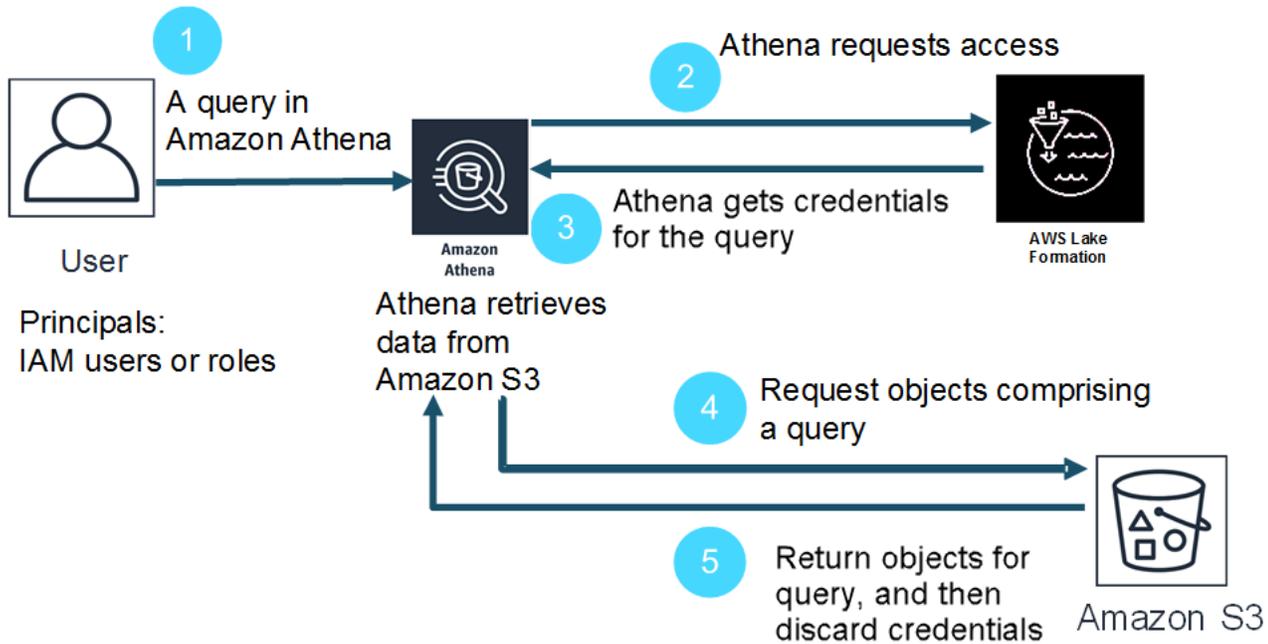
本節所述的存取工作流程僅適用於在 Simple Storage Service (Amazon S3) 位置，以及針對向 Lake Formation 註冊的中繼資料物件執行 Athena 查詢時。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[註冊資料湖](#)。除了註冊資料外，Lake Formation 管理員也會應用 Lake Formation 許可，授予或撤銷對資料目錄中繼資料以及 Simple Storage Service (Amazon S3) 資料位置的存取權限。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[中繼資料與資料的安全性與存取控制](#)。

每當 Athena 主體 (使用者、群組或角色) 對使用 Lake Formation 註冊的資料執行查詢時，Lake Formation 都會驗證主體是否有適合查詢的資料庫、資料表和 Simple Storage Service (Amazon S3) 位置的適當 Lake Formation 許可。如果主體有權存取，Lake Formation 會將暫時性憑證提供給 Athena，且查詢會執行。

下圖為上述流程的說明。



下圖顯示了在 Athena 的憑證自動售貨如何運作，以便在 Lake Formation 中註冊的 Amazon S3 位置的表格上進行假設性 SELECT 查詢：query-by-query



1. 主體在 Athena 中執行 SELECT 查詢。
2. Athena 會分析查詢並檢查 Lake Formation 許可，以查看是否授予主體對資料表和資料欄的存取權。
3. 如果主體有權存取，則 Athena 會向 Lake Formation 請求憑證。如果主體沒有存取權，Athena 會發出存取遭拒錯誤。
4. Lake Formation 會在從 Simple Storage Service (Amazon S3) 讀取資料時發出憑證給 Athena，以及允許的資料欄之清單。
5. Athena 使用 Lake Formation 臨時憑證來查詢來自 Simple Storage Service (Amazon S3) 的資料。查詢完成後，Athena 會捨棄憑證。

## 使用 Athena 查詢向 Lake Formation 註冊之資料的考量事項與限制

使用 Athena 查詢在 Lake Formation 中註冊的資料時請考慮以下項目。如需其他資訊，請參閱《AWS Lake Formation 開發人員指南》中的 [AWS Lake Formation 的已知問題](#)。

### 考量事項與限制

- [在 Avro 和 custom 的某些情況下，未經授權的用戶可以看到列元數據 SerDe](#)
- [使用檢視的 Lake Formation 許可](#)
- [Lake Formation 精細存取控制和 Athena 工作群組](#)
- [Amazon S3 中的 Athena 查詢結果位置未向 Lake Formation 註冊](#)

- [使用 Athena 工作群組限制查詢歷史記錄的存取](#)
- [跨帳戶存取資料目錄](#)
- [在 Lake Formation 註冊的 CSE-KMS 加密 Amazon S3 位置](#)
- [向 Lake Formation 註冊的分割資料位置必須位於資料表子目錄中](#)
- [Create Table As Select \(CTAS\) 查詢需要 Amazon S3 寫入許可](#)
- [預設資料庫需要 DESCRIBE 許可](#)

在 Avro 和 custom 的某些情況下，未經授權的用戶可以看到列元數據 SerDe

Lake Formation 資料欄層級授權可防止沒有 Lake Formation 許可的使用者存取資料欄中的資料。不過，在某些情況下，使用者可以存取描述資料表中所有資料欄的中繼資料，包括他們沒有資料許可的資料欄。

當資料行中繼資料儲存在使用 Apache Avro 儲存格式的資料表屬性中，或使用自訂序列化程式/還原序列化程式 (SerDe)，其中資料表結構定義與定義一起在資料表屬性中定義時，就會發生這種情況。SerDe 搭配 Lake Formation 使用 Athena 時，我們建議您檢閱向 Lake Formation 註冊的資料表屬性內容，並盡可能限制資料表屬性中所存放的資訊，以防止使用者看到任何敏感中繼資料。

### 使用檢視的 Lake Formation 許可

對於向 Lake Formation 註冊的資料，Athena 使用者只有在擁有資料表、資料欄，以及來源 Simple Storage Service (Amazon S3) 資料位置 (此為 VIEW 之根據) 之 Lake Formation 許可時，才能建立 VIEW。在 Athena 中建立 VIEW 之後，Lake Formation 許可即可套用至 VIEW。資料欄層級許可不適用於 VIEW。具有 VIEW 的 Lake Formation 許可，但沒有資料表和資料欄許可 (此為檢視之根據) 的使用者，無法使用 VIEW 來查詢資料。不過，具有此混合許可的使用者可以使用陳述式 (例如 DESCRIBE VIEW、SHOW CREATE VIEW 和 SHOW COLUMNS) 來查看 VIEW 中繼資料。因此，請務必針對每個 VIEW 調整 Lake Formation 許可以符合基礎資料表的許可。在資料表中定義的儲存格篩選條件不適用於針對該資料表的 VIEW。資源連結名稱必須與原始帳戶中的資源名稱相同。在跨帳戶設定中使用檢視時，還會有其他限制。如需有關設定跨帳戶的共用視圖的許可的詳細資訊，請參閱[跨帳戶存取資料目錄](#)。

### Lake Formation 精細存取控制和 Athena 工作群組

同一 Athena 工作群組中的使用者可以看到 Lake Formation 精細存取控制設定為可供工作群組存取的資料。如需有關在 Lake Formation 中使用精細定義存取控制的詳細資訊，請參閱 AWS 大數據部落格中的[使用 AWS Lake Formation 管理精細定義存取控制](#)。

## Amazon S3 中的 Athena 查詢結果位置未向 Lake Formation 註冊

Simple Storage Service (Amazon S3) 中的 Athena 查詢結果無法向 Lake Formation 註冊。Lake Formation 許可不會限制對這些位置的存取。除非您限制存取，否則 Athena 使用者可以在沒有 Lake Formation 的資料許可時存取查詢結果檔案和中繼資料。若要避免這種情況，我們建議您使用工作群組來指定查詢結果的位置，並使工作群組成員資格與 Lake Formation 許可保持一致。然後，您可以使用 IAM 許可政策來限制查詢結果位置的存取。如需有關查詢結果的詳細資訊，請參閱[使用查詢結果、近期查詢和輸出檔案](#)。

### 使用 Athena 工作群組限制查詢歷史記錄的存取

Athena 查詢歷史記錄會公開已儲存查詢和完整查詢字串的清單。除非您使用工作群組來分隔查詢歷程記錄的存取權限，否則未獲授權在 Lake Formation 中查詢資料的 Athena 使用者可以檢視對該資料執行的查詢字串，包括資料欄名稱、選取準則等。我們建議您使用工作群組來分隔查詢歷程記錄，並使 Athena 工作群組成員資格與 Lake Formation 許可保持一致，以限制存取。如需詳細資訊，請參閱[使用工作群組來控制查詢存取和成本](#)。

### 跨帳戶存取資料目錄

若要存取另一個帳戶中的資料目錄，您可以使用 Athena 的跨帳戶 AWS Glue 功能或在 Lake Formation 中設定跨帳戶存取。

### Athena 跨帳戶存取資料目錄

您可以使用 Athena 的跨帳戶 AWS Glue 目錄功能，在帳戶中註冊目錄。僅 Athena 引擎版本 2 及更新版本中提供此功能，且僅限於帳戶之間的相同區域使用。如需詳細資訊，請參閱[AWS Glue Data Catalog 從另一個帳戶註冊](#)。

如果要共用的資料目錄已在中設定資源策略 AWS Glue，則必須更新該策略，以允許存取帳戶 B AWS Resource Access Manager 並授與使用帳戶 A 的資料目錄的權限，如下列範例所示。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "ram.amazonaws.com"
    },
    "Action": "glue:ShareResource",
    "Resource": [
      "arn:aws:glue:<REGION>:<ACCOUNT-A>:table/*/*",
```

```

        "arn:aws:glue:<REGION>:<ACCOUNT-A>:database/*",
        "arn:aws:glue:<REGION>:<ACCOUNT-A>:catalog"
    ]
},
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::<ACCOUNT-B>:root"
    },
    "Action": "glue:*",
    "Resource": [
        "arn:aws:glue:<REGION>:<ACCOUNT-A>:table/*/*",
        "arn:aws:glue:<REGION>:<ACCOUNT-A>:database/*",
        "arn:aws:glue:<REGION>:<ACCOUNT-A>:catalog"
    ]
}
]
}

```

如需詳細資訊，請參閱 [跨帳戶存取 AWS Glue 資料目錄](#)。

在 Lake Formation 中設定跨帳戶存取

AWS Lake Formation 可讓您使用單一帳戶來管理中央資料目錄。您可以使用此功能來實作對資料目錄中繼資料和基礎資料的 [跨帳戶存取](#)。例如，擁有者帳戶可以授予另一個 (收件人) 帳戶對資料表的 SELECT 許可。

若要在 Athena 查詢編輯器中顯示共用的資料庫或資料表，您需在 Lake Formation 中對共用的資料庫或資料表 [建立資源連結](#)。當 Lake Formation 中的收件者帳戶查詢擁有者的資料表時，會將資料存取事件 [CloudTrail](#) 新增至收件者帳戶和擁有者帳戶的記錄中。

如需共用視圖，請謹記下列要點：

- 在目標資源連結上執行，而不是來源資料表或視圖上執行查詢，然後將輸出共用到目標帳戶。
- 僅共用視圖並不足夠。建立視圖所涉及的所有資料表必須是跨帳戶共用的一部分。
- 在共用資源上建立的資源連結的名稱必須與擁有者帳戶中的資源名稱相符。如果名稱不匹配，則出現錯誤消息，如分析存儲視圖「awsdata atalog 失敗」。 *my-lf-resource-link*。 *my-lf-view'*: 第 3 行:結構####不存在發生。

如需有關在 Lake Formation 中跨帳戶存取的詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的以下資源：

## [跨帳戶存取](#)

### [資源連結在 Lake Formation 中如何運作](#)

#### [跨帳戶 CloudTrail 記錄](#)

在 Lake Formation 註冊的 CSE-KMS 加密 Amazon S3 位置

Athena 無法查詢開放表格格式 (OTF) 表格，例如具有下列特性的 Apache 冰山：

- 這些表格是以在 Lake Formation 註冊的 Amazon S3 資料位置為基礎。
- Amazon S3 中的物件會使用用戶端加密 (CSE) 加密。
- 加密使用 AWS KMS 客戶管理的金鑰 (CSE\_KMS)。

若要查詢使用 CSE\_KMS 金鑰加密的非 OTF 資料表，請將下列區塊新增至您用於 CSE 加密的 AWS KMS 金鑰原則中。<KMS\_KEY\_ARN>是加密數據的密 AWS KMS 鑰的 ARN。<IAM-ROLE-ARN>是 IAM 角色的 ARN，可在 Lake Formation 中註冊 Amazon S3 位置。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "*"
  },
  "Action": "kms:Decrypt",
  "Resource": "<KMS-KEY-ARN>",
  "Condition": {
    "ArnLike": {
      "aws:PrincipalArn": "<IAM-ROLE-ARN>"
    }
  }
}
```

向 Lake Formation 註冊的分割資料位置必須位於資料表子目錄中

向 Lake Formation 註冊的分割資料表必須在目錄中擁有分割資料，這些目錄是 Simple Storage Service (Amazon S3) 中資料表的子目錄。例如，具有 `s3://DOC-EXAMPLE-BUCKET/mytable` 位置和分割區 `s3://DOC-EXAMPLE-BUCKET/mytable/dt=2019-07-11`、`s3://DOC-EXAMPLE-BUCKET/mytable/dt=2019-07-12` 等的資料表，可以向 Lake Formation 註冊並使用 Athena 進行查詢。另一方面，位置 `s3://DOC-EXAMPLE-BUCKET/mytable` 的資料表和位於 `s3://DOC-`

EXAMPLE-BUCKET/dt=2019-07-11、s3://DOC-EXAMPLE-BUCKET/dt=2019-07-12 等的分割區，無法向 Lake Formation 註冊。此類分割區不是 s3://DOC-EXAMPLE-BUCKET/mytable 的子目錄，因此也無法從 Athena 讀取。

Create Table As Select (CTAS) 查詢需要 Amazon S3 寫入許可

Create Table As Statements (CTAS) 需要對資料表的 Simple Storage Service (Amazon S3) 位置的寫入存取權限。若要對在 Lake Formation 註冊的資料執行 CTAS 查詢，Athena 使用者除了適當的 Lake Formation 許可讀取資料位置之外，還必須具有 IAM 許可才能寫入資料表 Simple Storage Service (Amazon S3) 位置。如需詳細資訊，請參閱[從查詢結果建立資料表 \(CTAS\)](#)。

預設資料庫需要 DESCRIBE 許可

default 資料庫需要 Lake Formation [DESCRIBE](#) 許可。下列範例 AWS CLI 命令會將 default 資料庫的 DESCRIBE 權限授與 AWS 帳戶 datalake\_user1 中的使用者 111122223333。

```
aws lakeformation grant-permissions --principal
  DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/datalake_user1 --
permissions "DESCRIBE" --resource '{"Database": {"Name":"default"}}
```

如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的 [Lake Formation 許可參考](#)。

## 管理 Lake Formation 和 Athena 使用者許可

Lake Formation 提供憑證來查詢向 Lake Formation 註冊的 Simple Storage Service (Amazon S3) 資料存放區。如果您之前使用 IAM 政策來允許或拒絕讀取 Simple Storage Service (Amazon S3) 中資料位置的許可，您可以改為使用 Lake Formation 許可。不過，仍需要其他 IAM 許可。

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

以下章節摘要說明使用 Athena 查詢在 Lake Formation 中註冊之資料所需的許可。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的 [AWS Lake Formation 的安全性](#)。

### 許可摘要

- [Lake Formation 和 Athena 的身分型許可](#)
- [Athena 查詢結果位置的 Amazon S3 許可](#)
- [查詢歷史記錄的 Athena 工作群組成員資格](#)
- [Lake Formation 的資料許可](#)
- [寫入 Amazon S3 位置的 IAM 許可](#)

- [加密資料、中繼資料和 Athena 查詢結果的許可](#)
- [外部帳戶中的 Amazon S3 儲存貯體以資源為基礎的許可 \(選用\)](#)

## Lake Formation 和 Athena 的身分型許可

任何使用 Athena 查詢向 Lake Formation 註冊之資料的人，都必須擁有允許 `lakeformation:GetDataAccess` 動作的 IAM 許可政策。[AWS 受管理的策略：AmazonAthenaFullAccess](#) 允許此動作。如果您使用內嵌政策，請務必更新許可政策來允許此動作。

在 Lake Formation 中，資料湖管理員具有建立中繼資料物件 (例如資料庫和資料表)、授予 Lake Formation 許可給其他使用者，以及註冊新 Simple Storage Service (Amazon S3) 位置的許可。若要註冊新的位置，需要 Lake Formation 的服務連結角色許可。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[建立資料湖管理員](#)和[Lake Formation 的服務連結角色許可](#)。

Lake Formation 使用者可以根據資料湖管理員授予的 Lake Formation 許可，使用 Athena 查詢資料庫、資料表、資料表資料欄和基礎 Simple Storage Service (Amazon S3) 資料存放區。使用者無法建立資料庫或資料表，或向 Lake Formation 註冊新的 Simple Storage Service (Amazon S3) 位置。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[建立資料湖使用者](#)。

在 Athena 中，身分型許可政策 (包括 Athena 工作群組的政策) 仍然控制 Amazon Web Services 帳戶使用者對 Athena 動作的存取。此外，還可能透過 Athena 驅動程式提供以 SAML 為基礎的身分驗證來提供聯合存取。如需詳細資訊，請參閱[使用工作群組來控制查詢存取和成本](#)、[用於存取工作群組的 IAM 政策](#)及[啟用對 Athena API 的聯合存取](#)。

如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[授予 Lake Formation 許可](#)。

## Athena 查詢結果位置的 Amazon S3 許可

Simple Storage Service (Amazon S3) 中的 Athena 查詢結果無法向 Lake Formation 註冊。Lake Formation 許可不會限制對這些位置的存取。除非您限制存取，否則 Athena 使用者可以在沒有 Lake Formation 的資料許可時存取查詢結果檔案和中繼資料。若要避免這種情況，我們建議您使用工作群組來指定查詢結果的位置，並使工作群組成員資格與 Lake Formation 許可保持一致。然後，您可以使用 IAM 許可政策來限制查詢結果位置的存取。如需有關查詢結果的詳細資訊，請參閱[使用查詢結果、近期查詢和輸出檔案](#)。

## 查詢歷史記錄的 Athena 工作群組成員資格

Athena 查詢歷史記錄會公開已儲存查詢和完整查詢字串的清單。除非您使用工作群組來分隔查詢歷程記錄的存取權限，否則未獲授權在 Lake Formation 中查詢資料的 Athena 使用者可以檢視對該資料執行的查詢字串，包括資料欄名稱、選取準則等。我們建議您使用工作群組來分隔查詢歷程記錄，並使

Athena 工作群組成員資格與 Lake Formation 許可保持一致，以限制存取。如需詳細資訊，請參閱[使用工作群組來控制查詢存取和成本](#)。

## Lake Formation 的資料許可

除了使用 Lake Formation 的基準許可之外，Athena 使用者還必須具備 Lake Formation 許可，才能存取其查詢的資源。這些許可是由 Lake Formation 管理員授予和管理。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》中的[中繼資料與資料的安全性與存取控制](#)。

## 寫入 Amazon S3 位置的 IAM 許可

對 Simple Storage Service (Amazon S3) 的 Lake Formation 許可不包含寫入 Simple Storage Service (Amazon S3) 的能力。Create Table As Statements (CTAS) 需要對資料表的 Simple Storage Service (Amazon S3) 位置的寫入存取權限。若要對在 Lake Formation 註冊的資料執行 CTAS 查詢，Athena 使用者除了適當的 Lake Formation 許可可以讀取資料位置之外，還必須具有 IAM 許可才能寫入資料表 Simple Storage Service (Amazon S3) 位置。如需詳細資訊，請參閱[從查詢結果建立資料表 \(CTAS\)](#)。

## 加密資料、中繼資料和 Athena 查詢結果的許可

在 Simple Storage Service (Amazon S3) 中的基礎來源資料，以及向 Lake Formation 註冊的資料目錄中繼資料都可以加密。使用 Athena 查詢向 Lake Formation 註冊的資料時，Athena 處理查詢結果加密的方式沒有變更。如需詳細資訊，請參閱[加密 Amazon S3 中存放的 Athena 查詢結果](#)。

- 加密來源資料 – 支援 Simple Storage Service (Amazon S3) 資料位置來源資料的加密。查詢向 Lake Formation 註冊的已加密 Simple Storage Service (Amazon S3) 位置的 Athena 使用者，需要加密和解密資料的許可。如需有關需求的詳細資訊，請參閱[支援的 Amazon S3 加密選項](#)和[Amazon S3 中加密資料的許可](#)。
- 加密中繼資料 – 支援加密資料目錄內的中繼資料。對於使用 Athena 的主體，身分型政策必須允許用於加密中繼資料之索引鍵的 "kms:GenerateDataKey"、"kms:Decrypt" 和 "kms:Encrypt" 動作。如需相關資訊，請參閱《AWS Glue 開發人員指南》和 [從 Athena 存取加密的中繼資料 AWS Glue Data Catalog](#) 中的[加密您的資料型錄](#)。

## 外部帳戶中的 Amazon S3 儲存貯體以資源為基礎的許可 (選用)

若要在不同帳戶中查詢 Simple Storage Service (Amazon S3) 資料位置，以資源為基礎的 IAM 政策 (儲存貯體政策) 必須允許存取該位置。如需詳細資訊，請參閱[Athena 中的 Amazon S3 儲存貯體跨帳戶存取](#)。

如需有關在其他帳戶中存取資料目錄的資訊，請參閱[Athena 跨帳戶存取資料目錄](#)。

## 將 Lake Formation 許可套用至現有資料庫和資料表

如果您是初次使用 Athena，且使用 Lake Formation 設定對查詢資料的存取，則不需要設定 IAM 政策以讓使用者能夠讀取 Simple Storage Service (Amazon S3) 資料和建立中繼資料。您可以使用 Lake Formation 來管理許可。

不需要向 Lake Formation 註冊資料和更新 IAM 許可政策。如果資料未向 Lake Formation 註冊，在 Amazon S3 中擁有適當許可的 Athena 使用者 (如果適用 AWS Glue) 可以繼續查詢未在 Lake Formation 註冊的資料。

如果您有查詢資料未向 Lake Formation 註冊的現有 Athena 使用者，則可以更新 Amazon S3 的 IAM 許可，以及更新適用的 IAM 許可 AWS Glue Data Catalog，以便您可以使用 Lake Formation 許可集中管理使用者存取權限。對於讀取 Simple Storage Service (Amazon S3) 資料位置的許可，您可更新以資源和以身分為基礎的政策，來修改 Simple Storage Service (Amazon S3) 許可。若要存取中繼資料，如果您設定了資源層級政策以進行精細的存取控制 AWS Glue，則可以改用 Lake Formation 權限來管理存取。

如需詳細資訊，請參閱[對於中的資料庫和資料表的精細存取 AWS Glue Data Catalog](#) 《AWS Lake Formation 開發人員指南》中的 [〈將資 AWS Glue 料權限升級至 AWS Lake Formation 模型〉](#)。

## 使用 Lake Formation 和 Athena JDBC 和 ODBC 驅動程式對 Athena 進行聯合存取

Athena JDBC 和 ODBC 驅動程式支援使用 Okta 和 Microsoft Active Directory 聯合服務 (AD FS) 身分提供者，實現以 SAML 2.0 為基礎的 Athena 聯合。透過整合 Amazon Athena AWS Lake Formation，您可以透過企業登入資料向 Athena 啟用基於 SAML 的身份驗證。使用 Lake Formation 和 AWS Identity and Access Management (IAM)，您可以對 SAML 使用者可用的資料維護精細的欄層級存取控制。透過 Athena JDBC 和 ODBC 驅動程式，聯合存取可用於工具或程式設計存取。

若要使用 Athena 存取由 Lake Formation 控制的資料來源，您需要透過設定身分提供者 (IdP) 和 AWS Identity and Access Management (IAM) 角色來啟用以 SAML 2.0 為基礎的聯合。如需詳細步驟，請參閱[教學課程：使用 Lake Formation 和 JDBC 設定 Okta 使用者對 Athena 的聯合存取](#)。

### 必要條件

若要使用 Amazon Athena 和 Lake Formation 進行聯合存取，您必須符合以下需求：

- 您使用現有以 SAML 為基礎的身分提供者來管理您的公司身分，例如 Okta 或 Microsoft Active Directory 聯合服務 (AD FS)。
- 您可以使用 AWS Glue Data Catalog 做為中繼資料存放區。

- 您在 Lake Formation 中定義和管理許可，以存取 AWS Glue Data Catalog 中的資料庫、資料表和資料欄。如需詳細資訊，請參閱《AWS Lake Formation 開發人員指南》<https://docs.aws.amazon.com/lake-formation/latest/dg/>。
- 您使用 2.0.14 版或更新版本的 [Athena JDBC 驅動程式](#)，或 1.1.3 版或更新版本的 [Athena ODBC 驅動程式](#)。

## 考量與限制

使用 Athena JDBC 或 ODBC 驅動程式和 Lake Formation 來設定對 Athena 的聯合存取時，請記住下列幾點：

- 目前，Athena JDBC 驅動程式和 ODBC 驅動程式支援 Okta、Microsoft Active Directory 聯合服務 (AD FS) 和 Azure AD 身分提供者。雖然 Athena JDBC 驅動程式具有可擴充至使用其他身分識別提供者的一般 SAML 類別，但是對於啟用其他身分識別提供者 (IdPs) 與 Athena 搭配使用的自訂擴充功能的支援可能會受到限制。
- 使用 JDBC 和 ODBC 驅動程式的聯合存取與 IAM Identity Center 信任身分傳播功能不相容。
- 目前，您無法使用 Athena 主控台來設定 IdP 和 SAML 與 Athena 搭配使用時的支援。若要設定此支援，請使用第三方身分提供者、Lake Formation 和 IAM 管理主控台，以及 JDBC 或 ODBC 驅動程式用戶端。
- 在設定身分提供者和 SAML 以與 Lake Formation 和 Athena 搭配使用之前，您應先了解 [SAML 2.0 規格](#) 以及其如何與您的身分提供者搭配使用。
- SAML 提供者以及 Athena JDBC 和 ODBC 驅動程式由第三方提供，因此透過與其使用相關問題 AWS 的支援可能會受到限制。

## 主題

- [教學課程：使用 Lake Formation 和 JDBC 設定 Okta 使用者對 Athena 的聯合存取](#)

教學課程：使用 Lake Formation 和 JDBC 設定 Okta 使用者對 Athena 的聯合存取

本教學課程說明如何設定 Okta AWS Lake Formation、AWS Identity and Access Management 權限和 Athena JDBC 驅動程式，以啟用以 SAML 為基礎的聯合使用 Athena。Lake Formation 為以 SAML 為基礎的使用者提供對 Athena 中可用資料的精細存取控制。若要設定此設定，本教學課程會使用 Okta 開發人員主控台、AWS IAM 和 Lake Formation 主控台，以及 SQL 工作台 /J 工具。

## 必要條件

此教學課程假設您已完成下列作業：

- 已建立 Amazon Web Services 帳戶。若要建立帳戶，請造訪 [Amazon Web Services 首頁](#)。
- 在 Amazon S3 中為 Athena [設定查詢結果位置](#)。
- 向 Lake Formation [註冊 Amazon S3 資料儲存貯體位置](#)
- 在 [AWS Glue 資料目錄](#) (指向您在 Amazon S3 中的資料) 上定義[資料庫](#)和[資料表](#)。
  - 如果您尚未定義資料表，請執行 [AWS Glue 爬行者程式](#)，或[使用 Athena 為您要存取的資料定義資料庫和一或多個表格](#)。
- 本教學課程使用以[紐約計程車行程資料集](#) (可於 [AWS上的開放資料登記處](#)取得) 為基礎的資料表。本教學課程會使用資料庫名稱 tripdb 和資料表名稱 nyctaxi。

## 教學步驟

- [步驟 1：建立 Okta 帳戶](#)
- [步驟 2：將使用者和群組新增至 Okta](#)
- [步驟 3：設定用於 SAML 身分驗證的 Okta 應用程式](#)
- [步驟 4：建立 AWS SAML 身分識別提供者和 Lake Formation 存取 IAM 角色](#)
- [步驟 5：將 IAM 角色和 SAML 身分提供者新增至 Okta 應用程式](#)
- [步驟 6：透過授予使用者和群組權限 AWS Lake Formation](#)
- [步驟 7：驗證透過 Athena JDBC 用戶端的存取](#)
- [結論](#)
- [相關資源](#)

### 步驟 1：建立 Okta 帳戶

本教學課程使用 Okta 作為以 SAML 為基礎的身分提供者。如果您還沒有 Okta 帳戶，您可以建立免費的帳戶。需要 Okta 帳戶，以便您可以建立用於 SAML 身分驗證的 Okta 應用程式。

#### 若要建立 Okta 帳戶

1. 若要使用 Okta，請導覽至 [Okta 開發人員註冊頁面](#)並建立免費的 Okta 試用帳戶。開發人員版服務免費提供，最高可達 Okta 在 [developer.okta.com/pricing](#) 指定的限制。
2. 收到啟用電子郵件時，請啟用您的帳戶。

Okta 網域名稱將會指派給您。儲存網域名稱以供參考。稍後，您可以在連線至 Athena 的 JDBC 字串中使用網域名稱 (`< okta-idp-domain >`)。

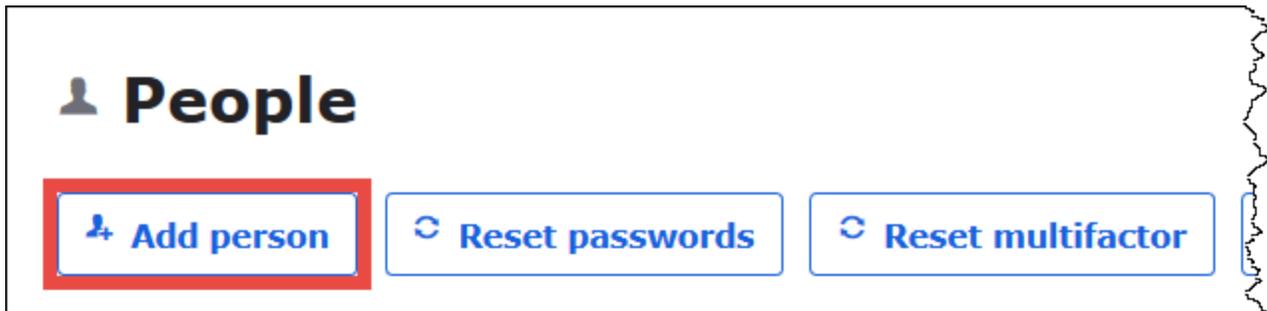
## 步驟 2：將使用者和群組新增至 Okta

在此步驟中，您可以使用 Okta 主控台來執行以下任務：

- 建立兩個 Okta 使用者。
- 建立兩個 Okta 群組。
- 在每個 Okta 群組中新增一個 Okta 使用者。

### 若要將使用者新增至 Okta

1. 啟用 Okta 帳戶之後，請以管理使用者身分登入指派的 Okta 網域。
2. 在左側導覽窗格中，選擇 Directory (目錄)，然後選擇 People (人員)。
3. 選擇 Add Person (新增人員)，新增一個將透過 JDBC 驅動程式存取 Athena 的新使用者。



4. 在 Add Person (新增人員) 對話方塊中，輸入所需資訊。
  - 輸入 First name (名字) 和 Last name (姓氏) 的值。本教學課程使用的是 *athena-okta-user*。
  - 輸入 Username (使用者名稱) 和 Primary email (主要電子郵件地址)。本教學課程使用 *athena-okta-user@anycompany.com*。
  - 對於 Password (密碼)，選擇 Set by admin (由管理員設定)，然後提供密碼。本教學課程會清除 User must change password on first login (使用者首次登入時必須變更密碼) 的選項；您的安全需求可能會有所不同。

## Add Person

User type <sup>?</sup>

User ▼

First name

athena-okta-user

Last name

athena-okta-user

Username

athena-okta-user@anycompany.com

Primary email

athena-okta-user@anycompany.com

Secondary email (optional)

Groups (optional)

Password <sup>?</sup>

Set by admin ▼

Enter password

User must change password on first login

Save

Save and Add Another

Cancel

5. 選擇 Save and Add Another (儲存並新增另一個)。
6. 輸入另一個使用者的資訊。此範例會新增商務分析師使用者 *athena-ba-user@anycompany .com*。

## Add Person

User type <sup>?</sup>

User ▼

First name

athena-ba-user

Last name

athena-ba-user

Username

athena-ba-user@anycompany.com

Primary email

athena-ba-user@anycompany.com

Secondary email (optional)

Groups (optional)

Password <sup>?</sup>

Set by admin ▼

Enter password

User must change password on first login

Save

Save and Add Another

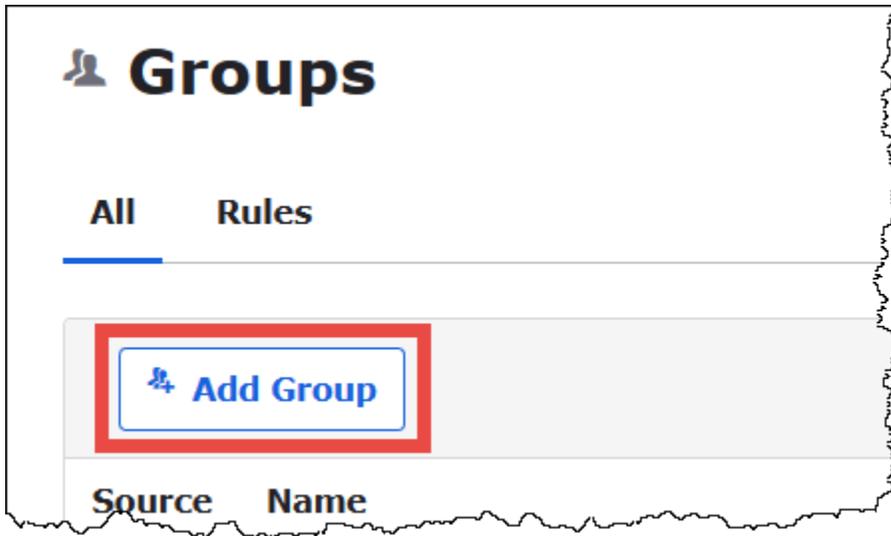
Cancel

## 7. 選擇儲存。

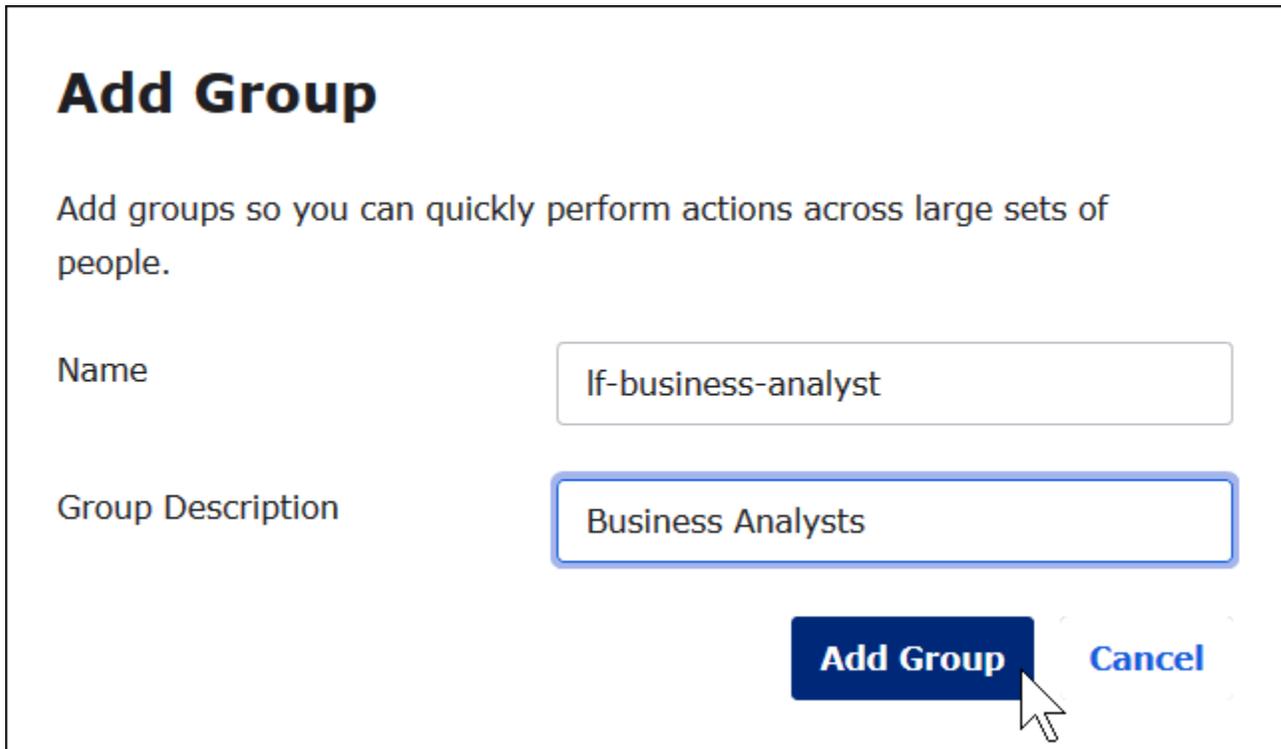
在下列程序中，您可以新增 "Business Analysts" (業務分析師) 群組和 "Developer" (開發人員) 群組，透過 Athena JDBC 驅動程式提供兩個 Okta 群組的存取權限。

若要新增 Okta 群組

1. 在 Okta 導覽窗格中選擇 Directory (目錄)，然後選擇 Groups (群組)。
2. 在 Groups (群組) 頁面上，選擇 Add Group (新增群組)。



3. 在 Add Group (新增群組) 對話方塊中，輸入所需資訊。
  - 針對名稱，輸入 *lf-business-analyst*。
  - 在 Group Description (群組描述) 中輸入 *Business Analysts* (業務分析師)。



## Add Group

Add groups so you can quickly perform actions across large sets of people.

Name

Group Description

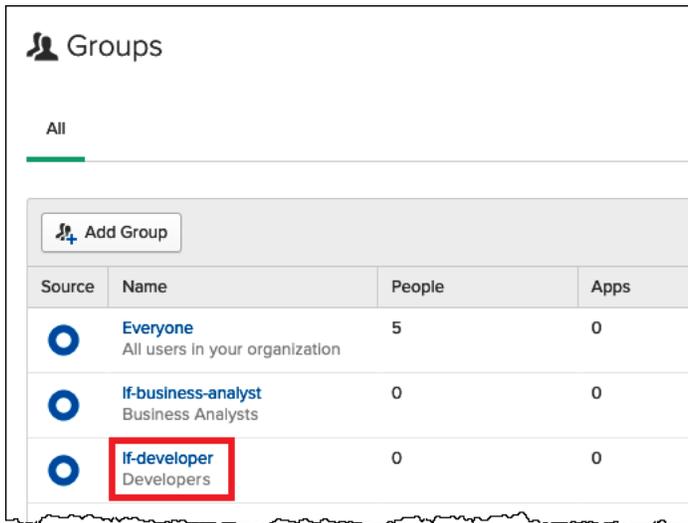
**Add Group** Cancel

4. 選擇 Add Group (新增群組)。
5. 在 Groups (群組) 頁面上，再次選擇 Add Group (新增群組)。現在輸入開發人員群組的資訊。
6. 輸入所需資訊。
  - 在 Name (名稱) 中輸入 *lf-developer*。
  - 在 Group Description (群組描述) 中輸入 *Developers* (開發人員)。
7. 選擇 Add Group (新增群組)。

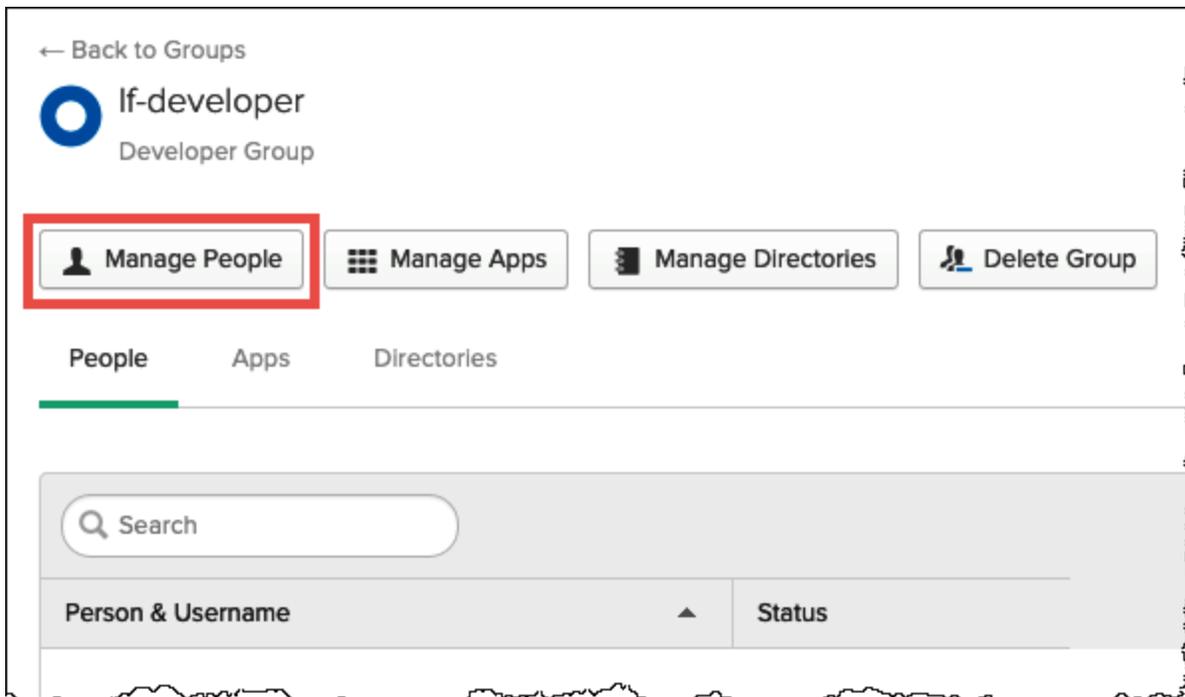
現在您已擁有兩個使用者和兩個群組，可以在每個群組中新增使用者。

若要將使用者新增至群組

1. 在 Groups (群組) 頁面上，選擇您剛建立的 lf-developer 群組。您需將您建立為開發人員的 Okta 使用者之一新增至此群組。



2. 選擇 Manage People (管理人員)。



3. 從「非成員」清單中選擇athena-okta-user。

← Back to Group

 **If-developer**  
Developers

---

### Add or remove people from the If-developer group

Search by person 

 **Not Members** Showing 1 - 4 of 4

Person & Username ▾

athena-ba-user athena-ba-user  
athena-ba-user@anycompany.com

**athena-okta-user athena-okta-user**    
athena-okta-user@anycompany.com

First Previous **1** Next Last

 **Members**

Person & Username ▲

First Previous Next Last

使用者的項目會從左側的 Not Members (非成員) 清單移動至右側的 Members (成員) 清單。

← Back to Group

**If-developer**  
Developers

Add or remove people from the If-developer group

Cancel Save

Search:  Person

**+ Add All** (3) **- Remove All** (1)

**Not Members** Showing 1 - 3 of 3

Person & Username

athena-ba-user athena-ba-user  
athena-ba-user@anycompany.com

First Previous **1** Next Last

**Members** Showing 1 - 1 of 1

Person & Username

athena-okta-user athena-okta-user  
athena-okta-user@anycompany.com

First Previous **1** Next Last

Cancel Save

4. 選擇儲存。
5. 選擇 Back to Group (返回群組)，或選擇 Directory (目錄)，接著選擇 Groups (群組)。
6. 選擇 If-business-analyst 群組。
7. 選擇 Manage People (管理人員)。
8. 新增 athena-ba-user 至 If-business-analyst 群組的 [成員] 清單，然後選擇 [儲存]。
9. 選擇 Back to Group (返回群組)，或依序選擇 Directory (目錄)、Groups (群組)。

Groups (群組) 頁面現在顯示每個群組都有一個 Okta 使用者。

Source	Name	People	Apps
	If-business-analyst Business Analyst	1	0
	If-developer Developer Group	1	0

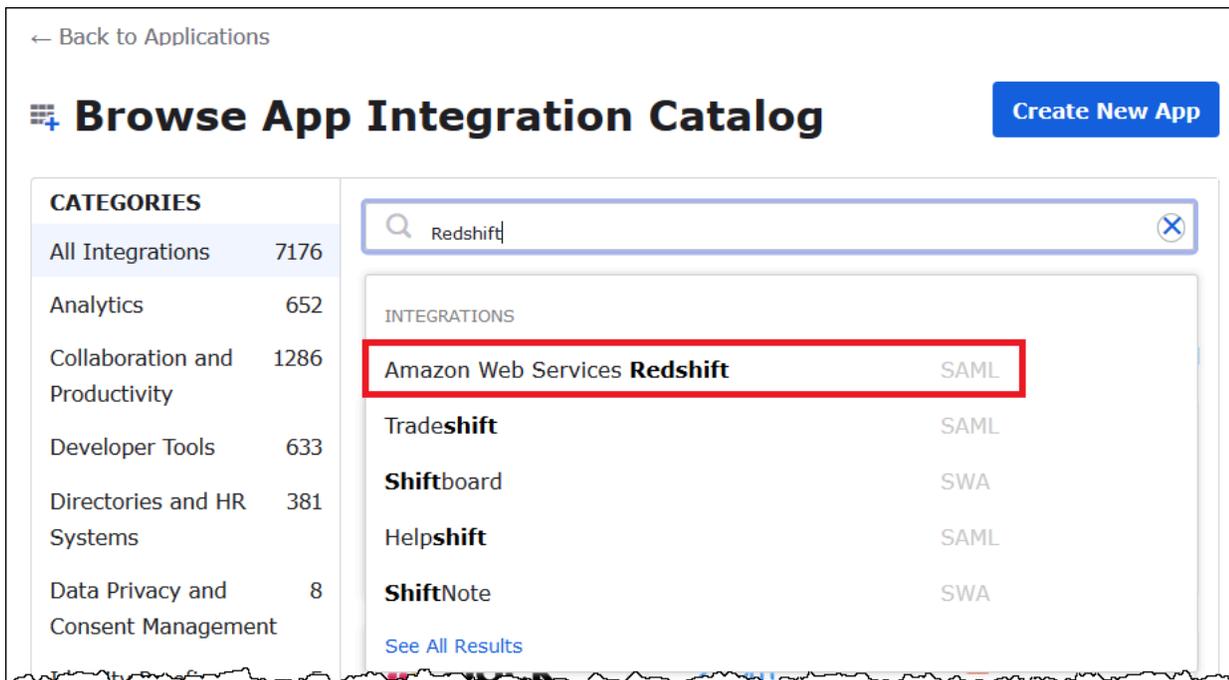
### 步驟 3：設定用於 SAML 身分驗證的 Okta 應用程式

在此步驟中，您需使用 Okta 開發人員主控台以執行下列任務：

- 新增要與 AWS 配合使用的 SAML 應用程式。
- 將應用程式指派給 Okta 使用者。
- 將應用程式指派給 Okta 群組。
- 下載產生的身分提供者中繼資料，以便稍後與 AWS 搭配使用。

若要新增應用程式以進行 SAML 身分驗證

1. 在 Okta 導覽窗格中，選擇 Applications (應用程式)、Applications (應用程式)，以便您可設定 Okta 應用程式以進行對 Athena 的 SAML 身分驗證。
2. 按一下 Browse App Catalog (瀏覽應用程式目錄)。
3. 在搜尋方塊中，輸入 **Redshift**。
4. 選擇 Amazon Web Services Redshift。本教學課程中的 Okta 應用程式對 Amazon Redshift 使用現有的 SAML 整合。



5. 在 Amazon Web Services Redshift 頁面上，選擇 Add (新增) 來為 Amazon Redshift 建立以 SAML 為基礎的應用程式。

← Back to Add Application

# Amazon Web Services Redshift

**Overview** Capabilities



**Add**

## Overview

Okta's integration with Amazon Web Services (AWS) Redshift allows end users to authenticate to AWS Redshift accounts using single sign-on with SAML. Once SAML SSO is configured you can use SQL client tools such as SQL Workbench/J to connect to redshift directly from the application.

CATEGORIES

[Security](#)

- 在 Application (應用程式) 標籤中輸入 Athena-LakeFormation-Okta，然後選擇 Done (完成)。

# Add Amazon Web Services Redshift

## 1 General Settings

### General Settings - Required

Application label

Athena-LakeFormation-Okta

This label displays under the app on your home page

Application Visibility

Do not display application icon to users

Do not display application icon in the Okta Mobile App

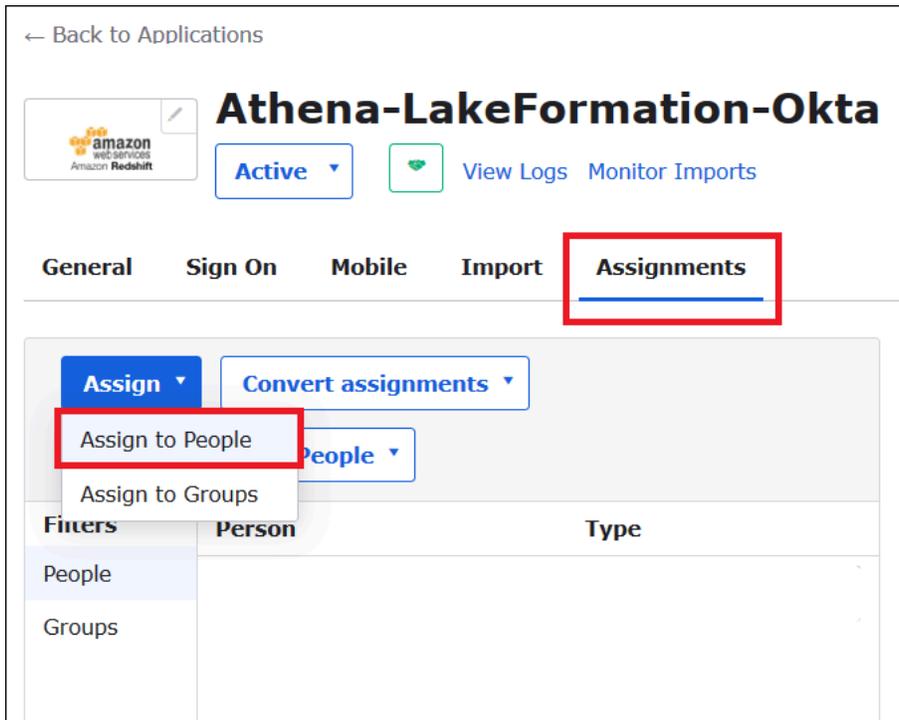
Cancel

Done

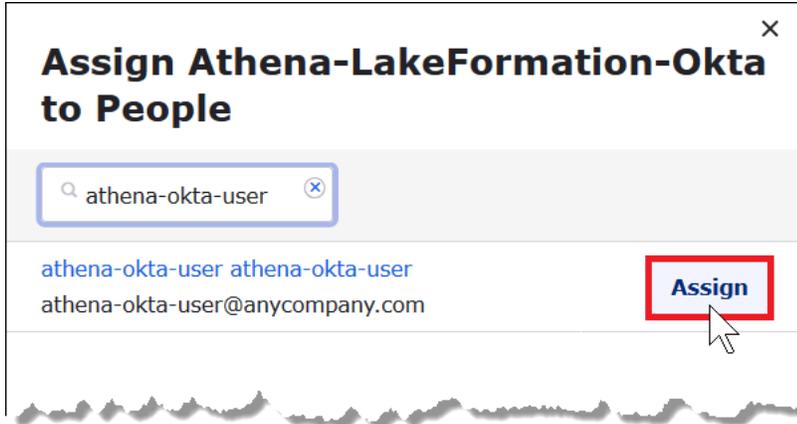
現在您已建立 Okta 應用程式，您可以將其指派給您建立的使用者和群組。

若要將應用程式指派給使用者和群組

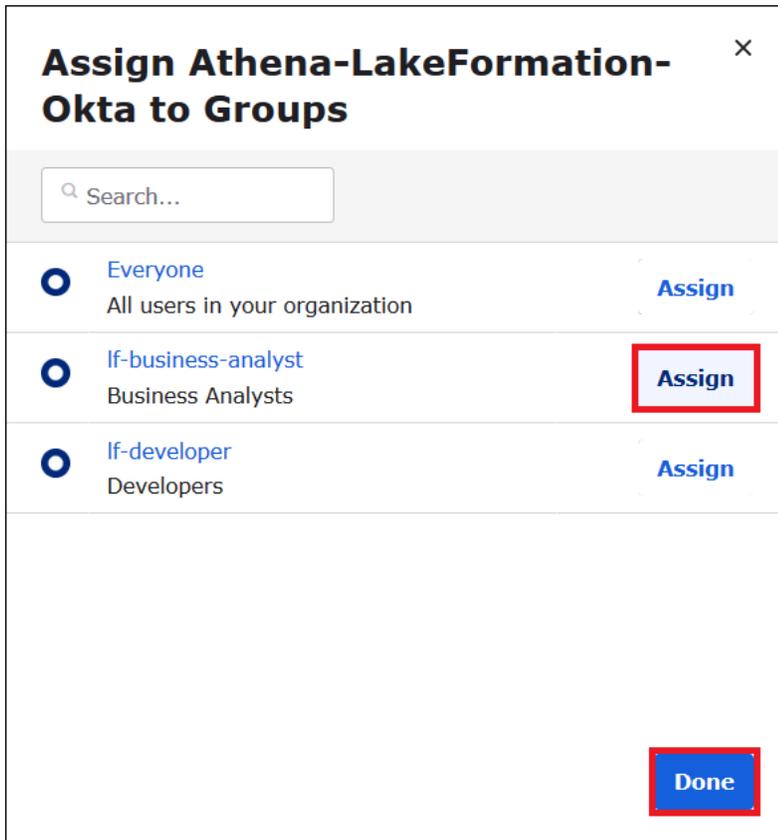
1. 在 [應用程式] 頁面上，選擇 Athena LakeFormation-Okta 應用程式。
2. 在 Assignments (指派) 索引標籤上，依序選擇 Assign (指派)、Assign to People (指派給人員)。



3. 在「將 Athena-LakeFormation-Okta 指派給人員」對話方塊中，找到您先前建立的athena-okta-user使用者。
4. 選擇 Assign (指派) 將使用者指派給應用程式。



5. 選擇 Save and Go Back (儲存並返回)。
6. 選擇 Done (完成)。
7. 在 Athena-LakeFormation Okta 應用程式的 [指派] 索引標籤上，選擇 [指派] > [指派給群組]。
8. 對於 If-business-analyst，選擇 [指派] 將 Athena LakeFormation-Okta 應用程式指派給If-business-analyst群組，然後選擇 [完成]。



該群組會出現在應用程式的群組清單中。

← Back to Applications

# Athena-LakeFormation-Okta

amazon web-services Amazon Redshift

Active View Logs Monitor Imports

General Sign On Mobile Import **Assignments**

Assign Convert assignments

Search... Groups

Filters	Priority	Assignment
People	1	If-business-analyst
Groups		Business Analysts

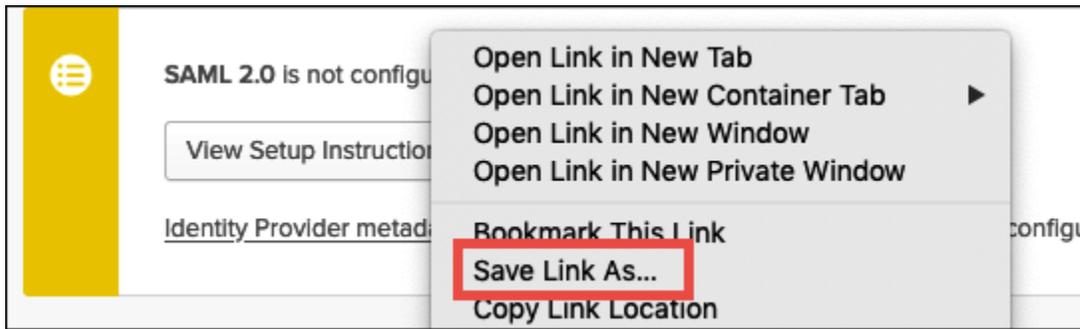
現在您已準備好下載身分提供者應用程式中繼資料，以與 AWS 搭配使用。

若要下載應用程式中繼資料

1. 選擇 Okta 應用程式 Sign On (登入) 索引標籤，然後以滑鼠右鍵按一下 Identity Provider metadata (身分提供者中繼資料)。

The screenshot shows the 'Sign On' configuration page in the AWS IAM console. At the top, the application name 'Athena-LakeFormation-Okta' is displayed, along with an 'Active' status and links for 'View Logs' and 'Monitor Imports'. Below this are navigation tabs for 'General', 'Sign On', 'Mobile', 'Import', and 'Assignments', with 'Sign On' highlighted. The main content area is titled 'Settings' and includes an 'Edit' button. Under the 'Sign on methods' section, 'SAML 2.0' is selected. A configuration box shows 'Default Relay State' and 'Attributes (Optional)'. A yellow warning banner states that SAML 2.0 is not configured until setup instructions are completed, with a 'View Setup Instructions' button. Below the banner, a red box highlights the 'Identity Provider metadata' link, which is described as available if the application supports dynamic configuration.

2. 選擇 Save Link As (另存連結)，將 XML 格式的身分提供者中繼資料儲存至檔案中。指定一個您認識的名稱 (例如 Athena-LakeFormation-idp-metadata.xml)。



#### 步驟 4：建立 AWS SAML 身分識別提供者和 Lake Formation 存取 IAM 角色

在此步驟中，您可以使用 AWS Identity and Access Management (IAM) 主控台來執行下列工作：

- 為 AWS 建立身分提供者。
- 為 Lake Formation 存取建立 IAM 角色。
- 將 AmazonAthenaFullAccess 受管理的原則新增至角色。
- 為 Lake Formation 和角色添 AWS Glue 加政策。
- 將 Athena 查詢結果的政策新增至該角色。

#### 若要建立 AWS SAML 身分識別提供者

1. 以 Amazon Web Services 帳戶管理員身分登入 Amazon Web Services 帳戶主控台，並導覽至 IAM 主控台 (<https://console.aws.amazon.com/iam/>)。
2. 在導覽窗格中，選擇 Identity providers (身分提供者)，然後按一下 Add provider (新增供應商)。
3. 在 Configure provider (設定供應商) 畫面上，輸入以下資訊：
  - 對於 Provider type (供應商類型)，選擇 SAML。
  - 對於 Provider name (供應商名稱)，輸入 AthenaLakeFormationOkta。
  - 對於 Metadata document (中繼資料文件)，使用 Choose file (選擇檔案) 選項，以上傳您下載的身分提供者 (IdP) 中繼資料 XML 檔案。
4. 選擇 Add provider (新增供應商)。

接下來，您建立用於 AWS Lake Formation 存取的 IAM 角色。您可以將兩個內嵌政策新增至該角色。一個策略提供訪問 Lake Formation 和 AWS Glue API 的權限。另一個政策可用於存取 Athena 和在 Amazon S3 中的 Athena 查詢結果位置。

## 若要建立 AWS Lake Formation 存取的 IAM 角色

1. 在 IAM 主控台導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
2. 在 Create role (建立角色) 頁面上，執行以下步驟：

**Create role**

1 2 3 4

Select type of trusted entity

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

**SAML 2.0 federation**  
Your corporate directory

Allows users that are federated with SAML 2.0 to assume this role to perform actions in your account. [Learn more](#)

Choose a SAML 2.0 provider

If you're creating a role for API access, choose an Attribute and then type a Value to include in the role. This restricts access to users with the specified attributes.

**SAML provider** AthenaLakeFormationOkta

[Create new provider](#) [Refresh](#)

Allow programmatic access only

Allow programmatic and AWS Management Console access

**Attribute** SAML:aud

**Value\*** https://signin.aws.amazon.com/saml

**Condition** [Add condition \(optional\)](#)

\* Required [Cancel](#) [Next: Permissions](#)

- a. 對於 Select type of trusted entity (選取信任的實體類型)，選擇 SAML 2.0 Federation (SAML 2.0 聯合)。
  - b. 對於 SAML 提供者，請選取 AthenaLakeFormationOkta。
  - c. 對於 SAML 提供者，請選取 [允許程式設計和 AWS Management Console 存取] 選項。
  - d. 選擇 Next: Permissions (下一步：許可)。
3. 在 Attach Permissions policies (連接許可政策) 頁面上，對於 Filter policies (篩選條件政策)，輸入 **Athena**。
  4. 選取受 AmazonAthenaFullAccess 管理的策略，然後選擇 [下一步:標記]。

## Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy ↻

Filter policies  Showing 2 results

	Policy name	Used as
<input checked="" type="checkbox"/>	▶ AmazonAthenaFullAccess	Permissions policy (3)
<input type="checkbox"/>	▶ AWSQuicksightAthenaAccess	None

▶ Set permissions boundary

\* Required Cancel Previous Next: Tags

5. 在 Add tags (新增標籤) 頁面上，選擇 Next: Review (下一步：檢閱)。
6. 在 [檢閱] 頁面上，對於角色名稱，輸入角色的名稱 (例如，*Athena-LakeFormation-OktaRole*)，然後選擇 [建立角色]。

## Create role

1 2 3 4

### Review

Provide the required information below and review this role before you create it.

**Role name\*** Athena-LakeFormation-OktaRole  
Use alphanumeric and '+, @, \_' characters. Maximum 64 characters.

**Role description**  
Maximum 1000 characters. Use alphanumeric and '+, @, \_' characters.

**Trusted entities** The identity provider(s) arn:aws:iam:::saml-provider/AthenaLakeFormationOkta

**Policies**  AmazonAthenaFullAccess [↗](#)

**Permissions boundary** Permissions boundary is not set

No tags were added.

\* Required Cancel Previous Create role

接下來，您可以新增內嵌政策，以便在 Amazon S3 中存取 Lake Formation、AWS Glue API 和 Athena 查詢結果。

每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

為 Lake Formation 的角色添加內嵌政策 AWS Glue

1. 從 IAM 主控台的角色清單中，選擇新建立的 Athena-LakeFormation-OktaRole。
2. 在角色的 Summary (摘要) 頁面之 Permissions (許可) 索引標籤上，選擇 Add inline policy (新增內嵌政策)。
3. 在 Create policy (建立政策) 頁面上，選擇 JSON。
4. 新增內嵌政策 (如下所示)，該政策提供對 Lake Formation 和 AWS Glue API 的存取權限。

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

```

        "lakeformation:GetDataAccess",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue>CreateDatabase",
        "glue:GetUserDefinedFunction",
        "glue:GetUserDefinedFunctions"
    ],
    "Resource": "*"
}
}

```

5. 選擇 Review policy (檢閱政策)。
6. 對於 Name (名稱)，輸入政策名稱 (例如 **LakeFormationGlueInlinePolicy**)。
7. 選擇 Create policy (建立政策)。

若要針對 Athena 查詢結果位置將內嵌政策新增至角色

1. 在 Athena-LakeFormation-OktaRole 角色的 Summary (摘要) 頁面之 Permissions (許可) 索引標籤上，選擇 Add inline policy (新增內嵌政策)。
2. 在 Create policy (建立政策) 頁面上，選擇 JSON。
3. 新增內嵌政策 (如下所示)，該政策允許對 Athena 查詢結果位置進行角色存取。將範例中的 *<athena-query-results-bucket>* 預留位置取代為 Amazon S3 儲存貯體的名稱。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AthenaQueryResultsPermissionsForS3",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::<athena-query-results-bucket>",
        "arn:aws:s3:::<athena-query-results-bucket>/*"
      ]
    }
  ]
}

```

```
]
}
```

4. 選擇 Review policy (檢閱政策)。
5. 對於 Name (名稱)，輸入政策名稱 (例如 **AthenaQueryResultsInlinePolicy**)。
6. 選擇 Create policy (建立政策)。

接下來，您要複製 Lake Formation 存取角色的 ARN 和您建立的 SAML 供應商的 ARN。在本教學課程的下一節中設定 Okta SAML 應用程式時，這些都是必要的程序。

若要複製角色 ARN 和 SAML 身分提供者 ARN

1. 在 IAM 主控台中，在 Athena-LakeFormation-OktaRole 角色的 Summary (摘要) 頁面上，選擇 Role ARN (角色 ARN) 旁的 Copy to clipboard (複製至剪貼簿) 圖示。ARN 採用下列格式：

```
arn:aws:iam::<account-id>:role/Athena-LakeFormation-OktaRole
```

2. 安全地儲存完整的 ARN 以供日後參考。
3. 在 IAM 主控台導覽窗格中，選擇 Identity providers (身分提供者)。
4. 選擇 AthenaLakeFormationOkta 提供者。
5. 在 Summary (摘要) 頁面上，選擇 Provider ARN (供應商 ARN) 旁的 Copy to clipboard (複製至剪貼簿) 圖示。ARN 看起來應該如下所示：

```
arn:aws:iam::<account-id>:saml-provider/AthenaLakeFormationOkta
```

6. 安全地儲存完整的 ARN 以供日後參考。

步驟 5：將 IAM 角色和 SAML 身分提供者新增至 Okta 應用程式

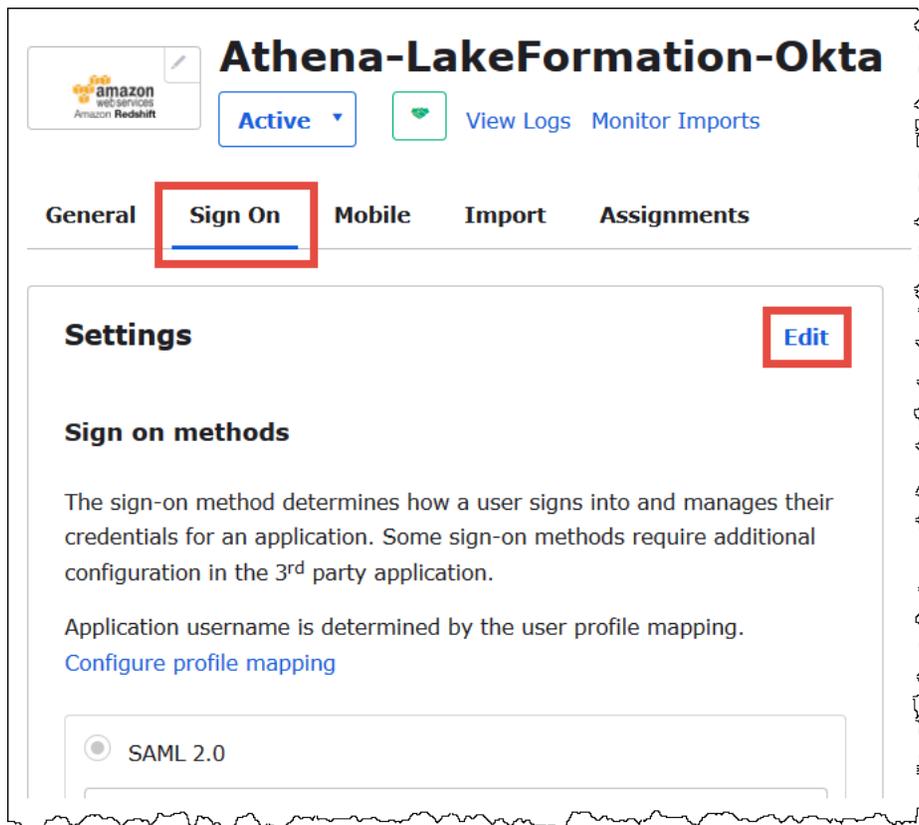
在此步驟中，您需返回 Okta 開發人員主控台並執行下列任務：

- 將使用者和群組的 Lake Formation URL 屬性新增至 Okta 應用程式。
- 將身分提供者的 ARN 和 IAM 角色的 ARN 新增至 Okta 應用程式。
- 複製 Okta 應用程式 ID。連接至 Athena 的 JDBC 設定檔中需要 Okta 應用程式 ID。

若要將使用者和群組的 Lake Formation URL 屬性新增至 Okta 應用程式

1. 登入 Okta 開發人員主控台。

2. 選擇 Applications (應用程式) 索引標籤，然後選擇 Athena-LakeFormation-Okta 應用程式。
3. 選擇應用程式的 Sign On (登入) 索引標籤，然後選擇 Edit (編輯)。



4. 選擇 Attributes (optional) (屬性 (選填)) 以將其展開。

**Athena-LakeFormation-Okta**

Active View Logs Monitor Imports

General **Sign On** Mobile Import Assignments

### Settings

**Sign on methods**

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3<sup>rd</sup> party application.

Application username is determined by the user profile mapping.  
[Configure profile mapping](#)

SAML 2.0 is the only sign-on option currently supported for this application.

SAML 2.0

Default Relay State  
 All IDP-initiated requests will include this RelayState.

Attributes (Optional) [Learn More](#)

Name	Name format (optional)	Value
http:	Unspecified	user.login

[Add Another](#)

5. 對於 Attribute Statements (optional) (屬性陳述式 (選填))，新增下列屬性：

- 對於 Name (名稱)，輸入 **https://lakeformation.amazon.com/SAML/Attributes/Username**。

- 對於 Value (值)，輸入 **user.login**
6. 在 Group Attribute Statements (optional) (群組屬性陳述式 (選填)) 下，新增下列屬性：
- 對於 Name (名稱)，輸入 **https://lakeformation.amazonaws.com/SAML/Attributes/Groups**。
  - 對於 Name format (名稱格式)，輸入 **Basic**
  - 對於 Filter (篩選條件)，選擇 Matches regex (符合 regex)，然後在篩選條件方塊中輸入 **.\***。

The screenshot shows the SAML 2.0 configuration interface. It includes a section for "Attributes (Optional)" with a table for "Attribute Statements (optional)". The table has columns for "Name", "Name format (optional)", and "Value". One entry is shown with "http:", "Unspecified", and "user.login". Below this is an "Add Another" button. A red box highlights the "Group Attribute Statements (optional)" section, which has columns for "Name", "Name format (optional)", and "Filter". One entry is shown with "https://la", "Basic", "Matches regex", and ".\*". Below this is another "Add Another" button. At the bottom of the configuration area is a "Preview SAML" button.

Name	Name format (optional)	Value
http:	Unspecified	user.login

Name	Name format (optional)	Filter
https://la	Basic	Matches regex

7. 向下捲動至 Advanced Sign-On Settings (進階登入設定) 區段，您可以在此將身分提供者和 IAM 角色 ARN 新增至 Okta 應用程式。

若要將身分提供者和 IAM 角色的 ARN 新增至 Okta 應用程式

1. <saml-arn><role-arn>對於 Idp ARN 和角色 ARN，請以逗號分隔值的格式輸入 AWS 身分識別提供者 ARN 和角色 ARN。組合的字串看起來應該如下所示：

```
arn:aws:iam::<account-id>:saml-provider/  
AthenaLakeFormationOkta,arn:aws:iam::<account-id>:role/Athena-LakeFormation-  
OktaRole
```

## Advanced Sign-on Settings

These fields may be required for a Amazon Web Services Redshift proprietary sign-on option or general setting.

**Idp ARN and Role ARN**

Enter your AWS IDP ARN and Role ARN as comma separated values (e.g. "arn:aws:iam::1234567890:saml-provider/OKTA,arn:aws:iam::1234567890:role/SAML\_ROLE").

**Session Duration**

Get the user data location in cond...

since this app is using SAML with no password.

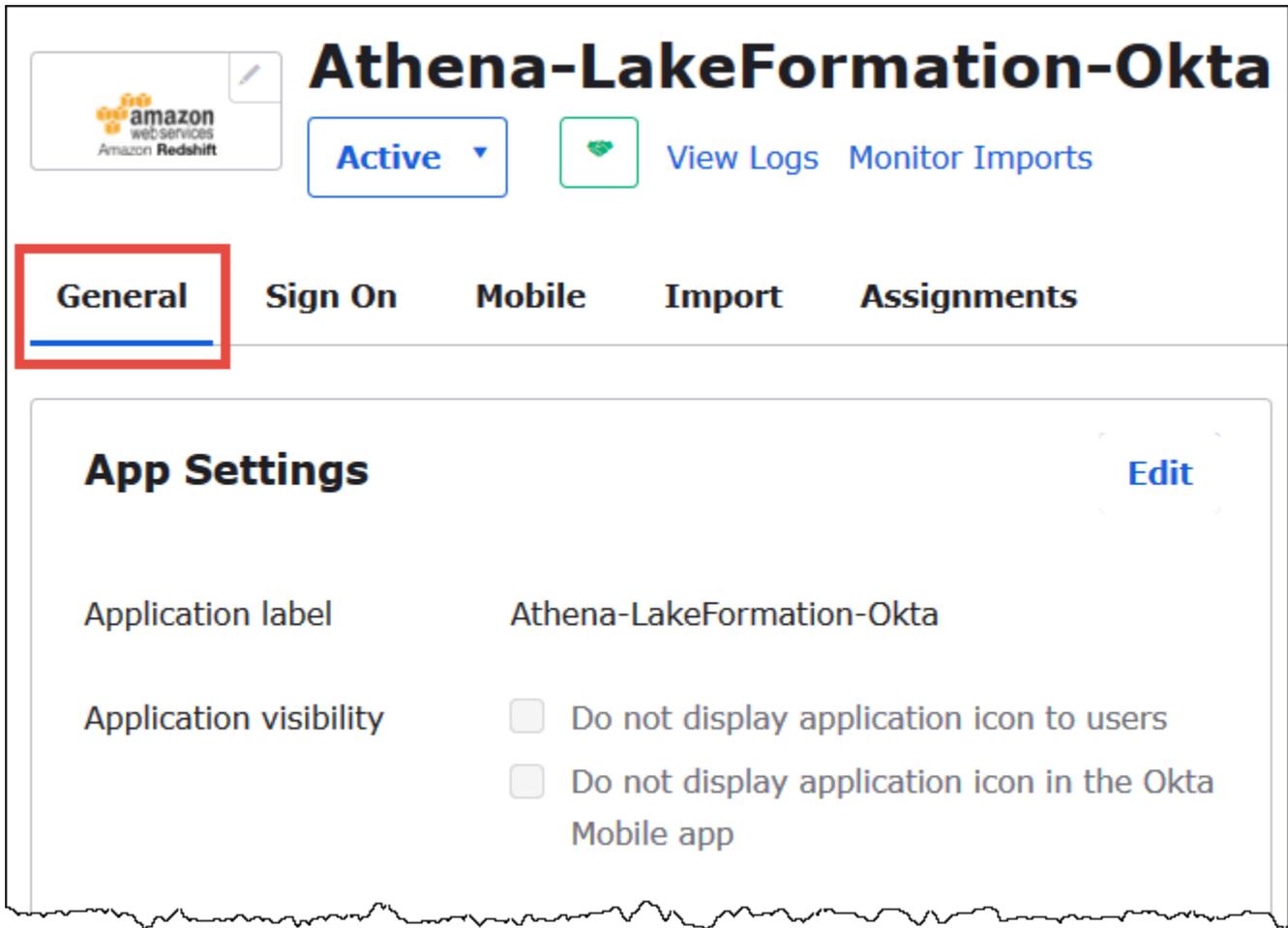
**Save**

## 2. 選擇儲存。

接下來，複製 Okta 應用程式 ID。您稍後需要此 ID，以用於連接至 Athena 的 JDBC 字串。

若要尋找並複製 Okta 應用程式 ID

### 1. 選擇 Okta 應用程式的 General (一般) 索引標籤。



**Athena-LakeFormation-Okta**

amazon web services Amazon Redshift

Active View Logs Monitor Imports

**General** Sign On Mobile Import Assignments

**App Settings** Edit

Application label Athena-LakeFormation-Okta

Application visibility  Do not display application icon to users  
 Do not display application icon in the Okta Mobile app

2. 向下捲動至 App Embed Link (應用程式內嵌連結) 區段。
3. 從 Embed Link (內嵌連結)，複製並安全地儲存 URL 的 Okta 應用程式 ID 部分。Okta 應用程式 ID 是 URL 中在 `amazon_aws_redshift/` 之後、下一個正斜線之前的部分。例如，如果 URL 包含 `amazon_aws_redshift/aaa/bbb`，則應用程式 ID 為 `aaa`。

**Note**

無法使用嵌入的連結直接登入 Athena 主控台以查看資料庫。僅當您使用 JDBC 或 ODBC 驅動程式向 Athena 提交查詢時，才能識別 SAML 使用者和群組的 Lake Formation 許可。若要查看資料庫，您可以使用 SQL Workbench/J 工具，其使用 JDBC 驅動程式連接至 Athena。在 [步驟 7：驗證透過 Athena JDBC 用戶端的存取](#) 中有 SQL Workbench/J 工具的說明。

**步驟 6：透過授予使用者和群組權限 AWS Lake Formation**

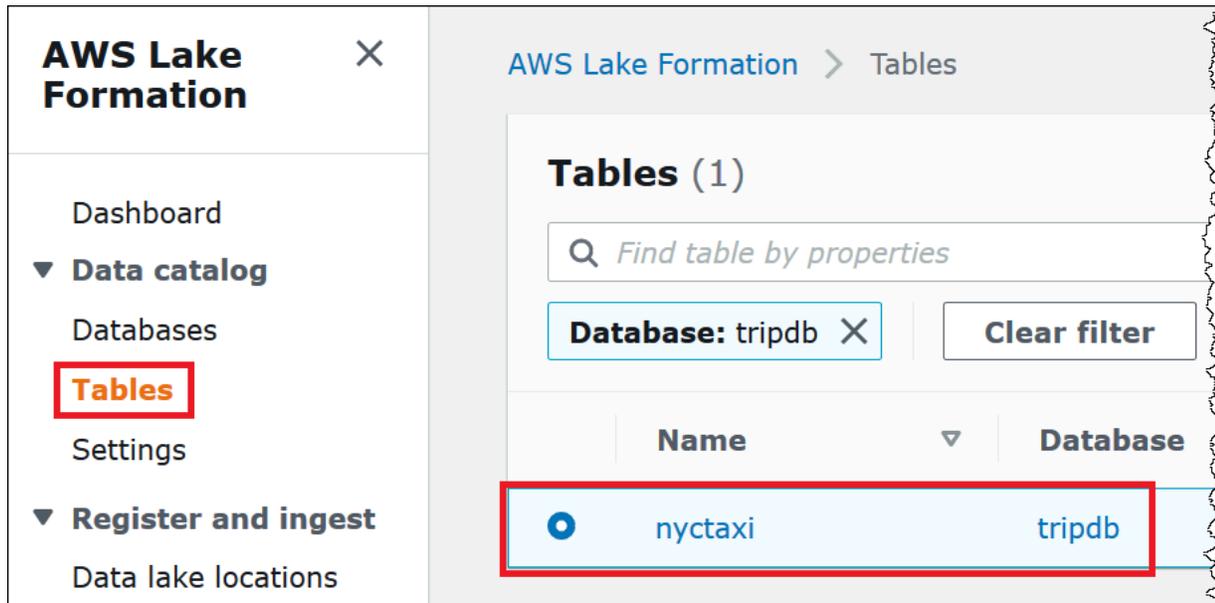
在此步驟中，您可以使用 Lake Formation 主控台，將資料表的許可授予 SAML 使用者和群組。執行以下任務：

- 指定 Okta SAML 使用者的 ARN 和資料表的相關使用者許可。
- 指定 Okta SAML 群組的 ARN 和資料表的相關群組許可。
- 確認您授予的許可。

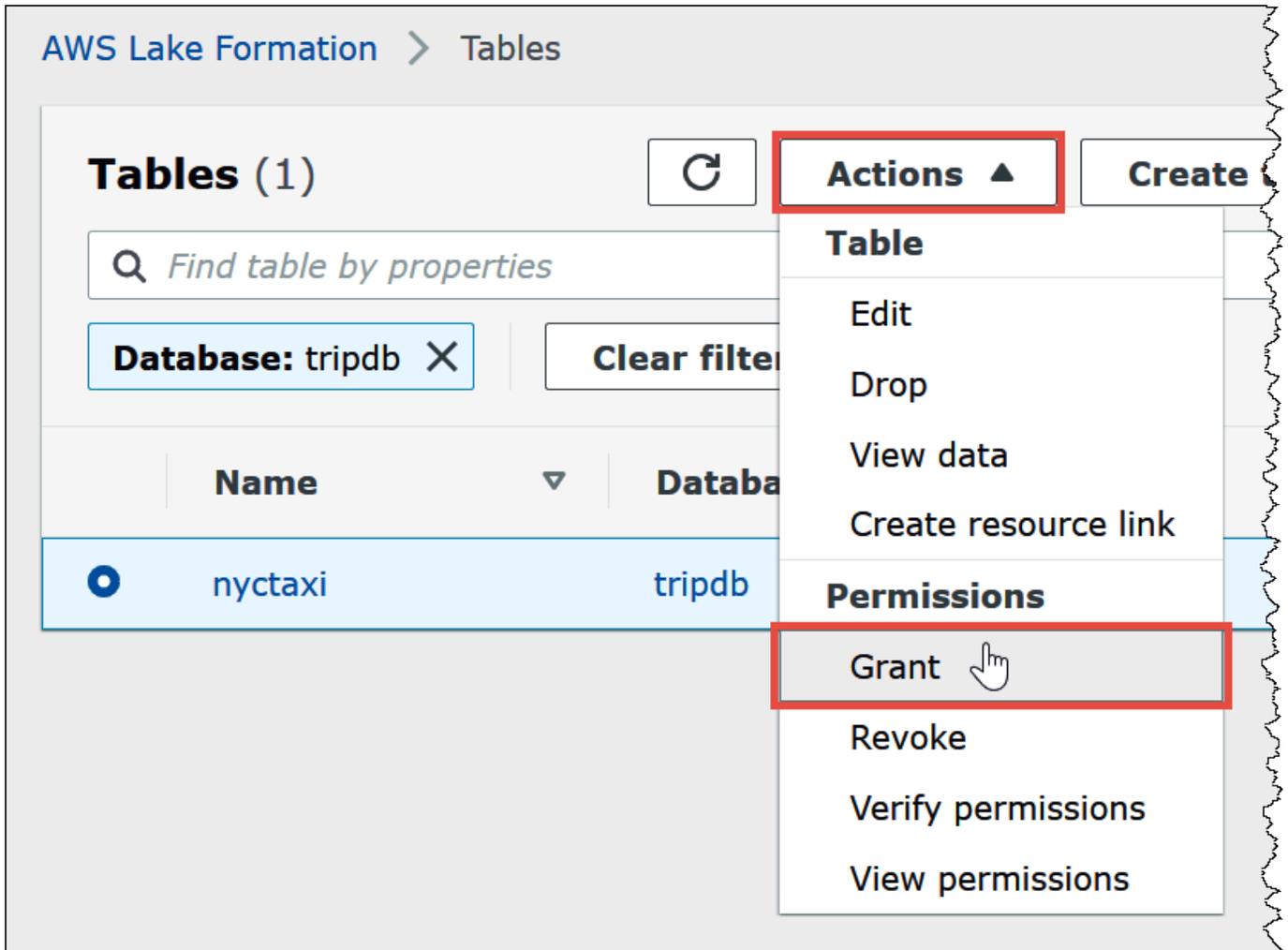
若要在 Lake Formation 中為 Okta 使用者授予許可

1. 以資料湖管理員身分登入 AWS Management Console。

2. 開啟 Lake Formation 主控台，網址為 <https://console.aws.amazon.com/lakeformation/>。
3. 從導覽窗格中，選擇 Tables (資料表)，然後選取您要為其授予許可的資料表。本教學課程使用來自 tripdb 資料庫的 nyctaxi 資料表。



4. 從 Actions (動作)，選擇 Grant (授予)。



5. 在 Grant permissions (授予許可) 對話方塊中，輸入下列資訊：

- a. 在 SAML 和 Amazon 使用 QuickSight 使用者和群組下，以下列格式輸入 Okta SAML 使用者 ARN：

```
arn:aws:iam::<account-id>:saml-provider/AthenaLakeFormationOkta:user/<athena-okta-user>@<anycompany.com>
```

- b. 對於 Columns (資料欄)、Choose filter type (選擇篩選條件類型)，選擇性地選擇 Include columns (包含資料欄) 或 Exclude columns (排除資料欄)。
- c. 使用篩選條件下的 Choose one or more columns (選擇一或多個資料欄) 下拉式清單，以指定您要包含或排除使用者的資料欄。
- d. 對於 Table permissions (資料表許可)，選擇 Select (選取)。本教學課程僅授予 SELECT 許可；您的需求可能會有所不同。

## 6. 選擇 Grant (授予)。

現在再針對 Okta 群組執行類似的步驟。

若要在 Lake Formation 中為 Okta 群組授予許可

1. 在 Lake Formation 主控台的 Tables (資料表) 頁面上，確定仍選取 nyctaxi 資料表。
2. 從 Actions (動作)，選擇 Grant (授予)。
3. 在 Grant permissions (授予許可) 對話方塊中，輸入下列資訊：
  - a. 在 SAML 和 Amazon QuickSight 使用者和群組下，以下列格式輸入 Okta SAML 群組 ARN：

```
arn:aws:iam::<account-id>:saml-provider/AthenaLakeFormationOkta:group/lf-business-analyst
```

- b. 對於 Columns (資料欄)、Choose filter type (選擇篩選條件類型)，選擇 Include columns (包含資料欄)。
- c. 對於 Choose one or more columns (選擇一或多個資料欄)，選擇資料表的前三個資料欄。

- d. 對於 Table permissions (資料表許可)，選擇要授予的特定存取許可。本教學課程僅授予 SELECT 許可；您的需求可能會有所不同。

My account  
User or role from this AWS account.

External account  
AWS account or AWS organization outside of my account.

**IAM users and roles**  
Add one or more IAM users or roles.  
Choose IAM principals to add

**SAML and Amazon QuickSight users and groups**  
Enter a SAML user or group ARN or Amazon QuickSight ARN. Press Enter to add additional ARNs.  
:saml-provider/AthenaLakeFormationOkta:group/lf-business-analyst

**Columns - optional**  
Choose filter type  
Include columns

**Include columns**  
Grant permissions to access the selected columns.  
Choose one or more columns

vendorid × lpep\_pickup\_datetime × lpep\_dropoff\_datetime ×  
bigint string string

**Table permissions**  
Choose the specific access permissions to grant.  
 Alter  Insert  Drop  Delete  Select

Super  
This permission is the union of the individual permissions above and supersedes them. [See here](#)

**Grantable permissions**  
Choose the permissions that may be granted to others.  
 Alter  Insert  Drop  Delete  Select

Super  
This permission allows the principal to grant any of the above permissions and supersedes those grantable permissions.

Cancel Grant

4. 選擇 Grant (授予)。
5. 若要確認您授予的許可，請選擇 Actions (動作)、View permissions (檢視許可)。

AWS Lake Formation > Tables

**Tables (1)** ↻ **Actions** ▲ Create

🔍 Find table by properties

**Database:** tripdb ✕ Clear filter

Name	Database
<input checked="" type="radio"/> nyctaxi	tripdb

- Table
  - Edit
  - Drop
  - View data
  - Create resource link
- Permissions
  - Grant
  - Revoke
  - Verify permissions
  - View permissions**

nyctaxi表格的 [資料權限] 頁面會顯示athena-okta-user和lf-business-analyst群組的權限。

**Data permissions (10)**  
Choose a database or table for which to review, grant or revoke user permissions.

🔍 Find by properties

**Database:** tripdb ✕ **Table:** nyctaxi ✕ Clear filter

Principal	Principal type	Resource type	Resource	Permissions
<input type="radio"/> lf-business-analyst	AD group	Column	Include: tripdb.nyctaxi. [lpep_dropoff_dateti me, lpep_pickup_datetim e, vendorid]	Select
<input type="radio"/> athena-okta- user@anycompany .com	AD user	Column	tripdb.nyctaxi.*	Select

## 步驟 7：驗證透過 Athena JDBC 用戶端的存取

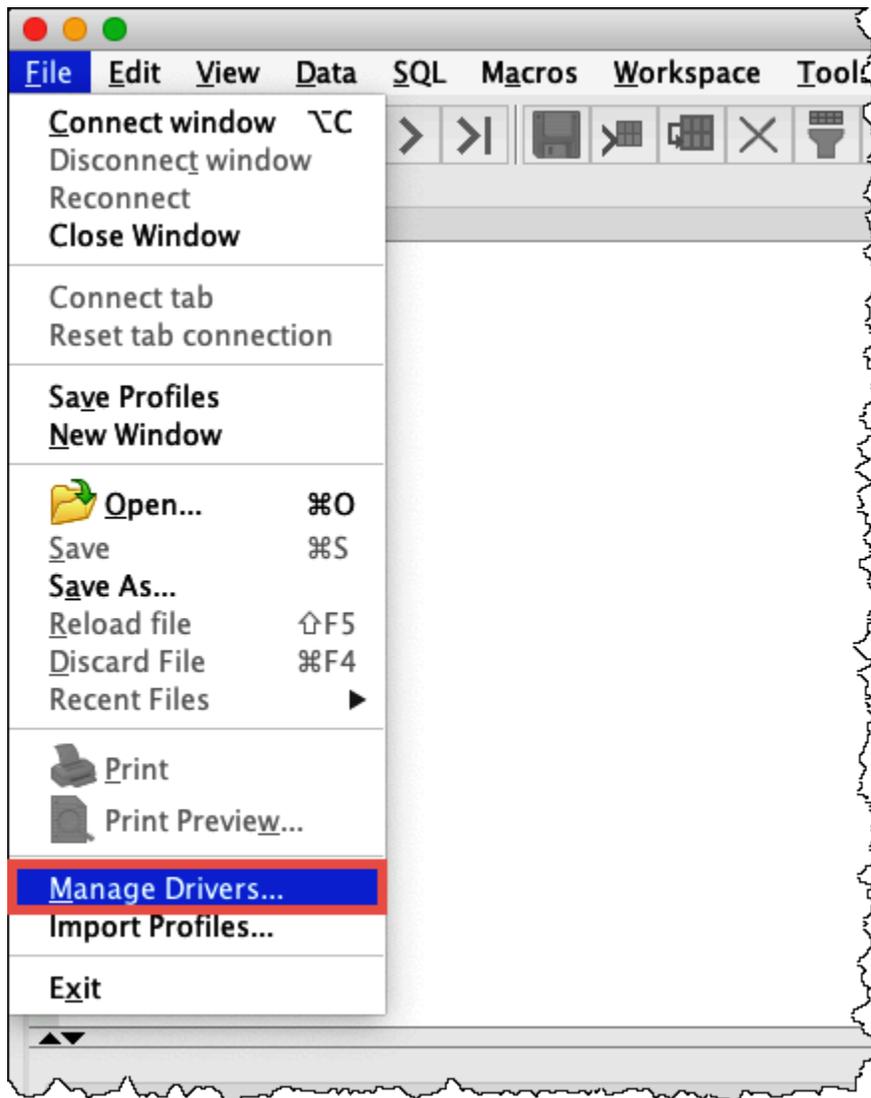
現在，您已經準備好使用 JDBC 用戶端，以 Okta SAML 使用者身分執行與 Athena 的測試連接。

在本節中，您需執行下列任務：

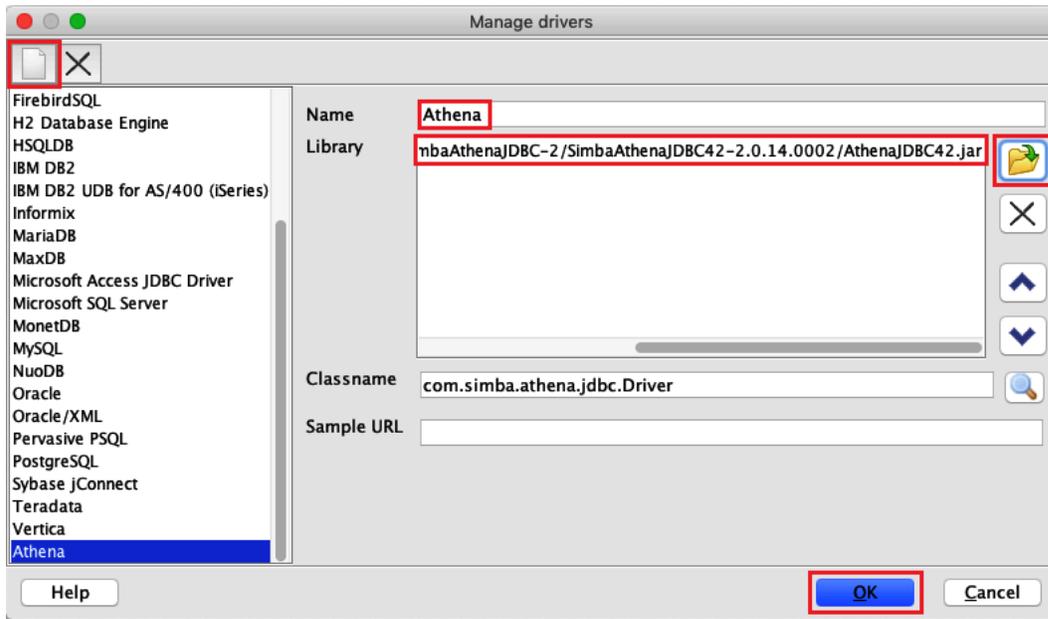
- 準備測試用戶端 – 下載 Athena JDBC 驅動程式，安裝 SQL Workbench，然後將驅動程式新增至 Workbench。本教學課程使用 SQL Workbench 透過 Okta 身分驗證來存取 Athena，並驗證 Lake Formation 許可。
- 在 SQL Workbench 中：
  - 為 Athena Okta 使用者建立連線。
  - 以 Athena Okta 使用者身分執行測試查詢。
  - 為業務分析師使用者建立並測試連線。
- 在 Okta 主控台中，將業務分析師使用者新增至開發人員群組。
- 在 Lake Formation 主控台中，設定開發人員群組的資料表許可。
- 在 SQL Workbench 中，以業務分析師使用者身分執行測試查詢，並確認許可變更如何影響結果。

### 若要準備測試用戶端

1. 從 [使用 JDBC 連接到 Amazon Athena](#) 下載並解壓縮 Lake Formation 相容的 Athena JDBC 驅動程式 (2.0.14 或更高版本)。
2. 下載並安裝免費的 [SQL Workbench/J](#) SQL 查詢工具 (可在修改後的 Apache 2.0 授權下取得)。
3. 在 SQL Workbench 中，選擇 File (檔案)，然後選擇 Manage Drivers (管理驅動程式)。



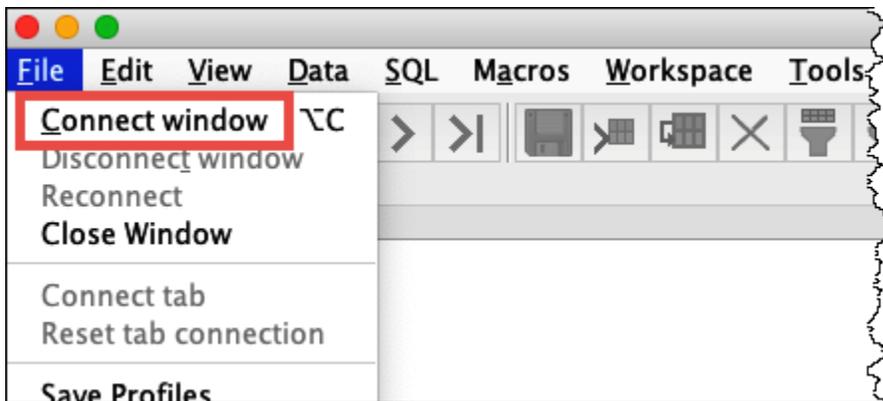
4. 在 Manage Drivers (管理驅動程式) 對話方塊中，執行以下步驟：
  - a. 選擇新驅動程式圖示。
  - b. 對於 Name (名稱)，輸入 **Athena**。
  - c. 對於 Library (程式庫)，瀏覽並選擇您剛下載的 Simba Athena JDBC .jar 檔案。
  - d. 選擇確定。



您現在可以開始建立並測試 Athena Okta 使用者的連線。

若要為 Okta 使用者建立連線

1. 依序選擇 File (檔案)、Connect window (連接視窗)。



2. 在 Connection profile (連線設定檔) 對話方塊中，輸入下列資訊來建立連線：

- 在名稱方塊中，輸入 **Athena\_Okta\_User\_Connection**。
- 對於 Driver (驅動程式)，選擇 Simba Athena JDBC 驅動程式。
- 對於 URL，執行下列其中一項作業：
  - 若要使用連線 URL，請輸入單行連線字串。以下範例新增了換行以方便閱讀。

```
jdbc:awsathena://AwsRegion=region-id;
```

```
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;  
AwsCredentialsProviderClass=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider;  
user=athena-okta-user@anycompany.com;  
password=password;  
idp_host=okta-idp-domain;  
App_ID=okta-app-id;  
SSL_Insecure=true;  
LakeFormationEnabled=true;
```

- 若要使用 AWS 基於設定檔的 URL，請執行下列步驟：

1. [設定具有認 AWS 證檔案的設定檔，如下列範例所示。AWS](#)

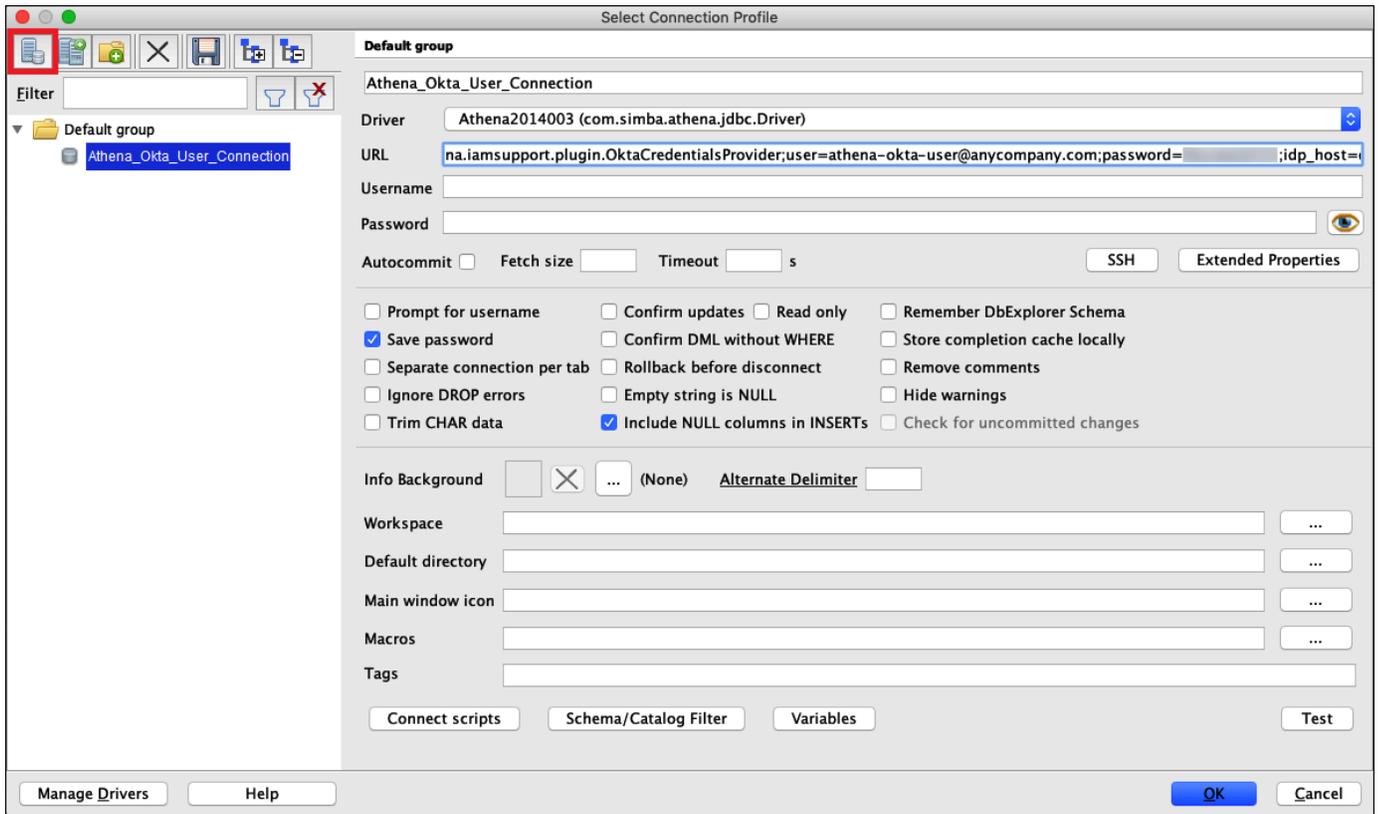
```
[athena_lf_dev]  
plugin_name=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider  
idp_host=okta-idp-domain  
app_id=okta-app-id  
uid=athena-okta-user@anycompany.com  
pwd=password
```

2. 對於 URL，輸入單行連線字串，如下列範例所示。該範例新增了換行以方便閱讀。

```
jdbc:awsathena://AwsRegion=region-id;  
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;  
profile=athena_lf_dev;  
SSL_Insecure=true;  
LakeFormationEnabled=true;
```

請注意，這些範例是連接至 Athena 所需的 URL 的基本表示。如需 URL 中支援的完整參數清單，請參閱 [JDBC 文件](#)。

下圖顯示使用連線 URL 的 SQL Workbench 連線設定檔。

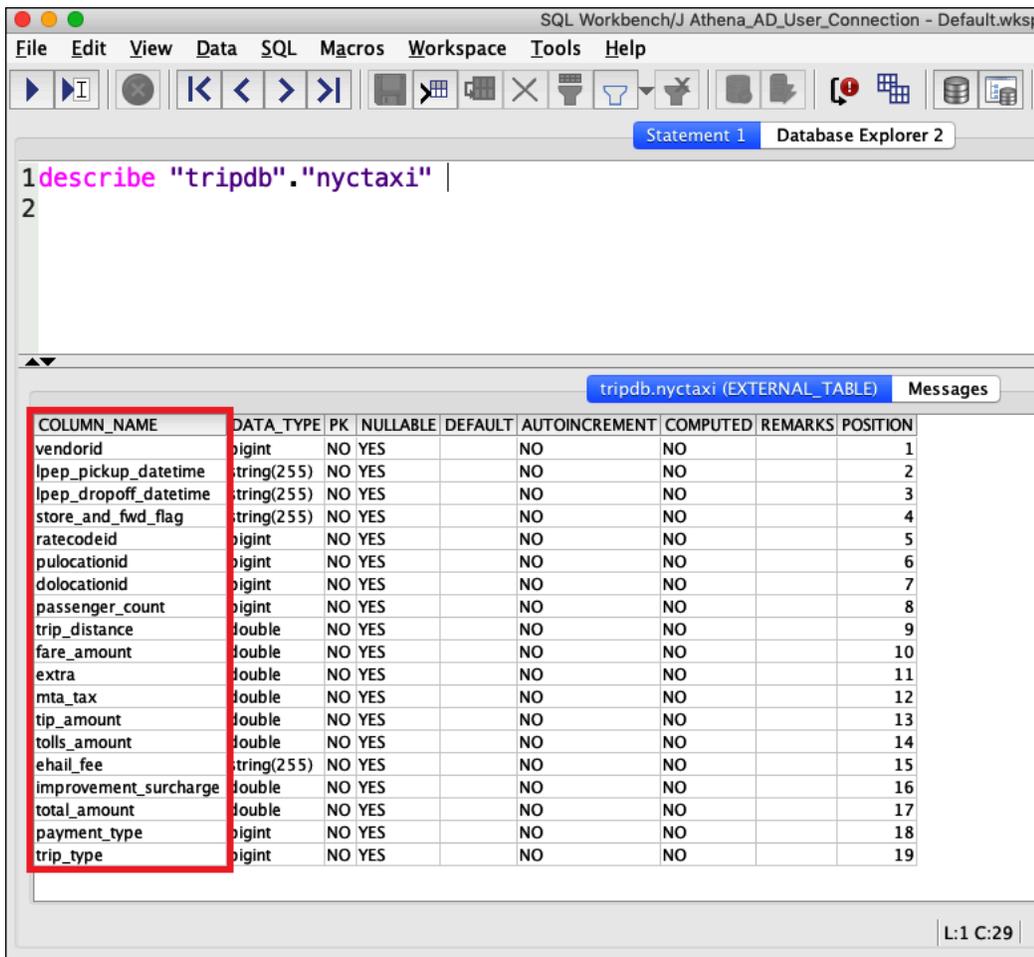


現在您已經為 Okta 使用者建立了連線，您可以擷取一些資料來進行測試。

若要測試 Okta 使用者的連線

1. 選擇 Test (測試)，然後確認連線成功。
2. 從 SQL Workbench Statement (陳述式) 視窗中，執行下列 SQL DESCRIBE 命令。確認已顯示所有資料欄。

```
DESCRIBE "tripdb"."nyctaxi"
```



Statement 1 Database Explorer 2

```
1 describe "tripdb"."nyctaxi" |
2
```

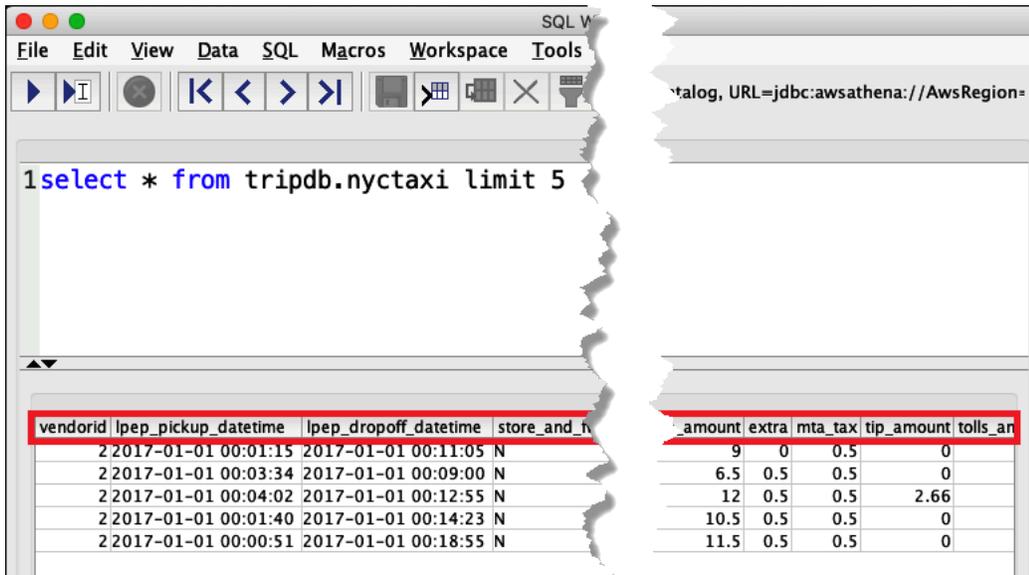
tripdb.nyctaxi (EXTERNAL\_TABLE) Messages

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DEFAULT	AUTOINCREMENT	COMPUTED	REMARKS	POSITION
vendorid	bigint	NO	YES		NO	NO		1
lpep_pickup_datetime	string(255)	NO	YES		NO	NO		2
lpep_dropoff_datetime	string(255)	NO	YES		NO	NO		3
store_and_fwd_flag	string(255)	NO	YES		NO	NO		4
ratecodeid	bigint	NO	YES		NO	NO		5
pulocationid	bigint	NO	YES		NO	NO		6
dolocationid	bigint	NO	YES		NO	NO		7
passenger_count	bigint	NO	YES		NO	NO		8
trip_distance	double	NO	YES		NO	NO		9
fare_amount	double	NO	YES		NO	NO		10
extra	double	NO	YES		NO	NO		11
mta_tax	double	NO	YES		NO	NO		12
tip_amount	double	NO	YES		NO	NO		13
tolls_amount	double	NO	YES		NO	NO		14
ehail_fee	string(255)	NO	YES		NO	NO		15
improvement_surcharge	double	NO	YES		NO	NO		16
total_amount	double	NO	YES		NO	NO		17
payment_type	bigint	NO	YES		NO	NO		18
trip_type	bigint	NO	YES		NO	NO		19

L:1 C:29

3. 從 SQL Workbench Statement (陳述式) 視窗中，執行下列 SQL SELECT 命令。確認已顯示所有資料欄。

```
SELECT * FROM tripdb.nyctaxi LIMIT 5
```



接下來，您確認身為lf-business-analyst群組的成員，只能存取您先前在 Lake Formation 中指定之表格的前三欄。athena-ba-user

若要驗證存取權 athena-ba-user

1. 在 SQL Workbench 的 Connection profile (連線設定檔) 對話方塊中，建立另一個連線設定檔。

- 對於連線設定檔名稱，輸入 **Athena\_Okta\_Group\_Connection**。
- 對於 Driver (驅動程式)，選擇 Simba Athena JDBC 驅動程式。
- 對於 URL，執行下列其中一項作業：
  - 若要使用連線 URL，請輸入單行連線字串。以下範例新增了換行以方便閱讀。

```
jdbc:awsathena://AwsRegion=region-id;  
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;  
AwsCredentialsProviderClass=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider;  
user=athena-ba-user@anycompany.com;  
password=password;  
idp_host=okta-idp-domain;  
App_ID=okta-application-id;  
SSL_Insecure=true;  
LakeFormationEnabled=true;
```

- 若要使用 AWS 基於設定檔的 URL，請執行下列步驟：

1. 設定具有認證檔案的設定檔，如下列範例所示。AWS

```
[athena_lf_ba]
plugin_name=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider
idp_host=okta-idp-domain
app_id=okta-application-id
uid=athena-ba-user@anycompany.com
pwd=password
```

- 對於 URL，輸入單行連線字串，如下所示。該範例新增了換行以方便閱讀。

```
jdbc:awsathena://AwsRegion=region-id;
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;
profile=athena_lf_ba;
SSL_Insecure=true;
LakeFormationEnabled=true;
```

- 選擇 Test (測試) 以確認連線是否成功。
- 從 SQL Statement (SQL 陳述式) 視窗中，執行與您之前執行相同的 DESCRIBE 和 SELECT SQL 命令並檢查結果。

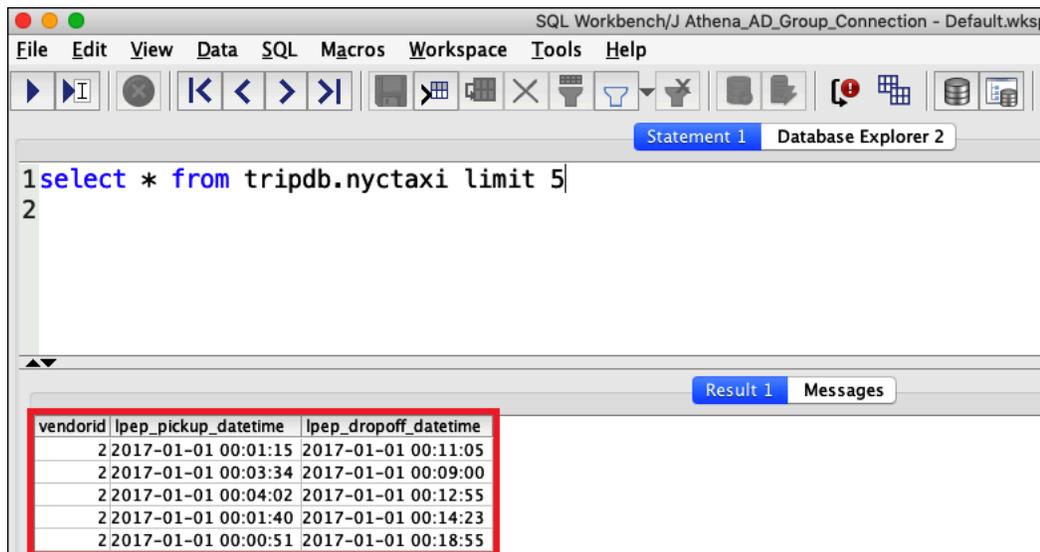
由於athena-ba-user是lf-business-analyst群組的成員，因此只會傳回您在 Lake Formation 主控台中指定的前三個資料行。

The screenshot shows the SQL Workbench/J interface. The SQL Statement window contains the following text:

```
1 describe tripdb.nyctaxi |
2
```

The Database Explorer window shows the table `tripdb.nyctaxi (EXTERNAL_TABLE)`. The result table is as follows:

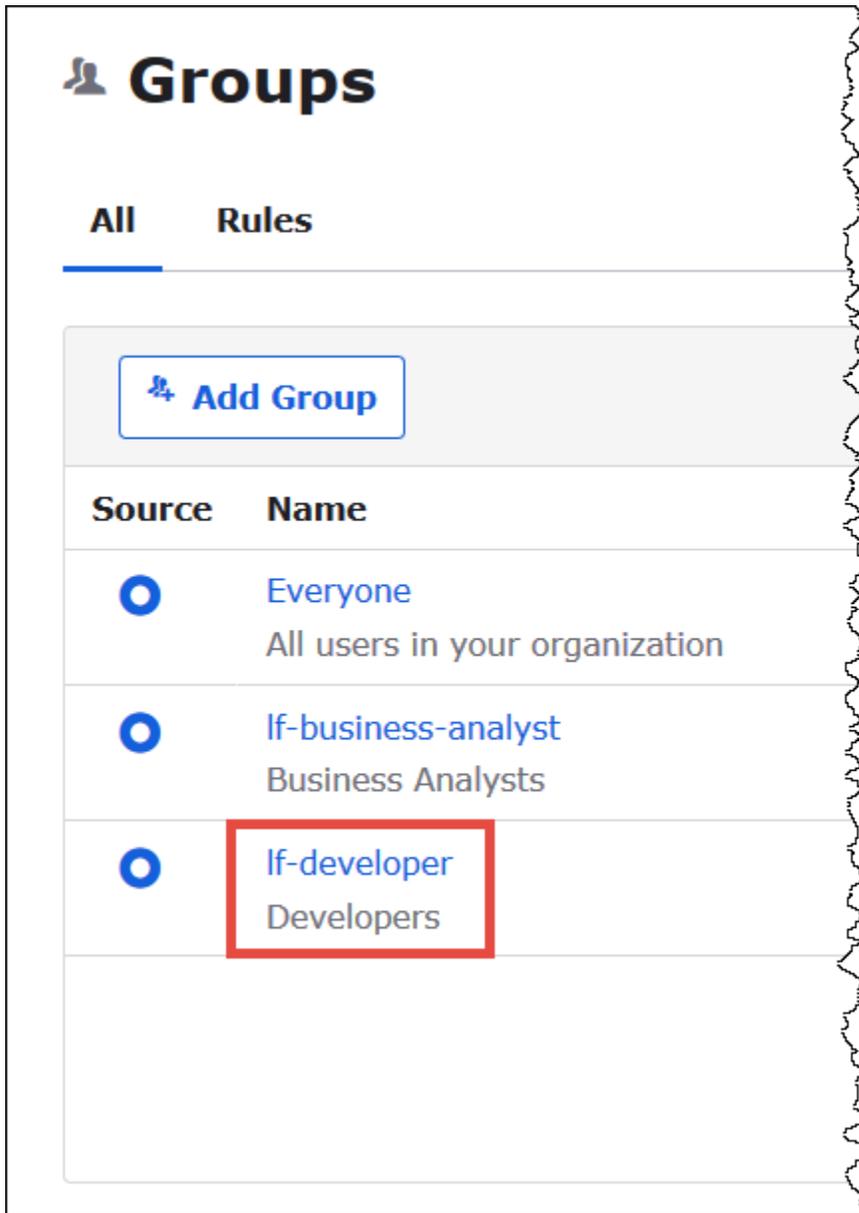
COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DEFAULT	AUTOINCREMENT	COMPUTED	REMARKS	POSITION
vendorid	bigint	NO	YES		NO	NO		1
lpep_pickup_datetime	string(255)	NO	YES		NO	NO		2
lpep_dropoff_datetime	string(255)	NO	YES		NO	NO		3



接下來，您需返回 Okta 主控台，將 `athena-ba-user` 新增至 `lf-developer` Okta 群組。

要添加到 LF-開 `athena-ba-user` 發人員組

1. 以指派的 Okta 網域的管理使用者身分登入 Okta 主控台。
2. 選擇 Directory (目錄)，然後選擇 Groups (群組)。
3. 在 Groups (群組) 頁面上，選擇 `lf-developer` 群組。



4. 選擇 Manage People (管理人員)。
5. 從不是成員列表中，選擇athena-ba-user將其添加到 If-developer 組。
6. 選擇儲存。

現在請返回 Lake Formation 主控台來設定 If-developer 群組的資料表許可。

若要設定資料表權限 If-developer-group

1. 以資料湖管理員身分登入 Lake Formation 主控台。
2. 在導覽窗格中，選擇 Tables (資料表)。

3. 選取 `nyctaxi` 資料表。
4. 依序選擇 Actions (動作) 和 Grant (授予)。
5. 在 Grant Permissions (授予許可) 對話方塊中，輸入下列資訊：
  - 對於 SAML 和 Amazon 的 QuickSight 使用者和群組，請以下列格式輸入 Okta SAML If 開發人員群組 ARN：
  - 對於 Columns (資料欄)、Choose filter type (選擇篩選條件類型)，選擇 Include columns (包含資料欄)。
  - 選擇 `trip_type` 資料欄。
  - 對於 Table permissions (資料表許可)，選擇 SELECT (選取)。
6. 選擇 Grant (授予)。

現在，您可以使用 SQL Workbench 來驗證 `If-developer` 群組的許可變更。這種變化應該反映在可用的數據中 `athena-ba-user`，誰現在是 `If-developer` 組的成員。

若要驗證的權限變更 `athena-ba-user`

1. 關閉 SQL Workbench 程式，然後再重新開啟。
2. Connect 到的設定檔 `athena-ba-user`。
3. 從 Statement (陳述式) 視窗中，發出您先前執行的相同 SQL 陳述式：

這一次將會顯示 `trip_type` 資料欄。

The screenshot shows the SQL Workbench interface with the following content:

```

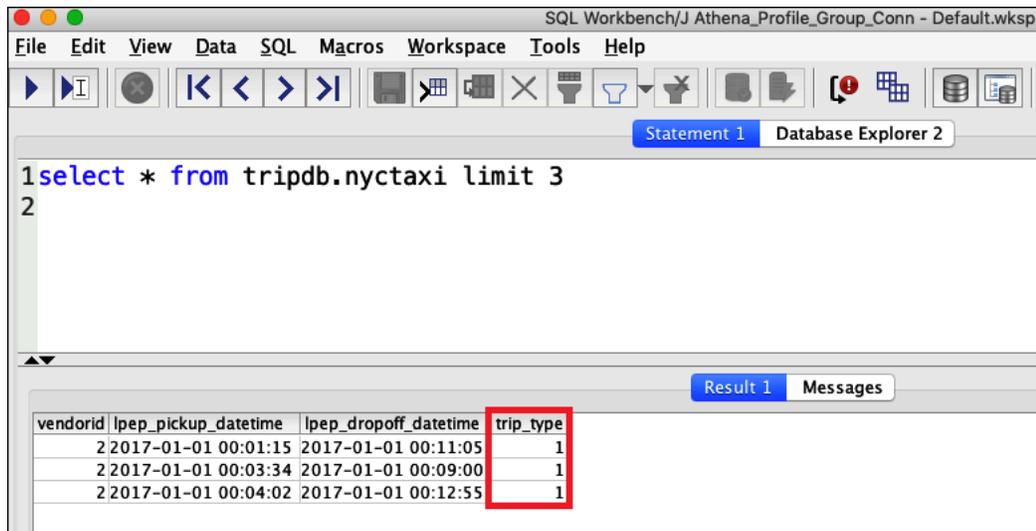
1 describe tripdb.nyctaxi
2

```

The results pane shows the table structure for `tripdb.nyctaxi (EXTERNAL_TABLE)`:

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DEFAULT	AUTOINCREMENT	COMPUTED	REMARKS	POSITION
vendorid	bigint	NO	YES		NO	NO		1
lpep_pickup_datetime	string(255)	NO	YES		NO	NO		2
lpep_dropoff_datetime	string(255)	NO	YES		NO	NO		3
trip_type	bigint	NO	YES		NO	NO		4

因為現在athena-ba-user是 lf-developer 和lf-business-analyst群組的成員，因此這些群組的 Lake Formation 權限組合會決定傳回的資料行。



## 結論

在本教學課程中，您將 Athena 整合設定為 AWS Lake Formation 使用 Okta 做為 SAML 提供者。您使用 Lake Formation 和 IAM 來控制資料湖資料目錄中可供 SAML 使用者使用的 AWS Glue 資源。

## 相關資源

如需相關資訊，請參閱下列資源。

- [使用 JDBC 連接到 Amazon Athena](#)
- [啟用對 Athena API 的聯合存取](#)
- [AWS Lake Formation 開發人員指南](#)
- 《AWS Lake Formation 開發人員指南》中的[授予和撤銷資料目錄許可](#)。
- 《IAM 使用者指南》中的身分提供者[身分提供者與聯合](#)。
- 《IAM 使用者指南》中的[建立 IAM SAML 身分提供者](#)。
- [啟用聯合以 AWS 使用AWS 安全性部落格上的 Windows 作用中目錄、ADFS 和 SAML 2.0。](#)

## 工作負載管理

您可以使用 Athena 的工作群組、容量管理、效能調校、壓縮支援、標籤和服務配額功能來管理工作負載。

## 主題

- [使用工作群組來控制查詢存取和成本](#)
- [管理查詢處理容量](#)
- [Athena 中的效能調校](#)
- [Athena 壓縮支援](#)
- [標記 Athena 資源](#)
- [Service Quotas](#)

## 使用工作群組來控制查詢存取和成本

使用工作群組來區隔使用者、團隊、應用程式或工作負載，以對每個查詢或整個工作群組可處理的資料量設定限制，並追蹤成本。因為工作群組扮演資源的角色，所以您可以使用資源層級的身分識別型政策來控制對特定工作群組的存取。您也可以 Amazon 中檢視與查詢相關的指標 CloudWatch、透過設定掃描的資料量限制來控制成本、建立閾值，以及在違反這些閾值時觸發動作 (例如 Amazon SNS)。

若要進一步控制成本，您可以使用指定的資料處理單位數目來建立容量保留，並將一或多個工作群組新增至保留。如需詳細資訊，請參閱 [管理查詢處理容量](#)。

工作群組與 IAM CloudWatch、Amazon 簡單通知服務以及 [AWS 成本和用量報告](#) 整合，如下所示：

- 具有資源層級許可的 IAM 身分型政策可控制誰可以在工作群組中執行查詢。
- 如果您啟用查詢量度 CloudWatch，Athena 會將工作群組查詢量度發佈至。
- 在 Amazon SNS 中，您可以建立 Amazon SNS 主題，當工作群組中查詢的資料用量控制超過您建立的閾值時，就對指定的工作群組使用者發出警示。
- 您在 Billing and Cost Management 主控台中使用被設定為成本分配標籤的標籤來標記工作群組時，與在該工作群組中正執行的查詢相關聯的成本將顯示在具有該成本分配標籤的「成本和用量報告」中。

## 主題

- [使用工作群組來執行查詢](#)
- [使用指標和事件控制成本並 CloudWatch 監控查詢](#)

另請參閱 AWS 大數據部落格文章 [使用 Amazon Athena 工作群組個別查詢和管理成本](#)，其中說明如何使用工作群組分隔工作負載、控制使用者存取以及管理查詢用量和成本。

## 使用工作群組來執行查詢

我們建議使用工作群組來隔離團隊、應用程式或不同工作負載的查詢。例如，您可以為組織中的兩個不同團隊建立個別的工作群組。您也可以分隔工作負載。例如，您可以建立兩個獨立工作群組，一個用於自動排定的應用程式，例如產生報告，另一個供分析師臨時使用。您可以切換工作群組。

### 主題

- [使用工作群組的優勢](#)
- [工作群組的運作方式](#)
- [設定工作群組](#)
- [用於存取工作群組的 IAM 政策](#)
- [工作群組設定](#)
- [管理工作群組](#)
- [使用啟用 IAM Identity Center 的 Athena 工作群組](#)
- [Athena 工作群組 API](#)
- [工作群組疑難排解](#)

### 使用工作群組的優勢

工作群組可讓您：

將使用者、團隊、應用程式或工作負載隔離成工作群組。

每個工作群組有其自己各別的查詢歷史記錄和儲存的查詢清單。如需詳細資訊，請參閱 [工作群組的運作方式](#)。

對於工作群組中的所有查詢，您可以選擇設定工作群組設定。其包含用於存放查詢結果的 Simple Storage Service (Amazon S3) 位置、預期的儲存貯體擁有者、加密和對寫入查詢結果儲存貯體之物件的控制。您也可以強制工作群組設定。如需詳細資訊，請參閱 [工作群組設定](#)。

強制成本限制。

對於工作群組中的查詢，您可以設定兩種類型的成本限制：

- 每次查詢限制是針對每次查詢所掃描的資料量閾值。當查詢超過指定的閾值時，Athena 就會取消查詢。此限制適用於工作群組內執行的每個查詢。您只能設定一個每一查詢限制，並視需要而更新。

- 每一工作群組限制是對於每個工作群組中的查詢所掃描的資料量，您可以為每個工作群組設定的閾值。超過閾值會啟用 Amazon SNS 警示，而觸發您選擇的動作，例如將電子郵件傳送給指定的使用者。您可以為每個工作群組設定多個每一工作群組限制。

如需詳細步驟，請參閱[設定資料用量控制限制](#)。

追蹤中所有工作群組  
查詢的查詢相關量度。  
CloudWatch

對於在工作群組中執行的每個查詢，如果您將工作群組設定為發佈指標，Athena 會將其發佈到 CloudWatch。您可以在 Athena 主控台中檢視每個工作群組的[查詢指標](#)。在中 CloudWatch，您可以建立自訂儀表板，並針對這些指標設定臨界值和警示。

## 工作群組的運作方式

Athena 中的工作群組具有下列特性：

- 在預設情況下，每個帳戶都有一個主要工作群組，而預設許可允許所有已驗證的使用者存取此工作群組。無法刪除主要工作群組。
- 您建立的每個群組只針對其中執行的查詢來顯示儲存的查詢和查詢歷史記錄，而不是針對帳戶中的所有查詢。這樣可隔開您的查詢和帳戶內的其他查詢，讓您更有效率尋找您自己儲存的查詢和歷史記錄中的查詢。
- 停用工作群組會阻止在其中執行查詢，直到又啟用為止。傳送到已停用的工作群組的查詢會失敗，直到再次啟用為止。
- 如有許可，您可以刪除空的群組，以及包含所儲存查詢的工作群組。在這種情況下，在刪除工作群組之前，Athena 會警告您將刪除儲存的查詢。在刪除其他使用者可存取的工作群組之前，請確定其使用者可以存取能夠讓他們繼續執行查詢的其他工作群組。
- 您可以設定工作群組整體設定，並強制工作群組中執行的所有查詢必須遵守。設定包含 Simple Storage Service (Amazon S3) 中查詢結果的位置、預期的儲存貯體擁有者、加密和對寫入查詢結果儲存貯體之物件的控制。

### Important

當您強制工作群組整體設定時，此工作群組中執行的所有查詢都會使用工作群組設定。即使其用戶端設定與工作群組設定可能不同，也是如此。如需相關資訊，請參閱[工作群組設定覆寫用戶端設定](#)。

## 工作群組限制

- 您可在帳戶中為每個區域建立最多 1000 個工作群組。
- 無法刪除主要工作群組。
- 在每個工作群組內，您最多可以開啟十個查詢索引標籤。在工作群組之間切換時，您的查詢索引標籤最多可以在三個工作群組中保持開啟。

## 設定工作群組

設定工作群組涉及到建立工作群組和規定其使用許可。首先，決定您的組織需要哪些工作群組，並建立這些工作群組。接著，設定 IAM 工作群組政策，以控制對 workgroup 資源的使用者存取和動作。能夠存取這些工作群組的使用者現在可以在其中執行查詢。

### Note

當您第一次開始使用工作群組時，請使用這些任務來設定工作群組。如果 Athena 帳戶已經使用工作群組，每個帳戶的使用者需要許可，才可在帳戶中的一或多個工作群組中執行查詢。執行查詢之前，請檢查 IAM 政策，以查明您可以存取哪些工作群組，視需要調整政策，並[切換](#)到您想要使用的工作群組。

在預設情況下，如果您尚未建立任何工作群組，則您帳戶中的所有查詢會在主要工作群組中執行。

Athena 會在位於主控台右上角的 Workgroup (工作群組) 選項中顯示目前工作群組。您可以使用此選項來切換工作群組。當您執行查詢時，它們會在目前工作群組中執行。您可以在主控台，或使用 API 操作、命令列界面或透過 JDBC 或 ODBC 驅動程式使用用戶端應用程式，在工作群組的內容中執行查詢。當您能夠存取工作群組時，您可以檢視工作群組的設定、指標和資料用量控制限制。利用額外的許可，您可以編輯設定和資料用量控制限制。

## 若要設定工作群組

1. 決定建立哪些工作群組。例如，您可以決定下列各項：
  - 誰可以在每個工作群組中執行查詢，以及誰擁有工作群組的組態。這會決定您建立的 IAM 政策。如需詳細資訊，請參閱 [用於存取工作群組的 IAM 政策](#)。
  - 對於每個工作群組中執行的查詢，Amazon S3 中哪些位置要用於存放查詢結果。在您可以為工作群組查詢結果指定某個位置前，Amazon S3 中必須存在該位置。所有使用工作群組的使用者必須能夠存取這個位置。如需詳細資訊，請參閱 [工作群組設定](#)。

- Simple Storage Service (Amazon S3) 查詢結果儲存貯體的擁有者是否具有對寫入儲存貯體的新物件的完整控制。例如，如果您的查詢結果位置係由另一個帳戶所擁有，則可以授予另一個帳戶對查詢結果的所有權和完整控制。如需詳細資訊，請參閱[AclConfiguration](#)。
  - 指定您希望成為輸出位置值區擁有者的識別碼。AWS 帳戶 這是一項選擇性的附加安全措施。如果儲存貯體擁有者的帳戶 ID 與您在此處指定的 ID 不相符，則嘗試輸出到儲存貯體的動作會失敗。如需詳細資訊，請參閱 Amazon S3 使用者指南中的[使用儲存貯體擁有者條件驗證儲存貯體擁有權](#)。此設定不適用於 CTAS、INSERT INTO 或 UNLOAD 陳述式。
  - 需要哪些加密設定，以及哪些工作群組的查詢必須加密。我們建議您為加密和未加密的查詢建立個別的工作群組。如此一來，您可以強制加密工作群組中執行的所有查詢。如需詳細資訊，請參閱 [加密 Amazon S3 中存放的 Athena 查詢結果](#)。
2. 視需要建立工作群組並新增標籤。如需這些步驟，請參閱 [建立工作群組](#)。
  3. 建立 IAM 政策以允許使用者、群組或角色存取工作群組。政策建立工作群組成員資格和在 workgroup 資源上的動作存取權。如需詳細步驟，請參閱 [用於存取工作群組的 IAM 政策](#)。如需 JSON 政策的範例，請參閱[存取工作群組和標籤](#)。
  4. 設定工作群組設定。指定 Simple Storage Service (Amazon S3) 中用於查詢結果的位置，並選擇性地指定預期的儲存貯體擁有者、加密設定和對寫入查詢結果儲存貯體之物件的控制。您可以強制工作群組設定。如需詳細資訊，請參閱[工作群組設定](#)。

#### Important

如果您[覆寫用戶端設定](#)，Athena 將使用工作群組的設定。這會影響您在主控台、透過驅動程式、命令列界面或 API 操作執行的查詢。

雖然查詢會繼續執行，但根據特定 Amazon S3 儲存貯體中是否有結果而建立的自動化可能失效。我們建議您在覆寫之前先通知使用者。在工作群組設定為覆寫之後，您可以在驅動程式或 API 中省略指定用戶端設定。

5. 將哪些工作群組要用於執行查詢告知使用者。傳送電子郵件，告知帳戶的使用者他們可以使用的工作群組名稱、所需的 IAM 政策和工作群組設定。
6. 設定查詢和工作群組的成本控制限制，也稱為資料用量控制限制。若要在超出閾值時接獲通知，請建立 Amazon SNS 主題並設定訂閱。如需詳細步驟，請參閱 [設定資料用量控制限制](#) 和《Amazon Simple Notification Service 開發人員指南》中的 [Amazon SNS 入門](#)。
7. 切換到工作群組，以便執行查詢。若要執行查詢，請切換到適當的工作群組。如需詳細步驟，請參閱[the section called “指定工作群組在其中執行查詢”](#)。

## 用於存取工作群組的 IAM 政策

若要控制對工作群組的存取，請使用資源層級 IAM 許可或以身分為基礎的 IAM 政策。每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

### Note

若要存取啟用受信任身分傳播的工作群組，IAM 身分中心使用者必須指派給 Athena [GetWorkGroup](#) API 動作回應所傳回的工作群組。IdentityCenterApplicationArn

下列程序專用於 Athena。

如需 IAM 特定的資訊，請參閱本節最後列出的連結。如需範例 JSON 工作群組政策的相關資訊，請參閱 [工作群組政策範例](#)。

若要在 IAM 主控台中使用視覺化編輯器來建立工作群組政策

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在左邊的導覽窗格中，選擇 Policies (政策)，然後選擇 Create policy (建立政策)。
3. 在 Visual editor (視覺化編輯器) 標籤上，選擇 Choose a service (選擇一項服務)。接著選擇 Athena 以新增到政策。
4. 選擇 Select actions (選取動作)，然後選擇要新增到政策的動作。視覺化編輯器會顯示 Athena 中可用的動作。如需詳細資訊，請參閱《服務授權參考》中的 [Amazon Athena 的動作、資源和條件索引鍵](#)。
5. 選擇 add actions (新增動作) 來輸入特定動作，或使用萬用字元 (\*) 來指定多個動作。

預設情況下，您建立的政策允許執行選擇的操作。對於 Athena 中的 workgroup 資源，如果您選擇的一個或多個動作支援資源層級許可，則編輯器會列出 workgroup 資源。

6. 選擇 Resources (資源) 來為您的政策指定工作群組。如需 JSON 工作群組政策的範例，請參閱 [工作群組政策範例](#)。
7. 如下所示指定 workgroup 資源：

```
arn:aws:athena:<region>:<user-account>:workgroup/<workgroup-name>
```

8. 選擇 Review policy (檢閱政策)，然後為您正在建立的政策輸入 Name (名稱) 與 Description (描述) (選用)。檢閱政策摘要來確認您已授予想要的許可。
9. 選擇 Create policy (建立政策) 儲存您的新政策。
10. 將此基於身分的政策連接到使用者、群組或角色。

如需詳細資訊，請參閱《服務授權參考》與《IAM 使用者指南》中的下列主題：

- [Amazon Athena 的操作、資料和條件索引鍵](#)
- [使用視覺化編輯器來建立政策](#)
- [新增和移除 IAM 政策](#)
- [控制資源的存取](#)

如需 JSON 工作群組政策的範例，請參閱[工作群組政策範例](#)。

如需 Amazon Athena 動作的完整清單，請參閱《[Amazon Athena API 參考](#)》中的 API 動作名稱。

#### 工作群組政策範例

本節包含可讓您在工作群組上用來啟用各種動作的政策範例。每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

工作群組是由 Athena 管理的 IAM 資源。因此，如果您的工作群組政策使用的動作接受 workgroup 做為輸入，您必須指定工作群組的 ARN，如下所示：

```
"Resource": [arn:aws:athena:<region>:<user-account>:workgroup/<workgroup-name>]
```

其中 *<workgroup-name>* 為工作群組的名稱。例如，假設工作群組名為 test\_workgroup，請如下所示將它指定為資源：

```
"Resource": ["arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"]
```

如需 Amazon Athena 動作的完整清單，請參閱《[Amazon Athena API 參考](#)》中的 API 動作名稱。如需有關 IAM 政策的詳細資訊，請參閱《IAM 使用者指南》中的[使用視覺化編輯器來建立政策](#)。如需有關為工作群組建立 IAM 政策的詳細資訊，請參閱[用於存取工作群組的 IAM 政策](#)。

- [Example policy for full access to all workgroups](#)
- [Example policy for full access to a specified workgroup](#)

- [Example policy for running queries in a specified workgroup](#)
- [Example policy for running queries in the primary workgroup](#)
- [Example policy for management operations on a specified workgroup](#)
- [Example policy for listing workgroups](#)
- [Example policy for running and stopping queries in a specific workgroup](#)
- [Example policy for working with named queries in a specific workgroup](#)
- [Example policy for working with Spark notebooks](#)

#### Example 適用於完整存取所有工作群組的政策範例

以下政策允許完整存取帳戶中可能存在的所有工作群組資源。對於您的帳戶中必須為所有其他使用者來管理工作群組的那些使用者，我們建議您使用此政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

#### Example 適用於完整存取指定工作群組的政策範例

以下政策允許完整存取名為 `workgroupA` 的單一特定的工作群組資源。對於能夠完全控制特定群組的使用者，您可以使用此政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListEngineVersions",

```

```

        "athena:ListWorkGroups",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "athena:BatchGetQueryExecution",
        "athena:GetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena:CreatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "athena>DeleteWorkGroup",
        "athena:UpdateWorkGroup",
        "athena:GetWorkGroup",
        "athena:CreateWorkGroup"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
    ]
}

```

```

    ]
  }
]
}

```

### Example 適用於在指定的工作群組中執行查詢的政策範例

在以下政策中，允許使用者在指定的 `workgroupA` 中執行查詢並檢視查詢。不允許使用者對工作群組本身執行管理任務，例如更新或刪除。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListEngineVersions",
        "athena:ListWorkGroups",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "athena:BatchGetQueryExecution",
        "athena:GetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena:CreatePreparedStatement",

```

```

        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
    ]
}
]
}

```

Example 適用於在主要工作群組中執行查詢的政策範例

您可以修改上述範例，以允許特定使用者也在主要工作群組中執行查詢。

#### Note

對於以其他方式設定為在指定的工作群組中執行查詢的所有使用者，我們建議您新增主要工作群組資源。為了以防其指定的工作群組遭到刪除或停用，將此資源新增到其工作群組使用者政策非常有用。在這種情況下，他們可以在主要工作群組中繼續執行查詢。

若要允許您帳戶中的使用者在主要工作群組中執行查詢，請將包含主要工作群組 ARN 的行新增到 [Example policy for running queries in a specified workgroup](#) 的資源區段，如以下範例所示。

```
arn:aws:athena:us-east-1:123456789012:workgroup/primary"
```

Example 適用於在指定的工作群組上執行管理操作的政策範例

在以下政策中，允許使用者建立、刪除、取得詳細資訊和更新工作群組 `test_workgroup`。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListEngineVersions"
      ],
      "Resource": "*"
    }
  ],
}

```

```
{
  "Effect": "Allow",
  "Action": [
    "athena:CreateWorkGroup",
    "athena:GetWorkGroup",
    "athena>DeleteWorkGroup",
    "athena:UpdateWorkGroup"
  ],
  "Resource": [
    "arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"
  ]
}
```

### Example 適用於列出工作群組的政策範例

以下政策可讓所有使用者列出所有工作群組：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListWorkGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

### Example 適用於在特定工作群組中執行和停止查詢的政策範例

在這個政策中，允許使用者在工作群組中執行查詢：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:StartQueryExecution",

```

```

        "athena:StopQueryExecution"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"
    ]
}
]
}

```

### Example 適用於在特定工作群組中處理具名查詢的政策範例

在以下政策中，使用者有許可在指定的工作群組中建立、刪除具名查詢及取得其相關資訊：

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena>DeleteNamedQuery"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"
      ]
    }
  ]
}

```

### Example 使用 Athena 中的 Spark 筆記本的範例政策

使用類似下列所示的政策來使用 Athena 中的 Spark 筆記本。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreatingWorkGroupWithDefaults",
      "Action": [
        "athena:CreateWorkGroup",
        "s3:CreateBucket",
        "iam:CreateRole",

```

```

        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "s3:GetBucketLocation",
        "athena:ImportNotebook"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/Demo*",
        "arn:aws:s3:::123456789012-us-east-1-athena-results-bucket-*",
        "arn:aws:iam::123456789012:role/service-role/
AWSAthenaSparkExecutionRole-*",
        "arn:aws:iam::123456789012:policy/service-role/
AWSAthenaSparkRolePolicy-*"
    ]
},
{
    "Sid": "AllowRunningCalculations",
    "Action": [
        "athena:ListWorkGroups",
        "athena:GetWorkGroup",
        "athena:StartSession",
        "athena:CreateNotebook",
        "athena:ListNotebookMetadata",
        "athena:ListNotebookSessions",
        "athena:GetSessionStatus",
        "athena:GetSession",
        "athena:GetNotebookMetadata",
        "athena:CreatePresignedNotebookUrl"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/Demo*"
},
{
    "Sid": "AllowListWorkGroupAndEngineVersions",
    "Action": [
        "athena:ListWorkGroups",
        "athena:ListEngineVersions"
    ],
    "Effect": "Allow",
    "Resource": "*"
}
]
}

```

## 工作群組設定

每個工作群組有以下設定：

- 唯一名稱。它可包含 1 到 128 個字元，包括英數字元、連字號和底線。在您建立工作群組後，便無法更改其名稱。不過，您可以使用相同的設定和不同的名稱來建立新的工作群組。
- 適用於工作群組中執行的所有查詢的設定。其中包括：
  - 對於此工作群組中執行的所有查詢，Amazon S3 中用於存放查詢結果的位置。當您建立工作群組時，此位置必須先存在，才能指定給工作群組。如需建立 Amazon S3 儲存貯體的相關資訊，請參閱[建立儲存貯體](#)。
  - 儲存貯體擁有者的查詢結果控制 – Simple Storage Service (Amazon S3) 查詢結果儲存貯體的擁有者是否具有對寫入儲存貯體的新物件的完整控制。例如，如果您的查詢結果位置係由另一個帳戶所擁有，則可以授予另一個帳戶對查詢結果的所有權和完整控制。
  - 預期值區擁有者 — 您預期作為查詢結果值區擁有者的 ID。AWS 帳戶 這是一項附加的安全措施。如果儲存貯體擁有者的帳戶 ID 與您在此處指定的 ID 不相符，則嘗試輸出到儲存貯體的動作會失敗。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[使用儲存貯體擁有者條件驗證儲存貯體擁有權](#)。

### Note

預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Amazon S3 輸出位置。它不適用於其他 Amazon S3 位置，例如外部 Amazon S3 儲存貯體、CTAS 和 INSERT INTO 目的地資料表位置、UNLOAD 陳述式輸出位置等資料來源位置，也不適用於溢出聯合查詢儲存貯體的操作，或針對另一個帳戶中的資料表執行的 SELECT 查詢。

- 對所有工作群組查詢使用加密時的加密設定。您只能加密工作群組中的所有查詢，而不能只加密其中一部分。最好建立個別的工作群組來包含加密或沒有加密的查詢。

此外，您的工作群組可以[覆寫用戶端設定](#)。在發布工作群組之前，您可以在 JDBC 或 ODBC 驅動程式中，或在 Athena 主控台的 Properties (屬性) 索引標籤中，將結果位置和加密選項指定為參數。也可以直接透過 API 操作指定這些設定。這些設定稱為「用戶端設定」。使用工作群組，您可以在工作群組層級進行這些設定，以控制用戶端層級上可用的選項。強制執行工作群組層級設定，也可避免使用者個別設定其用戶端設定。如果您為工作群組選取覆寫用戶端設定選項，則查詢會使用工作群組設定，並忽略用戶端設定。

如果選取 Override Client-Side Settings (覆寫用戶端設定)，主控台會向使用者告知其設定已變更。如果以這種方式強制工作群組設定，使用者可以省略對應的用戶端設定。然後，即使存在用戶端設定，在

主控台中執行的查詢仍會使用工作群組的設定。此外，當工作群組中的查詢透過 AWS CLI、API 作業或 JDBC 或 ODBC 驅動程式執行時，工作群組設定會覆寫用戶端設定 (例如查詢結果位置和加密)。若要查看工作群組的設定，請[檢視工作群組的詳細資訊](#)。

您也可以為工作群組中的查詢[設定查詢限制](#)。

### 工作群組設定覆寫用戶端設定

Create workgroup (建立工作群組) 和 Edit workgroup (編輯工作群組) 對話方塊有一個以 Override client-side settings (覆寫用戶端設定) 為標題的欄位。預設不選取此欄位。視您是否選取它而定，Athena 的行為如下：

- 如果未選取覆寫用戶端設定，則不會在用戶端層級強制工作群組設定。當未為工作群組選取覆寫用戶端設定選項時，Athena 將對工作群組中執行的所有查詢使用用戶端的設定，包括查詢結果位置、預期的儲存貯體擁有者、加密和對寫入查詢結果儲存貯體之物件的控制等的設定。每個使用者可以在主控台的設定選單中指定其自己的設定。如果不設定用戶端設定，則會套用工作群組整體設定。如果您使用 AWS CLI、API 動作或 JDBC 和 ODBC 驅動程式在不覆寫用戶端設定的工作群組中執行查詢，您的查詢會使用您在查詢中指定的設定。
- 如果選取覆寫用戶端設定，則會在工作群組層級強制工作群組中所有用戶端執行工作群組設定。當為工作群組選取覆寫用戶端設定選項時，Athena 將對工作群組中執行的所有查詢使用工作群組的設定，包括查詢結果位置、預期的儲存貯體擁有者、加密和對寫入查詢結果儲存貯體之物件的控制等的設定。當您使用主控台、API 動作或 JDBC 或 ODBC 驅動程式時，工作群組設定會覆寫您為查詢指定的任何用戶端設定。

如果您覆寫使用者端設定，則您或任何工作群組使用者下次開啟 Athena 主控台時，Athena 會向您告知此工作群組中的查詢使用工作群組的設定，並提示您確認此變更。

#### Important

如果您使用 API 動作 AWS CLI、或 JDBC 和 ODBC 驅動程式在覆寫用戶端設定的工作群組中執行查詢，請務必省略查詢中的用戶端設定，或更新它們以符合工作群組的設定。如果您在查詢中指定用戶端設定，但在覆寫設定的工作群組中執行這些設定，則將會使用工作群組設定執行查詢。如需有關檢視工作群組設定的資訊，請參閱 [檢視工作群組的詳細資訊](#)。

### 管理工作群組

在 <https://console.aws.amazon.com/athena/> 中，您可以執行下列任務：

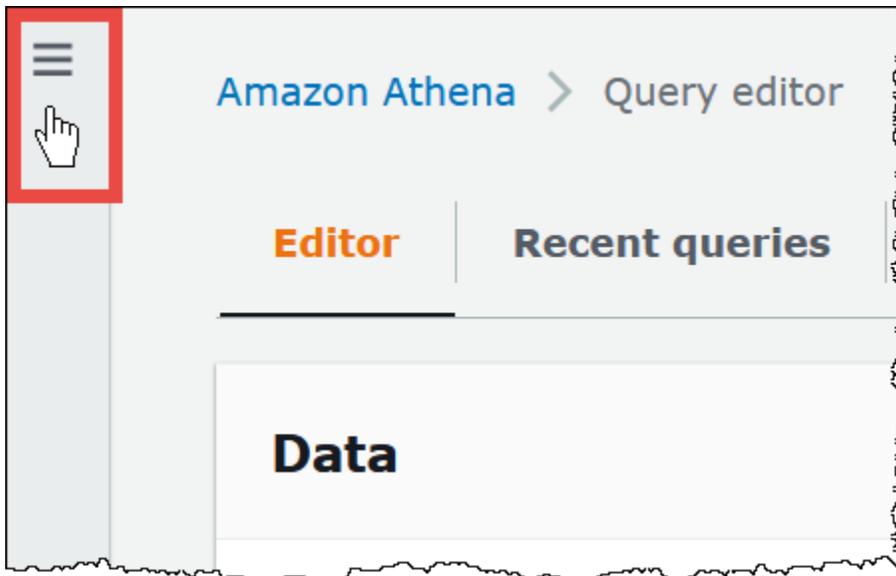
陳述式	描述
<a href="#">建立工作群組</a>	建立新的工作群組。
<a href="#">編輯工作群組</a>	編輯工作群組和變更其設定。您無法變更工作群組的名稱，但可以使用相同的設定和不同的名稱來建立新的工作群組。
<a href="#">檢視工作群組的詳細資訊</a>	檢視工作群組的詳細資訊，例如其名稱、描述、資料用量限制、查詢結果的位置、預期查詢結果儲存貯體擁有者、加密和對寫入查詢結果儲存貯體之物件的控制。如果已勾選 Override client-side settings (覆寫用戶端設定)，您也可以驗證此工作群組是否強制其設定。
<a href="#">刪除工作群組</a>	刪除工作群組。如果您刪除工作群組，將會刪除查詢歷史記錄、儲存的查詢、工作群組的設定和每一查詢資料限制控制。工作群組範圍的資料限制控制項會保留在中 CloudWatch，您可以個別刪除它們。  無法刪除主要工作群組。
<a href="#">切換工作群組</a>	在您可以存取的工作群組之間切換。
<a href="#">在工作群組之間複製儲存的查詢</a>	在工作群組之間複製儲存的查詢。例如，如果您在預覽工作群組中建立了查詢，並且想要在非預覽工作群組中使用該查詢，則您可能需要執行此操作。
<a href="#">啟用或停用工作群組</a>	啟用或停用工作群組。當工作群組停用時，其使用者無法執行查詢，或建立新的具名查詢。如果您可以存取此工作群組，則您仍可以檢視指標、資料用量限制控制、工作群組的設定、查詢歷史記錄和儲存的查詢。
<a href="#">指定工作群組在其中執行查詢</a>	您必須向 Athena 指定要使用的工作群組，才能執行查詢。您必須具有此工作群組的許可。
<a href="#">建立使用 IAM Identity Center 身分驗證的 Athena 工作群組</a>	若要將 IAM Identity Center 身分與 Athena 搭配使用，您必須建立啟用 IAM Identity Center 的工作群組。建立工作群組後，您可以使用 IAM Identity Center 主控台或 API，將 IAM Identity Center 使用者或群組指派給工作群組。

## 建立工作群組

建立工作群組需要有執行 CreateWorkgroup API 動作的許可。請參閱 [存取工作群組和標籤](#) 和 [用於存取工作群組的 IAM 政策](#)。如果您新增標籤，您也需要將許可新增到 TagResource。請參閱 [工作群組的標籤政策範例](#)。

### 在主控台建立工作群組

1. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



2. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
3. 在 Workgroups (工作群組) 頁面中，請選擇 Create workgroup (建立工作群組)。
4. 在 Create workgroup (建立工作群組) 頁面上，如下所示填寫欄位：

欄位	描述
Workgroup name (工作群組名稱)	必要。輸入工作群組的唯一名稱。使用 1 到 128 個字元。(A-Z、a-z、0-9、_、-、.)。無法變更此名稱。
Description (描述)	選用。輸入工作群組的描述。最多可包含 1024 個字元。
選擇引擎類型	如果您想對 <a href="#">Amazon S3 中的資料</a> 執行隨機操作 SQL 查詢，或想使用 <a href="#">預先建置的資料來源連接器</a> 對 Amazon S3 外部的各種資料來源執行 <a href="#">聯合查詢</a> ，則請選擇 Athena SQL。您可以使用 Athena 查詢編輯器、 <a href="#">AWS CLI</a> 或 <a href="#">Athena API</a> 執行查詢。

欄位	描述
	<p>如果您想要使用 Python 和 Apache Spark 建立、編輯和執行 Jupyter 筆記本應用程式，則請選擇 Apache Spark。Jupyter 筆記本包含一系列儲存格，其中可包含程式碼、文字、Markdown、數學、圖和豐富媒體。在 Athena 的互動式筆記本工作階段中，儲存格會依照計算順序執行。如需有關建立和設定已啟用 Spark 之工作群組的詳細資訊，則請參閱 <a href="#">在 Athena 中建立已啟用 Spark 的工作群組</a>。</p> <p>在您建立工作群組後，即可升級其分析引擎 (例如，從 Athena 引擎版本 2 升級為 Athena 引擎版本 3)，但無法變更其引擎類型。例如，Athena 引擎第 3 版工作群組無法變更為 PySpark 引擎第 3 版工作群組。</p>
更新查詢引擎	選擇在發布新的 Athena 引擎版本時工作群組的更新方式。您可以讓 Athena 決定何時更新工作群組或手動選擇引擎版本。如需詳細資訊，請參閱 <a href="#">Athena 引擎版本控制</a> 。
身分驗證方式	選擇 AWS Identity and Access Management (IAM) 以針對工作群組使用 IAM 身分驗證或聯合身分。如果您想要支援員工身分，例如來自 SAML 2.0 身分提供者 (例如 Microsoft Active Directory) 的使用者和群組，請選擇 IAM Identity Center。如需詳細資訊，請參閱《AWS IAM Identity Center 使用者指南》中的 <a href="#">跨應用程式的受信任身分傳播</a> 。
IAM Identity Center 存取的服務角色	Athena 需要 IAM 許可，才能代表您存取 IAM Identity Center。如需有關 IAM 服務角色的詳細資訊，請參閱《IAM 使用者指南》中的 <a href="#">建立角色以將許可委派給 AWS 服務</a> 。

欄位	描述
Location of query result (查詢結果的位置)	<p>選用。輸入 Amazon S3 儲存貯體的路徑或字首。這個儲存貯體和字首必須存在，您才能指定。</p> <div data-bbox="548 352 1507 758"><p><b>Note</b></p><p>如果您在主控制台執行查詢，則是否指定查詢結果位置是選擇性。如果您沒有為工作群組或沒有在 Settings (設定) 中指定此設定，Athena 會使用預設的查詢結果位置。如果您使用 API 或驅動程式執行查詢，則必須至少在下列兩個位置之一指定查詢結果位置：針對具有的個別查詢 <a href="#">OutputLocation</a>，或針對工作群組使用 <a href="#">WorkGroupConfiguration</a>。</p></div>
Expected bucket owner (預期的儲存貯體擁有者)	<p>選用。輸入您預期為輸出位置值區擁有者的識別碼。AWS 帳戶這是一項附加的安全措施。如果儲存貯體擁有者的帳戶 ID 與您在此處指定的 ID 不相符，則嘗試輸出到儲存貯體的動作會失敗。如需詳細資訊，請參閱 Amazon S3 使用者指南中的 <a href="#">使用儲存貯體擁有者條件驗證儲存貯體擁有權</a>。</p> <div data-bbox="548 1066 1507 1480"><p><b>Note</b></p><p>預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Amazon S3 輸出位置。它不適用於其他 Amazon S3 位置，例如外部 Amazon S3 儲存貯體、CTAS 和 INSERT INTO 目的地資料表位置、UNLOAD 陳述式輸出位置等資料來源位置，也不適用於溢出聯合查詢儲存貯體的操作，或針對另一個帳戶中的資料表執行的 SELECT 查詢。</p></div>

欄位	描述
指派儲存貯體擁有者對查詢結果的完整控制	<p>預設不選取此欄位。如果選取此項且查詢結果位置儲存貯體的 <a href="#">ACL 已啟用</a>，則可以授予儲存貯體擁有者對查詢結果的完整控制存取。例如，如果您的查詢結果位置係由另一個帳戶所擁有，則可以使用此選項授予另一個帳戶對查詢結果的所有權和完整控制。</p> <p>如果儲存貯體的 S3 物件所有權設定為 Bucket owner preferred (首選的儲存貯體擁有者) 時，則儲存貯體擁有者亦擁有從此工作群組寫入的所有查詢結果物件。例如，如果外部帳戶的工作群組啟用此選項，並將其查詢結果位置設定為您帳戶的 Simple Storage Service (Amazon S3) 儲存貯體，且該儲存貯體的 S3 物件所有權設定為 Bucket owner preferred (首選的儲存貯體擁有者) 時，則您擁有外部工作群組的查詢結果並具有完整的控制存取。</p> <p>當查詢結果儲存貯體的 S3 物件所有權設定為 Bucket owner enforced (強制的儲存貯體擁有者) 時，則選取此選項不會有任何作用。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的 <a href="#">物件所有權設定</a>。</p>
加密查詢結果	<p>選用。加密 Amazon S3 中存放的結果。如果選取此欄位，則工作群組中的所有查詢會加密。</p> <p>如果選取此欄位，您可以選擇 Encryption type (加密類型)、Encryption key (加密索引鍵)，並輸入 KMS Key ARN (KMS 索引鍵 ARN)。</p> <p>如果您沒有金鑰，請開啟 <a href="#">AWS KMS 主控台</a> 來建立金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 <a href="#">建立金鑰</a>。</p>
將 <i>encryption_type</i> 設定為最低加密	<p>選用。選取此選項可針對工作群組的所有使用者強制執行查詢結果的最低加密類型。選取此選項會顯示含有加密類型階層的資料表。此資料表也會顯示，當您將特定加密類型指定為最低加密類型時，工作群組使用者將允許使用的加密類型。若要使用此選項，則不得選取覆寫用戶端設定。</p> <p>如需詳細資訊，請參閱 <a href="#">為工作群組設定最低加密</a>。</p>

欄位	描述
啟用 S3 Access Grants	當您選擇 IAM Identity Center 做為身分驗證方式時，預設會選取此欄位。選取後，此選項會將 IAM Identity Center 以使用者或群組為基礎的許可套用至 Amazon S3 位置。
建立以使用者身分為基礎的 S3 字首	選取此選項後，Athena 會在存放查詢結果時建立 Amazon S3 字首。字首以使用者的 IAM Identity Center 使用者身分為基礎。
將查詢量度發佈至 CloudWatch	預設會選取此欄位。將查詢指標發佈至 CloudWatch。請參閱 <a href="#">使用 CloudWatch 指標監控 Athena 查詢</a> 。
Override client-side settings (覆寫用戶端設定)	預設不選取此欄位。如果您選取此欄位，則工作群組設定會套用至工作群組中的所有查詢，並覆寫用戶端設定。如需詳細資訊，請參閱 <a href="#">工作群組設定覆寫用戶端設定</a> 。
Requester Pays S3 buckets (申請者付款 S3 儲存貯體)	選用。如果工作群組使用者將對存放在設定為申請者付款的 Amazon S3 儲存貯體中的資料執行查詢，請選擇 Turn on queries on requester pays buckets in Amazon S3 (在 Amazon S3 中對申請者付款儲存貯體開啟查詢)。執行查詢的使用者帳戶需要支付與查詢相關聯的適用資料存取和資料傳輸費用。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 <a href="#">申請者付款儲存貯體</a> 。
Manage per query data usage control (管理每一查詢資料用量控制)	選用。設定允許查詢掃描的最大資料量限制。您只能為工作群組設定一個每一查詢限制。此限制會套用於工作群組中的所有查詢，如果查詢超出限制，則查詢將被取消。如需詳細資訊，請參閱 <a href="#">設定資料用量控制限制</a> 。
Workgroup data usage alerts (工作群組資料用量提醒)	選用。當工作群組中執行的查詢在特定時段內掃描指定數量的資料時，設定多個提醒閾值。提醒是使用 Amazon CloudWatch 警示實作，並套用至工作群組中的所有查詢。如需詳細資訊，請參閱 <a href="#">Amazon 使用 CloudWatch 者指南中的使用 Amazon CloudWatch 警示</a> 。

欄位	描述
Tags (標籤)	選用。將一或多個標籤新增到工作群組。標籤是您指派給 Athena 工作群組資源的標籤。由索引鍵和值組成。使用 AWS <a href="#">標籤最佳實踐</a> 建立一組一致的標籤，並依目的、擁有者或環境對工作群組進行分類。您也可以使用 IAM 政策中使用標籤，並用來控制帳單費用。請勿在相同的工作群組中使用重複的標籤索引鍵。如需詳細資訊，請參閱 <a href="#">the section called “標記 資源”</a> 。

5. 選擇建立工作群組。工作群組會出現在 Workgroups (工作群組) 頁面上的清單。

您也可以使用 [CreateWorkGroup](#) API 作業建立工作群組。

#### Important

在您建立工作群組後，請建立 [用於存取工作群組的 IAM 政策](#) IAM，這可讓您能夠執行工作群組相關的動作。

## 編輯工作群組

編輯工作群組需要有執行 UpdateWorkgroup API 操作的許可。請參閱 [存取工作群組和標籤](#) 和 [用於存取工作群組的 IAM 政策](#)。如果您新增或編輯標籤，您也需要有 TagResource 的許可。請參閱 [工作群組的標籤政策範例](#)。

### 在主控台編輯工作群組

1. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
2. 在 Workgroups (工作群組) 頁面上，選取您要編輯的工作群組的按鈕。
3. 選擇 Actions (動作)、Edit (編輯)。
4. 請依需求變更欄位。關於欄位清單，請參閱 [建立工作群組](#)。工作群組的名稱除外，您可以變更所有欄位。如果您需要變更名稱，請使用新的名稱和相同的設定建立另一個工作群組。
5. 選擇儲存變更。更新的工作群組會出現在 Workgroups (工作群組) 頁面上的清單。

## 檢視工作群組的詳細資訊

對於每個工作群組，您可以查看其詳細資訊。詳細資訊包括工作群組的名稱、描述、已啟用或停用，以及工作群組中執行的查詢所使用的設定，其中包括查詢結果的位置、預期的儲存貯體擁有者、加密和對寫入查詢結果儲存貯體之物件的控制。如果工作群組有資料用量限額，則也會顯示。

### 若要檢視工作群組的詳細資訊

1. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
2. 在 Workgroups (工作群組) 頁面上，請選擇您要檢視的工作群組連結。工作群組的 Overview Details (概觀詳細資料) 頁面隨即會顯示。

## 刪除工作群組

如果您有許可刪除工作群組，您可以這樣做。無法刪除主要工作群組。

如有許可，您可以隨時刪除空的工作群組。您也可以刪除包含儲存的查詢的工作群組。在這種情況下，在繼續刪除工作群組之前，Athena 會警告將刪除儲存的查詢。

如果您刪除自己所在的工作群組，主控台將焦點切換到主要工作群組。如果您有它的存取權，則可以執行查詢和檢視其設定。

如果您刪除工作群組，則其設定和每一查詢資料限制控制也會刪除。工作群組範圍的資料限制控制項會保留在中 CloudWatch，您可以視需要將其刪除。

### Important

在刪除工作群組之前，請確定其使用者也屬於能夠讓他們繼續執行查詢的其他工作群組。如果使用者的 IAM 政策僅允許他們在此工作群組中執行查詢，而且您刪除此工作群組，則他們再也沒有許可來執行查詢。如需詳細資訊，請參閱[Example policy for running queries in the primary workgroup](#)。

## 在主控台刪除工作群組

1. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
2. 在 Workgroups (工作群組) 頁面上，選取您要刪除的工作群組的按鈕。
3. 選擇 動作、刪除。

4. 在 Delete workgroup (刪除工作群組) 確認提示中，輸入工作群組的名稱，然後選擇 Delete (刪除)。

若要使用 API 操作來刪除工作群組，請使用 DeleteWorkGroup 動作。

### 切換工作群組

如果您對兩個工作群組都有許可，您可以從其中一個切換到另一個。

在每個工作群組內，您最多可以開啟十個查詢索引標籤。在工作群組之間切換時，您的查詢索引標籤最多可以在三個工作群組中保持開啟。

### 切換工作群組

1. 在 Athena 主控台中，使用右上角的 Workgroup (工作群組) 選項選擇工作群組。
2. 如果工作群組 *workgroup-name* 設定對話方塊出現時，請選擇 Acknowledge (確認)。

Workgroup (工作群組) 選項會顯示您已切換到其中的工作群組名稱。您現在可以在此工作群組中執行查詢。

### 在工作群組之間複製儲存的查詢

Athena 主控台目前沒有將儲存的查詢從某個工作群組直接複製到另一個工作群組的選項，但您可以使用下列程序手動執行相同的任務。

### 若要在工作群組之間複製儲存的查詢

1. 在 Athena 主控台中，從您想要從中複製查詢的工作群組中，選擇 Saved queries (儲存的查詢) 索引標籤。
2. 選擇您想要複製的已儲存查詢連結。Athena 會在查詢編輯器中開啟查詢。
3. 在查詢編輯器中，選取查詢文字，然後按下 **Ctrl+C** 以進行複製。
4. [切換](#)至目的地工作群組，或[建立工作群組](#)，然後切換至該工作群組。
5. 在查詢編輯器中開啟新索引標籤，然後按下 **Ctrl+V** 將文字貼到新索引標籤中。
6. 在查詢編輯器中，選擇 Save as (另存為)，將查詢儲存在目的地工作群組中。
7. 在 Choose a name (選擇名稱) 對話方塊中，輸入查詢的名稱和選擇性描述。
8. 選擇儲存。

## 啟用或停用工作群組

如果您有許可這樣做，您可以在主控台、使用 API 操作或透過 JDBC 和 ODBC 驅動程式來啟用或停用工作群組。

### 若要啟用或停用工作群組

1. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
2. 在 Workgroups (工作群組) 頁面上，請選擇工作群組的連結。
3. 在右上角，選擇 Enable workgroup (啟用工作群組) 或 Disable workgroup (停用工作群組)。
4. 在確認提示時，選擇 Enable (啟用) 或 Disable (停用)。如果您停用工作群組，則其使用者無法在其中執行查詢，或建立新的具名查詢。如果您啟用工作群組，使用者可以使用它來執行查詢。

### 指定工作群組在其中執行查詢

若要指定使用的工作群組，您必須具有此工作群組的許可。

### 指定工作群組使用

1. 請確認您的許可允許您在您想要使用在工作群組中執行查詢。如需詳細資訊，請參閱[the section called “用於存取工作群組的 IAM 政策”](#)。
2. 若要指定工作群組，請使用以下其中一個選項：
  - 如果您正在使用 Athena 主控台，請[切換工作群組](#)來設定工作群組。
  - 如果您是使用 Athena API 操作，請在 API 動作中指定工作群組名稱。例如，您可以在中設定工作群組名稱 [StartQueryExecution](#)，如下所示：

```
StartQueryExecutionRequest startQueryExecutionRequest = new
    StartQueryExecutionRequest()
        .withQueryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
        .withQueryExecutionContext(queryExecutionContext)
        .withWorkGroup(WorkgroupName)
```

- 如果您是使用 JDBC 或 ODBC 驅動程式，請在連線字串中使用 Workgroup 組態參數來設定工作群組名稱。驅動程式會將工作群組名稱傳遞給 Athena。在連線字串中指定工作群組參數，如下列範例所示：

```
jdbc:awsathena://AwsRegion=<AWSREGION>;UID=<ACCESSKEY>;
```

```
PWD=<SECRETKEY>;S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/<athena-  
output>-<AWSREGION>;  
Workgroup=<WORKGROUPNAME>;
```

## 為工作群組設定最低加密

身為 Athena SQL 工作群組的管理員，您可以在 Amazon S3 對工作群組的所有查詢結果強制執行最低層級的加密。您可以使用此功能，確保查詢結果絕不會儲存在處於未加密狀態的 Amazon S3 儲存貯體中。

若使用者位於啟用最低加密的工作群組中，則當其提交查詢時，只能將加密設定為您設定的最低層級，或將加密設定為較高層級 (如果有的話)。Athena 會在使用者執行查詢時，指定的層級或工作群組中設定的層級加密查詢結果。

以下為可用的層級：

- 基本 – 使用 Amazon S3 受管金鑰 (SSE-S3) 的 Amazon S3 伺服器端加密 (SSE\_S3)。
- 中級 – 使用 KMS 受管金鑰的伺服器端加密 (SSE\_KMS)。
- 進階 – 使用 KMS 受管金鑰的用戶端加密 (CSE\_KMS)。

## 考量與限制

- 最低加密功能不適用於已啟用 Apache Spark 的工作群組。
- 只有當工作群組未啟用 [覆寫用戶端設定](#) 選項時，最低加密功能才能正常運作。
- 如果工作群組已啟用覆寫用戶端設定選項，則會採用工作群組加密設定，且最低加密設定不會造成任何影響。
- 啟用此功能無需付費。

## 為工作群組啟用最低加密

您可以在建立或更新工作群組時，為 Athena SQL 工作群組的查詢結果啟用最低加密層級。若要這麼做，您可以使用 Athena 主控台、Athena API 或 AWS CLI。

## 使用 Athena 主控台啟用最低加密

若要開始使用 Athena 主控台建立或編輯工作群組，請參閱 [建立工作群組](#) 或 [編輯工作群組](#)。設定工作群組時，請使用下列步驟來啟用最低加密。

## 若要為工作群組查詢結果設定最低加密層級

1. 在其他組態區段中，展開設定。
2. 清除覆寫用戶端設定選項，或確認未選取該選項。
3. 在其他組態區段中，展開查詢結果組態。
4. 選取加密查詢結果選項。
5. 針對加密類型，請選取您希望 Athena 用於工作群組查詢結果的加密方法 (SSE\_S3、SSE\_KMS 或 CSE\_KMS)。這些加密類型對應至基本、中級和進階安全層級。
6. 若要針對所有使用者強制執行您選擇作為最低加密層級的加密方法，請選取將 ***encryption\_method*** 設定為最低加密。

選取此選項時，資料表會顯示，當您選擇的加密類型變為最低時，使用者將允許的加密階層和加密層級。

7. 建立工作群組或更新工作群組組態後，請選擇建立工作群組或儲存變更。

## 使用 Athena API 或啟 AWS CLI 用最低加密

當您使用 [CreateWorkGroup](#) 或 [UpdateWorkGroup](#) API 建立或更新 Athena SQL 工作群組時 `false`，請 [EnforceWorkGroupConfiguration](#) 將設定 [EnableMinimumEncryptionConfiguration](#) 為 `true`、並使用 [EncryptionOption](#) 來指定加密類型。

在中 AWS CLI，使用 [create-work-group](#) 或命 [update-work-group](#) 令搭配 `--configuration` 或 `--configuration-updates` 參數，並指定與 API 的選項相對應的選項。

## 使用啟用 IAM Identity Center 的 Athena 工作群組

的受信任身分傳播功能 AWS IAM Identity Center 可讓您的員工身分識別跨 AWS 分析服務使用。使用受信任的身分傳播，您不必執行服務特定身分提供者組態或 IAM 角色設定。

使用 IAM Identity Center，您可管理員工身分 (也稱為員工使用者) 的登入安全性。IAM 身分中心提供了一個位置，您可以在其中建立或連接員工使用者，並集中管理其所有 AWS 帳戶和應用程式的存取權限。您可使用多帳戶許可，將對 AWS 帳戶的存取權指派給這些使用者。您可使用應用程式指派功能，來指派使用者對啟用 IAM Identity Center 的應用程式、雲端應用程式和客戶安全性聲明標記語言 (SAML 2.0) 應用程式的存取權。如需詳細資訊，請參閱《AWS IAM Identity Center 使用者指南》中的 [跨應用程式的受信任身分傳播](#)。

目前，憑藉受信任身分傳播的 Athena SQL 支援，您可使用與 Amazon EMR Studio 相同的身分，以及 EMR Studio 中的 Athena SQL 介面。若要將 IAM Identity Center 身分與 EMR Studio 中的 Athena

SQL 搭配使用，您必須建立啟用 IAM Identity Center 的工作群組。然後，您可使用 IAM Identity Center 主控台或 API，將 IAM Identity Center 使用者或群組指派給啟用 IAM Identity Center 的 Athena 工作群組。從使用受信任身分傳播的 Athena 工作群組查詢，必須從啟用了 IAM Identity Center 的 EMR Studio 中的 Athena SQL 介面執行。

## 考量與限制

當您在 Amazon Athena 使用受信任的 Identity Center 時，請考慮下列幾點：

- 建立工作群組之後，無法變更工作群組的身分驗證方法。
  - 無法修改現有的 Athena SQL 工作群組來支援啟用了 IAM Identity Center 的工作群組。
  - 無法修改啟用了 IAM Identity Center 的工作群組，來支援資源層級 IAM 許可或以 Identity Center 為基礎的 IAM 政策。
- 若要存取啟用受信任身分傳播的工作群組，IAM 身分中心使用者必須指派給 Athena [GetWorkGroup](#) API 動作回應所傳回的工作群組。IdentityCenterApplicationArn
- Amazon S3 Access Grants 必須設定為使用受信任的身分傳播身分。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的 [S3 Access Grants 和公司目錄身分](#)。
- 啟用 IAM Identity Center 的 Athena 工作群組需要將 Lake Formation 設定為使用 IAM Identity Center 身分。如需組態資訊，請參閱《AWS Lake Formation 開發人員指南》中的 [整合 IAM Identity Center](#)。
- 依預設，在使用受信任身分傳播的工作群組中，查詢會在 30 分鐘後逾時。您可請求增加查詢逾時，但在受信任身分傳播工作群組中，可執行的查詢上限為一小時。
- 在受信任身分傳播工作群組中，使用者或群組權利變更可能需要最多一小時才會生效。
- 使用受信任身分傳播的 Athena 工作群組中的查詢，無法直接從 Athena 主控台執行。必須從啟用了 IAM Identity Center 的 EMR Studio 中的 Athena 介面執行。如需有關在 EMR Studio 中使用 Athena 的詳細資訊，請參閱《Amazon EMR 管理指南》中的在 [EMR Studio 中使用 Amazon Athena SQL 編輯器](#)。
- 受信任身分傳播與下列 Athena 功能不相容。
  - aws:CalledVia 內容索引鍵。
  - Athena for Spark 工作群組。
  - 對 Athena API 的聯合存取。
  - 使用 Lake Formation 以及 Athena JDBC 和 ODBC 驅動程式對 Athena 進行聯合存取。
- 您只能在下列情況下搭配 Athena 使用受信任的身分傳播 AWS 區域：
  - us-east-2 – 美國東部 (俄亥俄)

- us-east-1 – 美國東部 (維吉尼亞北部)
- us-west-1 – 美國西部 (加利佛尼亞北部)
- us-west-2 – 美國西部 (奧勒岡)
- af-south-1 – 非洲 (開普敦)
- ap-east-1 – 亞太區域 (香港)
- ap-southeast-3 – 亞太區域 (雅加達)
- ap-south-1 – 亞太區域 (孟買)
- ap-northeast-3 – 亞太區域 (大阪)
- ap-northeast-2 – 亞太區域 (首爾)
- ap-southeast-1 – 亞太區域 (新加坡)
- ap-southeast-2 – 亞太區域 (雪梨)
- ap-northeast-1 – 亞太區域 (東京)
- ca-central-1 – 加拿大 (中部)
- eu-central-1 – 歐洲 (法蘭克福)
- eu-west-1 – 歐洲 (愛爾蘭)
- eu-west-2 – 歐洲 (倫敦)
- eu-south-1 – 歐洲 (米蘭)
- eu-west-3 – 歐洲 (巴黎)
- eu-north-1 – 歐洲 (斯德哥爾摩)
- me-south-1 – 中東 (巴林)
- sa-east-1 – 南美洲 (聖保羅)

## 所需的許可

若管理員 IAM 使用者在 Athena 主控台中建立啟用了 IAM Identity Center 的工作群組，必須附加下列政策。

- AmazonAthenaFullAccess 受管政策。如需詳細資訊，請參閱 [AWS 受管理的策略：AmazonAthenaFullAccess](#)。
- 下列內嵌政策允許 IAM 和 IAM Identity Center 的以下動作：

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

```
"Statement": [
  {
    "Action": [
      "iam:createRole",
      "iam:CreatePolicy",
      "iam:AttachRolePolicy",
      "iam:ListRoles",
      "iam:PassRole",
      "identitystore:ListUsers",
      "identitystore:ListGroups",
      "identitystore:CreateUser",
      "identitystore:CreateGroup",
      "sso:ListInstances",
      "sso:CreateInstance",
      "sso>DeleteInstance",
      "sso:DescribeUser",
      "sso:DescribeGroup",
      "sso:ListTrustedTokenIssuers",
      "sso:DescribeTrustedTokenIssuer",
      "sso:ListApplicationAssignments",
      "sso:DescribeRegisteredRegions",
      "sso:GetManagedApplicationInstance",
      "sso:GetSharedSsoConfiguration",
      "sso:PutApplicationAssignmentConfiguration",
      "sso:CreateApplication",
      "sso>DeleteApplication",
      "sso:PutApplicationGrant",
      "sso:PutApplicationAuthenticationMethod",
      "sso:PutApplicationAccessScope",
      "sso:ListDirectoryAssociations",
      "sso:CreateApplicationAssignment",
      "sso>DeleteApplicationAssignment",
      "organizations:ListDelegatedAdministrators",
      "organizations:DescribeAccount",
      "organizations:DescribeOrganization",
      "organizations:CreateOrganization",
      "sso-directory:SearchUsers",
      "sso-directory:SearchGroups",
      "sso-directory:CreateUser"
    ],
    "Effect": "Allow",
    "Resource": [
      "*"
    ]
  }
]
```

```
}  
  ]  
}
```

## 建立啟用 IAM Identity Center 的 Athena 工作群組

下列程序顯示與建立啟用 IAM Identity Center 的 Athena 工作群組相關的步驟和選項。如需 Athena 工作群組可用的其他組態選項的說明，請參閱 [建立工作群組](#)。

在 Athena 主控台中建立啟用 SSO 的工作群組

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
3. 在 Workgroups (工作群組) 頁面中，請選擇 Create workgroup (建立工作群組)。
4. 在建立工作群組頁面上，針對工作群組名稱，輸入工作群組的名稱。
5. 針對分析引擎，使用 Athena SQL 預設值。
6. 針對身分驗證，選擇 IAM Identity Center。
7. 針對 IAM Identity Center 存取的服務角色，選擇現有的服務角色，或建立新的服務角色。

Athena 需要許可，才能代表您存取 IAM Identity Center。Athena 需要服務角色，才能執行此操作。服務角色是您管理的 IAM 角色，可授權 AWS 服務代表您存取其他 AWS 服務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [建立角色以將權限委派給 AWS 服務](#)。

8. 展開查詢結果組態，然後針對查詢結果的位置輸入或選擇 Amazon S3 路徑。
9. (選用) 選擇加密查詢結果。
10. (選用) 選擇建立以使用者身分為基礎的 S3 字首。

當您建立啟用 IAM Identity Center 的工作群組時，預設會選取啟用 S3 Access Grants 選項。您可使用 Amazon S3 Access Grants，來控制對 Amazon S3 中 Athena 查詢結果位置 (字首) 的存取權。如需有關 Amazon S3 Access Grants 的詳細資訊，請參閱 [使用 Amazon S3 Access Grants 管理存取權](#)。

在使用 IAM Identity Center 身分驗證的 Athena 工作群組中，您可建立由 Amazon S3 Access Grants 管控的以身分為基礎的查詢結果位置。藉助這些以使用者身分為基礎的 Amazon S3 字首，Athena 工作群組中的使用者可將查詢結果與相同工作群組中的其他使用者隔離。

啟用使用者字首選項時，Athena 會將使用者 ID 做為 Amazon S3 路徑字首附加至工作群組的查詢結果輸出位置 (例如 `s3://DOC-EXAMPLE-BUCKET/${user_id}`)。若要使用此功能，您必須將

Access Grants 設定為僅允許使用者存取具有 `user_id` 字首的位置。如需限制對 Athena 查詢結果存取權的 Amazon S3 存取授權位置角色政策範例，請參閱[角色政策範例](#)。

#### Note

選取使用者身分 S3 字首選項，會自動啟用工作群組的「覆寫用戶端設定」選項，如下一個步驟所述。「覆寫用戶端設定」選項是使用者身分字首功能的必要條件。

11. 展開設定，然後確認選取覆寫用戶端設定。

如果選取覆寫用戶端設定，則會在工作群組層級，強制工作群組中所有用戶端執行工作群組設定。如需詳細資訊，請參閱[工作群組設定覆寫用戶端設定](#)。

12. (選用) 如 [建立工作群組](#) 中所述，進行需要的任何其他組態設定。

13. 選擇建立工作群組。

14. 使用 Athena 主控台的「工作群組」區段，將您的 IAM 身分中心目錄中的使用者或群組指派給已啟用 IAM 身分中心的 Athena 工作群組。

## 角色政策範例

下列範例顯示將角色附加至 Amazon S3 存取授與位置的政策，該位置限制對 Athena 查詢結果的存取。

```
{
  "Statement": [{
    "Action": ["s3:*"],
    "Condition": {
      "ArnNotEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-grants/default"
      },
      "StringNotEquals": {
        "aws:ResourceAccount": "${account}"
      }
    },
    "Effect": "Deny",
    "Resource": "*",
    "Sid": "ExplicitDenyS3"
  }, {
    "Action": ["kms:*"],
    "Effect": "Deny",
```

```

    "NotResource": "arn:aws:kms:${region}:${account}:key/${keyid}",
    "Sid": "ExplicitDenyKMS"
  }, {
    "Action": ["s3:ListMultipartUploadParts", "s3:GetObject"],
    "Condition": {
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-
grants/default"
      },
      "StringEquals": {
        "aws:ResourceAccount": "${account}"
      }
    },
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::ATHENA-QUERY-RESULT-LOCATION/${identitystore:UserId}/
*",
    "Sid": "ObjectLevelReadPermissions"
  }, {
    "Action": ["s3:PutObject", "s3:AbortMultipartUpload"],
    "Condition": {
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-
grants/default"
      },
      "StringEquals": {
        "aws:ResourceAccount": "${account}"
      }
    },
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::ATHENA-QUERY-RESULT-LOCATION/${identitystore:UserId}/
*",
    "Sid": "ObjectLevelWritePermissions"
  }, {
    "Action": "s3:ListBucket",
    "Condition": {
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-
grants/default"
      },
      "StringEquals": {
        "aws:ResourceAccount": "${account}"
      },
      "StringLikeIfExists": {
        "s3:prefix": ["${identitystore:UserId}", "${identitystore:UserId}/*"]
      }
    }
  }

```

```
    }
  },
  "Effect": "Allow",
  "Resource": "arn:aws:s3:::ATHENA-QUERY-RESULT-LOCATION",
  "Sid": "BucketLevelReadPermissions"
}, {
  "Action": ["kms:GenerateDataKey", "kms:Decrypt"],
  "Effect": "Allow",
  "Resource": "arn:aws:kms:${region}:${account}:key/${keyid}",
  "Sid": "KMSPermissions"
}],
"Version": "2012-10-17"
}
```

## Athena 工作群組 API

以下是一些用於 Athena 工作群組的 REST API 操作。在以下所有操作中 (ListWorkGroups 除外)，您必須指定工作群組。在其他操作中，例如 StartQueryExecution，工作群組參數是選擇性，此處未列出這些操作。如需操作的完整清單，請參閱 [Amazon Athena API 參考](#)。

- [CreateWorkGroup](#)
- [DeleteWorkGroup](#)
- [GetWorkGroup](#)
- [ListWorkGroups](#)
- [UpdateWorkGroup](#)

## 工作群組疑難排解

使用以下秘訣進行工作群組疑難排解。

- 檢查您的帳戶中個別使用者的許可。他們必須有權存取查詢結果的位置，以及您們想要在其中執行查詢的工作群組。如果他們想要切換工作群組，則還需要有這兩個工作群組的許可。如需相關資訊，請參閱 [用於存取工作群組的 IAM 政策](#)。
- 請注意 Athena 主控台內容，了解您將在哪個工作群組中執行查詢。如果您使用驅動程式，請務必設定為您需要的工作群組。如需相關資訊，請參閱 [the section called “指定工作群組在其中執行查詢”](#)。

- 如果您使用 API 或驅動程式執行查詢，則必須使用下列其中一種方式指定查詢結果位置：針對個別查詢，請使用 [OutputLocation](#)(用戶端)。在工作群組中，使用[WorkGroupConfiguration](#)。如果未以任何方式指定位置，Athena 會在查詢執行期時發出錯誤。
- 如果您以工作群組設定覆寫用戶端設定，您可能會遇到查詢結果位置方面的錯誤。例如，工作群組的使用者可能沒有許可，無法在 Amazon S3 中的工作群組位置存放查詢結果。在這種情況下，請新增必要的許可。
- 工作群組會對 API 操作的行為造成改變。您帳戶中的使用者在 IAM 中，對工作群組需要有以資源為基礎的許可，才能在這些工作群組中呼叫以下現有的 API 操作。如果沒有工作群組和工作群組動作的權限，則下列 API 動作會擲回 `AccessDeniedException`：CreateNamedQueryDeleteNamedQueryGetNamedQueryListNamedQuery和 GetQueryResultsStream(此 API 動作僅適用於驅動程式使用，否則不會公開供公眾使用)。如需詳細資訊，請參閱《服務授權參考》中的 [Amazon Athena 的動作、資源和條件索引鍵](#)。

對BatchGetQueryExecution和 BatchGetNamedQueryAPI 作業的呼叫僅會傳回在使用者可存取的工作群組中執行之查詢的相關資訊。如果使用者無法存取此工作群組，這些 API 操作會在未處理的 ID 清單中傳回未經授權的查詢 ID。如需詳細資訊，請參閱[the section called “Athena 工作群組 API”](#)。

- 如果執行查詢的工作群組已設定有[強制的查詢結果位置](#)，請勿為 CTAS 查詢指定 `external_location`。在這種情況下，Athena 會發出錯誤，且指定 `external_location` 的查詢會失敗。例如，如果您覆寫用戶端的查詢結果位置設定，而強制工作群組使用其自己的位置，此查詢會失敗：`CREATE TABLE <DB>.<TABLE1> WITH (format='Parquet', external_location='s3://DOC-EXAMPLE-BUCKET/test/') AS SELECT * FROM <DB>.<TABLE2> LIMIT 10;`

您可能會看到下列錯誤。下表列出工作群組的一些相關錯誤和建議的解決方案。

### 工作群組錯誤

錯誤	發生時情況...
查詢狀態為 CANCELED (已取消)。超出掃描的位元組限制。	查詢達到每一查詢資料限制，並且被取消。請考慮重寫查詢，以讀取較少的資料，或聯絡您的帳戶管理員。
<pre>###:ARN: aw: IAM:: 123456789012: # ##/ABC #####:###:###:ARN: AW: # ###:#### 1:123456789012: ####/#### # # StartQueryExecution</pre>	使用者在工作群組中執行查詢，但沒有它的存取權。更新您的政策以提供此工作群組的存取權。

錯誤	發生時情況...
無效 _ 輸入。 WorkGroup <name>已停用。	使用者在工作群組中執行查詢，但此工作群組已停用。管理員可能停用您的工作群組。也可能是您沒有它的存取權。在這兩種情況下，請聯絡有權修改工作群組的管理員。
無效 _ 輸入。 WorkGroup <name>找不到。	使用者在工作群組中執行查詢，但此工作群組不存在。如果已刪除工作群組，則可能發生此情況。切換到另一個工作群組來執行您的查詢。
InvalidRequestException: 呼叫 StartQueryExecution 作業時：未提供輸出位置。需要透過工作群組結果組態設定或作為 API 輸入來提供輸出位置。	使用者以 API 執行查詢，但未指定查詢結果的位置。您必須使用下列兩種方式之一來設定查詢結果的輸出位置：針對個別查詢、使用 <a href="#">OutputLocation</a> (用戶端) 或在工作群組中使用 <a href="#">WorkGroup Configuration</a> 。
Create Table As Select 查詢失敗，因為該查詢已使用 'external_location' 屬性提交至 Athena 工作群組，而該工作群組會針對所有查詢強制執行集中輸出位置。請移除 'external_location' 屬性，然後重新提交查詢。	如果執行查詢的工作群組已設定 <a href="#">強制的查詢結果位置</a> ，而且您為 CTAS 查詢指定 external_location。在這種情況下，請移除 external_location 並重新執行查詢。
無法建立預備陳述式 <i>prepared_statement_name</i> 。此工作群組中的預備陳述式數目超過 1000 的上限。	工作群組包含的預備陳述式超過 1000 的上限。若要解決此問題，請使用 <a href="#">DEALLOCATE PREPARE</a> 以從工作群組中移除一或多個預備陳述式。或者，建立新的工作群組。

## 使用指標和事件控制成本並 CloudWatch 監控查詢

工作群組可讓您設定每個查詢或每個工作群組的資料使用控制限制、在超過這些限制時設定警示，以及將查詢量度發佈至 CloudWatch。

在每個工作群組中，您可以：

- 設定每一查詢和每一工作群組的 Data usage controls (資料用量控制)，並建立當查詢超出閾值時將採取的動作。

- 檢視和分析查詢量度，並將其發佈至 CloudWatch。如果您在主控台中建立工作群組，則會為您選取用 CloudWatch 於將指標發佈至的設定。如果您使用 API 操作，您必須[啟用發布指標](#)。發布指標時，其會顯示在 Workgroups (工作群組) 面板中的 Metrics (指標) 索引標籤。對於主要工作群組，預設會停用指標。

## 影片

以下視訊說明如何建立自訂儀表板，以及如何在主設定量度的警示和觸發器 CloudWatch。您可以直接從 Athena 主控台使用預先填入的儀表板，以使用這些查詢指標。

### [使用 Amazon 監控亞馬遜雅典娜 CloudWatch](#)

#### 主題

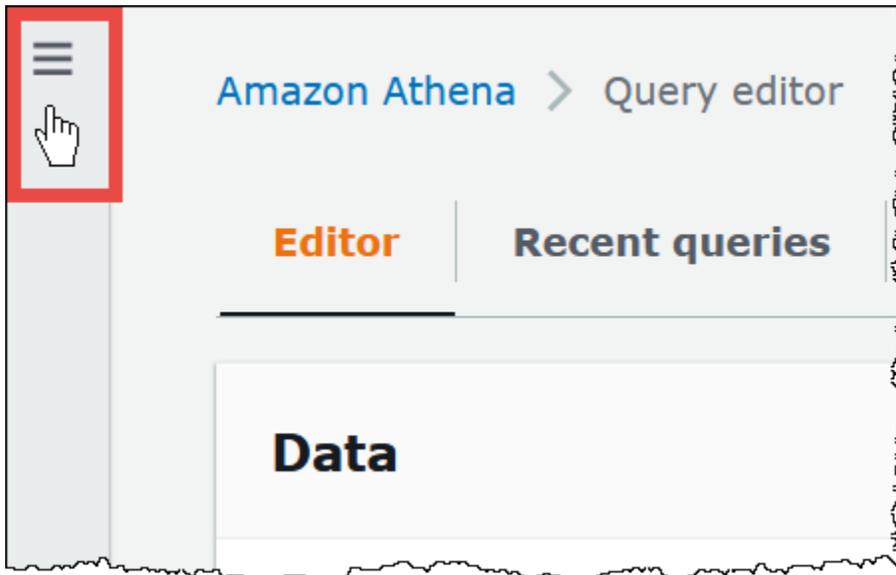
- [啟用 CloudWatch 查詢指標](#)
- [使用 CloudWatch 指標監控 Athena 查詢](#)
- [使用 Amazon EventBridge 事件監控 Athena 查](#)
- [監控 Athena 用量指標](#)
- [設定資料用量控制限制](#)

#### 啟用 CloudWatch 查詢指標

在主控台中建立工作群組時，依預設會選取用於發佈查詢指標的設定。CloudWatch

若要在 Athena 主控台中啟用或停用工作群組的查詢指標

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中，選擇 Workgroups (工作群組)。
4. 選擇您想要修改的工作群組連結。
5. 在工作群組的詳細資訊頁面上，選擇 Edit (編輯)。
6. 在「設定」區段中，選取或清除「將查詢量度發佈至」 AWS CloudWatch。

如果您使用 API 作業、命令列介面或具有 JDBC 驅動程式的用戶端應用程式來建立工作群組，以啟用發佈查詢量度，請 `PublishCloudWatchMetricsEnabled` 將設定為 `true` 中。[WorkGroupConfiguration](#) 以下範例只顯示指標組態，而省略其他組態：

```
"WorkGroupConfiguration": {  
  "PublishCloudWatchMetricsEnabled": "true"  
  ....  
}
```

### 使用 CloudWatch 指標監控 Athena 查詢

選取將查詢指標發佈到 CloudWatch 選項後 CloudWatch，Athena 會將[查詢相關指標發佈至](#) Amazon。您可以建立自訂儀表板、在中設定指標的警示和觸發器 CloudWatch，或直接從 Athena 主控台使用預先填入的儀表板。

當您為工作群組中的查詢啟用查詢指標時，Athena 主控台中每個工作群組的指標會顯示在 Workgroups (工作群組) 面板中的 Metrics (指標) 索引標籤中。

Athena 會將下列指標發佈至主 CloudWatch 控制台：

- `DPUAllocated` – 在容量保留中佈建的用於執行查詢的 DPU (資料處理單位) 總數。
- `DPUConsumed` – 在指定的時間，容量保留中處於 `RUNNING` 狀態的查詢主動使用的 DPU 數目。僅當工作群組與容量保留相關聯，並包含與保留關聯的所有工作群組時，才會發出指標。
- `DPUCount` – 查詢使用的 DPU 數目上限，僅當查詢完成時發布一次。
- `EngineExecutionTime` – 執行查詢所花費的毫秒數。
- `ProcessedBytes` – Athena 在每次 DML 查詢所掃描的位元組。
- `QueryPlanningTime` – Athena 規劃查詢處理流程所花費的毫秒數。
- `QueryQueueTime` – 查詢在查詢佇列中等待資源的毫秒數。
- `ServicePreProcessingTime` – 提交查詢至查詢引擎之前，Athena 預先處理查詢所花費的毫秒數。
- `ServiceProcessingTime` – 查詢引擎完成查詢的執行後，Athena 處理查詢結果所花費的毫秒數。
- `TotalExecutionTime` – Athena 執行 DDL 或 DML 查詢所花費的毫秒數。

如需更完整的描述，請參閱本文件稍後的 [Athena 的 CloudWatch 度量和維度清單](#)。

這些指標具有下列維度：

- `CapacityReservation`– 用於執行查詢的容量保留名稱 (如果適用)。
- `QueryState` – `SUCCEEDED`、`FAILED` 或 `CANCELED`
- `QueryType` – `DML`、`DDL` 或 `UTILITY`
- `WorkGroup` – 工作群組的名稱

Athena 會將下列指標發佈至命 `AmazonAthenaForApacheSpark` 名空間下的 CloudWatch 主控台：

- `DPUCount` – 工作階段期間用來執行計算所消耗的 DPU 數目。

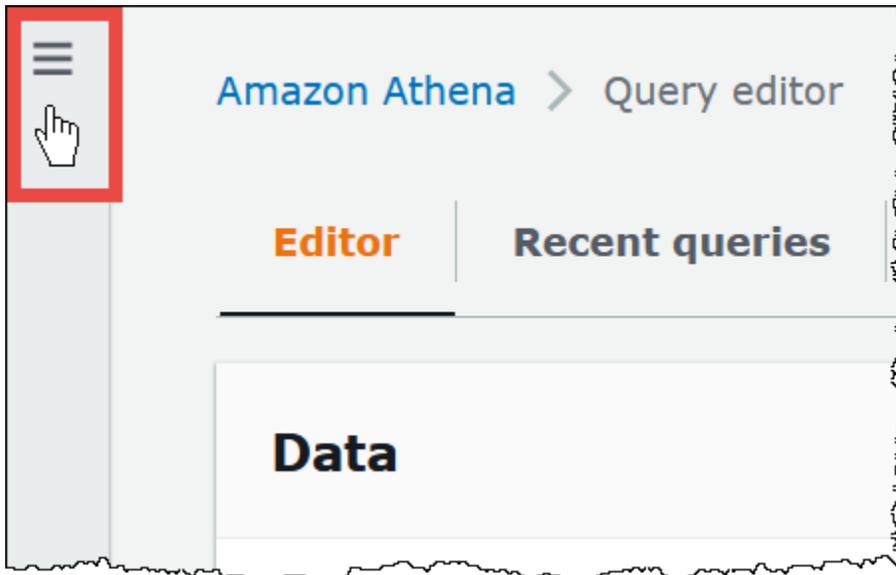
該指標具有下列兩個維度：

- `SessionId` – 要提交計算的工作階段 ID。
- `WorkGroup` – 工作群組的名稱。

如需詳細資訊，請參閱本主題稍後的 [Athena 的 CloudWatch 度量和維度清單](#)。如需有關 Athena 用量指標的資訊，請參閱 [監控 Athena 用量指標](#)。

## 在主控台檢視工作群組的查詢指標

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



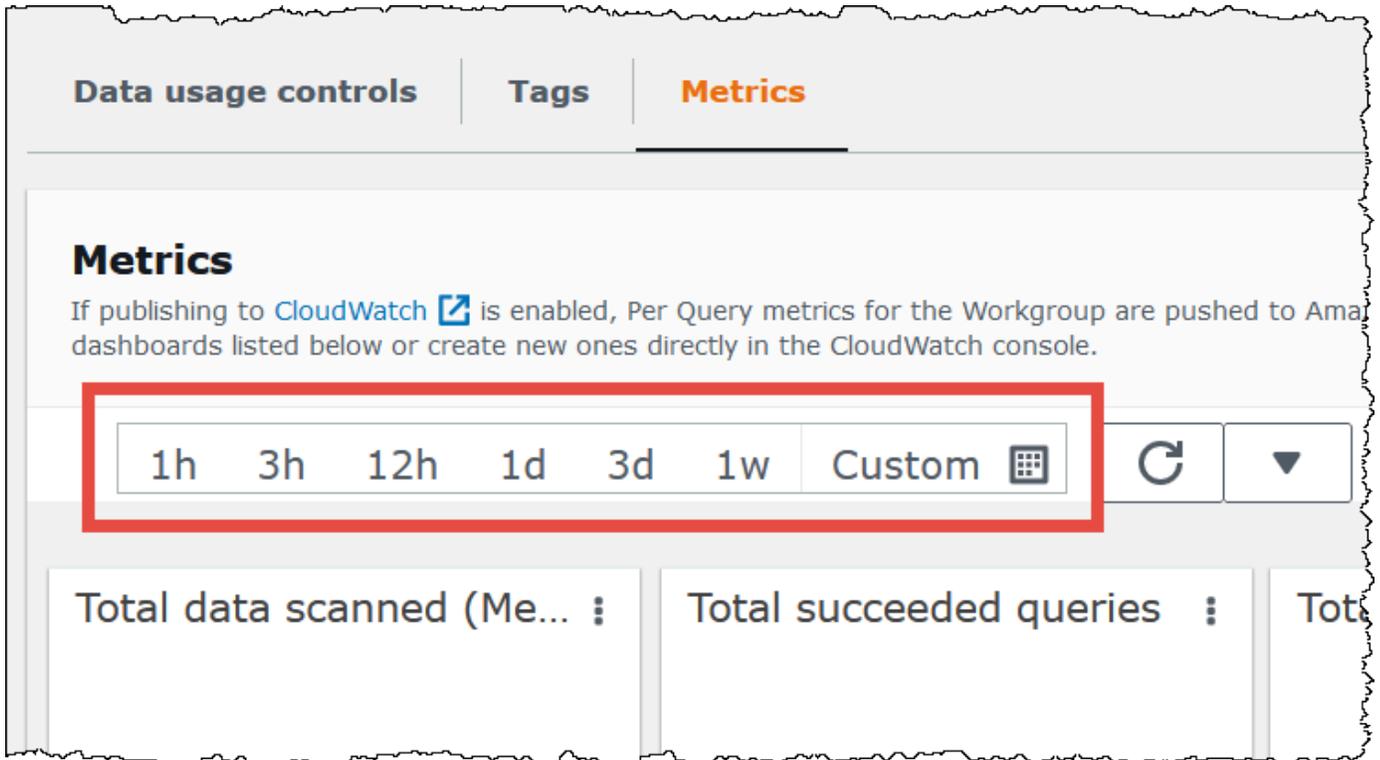
3. 在導覽窗格中，選擇 Workgroups (工作群組)。
4. 從清單中選擇所需的工作群組，然後選擇 Metrics (指標) 索引標籤。

指標儀表板隨即顯示。

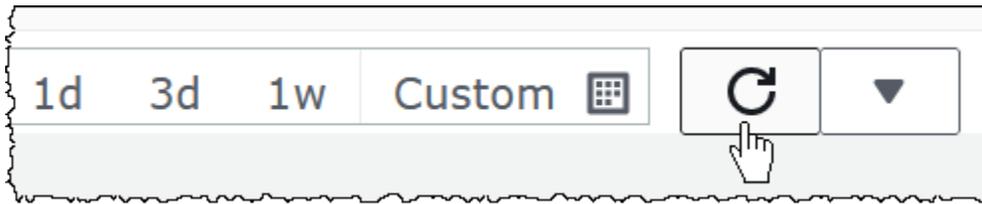
### Note

如果您最近剛啟用工作群組的指標，且/或最近沒有任何查詢活動，儀表板上的圖形可能會是空白的狀態。系統會 CloudWatch 根據您在下一個步驟中指定的間隔擷取查詢活動。

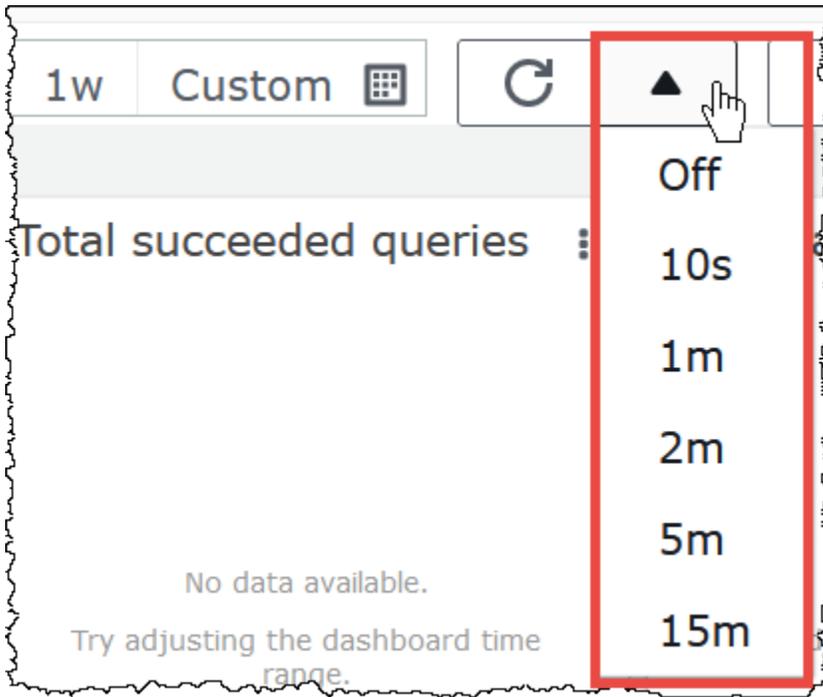
5. 在「量度」區段中，選擇 Athena 用來擷取查詢指標的度量間隔 CloudWatch，或指定自訂間隔。



- 若要重新整理顯示的指標，請選擇重新整理圖示。



- 按一下重新整理圖示旁的箭頭，以選擇您希望指標顯示的更新頻率。



若要在 Amazon CloudWatch 主控台中檢視指標

1. [請在以下位置開啟 CloudWatch 主控台。](https://console.aws.amazon.com/cloudwatch/) <https://console.aws.amazon.com/cloudwatch/>
2. 在導覽窗格中，選擇 Metrics (指標)、All metrics (所有指標)。
3. 選取 AWS/Athena 命名空間。

使用 CLI 來檢視指標

- 執行以下任意一項：
  - 若要列出 Athena 的指標，請開啟命令提示，然後使用下列命令：

```
aws cloudwatch list-metrics --namespace "AWS/Athena"
```

- 若要列出所有可用的指標，請使用以下命令：

```
aws cloudwatch list-metrics"
```

## Athena 的 CloudWatch 度量和維度清單

如果您已在 Athena 中啟用 CloudWatch 指標，則會將下列指標傳送給 CloudWatch 每個工作群組。下列指標使用 AWS/Athena 命名空間。

指標名稱	描述
DPUAllocated	在容量保留中佈建的用於執行查詢的 DPU (資料處理單位) 總數。
DPUConsumed	在給定的時間，保留區中處於 RUNNING 狀態的查詢主動使用的 DPU 數目。僅當工作群組與容量保留相關聯並包含與保留關聯的所有工作群組時，才會發出這個指標。如果您將工作群組從一個保留區移至另一個保留區，則該指標會包含該工作群組屬於第一個保留區時的資料。如需有關容量保留的詳細資訊，請參閱 <a href="#">管理查詢處理容量</a> 。
DPUCount	查詢使用的 DPU 數目上限，僅當查詢完成時發布一次。只有附加至容量保留的工作群組才會發出此指標。
EngineExecutionTime	查詢執行所花費的毫秒數。
ProcessedBytes	Athena 在每次 DML 查詢所掃描的位元組。對於取消的查詢 (無論是由使用者取消，或達到上限時自動取消)，這包括取消前掃描的資料量。DDL 查詢不會報告此指標。
QueryPlanningTime	Athena 規劃查詢處理流程所花費的毫秒數。這包括從資料來源擷取資料表分割區所花費的時間。請注意，由於查詢引擎會執行查詢計劃，因此查詢計劃時間是的子集 EngineExecutionTime。
QueryQueueTime	查詢在查詢佇列中等待資源的毫秒數。請注意，如果發生暫時性錯誤，查詢可能自動加回到佇列。
ServicePreProcessingTime	提交查詢至查詢引擎之前，Athena 預先處理查詢所花費的毫秒數。
ServiceProcessingTime	查詢引擎完成查詢的執行後，Athena 處理查詢結果所花費的毫秒數。

指標名稱	描述
TotalExecutionTime	Athena 執行 DDL 或 DML 查詢所花費的毫秒數。TotalExecutionTime 包括 QueryQueueTime、QueryPlanningTime、EngineExecutionTime、和 ServiceProcessingTime。

Athena 的這些指標具有下列維度。

維度	描述
CapacityReservation	用於執行查詢的容量保留名稱 (如果適用)。當未使用容量保留時，此維度不會傳回任何資料。
QueryState	查詢狀態。  有效的統計資訊：已成功、已失敗、已取消。
QueryType	查詢類型。  有效的統計資訊：DDL、DML 或 UTILITY。執行的查詢陳述式類型。DDL 表示 DDL (資料定義語言) 查詢陳述式。DML 表示 DML (資料處理語言) 查詢陳述式，例如 CREATE TABLE AS SELECT。UTILITY 表示除 DDL 和 DML 以外的查詢陳述式，例如 SHOW CREATE TABLE 或 DESCRIBE TABLE。
WorkGroup	工作群組的名稱。

## 使用 Amazon EventBridge 事件監控 Athena 查

您可以 EventBridge 將 Amazon Athena 與 Amazon 搭配使用，以接收有關查詢狀態的即時通知。當您提交轉換狀態的查詢時，Athena 會發佈事件，其中 EventBridge 包含該查詢狀態轉換的相關資訊。您可以針對感興趣的事件撰寫簡單的規則，並在事件符合規則時採取自動化動作。例如，您可以建立在查詢到達終端機狀態時叫用 AWS Lambda 函數的規則。盡可能發出事件。

在您為 Athena 建立事件規則之前，請執行下列動作：

- 熟悉中的事件、規則和目標。EventBridge 有關更多信息，請參閱 [什麼是 Amazon EventBridge？](#) 如需如何設定規則的詳細資訊，請參閱 [Amazon 入門 EventBridge](#)。

- 建立要在事件規則中使用的一或多個目標。

### Note

Athena 目前提供一種事件類型 (Athena 查詢狀態變更)，但可能會新增其他事件類型和詳細資訊。如果您是透過程式設計方式將事件 JSON 資料還原序列化，請確定您的應用程式在新增這些額外屬性時已準備好處理不明屬性。

## Athena 事件格式

以下是 Amazon Athena 事件的基本模式。

```
{
  "source": [
    "aws.athena"
  ],
  "detail-type": [
    "Athena Query State Change"
  ],
  "detail": {
    "currentState": [
      "SUCCEEDED"
    ]
  }
}
```

## Athena 查詢狀態變更事件

下列範例會顯示 `currentState` 值為 `SUCCEEDED` 的 Athena 狀態變更事件。

```
{
  "version": "0",
  "id": "abcdef00-1234-5678-9abc-def012345678",
  "detail-type": "Athena Query State Change",
  "source": "aws.athena",
  "account": "123456789012",
  "time": "2019-10-06T09:30:10Z",
  "region": "us-east-1",
  "resources": [
```

```

    ],
    "detail":{
      "versionId":"0",
      "currentState":"SUCCEEDED",
      "previousState":"RUNNING",
      "statementType":"DDL",
      "queryExecutionId":"01234567-0123-0123-0123-012345678901",
      "workgroupName":"primary",
      "sequenceNumber":"3"
    }
  }
}

```

下列範例會顯示 `currentState` 值為 `FAILED` 的 Athena 狀態變更事件。只有在 `currentState` 為 `FAILED` 時，才會顯示 `athenaError` 區塊。如需有關 `errorCategory` 和 `errorType` 值的詳細資訊，請參閱 [Athena 錯誤目錄](#)。

```

{
  "version":"0",
  "id":"abcdef00-1234-5678-9abc-def012345678",
  "detail-type":"Athena Query State Change",
  "source":"aws.athena",
  "account":"123456789012",
  "time":"2019-10-06T09:30:10Z",
  "region":"us-east-1",
  "resources":[
  ],
  "detail":{
    "athenaError": {
      "errorCategory": 2.0, //Value depends on nature of exception
      "errorType": 1306.0, //Type depends on nature of exception
      "errorMessage": "Amazon S3 bucket not found", //Message depends on nature
of exception
      "retryable":false //Retryable value depends on nature of exception
    },
    "versionId":"0",
    "currentState": "FAILED",
    "previousState": "RUNNING",
    "statementType":"DML",
    "queryExecutionId":"01234567-0123-0123-0123-012345678901",
    "workgroupName":"primary",
    "sequenceNumber":"3"
  }
}

```

```
}
```

## 輸出屬性

JSON 輸出包含以下屬性。

屬性	描述
<code>athenaError</code>	只有在 <code>currentState</code> 為 FAILED 時才會顯示。包含發生之錯誤的相關資訊，包括錯誤類別、錯誤類型、錯誤訊息，以及是否可以重試導致錯誤的動作。每個欄位的值視錯誤性質而定。如需有關 <code>errorCategory</code> 和 <code>errorType</code> 值的詳細資訊，請參閱 <a href="#">Athena 錯誤目錄</a> 。
<code>versionId</code>	詳細資訊物件之結構描述的版本號碼。
<code>currentState</code>	在事件發生時查詢所轉換成的狀態。
<code>previousState</code>	在事件發生時查詢從其轉換的狀態。
<code>statementType</code>	執行的查詢陳述式類型。
<code>queryExecutionId</code>	查詢執行的唯一識別碼。
<code>workgroupName</code>	執行查詢的工作群組名稱。
<code>sequenceNumber</code>	單調遞增的數字，允許刪除重複和排序涉及所執行單一查詢的傳入事件。當針對相同的狀態轉換發布重複事件時， <code>sequenceNumber</code> 值是相同的。當查詢經歷一次以上的狀態轉換時 (例如遇到鮮少列入佇列的查詢)，您可以使用 <code>sequenceNumber</code> 來排序 <code>currentState</code> 和 <code>previousState</code> 值相同的事件。

## 範例

下列範例會將事件發布至您已訂閱的 Amazon SNS 主題。查詢 Athena 時，您會收到一封電子郵件。該範例假設 Amazon SNS 主題存在且您已訂閱該主題。

## 若要將 Athena 事件發布至 Amazon SNS 主題

1. 建立 Amazon SNS 主題的目標。授與 EventBridge 事件服務主體發佈到 Amazon SNS 主題的 `events.amazonaws.com` 權限，如下列範例所示。

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "arn:aws:sns:us-east-1:111111111111:your-sns-topic"
}
```

2. 使用命 AWS CLI `events put-rule` 令建立 Athena 事件的規則，如下列範例所示。

```
aws events put-rule --name {ruleName} --event-pattern '{"source": ["aws.athena"]}'
```

3. 使用 AWS CLI `events put-targets` 命令將 Amazon SNS 主題目標附加至規則，如下列範例所示。

```
aws events put-targets --rule {ruleName} --targets Id=1,Arn=arn:aws:sns:us-east-1:111111111111:your-sns-topic
```

4. 查詢 Athena 並觀察被叫用的目標。您應該會收到來自該 Amazon SNS 主題的相應電子郵件。

## 使 AWS 使用者通知用 Amazon Athena

您可使用 [AWS 使用者通知](#) 來設定交付管道，以取得有關 Amazon Athena 事件的通知。當事件符合您指定的規則時，便會收到通知。您可以透過多個管道接收事件通知，包括電子郵件、[AWS Chatbot](#) 聊天通知或 [AWS Console Mobile Application](#) 推送通知。您也可以可以在「[主控台通知中心](#)」中查看通知。使用者通知 支援彙總，可減少您在特定事件期間收到的通知數量。

如需詳細資訊，請參閱《AWS 使用者通知 使用者指南》。

## 監控 Athena 用量指標

您可以透過在 CloudWatch 圖形和儀表板上顯示目前的服務 CloudWatch 使用情況，使用使用量指標來提供您帳戶使用資源的可見度。

對於 Athena，使用量可用性指標會對應於 Athena 的 AWS 服務配額。您可以設定警示，在您的用量接近服務配額時發出警示。如需有關 Athena 服務配額的詳細資訊，請參閱 [Service Quotas](#)。如需 AWS 使用量指標的詳細資訊，請參閱 Amazon [AWS 使用 CloudWatch 者指南中的使用量指標](#)。

Athena 在 AWS/Usage 命名空間中發布下列指標。

指標名稱	描述
ResourceCount	<p>AWS 區域 每個帳戶的所有佇列和執行查詢的總和，以查詢類型 (DML 或 DDL) 分隔。最大值是此測量結果唯一有用的統計資料。</p> <p>此指標每分鐘定期發布一次。如果您沒有執行任何查詢，則指標不會報告任何內容 (甚至不會報告 0)。僅當採用指標時正在執行啟用查詢時，指標才會發布。</p>

下列維度用於強化 Athena 所發布的用量指標。

維度	描述
Service	AWS 服務 包含資源的名稱。對 Athena 而言，此維度的數值為 Athena。
Resource	正在執行的資源類型。Athena 查詢用量的資源值為 ActiveQueryCount。
Type	正在報告的實體類型。目前，Athena 用量指標的唯一有效值為 Resource。
Class	正在追蹤的資源類別。對 Athena 而言，Class 可以是 DML 或 DDL。

在 CloudWatch 主控台中檢視 Athena 資源使用量度

您可以使用 CloudWatch 主控台查看 Athena 使用量指標圖表，並設定警示，以便在使用量接近服務配額時提醒您。

## 檢視 Athena 資源用量指標

1. [請在以下位置開啟 CloudWatch 主控台。](https://console.aws.amazon.com/cloudwatch/) <https://console.aws.amazon.com/cloudwatch/>
2. 在導覽窗格中，選擇 Metrics (指標)、All metrics (所有指標)。
3. 選擇用量，然後選擇依據 AWS 資源。

服務配額用量指標清單隨即出現。

4. 選取 Athena 和旁邊的核取方塊 ActiveQueryCount。
5. 選擇 Graphed metrics (圖表化指標) 標籤。

上圖顯示您目前的 AWS 資源使用量。

如需將服務配額新增至圖形，以及設定在接近服務配額時通知您的警示的相關資訊，請參閱 Amazon CloudWatch 使用者指南中的[視覺化服務配額和設定警示](#)。如需設定每個工作群組用量限制的詳細資訊，請參閱[設定資料用量控制限制](#)。

### 設定資料用量控制限制

Athena 可讓您設定兩種類型的成本控制：每一查詢限制和每一工作群組限制。對於每個工作群組，您只能設定一個每一查詢限制和多個每一工作群組限制。

- 每一查詢控制限制指定每次查詢所掃描的資料總量。如果工作群組中執行的任何查詢超過此限制，則會取消。您在工作群組中只能建立一個每一查詢控制限制，它會套用到其中執行的每個查詢。如果您需要變更限制，請編輯限制。如需詳細步驟，請參閱[建立每一查詢資料用量控制](#)。
- 工作群組整體資料用量控制限制指定在指定期間內針對此工作群組中執行的所有查詢所掃描的資料總量。您可以為每一工作群組建立多個限制。工作群組整體查詢限制可讓您針對工作群組中執行的查詢所掃描的資料，在每小時或每日彙總上設定多個閾值。

如果掃描的資料總量超過閾值，您可以將通知推送至 Amazon SNS 主題。若要這麼做，請在 Athena 主控台中設定 Amazon SNS 警示和動作，以便在超過限制時通知管理員。如需詳細步驟，請參閱[建立每一工作群組資料用量控制](#)。您也可以針對 Athena 從 CloudWatch 主控台發佈的任何指標建立警示和動作。例如，您可以對失敗的查詢次數設定提醒。如果次數超過特定閾值，這個提醒可以觸發將電子郵件寄給管理員。如果超過限制，將會有動作將 Amazon SNS 警示通知傳送給指定的使用者。

您可以採取的其他動作：

- 調用 Lambda 函數。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的[使用 Amazon SNS 通知叫用 Lambda 函數](#)。
- 停用工作群組，以停止執行任何進一步的查詢。如需這些步驟，請參閱[啟用或停用工作群組](#)。

每一查詢限制和每一工作群組限制彼此獨立。每當超過任一限制時就會採取指定的動作。如果兩個或更多使用者在同一個工作群組中同時執行查詢，則可能每個查詢都不超過任何指定的限制，但掃描的資料總數超過每一工作群組的資料用量限制。在這種情況下，Amazon SNS 警示會傳送給使用者。

### 建立每一查詢資料用量控制

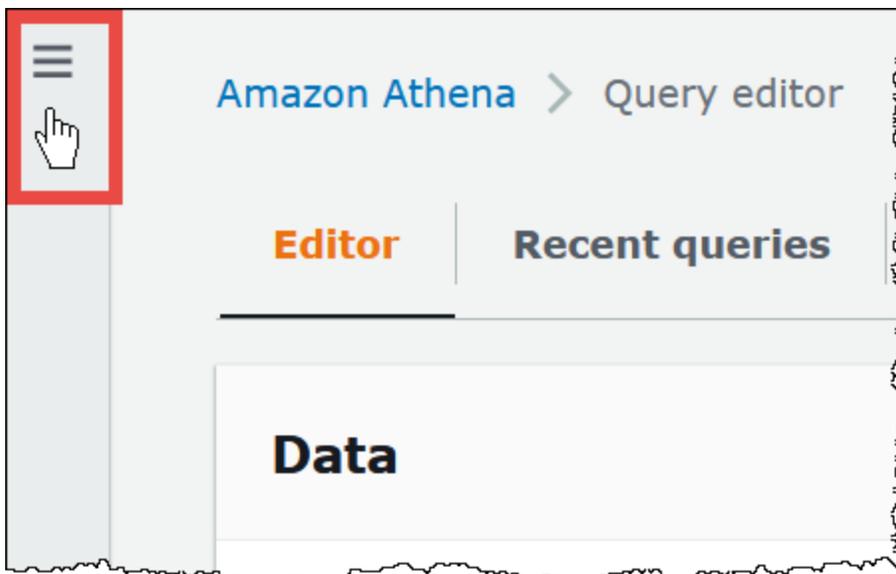
每一查詢控制限制指定每次查詢所掃描的資料總量。如果工作群組中執行的任何查詢超過此限制，則會取消。已取消的查詢會按照 [Amazon Athena 定價](#) 計費。

#### Note

如果是已取消或失敗的查詢，Athena 可能已將局部結果寫入 Amazon S3。在這種情況下，Athena 不會從存放結果的 Amazon S3 字首中刪除這些局部結果。您必須移除含有局部結果的 Amazon S3 字首。Athena 使用 Amazon S3 分段上傳，將資料寫入 Amazon S3。我們建議您設定儲存貯體生命週期政策，指定當查詢失敗時結束分段上傳。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [使用儲存貯體生命週期政策中止未完成的分段上傳](#)。

您在工作群組中只能建立一個每一查詢控制限制，它會套用到其中執行的每個查詢。如果您需要變更限制，請編輯限制。

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中，選擇 Workgroups (工作群組)。

4. 從清單中選擇工作群組名稱。
5. 在 Data usage controls (資料用量控制) 索引標籤的 Per query data usage control (每一查詢資料用量控制) 區段中，選擇 Manage (管理)。
6. 在 Manage per query data usage control (管理每一查詢資料用量控制) 頁面上，指定下列值：
  - 對於 Data limit (資料限制)，指定 10 MB (最小) 和 7 EB (最大) 之間的值。

 Note

對於工作群組內的資料用量控制，主控台會強制這些限制。它們不代表 Athena 中的任何查詢限制。

- 對於單位，請從下拉式清單中選擇單位值 (例如 KB 或 EB)。

預設動作是在查詢超過限制時取消查詢。此設定無法變更。

7. 選擇儲存。

### 建立或編輯每一工作群組資料用量提醒

當工作群組中執行的查詢在特定時段內掃描指定數量的資料時，您可以設定多個提醒閾值。提醒是使用 Amazon CloudWatch 警示實作，並套用至工作群組中的所有查詢。達到閾值後，您可以讓 Amazon SNS 向您指定的使用者傳送電子郵件。達到閾值時，查詢作業不會自動取消。

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 在導覽窗格中，選擇 Workgroups (工作群組)。
4. 從清單中選擇工作群組名稱。
5. 選擇 Edit (編輯) 以編輯工作群組的設定。
6. 向下捲動到並展開 Workgroup data usage alerts - optional (工作群組資料用量提醒 - 選用)。
7. 選擇 Add alert (新增提醒)。
8. 針對 Data usage threshold configuration (資料用量閾值組態)，指定值，如下所示：
  - 針對 Data threshold (資料閾值)，指定數字，然後從下拉式清單中選取單位值。
  - 對於 Time period (期間)，請從下拉式清單中選擇期間。

- 針對 SNS topic selection (SNS 主題選擇)，請從下拉式清單中選擇 Amazon SNS 主題。或者，選擇 Create SNS topic (建立 SNS 主題)，直接前往 [Amazon SNS 主控台](#)，建立 Amazon SNS 主題，並為 Athena 帳戶中的其中一個使用者設定主題訂閱。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 [Amazon SNS 入門](#)。

9. 如果您正在建立新提醒，請選擇 Add alert (新增提醒)，或選擇 Save (儲存) 來儲存現有提醒。

## 管理查詢處理容量

您可以使用容量保留來為您在 Athena 中執行的查詢獲取專用處理容量。透過容量保留，您可以利用工作負載管理功能，協助您優先處理序、控制和擴展最重要的互動式工作負載。例如，您可以隨時新增容量以增加可同時執行的查詢數量、控制哪些工作負載可以使用該容量，以及在工作負載之間共用容量。容量由 Athena 完全管理，並按您的要求為您預留。設定非常簡單，並且不需要變更 SQL 陳述式。

若要取得查詢的處理能力，您可建立容量保留、指定所需的資料處理單位 (DPU) 數目，然後將一或多個工作群組指派給保留。

當您使用容量保留時，工作群組扮演著重要角色。工作群組可讓您將查詢整理為邏輯分組。透過容量保留，您可以選擇性地將容量指派給工作群組，以便控制每個工作群組的查詢行為方式以及計費方式。如需有關工作群組的詳細資訊，請參閱 [使用工作群組來控制查詢存取和成本](#)。

將工作群組指派至保留，可讓您優先處理提交至指派的工作群組的查詢。例如，您可以將容量分配給用於時間敏感型財務報告查詢的工作群組，以將這些查詢與其他工作群組中較不重要的查詢隔離。這樣會啟用關鍵工作負載的一致查詢執行，同時允許其他工作負載獨立執行。

您可以結合使用容量保留和工作群組，以滿足不同的需求。以下為一些範例案例：

- 隔離 – 若要隔離重要的工作負載，您可以將單一工作群組指派給一個保留。僅來自指派的工作群組的查詢會使用所選保留區中的處理容量。
- 共用 – 多個工作負載會共用一個保留區的容量。例如，如果您想要為一組特定的工作負載產生可預測的每月費用，則可以將多個工作群組指派給單一保留。指派的工作群組共用保留的容量。
- 混合模型 – 您可以在同一帳戶中同時使用容量保留和每個查詢計費。例如，為確保可靠執行支援生產應用程式的查詢，您可以將這些查詢的工作群組指派給容量保留。在將查詢移至生產工作群組之前開發查詢時，您可以使用單獨的工作群組，這些工作群組不與保留建立關聯，因此使用每個查詢計費。

## 了解 DPU

容量是以資料處理單位 (DPU) 為單位測量而得。DPU 表示 Athena 用來代表您存取和處理資料的運算和記憶體資源。一個 DPU 可提供 4 個 vCPU 和 16 GB 的記憶體。您指定的 DPU 數目會影響您可以同時執行的查詢數目。例如，具有 256 個 DPU 的保留所支援的並行查詢數量大約是具有 128 個 DPU 的保留的兩倍。

您可建立多達 100 個容量保留，每個帳戶和區域可建立多達 1,000 個 DPU 總數目上限。您可以請求的最小 DPU 數量為 24。如果您的使用案例需要 1,000 個以上的 DPU，請聯絡 [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com)。

如需有關估算容量需求的資訊，請參閱 [判斷容量要求](#)。如需定價資訊，請參閱 [Amazon Athena 定價](#)。

## 考量與限制

- 該功能需要 [Athena 引擎版本 3](#)。
- 單一工作群組一次最多可以指派一個保留，而您最多可以將 20 個工作群組新增至保留。
- 您無法將已啟用 Spark 的工作群組新增至容量保留。
- 若要刪除已指派給保留的工作群組，請先從保留中移除該工作群組。
- 您可以佈建的最小 DPU 數量為 24。
- 您可建立多達 100 個容量保留，每個帳戶和區域可建立多達 1,000 個 DPU 總數目上限。
- 無法保證容量請求，且可能需要 30 分鐘才能完成。
- 最小計費期間為每個保留 1 小時。1 小時後，容量以每分鐘計費。如需定價資訊，請參閱 [Amazon Athena 定價](#)。
- 預留容量不可轉移至另一個容量保留、AWS 帳戶或 AWS 區域。
- 容量保留的 DDL 查詢會消耗 DPU。
- 在佈建容量上執行的查詢不會計入 DDL 和 DML 的作用中查詢限制。
- 如果所有 DPU 都在使用中，則提交的查詢會排入佇列。此類查詢不會遭到拒絕，也不會使用隨選容量。
- DPUConsumed CloudWatch 量度是以工作群組為單位，而非每個保留區。因此，如果您將工作群組從一個保留區移至另一個保留區，則 DPUConsumed 指標會包含該工作群組屬於第一個保留區時的資料。如需在 Athena 中使用 CloudWatch 量度的詳細資訊，請參閱 [使用 CloudWatch 指標監控 Athena 查詢](#)。
- 目前，該功能可用於以下內容 AWS 區域：

- 美國東部 (維吉尼亞北部)
- 美國東部 (俄亥俄)
- 美國西部 (奧勒岡)
- 亞太區域 (新加坡)
- 亞太區域 (雪梨)
- 亞太區域 (東京)
- 歐洲 (愛爾蘭)
- 歐洲 (西班牙)
- 歐洲 (斯德哥爾摩)
- 南美洲 (聖保羅)

## 主題

- [判斷容量要求](#)
- [建立容量保留](#)
- [管理保留](#)
- [容量保留的 IAM 政策](#)
- [Athena 容量保留 API](#)

## 判斷容量要求

建立容量保留之前，您可以預估所需的容量，以便為其指派正確的 DPU 數目。而且，在使用保留後，您可能需要檢查保留的容量是不足還是過多。本主題說明可用來進行這些估算值的技巧，並說明評估使用量和成本的一些 AWS 工具。

## 主題

- [預估所需容量](#)
- [需要更多容量的跡象](#)
- [檢查閒置容量](#)
- [評估容量需求和成本的工具](#)

## 預估所需容量

預估容量需求時，考慮兩個觀點非常有用：特定查詢可能需要多少容量，以及一般需要多少容量。

## 預估每個查詢的容量需求

若要判斷查詢可能需要的 DPU 數目，您可以使用下列指導方針：

- DDL 查詢會消耗 4 個 GPU。
- DML 查詢會消耗 4 到 124 個 GPU。

Athena 可判斷提交查詢時 DML 查詢所需的 DPU 數目。數目會根據資料大小、儲存格式、查詢建構和其他因素而異。一般而言，Athena 會嘗試選取最低、最有效率的 DPU 數目。如果 Athena 判斷需要更多的運算能力才能順利完成查詢，則會增加指派給查詢的 DPU 數目。

## 估算工作負載特定容量需求

若要判斷同時執行多個查詢時可能需要多少容量，請考慮下列資料表中的一般指導方針：

並行查詢	所需的 DPU
10	40 或以上
20	96 或以上
30 或以上	240 或以上

請注意，您實際需要的 DPU 數目取決於您的目標和分析模式。例如，如果您想要立即開始查詢而不排入佇列，請判斷尖峰並行查詢需求，然後相應地佈建 DPU 數目。

您可以佈建比尖峰需求更少的 DPU，但在發生尖峰需求時可能會導致佇列。進行佇列時，Athena 會將您的查詢保存在佇列中，並在容量可用時執行查詢。

如果您的目標是在固定預算內執行查詢，您可以使用 [AWS Pricing Calculator](#) 來判斷適合您預算的 DPU 數目。

最後請記住，資料大小、儲存格式和查詢的寫入方式會影響查詢所需的 DPU。若要提高查詢效能，您可以壓縮或分割資料，或將其轉換為單欄式格式。如需詳細資訊，請參閱 [Athena 中的效能調校](#)。

## 需要更多容量的跡象

容量不足錯誤訊息和查詢佇列是指派容量不足的兩個指示。

如果您的查詢失敗並顯示容量不足錯誤訊息，則容量保留的 DPU 數目太低，無法滿足您的查詢要求。例如，如果您的保留具有 24 個 DPU，且執行的查詢需要 24 個以上的 DPU，則查詢將失敗。若要監視此查詢錯誤，您可以使用 Athena 的 [EventBridge 事件](#)。嘗試新增更多 DPU，然後重新執行查詢。

如果有許多查詢排入佇列，則表示您的容量已被其他查詢充分利用。若要減少佇列，請執行下列任意一項：

- 將 DPU 新增至您的保留，以提高查詢並行性。
- 從保留中移除工作群組，以釋放容量供其他查詢使用。

若要檢查查詢佇列是否過多，請針對容 [CloudWatch](#) 量保留區中的工作群組使用 Athena 查詢佇列時間量度。如果該值超過您偏好的閾值，您可以將 DPU 新增至容量保留。

### 檢查閒置容量

若要檢查閒置容量，您可以減少保留中的 DPU 數目或增加其工作負載，然後觀察結果。

### 若要檢查閒置容量

1. 執行以下任意一項：
  - 減少保留中的 DPU 數目 (減少可用資源)
  - 將工作群組新增至您的保留 (增加工作負載)
2. 用 [CloudWatch](#) 於測量查詢佇列時間。
3. 如果佇列時間增加超過理想水平，請執行下列任意一項
  - 移除工作群組
  - 將 DPU 新增至您的容量保留
4. 每次變更之後，請檢查效能和查詢佇列時間。
5. 繼續調整工作負載及/或 DPU 計數，以達到所需的平衡。

如果您不想將容量維持在偏好期間以外，您可以 [取消](#) 保留並稍後建立另一個保留。但是，即使您最近取消了其他保留的容量，也無法保證請求新容量，而且建立新的保留需要一些時間。

### 評估容量需求和成本的工具

您可以使用中的下列服務和功能 AWS 來衡量您的 Athena 使用量和成本。

## CloudWatch 度量

您可以將 Athena 設定為 CloudWatch 在工作群組層級將查詢相關指標發佈到 Amazon。為工作群組啟用指標後，工作群組查詢的指標會顯示在工作群組詳細資訊頁面的 Athena 主控台中。

如需發佈至的 Athena 量度 CloudWatch 及其維度的相關資訊，請參閱[使用 CloudWatch 指標監控 Athena 查詢](#)。

## CloudWatch 使用量度

您可以透過在 CloudWatch 圖形和儀表板上顯示目前的服務 CloudWatch 使用情況，使用使用量指標來提供您帳戶使用資源的可見度。對於 Athena，使用量可用性指標會對應於 Athena 的 AWS [服務配額](#)。您可以設定警示，在您的用量接近服務配額時發出警示。

如需詳細資訊，請參閱 [監控 Athena 用量指標](#)。

## Amazon EventBridge 活動

您可以 EventBridge 將 Amazon Athena 與 Amazon 搭配使用，以接收有關查詢狀態的即時通知。當您提交變更狀態的查詢時，Athena 會發佈包 EventBridge 含查詢狀態轉換相關資訊的事件。您可以針對感興趣的事件撰寫簡單的規則，並在事件符合規則時採取自動化動作。

如需詳細資訊，請參閱下列資源。

- [使用 Amazon EventBridge 事件監控 Athena 查](#)
- [什麼是 Amazon EventBridge ?](#)
- [Amazon EventBridge 活動](#)

## 標籤

在 Athena 中，容量保留支援標籤。一個標籤均包含一個索引鍵和一個值。若要追蹤您在 Athena 的成本，您可以使用 AWS 產生的成本分配標籤。AWS 使用成本分配標籤，在「成本 [和用量報表](#)」中組織您的資源成本。這可讓您輕鬆地分類和追蹤 AWS 成本。若要啟用 Athena 的成本分配標籤，請使用 [AWS Billing and Cost Management 主控台](#)。

如需詳細資訊，請參閱下列資源。

- [標記 Athena 資源](#)
- [啟用 AWS 產生的成本分配標籤](#)
- [使用 AWS 成本分配標籤](#)

## 建立容量保留

若要開始使用，您可建立具有所需 DPU 數目的容量保留，然後指派一或多個工作群組，以使用該容量進行查詢。您可以稍後視需要調整容量，以提供更一致的效能或更好地管理成本。如需有關估算容量需求的資訊，請參閱 [判斷容量要求](#)。

### Important

無法保證容量請求，且可能需要 30 分鐘才能完成。

### 若要建立容量保留

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 選擇管理、容量保留。
4. 選擇建立容量保留。
5. 在建立容量保留頁面上，對於容量保留名稱，輸入名稱。名稱必須是唯一的，長度介於 1 至 128 個字元之間，且只能使用字元 a-z、A-Z、0-9、\_(底線)、.(句點) 和 -(連字號)。建立保留後，便無法變更名稱。
6. 對於 DPU，選擇或輸入所需的資料處理單位 (DPU) 數目，增量為 4。如需詳細資訊，請參閱 [了解 DPU](#)。
7. (選用) 展開標籤選項，然後選擇新增標籤，以新增一或多個鍵值組，進而與容量保留資源建立關聯。如需詳細資訊，請參閱 [標記 Athena 資源](#)。
8. 選擇檢閱。
9. 在確認建立容量保留提示下，確認 DPU 的數量及其他資訊。AWS 區域如果您接受，請選擇提交。

在詳細資訊頁面上，容量保留的狀態會顯示為待定。當您的預留容量可用於執行查詢時，其狀態會顯示為作用中。

此時，您可在保留中新增一或多個工作群組。如需這些步驟，請參閱 [將工作群組新增至保留](#)。

## 管理保留

您可以在容量保留頁面上檢視和管理您的容量保留。您可以執行管理任務，例如新增或減少 DPU、修改工作群組指派，以及標記或取消保留。

## 若要檢視和管理容量保留

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 選擇管理、容量保留。
4. 在容量保留頁面上，您可以執行下列任務：
  - 若要[建立](#)容量保留，請選擇建立容量保留。
  - 使用搜尋方塊，依名稱或 DPU 數目篩選保留。
  - 選擇狀態下拉式選單，依容量保留狀態進行篩選 (例如，作用中或已取消)。如需有關保留狀態的詳細資訊，請參閱 [了解保留狀態](#)。
  - 若要檢視容量保留的詳細資訊，請選擇保留的連結。保留的詳細資訊頁面包括以下選項：[編輯容量](#)、[新增工作群組](#)、[移除工作群組](#)以及[取消](#)保留。
  - 若要編輯保留 (例如，透過新增或移除 DPU)，請選取保留的按鈕，然後選擇編輯。
  - 若要取消預留，請選取保留按鈕，然後選擇取消。

## 了解保留狀態

下表說明容量保留的可能狀態值。

狀態	描述
待定	Athena 正在處理您的容量請求。容量尚未準備好執行查詢。
Active (作用中)	容量可用於執行查詢。
失敗	您的容量請求未成功完成。請注意，無法保證滿足容量請求。失敗的保留會計入您的帳戶 DPU 限制。若要釋出使用量，您必須取消保留。
更新待定	Athena 正在處理保留的變更。例如，在您編輯保留以新增或移除 DPU 之後，就會發生此狀態。
取消	Athena 正在處理取消保留的請求。仍在保留的工作群組中執行的查詢可以完成，但工作群組中的其他查詢將使用隨選 (未佈建的) 容量。

狀態	描述
已取消	<p>容量保留取消完成。已取消的保留會在主控台中保留 45 天。45 天後，Athena 將刪除保留。在這 45 天期間，您無法重新規劃或重複使用保留，但您可以參考其標籤並檢視其詳細資訊以供歷史參考。</p> <p>無法保證可在未來日期重新保留取消的容量。容量不能轉移到其他保留區，AWS 帳戶 或者 AWS 區域。</p>

## 了解作用中 DPU 和目標 DPU

在 Athena 主控台的容量保留清單中，您的保留會顯示兩個 DPU 值：作用中 DPU 和目標 DPU。

- 作用中 DPU – 保留中可用來執行查詢的 DPU 數目。例如，如果您請求 100 個 DPU，且您的請求已滿足，則作用中 DPU 會顯示 100。
- 目標 DPU – 您的保留正在移到的 DPU 數目。在建立保留或待定增加或減少 DPU 數目時，目標 DPU 顯示的值會與作用中 DPU 不同。

例如，在您提交請求以建立具有 24 個 DPU 的保留之後，保留狀態將為待定，作用中 DPU 將為 0，而目標 DPU 將為 24。

如果您的保留有 100 個 DPU，並編輯您的保留以請求增加 20 個 DPU，則狀態將為更新待定，作用中 DPU 將為 100，而目標 DPU 將為 120。

如果您的保留有 100 個 DPU，並編輯您的保留以請求減少 20 個 DPU，則狀態將為更新待定，作用中 DPU 將為 100，而目標 DPU 將為 80。

在這些轉換期間，Athena 會根據您的請求積極獲取或減少 DPU 數目。當作用中 DPU 等於目標 DPU 時，則達到目標數目，且沒有任何待定變更。

若要以程式設計方式擷取這些值，您可以呼叫 [GetCapacityReservation](#) API 動作。該 API 將作用中 DPU 和目標 DPU 稱為 AllocatedDpus 和 TargetDpus。

## 主題

- [編輯容量保留](#)
- [將工作群組新增至保留](#)
- [從保留移除工作群組](#)

- [取消容量保留](#)
- [刪除容量保留](#)

## 編輯容量保留

建立容量保留後，您可以調整其 DPU 數目，以及新增或移除其自訂標籤。

### 若要編輯容量保留

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 選擇管理、容量保留。
4. 在容量保留清單中，執行下列任意一項：
  - 選取保留旁邊的按鈕，然後選擇編輯。
  - 選擇保留連結，然後選擇編輯。
5. 對於 DPU，選擇或輸入所需的資料處理單位數目，增量為 4。您可以擁有的最小 DPU 數量為 24。如需詳細資訊，請參閱 [了解 DPU](#)。

#### Note

您可以隨時將 DPU 新增至現有的容量保留。不過，您必須得等到建立保留或新增 DPU 後 1 小時之後，才能減少 DPU 的數目。

6. (選用) 針對標籤，選擇移除以移除標籤，或選擇新增標籤以新增標籤。
7. 選擇提交。保留詳細資訊頁面會顯示更新的組態。

## 將工作群組新增至保留

建立容量保留後，您最多可以將 20 個工作群組新增至保留。將工作群組新增至保留會告訴 Athena 應該在預留容量上執行哪些查詢。來自未與保留關聯的工作群組查詢，會繼續使用預設的每 TB 掃描定價模式執行。

當保留有兩個或多個工作群組時，來自這些工作群組的查詢可以使用保留的容量。您可以隨時新增和移除工作群組。新增或移除工作群組時，不會中斷正在執行的查詢。

當您的保留處於待定狀態時，來自您新增的工作群組的查詢會繼續使用預設的每 TB 掃描定價模式執行，直到保留變為作用中。

## 若要將一或多個工作群組新增至您的容量保留

1. 在容量保留的詳細資訊頁面上，選擇新增工作群組。
2. 在新增工作群組頁面上，選取要新增的工作群組，然後選擇新增工作群組。您不能將一個工作群組指派至多個預留。

容量保留的詳細資訊頁面會列出您新增的工作群組。在這些工作群組中執行的查詢，將使用您在保留處於作用中狀態時保留的容量。

## 從保留移除工作群組

如果您不再需要工作群組專用容量，或想要將工作群組移至其自己的保留中，您可以隨時將其移除。從保留移除工作群組輕鬆簡單。從保留移除工作群組後，來自已移除的工作群組的查詢預設為使用隨選 (未佈建) 容量，並根據掃描的 TB 計費。

## 若要從保留移除一或多個工作群組

1. 在容量保留之詳細資料頁面上，選取您要移除的工作群組。
2. 選擇移除工作群組。是否移除工作群組？提示會通知您，在將工作群組從保留中移除之前，將會完成所有目前作用中的查詢。
3. 選擇移除。容量保留的詳細資訊頁面會顯示移除的工作群組不再存在。

## 取消容量保留

如果您不想再使用容量保留，則可將其取消。仍在使用保留的工作群組中執行的查詢將可完成，但工作群組中的其他查詢將不再使用保留。

### Note

無法保證可在未來日期重新保留取消的容量。容量不能轉移到其他保留區，AWS 帳戶 或者 AWS 區域。

## 若要取消容量保留

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 選擇管理、容量保留。

4. 在容量保留清單中，執行下列任意一項：
  - 選取保留旁邊的按鈕，然後選擇取消。
  - 選擇容量連結，然後選擇取消容量保留。
5. 畫面出現是否取消容量保留？提示時，輸入取消，然後選擇取消容量保留。

保留的狀態會變更為取消中，且進度橫幅會通知您正在取消。

取消完成後，容量保留仍會保留，但其狀態會顯示為已取消。保留將在取消後 45 天刪除。在這 45 天期間，您無法重新規劃或重複使用已取消的保留，但您可以參考其標籤並檢視其詳細資訊以供歷史參考。

## 刪除容量保留

如果您要移除已取消的容量保留的所有參考，您可以刪除保留。必須先取消保留，然後才可將其刪除。刪除的保留會立即從您的帳戶中移除，且無法再參考，包括其 ARN。

### 若要刪除容量保留

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 選擇管理、容量保留。
4. 在容量保留清單中，執行下列任意一項：
  - 選取已取消的保留旁邊的按鈕，然後選擇動作、刪除。
  - 選擇保留連結，然後選擇刪除。
5. 畫面出現是否刪除容量保留？提示時，選擇刪除。

畫面會出現橫幅，通知您已成功刪除容量保留。刪除的保留不再出現在容量保留清單中。

## 容量保留的 IAM 政策

若要控制對容量保留的存取，請使用資源層級 IAM 許可或以身為基礎的 IAM 政策。每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

下列程序專用於 Athena。

如需 IAM 特定的資訊，請參閱本節最後列出的連結。如需有關範例 JSON 容量保留政策的資訊，請參閱 [容量保留範例政策](#)。

若要在 IAM 主控台中使用視覺化編輯器來建立容量保留政策

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在左邊的導覽窗格中，選擇 Policies (政策)，然後選擇 Create policy (建立政策)。
3. 在 Visual editor (視覺化編輯器) 標籤上，選擇 Choose a service (選擇一項服務)。接著選擇 Athena 以新增到政策。
4. 選擇 Select actions (選取動作)，然後選擇要新增到政策的動作。視覺化編輯器會顯示 Athena 中可用的動作。如需詳細資訊，請參閱《服務授權參考》中的 [Amazon Athena 的動作、資源和條件索引鍵](#)。
5. 選擇新增動作來輸入特定動作，或使用萬用字元 (\*) 來指定多個動作。

預設情況下，您建立的政策允許執行選擇的操作。對於 Athena 中的 capacity-reservation 資源，如果您選擇的一個或多個動作支援資源層級許可，則編輯器會列出 capacity-reservation 資源。

6. 選擇資源來為您的政策指定特定容量保留。如需有關範例 JSON 容量保留政策的資訊，請參閱 [容量保留範例政策](#)。
7. 如下所示指定 capacity-reservation 資源：

```
arn:aws:athena:<region>:<user-account>:capacity-reservation/<capacity-reservation-name>
```

8. 選擇 Review policy (檢閱政策)，然後為您正在建立的政策輸入 Name (名稱) 與 Description (描述) (選用)。檢閱政策摘要來確認您已授予想要的許可。
9. 選擇 Create policy (建立政策) 儲存您的新政策。
10. 將此基於身分的政策連接到使用者、群組或角色。

如需詳細資訊，請參閱《服務授權參考》與《IAM 使用者指南》中的下列主題：

- [Amazon Athena 的操作、資料和條件索引鍵](#)
- [使用視覺化編輯器來建立政策](#)
- [新增和移除 IAM 政策](#)
- [控制資源的存取](#)

如需有關範例 JSON 容量保留政策的資訊，請參閱 [容量保留範例政策](#)。

如需 Amazon Athena 動作的完整清單，請參閱《[Amazon Athena API 參考](#)》中的 API 動作名稱。

## 容量保留範例政策

本節包含可讓您在容量保留上用來啟用各種動作的範例政策。每當您使用 IAM 政策時，請務必遵循 IAM 最佳實務。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

容量保留是由 Athena 管理的 IAM 資源。因此，如果您的容量保留政策使用 `capacity-reservation` 做為輸入的動作，您必須指定容量保留的 ARN，如下所示：

```
"Resource": [arn:aws:athena:<region>:<user-account>:capacity-reservation/<capacity-reservation-name>]
```

其中 `<capacity-reservation-name>` 為容量保留的名稱。例如，對於名為 `test_capacity_reservation` 的容量保留，將其指定為資源，如下所示：

```
"Resource": ["arn:aws:athena:us-east-1:123456789012:capacity-reservation/test_capacity_reservation"]
```

如需 Amazon Athena 動作的完整清單，請參閱《[Amazon Athena API 參考](#)》中的 API 動作名稱。如需有關 IAM 政策的詳細資訊，請參閱《IAM 使用者指南》中的 [使用視覺化編輯器來建立政策](#)。

- [Example policy to list capacity reservations](#)
- [Example policy for management operations](#)

Example 適用於列出容量保留的政策範例

以下政策允許所有使用者列出所有容量保留。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListCapacityReservations"
      ],
      "Resource": "*"
    }
  ]
}
```

```

    }
  ]
}

```

## Example 適用於管理操作的策略範例

下列政策可讓使用者建立、取消、取得詳細資訊，以及更新容量保留 `test_capacity_reservation`。該政策還允許使用者指派 `workgroupA` 和 `workgroupB` 給 `test_capacity_reservation`。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateCapacityReservation",
        "athena:GetCapacityReservation",
        "athena:CancelCapacityReservation",
        "athena:UpdateCapacityReservation",
        "athena:GetCapacityAssignmentConfiguration",
        "athena:PutCapacityAssignmentConfiguration"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:capacity-
reservation/test_capacity_reservation",
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA",
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupB"
      ]
    }
  ]
}

```

## Athena 容量保留 API

下列清單包含 Athena 容量保留 API 動作的參考連結。如需資料結構和其他 Athena API 動作，請參閱 [Amazon Athena API 參考](#)。

- [CancelCapacityReservation](#)
- [CreateCapacityReservation](#)
- [GetCapacityAssignmentConfiguration](#)

- [GetCapacityReservation](#)
- [ListCapacityReservations](#)
- [PutCapacityAssignmentConfiguration](#)
- [UpdateCapacityReservation](#)

## Athena 中的效能調校

本主題就改善 Athena 查詢效能以及如何解決與限制和資源使用量相關的錯誤，提供了一般資訊和具體建議。

### Service Quotas

Athena 會針對查詢執行時間、帳戶中的並行查詢數目以及 API 請求率等指標強制執行配額。如需有關這些配額的詳細資訊，請參閱 [Service Quotas](#)。超過這些配額會導致查詢失敗 – 無論是在提交查詢時還是在查詢執行期。

此頁面上的許多效能最佳化秘訣可協助縮短查詢的執行時間。最佳化可釋出容量，以便您可以在並行配額內執行更多查詢，並避免查詢因執行時間過長而被取消。

並行查詢和 API 請求數量的配額是每個 AWS 帳戶和 AWS 區域。我們建議每個工作負載 AWS 帳戶 (或使用個別佈建的容量保留區) 執行一個工作負載，以防止工作負載競爭相同的配額。

如果您在相同的帳戶中執行兩個工作負載，其中一個工作負載可能要執行突增的查詢。這可能會導致剩餘的工作負載遭受限流或封鎖，而無法執行查詢。若要避免這種情況，您可以將工作負載移至不同的帳戶，以便為每個工作負載提供自己的並行配額。為一個或兩個工作負載建立佈建的容量保留可完成相同的目標。

### 其他服務的配額

當 Athena 執行查詢時，它可以呼叫其他強制執行配額的服務。在查詢執行期間，Athena 可以對 Amazon S3 和其他 AWS 服務 (例如 IAM 和 AWS KMS. AWS Glue Data Catalog 如果您使用 [聯合查詢](#)，Athena 也會致電 AWS Lambda。所有這些服務都有自己的限制和配額，可以超過。當查詢執行遇到來自這些服務的錯誤時，其便會失敗並包含來源服務的錯誤。重試可復原的錯誤，但如果問題無法及時自行解決，查詢仍可能失敗。請務必徹底閱讀錯誤訊息，以判斷其是來自 Athena 還是來自其他服務。本文件涵蓋了一些相關錯誤。

如需有關解決 Amazon S3 Service Quotas 造成的錯誤的資訊，請參閱本文件稍後的 [避免檔案太多](#)。如需有關 Amazon S3 效能最佳化的詳細資訊，請參閱《Amazon S3 使用者指南》中的 [最佳實務設計模式：最佳化 Amazon S3 效能](#)。

## 資源限制

Athena 會在分散式查詢引擎中執行查詢。當您提交查詢時，Athena 引擎查詢規劃程式會預估執行查詢所需的運算容量，並相應地準備運算節點叢集。某些查詢 (例如 DDL 查詢) 只在一個節點上執行。大型資料集的複雜查詢可在更大的叢集上執行。節點是統一的，具有相同的記憶體、CPU 和磁碟組態。Athena 透過橫向擴展 (而非向上擴展) 來處理要求更高的查詢。

有時候，查詢的需求會超過執行查詢的叢集的可用資源。發生這種情況時，查詢會失敗，並顯示錯誤在此擴展因數下查詢耗盡的資源。

最常耗盡的資源是記憶體，但在極少數情況下，也可以是磁碟空間。當引擎執行聯結或視窗函數時，通常會發生記憶體錯誤，但也可能發生在不同的計數和彙總中。

即使查詢失敗並出現「資源不足」錯誤一次，當您再次執行時，其可能會成功。查詢執行不是確定性的。載入資料所花費時間以及如何在節點上分配中繼資料集等因素，可能會導致資源使用量不同。例如，假設一個查詢聯結兩個資料表，並且聯結條件的值分佈存在很大偏差。這樣的查詢在大多數時間都可以成功，但是當最常見的值最終由相同節點處理時，偶爾會失敗。

若要避免查詢超出可用資源，請使用本文件中提及的效能調校秘訣。具體而言，有關如何最佳化會耗盡可用資源之查詢的秘訣，請參閱 [最佳化聯結](#)、[最佳化視窗函數](#) 和 [使用近似值最佳化查詢](#)。

## 查詢最佳化技術

使用本節中描述的查詢最佳化技術，可加快查詢執行速度，或做為 Athena 中超出資源限制的查詢解決方法。

### 最佳化聯結

在分散式查詢引擎中執行聯結的策略有很多。最常見的兩個是分散式雜湊聯結和具有複雜聯結條件的查詢。

#### 分散式雜湊聯結

最常見的聯結類型使用對等比較做為聯結條件。Athena 以分散式雜湊聯結的形式執行此類聯結。

在分散式雜湊聯結中，引擎會從聯結的一側建置一個查詢表 (雜湊資料表)。這一側稱為建置端。建置端的記錄分散在多個節點上。每個節點為其子集建置一個查詢表。接著，聯結的另一端稱為探查端，會透過節點進行串流。探查端的記錄以與建置端相同的方式分散在節點上。這樣會啟用每個節點，以透過在其自己的查詢表中查詢相符記錄來執行聯結。

當從聯結的建置端建立的查詢表不適合放入記憶體中時，查詢可能會失敗。即使建置端的總大小小於可用記憶體，如果記錄的分佈存在很大偏差，查詢也可能會失敗。在極端情況下，所有記錄可能具有相同

的聯結條件值，並且必須放入單一節點上的記憶體中。如果將一組值傳送至相同節點，且值加起來超過可用記憶體，即使是查詢偏差較小，也可能會失敗。節點確實可以將記錄溢寫至磁碟，但溢出會降低查詢執行速度，並且可能不足以防止查詢失敗。

Athena 會嘗試重新排序聯結，以使用較大的關係做為探查端，而較小的關係做為建置端。不過，由於 Athena 不會管理資料表中的資料，所以資訊有限，而且通常必須假設第一個資料表較大，第二個資料表較小。

使用以等值為基礎的聯結條件撰寫聯結時，假設 JOIN 關鍵字左側的資料表是探查端，右側的資料表是建置端。確保正確的資料表 (建置端) 是較小的資料表。如果無法使聯結的建置端足夠小以放入記憶體中，請考慮執行多個聯結建置資料表子集的查詢。

### 其他聯結類型

具有複雜聯結條件的查詢 (例如，使用 LIKE、> 或其他運算子的查詢) 通常運算要求較高。在最壞的情況下，聯結一端的每條記錄必須與聯結另一端的每條記錄進行比較。由於執行時間會隨著記錄數量的平方而增加，因此此類查詢會有超過最大執行時間的風險。

若要了解 Athena 將如何事先執行查詢，您可以使用 EXPLAIN 陳述式。如需詳細資訊，請參閱 [在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#) 及 [了解 Athena EXPLAIN 陳述式結果](#)。

### 最佳化視窗函數

由於視窗函數是資源密集型操作，所以它們可能會使查詢執行緩慢甚至失敗，並顯示在此擴展因數下查詢耗盡的資源。視窗函數將其操作的所有記錄保留在記憶體中，以便計算其結果。當視窗非常大時，視窗函數可能會耗盡記憶體。

為了確保您的查詢在可用的記憶體限制內執行，請減少視窗函數操作的視窗大小。為此，您可以新增 PARTITIONED BY 子句或縮小現有分割區子句的範圍。

### 改用非視窗函數

有時，具有視窗函數的查詢可以在沒有視窗函數的情況下重寫。例如，您可以使用 ORDER BY 和 LIMIT，而不是使用 row\_number 來尋找前 N 個記錄。您可以使用彙總函數 (例如 [max\\_by](#)、[min\\_by](#) 和 [arbitrary](#))，而不是使用 row\_number 或 rank 來重複記錄。

例如，假設您的資料集包含來自感應器的更新。感應器會定期報告其電池狀態，並包含一些中繼資料，例如位置。如果您想知道每個感應器及其位置的最新電池狀態，可以使用以下查詢：

```
SELECT sensor_id,
       arbitrary(location) AS location,
       max_by(battery_status, updated_at) AS battery_status
```

```
FROM sensor_readings
GROUP BY sensor_id
```

因為每筆記錄的中繼資料 (例如位置) 都相同，因此您可以使用 `arbitrary` 函數從群組中選擇任何值。

要獲取最新的電池狀態，您可以使用 `max_by` 函數。`max_by` 函數從找到另一個資料欄的最大值的記錄中選擇資料欄的值。在此範例中，它會傳回群組內上次更新時間的記錄的電池狀態。與具有視窗函數的對等查詢相比，此查詢執行速度更快，而且使用的記憶體更少。

### 最佳化彙總

當 Athena 執行彙總時，它會使用 `GROUP BY` 子句中的資料欄，跨不同的工作節點分佈記錄。為了使相符記錄與群組的任務盡可能有效率，節點會嘗試將記錄放入記憶體中，但必要時會將記錄溢寫至磁碟。

避免在 `GROUP BY` 子句中包含備援資料欄也是個不錯的主意。由於較少的資料欄需要較少的記憶體，因此使用較少資料欄描述群組的查詢會更有效率。數值資料欄使用的記憶體也少於字串。例如，當您彙總同時具有數值類別 ID 和類別名稱的資料集時，請僅使用 `GROUP BY` 子句中的類別 ID 資料欄。

有時候，查詢會在 `GROUP BY` 子句中包含資料欄，以解決資料欄必須是 `GROUP BY` 子句的一部分或彙總表達式的事實。如果未遵循此規則，您可能會收到以下錯誤訊息：

EXPRESSION\_NOT\_AGGREGATE：行 1:8：「類別」必須是一個彙總表達式或出現在 `GROUP BY` 子句

若要避免在 `GROUP BY` 子句中新增備援資料欄，您可以使用 [任意](#) 函數，如下列範例所示。

```
SELECT country_id,
       arbitrary(country_name) AS country_name,
       COUNT(*) AS city_count
FROM world_cities
GROUP BY country_id
```

`ARBITRARY` 函數會從群組中傳回任意值。當您知道群組中的所有記錄都具有相同的資料欄值，但該值無法識別群組時，此函數非常有用。

### 最佳化前 N 個查詢

`ORDER BY` 子句會以排序順序傳回查詢的結果。Athena 使用分散式排序，在多個節點上平行執行排序操作。

如果您並不需要對結果進行排序，請避免新增 ORDER BY 子句。另外，如果並非絕對必要，請避免新增 ORDER BY 到內部查詢。在許多情況下，查詢規劃程式可以移除備援排序，但不保證一定如此。此規則的例外是，如果內部查詢正在執行前 N 個操作，例如尋找最新的 N 個值或最常用的 N 個值。

當 Athena 發現 ORDER BY 與 LIMIT 時，就會明白您正在執行前 N 個查詢，並相應地使用專用操作。

#### Note

雖然 Athena 也經常可以偵測視窗函數 (例如使用前 N 個的 `row_number`)，但我們建議使用 ORDER BY 和 LIMIT 的較簡單版本。如需詳細資訊，請參閱 [最佳化視窗函數](#)。

### 僅包含必填資料欄

如果您並不需要資料欄，請不要將其包含在查詢中。查詢必須處理的資料越少，執行速度就越快。這可以減少所需的記憶體數量，以及必須在節點之間傳送的資料量。如果您使用的是單欄式檔案格式，減少資料欄數量也會減少從 Amazon S3 讀取的資料量。

Athena 對結果中的資料欄數量沒有特定限制，但是執行查詢的方式會限制可能的資料欄合併大小。資料欄的合併大小包括它們的名稱和類型。

例如，下列錯誤是由超出關係描述項大小限制的關係造成的：

一般 \_ 內部錯誤：.airlift. 位元組碼。CompilationException

若要解決此問題，請減少查詢中的資料欄數量或建立子查詢，並使用擷取較少量資料的 JOIN。如果您有在最外層查詢中執行 SELECT \* 的查詢，則應將 \* 變更為僅包含所需資料欄的清單。

### 使用近似值最佳化查詢

Athena 支援[近似彙總函數](#)，用於計算獨特值、最常出現的值、百分位數 (包括近似中位數)，以及建立直方圖。每當不需要確切值時，請使用這些函數。

與 COUNT(DISTINCT col) 操作不同，[approx\\_distinct](#) 使用的記憶體更少且執行速度更快。同樣，使用 [numeric\\_histogram](#) (而不是[直方圖](#)) 使用了近似方法，因此所需的記憶體更少。

### 最佳化 LIKE

您可以使用 LIKE 來查找相符字串，但對於長字串而言，此為計算密集型。[regexp\\_like](#) 函數在大多數情況下是一個更快的替代方案，並且還提供了更多的靈活性。

通常，您可以透過錨定要尋找的子字串來最佳化搜尋。##### 'substr%' ### '%substr%'。或者，如果您使用的是 `regexp_like`，則為 `^substr'`。

## 使用 UNION ALL 而非 UNION

UNION ALL 和 UNION 是兩種將兩個查詢結果合併為一個結果的方法。UNION ALL 將第一個查詢中的記錄與第二個查詢中的記錄串連起來，且 UNION 會執行相同的操作，但也會移除重複項目。UNION 需要處理所有記錄並找到重複項目，此為記憶體和計算密集型，不過 UNION ALL 是一個相對快速的動作。除非您需要重複記錄，否則請使用 UNION ALL 以獲得最佳效能。

## 對大型結果集使用 UNLOAD

當查詢結果預期很大 (例如，數萬個資料列或更多) 時，請使用 UNLOAD 匯出結果。在大多數情況下，這比執行規則查詢更快，並且使用 UNLOAD 也可以讓您更好地控制輸出。

查詢完成執行後，Athena 會將結果以單一未壓縮的 CSV 檔案形式存放在 Amazon S3 上。這比 UNLOAD 需要更長的時間，不僅因為結果未壓縮，而且還因為操作無法平行化。相反地，UNLOAD 會直接從工作節點寫入結果，並充分利用運算叢集的平行處理。此外，您可以設定 UNLOAD，以壓縮格式和其他檔案格式 (例如 JSON 和 Parquet) 寫入結果。

如需詳細資訊，請參閱 [UNLOAD](#)。

## 使用 CTAS 或 Glue ETL 將常用的彙總具體化

「具體化」查詢是一種透過儲存預先計算的複雜查詢結果 (例如，彙總和聯結) 以便在後續查詢中重複使用，來加速查詢效能的方法。

如果您的許多查詢包含相同的聯結和彙總，您可以將通用子查詢具體化為新資料表，然後針對該資料表執行查詢。您可以使用 [從查詢結果建立資料表 \(CTAS\)](#) 或專用 ETL 工具 (例如 [Glue ETL](#)) 來建立新的資料表。

例如，假設您的儀表板中包含顯示訂單資料集的不同層面的小工具。每個小工具都有自己的查詢，但所有查詢會共用相同的聯結和篩選條件。訂單資料表會與明細項目資料表結合在一起，而且有一個篩選條件，可僅顯示過去三個月。如果您識別這些查詢的常用功能，您可以建立小工具可使用的新資料表。這樣可以減少重複並提高效率。缺點是您必須將資料表保持在最新狀態。

## 重複使用查詢結果

同一查詢在短時間內執行多次非常常見。例如，當多個人開啟相同的資料儀表板時，就會發生這種情況。執行查詢時，您可以告訴 Athena 重複使用先前計算的結果。您可以指定要重複使用的結果的最長期限。如果先前在該時間範圍內執行同一查詢，Athena 會傳回這些結果，而不是再次執行查詢。如需

詳細資訊，請參閱此處 [《Amazon Athena 使用者指南》](#) 中的 [重複使用查詢結果](#)，以及 AWS 大數據部落格中的 [使用 Amazon Athena 查詢結果重複使用降低成本和提升查詢效能](#)。

## 資料最佳化技術

效能不僅取決於查詢，而且還取決於資料集的組織方式，以及資料集使用的檔案格式和壓縮。

### 分割您的資料

分割會將您的資料表分為多個部分，並根據日期、國家或地區等屬性將相關資料保留在一起。分割區索引鍵可作為虛擬資料欄。您可以在建立資料表時定義分割區索引鍵，並使用它們來篩選查詢。當您篩選分割區索引鍵資料欄時，只會讀取相符分割區中的資料。例如，如果您的資料集依日期進行分割，而您的查詢具有僅符合上週的篩選條件，則只會讀取上週的資料。如需有關分割區的詳細資訊，請參閱 [在 Athena 中分割資料](#)。

### 選擇將支援您查詢的分割區索引鍵

由於分割會對查詢效能產生重大影響，因此在設計資料集和資料表時，請務必仔細考慮分割的方式。分割區索引鍵太多可能會導致資料集分段為過多的檔案且各個檔案過小。相反地，分割區索引鍵太少或完全沒有分割，會導致查詢掃描的資料超過必要的資料。

### 避免最佳化罕見查詢

一個好的策略是針對最常見的查詢進行最佳化，並避免針對罕見查詢進行最佳化。例如，如果您的查詢瀏覽時間跨度 (天數)，即使某些查詢會篩選至該級別，也不要按小時進行分割。如果您的資料具有精確的時間戳記資料欄，則依小時篩選的罕見查詢可以使用時間戳記資料欄。即使極少數情況掃描的資料超過必要的資料，為了極少數情況而降低整體效能通常不是一種很好的權衡。

若要減少查詢必須掃描的資料量，進而改善效能，請使用單欄式檔案格式，並保持記錄排序。請依時間戳記排序記錄，而不是依小時分割。對於在較短時間視窗上進行查詢，按時間戳記排序的效率幾乎與依小時分割一樣有效。此外，依時間戳記排序通常不會損害時間視窗上的查詢效能 (以天數計算)。如需詳細資訊，請參閱 [使用單欄式檔案格式](#)。

請注意，如果所有分割區索引鍵都有述詞，則對具有數萬個分割區的資料表進行查詢的效果會更好。這是為最常見的查詢設計分割結構的另一個原因。如需詳細資訊，請參閱 [依等式查詢分割區](#)。

### 使用分割區投影

分割區投影是一項 Athena 功能，它不會將分割區資訊儲存在中 AWS Glue Data Catalog，而是儲存在中表格屬性中的規則 AWS Glue。當 Athena 在設定了分割區投影的資料表上計劃查詢時，其會讀取資

料表的分割區投影規則。Athena 會根據查詢和規則，計算要在記憶體中讀取的分割區，而不是在 AWS Glue Data Catalog 中查找分割區。

除了簡化分割區管理之外，分割區投影還可以改善具有大量分割區的資料集的效能。當查詢包含範圍而不是分割區索引鍵的特定值時，在目錄中查找相符分割區所花費的時間越長，所需的分割區就越多。使用分割區投影，可以在記憶體中計算篩選條件而無需進入目錄，並且速度可以更快。

在某些情況下，分割區投影可能會導致效能變差。一個範例就是當資料表為「稀疏」時。稀疏資料表沒有分割區投影組態所描述的分割區索引鍵值的每個排列資料。使用稀疏資料表時，從查詢和分割區投影組態計算的一組分區都會列在 Amazon S3 上，即使它們沒有資料也一樣。

當您使用分割區投影時，請務必在所有分割區索引鍵上包含述詞。縮小可能值的範圍，以避免不必要的 Amazon S3 清單。假設一個分割區索引鍵的範圍包含一百萬個值，以及查詢在該分割區索引鍵上沒有任何篩選條件。若要執行查詢，Athena 必須執行至少一百萬個 Amazon S3 清單操作。查詢特定值時，無論您是使用分割區投影還是將分割區資訊儲存在目錄中，查詢速度都是最快的。如需詳細資訊，請參閱 [依等式查詢分割區](#)。

當您設定分割區投影的資料表時，請確定您指定的範圍是合理的。如果查詢不包含分割區索引鍵的述詞，則會使用該索引鍵範圍內的所有值。如果您的資料集是在特定日期建立的，請使用該日期做為任何日期範圍的起點。使用 NOW 作為日期範圍的結束。避免使用具有大量值的數值範圍，並考慮改用 [注入](#) 類型。

如需有關分割區投影的詳細資訊，請參閱 [使用 Amazon Athena 進行分割區投影](#)。

## 使用分割區索引

資料分割索引是中的一項功能 AWS Glue Data Catalog，可改善具有大量分割區之資料表的資料分割查閱效能。

目錄中的分割區清單就像是關聯式資料庫中的資料表。此資料表包含分割區索引鍵的資料欄，以及分割區位置的其他資料欄。當您查詢分割區資料表時，會掃描此資料表來查詢分割區位置。

就像關聯式資料庫一樣，您可以透過新增索引來提高查詢的效能。您可以新增多個索引，以支援不同的查詢模式。資料 AWS Glue Data Catalog 分割索引支援相等運算子和比較運算子 >，例如 >=、<，並與運 AND 算子 < 結合。如需詳細資訊，請參閱 [AWS Glue 發人員指南 AWS Glue 中的使用分割區索引](#) 和 AWS 大數據部落格中的 [使用 AWS Glue Data Catalog 分區索引改善 Amazon Athena 查詢效能](#)。

## 始終使用 STRING 做為分割區索引鍵類型

當您查詢分割區索引鍵時，請記住 Athena 要求分割區索引鍵為類型 STRING，才能將分割區篩選下推至 AWS Glue。如果分割區數目不小，使用其他類型可能會導致效能變差。如果您的分割區索引鍵值為 date-like 或 number-like，請將它們轉換為查詢中的適當類型。

## 移除舊的和空的分割區

如果您從 Amazon S3 上的分割區移除資料 (例如，使用 Amazon S3 [生命週期](#))，您也應該從 AWS Glue Data Catalog 中移除分割區項目。在查詢規劃期間，與查詢相符的任何分割區都會列在 Amazon S3 上。如果您有許多空的分割區，列出這些分割區的負荷可能會產生不利影響。

此外，如果您有數千個分割區，請考慮移除不再相關的舊資料的分割區中繼資料。例如，如果查詢從未查看超過一年的資料，您可以定期移除舊分割區的分割區中繼資料。如果分割區數目增加到數萬個，移除未使用的分割區可以加速查詢，其中這些查詢不包含所有分割區索引鍵的述詞。如需有關在查詢中包含所有分割區索引鍵的述詞的資訊，請參閱 [依等式查詢分割區](#)。

## 依等式查詢分割區

在所有分割區索引鍵上包含等式述詞的查詢執行速度更快，因為可以直接載入分割區中繼資料。避免查詢中一個或多個分割區索引鍵沒有述詞，或述詞選擇了範圍值。對於此類查詢，必須篩選所有分割區的清理，以找到相符的值。對於大多數資料表而言，負荷會降至最低，但對於具有數萬個或更多分割區的資料表而言，負荷可能會變得很大。

如果無法重寫查詢以透過等式篩選分割區，則可以嘗試分割區投影。如需詳細資訊，請參閱 [使用分割區投影](#)。

## 避免使用 MSCK REPAIR TABLE 進行分割區維護

由於 MSCK REPAIR TABLE 可能需要很長的時間才能執行，並且僅會新增分割區，而不會移除舊的分割區，因此不是管理分割區的有效方法 (請參閱 [考量與限制](#))。

使用 [AWS Glue Data Catalog API](#)、[ALTER TABLE ADD PARTITION](#) 或 [AWS Glue 爬蟲程式](#) 可以更好地手動管理分割區。或者，您也可以使用分割區投影，這樣就不需要完全管理分割區。如需詳細資訊，請參閱 [使用 Amazon Athena 進行分割區投影](#)。

## 驗證您的查詢是否與分割結構相容

您可以使用 [EXPLAIN](#) 陳述式預先檢查查詢將掃描的分割區。使用 EXPLAIN 關鍵字作為查詢字首，然後在 EXPLAIN 輸出底部附近查找每個資料表的來源片段 (例如，Fragment 2 [SOURCE])。查找右側被定義為分割區索引鍵的指派。下方的列包含一份清單，列出執行查詢時將要掃描的該分割區索引鍵的所有值。

例如，假設您對具有 dt 分割區索引鍵的資料表進行查詢，並使用 EXPLAIN 作為查詢字首。如果查詢中的值是日期，並且篩選條件選取了三天的範圍，則 EXPLAIN 輸出可能如下所示：

```
dt := dt:string:PARTITION_KEY
    :: [[2023-06-11], [2023-06-12], [2023-06-13]]
```

EXPLAIN 輸出顯示，規劃程式找到了此分割區索引鍵的三個值，且其與查詢相符。它還顯示了這些值是什麼。如需有關使用 EXPLAIN 的詳細資訊，請參閱 [在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#) 和 [了解 Athena EXPLAIN 陳述式結果](#)。

## 使用單欄式檔案格式

專為分散式分析工作負載而設計的單欄式檔案格式，例如 Parquet 和 ORC。它們依資料欄而非資料列整理資料。以單欄式格式整理資料具有下列優點：

- 僅會載入查詢所需的資料欄
- 減少需要載入的整體資料量
- 資料欄值會一起存放，因此可以有效地壓縮資料
- 檔案可以包含允許引擎略過載入不需要的資料的中繼資料

做為如何使用檔案中繼資料的範例，檔案中繼資料可以包含有關資料頁面中最小值和最大值的資訊。如果查詢的值不在中繼資料中註明的範圍內，則可以略過該頁面。

使用此中繼資料提高效能的一種方法是，確定對檔案中的資料進行排序。例如，假設您有查詢會尋找 `created_at` 項目在短時間範圍內的記錄。如果您的資料依 `created_at` 資料欄排序，Athena 可以使用檔案中繼資料中的最小值和最大值來略過資料檔案中不需要的部分。

使用單欄式檔案格式時，請確定檔案不會太小。如 [避免檔案太多](#) 中所述，含有許多小型檔案的資料集會造成效能問題。對於單欄式檔案格式而言，尤其如此。對於小型檔案，單欄式檔案格式的負荷超過優勢。

請注意，Parquet 和 ORC 在內部依資料列群組 (Parquet) 和條紋 (ORC) 進行整理。資料列群組的預設大小為 128 MB，而條紋的預設大小則為 64 MB。如果您有許多資料欄，您可以增加資料列群組和條紋大小，以提升效能。不建議將資料列群組或條紋大小減少至小於其預設值。

若要將其他資料格式轉換為鑲木地板或 ORC，您可以使用 AWS Glue ETL 或 Athena。如需有關使用 Athena for ETL 的詳細資訊，請參閱 [使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析](#)。

## 壓縮資料

Athena 支援各種壓縮格式。查詢壓縮資料的速度更快，也更便宜，因為您需要為解壓縮前掃描的位元組數量進行支付。

[gzip](#) 格式提供了良好的壓縮比，並且在其他工具和服務中具有廣泛的支援。[zstd](#) (Zstandard) 格式是一種較新的壓縮格式，可使效能和壓縮比之間達到良好的平衡。

壓縮文字檔案 (例如 JSON 和 CSV 資料) 時，請嘗試在檔案數量和檔案大小之間取得平衡。大多數壓縮格式都要求讀者從頭開始讀取檔案。這表示一般而言，壓縮的文字檔案無法平行處理。大型未壓縮的檔案通常會在工作者之間分割，以在查詢處理期間達到更高的平行處理能力，但大多數壓縮格式無法執行這項操作。

如 [避免檔案太多](#) 中所述，檔案最好不要太多也不宜過少。由於檔案數目是可處理查詢的工作者數量限制，因此對於壓縮檔案而言，此規則尤其如此。

如需有關在 Athena 中使用壓縮的詳細資訊，請參閱 [Athena 壓縮支援](#)。

### 使用歸納來查詢具有高基數的索引鍵

歸納是一種技術，可根據其中一個資料欄的值，將記錄分佈至個別檔案中。這能確保具有相同值的所有記錄都位於同一個檔案中。當您擁有高基數的索引鍵，而且許多查詢都會查詢索引鍵的特定值時，歸納功能非常有用。

例如，假設您查詢特定使用者的一組記錄。如果資料是依使用者 ID 歸納，Athena 會事先知道哪些檔案包含特定 ID 的記錄，哪些檔案不包含。這可讓 Athena 僅讀取包含 ID 的檔案，進而大幅減少讀取的資料量。這同意也可減少在資料中搜尋特定 ID 所需的運算時間。

### 歸納的缺點

當查詢經常在資料歸納的資料欄中搜尋多個值時，歸納的價值就不太重要。查詢的值越多，必須讀取所有或大部分檔案的可能性就越高。例如，如果您有三個儲存貯體，而查詢會尋找三個不同的值，則可能必須讀取所有檔案。當查詢在查找單一值時，歸納效果最佳。

如需詳細資訊，請參閱 [在 Athena 中分割和歸納](#)。

### 避免檔案太多

由許多小型檔案組成的資料集會導致整體查詢效能不佳。Athena 計劃查詢時，會列出所有分割區位置，而這需要花費一些時間。處理和請求每個檔案也會產生運算負載。因此，從 Amazon S3 載入單一較大的檔案比從許多較小的檔案載入相同記錄更快。

在極端情況下，您可能會遇到 Amazon S3 服務限制。針對單一索引分割區，Amazon S3 每秒最多可支援 5,500 個請求。最初，儲存貯體會被視為單一索引分割區，但隨著請求載入的增加，它可以分割成多個索引分割區。

Amazon S3 會根據索引鍵字首查詢請求模式和分割。如果您的資料集包含數千個檔案，則來自 Athena 的請求可能會超出請求的配額。即使檔案較少，如果對同一個資料集進行多個並行查詢，則可能會超過配額。存取相同檔案的其他應用程式可能會增加請求總數。

當超出請求率 limit 時，Amazon S3 會傳回下列錯誤。此錯誤已包含在 Athena 查詢的狀態資訊中。

SlowDown：請降低您的請求率

若要進行疑難排解，請先判斷錯誤是由單一查詢還是由可讀取相同檔案的多個查詢造成的。如果是後者，請協調查詢的執行，以便它們不會同時執行。為此，請在應用程式中新增佇列機制，甚至是重試。

如果執行單一查詢觸發錯誤，請嘗試合併資料檔案或修改查詢，以讀取較少的檔案。合併小型檔案的最佳時間是在將其寫入之前。為此，請考慮下列技巧：

- 變更寫入檔案的程序，以寫入較大的檔案。例如，您可以在寫入記錄之前緩衝一段較長的時間。
- 將檔案放在 Amazon S3 上的某個位置，然後使用 Glue ETL 等工具將其合併為較大的檔案。然後，將較大的檔案移至資料表所指向的位置。如需詳細資訊，請參閱 [AWS Glue 開發人員指南中的讀取較大群組的輸入檔案](#) 和 [如何設定 AWS Glue ETL 工作以輸出較大的檔案？](#) 在 AWS RE：發布知識中心。
- 減少分割區索引鍵的數量。當您的分割區索引鍵太多時，每個分割區可能只有少數記錄，因此會產生過多的小型檔案。如需有關決定要建立哪些分割區的資訊，請參閱 [選擇將支援您查詢的分割區索引鍵](#)。

避免分割區以外的其他儲存體階層

若要避免查詢規劃負荷，請將檔案儲存在每個分割區位置的單層式結構中。請勿使用任何其他目錄階層。

Athena 計劃查詢時，會列出與查詢相符的所有分割區中的所有檔案。雖然 Amazon S3 本身沒有目錄，但慣例是將 / 斜線解譯為目錄分隔符號。當 Athena 列出分割區位置時，其會遞歸列出找到的任何目錄。當分割區內的檔案整理成階層時，會出現多輪清單。

當所有檔案都直接位於分割區位置時，大多數情況下只需執行一個清單操作。但是，如果一個分割區中有 1000 個以上的檔案，則需要多個連續清單操作，因為 Amazon S3 每個清單操作只會傳回 1000 個物件。分區中有 1000 多個檔案也可能會導致其他更嚴重的效能問題。如需詳細資訊，請參閱 [避免檔案太多](#)。

僅在必要時使用 SymlinkTextInputFormat

使用 [SymlinkTextInputFormat](#) 技術可以是一種解決相關情況的方法，當資料表的檔案沒有整齊地整理到分割區中時。例如，當所有檔案都使用相同的字首，或是具有不同結構描述的檔案位於相同位置時，符號連結可能很有用。

但是，使用符號連結會為查詢執行增加間接層級。這些間接層級會影響整體效能。必須讀取符號連結檔案，並且必須列出它們定義的位置。如此會新增至 Amazon S3 的多次往返，而這並非常見 Hive 資料表所需的。總之，只有當沒有更好的選項 (例如整理檔案) 可用時，您才應使用 `SymlinkTextInputFormat`。

## 其他資源

如需 Athena 效能調校的其他相關資訊，請參閱下列資源：

- 閱讀 AWS 大數據部落格文章 [Amazon Athena 的十大效能調整秘訣](#)
- 如需有關使用述詞下推以改善聯合查詢中的效能的文章，請參閱 AWS 大數據部落格中的 [使用 Amazon Athena 中的述詞下推改善聯合查詢](#)。
- 如需 Athena 查詢引擎效能最佳化的文章，請參閱 AWS 大數據部落格中 [最新的 Amazon Athena 引擎，以快 3 倍的速度執行查詢，節省高達 70% 的成本](#)。
- 閱讀 [AWS 大數據部落格中的其他 Athena 文章](#)
- 使用 Amazon Athena 標籤在 [AWS re:Post](#) 上詢問問題
- 請參閱 [AWS 知識中心的 Athena 主題](#)
- 連絡人 AWS Support (在中 AWS Management Console，按一下 [Support]、[Support 中心])

## 防止 Amazon S3 限流

限流是限制您使用服務、應用程式或系統的速率的程序。在中 AWS，您可以使用節流來防止過度使用 Amazon S3 服務，並為所有使用者提高 Amazon S3 的可用性和回應能力。但是，由於限流限制了資料傳輸到 Amazon S3 或從 Amazon S3 傳輸的速率，因此請務必考慮防止您的互動受到限流。

### 減少服務層級的限流

若要避免 Amazon S3 在服務層級的限流，您可以監控您的用量並調整 [服務配額](#)，或者使用分割等特定技術。以下某些條件可能導致限流：

- 超出帳戶的 API 請求限制 – Amazon S3 具有基於帳戶類型和用量的預設 API 請求限制。如果您超過單一物件每秒的請求數目上限，則您的請求可能會受到限流，以防止 Amazon S3 服務多載。
- 資料分割不足 – 如果您沒有正確地分割資料並傳輸大量資料，Amazon S3 可能會對您的請求限流。如需有關分割的詳細資訊，請參閱本文件中的 [使用分割區](#) 部分。
- 大量小型物件 – 如果可能，請避免擁有大量小型檔案。Amazon S3 每個分割的字首具有每秒 [5500 個 GET 請求](#) 的限制，而您的 Athena 查詢也共用此相同限制。如果您在單一查詢中掃描數百萬個小型物件，您的查詢將很容易受到 Amazon S3 限流。

若要避免過度掃描，您可以使用 AWS Glue ETL 定期壓縮檔案，或者分割表格並新增分割索引鍵篩選器。如需詳細資訊，請參閱下列資源。

- [如何配置 AWS Glue ETL 任務以輸出更大的文件？](#) (AWS 知識中心)
- [讀取較大群組中的輸入檔案](#) (AWS Glue 開發人員指南)

## 優化資料表

如果遇到限流問題，建構資料很重要。雖然 Amazon S3 可以處理大量資料，但有時會因為資料的建構方式而發生限流。

下列各節就如何在 Amazon S3 中建構資料以避免限流問題提供了一些建議。

## 使用分割區

您可以限制在任何指定時間內存取的資料量，使用分割來減少限流。藉由分割特定資料欄上的資料，您可以將請求平均分散至多個物件，並減少單一物件的請求數目。減少必須掃描的資料量會改善查詢效能並降低成本。

您可以在建立資料表時，定義可做為虛擬資料欄的分割區。若要在 CREATE TABLE 陳述式中建立含有分割區的資料表，您可以使用 PARTITIONED BY (*column\_name data\_type*) 子句來定義分割資料的索引鍵。

若要限制查詢掃描的分割區，您可以在查詢的 WHERE 子句中將其指定為述詞。因此，經常用作篩選器的資料欄是分割的理想候選者。常見做法是根據時間間隔來分割資料，這通常會產生多層級的分割機制。

請注意，分割也有成本。當您增加資料表中的分割區數目時，擷取和處理分割區中繼資料所需的時間也會增加。因此，過度分割會消除您透過更明智分區而獲得的益處。如果您的資料明顯集中於一個分割區值，並且大多數查詢使用該值，則可能會產生額外開銷。

如需有關在 Athena 中分割的詳細資訊，請參閱 [什麼是分割？](#)

## 歸納您的資料

分割資料的另一個方法是將資料歸納在單一分割區內。使用歸納功能時，您可以指定一或多個資料欄，其中包含要組合在一起的資料列。然後，將這些資料列放入多個儲存貯體中。如此一來，您只會查詢必須讀取的儲存貯體，從而減少必須掃描的資料列數目。

當您選取要用於歸納的資料欄時，請選取具有高基數 (亦即，具有許多不同值)、均勻分散且經常用來篩選資料的資料欄。主索引鍵 (例如 ID 資料欄) 是用來歸納的良好資料欄範例。

如需有關在 Athena 中歸納的詳細資訊，請參閱 [什麼是歸納？](#)。

## 使用 AWS Glue 分割區索引

您可以使用資料 AWS Glue 分割索引，根據一或多個分割區的值來組織資料表中的資料。AWS Glue 資料分割索引可以減少資料傳輸次數、資料處理量，以及處理查詢的時間。

AWS Glue 分割區索引是一種中繼資料檔案，其中包含資料表中分割區的相關資訊，包括分割區索引鍵及其值。分割區索引存放在 Amazon S3 儲存貯體中，並在新分 AWS Glue 分割區新增至表格時自動更新。

存在 AWS Glue 分割區索引時，查詢會嘗試擷取分割區的字集，而不是載入資料表中的所有分割區。查詢只會在與查詢相關的資料子集上執行。

當您在中建立資料表時 AWS Glue，您可以在資料表上定義的任何分割區索引鍵組合上建立分割區索引。在資料表上建立一或多個分割區索引之後，您必須將屬性新增至啟用分割區篩選的資料表。然後，您可以從 Athena 查詢資料表。

如需有關在中建立分割區索引的資訊 AWS Glue，請參閱 [開關 AWS Glue 發人員指南 AWS Glue 中的使用分割區索引](#)。如需有關新增資料表屬性以啟用分割區篩選的資訊，請參閱 [AWS Glue 分割區索引和篩選](#)。

## 使用資料壓縮和檔案分割

如果檔案處於最佳大小，或可以將檔案分割成邏輯群組，則資料壓縮可以大幅加快查詢速度。一般而言，較高的壓縮比需要更多的 CPU 週期來壓縮和解壓縮資料。對於 Athena，建議您使用 Apache Parquet 或 Apache ORC，因為其預設會壓縮資料。如需有關 Athena 中資料壓縮的資訊，請參閱 [Athena 壓縮支援](#)。

分割檔案可讓 Athena 將讀取單一檔案的任務分散至多個讀取器，藉此提高平行處理能力。如果單一檔案不可分割，則只有一個讀取器可以讀取檔案，而其他讀取器則處於閒置狀態。Apache Parquet 和 Apache ORC 還支援可分割的檔案。

## 使用優化單欄式資料存放區

如果您將資料轉換為單欄式格式，則 Athena 查詢效能會大幅提升。當您產生單欄式檔案時，需要考量的一種優化技術是根據資料分割區索引鍵排序資料。

Apache Parquet 和 Apache ORC 是常用的開放原始碼單欄式資料存放區。如需有關將現有 Amazon S3 資料來源轉換為其中一種格式的資訊，請參閱 [轉換為單欄式格式](#)。

## 使用較大的 Parquet 區塊大小或 ORC 條紋大小

Parquet 和 ORC 具有資料儲存參數，您可以調整以進行優化。在 Parquet 中，您可以優化區塊大小。在 ORC 中，您可以優化條紋大小。區塊或條紋越大，您可以在其中儲存的資料列就越多。依預設，Parquet 區塊大小為 128 MB，而 ORC 條紋大小為 64 MB。

如果 ORC 條紋小於 8 MB (預設值為 `hive.orc.max_buffer_size`)，則 Athena 會讀取整個 ORC 條紋。對於較小的條紋，這是 Athena 在資料欄選取性和每秒輸入/輸出操作之間進行的權衡。

如果您的資料表包含大量資料欄，則較小的區塊或條紋大小可能會導致掃描的資料超過必要的數量。在這些情況下，較大的區塊大小可能會更有效率。

## 對複雜類型使用 ORC

目前，當您查詢存放在 Parquet 中且具有複雜資料類型 (例如，`array`、`map` 或 `struct`) 的資料欄時，Athena 會讀取整個資料列，而不是只選擇性地讀取指定的資料欄。這是 Athena 的已知問題。解決方法是考慮使用 ORC。

## 選擇壓縮演算法

您可以設定的另一個參數是資料區塊上的壓縮演算法。如需有關 Athena 中 Parquet 和 ORC 支援的壓縮演算法的資訊，請參閱 [Athena 壓縮支援](#)。

如需有關在 Athena 中最佳化單欄式儲存格式的詳細資訊，請參閱大數據部落格文章 [Amazon Athena 十 AWS 大效能調整秘訣](#) 中的「最佳化單欄式資料存放區產生」一節。

## 使用 Iceberg 資料表

Apache Iceberg 是開放式的資料表格式，專用於非常大型的分析資料集，旨在優化 Amazon S3 上的用量。您可以使用 Iceberg 資料表來協助減少 Amazon S3 中的限流。

Iceberg 資料表可為您提供下列優點：

- 您可以在一或多個資料欄上對 Iceberg 資料表進行分割。這樣可優化資料存取，並減少查詢必須掃描的資料量。
- 由於 Iceberg 物件儲存模式可將 Iceberg 資料表優化以與 Amazon S3 搭配使用，因此其可以處理大量資料和繁重的查詢工作負載。
- 物件儲存模式中的 Iceberg 資料表具備可擴展性、容錯能力和耐用性，有助於減少限流。
- ACID 交易支援意味著，多個使用者可以以原子方式新增和刪除 Amazon S3 物件。

如需有關 Apache Iceberg 的詳細資訊，請參閱 [Apache Iceberg](#)。如需有關在 Athena 中使用 Apache Iceberg 資料表的詳細資訊，請參閱 [使用 Iceberg 資料表](#)。

## 優化查詢

使用本節中的建議優化 Athena 中的 SQL 查詢。

### 使用 LIMIT 與 ORDER BY 子句

ORDER BY 子句會以排序順序傳回資料。這需要 Athena 將所有資料列傳送至單一工作節點，然後排序資料列。這種查詢類型可以執行很長一段時間，甚至失敗。

為了提高查詢的效率，請查看頂部或底部的  $N$  個值，然後使用 LIMIT 子句。這樣可以將排序和限制推送至個別工作節點，而不是單一工作節點，進而大幅降低排序的成本。

### 優化 JOIN 子句

當您聯結兩個資料表時，Athena 會將右側的資料表分散至工作節點，然後串流左側的資料表以執行聯結。

因此，請在聯結左側指定較大的資料表，並在聯結的右側指定較小的資料表。如此一來，Athena 會使用較少的記憶體，並以較低的延遲執行查詢。

另外，請注意以下重點：

- 當您使用多個 JOIN 命令時，請按從最大到最小的順序指定資料表。
- 除非查詢需要交叉聯結，否則請予以避免。

### 優化 GROUP BY 子句

GROUP BY 運算子會根據 GROUP BY 資料欄將資料列分散至工作節點。我們會在記憶體中引用這些資料欄，並在擷取資料列時比較這些值。當 GROUP BY 資料欄相符時，這些值彙總在一起。考慮到此程序的運作方式，建議按從最高基數到最低基數的順序排序資料欄。

### 使用數字而不是字串

由於相較於字串，數字需要的記憶體較少，而且處理速度更快，因此請盡可能使用數字而不是字串。

### 限制資料欄的數目

若要減少儲存資料所需的記憶體總量，請限制 SELECT 陳述式中指定的資料欄數。

## 使用規則表達式而不是 LIKE

包含子句 (例如大型字串上的 LIKE '%string%') 的查詢可能需要非常大量的運算。當您篩選字串資料欄上的多個值時，請改用 [regexp\\_like\(\)](#) 函數和規則表達式。當您比較一長串值時，這特別有用。

## 使用 LIMIT 子句

當您執行查詢時，請不要選取所有資料欄，而是使用 LIMIT 子句，以只傳回您需要的資料欄。這樣做可減少透過查詢執行管道處理的資料集大小。當您查詢具有大量字串型資料欄的資料表時，LIMIT 子句會更有幫助。當您對任何查詢執行多個聯結或彙總時，LIMIT 子句也很有用。

## 其他資源

《Amazon Simple Storage Service 使用者指南》中的[最佳實務設計模式：優化 Amazon S3 效能](#)。

## [Athena 中的效能調校](#)

# Athena 壓縮支援

## 主題

- [指定壓縮格式](#)
- [指定未壓縮](#)
- [備註和資源](#)
- [檔案格式支援 Hive 資料表壓縮](#)
- [檔案格式支援 Iceberg 資料表壓縮](#)
- [在 Athena 中使用 ZSTD 壓縮級別](#)

Athena 支援各種壓縮格式來讀取和寫入資料，包括從使用多種壓縮格式的資料表讀取。例如，當某些 Parquet 檔案使用 Snappy 壓縮而其他 Parquet 檔案使用 GZIP 壓縮時，Athena 可以成功讀取使用 Parquet 檔案格式的資料表中的資料。相同的原則適用於 ORC、文字檔案和 JSON 儲存格式。

Athena 支援以下壓縮格式：

- BZIP2 – 使用 Burrows-Wheeler 演算法的格式。
- DEFLATE – 基於 [LZSS](#) 和 [Huffman 編碼](#) 的壓縮演算法。[Deflate](#) 僅適用於 Avro 檔案格式。
- GZIP – 以 Deflate 為基礎的壓縮演算法。對於 Athena 引擎版本 2 和 3 中的 Hive 資料表，以及 Athena 引擎版本 2 中的 Iceberg 資料表，GZIP 是採用 Parquet 格式和文字檔案存儲格式之檔案的預設寫入壓縮格式。不支援 tar.gz 格式的檔案。

- LZ4 – Lempel-Ziv 77 (LZ7) 系列的這個成員也著重於壓縮和解壓縮速度，而不是資料的最大壓縮。LZ4 具有以下框架格式：
  - LZ4 原始/無框架 – LZ4 區塊壓縮格式的無框架標準實作。若要取得更多資訊，請參閱中的 [LZ4 區塊格式描述](#)。GitHub
  - LZ4 框架 – LZ4 的一般框架實作。如需詳細資訊，請參閱中的 [LZ4 框架格式說明](#)。GitHub
  - LZ4 Hadoop 相容 – LZ4 的 Apache Hadoop 實作。此實作會使用 .java 類別封裝 LZ4 壓縮 [BlockCompressorStream](#)。
- LZO – 使用 Lempel-Ziv-Oberhumer 演算法的格式，該演算法著重於高壓縮和解壓縮速度，而不是資料的最大壓縮。LZO 有兩種實作：
  - 標準 LZO – 如需詳細資訊，請參閱 Oberhumer 網站上的 LZO [摘要](#)。
  - LZO 哈多波相容 — [此實作會以 .java 類別封裝 LZO 演算法。BlockCompressorStream](#)
- SNAPPY – 屬於 Lempel-Ziv 77 (LZ7) 系列的壓縮演算法。Snappy 著重於高壓縮和解壓縮速度，而不是資料的最大壓縮。
- ZLIB – 根據 Deflate，ZLIB 是 ORC 資料儲存格式中檔案的預設寫入壓縮格式。如需詳細資訊，請參閱中的 [zlib](#) 頁面 GitHub。
- ZSTD – [Zstandard 即時資料壓縮演算法](#) 是一種提供高壓縮比的快速壓縮演算法。Zstandard (ZSTD) 程式庫是使用 BSD 授權作為開放原始碼所提供。ZSTD 是 Iceberg 資料表的預設壓縮。寫入 ZSTD 壓縮資料時，Athena 預設使用 ZSTD 壓縮級別 3。如需有關在 Athena 中使用 ZSTD 壓縮級別的詳細資訊，請參閱 [在 Athena 中使用 ZSTD 壓縮級別](#)。

## 指定壓縮格式

當您寫入 CREATE TABLE 或 CTAS 陳述式時，您可以指定壓縮屬性，以指定 Athena 寫入這些資料表時要使用的壓縮類型。

- 如需 CTAS 的資訊，請參閱 [CTAS 資料表屬性](#)。如需範例，請參閱 [CTAS 查詢的範例](#)。
- 如需 CREATE TABLE 的資訊，請參閱 [ALTER TABLE SET TBLPROPERTIES](#) 以取得壓縮資料表屬性的清單。

## 指定未壓縮

CREATE TABLE 陳述式支援寫入未壓縮的檔案。若要編寫未壓縮檔案，請使用下列語法：

- CREATE TABLE (文字檔案或 JSON) – 在 TBLPROPERTIES 中，指定 `write.compression = NONE`。

- CREATE TABLE (Parquet) – 在 TBLPROPERTIES 中，指定 `parquet.compression = UNCOMPRESSED`。
- CREATE TABLE (ORC) – 在 TBLPROPERTIES 中，指定 `orc.compress = NONE`。

## 備註和資源

- 目前 Athena 無法識別大寫的副檔名，諸如 `.GZ` 或 `.BZIP2`。避免使用大寫副檔名的資料集，或將資料副檔名重新以小寫字母命名。
- 若為 CSV、TSV 和 JSON 的資料，Athena 會根據副檔名決定壓縮類型。如果不存在任何副檔名，Athena 會將資料視為未壓縮的純文字。如果您的資料已壓縮，請確定檔案名稱包含副檔名，例如 `gz`。
- 不支援 ZIP 檔案格式。
- 若要查詢來自 Athena 的 Amazon 資料 Firehose 日誌，支援的格式包括 GZIP 壓縮或具有 SNAPPY 壓縮的 ORC 檔案。
- 如需使用壓縮的詳細資訊，請參閱 AWS 大數據部落格文章 [Amazon Athena 十大效能調整秘訣](#) 的第 3 節（「壓縮和分割檔案」）。

## 檔案格式支援 Hive 資料表壓縮

Athena 中的 Hive 壓縮支援取決於引擎版本。

### Athena 引擎版本 3 中的 Hive 壓縮支援

下列資料表摘要說明 Athena 引擎版本 3 對 Apache Hive 中的儲存檔案格式的壓縮格式支援。文字檔案格式包括 TSV、CSV、JSON 和文字的自訂 SerDes。儲存格中的「是」或「否」同樣適用於讀取和寫入操作，除非另有註明。基於本資料表的用途，CREATE TABLE、CTAS 和 INSERT INTO 會被視為寫入操作。如需有關在 Athena 中使用 ZSTD 壓縮級別的詳細資訊，請參閱 [在 Athena 中使用 ZSTD 壓縮級別](#)。

	Avro	Ion	ORC	Parquet	文字檔案
BZIP2	是	是	否	否	是
DEFLATE	是	否	否	否	否
GZIP	否	是	否	是	是

	Avro	Ion	ORC	Parquet	文字檔案
LZ4	否	是	是	是	是
LZO	否	寫入 - 否 讀取 - 是	否	是	寫入 - 否 讀取 - 是
SNAPPY	是	是	是	是	是
ZLIB	否	否	是	否	否
ZSTD	是	是	是	是	是
NONE	是	是	是	是	是

### Athena 引擎版本 2 中的 Hive 壓縮支援

下列資料表摘要說明 Athena 引擎版本 2 對 Apache Hive 的壓縮格式支援。文字檔案格式包括 TSV、CSV、JSON 和文字的自訂 SerDes。儲存格中的「是」或「否」同樣適用於讀取和寫入操作，除非另有註明。基於本資料表的用途，CREATE TABLE、CTAS 和 INSERT INTO 會被視為寫入操作。

	Avro	Ion	ORC	Parquet	文字檔案
BZIP2	是	是	否	否	是
DEFLATE	是	否	否	否	否
GZIP	否	是	否	是	是
LZ4	否	否	是	寫入 - 是 讀取 - 否	寫入 - 否 讀取 - 是
LZO	否	寫入 - 否 讀取 - 是	否	是	寫入 - 否 讀取 - 是
SNAPPY	是	是	是	是	是

	Avro	Ion	ORC	Parquet	文字檔案
ZLIB	否	否	是	否	否
ZSTD	否	是	是	是	是
NONE	是	是	是	是	是

## 檔案格式支援 Iceberg 資料表壓縮

Athena 中的 Apache Iceberg 壓縮支援取決於引擎版本。

### Athena 引擎版本 3 中的 Iceberg 壓縮支援

下列資料表摘要說明 Athena 引擎版本 3 對 Apache Iceberg 中的儲存檔案格式的壓縮格式支援。儲存格中的「是」或「否」同樣適用於讀取和寫入操作，除非另有註明。基於本資料表的用途，CREATE TABLE、CTAS 和 INSERT INTO 會被視為寫入操作。Athena 引擎版本 3 中的 Iceberg 的預設儲存格式是 Parquet。Athena 引擎版本 3 中的 Iceberg 的預設壓縮格式是 ZSTD。如需有關在 Athena 中使用 ZSTD 壓縮級別的詳細資訊，請參閱 [在 Athena 中使用 ZSTD 壓縮級別](#)。

	Avro	ORC	Parquet (預設)
BZIP2	否	否	否
GZIP	是	否	是
LZ4	否	是	否
SNAPPY	是	是	是
ZLIB	否	是	否
ZSTD	是	是	是 (預設)
NONE	是 (指定 None 或 Deflate)	是	是 (指定 None 或 Uncompressed )

## Athena 引擎版本 2 中的 Iceberg 壓縮支援

下列資料表摘要說明 Athena 引擎版本 2 對 Apache Iceberg 的壓縮格式支援。儲存格中的「是」或「否」同樣適用於讀取和寫入操作，除非另有註明。基於本資料表的用途，CREATE TABLE、CTAS 和 INSERT INTO 會被視為寫入操作。Athena 引擎版本 2 中的 Iceberg 的預設儲存格式是 Parquet。Athena 引擎版本 2 中的 Iceberg 的預設壓縮格式是 GZIP。

	Avro (不支援)	ORC (不支援)	Parquet (預設)
BZIP2	否	否	否
GZIP	否	否	是 (預設)
LZ4	否	否	否
SNAPPY	否	否	是
ZLIB	否	否	否
ZSTD	否	否	是
NONE	否	否	是

## 在 Athena 中使用 ZSTD 壓縮級別

[Zstandard 即時資料壓縮演算法](#) 是一種提供高壓縮比的快速壓縮演算法。Zstandard (ZSTD) 程式庫是開放原始碼軟體，使用 BSD 授權。Athena 支援讀取和寫入 ZStandard 壓縮的 ORC、Parquet 和文字檔案資料。

您可以使用 ZSTD 壓縮級別根據自己的要求調整壓縮比和速度。ZSTD 庫支援從 1 到 22 的壓縮級別。Athena 預設使用 ZSTD 壓縮級別 3。

壓縮級別在壓縮速度和達到的壓縮量之間提供了細微的權衡。壓縮級別越低，速度越快，但檔案大小更大。例如，如果速度最重要，則可以使用級別 1，如果大小最重要，則可以使用級別 22。級別 3 適用於許多使用案例，且為預設值。請謹慎使用大於 19 的級別，因為這些級別需要更多的記憶體。ZSTD 庫還提供負壓縮級別，其可擴展壓縮速度和比率的範圍。如需詳細資訊，請參閱 [Zstandard 壓縮 RFC](#)。

豐富的壓縮級別為微調提供了大量機會。但是，請確保在決定壓縮級別時測量資料並考慮權衡。建議使用預設級別 3，或 6 到 9 之間的級別，以便在壓縮速度與壓縮資料大小之間取得合理的權衡。對於大小最重要且壓縮速度不是問題的情況，預留級別 20 或更高。

## 考量與限制

在 Athena 中使用 ZSTD 壓縮級別時，請考慮下列幾點。

- 只有 Athena 引擎版本 3 中才支援 ZSTD `compression_level` 屬性。
- ALTER TABLE、CREATE TABLE、CREATE TABLE AS (CTAS) 以及 UNLOAD 陳述式支援 ZSTD `compression_level` 屬性。
- `compression_level` 屬性是選用屬性。
- 只有 ZSTD 壓縮才支援 `compression_level` 屬性。
- 可能的壓縮級別為 1 到 22。
- 預設壓縮級別為 3。

如需有關 Athena 中 Apache Hive ZSTD 壓縮支援的資訊，請參閱 [檔案格式支援 Hive 資料表壓縮](#)。

如需有關 Athena 中 Apache Iceberg ZSTD 壓縮支援的資訊，請參閱 [檔案格式支援 Iceberg 資料表壓縮](#)。

## 指定 ZSTD 壓縮級別

若要指定 ALTER TABLE、CREATE TABLE、CREATE TABLE AS 和 UNLOAD 陳述式的 ZSTD 壓縮級別，請使用 `compression_level` 屬性。若要指定 ZSTD 壓縮本身，您必須使用陳述式語法所使用的單獨壓縮屬性。

### ALTER TABLE SET TBLPROPERTIES

在 [ALTER TABLE SET TBLPROPERTIES](#) 陳述式 SET TBLPROPERTIES 子句中，使用 `'write.compression' = 'ZSTD'` 或 `'parquet.compression' = 'ZSTD'` 指定 ZSTD 壓縮。然後使用 `compression_level` 屬性來指定 1 到 22 之間的值 (例如，`'compression_level' = 5`)。如果您沒有指定壓縮級別屬性，則壓縮級別會預設為 3。

## 範例

下列範例會修改資料表 `existing_table`，以使用具有 ZSTD 壓縮和 ZSTD 壓縮級別 4 的 Parquet 檔案格式。請注意，壓縮級別值必須輸入為字串而不是整數。

```
ALTER TABLE existing_table
```

```
SET TBLPROPERTIES ('parquet.compression' = 'ZSTD', 'compression_level' = 4)
```

## CREATE TABLE

在 [CREATE TABLE](#) 陳述式 TBLPROPERTIES 子句中，指定 'write.compression' = 'ZSTD' 或 'parquet.compression' = 'ZSTD'，然後使用 `compression_level = compression_level` 並指定介於 1 到 22 之間的值。如果未指定 `compression_level` 屬性，則預設壓縮級別為 3。

### 範例

下列範例使用 ZSTD 壓縮和 ZSTD 壓縮級別 4 建立 Parquet 檔案格式的資料表。

```
CREATE EXTERNAL TABLE new_table (  
  `col0` string COMMENT '',  
  `col1` string COMMENT ''  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'  
TBLPROPERTIES ('write.compression' = 'ZSTD', 'compression_level' = 4)
```

## CREATE TABLE AS (CTAS)

在 [CREATE TABLE AS](#) 陳述式 WITH 子句中，指定 `write_compression = 'ZSTD'` 或 `parquet_compression = 'ZSTD'`，然後使用 `compression_level = compression_level` 並指定介於 1 到 22 之間的值。如果未指定 `compression_level` 屬性，則預設壓縮級別為 3。

### 範例

下列 CTAS 範例使用 ZSTD 壓縮和 ZSTD 壓縮級別 4，將 Parquet 指定為檔案格式。

```
CREATE TABLE new_table  
WITH ( format = 'PARQUET', write_compression = 'ZSTD', compression_level = 4)  
AS SELECT * FROM old_table
```

## UNLOAD

在 [UNLOAD](#) 陳述式 WITH 子句中，指定 `compression = 'ZSTD'`，然後使用 `compression_level = compression_level` 並指定介於 1 到 22 之間的值。如果未指定 `compression_level` 屬性，則預設壓縮級別為 3。

## 範例

下列範例會使用 Parquet 檔案格式、ZSTD 壓縮和 ZSTD 壓縮級別 4，將查詢結果卸載到指定的位置。

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/'
WITH (format = 'PARQUET', compression = 'ZSTD', compression_level = 4)
```

## 標記 Athena 資源

每個標籤皆包含由您定義的金鑰和值。標記 Athena 資源時，您可以將自訂中繼資料指派給該資源。您可以使用標籤以不同方式分類 AWS 資源，例如依用途、擁有者或環境。在 Athena 中，工作群組、資料目錄和容量保留等資源都是可標記的資源。例如，您可以為帳戶中的工作群組建立一組標籤，以協助您追蹤工作群組擁有者，或依用途來識別工作群組。如果您也在 Billing and Cost Management 主控台中啟用標籤作為成本分配標籤，則與執行查詢相關聯的成本會顯示在「成本和用量報告」中，並含有該成本分配標籤。我們建議您使用 AWS [標記最佳實務](#) 來建立一組一致的標籤，以滿足您組織的需求。

您可以使用 Athena 主控台或 API 操作來處理標籤。

### 主題

- [標籤基本概念](#)
- [標籤限制](#)
- [在主控台的工作群組上使用標籤](#)
- [使用標籤操作](#)
- [標籤型 IAM 存取控制政策](#)

## 標籤基本概念

標籤是您指派給 Athena 資源的標籤。每個標籤皆包含由您定義的一個金鑰與一個選用值。

標籤可讓您以不同方式來分類 AWS 資源。例如，您可以為您帳戶的工作群組定義一組標籤，以協助您追蹤每個工作群組擁有者或用途。

您可以在建立新的 Athena 工作群組或資料目錄時新增標籤，或從中新增、編輯或移除標籤。您可以在主控台編輯標籤。如果您使用 API 操作來編輯標籤，請移除舊標籤，再加入新標籤。如果您刪除資源，也會刪除任何該資源的標籤。

Athena 不會自動為您的資源指派標籤。您可以編輯標籤金鑰和值，並且可以隨時從資源移除標籤。您可以將標籤的值設為空白字串，但您無法將標籤的值設為 Null。請勿將重複的標籤鍵新增至相同的資

源。如果這樣做，Athena 會發出錯誤訊息。如果您使用 TagResource 動作來標記使用現有標籤鍵的資源，則新標籤值會覆寫舊值。

在 IAM 中，您可以控制 Amazon Web Services 帳戶中的哪些使用者具有建立、編輯、移除或列出標籤的許可。如需詳細資訊，請參閱[標籤型 IAM 存取控制政策](#)。

如需 Amazon Athena 標籤動作的完整清單，請參閱 [Amazon Athena API 參考](#) 中的 API 動作名稱。

您可以使用標籤來計費。如需詳細資訊，請參閱《AWS Billing and Cost Management 使用者指南》中的[使用標籤計費](#)。

如需詳細資訊，請參閱[標籤限制](#)。

## 標籤限制

標籤有以下限制：

- 在 Athena 中，您可以標記工作群組和資料目錄。您無法標記查詢。
- 每一資源標籤數最多為 50。若要保持在限制內，請檢閱和刪除未使用的標籤。
- 對於每一個資源，每個標籤金鑰必須是唯一的，且每個標籤金鑰只能有一個值。請勿將重複的標籤鍵同時新增到相同的工作群組。如果這樣做，Athena 會發出錯誤訊息。如果您在個別的 TagResource 動作中使用現有的標籤鍵來標記資源，新標籤值會覆寫舊值。
- 標籤索引鍵長度為 1-128 個 UTF-8 Unicode 字元。
- 標籤值長度為 0-256 個 UTF-8 Unicode 字元。

您需要指定工作群組資源的 ARN，才能執行標記操作，例如新增、編輯、移除或列出標籤。

- Athena 可讓您使用字母、數字、以 UTF-8 表示的空格，以及下列字元：+ - = . \_ : / @。
- 標籤金鑰與值皆區分大小寫。
- 標籤鍵中的 "aws:" 前綴已保留供 AWS 使用。您無法編輯或刪除含有此字首的標籤索引鍵。含有此字首的標籤不算在每一資源的標籤限制內。
- 您指派的標籤僅供 Amazon Web Services 帳戶使用。

## 在主控台的工作群組上使用標籤

您可以使用 Athena 主控台，查看您帳戶中每個工作群組正在使用的標籤。您只能依工作群組來檢視標籤。Athena 主控台也可讓您一次在一個工作群組中套用、編輯或移除標籤。

您可以使用您建立的標籤來搜尋工作群組。

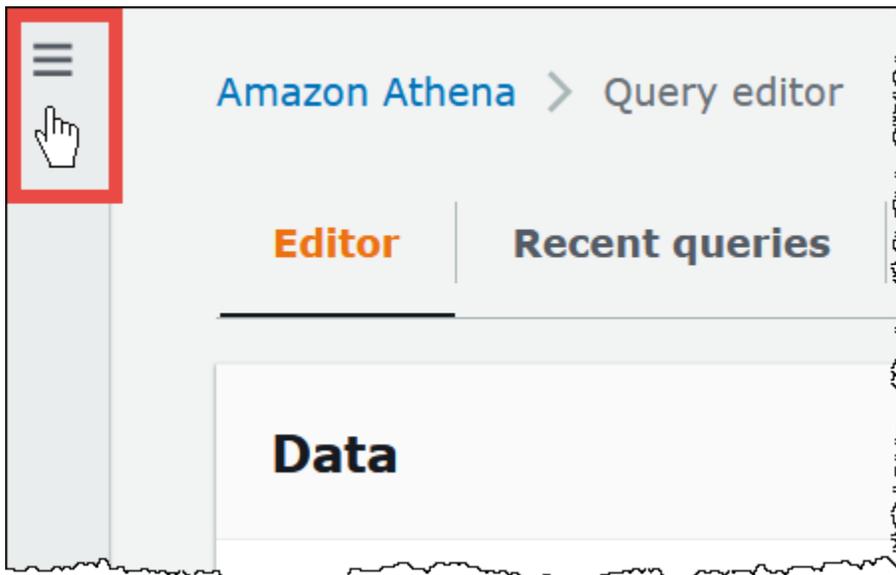
## 主題

- [顯示個別工作群組的標籤](#)
- [在個別工作群組上新增和刪除標籤](#)

### 顯示個別工作群組的標籤

在 Athena 主控台中顯示個別工作群組的標籤

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽選單上，選擇 Workgroups (工作群組)，然後選擇您要的工作群組。
4. 執行下列任意一項：
  - 選擇 Tags (標籤) 索引標籤。如果標籤清單很長，請使用搜尋方塊。
  - 選擇 Edit (編輯)，然後向下捲動到 Tags (標籤) 區段。

### 在個別工作群組上新增和刪除標籤

您可以直接從 Workgroups (工作群組) 索引標籤來管理個別工作群組的標籤。

**Note**

如果您希望使用者在主控台中建立工作群組時新增標籤，或在使用 CreateWorkGroup 動作時傳入標籤，請確定您授予使用者進行 TagResource 和 CreateWorkGroup 動作的 IAM 許可。

### 在建立新的工作群組時新增標籤

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在導覽選單中，選擇 Workgroups (工作群組)。
3. 選擇 Create workgroup (建立工作群組)，並視需要填寫值。如需詳細步驟，請參閱[建立工作群組](#)。
4. 在 Tags (標籤) 區段中，指定索引鍵和值來新增一或多個標籤。請勿將重複的標籤索引鍵同時新增到相同的工作群組。如果這樣做，Athena 會發出錯誤訊息。如需詳細資訊，請參閱[標籤限制](#)。
5. 完成時，選擇 Create Workgroup (建立工作群組)。

### 在現有的工作群組中新增或編輯標籤

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在導覽窗格中，選擇 Workgroups (工作群組)。
3. 選擇您想要修改的工作群組。
4. 執行下列任意一項：
  - 選擇 Tags (標籤) 索引標籤，然後選擇 Manage tags (管理標籤)。
  - 選擇 Edit (編輯)，然後向下捲動到 Tags (標籤) 區段。
5. 指定每個標籤的索引鍵和數值。如需詳細資訊，請參閱[標籤限制](#)。
6. 選擇 Save (儲存)。

### 從個別工作群組中刪除標籤

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 在導覽窗格中，選擇 Workgroups (工作群組)。
3. 選擇您想要修改的工作群組。
4. 執行下列任意一項：

- 選擇 Tags (標籤) 索引標籤，然後選擇 Manage tags (管理標籤)。
  - 選擇 Edit (編輯)，然後向下捲動到 Tags (標籤) 區段。
5. 在標籤清單中，選取您要刪除標籤的 Remove (移除)，然後選擇 Save (儲存)。

## 使用標籤操作

使用下列標籤操作來新增、移除或列出資源上的標籤。

API	CLI	動作描述
TagResource	tag-resource	在具有指定 ARN 的資源上新增或覆寫一或多個標籤。
UntagResource	untag-resource	從具有指定 ARN 的資源中刪除一個或多個標籤。
ListTagsForResource	list-tags-for-resource	列出具有指定 ARN 之資源的一個或多個標籤。

### 建立資源時新增標籤

若要在建立工作群組或資料目錄時新增標籤，請將 tags 參數搭配 CreateWorkGroup 或 CreateDataCatalog API 操作或 AWS CLI create-work-group 或 create-data-catalog 命令使用。

### 使用 API 操作管理標籤

本節中的範例說明如何使用標籤 API 操作來管理工作群組和資料目錄上的標籤。這些範例使用 Java 程式語言。

#### Example TagResource

下列範例將兩個標籤新增至工作群組 workgroupA：

```
List<Tag> tags = new ArrayList<>();
tags.add(new Tag().withKey("tagKey1").withValue("tagValue1"));
tags.add(new Tag().withKey("tagKey2").withValue("tagValue2"));
```

```
TagResourceRequest request = new TagResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA")
    .withTags(tags);

client.tagResource(request);
```

下列範例將兩個標籤新增至資料目錄 datacatalogA :

```
List<Tag> tags = new ArrayList<>();
tags.add(new Tag().withKey("tagKey1").withValue("tagValue1"));
tags.add(new Tag().withKey("tagKey2").withValue("tagValue2"));

TagResourceRequest request = new TagResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA")
    .withTags(tags);

client.tagResource(request);
```

#### Note

請勿將重複的標籤鍵新增至相同的資源。如果這樣做，Athena 會發出錯誤訊息。如果您在個別的 TagResource 動作中使用現有的標籤鍵來標記資源，新標籤值會覆寫舊值。

## Example UntagResource

下列範例將 tagKey2 從工作群組 workgroupA 中移除：

```
List<String> tagKeys = new ArrayList<>();
tagKeys.add("tagKey2");

UntagResourceRequest request = new UntagResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA")
    .withTagKeys(tagKeys);

client.untagResource(request);
```

下列範例將 tagKey2 從資料目錄 datacatalogA 中移除：

```
List<String> tagKeys = new ArrayList<>();
tagKeys.add("tagKey2");
```

```
UntagResourceRequest request = new UntagResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA")
    .withTagKeys(tagKeys);

client.untagResource(request);
```

## Example ListTagsForResource

下列範例列出工作群組 workgroupA 的標籤：

```
ListTagsForResourceRequest request = new ListTagsForResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA");

ListTagsForResourceResult result = client.listTagsForResource(request);

List<Tag> resultTags = result.getTags();
```

下列範例列出資料目錄 datacatalogA 的標籤：

```
ListTagsForResourceRequest request = new ListTagsForResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA");

ListTagsForResourceResult result = client.listTagsForResource(request);

List<Tag> resultTags = result.getTags();
```

## 使用 AWS CLI 管理標籤

以下各節說明如何使用 AWS CLI 來建立和管理資料目錄上的標籤。

將標籤新增至資源：Tag-resource

tag-resource 命令將一或多個標籤新增到指定的資源。

### 語法

```
aws athena tag-resource --resource-arn
arn:aws:athena:region:account_id:datacatalog/catalog_name --tags
Key=string,Value=string Key=string,Value=string
```

--resource-arn 參數會指定要新增標籤的資源。--tags 參數會指定以空格分隔的鍵值組清單，以做為標籤新增至資源。

## Example

下列範例會將標籤新增至 mydatacatalog 資料目錄。

```
aws athena tag-resource --resource-arn arn:aws:athena:us-east-1:111122223333:datacatalog/mydatacatalog --tags Key=Color,Value=Orange
Key=Time,Value=Now
```

若要顯示結果，請使用 list-tags-for-resource 命令。

如需使用 create-data-catalog 命令時新增標籤的相關資訊，請參閱 [註冊目錄：Create-data-catalog](#)。

列出資源的標籤：List-tags-for-resource

list-tags-for-resource 命令會列出指定資源的標籤。

### 語法

```
aws athena list-tags-for-resource --resource-arn
arn:aws:athena:region:account_id:datacatalog/catalog_name
```

--resource-arn 參數會指定列出標籤的資源。

下列範例列出 mydatacatalog 資料目錄的標籤。

```
aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-east-1:111122223333:datacatalog/mydatacatalog
```

以下樣本結果為 JSON 格式。

```
{
  "Tags": [
    {
      "Key": "Time",
      "Value": "Now"
    },
    {
      "Key": "Color",
      "Value": "Orange"
    }
  ]
}
```

## 從資源中刪除標籤：Untag-resource

`untag-resource` 命令會從指定的資源中刪除指定的標籤鍵及其關聯的數值。

### 語法

```
aws athena untag-resource --resource-arn  
arn:aws:athena:region:account_id:datacatalog/catalog_name --tag-keys  
key_name [key_name ...]
```

`--resource-arn` 參數會指定要從中移除標籤的資源。`--tag-keys` 參數會取得以空格分隔的索引鍵名稱清單。對於指定的每個索引鍵名稱，`untag-resource` 命令會同時移除索引鍵及其數值。

下列範例會從 `mydatacatalog` 目錄資源中移除 `Color` 和 `Time` 索引鍵及其數值。

```
aws athena untag-resource --resource-arn arn:aws:athena:us-  
east-1:111122223333:datacatalog/mydatacatalog --tag-keys Color Time
```

## 標籤型 IAM 存取控制政策

使用標籤可讓您撰寫包含 `Condition` 區塊的 IAM 政策，以根據資源的標籤來控制對資源的存取。

### 工作群組的標籤政策範例

#### Example 1. 基本標記政策

以下 IAM 政策可讓您執行查詢，並與名為 `workgroupA` 的工作群組的標籤互動：

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "athena:ListWorkGroups",  
        "athena:ListEngineVersions",  
        "athena:ListDataCatalogs",  
        "athena:ListDatabases",  
        "athena:GetDatabase",  
        "athena:ListTableMetadata",  
        "athena:GetTableMetadata"  
      ],  
      "Resource": "*"   
    },  
  ],  
}
```

```

    {
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource",
        "athena:StartQueryExecution",
        "athena:GetQueryExecution",
        "athena:BatchGetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena:CreatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
      ],
      "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
    }
  ]
}

```

### Example 2 : 根據標籤索引鍵和標籤值配對在工作群組上拒絕動作的政策區塊

與資源 (如工作群組) 相關聯的標籤稱為資源標籤。資源標籤可讓您編寫政策區塊，如以下項目在以鍵值組所標記的任何工作群組上拒絕列出的動作，例如 `stack`、`production`。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "athena:GetWorkGroup",
        "athena:UpdateWorkGroup",

```

```

        "athena:DeleteWorkGroup",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource",
        "athena:StartQueryExecution",
        "athena:GetQueryExecution",
        "athena:BatchGetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena:CreatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
    ],
    "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/stack": "production"
        }
    }
}
]
}

```

### Example 3. 限制對指定的標籤提出標籤變更動作請求的政策區塊

做為參數傳入至變更標籤之操作的標籤 (例如帶有標籤的 TagResource、UntagResource 或 CreateWorkGroup) 稱為請求標籤。下列範例政策區塊只有在傳遞的其中一個標籤具有索引鍵 costcenter 和數值 1、2 或 3 時，才允許 CreateWorkGroup 操作。

#### Note

如果您想要允許 IAM 角色在 CreateWorkGroup 操作中傳入標籤，請確定您授予角色進行 TagResource 和 CreateWorkGroup 動作的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateWorkGroup",
        "athena:TagResource"
      ],
      "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/costcenter": [
            "1",
            "2",
            "3"
          ]
        }
      }
    }
  ]
}
```

## 資料目錄的標籤政策範例

### Example 1. 基本標記政策

以下 IAM 政策可讓您與名為 datacatalogA 的資料目錄的標籤互動：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListWorkGroups",
        "athena:ListEngineVersions",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata"
      ],
    }
  ]
}
```

```

    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "athena:GetWorkGroup",
      "athena:TagResource",
      "athena:UntagResource",
      "athena:ListTagsForResource",
      "athena:StartQueryExecution",
      "athena:GetQueryExecution",
      "athena:BatchGetQueryExecution",
      "athena:ListQueryExecutions",
      "athena:StopQueryExecution",
      "athena:GetQueryResults",
      "athena:GetQueryResultsStream",
      "athena:CreateNamedQuery",
      "athena:GetNamedQuery",
      "athena:BatchGetNamedQuery",
      "athena:ListNamedQueries",
      "athena>DeleteNamedQuery"
    ],
    "Resource": [
      "arn:aws:athena:us-east-1:123456789012:workgroup/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "athena:CreateDataCatalog",
      "athena:GetDataCatalog",
      "athena:UpdateDataCatalog",
      "athena>DeleteDataCatalog",
      "athena:ListDatabases",
      "athena:GetDatabase",
      "athena:ListTableMetadata",
      "athena:GetTableMetadata",
      "athena:TagResource",
      "athena:UntagResource",
      "athena:ListTagsForResource"
    ],
    "Resource": "arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA"
  }
}

```

```
]
}
```

### Example 2：根據標籤索引鍵和標籤值組在資料目錄上拒絕動作的政策區塊

您可以使用資源標籤撰寫政策區塊，以拒絕針對以特定標籤鍵值組標記的資料目錄上的特定動作。下列範例政策會拒絕具有標籤鍵值組 `stack`、`production` 的資料目錄上的動作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "athena:CreateDataCatalog",
        "athena:GetDataCatalog",
        "athena:UpdateDataCatalog",
        "athena>DeleteDataCatalog",
        "athena:GetDatabase",
        "athena:ListDatabases",
        "athena:GetTableMetadata",
        "athena:ListTableMetadata",
        "athena:StartQueryExecution",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource"
      ],
      "Resource": "arn:aws:athena:us-east-1:123456789012:datacatalog/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stack": "production"
        }
      }
    }
  ]
}
```

### Example 3. 限制對指定的標籤提出標籤變更動作請求的政策區塊

做為參數傳入至變更標籤之操作的標籤 (例如帶有標籤的 `TagResource`、`UntagResource` 或 `CreateDataCatalog`) 稱為請求標籤。下列範例政策區塊只有在傳遞的其中一個標籤具有索引鍵 `costcenter` 和數值 1、2 或 3 時，才允許 `CreateDataCatalog` 操作。

**Note**

如果您想要允許 IAM 角色在 CreateDataCatalog 操作中傳入標籤，請確定您授予角色進行 TagResource 和 CreateDataCatalog 動作的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateDataCatalog",
        "athena:TagResource"
      ],
      "Resource": "arn:aws:athena:us-east-1:123456789012:datacatalog/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/costcenter": [
            "1",
            "2",
            "3"
          ]
        }
      }
    }
  ]
}
```

## Service Quotas

**Note**

Service Quotas 主控台提供有關 Amazon Athena 配額的資訊。您也可以使用 Service Quotas 主控台，針對可調整的配額[請求提高配額](#)。如需 AWS Glue 相關的結構描述限制，請參閱 [AWS Glue 端點和配額](#) 頁面。如需有關 AWS 服務配額的一般資訊，請參閱 [AWSAWS 一般參考](#)。

## 查詢

您的帳戶於 Amazon Athena 有下列查詢相關的配額：如需詳細資訊，請參閱 AWS 一般參考的 [Amazon Athena 端點和配額](#) 頁面。

- Active DDL queries (作用中 DDL 查詢) - 作用中 DDL 查詢的數量。DDL 查詢包括 CREATE TABLE 和 ALTER TABLE ADD PARTITION 查詢。
- DDL query timeout (DDL 查詢逾時) - DDL 查詢在取消之前可以執行的時間上限，以分鐘為單位。
- Active DML queries (作用中 DML 查詢) - 作用中 DML 查詢的數量。DML 查詢包括 SELECT、CREATE TABLE AS (CTAS) 和 INSERT INTO 查詢。具體配額因 AWS 區域而異。
- DML query timeout (DML 查詢逾時) - DML 查詢在取消之前可以執行的時間上限，以分鐘為單位。您可以要求將此逾時提高到最多 240 分鐘。

若要請求增加配額，可使用 [Athena Service Quotas](#) 主控台。

Athena 會根據整體服務負載和傳入請求數量，以指派資源來處理查詢。系統可能會在執行您的查詢之前，先將其暫時排入佇列。非同步處理會從佇列中接收查詢，並在資源可用且帳戶設定允許的情況下，立即在實體資源上執行查詢。

DML 或 DDL 查詢配額包括執行中和佇列中的查詢。例如，如果您的 DML 查詢配額為 25，而執行中和已排入佇列的查詢總數為 26，則查詢 26 會導致錯誤。TooManyRequestsException

### Note

若您要直接控制在 Athena 中執行的查詢的並行，則可以使用容量保留。如需詳細資訊，請參閱 [管理查詢處理容量](#)。

## 查詢字串長度

允許的查詢字串長度上限是 262144 位元組，其中字串以 UTF-8 編碼。這不是可調整的配額。不過，您可以將長查詢分割成多個較小的查詢，以解決這項限制。如需詳細資訊，請參閱 AWS 知識中心中的 [如何在 Athena 中增加最大查詢字串長度？](#)。

## 工作群組

使用 Athena 工作群組時，請記住以下幾點：

- Athena Service Quotas 會在帳戶中的所有工作群組之間共用。

- 可以在帳戶中為每個區域建立的工作群組數量上限為 1000 個。
- 工作群組中預備陳述式的數量上限為 1000。
- 每一工作階段的標籤數上限為 50。如需詳細資訊，請參閱[標籤限制](#)。

## 資料庫、資料表和分割區

- 如果您 AWS Glue Data Catalog 搭配 Athena 使用，請參閱表格、資料庫和分割區的服務配額的[AWS Glue 端點和配額](#)，例如，每個帳戶的資料庫或表格數目上限。
  - 雖然 Athena 支援查詢具有 1000 萬個分割區的 AWS Glue 資料表，但是 Athena 在一次掃描中無法讀取超過 100 萬個分割區。
- 如果您未使用 AWS Glue Data Catalog，則每個表的分區數量為 20,000。您可以[要求增加配額](#)。

## Amazon S3 儲存貯體

在您使用 Amazon S3 儲存貯體時，請記住以下幾點：

- Amazon S3 的預設服務配額為每個帳戶 100 個儲存貯體。
- Athena 需有一個單獨的儲存貯體來記錄結果。
- 您可以要求每個 AWS 帳戶增加最多 1,000 個 Amazon S3 儲存貯體的配額。

## 每個帳戶 API 呼叫配額

Athena API 關於每個帳戶 (而非每個查詢) 的 API 呼叫數量的預設配額如下：

API 名稱	預設的每秒呼叫次數	高載容量
BatchGetNamedQuery , ListNamedQueries , ListQueryExecutions	5	最多 10 個
CreateNamedQuery , DeleteNamedQuery , GetNamedQuery	5	最多 20 個
BatchGetQueryExecution	20	最多 40 個
StartQueryExecution , StopQueryExecution	20	最多 80 個

API 名稱	預設的每秒呼叫次數	高載容量
GetQueryExecution , GetQueryResults	100	最多 200 個

例如，若為 StartQueryExecution，您每秒最多可以呼叫 20 次。此外，如果在 4 秒內沒有呼叫此 API，您的帳戶會累積爆發容量，最多 80 個呼叫。在這種情況下，您的應用程式可以在爆量模式下對此 API 呼叫最多 80 次。

如果您使用這些 API 中的任何一個，且超出每秒呼叫次數或您帳戶中的高載容量的預設配額，則 Athena API 會發出類似下列內容的錯誤 ClientError："`""`：呼叫作業時發生錯誤 (ThrottlingException)：超過速率。」 <API\_name> 請減少每秒呼叫次數，或此帳戶的 API 爆發容量。

無法在 Athena Service Quotas 主控台中變更每個帳戶 API 呼叫的 Athena 配額。若要請求增加 Athena API 呼叫的配額，請導覽至 AWS Support [提升服務限額](#) 頁面，然後填妥並提交表格。

## Athena 引擎版本控制

Athena 偶爾會發行新的引擎版本，以提供改進的效能、功能和程式碼修正。當有新的引擎版本可供使用時，Athena 會透過 Athena 主控台和 [AWS Health Dashboard](#) 通知您。您會 AWS Health Dashboard 通知您可能會影響您的 AWS 服務或帳戶的事件。如需 [有關的詳細資訊 AWS Health Dashboard](#)，請參閱 [AWS Health Dashboard](#)。

引擎版本控制是按 [工作群組](#) 設定。您可以使用工作群組來控制查詢使用哪個查詢引擎，以及是否讓 Athena 自動升級您的工作群組。使用中的查詢引擎會顯示在查詢編輯器的工作群組詳細資料頁面上，並可透過 Athena API 取得。

- 依預設，工作群組規劃為 auto 升級。當工作群組設定為 auto 升級時，Athena 會為您升級工作群組，除非發現不相容。
- 如果您將工作群組設定為使用指定版本，Athena 將不會變更工作群組的版本。

在這兩種情況下，當版本不再可用時，Athena 都會升級您的工作群組。Athena 通知您 [AWS Health Dashboard](#) 何時不再提供引擎版本。您會 AWS Health Dashboard 通知您可能會影響您的 AWS 服務或帳戶的事件。如需 [有關的詳細資訊 AWS Health Dashboard](#)，請參閱 [AWS Health Dashboard](#)。

當您開始使用新的引擎版本時，一小部分的查詢子集可能會因為不相容而中斷。當新的 Athena 版本發布時，將宣布重大更改。您應該透過建立使用新引擎的測試工作群組或測試升級既有工作群組，來在升級之前使用工作群組測試您的查詢。如需更多詳細資訊，請參閱 [在引擎版本升級之前測試查詢](#)。

## 主題

- [變更 Athena 引擎版本](#)
- [Athena 引擎版本參考](#)

## 變更 Athena 引擎版本

Athena 偶爾會發行新的引擎版本，以提供改進的效能、功能和程式碼修正。當有新的引擎版本可供使用時，Athena 會在主控台中通知您。您可以選擇讓 Athena 決定何時升級，或手動為每個工作群組指定 Athena 引擎版本。

## 主題

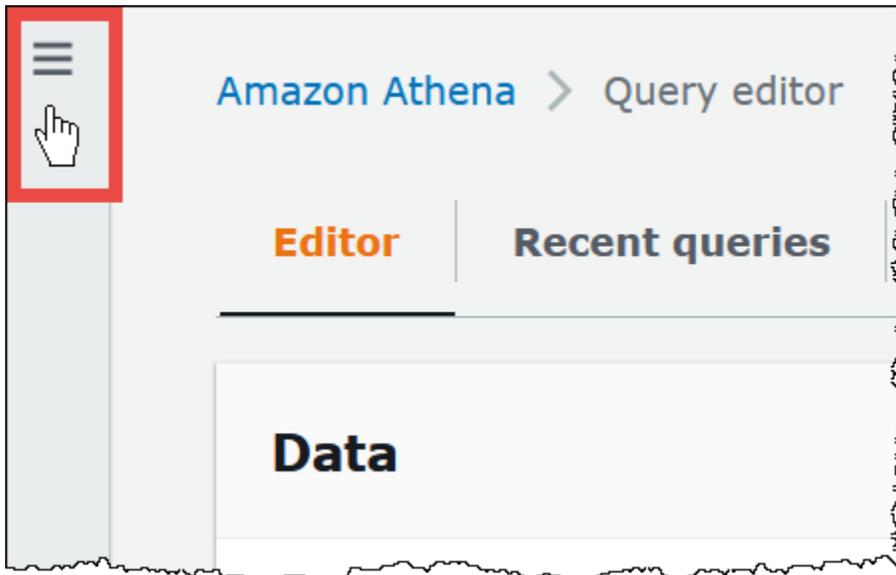
- [尋找工作群組的查詢引擎版本](#)
- [在 Athena 主控台中變更引擎版本](#)
- [使用變更引擎版本 AWS CLI](#)
- [建立工作群組時指定引擎版本](#)
- [在引擎版本升級之前測試查詢](#)
- [對失敗的查詢進行疑難排解](#)

## 尋找工作群組的查詢引擎版本

您也可以使用 Workgroups (工作群組) 頁面，來尋找任何工作群組目前使用的引擎版本。

若要尋找任意工作群組目前的引擎版本

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
4. 在 Workgroups (工作群組) 頁面，尋找您想要的工作群組。工作群組的 Query engine version (查詢引擎版本) 欄位顯示查詢引擎版本。

## 在 Athena 主控台中變更引擎版本

當有新的引擎版本可用時，您可以選擇讓 Athena 決定何時升級工作群組，或手動指定工作群組使用的 Athena 引擎版本。如果目前只有一個版本可以使用，則無法手動指定不同版本。

### Note

若要變更工作群組的引擎版本，必須擁有許可，才能對該工作群組執行 `athena:ListEngineVersions` 動作。如需 IAM 政策範例，請參閱 [工作群組政策範例](#)。

### 讓 Athena 決定何時升級工作群組

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 在主控台導覽窗格中，選擇 Workgroups (工作群組)。
4. 在工作群組清單中，選擇您想要設定的工作群組連結。
5. 選擇編輯。

6. 在 Query engine version (查詢引擎版本) 中，對於 Update query engine (更新查詢引擎)，選擇 Automatic (自動) 讓 Athena 選擇何時升級您的工作群組。這是預設設定。
7. 選擇儲存變更。

在工作群組清單中，工作群組的 Query engine update status (查詢引擎更新狀態) 會顯示為 Automatic (自動)。

### 若要手動選擇引擎版本

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 在主控台導覽窗格中，選擇 Workgroups (工作群組)。
4. 在工作群組清單中，選擇您想要設定的工作群組連結。
5. 選擇編輯。
6. 在 Query engine version (查詢引擎版本) 下，對於 Update query engine (更新查詢引擎)，選擇 Manually (手動) 即可手動選擇引擎版本。
7. 使用 Query engine version (查詢引擎版本) 選項，選擇您希望工作群組使用的引擎版本。如果不同引擎版本無法使用，則無法指定不同引擎版本。
8. 選擇儲存變更。

在工作群組清單中，工作群組的 Query engine update status (查詢引擎更新狀態) 會顯示為 Manual (手動)。

## 使用變更引擎版本 AWS CLI

若要使用變更引擎版本 AWS CLI，請使用下列範例中的語法。

```
aws athena update-work-group --work-group workgroup-name --configuration-updates  
EngineVersion={SelectedEngineVersion='Athena engine version 3'}
```

## 建立工作群組時指定引擎版本

當您建立工作群組時，您可以指定工作群組使用的引擎版本，或讓 Athena 決定何時升級工作群組。如果新的引擎版本可供使用，則最佳實務則是建立工作群組，以測試新引擎，然後再升級其他工作群組。若要指定工作群組的引擎版本，您必須擁有工作群組的 `athena:ListEngineVersions` 許可。如需 IAM 政策範例，請參閱 [工作群組政策範例](#)。

若要在建立工作群組時指定引擎版本

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 在主控台導覽窗格中，選擇 Workgroups (工作群組)。
4. 在 Workgroups (工作群組) 頁面中，請選擇 Create workgroup (建立工作群組)。
5. 在 Create workgroup (建立工作群組) 頁面中，請於 Query engine version (查詢引擎版本) 區段執行下列其中一項動作：
  - 選擇 Automatic (自動)，便可讓 Athena 選擇何時升級您的工作群組。這是預設設定。
  - 如果有不同引擎版本可供使用，透過選擇 Manual (手動)，便可手動選擇該引擎版本。
6. 視需要輸入其他欄位的資訊。如需其他欄位的相關資訊，請參閱 [建立工作群組](#)。
7. 選擇 Create workgroup (建立工作群組)。

## 在引擎版本升級之前測試查詢

當工作群組升級到新的引擎版本時，您的部分查詢可能會因為不相容而中斷。若要確保您的引擎版本升級順利進行，您可以事先測試您的查詢。

若要在引擎版本升級之前測試您的查詢

1. 請驗證您所使用的工作群組引擎版本。您正在使用的引擎版本顯示在 Workgroups (工作群組) 頁面上用於該工作群組的 Query engine version (查詢引擎版本) 欄位中。如需詳細資訊，請參閱 [尋找工作群組的查詢引擎版本](#)。
2. 建立使用新引擎版本的測試工作群組。如需詳細資訊，請參閱 [建立工作群組時指定引擎版本](#)。
3. 使用新工作群組來執行您要測試的查詢。
4. 如果查詢失敗，請使用 [Athena 引擎版本參考](#) 來檢查可能會影響查詢的重大變更。某些變更可能需要您更新查詢的語法。
5. 如果您的查詢仍然失敗，請聯絡 AWS Support 尋求協助。在 AWS Management Console 中，選擇 Support (支援)、Support Center (支援中心)，或使用 Amazon Athena 標籤在 [AWS re:Post](#) 上詢問問題。

## 對失敗的查詢進行疑難排解

如果查詢在引擎版本升級後失敗，請使用 [Athena 引擎版本參考](#) 來檢查是否有重大變更，包括可能會影響查詢語法的變更。

如果您的查詢仍然失敗，請聯絡 AWS Support 尋求協助。在中 AWS Management Console，選擇「Support」、「Support 中心」，或使用 Amazon Athena 標籤在 [AWS Re: Post](#) 上詢問問題。

## Athena 引擎版本參考

本節列出 Athena 查詢引擎的各項異動。

### 主題

- [Athena 引擎版本 3](#)
- [Athena 引擎版本 2](#)

## Athena 引擎版本 3

Athena 引擎版本 3 也仍然採用持續整合方法，管理開放原始碼軟體，提高 [Trino](#) 和 [Presto](#) 專案的並行數，讓您可以更快速地使用到各種改善後的社群功能，並在 Athena 引擎中執行整合和調整。

此次推出的 Athena 引擎版本 3 支援 Athena 引擎版本 2 的所有功能。本文件會重點說明 Athena 引擎版本 2 與 Athena 引擎版本 3 之間的主要差異。如需詳細資訊，請參閱 AWS 大數據部落格文章「[升級至 Athena 引擎版本 3，以提升查詢效能並存取更多分析功能](#)」。

- [開始使用](#)
- [改善和新功能](#)
  - [新增功能](#)
  - [新增的函數](#)
  - [效能改進](#)
  - [可靠性增強功能](#)
  - [查詢語法增強功能](#)
  - [資料格式和資料類型增強功能](#)
- [重大變更](#)
  - [查詢語法變更](#)
  - [資料處理變更](#)
  - [時間戳記變更](#)
- [限制](#)

## 開始使用

若要開始使用，請建立使用 Athena 引擎版本 3 的新 Athena 工作群組，或將現有工作群組設定為使用版本 3。任何 Athena 工作群組都可以從引擎版本 2 升級為引擎版本 3，且不會中斷提交查詢的功能。

如需詳細資訊，請參閱[變更 Athena 引擎版本](#)。

## 改善和新功能

所列之功能和更新內容包括 Athena 本身的功能改進，以及從開放原始碼 Trino 整合的功能改進。如需有關 SQL 查詢運算子和函數的詳盡清單，請參閱[Trino 文件](#)。

## 新增功能

### Apache Spark 歸納演算法支援

Athena 可讀取 Spark 雜湊演算法產生的儲存貯體。若要指定最初由 Spark 雜湊演算法撰寫該資料，請將 ('bucketing\_format'='spark') 放入 CREATE TABLE 陳述式的 TBLPROPERTIES 子句中。若未指定此屬性，則系統會使用 Hive 雜湊演算法。

```
CREATE EXTERNAL TABLE `spark_bucket_table`(  
  `id` int,  
  `name` string  
)  
CLUSTERED BY (`name`)  
INTO 8 BUCKETS  
STORED AS PARQUET  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/to/bucketed/table/'  
TBLPROPERTIES ('bucketing_format'='spark')
```

## 新增的函數

本節中的函數是 Athena 引擎版本 3 中新增的函數。

### 彙總函數

listagg(x, separator) – 傳回串連的輸入值，由分隔符號字串分開。

```
SELECT listagg(value, ',') WITHIN GROUP (ORDER BY value) csv_value  
FROM (VALUES 'a', 'c', 'b') t(value);
```

## 陣列函數

`contains_sequence(x, seq)` – 如果陣列 `x` 包含所有陣列序列，並作為循序子集 (相同連續順序中的所有值)，則會傳回 `true`。

```
SELECT contains_sequence(ARRAY [1,2,3,4,5,6], ARRAY[1,2]);
```

## 二進位函數

`murmur3 (二進制)` -計算二進製文件的 128 位 MurmurHash 3 哈希值。

```
SELECT murmur3(from_base64('aaaaaa'));
```

## 轉換函數

`format_number(number)` – 傳回使用單位符號的格式化字串。

```
SELECT format_number(123456); -- '123K'
```

```
SELECT format_number(1000000); -- '1M'
```

## 日期和時間函數

`timezone_hour(timestamp)` – 傳回與時間戳記偏離的時區小時數。

```
SELECT EXTRACT(TIMEZONE_HOUR FROM TIMESTAMP '2020-05-10 12:34:56 +08:35');
```

`timezone_minute(timestamp)` – 傳回與時間戳記偏離的時區分鐘數。

```
SELECT EXTRACT(TIMEZONE_MINUTE FROM TIMESTAMP '2020-05-10 12:34:56 +08:35');
```

## 地理空間函數

`to_encoded_polyline(Geometry)` – 將 `linestring` 或 `multipoint` 編碼為 `polyline`。

```
SELECT to_encoded_polyline(ST_GeometryFromText(
  'LINESTRING (-120.2 38.5, -120.95 40.7, -126.453 43.252)');
```

`from_encoded_polyline(varchar)` – 將 `polyline` 解碼為 `linestring`。

```
SELECT ST_AsText(from_encoded_polyline('_p~iF~ps|U_u1LnnqC_mqNvxq`@'));
```

`幾何 (SphericalGeography)` — 以 GeoJSON 格式傳回指定的球形地理位置。

```
SELECT to_geojson_geometry(to_spherical_geography(ST_GeometryFromText(
  'LINESTRING (0 0, 1 2, 3 4)')));
```

`from_geojson_geometry(varchar)` – 以 GeoJSON 表示法傳回剝離非幾何鍵/值的球形地理類型物件。  
不支援 `Feature` 和 `FeatureCollection`。

```
SELECT
  from_geojson_geometry(to_geojson_geometry(to_spherical_geography(ST_GeometryFromText(
    'LINESTRING (0 0, 1 2, 3 4)'))));
```

`geometry_nearest_points(Geometry, Geometry)` – 傳回每個幾何形狀上互相最近的點。如果任一幾何形狀為空，則會傳回 `NULL`。否則，便會傳回由兩個 `Point` 物件組成的資料列，這兩個物件與幾何形狀上任兩點的距離為最小距離。第一個點來自第一個 `Geometry` 引數，第二個點來自第二個 `Geometry` 引數。如果有多個具有相同最小距離的配對，則會任意選擇一對。

```
SELECT geometry_nearest_points(ST_GeometryFromText(
  'LINESTRING (50 100, 50 200)'), ST_GeometryFromText(
  'LINESTRING (10 10, 20 20)'));
```

**Set Digest 函數**

`make_set_digest(x)` – 將 `x` 的所有輸入值撰寫成一個 `Setdigest`。

```
SELECT make_set_digest(value) FROM (VALUES 1, 2, 3) T(value);
```

**字串函數**

`soundex(char)` – 傳回包含字元音標表示法的字元串。

```
SELECT name
FROM nation
WHERE SOUNDEX(name) = SOUNDEX('CHYNA'); -- CHINA
```

`concat_ws(string0, string1, ..., stringN)` – 傳回串連的 `string1, string2, ..., stringN` (使用 `string0` 作分隔符號)。如果 `string0` 為 `NULL`，則傳回值為 `NULL`。系統會略過在分隔符號後以引數方式提供的任何 `Null` 值。

```
SELECT concat_ws(',', 'def', 'pqr', 'mno');
```

## 範圍函數

**GROUPS** – 新增對群組型範圍框的支援。

```
SELECT array_agg(a) OVER(
  ORDER BY a ASC NULLS FIRST GROUPS BETWEEN 1 PRECEDING AND 2 FOLLOWING)
FROM (VALUES 3, 3, 3, 2, 2, 1, null, null) T(a);
```

## 效能改進

Athena 引擎版本 3 的效能改進包含以下內容。

- 更快速的 AWS Glue 表格中繼資料擷取 — 涉及多個資料表的查詢將減少查詢規劃時間。
- **RIGHT JOIN** 的動態篩選 – 現可針對具有相等聯結條件的右端聯結啟用動態篩選，如以下範例所示。

```
SELECT *
FROM lineitem RIGHT JOIN tpch.tiny.supplier
ON lineitem.suppkey = supplier.suppkey
WHERE supplier.name = 'abc';
```

- 大型預備陳述式 – 預設 HTTP 請求/回應標頭大小已增加至 2 MB，以允許使用大型預備陳述式。
- `approx_percentile()` – `approx_percentile` 函數現在使用 `tdigest` 而不是 `qdigest` 來從分佈擷取近似分位數值。這會導致更高的效能以及更低的記憶體使用量。請注意，由於此變更，函數傳回的結果與 Athena 引擎版本 2 中的結果不同。如需詳細資訊，請參閱 [approx\\_percentile 函數會傳回不同的結果](#)。

## 可靠性增強功能

Athena 引擎版本 3 的一般引擎記憶體用量和追蹤功能已改進。大型查詢較不易因節點當機而失敗。

## 查詢語法增強功能

**INTERSECT ALL** – 已新增對 **INTERSECT ALL** 的支援。

```
SELECT * FROM (VALUES 1, 2, 3, 4) INTERSECT ALL SELECT * FROM (VALUES 3, 4);
```

EXCEPT ALL – 已新增對 EXCEPT ALL 的支援。

```
SELECT * FROM (VALUES 1, 2, 3, 4) EXCEPT ALL SELECT * FROM (VALUES 3, 4);
```

RANGE PRECEDING – 已新增對範圍函數中 RANGE PRECEDING 的支援。

```
SELECT sum(x) over (order by x range 1 preceding)
FROM (values (1), (1), (2), (2)) t(x);
```

MATCH\_RECOGNIZE – 已新增對資料列模式比對的支援，如以下範例所示。

```
SELECT m.id AS row_id, m.match, m.val, m.label
FROM (VALUES(1, 90),(2, 80),(3, 70),(4, 70)) t(id, value)
MATCH_RECOGNIZE (
  ORDER BY id
  MEASURES match_number() AS match,
  RUNNING LAST(value) AS val,
  classifier() AS label
  ALL ROWS PER MATCH
  AFTER MATCH SKIP PAST LAST ROW
  PATTERN (() | A) DEFINE A AS true
) AS m;
```

## 資料格式和資料類型增強功能

Athena 引擎版本 3 具有以下資料格式和資料類型增強功能。

- LZ4 和 ZSTD – 已新增對讀取 LZ4 和 ZSTD 壓縮 Parquet 資料的支援。已新增對寫入 ZSTD 壓縮 ORC 資料的支援。
- 符號連結型資料表 – 已新增對在 Avro 檔案上建立符號連結型資料表的支援。範例如下。

```
CREATE TABLE test_avro_symlink
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
...
INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
```

- SphericalGeography— 此 SphericalGeography 類型為地理座標上表示的空間圖徵提供原生支援 (有時稱為大地座標、緯度/經度或 LON/lat)。地理座標是以角度單位 (度) 來表示的球面座標。

`to_spherical_geography` 函數會從幾何 (平面) 座標傳回地理 (球形) 座標，如以下範例所示。

```
SELECT to_spherical_geography(ST_GeometryFromText(
  'LINESTRING (-40.2 28.9, -40.2 31.9, -37.2 31.9)');
```

## 重大變更

從 Athena 引擎版本 2 遷移至 Athena 引擎版本 3 時，特定變更可能會影響資料表結構描述、語法或資料類型用量。本節會列出相關的錯誤訊息，並提供建議的因應措施。

### 查詢語法變更

IGNORE NULLS 不能與非值視窗函數一起使用

錯誤訊息：無法為 `bool_or` 函數指定 Null 處理方式子句。

原因：IGNORE NULLS 現在只能與[值函數](#) `first_value`、`last_value`、`nth_value`、`lead` 和 `lag` 一起使用。此變更之目的是符合 ANSI SQL 規格。

建議的解決方案：從查詢字符串中的非值視窗函數中移除 IGNORE NULLS。

CONCAT 函數必須有兩個或多個引數

錯誤信息：INVALID\_FUNCTION\_ARGUMENT: There must be two or more concatenation arguments (INVALID\_FUNCTION\_ARGUMENT：必須有兩個或多個串聯參數)

原因：先前，CONCAT 字串函數接受了單一引數。在 Athena 引擎版本 3 中，CONCAT 函數需要至少兩個引數。

建議的解決方案：將出現次數從 `CONCAT(str)` 變更為 `CONCAT(str, '')`。

在 Athena 引擎版本 3 中，函數中不可超過 127 個引數。如需詳細資訊，請參閱 [函數呼叫的引數過多](#)。

`approx_percentile` 函數會傳回不同的結果

`approx_percentile` 函數會在 Athena 引擎版本 3 中傳回的結果與 Athena 引擎版本 2 中傳回的結果不同。

錯誤訊息：無。

原因：`approx_percentile` 函數可能因版本變更而變更。

**⚠ Important**

由於 `approx_percentile` 函數的輸出是近似值，並且近似值可能會從一個版本變更為下一個版本，因此關鍵應用程式不得倚賴 `approx_percentile` 函數。

建議的解決方案：若要將 Athena 引擎版本 2 行為估算為 `approx_percentile`，您可以在 Athena 引擎版本 3 中使用不同的函數組。例如，假設您在 Athena 引擎版本 2 中有下列查詢：

```
SELECT approx_percentile(somecol, 2E-1)
```

若要估計 Athena 引擎版本 3 中的相同輸出，您可以嘗試使用 `qdigest_agg` 和 `value_at_quantile` 函數，如下列範例所示。請注意，即使使用此解決方法，也不能保證行為相同。

```
SELECT value_at_quantile(qdigest_agg(somecol, 1), 2E-1)
```

地理空間函數不支援 `varbinary` 輸入

錯誤訊息：FUNCTION\_NOT\_FOUND for st\_XXX

原因：一些地理空間函數不再支援舊版 `VARBINARY` 輸入類型或與文字相關的函數簽章。

建議的解決方案：使用地理空間函數將輸入類型轉換為受支援的類型。錯誤訊息會指出受支援的輸入類型。

在 `GROUP BY` 子句中，巢狀資料欄必須使用雙引號

錯誤訊息：`"column_name"."nested_column"` 必須是彙總表達式，或出現在 `GROUP BY` 子句中

原因：Athena 引擎版本 3 要求 `GROUP BY` 子句中的巢狀資料欄名稱必須加上雙引號。例如，下列查詢會產生錯誤，因為在 `GROUP BY` 子句中，`user.name` 未使用雙引號。

```
SELECT "user"."name" FROM dataset
GROUP BY user.name
```

建議的解決方案：在 `GROUP BY` 子句中的巢狀資料欄名稱周圍使用雙引號，如下列範例所示。

```
SELECT "user"."name" FROM dataset
```

```
GROUP BY "user"."name"
```

在冰山表上使用 OPTIMIZE 時出 FilterNode 現意外錯誤

錯誤訊息：在計劃中 FilterNode發現未預期；可能連接器無法處理提供的 WHERE 運算式。

原因：在 Iceberg 表上運行的OPTIMIZE語WHERE句使用了在其過濾器表達式中包含非分區列的子句。

建議的解決方案：OPTIMIZE陳述式僅支援按分割區進行篩選。當您在分區資料表OPTIMIZE上執行時，在WHERE子句中只包含資料分割資料行。如果您OPTIMIZE在非資料分割資料表上執行，請勿指定子WHERE句。

Log() 函數引數的順序

在 Athena 引擎第 2 版中，log() 函數的引數順序為 log(*value*, *base*)。在 Athena 引擎第 3 版中，此順序已變更為 log(*base*, *value*)，以符合 SQL 標準。

Minute() 函數不支援 interval year to month

錯誤訊息：Unexpected parameters (interval year to month) for function minute.(Minute 函數的意外參數 (間隔使用年份和月份)。) Expected: minute(timestamp with time zone), minute(time with time zone), minute(timestamp), minute(time), minute(interval day to second). (函數 minute 出現未預期參數 (interval year to month)。預期：minute(timestamp with time zone)、minute(time with time zone)、minute(timestamp)、minute(time)、minute(interval day to second)。)

原因：在 Athena 引擎版本 3 中，已依據 ANSI SQL 規格，更精確地為 EXTRACT 執行類型檢查。

建議的解決方案：更新查詢，以確保類型與建議的函數簽章相符。

ORDER BY 表達式必須出現在 SELECT 清單中。

錯誤訊息：For SELECT DISTINCT, ORDER BY expressions must appear in SELECT list (若為 SELECT DISTINCT，ORDER BY 表達式必須出現在 SELECT 清單中)

原因：SELECT 子句中使用了不正確的資料表別名方式。

建議的解決方案：再次檢查 ORDER BY 表達式中的所有資料欄是否在 SELECT DISTINCT 子句中皆具有正確參考。

比較從子查詢傳回的多個資料欄時，查詢失敗

範例錯誤訊息：值表達式和子查詢的結果必須是相同的類型：資料列(varchar, varchar) 與資料列(row(varchar, varchar))

原因：由於 Athena 引擎版本 3 中的語法更新，當查詢嘗試比較子查詢傳回的多個值，而子查詢 SELECT 陳述式會包括其資料欄清單 (用括號) 時，就會發生此錯誤，如下列範例所示。

```
SELECT *
FROM table1
WHERE (t1_col1, t1_col2)
IN (SELECT (t2_col1, t2_col2) FROM table2)
```

解決方案：在 Athena 引擎版本 3 中，移除子查詢 SELECT 陳述式中資料欄清單周圍的括號，如下列更新的範例查詢所示。

```
SELECT *
FROM table1
WHERE (t1_col1, t1_col2)
IN (SELECT t2_col1, t2_col2 FROM table2)
```

SKIP 是 DML 查詢的保留字詞

單字 SKIP 是 DML 查詢的保留字詞，例如 SELECT。若要在 DML 查詢中將 SKIP 用作識別符，請將其括在雙引號中。

如需有關 Athena 中保留字詞的詳細資訊，請參閱 [保留的關鍵字](#)。

時間歷程已取代 SYSTEM\_TIME 和 SYSTEM\_VERSION 子句。

錯誤訊息：mismatched input 'SYSTEM\_TIME'. (不相符的輸入 'SYSTEM\_TIME'。) 預期：'TIMESTAMP'、'VERSION'

原因：在 Athena 引擎版本 2 中，Iceberg 資料表使用了 FOR SYSTEM\_TIME AS OF 和 FOR SYSTEM\_VERSION AS OF 子句作為時間戳記和版本時間歷程。Athena 引擎版本 3 使用了 FOR TIMESTAMP AS OF 和 FOR VERSION AS OF 子句。

建議的解決方案：更新 SQL 查詢，以使用 TIMESTAMP AS OF 和 VERSION AS OF 子句作為時間歷程操作，如下列範例所示。

依時間戳記的時間歷程：

```
SELECT * FROM TABLE FOR TIMESTAMP AS OF (current_timestamp - interval '1' day)
```

依版本的時間歷程：

```
SELECT * FROM TABLE FOR VERSION AS OF 949530903748831860
```

### 陣列建構的引數過多

錯誤訊息：TOO\_MANY\_ARGUMENTS：陣列建構的引數過多。

原因：陣列建構中的元素數目上限現已設定為 254。

建議的解決方案：將元素分解為多個陣列，每個陣列具有 254 個或更少的元素，並使用 CONCAT 函數來串連陣列，如下列範例所示。

```
CONCAT(  
  ARRAY[x1,x2,x3...x254],  
  ARRAY[y1,y2,y3...y254],  
  ...  
)
```

### 不允許使用零長度分隔的識別碼

錯誤訊息：Zero-length delimited identifier not allowed. (不允許使用零長度分隔的識別碼。)

原因：查詢使用了空白字串作為資料欄別名。

建議的解決方案：更新查詢以使用非空白的資料欄別名。

### 資料處理變更

#### 儲存貯體驗證

錯誤訊息：HIVE\_INVALID\_BUCKET\_FILES：Hive 資料表已損壞。

原因：該資料表可能已損壞。為確定歸納資料表的查詢正確性，Athena 引擎版本 3 會在歸納資料上啟用額外的驗證，以確保查詢的正確性並避免執行期發生非預期的失敗。

建議的解決方案：使用 Athena 引擎版本 3 重新建立資料表。

現在，將結構轉換為 JSON，會傳回欄位名稱

當您在 Athena 引擎第 3 版的 SELECT 查詢中將 struct 轉換為 JSON 時，該轉換現在會傳回欄位名稱和值 (例如 "useragent":null) 而非只是值 (例如，null)。

#### Iceberg 資料表資料欄層級安全性強制執行變更

錯誤信息：Access Denied: Cannot select from columns (存取遭拒：無法從資料欄中選取)

原因：Iceberg 資料表是在 Athena 之外建立的，並使用早於 0.13.0 的 [Apache Iceberg SDK](#) 版本。由於舊版 SDK 不會在中填入資料行 AWS Glue，因此 Lake Formation 無法判斷授權存取的資料行。

建議的解決方案：使用 Athena [ALTER TABLE SET PROPERTIES](#) 陳述式執行更新，或使用最新的 Iceberg SDK 來修正資料表並更新 AWS Glue 中的資料欄資訊。

List 資料類型中的 Null 值現已傳播至 UDF

錯誤訊息：Null Pointer Exception (Null 指標異常情形)

原因：如果您使用 UDF 連接器並已執行使用者定義的 Lambda 函數，則此問題可能會對您產生影響。

Athena 引擎版本 2 會篩選出 List 資料類型中已傳遞給使用者定義函數的 Null 值。在 Athena 引擎版本 3 中，會保留 Null 值並傳遞給 UDF。如果 UDF 嘗試在未檢查的情況下取消參考 Null 元素，這可能會導致 Null 指標異常情形。

例如，若您在如 DynamoDB 等原始資料來源中具有資料 [null, 1, null, 2, 3, 4]，則系統會將以下項目傳遞至使用者定義的 Lambda 函數：

Athena 引擎版本 2：[1,2,3,4]

Athena 引擎版本 3：[null, 1, null, 2, 3, 4]

建議的解決方案：確保使用者定義的 Lambda 函數會處理 List 資料類型中的 Null 元素。

字元陣列中的子字串不再包含已填補的空格

錯誤訊息：No error is thrown, but the string returned no longer contains padded spaces. (未擲回任何錯誤，但傳回的字串不再包含已填補的空格。) 例如，`substr(char[20],1,100)` 現在會傳回長度為 20 而非 100 的字串。

建議的解決方案：無需採取任何動作。

不支援的十進位資料欄類型強制轉換

錯誤消息 `#HIVE_CURSOR_ #####s3://DOC-EXAMPLE-BUCKET/ ##/###. #####  
#####varchar#####[##]`

原因：當嘗試將資料類型強制從 `varchar` 轉換為 `decimal` 時，Athena 引擎版本 2 偶有成功 (但經常失敗)。由於 Athena 引擎版本 3 會有類型驗證，即會在嘗試讀取值之前先檢查類型是否相容，因此類型嘗試強制執行現在總會失敗。

建議的解決方案：對於 Athena 引擎版本 2 和 Athena 引擎版本 3，請修改中的結構描述，AWS Glue 以使用數值資料類型，而不是 Parquet 檔案中的十進位資料欄。varchar 重新編目資料並確定新的資料欄資料類型為 decimal 類型，或在 Athena 中手動重新建立資料表，然後使用語法 decimal(*precision*, *scale*) 指定資料欄的 [decimal](#) 資料類型。

Float 或 double NaN 值不能再轉換為 bigint

錯誤訊息：INVALID\_CAST\_ARGUMENT：無法將 real/double NaN 轉換為 bigint

原因：在 Athena 引擎版本 3 中，NaN 無法再以 bigint 的形式轉換為 0。

建議的解決方案：當您轉換為 bigint 時，請確保 NaN 值不存在於 float 或 double 資料欄。

uuid() 函數傳回類型變更

下列問題會影響資料表和檢視。

錯誤訊息：不支援的 Hive 類型：uuid

原因：在 Athena 引擎版本 2 中，uuid() 函數傳回了一個字串，但在 Athena 引擎版本 3 中，它會傳回一個偽隨機產生的 UUID (類型 4)。由於 Athena 中不支援 UUID 資料欄資料類型，因此無法在 CTAS 查詢中直接使用 uuid() 函數，以在 Athena 引擎版本 3 中產生 UUID 資料欄。

例如，下列 CREATE TABLE 陳述式在 Athena 引擎版本 2 中成功完成；但在 Athena 引擎版本 3 中，則會傳回 NOT\_SUPPORTED: 不支援的 Hive 類型：uuid：

```
CREATE TABLE uuid_table AS
SELECT uuid() AS myuuid
```

同樣地，下列 CREATE VIEW 陳述式在 Athena 引擎版本 2 中成功完成；但在 Athena 引擎版本 3 中，則會傳回 資料欄 myuuid 的資料欄類型無效：不支援的 Hive 類型：uuid：

```
CREATE VIEW uuid_view AS
SELECT uuid() AS myuuid
```

在 Athena 引擎版本 3 中查詢在 Athena 引擎版本 2 中建立的檢視時，會發生類似下列情況的錯誤：

VIEW\_IS\_STALE：行 1:15:：檢視 'awsdatacatalog.mydatabase.uuid\_view' 已過時或處於無效狀態：從位置 0 查詢檢視投影之 uuid 類型的資料欄 [myuuid] 無法強制轉換為儲存在檢視定義中的 varchar 類型的資料欄 [myuuid]

建議的解決方案：建立資料表或檢視時，請使用 `cast()` 函數將 `uuid()` 的輸出轉換為 `varchar`，如下列範例所示：

```
CREATE TABLE uuid_table AS
SELECT CAST(uuid() AS VARCHAR) AS myuuid
```

```
CREATE VIEW uuid_view AS
SELECT CAST(uuid() AS VARCHAR) AS myuuid
```

## CHAR 和 VARCHAR 強制問題

如果您在 Athena 引擎版本 3 中遇到 `varchar` 和 `char` 強制問題，請使用本節的解決方法。如果您無法使用這些因應措施，請聯絡 [AWS Support](#)

### 混合 CHAR 和 VARCHAR 輸入的 CONCAT 函數失敗

問題：Athena 引擎版本 2 會成功執行下列查詢。

```
SELECT concat(CAST('abc' AS VARCHAR(20)), '12', CAST('a' AS CHAR(1)))
```

但是，在 Athena 引擎版本 3 上，相同的查詢會失敗，並顯示下列情況：

錯誤訊息：FUNCTION\_NOT\_FOUND：行 1:8：函數 concat 的意外參數 (varchar(20), varchar(2), char(1))。預期：concat(char(x), char(y)), concat(array(E), E) E、concat(E, array(E)) E、concat(array(E)) E、concat(varchar)、concat(varbinary)

建議的解決方案：使用 `concat` 函數時 `varchar`，可轉換為 `char` 或，但不能混合兩者。

### SQL || CHAR 和 VARCHAR 輸入的串聯失敗

在 Athena 引擎版本 3 中，雙垂直長條圖 `||` 串聯操作員要求 `varchar` 做為輸入。輸入不能是 `varchar` 和 `char` 類型的組合。

錯誤訊息：TYPE\_NOT\_FOUND：行 1:26：未知類型：char(65537)

原因：使用 `||` 來串連 `char` 和 `varchar` 的查詢可能會產生錯誤，如下列範例所示。

```
SELECT CAST('a' AS CHAR) || CAST('b' AS VARCHAR)
```

建議的解決方案：串連 `varchar` 和 `varchar`，如下列範例所示。

```
SELECT CAST('a' AS VARCHAR) || CAST('b' AS VARCHAR)
```

## CHAR 和 VARCHAR UNION 查詢失敗

錯誤訊息：NOT\_SUPPORTED：不支援的 Hive 類型：char(65536)。支援的 CHAR 類型：CHAR(<=255)

原因：嘗試合併 char 和 varchar 的查詢，如下列範例所示：

```
CREATE TABLE t1 (c1) AS SELECT CAST('a' as CHAR) as c1 UNION ALL SELECT CAST('b' AS VARCHAR) AS c1
```

建議的解決方案：在範例查詢中，將 'a' 轉換為 varchar 而不是 char。

## CHAR 或 VARCHAR 強制後不需要的空白空格

在 Athena 引擎版本 3 中，當 char(X) 和 varchar 資料在形成陣列或單一資料欄時強制轉換為單一類型，則 char(65535) 是目標類型，而且每個欄位都包含許多不需要的尾隨空格。

原因：Athena 引擎版本 3 強制將 varchar 和 char(X) 轉換為 char(65535)，然後右邊填補了空格的資料。

建議的解決方案：明確地將每個欄位轉換為 varchar。

## 時間戳記變更

將 Timestamp with Time Zone 轉換為 varchar 的行為變更

在 Athena 引擎版本 2 中，將 Timestamp with time zone 轉換為 varchar 時，會導致部分時區常值變更 (例如，US/Eastern 變更為 America/New\_York)。Athena 引擎版本 3 不會發生此行為。

## 日期時間戳記溢出會擲出錯誤

錯誤訊息：Millis overflow: XXX (毫秒溢出：XXX)

原因：由於在 Athena 引擎版本 2 中未檢查 ISO 8601 日期是否溢出，因此部分日期會產生負數時間戳記。Athena 引擎版本 3 會檢查此溢出，並擲出例外狀況。

建議的解決方案：確保時間戳記在範圍內。

## 不支援具有 TIME 的政治時區

錯誤訊息：INVALID LITERAL (無效常值)

原因：如 `SELECT TIME '13:21:32.424 America/Los_Angeles'` 等查詢。

建議的解決方案：避免使用具有 TIME 的政治時區。

Timestamp 資料欄中的精確度不符會造成序列化錯誤

錯誤訊息：SERIALIZATION\_ERROR: Could not serialize column '*COLUMNZ*' of type 'timestamp(3)' at position *X:Y* (SERIALIZATION\_ERROR：無法在位置 *X:Y* 處序列化類型為 'timestamp(3)' 的資料欄 '*COLUMNZ*')

*COLUMNZ* 是造成該問題之資料欄的輸出名稱。數字 *X:Y* 表示資料欄在輸出中的位置。

原因：Athena 引擎版本 3 會執行檢查，以確保資料中時間戳記的精確度與資料表規格中資料欄資料類型指定的精確度相同。目前，此精確度一直是 3。如果資料的精確度大於此值，則查詢會失敗，且系統會指出錯誤。

建議的解決方案：檢查您的資料，以確保您的時間戳記為毫秒精確度。

Iceberg 表的 UNLOAD 和 CTAS 查詢中的時間戳記精度不正確

錯誤訊息：時間戳記 (6) 的時間戳記精確度不正確；設定的精確度為毫秒

原因：Athena 引擎版本 3 會執行檢查，以確保資料中時間戳記的精確度與資料表規格中資料欄資料類型指定的精確度相同。目前，此精確度一直是 3。如果資料的精確度大於此值 (例如以微秒而非毫秒)，則查詢會失敗且系統會指出錯誤。

解決方案：若要解決此問題，請先將時間戳記精確度 CAST 為 6，如下列建立 Iceberg 資料表的 CTAS 範例所示。請注意，必須將精確度指定為 6 而不是 3，以避免錯誤：Iceberg 不支援時間戳記精度 (3)。

```
CREATE TABLE my_iceberg_ctas
WITH (table_type = 'ICEBERG', location = 's3://DOC-EXAMPLE-BUCKET/table_ctas/',
format = 'PARQUET')
AS SELECT id, CAST(dt AS timestamp(6)) AS "dt"
FROM my_iceberg
```

然後，由於 Athena 不支援時間戳記 6，因此請將值再次轉換為時間戳記 (例如，在檢視中)。下列範例從 `my_iceberg_ctas` 資料表建立檢視。

```
CREATE OR REPLACE VIEW my_iceberg_ctas_view AS
SELECT cast(dt AS timestamp) AS dt
FROM my_iceberg_ctas
```

現在，將 ORC 檔案中的 Long 類型讀取為 Timestamp 會造成不正確的 ORC 檔案錯誤，反之亦然

錯誤訊息：Error opening Hive split 'FILE (SPLIT POSITION)' Malformed ORC file. (開啟 Hive 分割 'FILE (SPLIT POSITION)' 格式不正確的 ORC 檔案時發生錯誤。) Cannot read SQL type timestamp from ORC stream .long\_type of type LONG (開啟 Hive 分割 'FILE (SPLIT POSITION)' 不正確的 ORC 檔案時發生錯誤。無法從 LONG 類型的 ORC 串流 .long\_type 讀取 SQL 類型的時間戳記)

原因：Athena 引擎版本 3 現在會拒絕從 Long 資料類型隱含強制轉換至 Timestamp，或從 Timestamp 轉換至 Long。過去，若 Long 值為 epoch 毫秒，便會被隱含強制轉換為時間戳記。

建議的解決方案：使用 `from_unixtime` 函數來明確轉換資料欄，或者使用 `from_unixtime` 函數為以後的查詢建立一個額外的資料欄。

不支援 time 和 interval year to month

錯誤訊息：TYPE MISMATCH (類型不相符)

原因：Athena 引擎版本 3 不支援 time 和 interval year to month (例如 `SELECT TIME '01:00' + INTERVAL '3' MONTH`)。

Int96 Parquet 格式的時間戳記溢出

錯誤信息：無效的 timeOfDay 納諾斯

原因：int96 Parquet 格式的時間戳記溢出。

建議的解決方案：辨識發生問題的特定檔案。然後使用知名的實木複合地板資料庫再次產生資料檔案，或使用 Athena CTAS。up-to-date 如果問題仍持續發生，請聯絡 Athena 支援，並告知我們產生資料檔案的方法。

從字串轉換為時間戳記時，日期和時間值之間需要的空格

錯誤訊息：INVALID\_CAST\_ARGUMENT：值無法轉換為時間戳記。

原因：Athena 引擎版本 3 不再接受連字號做為輸入字串中日期和時間值之間的有效分隔符。cast 例如，下列查詢適用於 Athena 引擎版本 2，但不適用於 Athena 引擎版本 3：

```
SELECT CAST('2021-06-06-23:38:46' AS timestamp) AS this_time
```

建議的解決方案：在 Athena 引擎版本 3 中，將日期與時間之間的連字號取代為空格，如下列範例所示。

```
SELECT CAST('2021-06-06 23:38:46' AS timestamp) AS this_time
```

## to\_iso8601() 時間戳記傳回值變更

錯誤訊息：無

原因：在 Athena 引擎版本 2 中，即使傳遞給 to\_iso8601 函數的值不包含時區，該函數也會傳回帶有時區的時間戳記。在 Athena 引擎版本 3 中，只有當傳遞的引數包含時區時，to\_iso8601 函數才會傳回時間戳記與時區。

例如，下列查詢會將目前日期傳遞給 to\_iso8601 函數兩次：首先是做為具有時區的時間戳記，然後是做為時間戳記。

```
SELECT TO_ISO8601(CAST(CURRENT_DATE AS TIMESTAMP WITH TIME ZONE)),
       TO_ISO8601(CAST(CURRENT_DATE AS TIMESTAMP))
```

下列輸出會顯示每個引擎中的查詢結果。

Athena 引擎版本 2：

#	_col0	_col1
1	2023-02-24T00:00:00.000Z	2023-02-24T00:00:00.000Z

Athena 引擎版本 3：

#	_col0	_col1
1	2023-02-24T00:00:00.000Z	2023-02-24T00:00:00.000

建議的解決方案：要複製先前的行為，您可以先將時間戳記值傳遞至 with\_timezone 函數，然後再將其傳遞給 to\_iso8601，如下列範例所示：

```
SELECT to_iso8601(with_timezone(TIMESTAMP '2023-01-01 00:00:00.000', 'UTC'))
```

結果

#	_col0
1	2023-01-01T00:00:00.000Z

## at\_timezone() 第一個參數必須指定一個日期

問題：在 Athena 引擎版本 3 中，at\_timezone 函數無法將 time\_with\_timezone 值做為第一個參數。

原因：如果沒有日期資訊，則無法判斷傳遞的值是夏令時間還是標準時間。例

如，at\_timezone('12:00:00 UTC', 'America/Los\_Angeles') 並不明確，因為無法判斷傳遞的值是太平洋夏令時間 (PDT) 還是太平洋標準時間 (PST)。

## 限制

Athena 引擎版本 3 有以下限制。

- 查詢效能 – 許多查詢在 Athena 引擎版本 3 上的執行速度更快，但部分查詢計畫可能與 Athena 引擎版本 2 不同。因此，部分查詢的延遲或成本可能有所不同。
- Trino 和 Presto 連接器 – 不支援 [Trino](#) 和 [Presto](#) 連接器。使用 Amazon Athena 聯合查詢來連接資料來源。如需詳細資訊，請參閱 [使用 Amazon Athena 聯合查詢](#)。
- 容錯執行 – 不支援 Trino [容錯執行](#) (Trino Tardigrade)。
- 函數參數限制 – 函數使用的參數不能超過 127 個。如需詳細資訊，請參閱 [函數呼叫的引數過多](#)。

Athena 引擎版本 2 中引入了下列限制，以確保查詢不會因資源限制而失敗。使用者無法設定這些限制。

- 結果元素的數量 – 下列函數的結果元素 n 的數量限制為 10,000 或更少：min(col, n)、max(col, n)、min\_by(col1, col2, n) 以及 max\_by(col1, col2, n)。
- GROUPING SETS – 分組集中的配量數目上限為 2048。
- 文字檔案一行長度上限 – 文字檔案預設的一行長度上限為 200 MB。
- 序列函數結果大小上限 – 序列函數的結果大小上限為 50,000 個項目。例如：SELECT sequence(0,45000,1) 成功，但 SELECT sequence(0,55000,1) 失敗並顯示錯誤訊息 The result of the sequence function must not have more than 50000 entries (序列函數的結果不能有 50,000 個以上項目)。此限制適用於序列函數 (包括時間戳記) 的所有輸入類型。

## Athena 引擎版本 2

Athena 引擎版本 2 推出下列變更。

- [改善和新功能](#)

- [分組、聯結及子查詢改善](#)
- [資料類型增強功能](#)
- [新增的函數](#)
- [效能改進](#)
- [JSON 相關的改善](#)
- [重大變更](#)
  - [錯誤修正](#)
  - [對空間函數的變更](#)
  - [ANSI SQL 合規](#)
  - [替換的函數](#)
  - [限制](#)

## 改善和新功能

- EXPLAIN 和 EXPLAIN ANALYZE – 您現在可以在 Athena 中使用 EXPLAIN 陳述式來檢視 SQL 查詢的執行計劃。使用 EXPLAIN ANALYZE 來檢視 SQL 查詢的分散式執行計劃，以及每項操作的成本。如需詳細資訊，請參閱 [在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)。
- 聯合查詢 – Athena 引擎版本 2 支援聯合查詢。如需詳細資訊，請參閱 [使用 Amazon Athena 聯合查詢](#)。
- 地理空間函數 – 新增超過 25 個地理空間函數。如需詳細資訊，請參閱 [Athena 引擎版本 2 中的新地理空間函數](#)。
- 巢狀結構描述 – 已新增對讀取巢狀結構描述的支援，可降低成本。
- 預備陳述式 – 使用預備陳述式來重複執行具有不同查詢參數的相同查詢。預備陳述式包含您在執行時間傳遞其值的預留位置參數。預備陳述式有助於防止 SQL injection 攻擊。如需詳細資訊，請參閱 [使用參數化查詢](#)。
- 結構描述演變支援 – 已新增 Prquet 格式資料的結構描述演進支援。
  - 新增了對讀取陣列、映射或分割區列類型資料欄的支援，其中分割區結構描述與資料表結構描述不同。當資料表結構描述在建立分割區之後更新時，就會發生這種情況。已變更資料欄類型必須相容。對於列類型，可以新增或捨棄結尾欄位，但對應欄位 (按序數) 必須具有相同的名稱。
  - ORC 檔案現在可以有遺漏欄位的結構資料欄。這允許在不重寫 ORC 檔案的情況下變更資料表結構描述。
  - ORC 結構資料欄現在按名稱映射，而不是按序數映射。這樣可正確處理 ORC 檔案中缺少或額外的結構欄位。

- SQL 位移 – SELECT 陳述式現已支援 SQL OFFSET 子句。如需詳細資訊，請參閱 [SELECT](#)。
- UNLOAD 陳述式 – 您可以將 UNLOAD 陳述式用於 SELECT 查詢的輸出，便可寫入 PARQUET、ORC、AVRO 和 JSON 格式。如需詳細資訊，請參閱 [UNLOAD](#)。

### 分組、聯結及子查詢改善

- 複雜分組 – 新增了對複雜分組操作的支援。
- 相互關聯的子查詢 – 新增了對 IN 述詞中相互關聯子查詢以及對需要強制的相互關聯子查詢的支援。
- CROSS JOIN – 新增了對依據 LATERAL 衍生資料表執行 CROSS JOIN 的支援。
- GROUPING SETS – 在使用 GROUPING SETS 的查詢彙總中新增了對 ORDER BY 子句的支援。
- Lambda 表達式 – 在 Lambda 表達式中新增了取消列欄位參考的支援。
- 半聯結中的 Null 值 – 在半聯結的左側新增了對 Null 值的支援 (也就是有子查詢的 IN 述詞)。
- 空間聯結 – 新增了對廣播空間聯結和空間左聯結的支援。
- 溢寫至磁碟 – 對於佔用大量記憶體 INNER JOIN 和 LEFT JOIN 操作，Athena 會卸載中繼操作結果至磁碟。這會啟用需要大量記憶體的查詢執行。

### 資料類型增強功能

- INTEGER 的 INT – 新增了對 INT 的支援，作為 INTEGER 資料類型的別名。
- INTERVAL 類型 – 新增了對轉換為 INTERVAL 類型的支援。
- IPADDRESS — 已新增新 IPADDRESS 類型，以代表 DML 查詢中的 IP 位址。新增了對 VARBINARY 類型和 IPADDRESS 類型之間轉換的支援。在 DDL 查詢中無法識別該 IPADDRESS 類型。
- IS DISTINCT FROM – 新增了對 JSON 和 IPADDRESS 類型的 IS DISTINCT FROM 支援。
- Null 相等檢查 – 現在支援對 ARRAY、MAP 和 ROW 資料結構中的 null 值進行相等檢查。例如，表達式 `ARRAY ['1', '3', null] = ARRAY ['1', '2', null]` 傳回 false。先前，null 元素會傳回錯誤訊息 comparison not supported (不支援比較)。
- 列類型強制 – 不論欄位名稱為何，現在都允許在列類型之間強制執行。先前，只有在來源類型中的欄位名稱符合目標類型，或目標類型具有匿名欄位名稱時，才會強制列類型為另一個列類型。
- 時間減法 – 為所有 TIME 和 TIMESTAMP 類型實作了減法。
- Unicode – 對字串常值中逸出的 Unicode 序列新增了支援。
- VARBINARY 串連 – 新增了對 VARBINARY 值串聯的支援。

視窗值函數 – 視窗值函數現在支援 IGNORE NULLS 和 RESPECT NULLS。

## 函數的其他輸入類型

下列函數現在可接受其他輸入類型。如需每個函數的詳細資訊，請造訪 Presto 文件的對應連結。

- `approx_distinct()` – [approx\\_distinct\(\)](#) 函數現在支援以下類型：INTEGER、SMALLINT、TINYINT、DECIMAL、REAL、DATE、TIMESTAMP、TIMESTAMP WITH TIME ZONE、TIME、TIME WITH TIME ZONE、IPADDRESS 以及 CHAR。
- `avg()`, `sum()` – [avg\(\)](#) 和 [sum\(\)](#) 彙總函數現在支援 INTERVAL 資料類型。
- `lpad()`, `rpadd()` – [lpad](#) 和 [rpad](#) 函數現在可在 VARBINARY 輸入上運作。
- `min()`, `max()` – [min\(\)](#) 和 [max\(\)](#) 彙總函數現在允許在查詢分析時使用未知的輸入類型，以便您可使用擁有 NULL 常值的函數。
- `regexp_replace()` – 新增了 [regexp\\_replace\(\)](#) 函數的變體，可為每個替換執行 Lambda 函數。
- `sequence()` – 新增了 [sequence\(\)](#) 函數的 DATE 變體，包括帶有隱含一天逐步增量的變體。
- `ST_Area()` – [ST\\_Area\(\)](#) 空間函數現在支援所有幾何類型。
- `substr()` – [substr](#) 函數現在可在 VARBINARY 輸入上運作。
- `zip_with()` – 長度不匹配的陣列現在可搭配 [zip\\_with\(\)](#) 使用。缺少的位置填充 null。先前，當傳遞不同長度的陣列時，會引發錯誤。這項變更可能導致難以區分原本為 null 的值與新增以填充陣列至相同長度的值。

## 新增的函數

下列清單包含 Athena 引擎版本 2 中新採用的函數。此清單不包括空間函數。如需空間函數的清單，請參閱 [Athena 引擎版本 2 中的新地理空間函數](#)。

如需每個函數的詳細資訊，請造訪 Presto 文件的對應連結。

## 彙總函數

### [reduce\\_agg\(\)](#)

## 陣列函數和運算子

[array\\_sort\(\)](#) - 新增了這個函數的變體，其將 Lambda 函數作為比較程式。

### [ngrams\(\)](#)

## 二進位函數和運算子

### [from\\_big\\_endian\\_32\(\)](#)

[from\\_ieee754\\_32\(\)](#)

[from\\_ieee754\\_64\(\)](#)

[hmac\\_md5\(\)](#)

[hmac\\_sha1\(\)](#)

[hmac\\_sha256\(\)](#)

[hmac\\_sha512\(\)](#)

[spooky\\_hash\\_v2\\_32\(\)](#)

[spooky\\_hash\\_v2\\_64\(\)](#)

[to\\_big\\_endian\\_32\(\)](#)

[to\\_ieee754\\_32\(\)](#)

[to\\_ieee754\\_64\(\)](#)

日期與時間函數和運算子

[millisecond\(\)](#)

[parse\\_duration\(\)](#)

[to\\_milliseconds\(\)](#)

對應函數和運算子

[multimap\\_from\\_entries\(\)](#)

數學函數和運算子

[inverse\\_normal\\_cdf\(\)](#)

[wilson\\_interval\\_lower\(\)](#)

[wilson\\_interval\\_upper\(\)](#)

分位數摘要函數

[分位數摘要函數](#)與新增的 `qdigest` 分位數摘要類型。

## 字串函數和運算子

### [hamming\\_distance\(\)](#)

### [split\\_to\\_multimap\(\)](#)

## 效能改進

Athena 引擎版本 2 中以下功能的效能已改善。

### 查詢效能

- 廣播聯結效能 – 藉由在工作節點中套用動態分割區剪清除來改善廣播聯結效能。
- 歸納的資料表 – 在寫入的資料已經適當分割時 (例如，當輸出來自歸納的聯結時)，改善了寫入歸納的資料表的效能。
- DISTINCT – 改善了某些使用 DISTINCT 之查詢的效能。

動態篩選和分割區清除 – 改善項目可提高效能並降低某些查詢中的掃描資料量。

- 篩選和投影操作 – 篩選和投影操作現在一律按資料欄處理 (如果可能)。引擎會在有效的情況下自動利用字典編碼。
- 收集交換 – 改善了使用收集交換的查詢效能。
- 全域彙總 – 改善了某些執行篩選全域彙總之查詢的效能。
- GROUPING SETS、CUBE、ROLLUP – 改善了涉及 GROUPING SETS、CUBE 或 ROLLUP 之查詢的效能，您可以用這些程式碼在單一查詢中彙總多組資料欄。
- 高選擇性的篩選條件 – 改善了使用高選擇性篩選條件的查詢效能。
- JOIN 和 AGGREGATE 操作 – 已經增強 JOIN 和 AGGREGATE 操作的效能。
- LIKE – 改善了在 information\_schema 資料表資料欄中使用 LIKE 述詞的查詢效能。
- ORDER BY 和 LIMIT – 針對涉及 ORDER BY 和 LIMIT 的查詢改善了計劃、效能和記憶體使用量，以避免不必要的資料交換。
- ORDER BY – ORDER BY 操作現在會依預設分配，讓較大的 ORDER BY 子句得到使用。
- ROW 類型轉換 – 改善了在 ROW 類型之間轉換的效能。
- 結構類型 – 改善了處理結構類型並包含掃描、聯結、彙總或資料表寫入的查詢效能。
- 資料表掃描 – 已推出新版最佳化規則，可避免在某些情況下重複掃描資料表。
- UNION – 改善了 UNION 查詢的效能。

## 查詢規劃效能

- 規劃效能 – 針對聯結包含大量資料欄的多個資料表的查詢，改善了規劃效能。
- 述詞評估 – 改善了規劃中述詞下推期間的述詞評估效能。
- 用於轉換的述詞下推支援 – 支援 `<column> IN <values list>` 述詞的述詞下推，其中值清單中的值需要轉換以符合資料欄的類型。
- 述詞推斷和下推 – 針對使用 `<symbol> IN <subquery>` 述詞的查詢，擴展了述詞推斷和下推。
- 逾時 – 已修正在極少數情況下可能導致查詢規劃逾時的錯誤。

## 聯結效能

- 與映射資料欄聯結 – 改善了包含映射資料欄之聯結和彙總的效能。
- 僅與不等式條件聯結 – 透過使用巢狀迴圈聯結，而非雜湊聯結，改善了僅與不等式條件聯結的效能。
- 外部聯結 – 現在會針對涉及外部聯結的查詢自動選取聯結分配類型。
- 函數聯結的範圍 – 改善了條件為函數範圍 (例如 `a JOIN b ON b.x < f(a.x) AND b.x > g(a.x)`) 之聯結的效能。
- S spill-to-disk — 修正 spill-to-disk 相關錯誤和記憶體問題，以提升效能並減少 JOIN 作業中的記憶體錯誤。

## 子查詢效能

- 相互關聯的 EXISTS 子查詢 – 改善了相互關聯的 EXISTS 子查詢的效能。
- 擁有等式的相互關聯子查詢 – 改善了對包含等式述詞的相互關聯子查詢的支援。
- 擁有不等式的相互關聯子查詢 – 改善了包含不等式的相互關聯子查詢的效能。
- 子查詢 Count(\*) 彙總 – 改善了具有已知常數基數的子查詢 `count(*)` 彙總的效能。
- 外部查詢篩選條件傳播 – 改善了在外部查詢的篩選條件可傳播至子查詢時相互關聯子查詢的效能。

## 函數效能

- 彙總視窗函數 – 改善了彙總視窗函數的效能。
- `element_at()` – 針對映射為常數時間，而不是與映射的大小成正比，改善了 `element_at()` 的效能。
- `Grouping()` – 改善了涉及 `grouping()` 的查詢效能。

- JSON 切換 – 改善了從 JSON 轉換至 ARRAY 或 MAP 類型的效能。
- 映射傳回函數 – 改善了傳回映射之函數的效能。
- Map-to-map 鑄造 – 改善了 map-to-map 鑄造的性能。
- Min() 和 max() – min() 和 max() 函數已經過最佳化處理，以避免建立不需要的物件，從而減少廢棄項目收集負荷。
- row\_number() – 針對使用 row\_number() 後對產生的資料行號進行篩選的查詢，改善了效能和記憶體使用量。
- 視窗函數 – 針對包含具有相同 PARTITION BY 和 ORDER BY 子句的視窗函數的查詢，改善了效能。
- 視窗函數 – 改善了某些具有類似規格的視窗函數 (例如 LAG) 的效能。

### 地理空間效能

- 幾何序列化 – 改善了幾何值的序列化效能。
- 地理空間函數 – 改善了 ST\_Intersects()、ST\_Contains()、ST\_Touches()、ST\_Within()、ST\_Overlaps()、ST\_Distance() 以及 array\_intersect() 的效能。
- ST\_Distance() – 改善了涉及 ST\_Distance() 函數的聯結查詢效能。
- ST\_Intersection() – 針對與座標軸對齊的矩形 (例如，由 ST\_Envelope() 和 bing\_tile\_polygon() 函數產出的多邊形)，最佳化了 ST\_Intersection() 函數。

### JSON 相關的改善

#### 對應函數

- 在所有案例中，改善了從  $O(n)$  至  $O(1)$  的映射下標效能。先前，只有特定函數和讀取器所產生的映射才能利用這項改善。
- 新增了 map\_from\_entries() 和 map\_entries() 函數。

#### 轉換

- 新增了從 REAL、TINYINT 或 SMALLINT 轉換至 JSON 的能力。
- 現在可以將 JSON 轉換至 ROW，即使 JSON 不包含 ROW 中的每個欄位，也可以做到。
- 改善了 CAST(json\_parse(...) AS ...) 的效能。

- 改善了從 JSON 轉換至 ARRAY 或 MAP 類型的效能。

## 新 JSON 函數

- [is\\_json\\_scalar\(\)](#)

## 重大變更

重大變更包括錯誤修正、地理空間函數的變更、替換的函數以及引入限制。ANSI SQL 合規的改善可能會中斷依賴於非標準行為的查詢。

## 錯誤修正

下列變更會修正造成查詢成功執行但結果不正確的行為問題。

- 現在接受 `fixed_len_byte_array` parquet 資料欄作為 DECIMAL – Parquet 資料欄類型  
`fixed_len_byte_array` 的查詢會成功並傳回正確的值，前提是將它們標註為 Parquet 結構描述中的 DECIMAL。沒有 DECIMAL 標註的 `fixed_len_byte_array` 資料欄查詢會失敗並顯示錯誤。先前，`fixed_len_byte_array` 資料欄查詢如果沒有 DECIMAL 標註會成功，但會傳回無法理解的值。
- `json_parse()` 不再忽略結尾字元 – 先前，例如 `[1,2]abc` 的輸入都會成功剖析為 `[1,2]`。使用結尾字元現在會產生錯誤訊息 `Cannot convert '[1, 2]abc' to JSON (無法將 '[1, 2]abc' 轉換為 JSON)`。
- `Round()` 小數精確度已修正 – 當 `x` 為 DECIMAL 時，或者當 `x` 為小數位數為 0 的 DECIMAL 且 `d` 是負整數時，`round(x, d)` 現在會正確四捨五入 `x`。先前，在這些情況下不會四捨五入。
- `round(x, d)` 和 `truncate(x, d)` – 函數 `round(x, d)` 和 `truncate(x, d)` 的簽章參數 `d` 現在是類型 INTEGER。先前，`d` 可以是類型 BIGINT。
- `Map()` 具有重複金鑰 – `map()` 現在發現重複金鑰會引發錯誤，而不會以無提示的方式產生毀損的映射。目前使用重複金鑰建構映射值的查詢現在會失敗並顯示錯誤。
- `map_from_entries()` 引發 null 項目錯誤 – 輸入陣列包含 null 項目時，`map_from_entries()` 現在會引發錯誤。透過傳遞 NULL 為值來構建映射的查詢現在會失敗。
- 資料表 – 無法再建立具有不支援分割區類型的資料表。
- 改善了統計函數的數字穩定性 – 統計函數 `corr()`、`covar_samp()`、`regr_intercept()` 以及 `regr_slope()` 的數字穩定性已改善。
- 在 parquet 中定義的 TIMESTAMP 精確度現在得以遵守 – TIMESTAMP 值的精確度和 Parquet 結構描述中為 TIMESTAMP 資料欄定義的精確度現在必須相符。不相符的精確度會導致不正確的時間戳記。

- 時區資訊 – 現在，時區資訊會使用 Java 1.8 軟體開發套件的 [java.time](#) 套件來計算。
- INTERVAL\_DAY\_TO\_SECOND 和 INTERVAL\_YEAR\_TO\_MONTH 資料類型的 SUM – 您無法再直接使用 SUM(NULL)。為了使用 SUM(NULL)，需將 NULL 轉換為資料類型，如 BIGINT、DECIMAL、REAL、DOUBLE、INTERVAL\_DAY\_TO\_SECOND 或 INTERVAL\_YEAR\_TO\_MONTH。

## 對空間函數的變更

對空間函數所做的變更包括以下內容。

- 函數名稱變更 – 某些函數名稱已變更。如需詳細資訊，請參閱 [Athena 引擎版本 2 中地理空間函數名稱變更](#)。
- VARBINARY 輸入 – VARBINARY 類型不再直接支援輸入到地理空間函數。例如，若要直接計算幾何圖形的面積，現在必須以 VARCHAR 或 GEOMETRY 格式輸入幾何圖形。解決方式是使用轉換函數，如下列範例所示。
  - 若要使用 ST\_area() 來計算已知二進位 (WKB) 格式的 VARBINARY 輸入的面積，請先將輸入傳遞至 ST\_GeomFromBinary()，例如：

```
ST_area(ST_GeomFromBinary(<wkb_varbinary_value>))
```

- 若要使用 ST\_area() 來計算舊版二進位格式的 VARBINARY 輸入的面積，請先將相同的輸入傳遞至 ST\_GeomFromLegacyBinary() 函數，例如：

```
ST_area(ST_GeomFromLegacyBinary(<legacy_varbinary_value>))
```

- ST\_ExteriorRing () 和 st\_Polygon () — [ST\\_ExteriorRing\(\)ST\\_Polygon\(\)](#) 現在只接受多邊形作為輸入。先前，這些函數錯誤地接受了其他幾何圖形。
- ST\_Distance () – 根據 [SQL/MM 規格](#) 的要求，如果其中一個輸入是空幾何圖形，[ST\\_Distance\(\)](#) 函數現在會傳回 NULL。先前，會傳回 NaN。

## ANSI SQL 合規

已修正下列語法和行為問題，以遵循 ANSI SQL 標準。

- Cast() 操作 – 從 REAL 或 DOUBLE 到 DECIMAL 的 Cast() 操作現在符合 SQL 標準。例如：cast (double '10000000000000000000000000000000' as

decimal(38)) 先前會傳回 100000000000000000005366162204393472，但現在會傳回 1000000000000000000000000000000000。

- JOIN ... USING – JOIN ... USING 現在符合標準 SQL 語義。先前，JOIN ... USING 需要使資料欄中的資料表名稱符合資格，並且兩個資料表的資料欄會出現在輸出中。資料表資格現在無效，且資料欄僅存在於輸出中一次。
- 已移除 ROW 類型文字 – 不再支援 ROW 類型文字格式 ROW<int, int>(1, 2)。改用語法 ROW(1 int, 2 int)。
- 分組彙總語義 – 分組彙總使用 IS NOT DISTINCT FROM 語義，而不是相等語義。分組彙總現在會傳回正確的結果，並在 NaN 浮點值分組時展現出更高的效能。支援對映射、清單和包含 null 的列類型進行分組。
- 不再允許帶引號的類型 – 根據 ANSI SQL 標準，資料類型不能再用引號括住。例如：SELECT "date" '2020-02-02' 不再是有效的查詢。請改用語法 SELECT date '2020-02-02'。
- 匿名資料行欄位存取 – 無法再使用語法 [.field0, .field1, ...] 存取匿名資料行欄位。
- 複雜分組操作 – 複雜的分組操作 GROUPING SETS、CUBE 以及 ROLLUP 不支援對由輸入資料欄組成的表達式進行分組。只允許資料欄名稱。

## 替換的函數

不再支援下列函數，並以產生相同結果的語法替換。

- information\_schema.\_\_internal\_partitions\_\_ – Athena 引擎版本 2 不再支援使用 \_\_internal\_partitions\_\_。針對相等語法，請使用 SELECT \* FROM "<table\_name>\$partitions" 或 SHOW PARTITIONS。如需詳細資訊，請參閱[列出特定資料表的分割區](#)。
- 替換的地理空間函數 – 如需名稱已變更之地理空間函數的清單，請參閱[Athena 引擎版本 2 中地理空間函數名稱變更](#)。

## 限制

Athena 引擎版本 2 中引入了下列限制，以確保查詢不會因資源限制而失敗。使用者無法設定這些限制。

- 結果元素的數量 – 下列函數的結果元素 n 的數量限制為 10,000 或更少：min(col, n)、max(col, n)、min\_by(col1, col2, n) 以及 max\_by(col1, col2, n)。
- GROUPING SETS – 分組集中的配量數目上限為 2048。

- 文字檔案一行長度上限 – 文字檔案預設的一行長度上限為 200 MB。
- 序列函數結果大小上限 – 序列函數的結果大小上限為 50,000 個項目。例如：SELECT sequence(0,45000,1) 成功，但 SELECT sequence(0,55000,1) 失敗並顯示錯誤訊息 The result of the sequence function must not have more than 50000 entries (序列函數的結果不能有 50,000 個以上項目)。此限制適用於序列函數 (包括時間戳記) 的所有輸入類型。

## Athena 的 SQL 參考

Amazon Athena 支援一部分的資料定義語言 (DDL) 和資料處理語言 (DML) 陳述式、函數、運算子和資料類型。除了一些例外情況下，Athena DDL 是基於 HiveQL DDL 和 Athena DML 是基於特里諾。如需有關 Athena 引擎版本的資訊，請參閱[Athena 引擎版本控制](#)。

### 主題

- [Amazon Athena 中的資料類型](#)
- [DML 查詢、函數和運算子](#)
- [DDL 陳述式](#)
- [在 Amazon Athena 中進行 SQL 查詢的考量事項與限制](#)

## Amazon Athena 中的資料類型

當您執行時 CREATE TABLE，您可以指定資料行名稱和每個資料行可包含的資料類型。您建立的表格會儲存在中 AWS Glue Data Catalog。

為了促進與其他查詢引擎的互操作性，Athena 使用 [Apache Hive](#) 資料類型名稱，例如 CREATE TABLE DDL 陳述式。對於像、和這樣的 DML 查詢 SELECT CTASINSERT INTO，Athena 使用 [Trino](#) 資料類型名稱。下表顯示 Athena 支援的資料類型。其中 DDL 和 DML 類型在名稱、可用性或語法方面有所不同，它們會顯示在不同的資料行中。

DDL	DML	描述
BOOLEAN		值為 true 和 false。
TINYINT		一個 8 位有符號的整數，採用二進制補碼格式，最小值為 $-2^7$ ，最大值為 $2^7 - 1$ 。

DDL	DML	描述
SMALLINT		16 位元帶正負號整數，採用二進制補碼格式，最小值為 $-2^{15}$ ，最大值為 $2^{15}-1$ 。
INT、INTEGER		32 位元帶正負號值，採用二進制補碼格式，最小值為 $-2^{31}$ ，最大值為 $2^{31}-1$ 。
BIGINT		一個 64 位有符號的整數，採用二進制補碼格式，最小值為 $-2^{63}$ ，最大值為 $2^{63}-1$ 。
FLOAT	REAL	32 位元帶正負號的單精度浮點數。範圍為正值或負值。遵循浮點運算的 IEEE 標準 (IEEE 754)。
DOUBLE		64 位有符號的雙精度浮點數。該範圍是正或負的範圍是 遵循浮點運算的 IEEE 標準 (IEEE 754)。
十進制 ( <i>##</i> , <i>#度</i> )		<i>precision</i> 是位數的總數。 <i>scale</i> (選擇性) 是小數部分的位數，預設值為 0。例如，使用這些類型定義： <code>decimal(11,5)</code> 、 <code>decimal(15)</code> 。最大 <i>###</i> 為 38，而最大 <i>##</i> 值為 38。
字符，字符 ( <i>##</i> )		固定長度字元資料，指定長度介於 1 到 255 之間，例如 <code>char(10)</code> 。如果指定 <i>length</i> ，字串會在讀取時以指定的長度截斷。如果基礎資料字串較長，則基礎資料字串會保持不變。  如需詳細資訊，請參閱 <a href="#">CHAR Hive 資料類型</a> 。
STRING	VARCHAR	可變長度字符數據。
<i>#####</i>		具有最大讀取長度的可變長度字符數據。讀取時，字串會以指定的長度截斷。如果基礎資料字串較長，則基礎資料字串會保持不變。
BINARY	VARBINARY	可變長度的二進制數據。
TIME		精確度為毫秒的一天時間。

DDL	DML	描述
無	時間 (###)	具有特定精確度的一天中的時間。 TIME(3)相當於TIME。
無	TIME WITH TIME ZONE	某個時區中的一天中的時間。時區應指定為與 UTC 的偏移量。
DATE		帶有年，月和日的日曆日期。
TIMESTAMP	時間戳記，無時區的時間戳記	具有毫秒精確度的日曆日期和時間。
無	時間戳 (##) ， 時間戳 (##) ， 沒有時區	具有特定精確度的日曆日期和時間。 TIMESTAMP(3) 相當於TIMESTAMP 。
無	TIMESTAMP WITH TIME ZONE	時區中的日曆日期和時間。您可以將時區指定為與 UTC 的偏移量、IANA 時區名稱，或使用 UTC、UT、Z 或 GMT。
無	具有時區的時間戳記 (##)	在時區中具有特定精確度的日曆日期和時間。
無	INTERVAL YEAR TO MONTH	一個或多個整個月的間隔
無	INTERVAL DAY TO SECOND	一或多秒、分、小時或天的間隔
## < ##### >	陣列 [##類型]	值的陣列。所有值都必須具有相同的類型。
## < ##### >	MAP## _ #####	一個地圖，其中值可以通過鍵查找。所有鍵必須具有相同的值，並且所有值都必須具有相同的值。

DDL	DML	描述
<pre>## &lt; #### _1### _ ## _1##### _2## ### _2#...</pre>	<pre>##### _1 # ### _1#### # _2 ##### _2#...#</pre>	具有命名欄位及其值的資料結構。
無	JSON	JSON 值類型，可以是 JSON 物件、JSON 陣列、JSON 數字、JSON 字串true, false或null.
無	UUID	UUID (通用唯一識別碼)。
無	IP 地址	一個 IPv4 或 IPv6 位址。
無	<a href="#">HyperLogLog</a> <a href="#">P4 HyperLogLog</a> <a href="#">SetDigest</a> <a href="#">Q 文摘</a> <a href="#">TDigest</a>	這些數據類型支持近似函數內部。有關每種類型的更多信息，請訪問 Trino 文檔中相應條目的鏈接。

## 資料類型範例

下表顯示 DML 資料類型的範例常值。

資料類型	範例
BOOLEAN	true false
TINYINT	TINYINT '123'
SMALLINT	SMALLINT '123'
INT、INTEGER	123456790

資料類型	範例
BIGINT	BIGINT '1234567890'  2147483648
REAL	'123456.78'
DOUBLE	1.234
十進制 ( ## , #度 )	DECIMAL '123.456'
字符, 字符 ( ## )	CHAR 'hello world', CHAR 'hello 'world''!
#####	VARCHAR 'hello world', VARCHAR 'hello 'world''!
VARBINARY	X'00 01 02'
時間、時間 ( ### )	TIME '10:11:12' , TIME '10:11:12.345'
TIME WITH TIME ZONE	TIME '10:11:12.345 -06:00'
DATE	DATE '2024-03-25'
時間戳, 沒有時區的時間戳, 時間戳 ( ## ), 沒有時區的時間戳 ( ## )	TIMESTAMP '2024-03-25 11:12:13' , TIMESTAMP '2024-03-25 11:12:13.456'
帶有時區的時間戳, 帶有時區的時間戳 ( ## )	TIMESTAMP '2024-03-25 11:12:13.456 Europe/Berlin'
INTERVAL YEAR TO MONTH	INTERVAL '3' MONTH
INTERVAL DAY TO SECOND	INTERVAL '2' DAY
陣列 [ ## 類型 ]	ARRAY['one', 'two', 'three']

資料類型	範例
<i>MAP## _ #####</i>	<pre>MAP(ARRAY['one', 'two', 'three'], ARRAY[1, 2, 3])</pre> <p>請注意，映射是從鍵數組和值數組創建的。</p>
<i>##### _1 ##### _1#### # _2 ##### _2#...#</i>	<pre>ROW('one', 'two', 'three')</pre> <p>請注意，以這種方式創建的行沒有列名。若要新增欄名稱，您可以使用CAST，如下列範例所示：</p> <pre>CAST(ROW(1, 2, 3) AS ROW(one INT, two INT, three INT))</pre>
JSON	<pre>JSON '{"one":1, "two": 2, "three": 3}'</pre>
UUID	<pre>UUID '12345678-90ab-cdef-1234-567890abcdef'</pre>
IP 地址	<pre>IPADDRESS '10.0.0.1'</pre> <pre>IPADDRESS '2001:db8::1'</pre>

## 資料類型的考量

### 大小限制

對於未指定大小限制的資料類型，請記住，單一資料列中所有資料的實際限制為 32MB。如需詳細資訊，請參閱 [在 Amazon Athena 中進行 SQL 查詢的考量事項與限制](#) 中的 [Row or column size limitation](#)。

### 字符和瓦爾查爾

CHAR(*n*) 值始終具有 *n* 字符數。例如，如果您將 'abc' 轉換為 CHAR(7)，則會添加 4 個尾隨空格。

CHAR 值的比較包括前導和尾隨空格。

如果為 CHAR 或指定長度 VARCHAR，字串會在讀取時以指定的長度截斷。如果基礎資料字串較長，則基礎資料字串會保持不變。

若要逸出 CHAR 或中的單引號 VARCHAR，請使用額外的單引號。

若要將非字串資料類型轉換為 DML 查詢中的字串，請轉換為 VARCHAR 資料類型。

若要使用 `substr` 函數從 CHAR 資料類型傳回指定長度的子字串，您必須先將 CHAR 值轉換為 VARCHAR。在下列範例中，`col1` 使用 CHAR 資料類型。

```
substr(CAST(col1 AS VARCHAR), 1, 4)
```

## DECIMAL

若要在 SELECT 查詢中將十進位值指定為常值，例如在選取具有特定十進位值的資料列時，您可以在查詢中指定 DECIMAL 類型，並將小數值列為單引號中的常值，如下列範例所示。

```
SELECT * FROM my_table  
WHERE decimal_value = DECIMAL '0.12'
```

```
SELECT DECIMAL '44.6' + DECIMAL '77.2'
```

## 使用時間戳記資料

本節說明在 Athena 中使用時間戳記資料的一些考量。

### Note

Athena 引擎版本 2 與 Athena 引擎版本 3 之間的時間戳記處理方式發生變更。如需有關 Athena 引擎版本 3 中可能發生的時間戳記相關錯誤及建議的解決方案的資訊，請參閱 [Athena 引擎版本 3 參考中的 時間戳記變更](#)。

將時間戳記資料寫入 Amazon S3 物件的格式

將時間戳記資料寫入 Amazon S3 物件的格式取決於資料行資料類型和您使用的程式 [SerDe 庫](#)。

- 如果您有類型為 DATE 的資料表資料欄，Athena 會預期資料的對應資料欄或屬性是 ISO 格式的字串 YYYY-MM-DD，或是內建的日期類型 (例如 Parquet 或 ORC 的日期類型)。
- 如果您有類型為 TIME 的資料表資料欄，Athena 會預期資料的對應資料欄或屬性是 ISO 格式的字串 HH:MM:SS，或是內建的時間類型 (例如 Parquet 或 ORC 的時間類型)。
- 如果您有類型為 TIMESTAMP 的資料表資料欄，Athena 會預期資料的對應資料欄或屬性是格式的字串 YYYY-MM-DD HH:MM:SS.SSS (注意日期和時間之間的空格)，或是內建的時間類型 (例如 Parquet、ORC 或 Ion 的時間類型)。

**Note**

OpenCSV SerDe 時間戳記是例外狀況，必須編碼為毫秒解析度的 UNIX 紀元。

## 確保時間分割資料符合記錄中的時間戳記欄位

資料的產生者必須確定分割區值與分割區內的資料一致。例如，如果您的資料具有 timestamp 屬性，而您使用 Firehose 將資料載入 Amazon S3，則必須使用 [動態分割](#)，因為 Firehose 的預設分割為 wall-clock-based

## 使用字串做為分割區索引鍵的資料類型

出於效能考量，最好將 STRING 用作分割區索引鍵的資料類型。雖然 Athena 將格式 YYYY-MM-DD 的分割區值識別為您使用 DATE 類型時的日期，而這可能造成效能不彰。因此，我們建議您改用 STRING 資料類型做為分割區索引鍵。

## 如何寫入同樣按時間分割的時間戳記欄位的查詢

如何寫入同樣按時間分割的時間戳記欄位的查詢，取決於您要查詢的資料表的類型。

### Hive 資料表

使用 Athena 中最常用的 Hive 資料表，查詢引擎對資料欄和分割區索引鍵之間的關係一無所知。因此，您必須始終在資料欄和分割區索引鍵的查詢中新增述詞。

例如，假設您有 event\_time 資料欄和 event\_date 分割區索引鍵，而且想要查詢 23:00 到 03:00 之間的事件。在此案例中，您必須在查詢資料欄和分割區索引鍵中包含述詞，如下列範例所示。

```
WHERE event_time BETWEEN start_time AND end_time
      AND event_date BETWEEN start_time_date AND end_time_date
```

### Iceberg 資料表

使用 Iceberg 資料表，您可以使用運算的分割區值，從而簡化查詢。例如，假設您的 Iceberg 資料表是使用如下 PARTITIONED BY 子句建立的：

```
PARTITIONED BY (event_date month(event_time))
```

在此案例下，查詢引擎會根據 `event_time` 述詞的值自動剔除分割區。因此，您的查詢只需要指定 `event_time` 的述詞，如下列範例所示。

```
WHERE event_time BETWEEN start_time AND end_time
```

如需更多詳細資訊，請參閱 [建立 Iceberg 資料表](#)。

## DML 查詢、函數和運算子

Athena DML 查詢引擎通常支援 Trino 和 Presto 語法，並且自行對其進行了改進。Athena 不支援 Trino 或 Presto 的所有功能。如需詳細資訊，請參閱本節中特定陳述式的主題以及 [考量與限制](#)。如需有關函數資訊，請參閱 [Amazon Athena 中的函數](#)。如需有關 Athena 引擎版本的資訊，請參閱 [Athena 引擎版本控制](#)。

如需 DDL 陳述式的資訊，請參閱 [DDL 陳述式](#)。如需不支援的 DDL 陳述式之清單，請參閱 [不支援的 DDL](#)。

## SELECT

從零個或多個資料表中擷取資料列。

### Note

本主題提供摘要資訊以供參考。關於使用 SELECT 和 SQL 語言的完整資訊已超出本文件的範圍。如需有關使用 Athena 特定 SQL 的資訊，請參閱 [在 Amazon Athena 中進行 SQL 查詢的考量事項與限制](#) 和 [使用 Amazon Athena 執行 SQL 查詢](#)。如需在 Athena 建立資料庫、建立資料表和執行資料表 SELECT 查詢的範例，請參閱 [開始使用](#)。

## 概要

```
[ WITH with_query [, ...] ]  
SELECT [ ALL | DISTINCT ] select_expression [, ...]  
[ FROM from_item [, ...] ]  
[ WHERE condition ]  
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]  
[ HAVING condition ]  
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
```

```
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]  
[ OFFSET count [ ROW | ROWS ] ]  
[ LIMIT [ count | ALL ] ]
```

### Note

SQL SELECT 陳述式中的保留字必須以雙引號括住。如需詳細資訊，請參閱[SQL SELECT 陳述式中的預留關鍵字清單](#)。

## 參數

```
[ WITH with_query [, ...] ]
```

您可以使用 WITH 來展開巢狀查詢，或簡化子查詢。

Athena 引擎版本 3 開始支援使用 WITH 子句建立遞迴查詢。最大遞歸深度為 10。

WITH 子句位於查詢中的 SELECT 清單前面，可定義一或多個子查詢以用於 SELECT 查詢內。

每個子查詢定義臨時資料表，類似於檢視定義，供您在 FROM 子句中參考。這些資料表只在查詢執行時才使用。

with\_query 語法是：

```
subquery_table_name [ ( column_name [, ...] ) ] AS (subquery)
```

其中：

- subquery\_table\_name 是臨時資料表的唯一名稱，定義 WITH 子句子查詢的結果。每個 subquery 必須有可在 FROM 子句中參考的資料表名稱。
- column\_name [, ...] 是選用的輸出欄名稱清單。欄名稱的數目必須等於或少於 subquery 所定義的欄數。
- subquery 是任何查詢陳述式。

[全部 | 不同] 選擇表達式

select\_expression 決定要選取的列。A select\_expression 可以使用下列格式之一：

```
expression [ [ AS ] column_alias ] [, ...]
```

```
row_expression.* [ AS ( column_alias [, ...] ) ]
```

```
relation.*
```

```
*
```

- `expression [ [ AS ] column_alias ]`語法指定一個輸出列。可選`[AS]` `column_alias`語法指定要用於輸出中欄的自訂標題名稱。
- For `row_expression.* [ AS ( column_alias [, ...] ) ]` `row_expression` 是資料類型的任意運算式ROW。資料列的欄位定義要包含在結果中的輸出資料行。
- 對於`relation.*`，的欄`relation`會包括在結果中。此語法不允許使用資料行別名。
- 星號`*`指定所有資料行都包含在結果集中。
- 在結果集中，欄的順序與 `select` 運算式的規格順序相同。如果 `select` 表達式返回多個列，則列順序遵循源關係或行類型表達式中使用的順序。
- 指定欄別名後，別名會覆寫預先存在的欄或列欄位名稱。如果 `select` 表達式沒有列名，則輸出中將顯示零索引的匿名列名稱 ( `_col0_col1` , , `_col2`, ... )。
- 預設值為 `ALL`。 `ALL` 使用與省略不用都一樣；所有欄的所有列都會選取，重複的列也會保留。
- 當欄包含重複值時，可使用 `DISTINCT` 只傳回獨特值。

```
FROM from_item [, ...]
```

指出查詢的輸入，其中 `from_item` 可以是檢視、聯結建構或子查詢，如下所示。

`from_item` 可以是以下任一項：

- `table_name [ [ AS ] alias [ (column_alias [, ...]) ] ]`

其中，`table_name` 是從中選取列的目標表格的名稱，`alias` 是提供給 `SELECT` 陳述式輸出的名稱，`column_alias` 為指定的 `alias` 定義欄。

-或-

- `join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]`

其中 `join_type` 是以下其中一個：

- `[ INNER ] JOIN`

- LEFT [ OUTER ] JOIN
- RIGHT [ OUTER ] JOIN
- FULL [ OUTER ] JOIN
- CROSS JOIN
- ON *join\_condition* | USING (*join\_column* [, ...]) 其中，使用 *join\_condition* 可讓您指定多個資料表中聯結索引鍵的欄名稱，而使用 *join\_column* 之前，兩個資料表中必須都有 *join\_column*。

### [ WHERE condition ]

根據您指定的 *condition* 篩選結果，其中 *condition* 通常具有以下語法。

```
column_name operator value [[[AND | OR] column_name operator value] ...]
```

*operator* 可以是其中一個比較子：=、>、<、>=、<=、<>、!=。

下面的子查詢表達式也可以用在 WHERE 子句中。

- [NOT] BETWEEN *integer\_A* AND *integer\_B* – 指定兩個整數之間的範圍，如下列範例所示。如果資料欄資料類型是 *varchar*，則必須先將資料欄轉換為整數。

```
SELECT DISTINCT processid FROM "webdata"."impressions"
WHERE cast(processid as int) BETWEEN 1500 and 1800
ORDER BY processid
```

- [NOT] LIKE *value* – 搜尋指定的模式。使用百分比符號 (%) 作為萬用字元，如下列範例所示。

```
SELECT * FROM "webdata"."impressions"
WHERE referrer LIKE '%.org'
```

- [NOT] IN (*value* [, *value* [, ...]]) – 指定資料欄可能的值之清單，如下列範例所示。

```
SELECT * FROM "webdata"."impressions"
WHERE referrer IN ('example.com','example.net','example.org')
```

### [ GROUP BY [ ALL | DISTINCT ] grouping\_expressions [, ...] ]

將 SELECT 陳述式的輸出分割成具有相符值的列。

ALL 和 DISTINCT 指定重複的分組集是否各產生相異的輸出列。如果省略，會採用 ALL。

`grouping_expressions` 可讓您執行複雜的分組操作。您可以使用複雜的分組操作來執行需要對單一查詢中的多組資料欄進行彙總的分析。

`grouping_expressions` 元素可以是任何函數，例如對輸入資料欄執行的 SUM、AVG 或 COUNT。

GROUP BY 表達式可依未出現在 SELECT 陳述式輸出中的輸入欄名稱，將輸出分組。

所有輸出表達式必須是彙總函數，或存在於 GROUP BY 子句中的欄。

您可以使用單一查詢來執行需要彙總多個欄集的分析。

Athena 支援使用 GROUPING SETS、CUBE 和 ROLLUP 的複雜彙總。GROUP BY GROUPING SETS 指定要分組的多個資料欄清單。GROUP BY CUBE 會針對一組給定的資料欄，產生所有可能的分組集。GROUP BY ROLLUP 會針對一組給定的資料欄，產生所有可能的小計。複雜的分組操作不支援對由輸入資料欄組成的表達式進行分組。只允許資料欄名稱。

您通常可以使用 UNION ALL 來達到與這些 GROUP BY 操作相同的結果，但使用 GROUP BY 的查詢好處是只讀取資料一次，而 UNION ALL 會讀取基礎資料三次，而且當資料來源易於變更時，可能會產生不一致的結果。

#### [ HAVING condition ]

與彙總函數和 GROUP BY 子句一起使用。控制要選取哪些群組，排除不符合 condition 的群組。計算群組和彙總之後會執行此篩選。

#### [ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] union\_query ]

UNION、INTERSECT 和 EXCEPT 將一個以上的 SELECT 陳述式的結果結合為單一查詢。ALL 或 DISTINCT 控制最終結果集包含的資料行的唯一性。

UNION 將第一個查詢產生的資料行與第二個查詢產生的資料行組合起來。為消除重複項，UNION 構建了一個消耗記憶體之雜湊資料表。為了更好的性能，如果您的查詢不需要消除重複項，則請考慮使用 UNION ALL。除非您使用括號來明確定義處理順序，否則多個 UNION 子句是由左向右處理。

INTERSECT 只傳回存在於第一個和第二個查詢之結果中的資料行。

EXCEPT 傳回第一個查詢之結果的資料行，不包括第二個查詢找到的資料行。

ALL 會包含所有資料行，即使資料行都相同也一樣。

DISTINCT 只會讓唯一的資料行包含在合併結果集中。

```
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
```

依一或多個輸出 `expression` 排序結果集。

當子句包含多個表達式時，結果集會根據第一個 `expression` 排序。然後，第二個 `expression` 套用的列有來自第一個表達式的相符值，以此類推。

每個 `expression` 可以指定 `SELECT` 的輸出欄，或依位置指定輸出欄的序數 (從 1 開始)。

`ORDER BY` 是在任何 `GROUP BY` 或 `HAVING` 子句之後當作最後一個步驟來評估。`ASC` 和 `DESC` 決定結果依遞增或遞減順序來排序。

無論是遞增或遞減排序，預設空值排序是 `NULLS LAST`。

```
[ OFFSET count [ ROW | ROWS ] ]
```

使用 `OFFSET` 子句來捨棄結果集中的數個前導列。如果 `ORDER BY` 子句存在時，`OFFSET` 子句會根據排序的結果集進行評估，並且該集會在跳過的資料列遭到捨棄後保持排序。如果查詢沒有 `ORDER BY` 子句，則可以任意捨棄資料列。如果 `OFFSET` 指定的計數等於或超過結果集的大小，則最終結果為空白。

```
LIMIT [ count | ALL ]
```

將結果集的列數限制為 `count`。`LIMIT ALL` 和省略 `LIMIT` 子句的效果相同。如果查詢沒有 `ORDER BY` 子句，則結果是任意順序。

```
TABLESAMPLE [ BERNOULLI | SYSTEM ] (percentage)
```

根據取樣方法從資料表中選擇列的選用運算子。

`BERNOULLI` 以機率 `percentage` 選取要放入資料表樣本中的每一列。將會掃描資料表的所有實體區塊，並根據範本 `percentage` 和執行時間計算的隨機值之間的比較，略過特定的資料行。

使用 `SYSTEM` 時，資料表會分割為邏輯的資料區段，並以此精細程式來取樣資料表。

將會選取特定區段的所有資料行，或根據取樣 `percentage` 和執行時間計算的隨機值之間的比較，略過區段。`SYSTEM` 取樣依賴連接器。這個方法不保證獨立取樣機率。

```
[ UNNEST (array_or_map) [WITH ORDINALITY] ]
```

將陣列或映射展開成關係。陣列會展開成單一欄。映射會展開成兩欄 (索引鍵、值)。

您可以使用 `UNNEST` 搭配多個參數，以展開成多個欄，且列數與最高基數引數相等。

其他欄以空值填補。

WITH ORDINALITY 子句會將基數欄新增至尾端。

UNNEST 通常與 JOIN 一起使用，且可以參考 JOIN 左側關係中的欄。

在 Amazon S3 中取得來源資料的檔案位置

若要查看資料表資料行中資料的 Amazon S3 檔案位置，可以使用 SELECT 查詢中的 "\$path"，如下列範例所示：

```
SELECT "$path" FROM "my_database"."my_table" WHERE year=2019;
```

此舉會傳回如下所示的結果：

```
s3://DOC-EXAMPLE-BUCKET/datasets_mytable/year=2019/data_file1.json
```

若要傳回資料表中資料的 S3 檔案名稱路徑經排序的單一清單，可以使用 SELECT DISTINCT 和 ORDER BY，如下範例所示。

```
SELECT DISTINCT "$path" AS data_source_file  
FROM sampledb.elb_logs  
ORDER By data_source_file ASC
```

若要僅傳回沒有路徑的檔案名稱，則可將 "\$path" 作為參數傳遞給 regexp\_extract 函數，如下列範例所示。

```
SELECT DISTINCT regexp_extract("$path", '[^/]+$') AS data_source_file  
FROM sampledb.elb_logs  
ORDER By data_source_file ASC
```

若要從特定檔案傳回資料，請在 WHERE 子句中指定檔案，如下列範例所示。

```
SELECT *, "$path" FROM my_database.my_table WHERE "$path" = 's3://DOC-EXAMPLE-BUCKET/  
my_table/my_partition/file-01.csv'
```

如需詳細資訊和範例，請參閱知識中心文章：[如何查看 Athena 資料表中某資料行的 Amazon S3 來源檔案？](#)。

**Note**

在 Athena 中，檢視不支援 Hive 或 Iceberg 隱藏的中繼資料欄 `$bucket`、`$file_modified_time`、`$file_size` 和 `$partition`。

## 逸出單引號

若要逸出單引號，請在它前面加上另一個單引號，如下列範例所示。請勿將此與雙引號混淆。

```
Select '0''Reilly'
```

## 結果

```
0'Reilly
```

## 其他資源

如需有關使用 Athena 中 SELECT 陳述式的詳細資訊，請參閱以下資源。

如需有關以下內容的資訊	參閱以下資源
在 Athena 中執行查詢	<a href="#">使用 Amazon Athena 執行 SQL 查詢</a>
使用 SELECT 建立資料表	<a href="#">從查詢結果建立資料表 (CTAS)</a>
將一個 SELECT 查詢的資料插入其他資料表	<a href="#">INSERT INTO</a>
使用 SELECT 陳述式中內建的函數	<a href="#">Amazon Athena 中的函數</a>
使用 SELECT 陳述式中使用使用者定義函數	<a href="#">利用使用者定義函數來查詢</a>
查詢資料目錄中繼資料	<a href="#">查詢 AWS Glue Data Catalog</a>

## INSERT INTO

根據來源資料表上執行的 SELECT 查詢陳述式，或根據作為該陳述式的一部分提供的一組 VALUES，將新資料行插入目標資料表。當來源資料表以某一格式 (例如 CSV 或 JSON) 的基礎資料為基礎時，而目標資料表以另一種格式 (例如 Parquet 或 ORC) 為基礎，則您可以使用 INSERT INTO 查詢，將選取的資料轉換為目標資料表的格式。

## 考量與限制

搭配 Athena 使用 INSERT 查詢時，請考慮以下情況：

- 在包含於 Amazon S3 中加密的基礎資料的資料表上執行 INSERT 查詢時，INSERT 查詢寫入的輸出檔案依預設不會加密。如果您要插入具有加密資料的資料表，建議您加密 INSERT 查詢結果。

如需使用主控台加密查詢結果的詳細資訊，請參閱[加密 Amazon S3 中存放的 Athena 查詢結果](#)。若要使用 AWS CLI 或 Athena API 啟用加密，請使用[StartQueryExecution](#)動作的 EncryptionConfiguration 屬性根據您的需求指定 Amazon S3 加密選項。

- 對於 INSERT INTO 陳述式，預期的儲存貯體擁有者設定不適用於 Amazon S3 中的目的地資料表位置。預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Amazon S3 輸出位置。如需詳細資訊，請參閱[使用 Athena 主控台指定查詢結果位置](#)。
- 如需 ACID 合規 INSERT INTO 陳述式，則請參閱[更新 Iceberg 資料表資料](#)的 INSERT INTO 一節。

## 支援的格式和 SerDes

您可以在使用下列格式和的資料建立的資料表上執行 INSERT 查詢 SerDes。

資料格式	SerDe
Avro	或. 阿帕切 .hadoop.HIVE. 服務 2. AvroSerDe
Ion	am. 亞馬遜。離子組織. IonHiveSerDe
JSON	或者 .apache.h目錄/數據。 JsonSerDe
ORC	組織. 阿帕切. 哈多普. 生存 .QL.io. OrcSerde
Parquet	或者阿帕奇 .hadoop.hive.ql.io.parquet.serde. ParquetHiveSerDe
文字檔案	或者 .apache.hadoop.hive. 服務器 2. LazySimpleSerDe

 **Note**  
支援 CSV、TSV 和自訂分隔檔案。

## 不支援分區資料表

在分區資料表上不支援 INSERT INTO。如需詳細資訊，請參閱 [在 Athena 中分割和歸納](#)。

## 不支援聯合查詢

INSERT INTO 不支援聯合查詢。嘗試執行這項操作可能會產生錯誤訊息：外部目錄目前不支援此操作。如需聯合查詢的詳細資訊，請參閱 [使用 Amazon Athena 聯合查詢](#)。

## 分割

搭配 INSERT INTO 或 CREATE TABLE AS SELECT 查詢使用分割區時，請將本章節的幾個重點納入考量。

## 限制

INSERT INTO 陳述式支援最多將 100 個分割區寫入目標資料表。如果您在分割區超過 100 個的資料表上執行 SELECT 子句，除非 SELECT 查詢限制在 100 個分割區或更少，否則查詢會失敗。

如需解決此限制的相關資訊，請參閱 [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)。

## 資料欄排序

INSERT INTO 或 CREATE TABLE AS SELECT 陳述式會預期分割的資料欄為 SELECT 陳述式中投影資料欄清單的最後一個資料欄。

如果來源資料表未經分割，或與目標資料表相比在不同的資料欄上分割，則如 INSERT INTO *destination\_table* SELECT \* FROM *source\_table* 的查詢會將來源資料表中的最後一個資料欄的值，認定為目標資料表中的分割區資料欄的值。嘗試從未分割的資料表建立分割的資料表時，請注意以上提醒。

## 資源

如需有關搭配分割使用 INSERT INTO 的詳細資料，請參閱以下資源。

- 如需將分割的資料插入分割的資料表，請參閱 [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)。
- 如需將未分割的資料插入分割的資料表，請參閱 [使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析](#)。

## 寫入 Amazon S3 的檔案

INSERT 命令執行之後，Athena 會將檔案寫入 Amazon S3 中的來源資料位置。每個 INSERT 操作都會建立新的檔案，而不是附加到現有的檔案。檔案位置取決於資料表和 SELECT 查詢的結構 (若有)。Athena 會為每個 INSERT 查詢生成一個資訊清單檔案。資訊清單會追蹤查詢寫入的檔案。它會儲存到 Amazon S3 中的 Athena 查詢結果位置。如需詳細資訊，請參閱 [識別查詢輸出檔案](#)。

### 避免高度交易更新

當您使用將資料列新增 INSERT INTO 至 Amazon S3 中的資料表時，Athena 不會重寫或修改現有檔案。相反地，它會將資料列寫入為一或多個新檔案。由於包含 [許多小檔案的資料表會導致查詢效能降低](#)，而且寫入和讀取操作 (例如 PutObject Amazon S3 會產生較高的成本)，GetObject 因此在使用時請考慮下列選項 INSERT INTO：

- 在較大批次的資料列上執行 INSERT INTO 作業的頻率較低。
- 對於大型資料擷取磁碟區，請考慮使用 [Amazon 資料 Fire hose](#) 之類的服務。
- INSERT INTO 完全避免使用。相反地，將資料列累積成較大的檔案，然後直接上傳到 Amazon S3，Athena 就能查詢這些資料列。

### 尋找孤立檔案

如果 CTAS 或 INSERT INTO 陳述式失敗，孤立的資料可以留在資料位置，而且可能會在後續查詢中讀取。若要尋找孤立檔案以進行檢測或刪除，您可以使用 Athena 提供的資料資訊清單檔案來追蹤要寫入的檔案清單。如需詳細資訊，請參閱 [識別查詢輸出檔案](#) 及 [DataManifestLocation](#)。

## INSERT INTO...SELECT

指定在一個資料表上執行的查詢：source\_table，這會決定要插入第二個資料表的資料列：destination\_table。如果 SELECT 查詢指定 source\_table 中的資料欄，則該資料欄必須精確地符合 destination\_table 中的資料欄。

如需 SELECT 查詢的詳細資訊，請參閱 [SELECT](#)。

### 概要

```
INSERT INTO destination_table
SELECT select_query
FROM source_table_or_view
```

## 範例

選取 `vancouver_pageviews` 資料表中的所有列，並將其插入 `canada_pageviews` 資料表：

```
INSERT INTO canada_pageviews
SELECT *
FROM vancouver_pageviews;
```

僅選取 `vancouver_pageviews` 資料表中的列，其中 `date` 資料欄有介於 2019-07-01 到 2019-07-31 間的值，然後將其插入 `canada_july_pageviews`：

```
INSERT INTO canada_july_pageviews
SELECT *
FROM vancouver_pageviews
WHERE date
      BETWEEN date '2019-07-01'
            AND '2019-07-31';
```

請僅從 `country` 資料欄中 `usa` 的值選取 `cities_world` 資料表中 `city` 與 `state` 欄內的值，並將其插入 `cities_usa` 資料表中的 `city` 和 `state` 欄：

```
INSERT INTO cities_usa (city,state)
SELECT city,state
FROM cities_world
      WHERE country='usa'
```

## INSERT INTO... VALUES

透過指定資料欄和值，將資料列插入現有的資料表。指定的資料欄和相關聯的資料類型必須精確地符合目標資料表中的資料欄和資料類型。

### Important

我們不建議使用 `VALUES` 插入資料行，因為 Athena 會為每個 `INSERT` 操作產生檔案。這可能會導致建立許多小型檔案，並降低資料表的查詢效能。若要識別 `INSERT` 查詢建立的檔案，請檢查資料資訊清單檔案。如需詳細資訊，請參閱 [使用查詢結果、近期查詢和輸出檔案](#)。

## 概要

```
INSERT INTO destination_table [(col1,col2,...)]
```

```
VALUES (col1value,col2value,...)[,  
        (col1value,col2value,...)][,  
        ...]
```

## 範例

在下列範例中，城市資料表有三個資料欄：id、city、state、state\_motto。id 資料欄的類型是 INT，而其他所有欄的類型都是 VARCHAR。

將單一系列插入 cities 資料表，並指定所有資料欄值：

```
INSERT INTO cities  
VALUES (1,'Lansing','MI','Si quaeris peninsulam amoenam circumspice')
```

在 cities 資料表中插入兩列：

```
INSERT INTO cities  
VALUES (1,'Lansing','MI','Si quaeris peninsulam amoenam circumspice'),  
        (3,'Boise','ID','Esto perpetua')
```

## DELETE

刪除 Apache Iceberg 資料表中的資料列。DELETE 是交易型的並且僅由 Apache Iceberg 資料表支援。

### 概要

若要刪除 Iceberg 資料表中的資料列，請使用下列語法。

```
DELETE FROM [db_name.]table_name [WHERE predicate]
```

如需詳細資訊和範例，請參閱 [更新 Iceberg 資料表資料](#) 中的 DELETE 一節。

## UPDATE

更新 Apache Iceberg 資料表中的資料列。UPDATE 是交易型的並且僅由 Apache Iceberg 資料表支援。

### 概要

若要更新 Iceberg 資料表中的資料列，請使用下列語法。

```
UPDATE [db_name.]table_name SET xx=yy[,...] [WHERE predicate]
```

如需詳細資訊和範例，請參閱 [更新 Iceberg 資料表資料](#) 中的 UPDATE 一節。

## 合併為

有條件地更新、刪除資料列或將其插入至 Apache Iceberg 資料表。單一陳述式可以組合更新、刪除和插入動作。

### Note

MERGE INTO 是交易型的並且僅由 Athena 引擎版本 3 中的 Apache Iceberg 資料表支援。

## 概要

若要有條件地更新、刪除或插入 Iceberg 資料表中的資料列，請使用下列語法。

```
MERGE INTO target_table [ [ AS ] target_alias ]  
USING { source_table | query } [ [ AS ] source_alias ]  
ON search_condition  
when_clause [...]
```

*when\_clause* 是下列其中一項：

```
WHEN MATCHED [ AND condition ]  
THEN DELETE
```

```
WHEN MATCHED [ AND condition ]  
THEN UPDATE SET ( column = expression [, ...] )
```

```
WHEN NOT MATCHED [ AND condition ]  
THEN INSERT (column_name[, column_name ...]) VALUES (expression, ...)
```

MERGE 支援具有不同 MATCHED 條件之任意數量的 WHEN 子句。條件子句會在由 MATCHED 狀態和符合條件選取的第一個 WHEN 子句中執行 DELETE、UPDATE 或 INSERT 運算。

針對每個來源資料列，會依序處理 WHEN 子句。只會執行第一個相符的 WHEN 子句。其餘子句會遭到忽略。當單一目標資料表資料列與多個來源資料列相符時，就會引起使用者錯誤。

如果來源資料列不與任何 WHEN 子句相符且沒有 WHEN NOT MATCHED 子句，則會忽略來源資料列。

在具有 UPDATE 運算的 WHEN 子句中，資料欄值表達式可以指代目標或來源的任何欄位。若是 NOT MATCHED，INSERT 表達式可以指代來源的任何欄位。

## 範例

下列範例會將第二個資料表中的資料列合併至第一個資料表 (如果第一個資料表中不存在資料列)。請注意，VALUES 子句中列出的資料欄必須加上來源資料表別名的字首。INSERT 子句中列出的目標資料欄不得使用字首。

```
MERGE INTO iceberg_table_sample as ice1
USING iceberg2_table_sample as ice2
ON ice1.col1 = ice2.col1
WHEN NOT MATCHED
THEN INSERT (col1)
      VALUES (ice2.col1)
```

如需更多 MERGE INTO 範例，請參閱 [更新 Iceberg 資料表資料](#)。

## OPTIMIZE

根據相關刪除檔案的大小和數量，將資料檔案重寫成更好的版面配置，進而最佳化 Apache Iceberg 資料表中的列。

### Note

OPTIMIZE 是交易型的並且僅由 Apache Iceberg 資料表支援。

## 語法

以下語法摘要說明最佳化處理 Iceberg 資料表的方式。

```
OPTIMIZE [db_name.]table_name REWRITE DATA USING BIN_PACK
[WHERE predicate]
```

### Note

WHERE 子句 ## 中只允許資料分割資料行。指定非分割區資料行將導致查詢失敗。

壓縮動作按重寫過程中掃描的資料量計費。REWRITE DATA 動作使用述詞來選擇包含對應資料列的檔案。如果檔案中的任何資料列與述詞相對應，則會選擇該檔案進行最佳化處理。因此，為了控制壓縮時包含的文件數量，請指定 WHERE 子句。

## 設定壓縮屬性

為了控制要選擇進行壓縮的檔案大小以及壓縮後產生的檔案大小，請使用資料表屬性參數。您可以使用 [ALTER TABLE SET PROPERTIES](#) 命令來設定相關的[資料表屬性](#)。

## 其他資源

### [最佳化處理 Iceberg 資料表](#)

## VACUUM

VACUUM 陳述式會移除不再需要的資料檔案，進而在 Apache Iceberg 資料表上執行資料表維護。

### Note

VACUUM 是交易型的並且僅由 Athena 引擎版本 3 中的 Apache Iceberg 資料表支援。

建議您在 Iceberg 資料表上執行 VACUUM 陳述式，以移除不再相關的資料檔案，並減少中繼資料大小和儲存消耗。請注意，由於 VACUUM 陳述式會對 Amazon S3 進行 API 呼叫，向 Amazon S3 發出相關請求需支付費用。

### Warning

如果您執行快照過期操作，則無法再回溯至過期的快照。

## 概要

若要移除 Iceberg 資料表中不再需要的資料檔案，請使用下列語法。

```
VACUUM [database_name.]target_table
```

若要在名稱開頭為底線 (例如，`_mytable`) 的資料表 VACUUM 上執行，請將資料表名稱括在反引號中，如下列範例所示。如果您在資料表名稱前面加上資料庫名稱，請勿將資料庫名稱括在反引號中。請注意，雙引號將無法代替反引號。

這種行為是特有的VACUUM。對於以底線開頭的資料表名稱，CREATE和INSERT INTO陳述式不需要反引號。

```
VACUUM `mytable`  
VACUUM my_database.`mytable`
```

另請注意，VACUUM希望冰山數據位於 Amazon S3 文件夾中，而不是在 Amazon S3 存儲桶中。例如，如果您的冰山資料位於s3://DOC-EXAMPLE-BUCKET/而不是，VACUUM陳述式會失敗s3://DOC-EXAMPLE-BUCKET/myicebergfolder/，並顯示錯誤訊息 GENERIC\_INAL\_ERROR：檔案系統位置中遺失路徑：。s3://DOC-EXAMPLE-BUCKET

## 執行的作業

VACUUM 執行以下操作：

- 移除早於 vacuum\_max\_snapshot\_age\_seconds 資料表屬性指定之時間的快照。依預設，此屬性已設為 432000 秒 (5 天)。
- 移除不在保留期間內且超過 vacuum\_min\_snapshots\_to\_keep 資料表屬性所指定數目的快照。預設為 1。

您可以在 CREATE TABLE 陳述式中指定這些資料表屬性。建立資料表之後，您就可以使用 [ALTER TABLE SET PROPERTIES](#) 陳述式來進行更新。

- 移除任何因為快照移除而無法取得的中繼資料和資料檔案。您可以設定 vacuum\_max\_metadata\_files\_to\_keep 資料表屬性，進而設定要保留的舊中繼資料檔案數目。預設值為 100。
- 移除早於 vacuum\_max\_snapshot\_age\_seconds 資料表屬性中指定之時間的孤立檔案。孤立檔案是資料表的資料目錄中不屬於資料表狀態的檔案。

如需有關在 Athena 中建立和管理 Apache Iceberg 資料表的詳細資訊，則請參閱 [建立 Iceberg 資料表](#) 和 [管理 Iceberg 資料表](#)。

## 在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE

EXPLAIN 陳述式會顯示指定 SQL 陳述式的邏輯或分散式執行計劃，或驗證 SQL 陳述式。您可以以文字格式或資料格式輸出結果，以便轉譯成圖形。

**Note**

您可以在 Athena 主控台中檢視查詢的邏輯和分散式計劃的圖形呈現，無需使用 EXPLAIN 語法。如需詳細資訊，請參閱 [檢視 SQL 查詢的執行計劃](#)。

EXPLAIN ANALYZE 陳述式會顯示指定 SQL 陳述式的分散式執行計劃，以及 SQL 查詢中每個操作的運算成本。您可以將結果輸出為文字或 JSON 格式。

**考量與限制**

在 Athena 的 EXPLAIN 和 EXPLAIN ANALYZE 陳述式有下列限制。

- EXPLAIN 查詢不會掃描任何資料，因此 Athena 不會為此收取費用。然而，由於 EXPLAIN 查詢會呼叫 AWS Glue 來擷取資料表中繼資料，如果呼叫超過 [Glue 免費方案限制](#) 的次數，則會從 Glue 收取費用。
- 因為已執行 EXPLAIN ANALYZE 查詢，它們會掃描資料，並且 Athena 會針對掃描的資料量收取費用。
- 在 Lake Formation 中定義的資料列或儲存格篩選資訊及查詢統計資料資訊未在 EXPLAIN 和 EXPLAIN ANALYZE 的輸出中顯示。

**EXPLAIN 語法**

```
EXPLAIN [ ( option [, ...]) ] statement
```

*option* 可為下列項目之一：

```
FORMAT { TEXT | GRAPHVIZ | JSON }  
TYPE { LOGICAL | DISTRIBUTED | VALIDATE | IO }
```

如果未指定 FORMAT 選項，則輸出預設為 TEXT 格式。IO 類型提供有關查詢讀取的資料表和結構描述的資訊。IO 僅在 Athena 引擎版本 2 中受支援，且只能以 JSON 格式傳回。

**EXPLAIN ANALYZE 語法**

除了包含在 EXPLAIN，EXPLAIN ANALYZE 輸出也包含指定查詢的執行時間統計數字，例如 CPU 用量、資料列輸入數目以及資料列輸出數目。

```
EXPLAIN ANALYZE [ ( option [, ...] ) ] statement
```

*option* 可為下列項目之一：

```
FORMAT { TEXT | JSON }
```

如果未指定 FORMAT 選項，則輸出預設為 TEXT 格式。因為 EXPLAIN ANALYZE 的所有查詢為 DISTRIBUTED，則 TYPE 選項不可為 EXPLAIN ANALYZE。

### 可為下列其中之一：

```
SELECT  
CREATE TABLE AS SELECT  
INSERT  
UNLOAD
```

## EXPLAIN 範例

下列針對 EXPLAIN 的範例從較簡單的內容進展到較複雜的內容。

### EXPLAIN 範例 1：使用 EXPLAIN 陳述式以文字格式顯示查詢計劃

在下列範例中，EXPLAIN 會顯示對 Elastic Load Balancing 日誌的 SELECT 查詢的執行計劃。格式預設為文字輸出。

```
EXPLAIN  
SELECT  
    request_timestamp,  
    elb_name,  
    request_ip  
FROM sampledb.elb_logs;
```

## 結果

```
- Output[request_timestamp, elb_name, request_ip] => [[request_timestamp, elb_name,  
request_ip]]  
  - RemoteExchange[GATHER] => [[request_timestamp, elb_name, request_ip]]  
    - TableScan[awsdatacatalog:HiveTableHandle{schemaName=sampled,  
tableName=elb_logs,
```

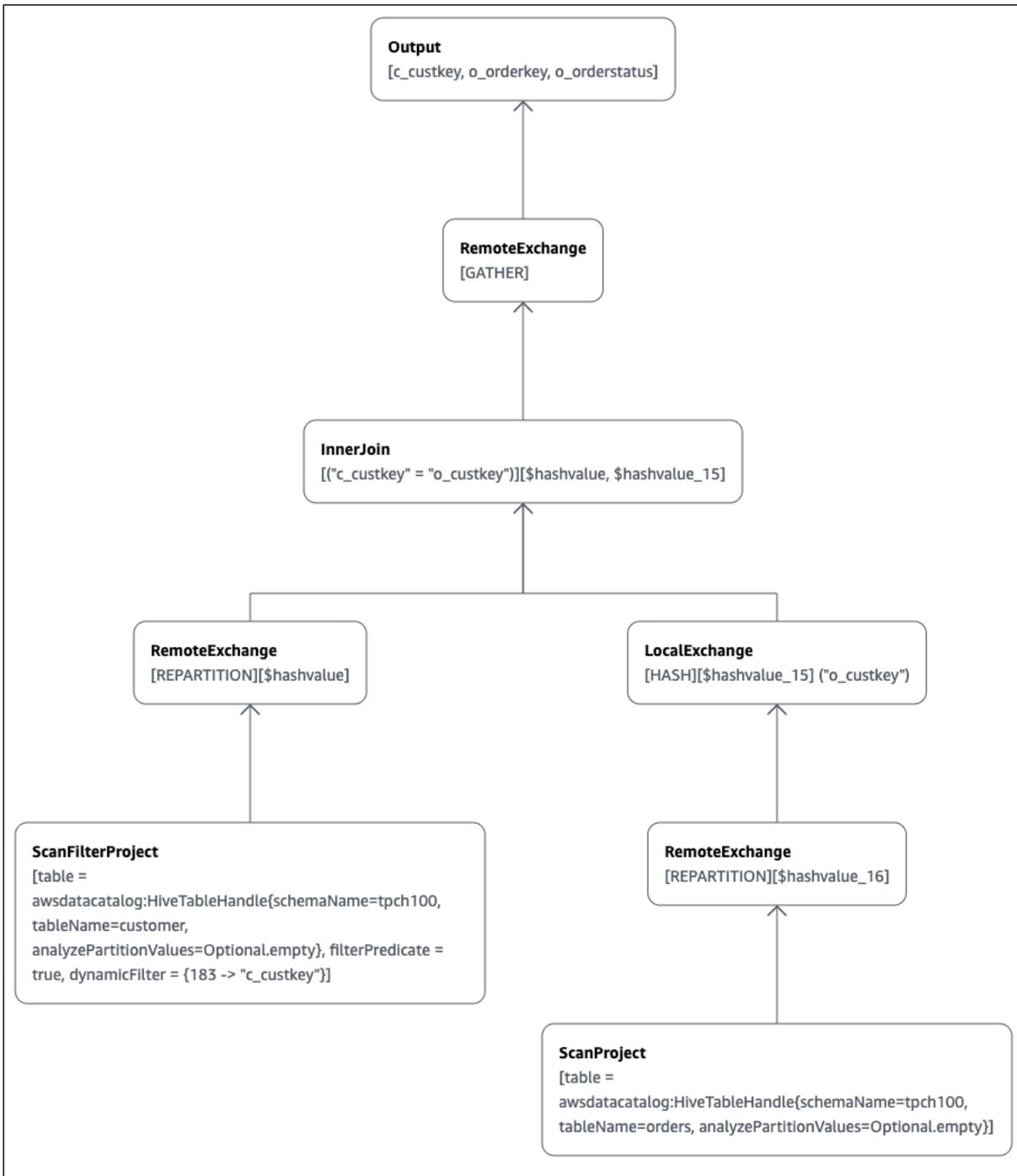
```
analyzePartitionValues=Optional.empty}} => [[request_timestamp, elb_name, request_ip]]
      LAYOUT: sampled.elb_logs
      request_ip := request_ip:string:2:REGULAR
      request_timestamp := request_timestamp:string:0:REGULAR
      elb_name := elb_name:string:1:REGULAR
```

## EXPLAIN 範例 2：繪製查詢計劃的圖形

您可以使用 Athena 主控台來繪製查詢計劃圖。在查詢編輯器中輸入類似以下的 SELECT 陳述式，然後選擇 EXPLAIN。

```
SELECT
  c.c_custkey,
  o.o_orderkey,
  o.o_orderstatus
FROM tpch100.customer c
JOIN tpch100.orders o
  ON c.c_custkey = o.o_custkey
```

Athena 查詢編輯器的 Explain (解釋) 頁面隨即開啟，並顯示查詢的分散式計劃和邏輯計劃。下圖顯示範例的邏輯計劃。



### ⚠ Important

目前，即使 Athena 確實將這些篩選條件套用至您的查詢，有些分割區篩選條件可能不會顯示在巢狀運算子樹狀結構圖形中。若要驗證此等篩選條件的效果，請在查詢上執行 `EXPLAIN` 或 `EXPLAIN ANALYZE` 並查看結果。

如需在 Athena 主控台中使用查詢計劃圖形功能的詳細資訊，請參閱 [檢視 SQL 查詢的執行計劃](#)。

### EXPLAIN 範例 3：使用 EXPLAIN 陳述式來驗證分割區剔除

當您對分割的金鑰使用篩選述詞來查詢分割的資料表時，查詢引擎會將述詞套用至分割的金鑰，以減少讀取的資料量。

以下範例使用 EXPLAIN 查詢，對分割資料表上的 SELECT 查詢驗證分割區剔除。首先，CREATE TABLE 陳述式會建立 tpch100.orders\_partitioned 資料表。該資料表在資料欄 o\_orderdate 上分割。

```
CREATE TABLE `tpch100.orders_partitioned` (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_totalprice` double,  
  `o_orderpriority` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_comment` string)  
PARTITIONED BY (  
  `o_orderdate` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/<your_directory_path>/'
```

tpch100.orders\_partitioned 資料表在 o\_orderdate 上有多個分割區，如 SHOW PARTITIONS 命令所示。

```
SHOW PARTITIONS tpch100.orders_partitioned;  
  
o_orderdate=1994  
o_orderdate=2015  
o_orderdate=1998  
o_orderdate=1995  
o_orderdate=1993  
o_orderdate=1997
```

```
o_orderdate=1992
o_orderdate=1996
```

下列 EXPLAIN 查詢會對指定 SELECT 陳述式驗證分割區別除。

```
EXPLAIN
SELECT
  o_orderkey,
  o_custkey,
  o_orderdate
FROM tpch100.orders_partitioned
WHERE o_orderdate = '1995'
```

## 結果

```
Query Plan
- Output[o_orderkey, o_custkey, o_orderdate] => [[o_orderkey, o_custkey, o_orderdate]]
  - RemoteExchange[GATHER] => [[o_orderkey, o_custkey, o_orderdate]]
    - TableScan[awsdatacatalog:HiveTableHandle{schemaName=tpch100,
      tableName=orders_partitioned,
      analyzePartitionValues=Optional.empty}] => [[o_orderkey, o_custkey, o_orderdate]]
      LAYOUT: tpch100.orders_partitioned
      o_orderdate := o_orderdate:string:-1:PARTITION_KEY
      :: [[1995]]
      o_custkey := o_custkey:int:1:REGULAR
      o_orderkey := o_orderkey:int:0:REGULAR
```

結果中的粗體文字顯示述詞 `o_orderdate = '1995'` 已套用於 `PARTITION_KEY`。

## EXPLAIN 範例 4：使用 EXPLAIN 查詢來檢查聯結順序和聯結類型

下列 EXPLAIN 查詢會檢查 SELECT 陳述式的聯結順序和聯結類型。使用這樣的查詢來檢查查詢記憶體的使用情況，以便減少取得 `EXCEEDED_LOCAL_MEMORY_LIMIT` 錯誤的可能性。

```
EXPLAIN (TYPE DISTRIBUTED)
SELECT
  c.c_custkey,
  o.o_orderkey,
  o.o_orderstatus
FROM tpch100.customer c
JOIN tpch100.orders o
```

```
ON c.c_custkey = o.o_custkey
WHERE c.c_custkey = 123
```

## 結果

### Query Plan

#### Fragment 0 [SINGLE]

```
Output layout: [c_custkey, o_orderkey, o_orderstatus]
Output partitioning: SINGLE []
Stage Execution Strategy: UNGROUPED_EXECUTION
- Output[c_custkey, o_orderkey, o_orderstatus] => [[c_custkey, o_orderkey,
o_orderstatus]]
  - RemoteSource[1] => [[c_custkey, o_orderstatus, o_orderkey]]
```

#### Fragment 1 [SOURCE]

```
Output layout: [c_custkey, o_orderstatus, o_orderkey]
Output partitioning: SINGLE []
Stage Execution Strategy: UNGROUPED_EXECUTION
- CrossJoin => [[c_custkey, o_orderstatus, o_orderkey]]
  Distribution: REPLICATED
  - ScanFilter[table = awsdatacatalog:HiveTableHandle{schemaName=tpch100,
tableName=customer, analyzePartitionValues=Optional.empty}, grouped = false,
filterPredicate = ("c_custkey" = 123)] => [[c_custkey]]
    LAYOUT: tpch100.customer
    c_custkey := c_custkey:int:0:REGULAR
  - LocalExchange[SINGLE] () => [[o_orderstatus, o_orderkey]]
    - RemoteSource[2] => [[o_orderstatus, o_orderkey]]
```

#### Fragment 2 [SOURCE]

```
Output layout: [o_orderstatus, o_orderkey]
Output partitioning: BROADCAST []
Stage Execution Strategy: UNGROUPED_EXECUTION
- ScanFilterProject[table = awsdatacatalog:HiveTableHandle{schemaName=tpch100,
tableName=orders, analyzePartitionValues=Optional.empty}, grouped = false,
filterPredicate = ("o_custkey" = 123)] => [[o_orderstatus, o_orderkey]]
  LAYOUT: tpch100.orders
  o_orderstatus := o_orderstatus:string:2:REGULAR
  o_custkey := o_custkey:int:1:REGULAR
  o_orderkey := o_orderkey:int:0:REGULAR
```

查詢範例在交叉聯結中得到最佳化處理，以獲得更好的性能。結果顯示 `tpch100.orders` 將作為 BROADCAST 分佈類型進行分配。這意味著 `tpch100.orders` 資料表將分配到執行聯結操作的所有節

點。BROADCAST 分佈類型會要求 `tpch100.orders` 資料表的所有篩選結果納入執行聯結操作的每個節點的記憶體。

不過，`tpch100.customer` 資料表比 `tpch100.orders` 小。由於 `tpch100.customer` 需要更少的記憶體，您可以將查詢重寫為 BROADCAST `tpch100.customer`，而非 `tpch100.orders`。這可減少查詢取得 EXCEEDED\_LOCAL\_MEMORY\_LIMIT 錯誤的可能性。此策略假設下列要點：

- 在 `tpch100.customer` 資料表中 `tpch100.customer.c_custkey` 為唯一。
- `tpch100.customer` 和之間存在 one-to-many 對應關係 `tpch100.orders`。

下列範例顯示重寫的查詢。

```
SELECT
  c.c_custkey,
  o.o_orderkey,
  o.o_orderstatus
FROM tpch100.orders o
JOIN tpch100.customer c -- the filtered results of tpch100.customer are distributed to
  all nodes.
  ON c.c_custkey = o.o_custkey
WHERE c.c_custkey = 123
```

### EXPLAIN 範例 5：使用 EXPLAIN 查詢移除沒有效果的述詞

您可以使用 EXPLAIN 查詢來檢查篩選述詞的有效性。您可以使用結果移除沒有效果的述詞，如下列範例所示。

```
EXPLAIN
  SELECT
    c.c_name
  FROM tpch100.customer c
  WHERE c.c_custkey = CAST(RANDOM() * 1000 AS INT)
  AND c.c_custkey BETWEEN 1000 AND 2000
  AND c.c_custkey = 1500
```

### 結果

```
Query Plan
- Output[c_name] => [[c_name]]
  - RemoteExchange[GATHER] => [[c_name]]
    - ScanFilterProject[table =
```

```
awsdatacatalog:HiveTableHandle{schemaName=tpch100,
tableName=customer, analyzePartitionValues=Optional.empty},
filterPredicate = (("c_custkey" = 1500) AND ("c_custkey" =
CAST(("random"() * 1E3) AS int)))) => [[c_name]]
      LAYOUT: tpch100.customer
      c_custkey := c_custkey:int:0:REGULAR
      c_name := c_name:string:1:REGULAR
```

結果中的 `filterPredicate` 顯示優化器將原始三個述詞合併為兩個述詞，並更改了應用程式的順序。

```
filterPredicate = (("c_custkey" = 1500) AND ("c_custkey" = CAST(("random"() * 1E3) AS
int)))
```

因為結果顯示述詞 `AND c.c_custkey BETWEEN 1000 AND 2000` 沒有任何效果，您可以在不更改查詢結果的情況下移除此述詞。

如需有關 EXPLAIN 查詢結果中所使用術語的詳細資訊，請參閱 [了解 Athena EXPLAIN 陳述式結果](#)。

## EXPLAIN ANALYZE 範例

下列範例將顯示範例 EXPLAIN ANALYZE 查詢和輸出。

### EXPLAIN ANALYZE 範例 1：使用 EXPLAIN ANALYZE 以文字格式顯示查詢計劃和運算成本

在下列範例中，EXPLAIN ANALYZE 顯示 CloudFront 記錄檔 SELECT 查詢的執行計畫和計算成本。格式預設為文字輸出。

```
EXPLAIN ANALYZE SELECT FROM cloudfront_logs LIMIT 10
```

## 結果

```
Fragment 1
  CPU: 24.60ms, Input: 10 rows (1.48kB); per task: std.dev.: 0.00, Output: 10 rows
(1.48kB)
  Output layout: [date, time, location, bytes, requestip, method, host, uri, status,
referrer,\
  os, browser, browserversion]
Limit[10] => [[date, time, location, bytes, requestip, method, host, uri, status,
referrer, os,\
  browser, browserversion]]
      CPU: 1.00ms (0.03%), Output: 10 rows (1.48kB)
      Input avg.: 10.00 rows, Input std.dev.: 0.00%
```

```

LocalExchange[SINGLE] () => [[date, time, location, bytes, requestip, method, host,
uri, status, referrer, os,\
browser, browserversion]]
      CPU: 0.00ns (0.00%), Output: 10 rows (1.48kB)
      Input avg.: 0.63 rows, Input std.dev.: 387.30%
RemoteSource[2] => [[date, time, location, bytes, requestip, method, host, uri, status,
referrer, os,\
browser, browserversion]]
      CPU: 1.00ms (0.03%), Output: 10 rows (1.48kB)
      Input avg.: 0.63 rows, Input std.dev.: 387.30%

Fragment 2
      CPU: 3.83s, Input: 998 rows (147.21kB); per task: std.dev.: 0.00, Output: 20 rows
(2.95kB)
      Output layout: [date, time, location, bytes, requestip, method, host, uri, status,
referrer, os,\
      browser, browserversion]
LimitPartial[10] => [[date, time, location, bytes, requestip, method, host, uri,
status, referrer, os,\
browser, browserversion]]
      CPU: 5.00ms (0.13%), Output: 20 rows (2.95kB)
      Input avg.: 166.33 rows, Input std.dev.: 141.42%
TableScan[awsdatacatalog:HiveTableHandle{schemaName=default, tableName=cloudfront_logs,
\
analyzePartitionValues=Optional.empty},
grouped = false] => [[date, time, location, bytes, requestip, method, host, uri, st
CPU: 3.82s (99.82%), Output: 998 rows (147.21kB)
Input avg.: 166.33 rows, Input std.dev.: 141.42%
LAYOUT: default.cloudfront_logs
date := date:date:0:REGULAR
referrer := referrer:string:9:REGULAR
os := os:string:10:REGULAR
method := method:string:5:REGULAR
bytes := bytes:int:3:REGULAR
browser := browser:string:11:REGULAR
host := host:string:6:REGULAR
requestip := requestip:string:4:REGULAR
location := location:string:2:REGULAR
time := time:string:1:REGULAR
uri := uri:string:7:REGULAR
browserversion := browserversion:string:12:REGULAR
status := status:int:8:REGULAR

```

## EXPLAIN ANALYZE 範例 2：使用 EXPLAIN ANALYZE 以 JSON 格式顯示查詢計畫

下列範例顯示在 CloudFront 記錄檔上 SELECT 查詢的執行計畫和計算成本。此範例指定 JSON 作為輸出格式。

```
EXPLAIN ANALYZE (FORMAT JSON) SELECT * FROM cloudfront_logs LIMIT 10
```

### 結果

```
{
  "fragments": [{
    "id": "1",

    "stageStats": {
      "totalCpuTime": "3.31ms",
      "inputRows": "10 rows",
      "inputDataSize": "1514B",
      "stdDevInputRows": "0.00",
      "outputRows": "10 rows",
      "outputDataSize": "1514B"
    },
    "outputLayout": "date, time, location, bytes, requestip, method, host,\
      uri, status, referrer, os, browser, browserversion",

    "logicalPlan": {
      "1": [{
        "name": "Limit",
        "identifier": "[10]",
        "outputs": ["date", "time", "location", "bytes", "requestip", "method",
"host",\
      "uri", "status", "referrer", "os", "browser", "browserversion"],
        "details": "",
        "distributedNodeStats": {
          "nodeCpuTime": "0.00ns",
          "nodeOutputRows": 10,
          "nodeOutputDataSize": "1514B",
          "operatorInputRowsStats": [{
            "nodeInputRows": 10.0,
            "nodeInputRowsStdDev": 0.0
          }]
        }
      ]},
      "children": [{
        "name": "LocalExchange",
```

```

        "identifier": "[SINGLE] ()",
        "outputs": ["date", "time", "location", "bytes", "requestip",
"method", "host",\
            "uri", "status", "referrer", "os", "browser", "browserversion"],
        "details": "",
        "distributedNodeStats": {
            "nodeCpuTime": "0.00ns",
            "nodeOutputRows": 10,
            "nodeOutputDataSize": "1514B",
            "operatorInputRowsStats": [{
                "nodeInputRows": 0.625,
                "nodeInputRowsStdDev": 387.2983346207417
            }]
        },
        "children": [{
            "name": "RemoteSource",
            "identifier": "[2]",
            "outputs": ["date", "time", "location", "bytes", "requestip",
"method", "host",\
                "uri", "status", "referrer", "os", "browser",
"browserversion"],
            "details": "",
            "distributedNodeStats": {
                "nodeCpuTime": "0.00ns",
                "nodeOutputRows": 10,
                "nodeOutputDataSize": "1514B",
                "operatorInputRowsStats": [{
                    "nodeInputRows": 0.625,
                    "nodeInputRowsStdDev": 387.2983346207417
                }]
            },
            "children": []
        }]
    }]
}
}, {
    "id": "2",

    "stageStats": {
        "totalCpuTime": "1.62s",
        "inputRows": "500 rows",
        "inputDataSize": "75564B",
        "stdDevInputRows": "0.00",

```

```

        "outputRows": "10 rows",
        "outputDataSize": "1514B"
    },
    "outputLayout": "date, time, location, bytes, requestip, method, host, uri,
status,\
referrer, os, browser, browserversion",

    "logicalPlan": {
        "1": [{
            "name": "LimitPartial",
            "identifier": "[10]",
            "outputs": ["date", "time", "location", "bytes", "requestip", "method",
"host", "uri",\
                "status", "referrer", "os", "browser", "browserversion"],
            "details": "",
            "distributedNodeStats": {
                "nodeCpuTime": "0.00ns",
                "nodeOutputRows": 10,
                "nodeOutputDataSize": "1514B",
                "operatorInputRowsStats": [{
                    "nodeInputRows": 83.33333333333333,
                    "nodeInputRowsStdDev": 223.60679774997897
                }]
            },
            "children": [{
                "name": "TableScan",
                "identifier": "[awsdatacatalog:HiveTableHandle{schemaName=default,\
                tableName=cloudfront_logs,
analyzePartitionValues=Optional.empty},\
                grouped = false]",
                "outputs": ["date", "time", "location", "bytes", "requestip",
"method", "host", "uri",\
                    "status", "referrer", "os", "browser", "browserversion"],
                "details": "LAYOUT: default.cloudfront_logs\ndate :=
date:date:0:REGULAR\nreferrer :=\
                    referrer: string:9:REGULAR\nos := os:string:10:REGULAR
\nmethod := method:string:5:\
                    REGULAR\nbytes := bytes:int:3:REGULAR\nbrowser :=
browser:string:11:REGULAR\nhost :=\
                    host:string:6:REGULAR\nrequestip := requestip:string:4:REGULAR
\nlocation :=\
                    location:string:2:REGULAR\ntime := time:string:1: REGULAR
\nuri := uri:string:7:\

```

```

REGULAR\nbrowserversion := browserversion:string:12:REGULAR
\nstatus :=\
    status:int:8:REGULAR\n",
    "distributedNodeStats": {
        "nodeCpuTime": "1.62s",
        "nodeOutputRows": 500,
        "nodeOutputDataSize": "75564B",
        "operatorInputRowsStats": [{
            "nodeInputRows": 83.33333333333333,
            "nodeInputRowsStdDev": 223.60679774997897
        }]
    },
    "children": []
}]
}

```

## 其他資源

如需其他資訊，請參閱以下資源。

- [了解 Athena EXPLAIN 陳述式結果](#)
- [檢視 SQL 查詢的執行計劃](#)
- [檢視已完成查詢的統計資料和執行詳細資訊](#)
- Trino [EXPLAIN](#) 文件
- Trino [EXPLAIN ANALYZE](#) 文件
- AWS 大數據部落格中的 [使用 Amazon Athena 中的 EXPLAIN 和 EXPLAIN ANALYZE 優化聯合查詢效能](#)。

## 了解 Athena EXPLAIN 陳述式結果

本主題提供 Athena EXPLAIN 陳述式結果中使用的操作術語的簡要指南。

### EXPLAIN 陳述式輸出類型

EXPLAIN 陳述式輸出可為以下兩種類型之一：

- **邏輯計劃** – 顯示 SQL 引擎用來執行陳述式的邏輯計劃。此選項的語法為 EXPLAIN 或 EXPLAIN (TYPE LOGICAL)。

- 分散式計劃 – 顯示分散式環境中的執行計劃。輸出顯示片段，這是處理階段。每個計劃片段會由一或多個節點處理。資料可以在處理片段的節點之間進行交換。此選項的語法為 EXPLAIN (TYPE DISTRIBUTED)。

在分散式計劃的輸出中，用 Fragment *number* [*fragment\_type*] 表示片段 (處理階段)，其中 *number* 是一個以零為起始的整數，以及 *fragment\_type* 指定片段如何由節點執行。片段類型提供資料交換配置的深入資訊，如下表所述。

#### 分散式計劃片段類型

片段類型	描述
SINGLE	片段在單個節點上執行。
HASH	該片段在固定數量的節點上執行。輸入資料使用雜湊函數分佈。
ROUND_ROB IN	該片段在固定數量的節點上執行。輸入資料以循環配置資源的方式分配。
BROADCAST	該片段在固定數量的節點上執行。輸入資料經廣播到所有節點。
SOURCE	該片段在存取輸入拆分的節點上執行。

## Exchange

交換相關術語描述如何在工作節點之間交換資料。傳輸可以是本機或遠端。

### LocalExchange [##類型]

針對查詢的不同階段，在工作節點內本機傳輸資料。*exchange\_type* 的值可以是邏輯或分散式交換類型其中一種，如本節後文所述。

### RemoteExchange [##類型]

針對查詢的不同階段，在工作節點之間傳輸資料。*exchange\_type* 的值可以是邏輯或分散式交換類型其中一種，如本節後文所述。

## 邏輯 Exchange 類型

下列交換類型說明邏輯計劃的交換階段期間所採取的動作。

- **GATHER** – 單一工作節點會收集所有其他工作節點的輸出。例如，選取查詢的最後階段會收集所有節點的結果，並將結果寫入 Amazon S3。
- **REPARTITION** – 根據套用至下一個運算子所需的分割結構描述，將資料行資料傳送至特定工作者。
- **REPLICATE** – 將資料行資料複製到所有工作者。

## 分散式交換類型

下列交換類型指出在分散式計劃中的節點之間交換資料時的資料配置。

- **HASH** – 交換會使用雜湊函數將資料分配到多個目的地。
- **SINGLE** – 交換會將資料分配到單一目的地。

## 掃描

下列術語描述查詢期間如何掃描資料。

### TableScan

掃描來自 Amazon S3 或 Apache Hive 連接器的資料表來源資料，並套用從篩選述詞產生的分割區剔除。

### ScanFilter

掃描來自 Amazon S3 或 Hive 連接器的資料表來源資料，並套用從篩選述詞產生的分割區剔除，以及從沒有透過分割區剔除套用的其他篩選述詞產生的分割區剔除。

### ScanFilterProject

首先，掃描來自 Amazon S3 或 Hive 連接器的資料表來源資料，並套用從篩選述詞產生的分割區剔除，以及從沒有透過分割區剔除套用的其他篩選述詞產生的分割區剔除。然後，將輸出資料的記憶體配置修改為新的投影，以改善後續階段的效能。

## Join

聯結兩個資料表之間的資料。聯結可以依聯結類型和分佈類型來分類。

### 聯結類型

聯結類型定義聯結操作發生的方式。

**CrossJoin**— 產生連接的兩個表的笛卡爾乘積。

**InnerJoin**— 選取兩個表格中具有相符值的記錄。

**LeftJoin**— 從左側表格中選取所有記錄，並從右側表格中選取相符的記錄。如果沒有發生匹配，右側的結果是 NULL。

**RightJoin**— 從右側表格中選取所有記錄，並從左側表格中選取相符的記錄。如果沒有發生匹配，左側的結果是 NULL。

**FullJoin**— 選取左側或右側表格記錄中相符項的所有記錄。聯結資料表包含來自兩個資料表的所有記錄，並在兩側不符的地方以 NULL 填充。

### Note

基於效能考量，查詢引擎可以將聯結查詢重新寫入不同的聯結類型，以產生相同的結果。例如，在一個資料表上具有述詞的內部聯結查詢可以重寫入 `CrossJoin`。這會將述詞推送到資料表的掃描階段，以便掃描較少的資料。

## 聯結分佈類型

分佈類型定義執行聯結操作時，如何在工作節點之間交換資料。

**已分割** – 左側和右側資料表會在所有工作節點間進行雜湊分割。已分割分佈會在每個節點消耗較少的記憶體。已分割分佈可以比複寫的聯結慢得多。當您聯結兩個大資料表時適合使用已分割聯結。

**已複寫** – 一個資料表會在所有工作節點之間進行雜湊分割，另一個資料表會複寫到所有工作節點，以執行聯結操作。複寫的分佈可以比已分割聯結快得多，但它會在每個工作節點中耗用更多的記憶體。如果複寫的資料表太大，Worker 節點可能會 `out-of-memory` 發生錯誤。當其中一個聯結的資料表很小時，就適合使用複寫的聯結。

## PREPARE

使用名稱 `statement_name` 建立 SQL 陳述式，以供之後執行。陳述式可以包含以問號表示的參數。若要提供值給參數並執行預備陳述式，請使用 [EXECUTE](#)。

### 概要

```
PREPARE statement_name FROM statement
```

下表描述了參數。

參數	描述
statement_name	要預備之陳述式的名稱。該名稱在工作群組內必須是唯一的。
statement	SELECT、CTAS 或 INSERT INTO 查詢。

 Note

工作群組中預備陳述式的數量上限為 1000。

## 範例

以下範例準備一個不含參數的選擇查詢。

```
PREPARE my_select1 FROM
SELECT * FROM nation
```

以下範例準備一個包含參數的選擇查詢。productid 和 quantity 的值將由 EXECUTE 陳述式的 USING 子句提供：

```
PREPARE my_select2 FROM
SELECT order FROM orders WHERE productid = ? and quantity < ?
```

以下範例準備一個插入查詢。

```
PREPARE my_insert FROM
INSERT INTO cities_usa (city, state)
SELECT city, state
FROM cities_world
WHERE country = ?
```

## 其他資源

[使用預備陳述式查詢](#)

[EXECUTE](#)

[DEALLOCATE PREPARE](#)

## [INSERT INTO](#)

### EXECUTE

執行名為 `statement_name` 的預備陳述式。預備陳述式中問號的參數值定義在以逗號分隔的清單的 `USING` 子句中。若要建立預備陳述式，使用 [PREPARE](#)。

#### 概要

```
EXECUTE statement_name [ USING parameter1[, parameter2, ... ] ]
```

#### 範例

以下範例準備並執行不含任何參數的查詢。

```
PREPARE my_select1 FROM  
SELECT name FROM nation  
EXECUTE my_select1
```

以下範例準備並執行含有一個參數的查詢。

```
PREPARE my_select2 FROM  
SELECT * FROM "my_database"."my_table" WHERE year = ?  
EXECUTE my_select2 USING 2012
```

這相當於：

```
SELECT * FROM "my_database"."my_table" WHERE year = 2012
```

以下範例準備並執行含有兩個參數的查詢。

```
PREPARE my_select3 FROM  
SELECT order FROM orders WHERE productid = ? and quantity < ?  
EXECUTE my_select3 USING 346078, 12
```

#### 其他資源

#### [使用預備陳述式查詢](#)

#### [PREPARE](#)

#### [INSERT INTO](#)

## DEALLOCATE PREPARE

從目前工作群組的預備陳述式中移除具有指定名稱的預備陳述式。

### 概要

```
DEALLOCATE PREPARE statement_name
```

### 範例

下列範例會從目前的工作群組中移除 my\_select1 預備陳述式。

```
DEALLOCATE PREPARE my_select1
```

### 其他資源

#### [使用預備陳述式查詢](#)

#### [PREPARE](#)

## UNLOAD

從 SELECT 陳述式寫入查詢結果至指定的資料格式。針對 UNLOAD 支援的格式包括 Apache Parquet、ORC、Apache Avro 和 JSON。CSV 是 Athena SELECT 命令唯一支援的輸出格式，但您可以使用支援多種輸出格式的 UNLOAD 命令，將 SELECT 查詢括起來，並將其輸出重新寫入為其中一種 UNLOAD 支援的格式。

雖然您可以使用 CTAS 陳述式以 CSV 以外的格式輸出資料，但這些陳述式也需要在 Athena 中建立資料表。UNLOAD 陳述式在您想以非 CSV 格式輸出 SELECT 查詢結果，但不需要相關聯的資料表時十分有用。例如，下游應用程式可能需要 SELECT 查詢結果使用 JSON 格式，而且如果您打算對其他分析使用 SELECT 查詢結果，則 Parquet 或 ORC 的效能可能優於 CSV。

### 考量與限制

當您在 Athena 中使用 UNLOAD 陳述式時，請謹記以下幾點：

- 沒有檔案的全域排序 – UNLOAD 結果會同時寫入多個檔案。如果 UNLOAD 陳述式中的 SELECT 查詢會指定排序順序，則每個檔案的內容會依排序順序排列，但檔案並不會彼此排序。
- 未刪除孤立資料 – 在發生故障的情況下，Athena 不會嘗試刪除孤立的資料。此行為同於 CTAS 和 INSERT INTO 陳述式。
- 分割區數量上限 – 可搭配 UNLOAD 使用的分割區數量上限為 100。

- 中繼資料和資訊清單檔案 – Athena 會為每個 UNLOAD 查詢產生中繼資料檔案和資訊清單檔案。資訊清單會追蹤查詢寫入的檔案。這兩個檔案都會儲存到 Amazon S3 中的 Athena 查詢結果位置。如需詳細資訊，請參閱[識別查詢輸出檔案](#)。
- 加密 – UNLOAD 輸出檔案會根據用於 Amazon S3 的加密組態進行加密。若要設定加密設定以加密 UNLOAD 結果，您可以使用 [EncryptionConfiguration API](#)。
- 預備陳述式 – UNLOAD 可以與預備的陳述式一起使用。如需有關 Athena 中的預備陳述式的資訊，請參閱[使用參數化查詢](#)。
- Service Quotas – UNLOAD 會使用 DML 查詢配額。如需有關配額的資訊，請參閱 [Service Quotas](#)。
- 預期的儲存貯體擁有者 – 預期的儲存貯體擁有者設定不適用於 UNLOAD 查詢中指定的目的地 Amazon S3 位置。預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Amazon S3 輸出位置。如需詳細資訊，請參閱[使用 Athena 主控台指定查詢結果位置](#)。

## 語法

UNLOAD 陳述式使用下列語法。

```
UNLOAD (SELECT col_name [, ...] FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/my_folder/'
WITH ( property_name = 'expression' [, ...] )
```

除了寫入分區時，TO 目的地必須在 Amazon S3 中指定沒有資料的位置。在 UNLOAD 查詢寫入指定位置之前，它會驗證儲存貯體位置是否為空。由於 UNLOAD 在指定位置已經有資料時不會將資料寫入該位置，UNLOAD 不會覆寫現有資料。若要重複使用儲存貯體位置作為 UNLOAD 的目標，請刪除儲存貯體位置中的資料，然後再次執行查詢。

請注意，UNLOAD 寫入分區時，這種行為是不同的。如果您多次執行具有相同 SELECT 陳述式、相同 TO 位置和相同分區的相同 UNLOAD 查詢，則每個 UNLOAD 查詢都會在指定的位置和分區將資料卸載到 Amazon S3。

## 參數

*property\_name* 可能的值如下。

格式 = '*file\_format*'

必要。指定輸出的檔案格式。*file\_format* 的可能值為 ORC、PARQUET、AVRO、JSON 或 TEXTFILE。

**壓縮 = 'compression\_format'**

選用。此選項僅適用於 ORC 和 Parquet 格式。ORC 的預設值為 zlib，而 Parquet 的預設值則為 gzip。如需有關支援的壓縮格式的資訊，請參閱 [Athena 壓縮支援](#)。

**Note**

此選項不適用於 AVRO 格式。Athena 對 JSON 和 TEXTFILE 格式使用 gzip。

**compression\_level = compression\_level**

選用。用於 ZSTD 壓縮的壓縮級別。此屬性僅適用於 ZSTD 壓縮。如需詳細資訊，請參閱 [在 Athena 中使用 ZSTD 壓縮級別](#)。

**field\_delimiter = 'delimiter'**

選用。指定 CSV、TSV 和其他文字格式檔案的單一字元欄位分隔符號。以下範例指定逗點分隔符號。

```
WITH (field_delimiter = ',')
```

目前不支援多字元欄位分隔符號。如果您未指定欄位分隔符號，系統會使用八進位制字元 \001 (^A)。

**partitioned\_by = ARRAY[ col\_name[,...] ]**

選用。資料欄的陣列清單，輸出依這些資料欄進行分割。

**Note**

在 SELECT 陳列式中，請確保已分割資料欄的名稱位於資料欄清單中的最後。

**範例**

下列範例會使用 JSON 格式，將 SELECT 查詢的輸出寫入 Amazon S3 位置 s3://DOC-EXAMPLE-BUCKET/unload\_test\_1/。

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/unload_test_1/'
WITH (format = 'JSON')
```

下列範例會採用 Snappy 壓縮方式以 Parquet 格式寫入 SELECT 查詢的輸出。

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/'
WITH (format = 'PARQUET',compression = 'SNAPPY')
```

下列範例會以文字格式寫入四個資料欄，且輸出依最後一個資料欄分割。

```
UNLOAD (SELECT name1, address1, comment1, key1 FROM table1)
TO 's3://DOC-EXAMPLE-BUCKET/ partitioned/'
WITH (format = 'TEXTFILE', partitioned_by = ARRAY['key1'])
```

下列範例會使用 Parquet 檔案格式、ZSTD 壓縮和 ZSTD 壓縮級別 4，將查詢結果卸載到指定的位置。

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/'
WITH (format = 'PARQUET', compression = 'ZSTD', compression_level = 4)
```

## 其他資源

- AWS 大數據部落格中的[使用 Amazon Athena UNLOAD 特徵簡化 ETL 和 ML 管道](#)。

## Amazon Athena 中的函數

如需有關 Athena 引擎版本之間函數的變更，請參閱 [Athena 引擎版本參考](#)。如需可與 AT TIME ZONE 運算子搭配使用的時區清單，請參閱[支援的時區](#)。

### Athena 引擎版本 3

Athena 引擎版本 3 中的函數以 Trino 為基礎。如需有關 Trino 函數、運算子和表達式的資訊，請參閱 Trino 文件中的 [Functions and operators](#) (函數和運算子)，以及以下具體子節。

- [Aggregate](#)
- [Array](#) (陣列)
- [: 二進位](#)
- [Bitwise](#) (位元)
- [Color](#) (顏色)
- [Comparison](#) (比較)

- [有條件](#)
- [Conversion](#) (轉換)
- [日期和時間](#)
- [Decimal](#) (小數)
- [Geospatial](#) (地理空間)
- [HyperLogLog](#)
- [IP Address](#) (IP 地址)
- [JSON](#)
- [Lambda](#)
- [Logical](#) (邏輯)
- [Machine learning](#) (機器學習)
- [Map](#)
- [Math](#) (數學)
- [Quantile digest](#) (分位數摘要)
- [Regular expression](#) (常規表達式)
- [Session](#) (工作階段)
- [Set Digest](#)
- [: 字串](#)
- [資料表](#)
- [Teradata](#)
- [T-Digest](#)
- [URL](#)
- [UUID](#)
- [視窗](#)

調用主要 ( ) 函數

該 `invoker_principal` 功能是 Athena 引擎版本 3 獨有的，在 Trino 中找不到。

傳回包 `VARCHAR` 含執行查詢呼叫函式之主體 (IAM 角色或識別中心身分識別) 的 ARN。例如，如果查詢叫用者使用 IAM 角色的許可執行查詢，則函數會傳回 IAM 角色的 ARN。執行查詢的角色必須允許 `LakeFormation:GetDataLakePrincipal` 動作。

## 用量

```
SELECT invoker_principal()
```

下表顯示範例結果。

#	_col0
1	<i>##:AWN: #: 111122223333: ##/###</i>

## Athena 引擎版本 2

Athena 引擎版本 2 中的函數以 [Presto 0.217](#) 為基礎。如需 Athena 引擎版本 2 的地理空間函數，請參閱 [Athena 引擎版本 2 中的地理空間函數](#)。

### Note

已不再提供 Presto 0.217 函數的版本特定文件。如需有關目前 Presto 函數、運算子和表達式的資訊，請參閱 [Presto functions and operators](#) (Presto 函數和運算子)，或造訪此部分中的子類別連結。

- [邏輯運算子](#)
- [比較函數和運算子](#)
- [條件表達式](#)
- [轉換函數](#)
- [數學函數和運算子](#)
- [位元函數](#)
- [Decimal 函數和運算子](#)
- [字串函數和運算子](#)
- [二進位函數](#)
- [日期與時間函數和運算子](#)
- [規則運算式函數](#)
- [JSON 函數和運算子](#)
- [URL 函數](#)

- [彙總函數](#)
- [視窗函數](#)
- [色彩函數](#)
- [陣列函數和運算子](#)
- [對應函數和運算子](#)
- [Lambda 表達式和函數](#)
- [Teradata 函數](#)

## 支援的時區

您可以在 SELECT timestamp 陳述式中使用 AT TIME ZONE 運算子來指定傳回之時間戳記的時區，如以下範例所示：

```
SELECT timestamp '2012-10-31 01:00 UTC' AT TIME ZONE 'America/Los_Angeles' AS la_time;
```

### 結果

```
la_time
```

```
2012-10-30 18:00:00.000 America/Los_Angeles
```

下列清單包含可搭配 Athena 中 AT TIME ZONE 運算子使用的時區。如需其他與時區相關的函數和範例，請參閱 [時區函數和範例](#)。

```
Africa/Abidjan  
Africa/Accra  
Africa/Addis_Ababa  
Africa/Algiers  
Africa/Asmara  
Africa/Asmera  
Africa/Bamako  
Africa/Bangui  
Africa/Banjul  
Africa/Bissau  
Africa/Blantyre  
Africa/Brazzaville  
Africa/Bujumbura  
Africa/Cairo  
Africa/Casablanca
```

Africa/Ceuta  
Africa/Conakry  
Africa/Dakar  
Africa/Dar\_es\_Salaam  
Africa/Djibouti  
Africa/Douala  
Africa/El\_Aaiun  
Africa/Freetown  
Africa/Gaborone  
Africa/Harare  
Africa/Johannesburg  
Africa/Juba  
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/Adak  
America/Anchorage  
America/Anguilla  
America/Antigua  
America/Araguaina

America/Argentina/Buenos\_Aires  
America/Argentina/Catamarca  
America/Argentina/ComodRivadavia  
America/Argentina/Cordoba  
America/Argentina/Jujuy  
America/Argentina/La\_Rioja  
America/Argentina/Mendoza  
America/Argentina/Rio\_Gallegos  
America/Argentina/Salta  
America/Argentina/San\_Juan  
America/Argentina/San\_Luis  
America/Argentina/Tucuman  
America/Argentina/Ushuaia  
America/Aruba  
America/Asuncion  
America/Atikokan  
America/Atka  
America/Bahia  
America/Bahia\_Banderas  
America/Barbados  
America/Belem  
America/Belize  
America/Blanc-Sablon  
America/Boa\_Vista  
America/Bogota  
America/Boise  
America/Buenos\_Aires  
America/Cambridge\_Bay  
America/Campo\_Grande  
America/Cancun  
America/Caracas  
America/Catamarca  
America/Cayenne  
America/Cayman  
America/Chicago  
America/Chihuahua  
America/Coral\_Harbour  
America/Cordoba  
America/Costa\_Rica  
America/Creston  
America/Cuiaba  
America/Curacao  
America/Danmarkshavn  
America/Dawson

America/Dawson\_Creek  
America/Denver  
America/Detroit  
America/Dominica  
America/Edmonton  
America/Eirunepe  
America/El\_Salvador  
America/Ensenada  
America/Fort\_Nelson  
America/Fort\_Wayne  
America/Fortaleza  
America/Glace\_Bay  
America/Godthab  
America/Goose\_Bay  
America/Grand\_Turk  
America/Grenada  
America/Guadeloupe  
America/Guatemala  
America/Guayaquil  
America/Guyana  
America/Halifax  
America/Havana  
America/Hermosillo  
America/Indiana/Indianapolis  
America/Indiana/Knox  
America/Indiana/Marengo  
America/Indiana/Petersburg  
America/Indiana/Tell\_City  
America/Indiana/Vevay  
America/Indiana/Vincennes  
America/Indiana/Winamac  
America/Indianapolis  
America/Inuvik  
America/Iqaluit  
America/Jamaica  
America/Jujuy  
America/Juneau  
America/Kentucky/Louisville  
America/Kentucky/Monticello  
America/Knox\_IN  
America/Kralendijk  
America/La\_Paz  
America/Lima  
America/Los\_Angeles

America/Louisville  
America/Lower\_Princes  
America/Maceio  
America/Managua  
America/Manaus  
America/Marigot  
America/Martinique  
America/Matamoros  
America/Mazatlan  
America/Mendoza  
America/Menominee  
America/Merida  
America/Metlakatla  
America/Mexico\_City  
America/Miquelon  
America/Moncton  
America/Monterrey  
America/Montevideo  
America/Montreal  
America/Montserrat  
America/Nassau  
America/New\_York  
America/Nipigon  
America/Nome  
America/Noronha  
America/North\_Dakota/Beulah  
America/North\_Dakota/Center  
America/North\_Dakota/New\_Salem  
America/Ojinaga  
America/Panama  
America/Pangnirtung  
America/Paramaribo  
America/Phoenix  
America/Port-au-Prince  
America/Port\_of\_Spain  
America/Porto\_Acre  
America/Porto\_Velho  
America/Puerto\_Rico  
America/Punta\_Arenas  
America/Rainy\_River  
America/Rankin\_Inlet  
America/Recife  
America/Regina  
America/Resolute

America/Rio\_Branco  
America/Rosario  
America/Santa\_Isabel  
America/Santarem  
America/Santiago  
America/Santo\_Domingo  
America/Sao\_Paulo  
America/Scoresbysund  
America/Shiprock  
America/Sitka  
America/St\_Barthelemy  
America/St\_Johns  
America/St\_Kitts  
America/St\_Lucia  
America/St\_Thomas  
America/St\_Vincent  
America/Swift\_Current  
America/Tegucigalpa  
America/Thule  
America/Thunder\_Bay  
America/Tijuana  
America/Toronto  
America/Tortola  
America/Vancouver  
America/Virgin  
America/Whitehorse  
America/Winnipeg  
America/Yakutat  
America/Yellowknife  
Antarctica/Casey  
Antarctica/Davis  
Antarctica/DumontDURville  
Antarctica/Macquarie  
Antarctica/Mawson  
Antarctica/McMurdo  
Antarctica/Palmer  
Antarctica/Rothera  
Antarctica/South\_Pole  
Antarctica/Syowa  
Antarctica/Troll  
Antarctica/Vostok  
Arctic/Longyearbyen  
Asia/Aden  
Asia/Almaty

Asia/Amman  
Asia/Anadyr  
Asia/Aqtau  
Asia/Aqtobe  
Asia/Ashgabat  
Asia/Ashkhabad  
Asia/Atyrau  
Asia/Baghdad  
Asia/Bahrain  
Asia/Baku  
Asia/Bangkok  
Asia/Barnaul  
Asia/Beirut  
Asia/Bishkek  
Asia/Brunei  
Asia/Calcutta  
Asia/Chita  
Asia/Choibalsan  
Asia/Chongqing  
Asia/Chungking  
Asia/Colombo  
Asia/Dacca  
Asia/Damascus  
Asia/Dhaka  
Asia/Dili  
Asia/Dubai  
Asia/Dushanbe  
Asia/Gaza  
Asia/Harbin  
Asia/Hebron  
Asia/Ho\_Chi\_Minh  
Asia/Hong\_Kong  
Asia/Hovd  
Asia/Irkutsk  
Asia/Istanbul  
Asia/Jakarta  
Asia/Jayapura  
Asia/Jerusalem  
Asia/Kabul  
Asia/Kamchatka  
Asia/Karachi  
Asia/Kashgar  
Asia/Kathmandu  
Asia/Katmandu

Asia/Khandyga  
Asia/Kolkata  
Asia/Krasnoyarsk  
Asia/Kuala\_Lumpur  
Asia/Kuching  
Asia/Kuwait  
Asia/Macao  
Asia/Macau  
Asia/Magadan  
Asia/Makassar  
Asia/Manila  
Asia/Muscat  
Asia/Nicosia  
Asia/Novokuznetsk  
Asia/Novosibirsk  
Asia/Omsk  
Asia/Oral  
Asia/Phnom\_Penh  
Asia/Pontianak  
Asia/Pyongyang  
Asia/Qatar  
Asia/Qyzylorda  
Asia/Rangoon  
Asia/Riyadh  
Asia/Saigon  
Asia/Sakhalin  
Asia/Samarkand  
Asia/Seoul  
Asia/Shanghai  
Asia/Singapore  
Asia/Srednekolymsk  
Asia/Taipei  
Asia/Tashkent  
Asia/Tbilisi  
Asia/Tehran  
Asia/Tel\_Aviv  
Asia/Thimbu  
Asia/Thimphu  
Asia/Tokyo  
Asia/Tomsk  
Asia/Ujung\_Pandang  
Asia/Ulaanbaatar  
Asia/Ulan\_Bator  
Asia/Urumqi

Asia/Ust-Nera  
Asia/Vientiane  
Asia/Vladivostok  
Asia/Yakutsk  
Asia/Yangon  
Asia/Yekaterinburg  
Asia/Yerevan  
Atlantic/Azores  
Atlantic/Bermuda  
Atlantic/Canary  
Atlantic/Cape\_Verde  
Atlantic/Faeroe  
Atlantic/Faroe  
Atlantic/Jan\_Mayen  
Atlantic/Madeira  
Atlantic/Reykjavik  
Atlantic/South\_Georgia  
Atlantic/St\_Helena  
Atlantic/Stanley  
Australia/ACT  
Australia/Adelaide  
Australia/Brisbane  
Australia/Broken\_Hill  
Australia/Canberra  
Australia/Currie  
Australia/Darwin  
Australia/Eucla  
Australia/Hobart  
Australia/LHI  
Australia/Lindeman  
Australia/Lord\_Howe  
Australia/Melbourne  
Australia/NSW  
Australia/North  
Australia/Perth  
Australia/Queensland  
Australia/South  
Australia/Sydney  
Australia/Tasmania  
Australia/Victoria  
Australia/West  
Australia/Yancowinna  
Brazil/Acre  
Brazil/DeNoronha

Brazil/East  
Brazil/West  
CET  
CST6CDT  
Canada/Atlantic  
Canada/Central  
Canada/Eastern  
Canada/Mountain  
Canada/Newfoundland  
Canada/Pacific  
Canada/Saskatchewan  
Canada/Yukon  
Chile/Continental  
Chile/EasterIsland  
Cuba  
EET  
EST5EDT  
Egypt  
Eire  
Europe/Amsterdam  
Europe/Andorra  
Europe/Astrakhan  
Europe/Athens  
Europe/Belfast  
Europe/Belgrade  
Europe/Berlin  
Europe/Bratislava  
Europe/Brussels  
Europe/Bucharest  
Europe/Budapest  
Europe/Busingen  
Europe/Chisinau  
Europe/Copenhagen  
Europe/Dublin  
Europe/Gibraltar  
Europe/Guernsey  
Europe/Helsinki  
Europe/Isle\_of\_Man  
Europe/Istanbul  
Europe/Jersey  
Europe/Kaliningrad  
Europe/Kiev  
Europe/Kirov  
Europe/Lisbon

Europe/Ljubljana  
Europe/London  
Europe/Luxembourg  
Europe/Madrid  
Europe/Malta  
Europe/Mariehamn  
Europe/Minsk  
Europe/Monaco  
Europe/Moscow  
Europe/Nicosia  
Europe/Oslo  
Europe/Paris  
Europe/Podgorica  
Europe/Prague  
Europe/Riga  
Europe/Rome  
Europe/Samara  
Europe/San\_Marino  
Europe/Sarajevo  
Europe/Simferopol  
Europe/Skopje  
Europe/Sofia  
Europe/Stockholm  
Europe/Tallinn  
Europe/Tirane  
Europe/Tiraspol  
Europe/Ulyanovsk  
Europe/Uzhgorod  
Europe/Vaduz  
Europe/Vatican  
Europe/Vienna  
Europe/Vilnius  
Europe/Volgograd  
Europe/Warsaw  
Europe/Zagreb  
Europe/Zaporozhye  
Europe/Zurich  
GB  
GB-Eire  
Hongkong  
Iceland  
Indian/Antananarivo  
Indian/Chagos  
Indian/Christmas

Indian/Cocos  
Indian/Comoro  
Indian/Kerguelen  
Indian/Mahe  
Indian/Maldives  
Indian/Mauritius  
Indian/Mayotte  
Indian/Reunion  
Iran  
Israel  
Jamaica  
Japan  
Kwajalein  
Libya  
MET  
MST7MDT  
Mexico/BajaNorte  
Mexico/BajaSur  
Mexico/General  
NZ  
NZ-CHAT  
Navajo  
PRC  
PST8PDT  
Pacific/Apia  
Pacific/Auckland  
Pacific/Bougainville  
Pacific/Chatham  
Pacific/Chuuk  
Pacific/Easter  
Pacific/Efate  
Pacific/Enderbury  
Pacific/Fakaofu  
Pacific/Fiji  
Pacific/Funafuti  
Pacific/Galapagos  
Pacific/Gambier  
Pacific/Guadalcanal  
Pacific/Guam  
Pacific/Honolulu  
Pacific/Johnston  
Pacific/Kiritimati  
Pacific/Kosrae  
Pacific/Kwajalein

Pacific/Majuro  
Pacific/Marquesas  
Pacific/Midway  
Pacific/Nauru  
Pacific/Niue  
Pacific/Norfolk  
Pacific/Noumea  
Pacific/Pago\_Pago  
Pacific/Palau  
Pacific/Pitcairn  
Pacific/Pohnpei  
Pacific/Ponape  
Pacific/Port\_Moresby  
Pacific/Rarotonga  
Pacific/Saipan  
Pacific/Samoa  
Pacific/Tahiti  
Pacific/Tarawa  
Pacific/Tongatapu  
Pacific/Truk  
Pacific/Wake  
Pacific/Wallis  
Pacific/Yap  
Poland  
Portugal  
ROK  
Singapore  
Turkey  
US/Alaska  
US/Aleutian  
US/Arizona  
US/Central  
US/East-Indiana  
US/Eastern  
US/Hawaii  
US/Indiana-Starke  
US/Michigan  
US/Mountain  
US/Pacific  
US/Pacific-New  
US/Samoa  
W-SU  
WET

## 時區函數和範例

以下是一些其他與時區相關的函數和範例。

- `at_timezone(timestamp, zone)` – 傳回##相應當地時間的####值。

### 範例

```
SELECT at_timezone(timestamp '2021-08-22 00:00 UTC', 'Canada/Newfoundland')
```

### 結果

```
2021-08-21 21:30:00.000 Canada/Newfoundland
```

- `timezone_hour(timestamp)` – 以 `bigint` 傳回時區偏離時間戳記的小時數。

### 範例

```
SELECT timezone_hour(timestamp '2021-08-22 04:00 UTC' AT TIME ZONE 'Canada/Newfoundland')
```

### 結果

```
-2
```

- `timezone_minute(timestamp)` – 以 `bigint` 傳回時區偏離####的分鐘數。

### 範例

```
SELECT timezone_minute(timestamp '2021-08-22 04:00 UTC' AT TIME ZONE 'Canada/Newfoundland')
```

### 結果

```
-30
```

- `with_timezone(timestamp, zone)` – 從指定的####和##值傳回時間戳記和時區。

### 範例

```
SELECT with_timezone(timestamp '2021-08-22 04:00', 'Canada/Newfoundland')
```

## 結果

```
2021-08-22 04:00:00.000 Canada/Newfoundland
```

## DDL 陳述式

在 Athena 中直接使用以下 DDL 陳述式。

Athena 查詢引擎部分以 [HiveQL DDL](#) 為基礎。

Athena 不支援所有 DDL 陳述式，而且 HiveQL DDL 和 Athena DDL 之間有一些差異。如需詳細資訊，請參閱本節中的參考主題和[不支援的 DDL](#)。

### 主題

- [不支援的 DDL](#)
- [ALTER DATABASE SET DBPROPERTIES](#)
- [ALTER TABLE ADD COLUMNS](#)
- [ALTER TABLE ADD PARTITION](#)
- [ALTER TABLE DROP PARTITION](#)
- [ALTER TABLE RENAME PARTITION](#)
- [ALTER TABLE REPLACE COLUMNS](#)
- [ALTER TABLE SET LOCATION](#)
- [ALTER TABLE SET TBLPROPERTIES](#)
- [CREATE DATABASE](#)
- [CREATE TABLE](#)
- [CREATE TABLE AS](#)
- [CREATE VIEW](#)
- [DESCRIBE](#)
- [DESCRIBE VIEW](#)
- [DROP DATABASE](#)
- [DROP TABLE](#)

- [DROP VIEW](#)
- [MSCK REPAIR TABLE](#)
- [SHOW COLUMNS](#)
- [SHOW CREATE TABLE](#)
- [SHOW CREATE VIEW](#)
- [SHOW DATABASES](#)
- [SHOW PARTITIONS](#)
- [SHOW TABLES](#)
- [SHOW TBLPROPERTIES](#)
- [SHOW VIEWS](#)

## 不支援的 DDL

Athena 不支援下列 DDL 陳述式：

- ALTER INDEX
- ALTER TABLE *table\_name* ARCHIVE PARTITION
- ALTER TABLE *table\_name* CLUSTERED BY
- ALTER TABLE *table\_name* EXCHANGE PARTITION
- ALTER TABLE *table\_name* CLUSTERED BY
- ALTER TABLE *table\_name* NOT SKEWED
- ALTER TABLE *table\_name* NOT SORTED
- ALTER TABLE *table\_name* NOT STORED AS DIRECTORIES
- ALTER TABLE *table\_name* partitionSpec CHANGE COLUMNS
- ALTER TABLE *table\_name* partitionSpec COMPACT
- ALTER TABLE *table\_name* partitionSpec CONCATENATE
- ALTER TABLE *table\_name* partitionSpec SET FILEFORMAT
- ALTER TABLE *table\_name* 設定服務器屬性
- ALTER TABLE *table\_name* SET SKEWED LOCATION
- ALTER TABLE *table\_name* SKEWED BY
- ALTER TABLE *table\_name* TOUCH

- ALTER TABLE *table\_name* UNARCHIVE PARTITION
- COMMIT
- CREATE INDEX
- CREATE ROLE
- CREATE TABLE *table\_name* LIKE *existing\_table\_name*
- CREATE TEMPORARY MACRO
- DELETE FROM
- DESCRIBE DATABASE
- DFS
- DROP INDEX
- DROP ROLE
- DROP TEMPORARY MACRO
- EXPORT TABLE
- GRANT ROLE
- IMPORT TABLE
- LOCK DATABASE
- LOCK TABLE
- REVOKE ROLE
- ROLLBACK
- SHOW COMPACTIONS
- SHOW CURRENT ROLES
- SHOW GRANT
- SHOW INDEXES
- SHOW LOCKS
- SHOW PRINCIPALS
- SHOW ROLE GRANT
- SHOW ROLES
- SHOW STATS
- SHOW TRANSACTIONS
- START TRANSACTION

- UNLOCK DATABASE
- UNLOCK TABLE

## ALTER DATABASE SET DBPROPERTIES

為資料庫建立一或多個屬性。DATABASE 和 SCHEMA 可互換使用，兩者的意義相同。

### 概要

```
ALTER {DATABASE|SCHEMA} database_name
  SET DBPROPERTIES ('property_name'='property_value' [, ...] )
```

### 參數

```
SET DBPROPERTIES ('property_name'='property_value' [, ...])
```

指定名為 `property_name` 之資料庫之一或多個屬性，並以 `property_value` 分別為每個屬性建立值。如果 `property_name` 已存在，`property_value` 會覆寫舊值。

### 範例

```
ALTER DATABASE jd_datasets
  SET DBPROPERTIES ('creator'='John Doe', 'department'='applied mathematics');
```

```
ALTER SCHEMA jd_datasets
  SET DBPROPERTIES ('creator'='Jane Doe');
```

## ALTER TABLE ADD COLUMNS

將一個或多個資料欄新增至現有資料表中。使用選用的 PARTITION 語法時，會更新分割區中繼資料。

### 概要

```
ALTER TABLE table_name
  [PARTITION
    (partition_col1_name = partition_col1_value
    [,partition_col2_name = partition_col2_value][,...])]
  ADD COLUMNS (col_name data_type)
```

## 參數

`PARTITION (partition_col_name = partition_col_value [...])`

使用您指定的資料欄名稱/值組合來建立分割區。只有當資料欄的資料類型為字串時，才需要用引號括住 `partition_col_value`。

`ADD COLUMNS (col_name data_type [,col_name data_type,...])`

將資料行新增在現有資料欄之後，但在分割區資料欄之前。

## 範例

```
ALTER TABLE events ADD COLUMNS (eventowner string)
```

```
ALTER TABLE events PARTITION (awsregion='us-west-2') ADD COLUMNS (event string)
```

```
ALTER TABLE events PARTITION (awsregion='us-west-2') ADD COLUMNS (eventdescription  
string)
```

## 備註

- 若要於執行 `ALTER TABLE ADD COLUMNS` 之後在 Athena 查詢編輯器導覽窗格中查看新資料表資料欄，請手動重新整理編輯器中的資料表清單，然後再次展開資料表。
- `ALTER TABLE ADD COLUMNS` 不適用於具有 `date` 資料類型的資料欄。若要解決這個問題，請轉為使用 `timestamp` 資料類型。

## ALTER TABLE ADD PARTITION

為資料表建立一或多個分割區欄。每個分割區由一或多個不同的欄名稱/值組合組成。每種指定的組合都會另外建立一個資料目錄，這樣可在某些情況下改善查詢效能。分割區的欄本身不存在於資料表的資料中，因此，如果您使用的分割區欄名稱和資料表中的欄名稱相同，就會發生錯誤。如需詳細資訊，請參閱 [在 Athena 中分割資料](#)。

在 Athena 中，資料表及其分割區必須使用相同的資料格式，但其結構描述可能不同。如需詳細資訊，請參閱 [資料表和分割區中的更新](#)。

如需有關 IAM 政策中所需資源層級許可的資訊 (包括 `glue:CreatePartition`)，請參閱 [《AWS Glue API 許可：動作和資源參考》](#) 以及 [對於中的資料庫和資料表的精細存取 AWS Glue Data](#)

[Catalog](#)。如需有關使用 Athena 時許可的疑難排解資訊，請參閱 [Athena 中的疑難排解](#) 主題的 [許可](#) 章節。

## 概要

```
ALTER TABLE table_name ADD [IF NOT EXISTS]
PARTITION
(partition_col1_name = partition_col1_value
[,partition_col2_name = partition_col2_value]
[,...])
[LOCATION 'location1']
[PARTITION
(partition_colA_name = partition_colA_value
[,partition_colB_name = partition_colB_value
[,...]])]
[LOCATION 'location2']
[,...]
```

## 參數

當新增分割區時，您可以為分割區指定一或多個資料欄名稱/值組，以及該分割區的資料檔案所在的 Simple Storage Service (Amazon S3) 路徑。

### [IF NOT EXISTS]

如果已有相同定義的分割區，則會造成錯誤隱藏。

PARTITION (partition\_col\_name = partition\_col\_value [...])

使用您指定的資料欄名稱/值組合來建立分割區。只有當欄的資料類型是字串時，才需要以字串字元圍住 partition\_col\_value。

### [LOCATION 'location']

指定目錄，將在其中存放先前陳述式定義的分割區。當資料使用 Hive 樣式分割 (pk1=v1/pk2=v2/pk3=v3) 時，LOCATION 子句為選用的。使用 Hive 樣式分割時，會從資料表的位置、分割區索引鍵名稱和分割區索引鍵值自動建構完整的 Amazon S3 URI。如需詳細資訊，請參閱 [在 Athena 中分割資料](#)。

## 考量事項

Amazon Athena 不會對您可以在單一 ALTER TABLE ADD PARTITION DDL 陳述式中新增的分割區數量施加特定限制。不過，如果您需要新增大量的分割區，請考慮將作業分成較小的批次，以避免潛在

的效能問題。下列範例會使用連續的指令來個別新增分割區，並用 IF NOT EXISTS 來避免新增重複項目。

```
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION (ds='2023-01-01')
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION (ds='2023-01-02')
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION (ds='2023-01-03')
```

在 Athena 中使用分割區時，請記住以下幾點：

- 雖然 Athena 支援查詢具有 1000 萬個分割區的 AWS Glue 資料表，但是 Athena 在一次掃描中無法讀取超過 100 萬個分割區。
- 若要最佳化您的查詢並減少掃描的分割區數目，請考慮分割區修剪或使用分割區索引等策略。
- 如果您未使用 AWS Glue Data Catalog，則每個表的最大分區數量為 20,000 個。您可以要求增加配額。

如需在 Athena 中使用分割區的其他考量，請參閱 [在 Athena 中分割資料](#)。

## 範例

下列範例會將單一分割區新增至資料表，以取得 HIVE 樣式分割資料。

```
ALTER TABLE orders ADD
PARTITION (dt = '2016-05-14', country = 'IN');
```

下列範例會將多個分割區新增至資料表，以取得 HIVE 樣式分割資料。

```
ALTER TABLE orders ADD
PARTITION (dt = '2016-05-31', country = 'IN')
PARTITION (dt = '2016-06-01', country = 'IN');
```

如果資料表不適用於 HIVE 樣式的分割資料，則 LOCATION 子句是必需的，且應該是包含分割區資料之字首的完整 Amazon S3 URI。

```
ALTER TABLE orders ADD
PARTITION (dt = '2016-05-31', country = 'IN') LOCATION 's3://DOC-EXAMPLE-BUCKET/path/to/INDIA_31_May_2016/'
PARTITION (dt = '2016-06-01', country = 'IN') LOCATION 's3://DOC-EXAMPLE-BUCKET/path/to/INDIA_01_June_2016/';
```

若要在分割區已存在時忽略錯誤，請使用 IF NOT EXISTS 子句，如下列範例所示。

```
ALTER TABLE orders ADD IF NOT EXISTS
PARTITION (dt = '2016-05-14', country = 'IN');
```

### 零位元組 `_$folder$` 檔案

如果您執行 ALTER TABLE ADD PARTITION 語句並錯誤地指定已存在的分區和不正確的 Simple Storage Service (Amazon S3) 位置，則格式為 `partition_value_$folder$` 的零位元組預留位置檔案會在 Simple Storage Service (Amazon S3) 中建立。您必須手動移除這些檔案。

若要防止這種情況發生，請使用 ALTER TABLE ADD PARTITION 陳述式中的 ADD IF NOT EXISTS 語法，如下列範例所示。

```
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION [...]
```

## ALTER TABLE DROP PARTITION

捨棄指定資料表的一或多個指定分割區。

### 概要

```
ALTER TABLE table_name DROP [IF EXISTS] PARTITION (partition_spec) [, PARTITION
(partition_spec)]
```

### 參數

#### [IF EXISTS]

如果指定的分割區不存在，會造成錯誤訊息隱藏。

#### PARTITION (partition\_spec)

每個 partition\_spec 會以 partition\_col\_name = partition\_col\_value [,...] 的形式指定一個資料欄名稱/值組合。

### 範例

```
ALTER TABLE orders
DROP PARTITION (dt = '2014-05-14', country = 'IN');
```

```
ALTER TABLE orders
DROP PARTITION (dt = '2014-05-14', country = 'IN'), PARTITION (dt = '2014-05-15',
country = 'IN');
```

## 備註

ALTER TABLE DROP PARTITION 陳述式不會提供單一語法來一次捨棄所有分割區，或支援篩選條件來指定要捨棄的分割區範圍。

因應措施是您可以在指令碼中使用 AWS Glue API [GetPartitions](#) 和 [BatchDeletePartition](#) 動作。GetPartitions 動作支援類似於 SQL WHERE 表達式中的複雜篩選條件表達式。使用 GetPartitions 建立要刪除的分割區篩選清單後，您可以使用 BatchDeletePartition 動作以刪除以 25 為單位批次的分割區。

### Important

由於已知問題，當 ALTER TABLE DROP PARTITION 陳述式指定了無效的分割區時，資料表的所有分割區都會在 AWS Glue 中被刪除。例如，即使指定的分割區不存在，下列陳述式仍會刪除 *my\_table* 資料表的所有分割區。解決方法是在執行 ALTER TABLE DROP PARTITION 陳述式之前，確保您已正確輸入分割區資訊。

```
ALTER TABLE my_table DROP IF EXISTS PARTITION(zzz='');
```

## ALTER TABLE RENAME PARTITION

重新命名分割區值。

### Note

ALTER TABLE RENAME PARTITION 不會重新命名分割區資料行。若要變更分割區欄名稱，您可以使用主 AWS Glue 控制台。如需詳細資訊，請參閱本文件稍後的 [重命名分區列 AWS Glue](#)。

## 概要

對於名為的資料表 *table\_name*，*partition\_spec* 將由指定的分割區值重新命名為指定的 *new\_partition\_spec* 值。

```
ALTER TABLE table_name PARTITION (partition_spec) RENAME TO PARTITION
(new_partition_spec)
```

## 參數

### PARTITION (partition\_spec)

每個 partition\_spec 會以 partition\_col\_name = partition\_col\_value [,...] 的形式指定一個資料欄名稱/值組合。

## 範例

```
ALTER TABLE orders
PARTITION (dt = '2014-05-14', country = 'IN') RENAME TO PARTITION (dt = '2014-05-15',
country = 'IN');
```

## 重命名分區列 AWS Glue

請使用下列程序在 AWS Glue 主控台中重新命名分割區資料行名稱。

若要在 AWS Glue 主控台中重新命名表格分割區資料行

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，網址為 <https://console.aws.amazon.com/glue/>。
2. 在導覽窗格中，選擇 Tables (資料表)。
3. 在「表格」頁面上，使用篩選表格搜尋方塊來尋找您要變更的表格。
4. 在「名稱」欄中，選擇您要變更之表格的連結。
5. 在表格的詳細資訊頁面的「綱要」段落中，執行下列其中一個動作：
  - 若要變更 JSON 格式的名稱，請選擇「將結構定義編輯為 JSON」。
  - 若要直接變更名稱，請選擇 [編輯綱要]。此程序會選擇「編輯綱要」。
6. 選取您要重新命名之分割資料行的核取方塊，然後選擇 [編輯]。
7. 在 [編輯架構項目] 對話方塊中，對於 [名稱]，輸入分割區資料欄的新名稱。
8. 選擇「另存為新表格版本」。此動作會更新分割區資料欄名稱並保留結構描述演進歷程記錄，而不需要另外建立資料的實體副本。
9. 若要比較表格版本，請在表格的詳細資訊頁上選擇「動作」，然後選擇「比較版本」。

## 其他資源

如需有關分割區的詳細資訊，請參閱 [在 Athena 中分割資料](#)。

## ALTER TABLE REPLACE COLUMNS

從使用建立的表格中移除所有現有欄，[LazySimpleSerDe](#)並以指定的欄集取代它們。使用選用的 PARTITION 語法時，會更新分割區中繼資料。您也可以使用 ALTER TABLE REPLACE COLUMNS 透過僅指定您想要保留的資料欄來刪除其他資料欄。

### 概要

```
ALTER TABLE table_name
  [PARTITION
    (partition_col1_name = partition_col1_value
    [,partition_col2_name = partition_col2_value][,...])]
  REPLACE COLUMNS (col_name data_type [, col_name data_type, ...])
```

### 參數

PARTITION (partition\_col\_name = partition\_col\_value [...])

以您指定的資料欄名稱/值組合來指定分割區。只有當資料欄的資料類型為字串時，才需要用引號括住 partition\_col\_value。

REPLACE COLUMNS (col\_name data\_type [,col\_name data\_type,...])

以指定的資料欄名稱和資料類型取代現有資料欄。

### 備註

- 如需在執行 ALTER TABLE REPLACE COLUMNS 之後，查看 Athena 查詢編輯器導覽窗格中的資料表資料表欄位變更，請手動重新整理編輯器中的資料表清單，然後再次展開資料表。
- ALTER TABLE REPLACE COLUMNS 不適用於具有 date 資料類型的資料欄。若要解決這個問題，請轉為使用資料表中的 timestamp 資料類型。
- 請注意，即使您只取代單一資料欄，語法必須是 ALTER TABLE *table-name* REPLACE COLUMNS，以及複數形式的資料欄。您不僅必須指定要取代的資料欄，還必須指定要保留的資料欄 – 如果不是，您未指定的資料欄將會遭到捨棄。這種語法和行為從 Apache Hive DDL 衍生。如需參考，請參閱 Apache 文件中的 [新增/取代資料欄](#)。

## 範例

在下列範例中，使用建立的資料表 `names_cities` 具有三個名為 `col1`、`col2`、和的欄 `col3`。[LazySimpleSerDe](#) 所有資料欄均為 `string` 類型。若要顯示資料表中的資料欄，下列命令會使用 [SHOW COLUMNS](#) 陳述式。

```
SHOW COLUMNS IN names_cities
```

查詢的結果：

```
col1
col2
col3
```

下列 `ALTER TABLE REPLACE COLUMNS` 命令將資料欄名稱取代為 `first_name`、`last_name` 和 `city`。基礎來源資料不受影響。

```
ALTER TABLE names_cities
REPLACE COLUMNS (first_name string, last_name string, city string)
```

若要測試結果，請重新執行 `SHOW COLUMNS`。

```
SHOW COLUMNS IN names_cities
```

查詢的結果：

```
first_name
last_name
city
```

顯示新資料欄名稱的另一種方法是在 Athena 查詢編輯器中[預覽資料表](#)，或執行您自己的 `SELECT` 查詢。

## ALTER TABLE SET LOCATION

變更為 `table_name` 之資料表的位置，以及 (選擇性) 含 `partition_spec` 的分割區。

### 概要

```
ALTER TABLE table_name [ PARTITION (partition_spec) ] SET LOCATION 'new location'
```

## 參數

### PARTITION (partition\_spec)

以 `partition_spec` 參數指定您想要變更位置的分割區。 `partition_spec` 以 `partition_col_name = partition_col_value` 的形式指定一個欄名稱/值組合。

### SET LOCATION 'new location'

指定新的位置，其必須是 Amazon S3 位置。如需有關語法的資訊，請參閱 [Amazon S3 中的資料表位置](#)。

## 範例

```
ALTER TABLE customers PARTITION (zip='98040', state='WA') SET LOCATION 's3://DOC-EXAMPLE-BUCKET/custdata/';
```

## ALTER TABLE SET TBLPROPERTIES

將自訂或預定義的中繼資料屬性新增到資料表中，並設定他們的指派值。若要查看資料表中的屬性，請使用 [SHOW TBLPROPERTIES](#) 命令。

由於不支援 Apache Hive [受管資料表](#)，設定 'EXTERNAL'='FALSE' 不會有任何作用。

## 概要

```
ALTER TABLE table_name SET TBLPROPERTIES ('property_name' = 'property_value' [ , ... ])
```

## 參數

### SET TBLPROPERTIES ('property\_name' = 'property\_value' [ , ... ])

指定以 `property_name` 新增的中繼資料屬性，以及以 `property value` 新增的個別值。如果 `property_name` 已存在，它的值會設定為新指定的 `property_value`。

下列預先定義的資料表屬性有特殊用途。

預先定義的屬性	描述
<code>classification</code>	指示的資料類型 AWS Glue。可能值為 <code>csv</code> 、 <code>parquet</code> 、 <code>orc</code> 、 <code>avro</code> 或 <code>json</code> 。在 CloudTrail 主控台中為 Athena 建立的表格會新增 <code>cloudtrai</code>

預先定義的屬性	描述
	l 為 classification 屬性的值。如需詳細資訊，請參閱 <a href="#">CREATE TABLE</a> 的 TBLPROPERTIES 章節。
has_encrypted_data	指出 LOCATION 指定的資料集是否已加密。如需詳細資訊，請參閱 <a href="#">CREATE TABLE</a> 和 <a href="#">根據 Amazon S3 中加密的資料集建立資料表</a> 的 TBLPROPERTIES 章節。
orc.compress	以 ORC 格式指定資料的壓縮格式。如需詳細資訊，請參閱 <a href="#">獸人 SerDe</a> 。
parquet.compression	指定 Parquet 格式之資料的壓縮格式。如需詳細資訊，請參閱 <a href="#">木地板 SerDe</a> 。
write.compression	以文字檔案格式或 JSON 格式指定資料的壓縮格式。如果是 Parquet 和 ORC 格式，請分別使用 parquet.compression 和 orc.compress 屬性。
compression_level	指定要使用的壓縮級別。此屬性僅適用於 ZSTD 壓縮。可能的值為 1 到 22。預設值為 3。如需詳細資訊，請參閱 <a href="#">在 Athena 中使用 ZSTD 壓縮級別</a> 。
projection.*	分割區投影中使用的自訂屬性可讓 Athena 知道在資料表上執行查詢時預期的分割區模式。如需詳細資訊，請參閱 <a href="#">使用 Amazon Athena 進行分割區投影</a> 。
skip.header.line.count	在定義表格時忽略資料中的標頭。如需詳細資訊，請參閱 <a href="#">忽略標頭</a> 。
storage.location.template	為投影分割區指定自訂 Amazon S3 路徑範本。如需詳細資訊，請參閱 <a href="#">設定分割區投影</a> 。

## 範例

下列範例會將註解附註新增至資料表屬性。

```
ALTER TABLE orders
```

```
SET TBLPROPERTIES ('notes'="Please don't drop this table.");
```

下列範例會修改資料表 `existing_table`，以使用具有 ZSTD 壓縮和 ZSTD 壓縮級別 4 的 Parquet 檔案格式。

```
ALTER TABLE existing_table  
SET TBLPROPERTIES ('parquet.compression' = 'ZSTD', 'compression_level' = 4)
```

## CREATE DATABASE

建立資料庫。DATABASE 和 SCHEMA 可互換使用。它們的意義相同。

### Note

如需在 Athena 建立資料庫、建立資料表和執行資料表 SELECT 查詢的範例，請參閱 [開始使用](#)。

## 概要

```
CREATE {DATABASE|SCHEMA} [IF NOT EXISTS] database_name  
  [COMMENT 'database_comment']  
  [LOCATION 'S3_loc']  
  [WITH DBPROPERTIES ('property_name' = 'property_value') [, ...]]
```

## 參數

### [IF NOT EXISTS]

如果名為 `database_name` 的資料表已存在，會造成錯誤隱藏。

### [COMMENT database\_comment]

為名為 `comment` 的內建中繼資料屬性以及您為 `database_comment` 提供的值，建立中繼資料值。在中 AWS Glue，內 COMMENT 容會寫入資料庫屬性的 Description 欄位。

### [LOCATION S3\_loc]

以 `S3_loc` 指定資料庫檔案和中繼存放區將存在的位置。此位置必須是 Amazon S3 位置。

### [WITH DBPROPERTIES ('property\_name' = 'property\_value') [, ...]]

可讓您指定資料庫定義的自訂中繼資料屬性。

## 範例

```
CREATE DATABASE clickstreams;
```

```
CREATE DATABASE IF NOT EXISTS clickstreams
  COMMENT 'Site Foo clickstream data aggregates'
  LOCATION 's3://DOC-EXAMPLE-BUCKET/clickstreams/'
  WITH DBPROPERTIES ('creator'='Jane D.', 'Dept.'='Marketing analytics');
```

## 檢視資料庫屬性

若要檢視您在使用中建立之資料庫的資料庫屬性CREATE DATABASE，您可以 AWSDataCatalog 使用 AWS CLI 命令 [aws glue get-database](#)，如下列範例所示：

```
aws glue get-database --name <your-database-name>
```

在 JSON 輸出中，其結果看起來如下：

```
{
  "Database": {
    "Name": "<your-database-name>",
    "Description": "<your-database-comment>",
    "LocationUri": "s3://DOC-EXAMPLE-BUCKET",
    "Parameters": {
      "<your-database-property-name>": "<your-database-property-value>"
    },
    "CreateTime": 1603383451.0,
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ]
  }
}
```

若要取得有關的更多資訊 AWS CLI，請參閱 [《AWS Command Line Interface 使用者指南》](#)。

## CREATE TABLE

以您指定的名稱和參數建立資料表。

### Note

此頁包含摘要參考資訊。如需在 Athena 中建立資料表的詳細資訊和範例 CREATE TABLE 陳述式，請參閱[在 Athena 中建立資料表](#)。如需在 Athena 建立資料庫、建立資料表和執行資料表 SELECT 查詢的範例，請參閱[開始使用](#)。

### 概要

```
CREATE EXTERNAL TABLE [IF NOT EXISTS]
  [db_name.]table_name [(col_name data_type [COMMENT col_comment] [, ...] )]
  [COMMENT table_comment]
  [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
  [CLUSTERED BY (col_name, col_name, ...) INTO num_buckets BUCKETS]
  [ROW FORMAT row_format]
  [STORED AS file_format]
  [WITH SERDEPROPERTIES (...)]
  [LOCATION 's3://DOC-EXAMPLE-BUCKET/[folder]/']
  [TBLPROPERTIES ( ['has_encrypted_data'='true | false',]
  ['classification'='aws_glue_classification',] property_name=property_value [, ...] ) ]
```

### 參數

#### EXTERNAL

指定資料表依據您指定的 LOCATION 中 Amazon S3 現有的基礎資料檔案。除了在建立 [Iceberg](#) 資料表時，請始終使用 EXTERNAL 關鍵字。如果您在非 Iceberg 資料表上使用 CREATE TABLE 卻沒有帶 EXTERNAL 關鍵字，則 Athena 會發出錯誤。當您建立外部資料表，被參考的資料必須符合預設格式或您以 ROW FORMAT、STORED AS 及 WITH SERDEPROPERTIES 子句指定的格式。

#### [IF NOT EXISTS]

此參數會檢查具有相同名稱的資料表是否已經存在。如果存在，參數會傳回 TRUE，Amazon Athena 會取消 CREATE TABLE 動作。由於取消發生在 Athena 呼叫資料目錄之前，因此不會發出 AWS CloudTrail 事件。

`[db_name.]table_name`

指定要建立的資料表名稱。選用的 `db_name` 參數可指定資料表所在的資料庫。如果省略，會採用目前的資料庫。如果資料表名稱包含號碼，以引號括住 `table_name`，例如 "table123"。如果 `table_name` 以底線開頭，請使用反引號，例如 ``_mytable``。不支援特殊字元 (底線除外)。

Athena 資料表名稱不區分大小寫；不過，如果您使用 Apache Spark，則 Spark 的資料表名稱需為小寫字母。

`[ ( col_name data_type [COMMENT col_comment] [, ...] ) ]`

指定要建立的每一欄名稱以及欄的資料類型。欄名稱不可使用底線 (`_`) 以外的特殊字元。如果 `col_name` 以底線開頭，以反引號括住欄名稱，例如 ``_mycolumn``。

`data_type` 可以是下列任何值：

- `boolean` – 值為 `true` 和 `false`。
- `tinyint` – 以兩的補數格式表示的 8 位元的有號整數，最小值為  $-2^7$ ，最大值為  $2^7-1$ 。
- `smallint` – 以兩的補數格式表示的 16 位元的有號整數，最小值為  $-2^{15}$ ，最大值為  $2^{15}-1$ 。
- `int` – 在資料定義語言 (DDL) 查詢中，如 `CREATE TABLE`，請使用 `int` 關鍵字來表示整數。在其他查詢中，使用關鍵字 `integer`，該處為 `integer` 以兩的補數格式表示的 32 位元有號值，最小值為  $-2^{31}$ ，最大值為  $2^{31}-1$ 。在 JDBC 驅動程式中，會傳回 `integer`，以確保與商業分析應用程式的相容性。
- `bigint` – 以兩的補數格式表示的 64 位元的有號整數，最小值為  $-2^{63}$ ，最大值為  $2^{63}-1$ 。
- `double` – 64 位元雙精度浮點數。該範圍為 `4.94065645841246544e-324d` 至 `1.79769313486231570e+308d`，正極或負極。`double` 遵循浮點數運算的 IEEE 標準 (IEEE 754)。
- `float` – 32 位元單精度浮點數。該範圍為 `1.40129846432481707e-45` 至 `3.40282346638528860e+38`，正極或負極。`float` 遵循浮點數運算的 IEEE 標準 (IEEE 754)。等同於 Presto 中的 `real`。在 Athena 中，請使用 DDL 陳述式中的 `float`，例如 `CREATE TABLE`，以及 SQL 函數中的 `real`，例如 `SELECT CAST`。AWS Glue 爬蟲程式會傳回中的值 `float`，而 Athena 會在內部進行翻譯 `real` 和輸入 `float` (請參閱 [2018 年 6 月 5 日版本說明](#))。
- `decimal [ (precision, scale) ]`，其中 *precision* 是位數的總數，*scale* (選用) 則是分數部分的位數總數，預設值為 0。例如，使用這些類型定義：`decimal(11,5)`、`decimal(15)`。最大###為 38，而最大##值為 38。

若要指定小數值為常值，例如在查詢 DDL 表達式中以特定小數值選取列，可指定 decimal 類型定義，並 (在單引號中) 列出小數值為常值，如同此範例中的 `decimal_value = decimal '0.12'`。

- `char` – 固定長度的字元資料，介於 1 到 255 的指定長度，例如 `char(10)`。如需詳細資訊，請參閱 [CHAR Hive 資料類型](#)。
- `varchar` – 可變長度的字元資料，介於 1 到 65535 的指定長度，例如 `varchar(10)`。如需詳細資訊，請參閱 [VARCHAR Hive 資料類型](#)。
- `string` – 用單引號或雙引號括住的字串文字。

**Note**

非字串資料類型不能轉換為 Athena 中的 `string`；請改將它們轉換為 `varchar`。

- `binary` – (用於 Parquet 的資料)
- `date` – ISO 格式的日期，例如 `YYYY-MM-DD`。例如：`date '2008-09-15'`。OpenCSV SerDe 使用自 1970 年 1 月 1 日以來經過的天數是例外狀況。如需詳細資訊，請參閱 [SerDe 用於處理 CSV 的開啟 CSV](#)。
- `timestamp` – 在 [java.sql.Timestamp](#) 相容格式下的日期和時間常數，最大解析度為毫秒，例如 `yyyy-MM-dd HH:mm:ss[.f...]`。例如：`timestamp '2008-09-15 03:04:05.324'`。OpenCSV 是一個例外 SerDe，它使用 UNIX 數字格式的 `TIMESTAMP` 資料 (例如，1579059880000)。如需詳細資訊，請參閱 [SerDe 用於處理 CSV 的開啟 CSV](#)。
- `array < data_type >`
- `map < primitive_type, data_type >`
- `struct < col_name : data_type [comment col_comment] [, ...] >`

[COMMENT table\_comment]

建立 `comment` 資料表屬性，並填入您指定的 `table_comment`。

[PARTITIONED BY (col\_name data\_type [ COMMENT col\_comment ], ... )]

建立有分割區的資料表，具有包含指定的 `col_name`、`data_type`、`col_comment` 的一或多個分割區欄。資料表可以有一或多個分割區，各自由不同欄名稱和值組合而成。每種指定的組合都會另外建立一個資料目錄，這樣可在某些情況下改善查詢效能。資料表資料內並未包含分割資料欄。如果您使用的 `col_name` 值與資料表欄相同，則會發生錯誤。如需詳細資訊，請參閱 [分割資料](#)。

**Note**

在您建立分割區資料表後，執行包含 [MSCK REPAIR TABLE](#) 子句的後續查詢 (例如，`MSCK REPAIR TABLE cloudfront_logs;`) 以重新整理分割區中繼資料。對於與 Hive 不相容的分割區，請使用 [ALTER TABLE ADD PARTITION](#) 載入分割區，以便查詢資料。

[CLUSTERED BY (col\_name, col\_name, ...) INTO num\_buckets BUCKETS]

透過或不透過分割，將指定的 col\_name 資料欄中的資料分割為稱為儲存貯體的資料子集。num\_buckets 參數會指定要建立的儲存貯體數目。歸納可以改善對大型資料集的某些查詢的效能。

[ROW FORMAT row\_format]

指定資料表的列格式及其基礎來源資料 (如果適用)。在 row\_format 可以使用 DELIMITED 子句來指定一個或多個分隔符號，或者，使用 SERDE 子句，如下所示。如 ROW FORMAT 果省略或 ROW FORMAT DELIMITED 指定，則使 SerDe 用原生。

- [DELIMITED FIELDS TERMINATED BY char [ESCAPED BY char]]
- [DELIMITED COLLECTION ITEMS TERMINATED BY char]
- [MAP KEYS TERMINATED BY char]
- [LINES TERMINATED BY char]
- [NULL DEFINED AS char]

僅適用於 Hive 0.13 及 TEXTFILE 的 STORED AS 檔案格式。

--或--

- SERDE 'serde\_name' [WITH SERDEPROPERTIES ("property\_name" = "property\_value", "property\_name" = "property\_value" [, ...] )]

表 serde\_name 示 SerDe 要使用的。子 WITH SERDEPROPERTIES 句可讓您提供一或多個允許的自訂內容 SerDe。

[STORED AS file\_format]

指定資料表中資料的檔案格式。如果省略，TEXTFILE 為預設值。file\_format 的選項有：

- SEQUENCEFILE

- TEXTFILE
- RCFILE
- ORC
- PARQUET
- AVRO
- ION
- INPUTFORMAT `input_format_classname` OUTPUTFORMAT `output_format_classname`

[所在位置 's3://DOC-EXAMPLE-BUCKET/[folder]/']

指定 Amazon S3 中基礎資料的位置，資料表根據此資料建立。位置路徑必須是儲存貯體名稱或儲存貯體和一或多個資料夾。如果您使用的是分割區，請指定分割資料的根目錄。如需資料表位置的詳細資訊，請參閱 [Amazon S3 中的資料表位置](#)。如需資料格式和許可的資訊，請參閱 [Athena 資料表和 Amazon S3 資料的要求](#)。

為您的資料夾或儲存貯體使用尾隨斜線。請勿使用檔案名稱或 glob 字元。

使用：

```
s3://DOC-EXAMPLE-BUCKET/
```

```
s3://DOC-EXAMPLE-BUCKET/folder/
```

```
s3://DOC-EXAMPLE-BUCKET/folder/anotherfolder/
```

不要使用：

```
s3://DOC-EXAMPLE-BUCKET
```

```
s3://DOC-EXAMPLE-BUCKET/*
```

```
s3://DOC-EXAMPLE-BUCKET/mydatafile.dat
```

```
[TBLPROPERTIES ( ['has_encrypted_data']='true | false',] ['classification']='classification_value',]
property_name=property_value [, ...] ) ]
```

除了預先定義的資料表屬性 (例如 "comment")，也為資料表定義指定自訂中繼資料鍵值組。

`has_encrypted_data` – Athena 有一個內建的屬性 `has_encrypted_data`。將此屬性設為 `true`，表示 LOCATION 指定的基礎資料集都會加密。如果省略且工作群組的設定不會覆寫用戶端設定，則

假定為 false。如果省略或設為 false 時基礎資料有加密，查詢會導致錯誤。如需詳細資訊，請參閱 [靜態加密](#)。

分類 — 在 CloudTrail 主控台中為 Athena 建立的表格會新增 cloudtrail 為 classification 屬性的值。若要執行 ETL 工作，您 AWS Glue 必須建立具有 classification 內容的資料表，以指出 AWS Glue 為 csv、parquet 或 orc 的資料類型。例如：'classification'='csv'。如果您沒有指定此屬性，ETL 工作會失敗。之後，您可以使用 AWS Glue 主控台、API 或 CLI 來進行指定。如需詳細資訊，請參閱 [AWS Glue 開發人員指南中的使用與 AWS Glue Athena 的 ETL 工作和在 AWS Glue 中編寫工作](#)。

compression\_level – compression\_level 屬性指定要使用的壓縮級別。此屬性僅適用於 ZSTD 壓縮。可能的值為 1 到 22。預設值為 3。如需詳細資訊，請參閱 [在 Athena 中使用 ZSTD 壓縮級別](#)。

如需有關其他資料表屬性的詳細資訊，請參閱 [ALTER TABLE SET TBLPROPERTIES](#)。

## 範例

下列範例 CREATE TABLE 陳述式會根據 Amazon S3 中存放的標籤分隔行星資料建立資料表。

```
CREATE EXTERNAL TABLE planet_data (  
  planet_name string,  
  order_from_sun int,  
  au_to_sun float,  
  mass float,  
  gravity_earth float,  
  orbit_years float,  
  day_length float  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE  
LOCATION 's3://DOC-EXAMPLE-BUCKET/tsv/'
```

請注意以下重點：

- 該 ROW FORMAT DELIMITED 子句表示數據由特定字符分隔。
- 該 FIELDS TERMINATED BY '\t' 子句指定 TSV 數據中的字段由製表符 ('\t') 分隔。
- 該 STORED AS TEXTFILE 子句指出資料是以純文字檔案形式存放在 Amazon S3 中。

要查詢數據，您可以使用如下簡單的SELECT語句：

```
SELECT * FROM planet_data
```

若要使用此範例在 Athena 中建立您自己的 TSV 表格，請以您自己表格和欄的名稱和資料類型取代資料表和資料類型，然後更新LOCATION子句以指向儲存 TSV 檔案的 Amazon S3 路徑。

如需建立資料表的詳細資訊，請參閱 [在 Athena 中建立資料表](#)。

## CREATE TABLE AS

建立新的資料表並填入 [SELECT](#) 查詢的結果。若要建立空白資料表，請使用 [CREATE TABLE](#)。CREATE TABLE AS 結合使用 CREATE TABLE DDL 陳述式與 SELECT DML 陳述式，因此技術上同時包含 DDL 和 DML。請注意，雖然 CREATE TABLE AS 在此處與其他 DDL 陳述式一起分組，但是針對 Service Quotas 而言，Athena 中的 CTAS 查詢仍會被視為 DML。如需有關 Athena 中的 Service Quotas 的相關資訊，請參閱 [Service Quotas](#)。

### Note

對於 CTAS 陳述式，預期的儲存貯體擁有者設定不適用於 Simple Storage Service (Amazon S3) 中的目的地資料表位置。預期的儲存貯體擁有者設定僅適用於您為 Athena 查詢結果指定的 Simple Storage Service (Amazon S3) 輸出位置。如需詳細資訊，請參閱 [使用 Athena 主控台指定查詢結果位置](#)。

如需有關超出此參考主題範圍的 CREATE TABLE AS 的其他資訊，請參閱 [從查詢結果建立資料表 \(CTAS\)](#)。

### 主題

- [概要](#)
- [CTAS 資料表屬性](#)
- [範例](#)

### 概要

```
CREATE TABLE table_name  
[ WITH ( property_name = expression [, ...] ) ]  
AS query
```

```
[ WITH [ NO ] DATA ]
```

其中：

```
WITH ( property_name = expression [, ...] )
```

CTAS 資料表選用屬性清單，其中一些是特定於資料儲存體格式。請參閱[CTAS 資料表屬性](#)。

query

用於建立新資料表的 [SELECT](#) 查詢。

#### Important

如果您打算以分割區建立查詢，在 SELECT 陳述式的欄清單最後指定分割區各欄的名稱。

```
[ WITH [ NO ] DATA ]
```

如果使用 WITH NO DATA，新的空資料表具有的結構描述會與建立的原始資料表相同。

#### Note

若要將資料欄標題包含在查詢結果輸出中，您可以使用簡單的 SELECT 查詢而不是 CTAS 查詢。您可以從查詢結果位置擷取結果，或直接使用 Athena 主控台下載結果。如需詳細資訊，請參閱 [使用查詢結果、近期查詢和輸出檔案](#)。

## CTAS 資料表屬性

Athena 中的每個 CTAS 資料表會有一個您使用 WITH (property\_name = expression [, ...] ) 指定的 CTAS 資料表選用屬性清單。如需這些參數的用法詳細資訊，請參閱 [CTAS 查詢的範例](#)。

```
WITH (property_name = expression [, ...], )  
      table_type = ['HIVE', 'ICEBERG']
```

選用。預設值為 HIVE。指定產生之資料表的資料表類型

範例：

```
WITH (table_type = 'ICEBERG')
```

## **external\_location = [location]**

### Note

由於 Iceberg 資料表並非外部資料表，因此此屬性不適用於 Iceberg 資料表。若要在 CTAS 陳述式中定義 Iceberg 資料表的根位置，請使用本節稍後所述的 `location` 屬性。

選用。Athena 在 Amazon S3 中儲存您的 CTAS 查詢的位置。

範例：

```
WITH (external_location = 's3://DOC-EXAMPLE-BUCKET/tables/parquet_table/')
```

Athena 不會對查詢結果使用相同的路徑兩次。如果您手動指定位置，請確定您指定的 Amazon S3 位置沒有資料。Athena 絕不會嘗試刪除您的資料。如果您想要再次使用相同的位置，請手動刪除資料，否則您的 CTAS 查詢將會失敗。

如果您執行一個在[強制執行查詢結果位置](#)的工作群組中指定 `external_location` 的 CTAS 查詢，則該查詢會失敗並顯示錯誤訊息。若要查看為工作群組指定的查詢結果位置，請[檢視工作群組的詳細資訊](#)。

如果您的工作群組覆寫查詢結果位置的用戶端設定，則 Athena 會在以下位置建立資料表：

```
s3://DOC-EXAMPLE-BUCKET/tables/query-id/
```

如果您未使用 `external_location` 屬性來指定位置，且您的工作群組未覆寫用戶端設定，則 Athena 會使用查詢結果位置的[用戶端設定](#)，在下列位置建立資料表：

```
s3://DOC-EXAMPLE-BUCKET/Unsaved-or-query-name/year/month/date/tables/query-id/
```

## **is\_external = [boolean]**

選用。指出資料表是否為外部資料表。預設值為 `true`。對於 Iceberg 資料表，必須將其設定為 `false`。

範例：

```
WITH (is_external = false)
```

### **location = [location]**

Iceberg 資料表的必要項目。指定要從查詢結果建立之 Iceberg 資料表的根位置。

範例：

```
WITH (location = 's3://DOC-EXAMPLE-BUCKET/tables/iceberg_table/')
```

### **field\_delimiter = [delimiter]**

選用，專門用於文字型資料儲存格式。CSV、TSV、文字檔案中檔案的單一字元欄位分隔符號。例如：WITH (field\_delimiter = ',')。目前，CTAS 查詢不支援多字元欄位分隔符號。如果不指定欄位分隔符號，預設會使用 \001。

### **format = [storage\_format]**

CTAS 查詢結果的儲存格式，例如 ORC、PARQUET、AVRO、JSON、ION 或 TEXTFILE。對於 Iceberg 資料表，允許的格式為 ORC、PARQUET 和 AVRO。若省略，預設會使用 PARQUET。此參數名稱 format 必須小寫，否則您的 CTAS 查詢將會失敗。

範例：

```
WITH (format = 'PARQUET')
```

### **bucketed\_by = ARRAY[ column\_name[,...], bucket\_count = [int] ]**

#### Note

此屬性不適用於 Iceberg 資料表。對於 Iceberg 資料表，請使用含儲存貯體轉換的分割。

將歸納資料的儲存貯體陣列清單。如果省略，Athena 不會歸納此查詢中的資料。

**bucket\_count = [int]****Note**

此屬性不適用於 Iceberg 資料表。對於 Iceberg 資料表，請使用含儲存貯體轉換的分割。

歸納您資料的儲存貯體數目。如果省略，Athena 不會歸納您的資料。範例：

```
CREATE TABLE bucketed_table WITH (
  bucketed_by = ARRAY[column_name],
  bucket_count = 30, format = 'PARQUET',
  external_location = 's3://DOC-EXAMPLE-BUCKET/tables/parquet_table/'
) AS
SELECT
  *
FROM
  table_name
```

**partitioned\_by = ARRAY[ col\_name[,...] ]****Note**

此屬性不適用於 Iceberg 資料表。若要針對 Iceberg 資料表使用分割區轉換，請使用本節稍後所述的 partitioning 屬性。

選用。欄的陣列清單，CTAS 資料表將依此分割。請確定分割區各欄的名稱，是列 SELECT 陳述式中欄清單的最後一欄。

**partitioning = ARRAY[partition\_transform, ...]**

選用。指定要建立之 Iceberg 資料表的分割區。Iceberg 支援多種分割區轉換和分割區演變。下表總結說明了分割區轉換。

轉換	描述
year(ts)	每年建立一個分割區。分割區值是介於 ts 和 1970 年 1 月 1 日間年份的整數差異。

轉換	描述
month(ts)	每年每月建立一個分割區。分割區值是介於 ts 和 1970 年 1 月 1 日間月份的整數差異。
day(ts)	每年每天建立一個分割區。分割區值是介於 ts 和 1970 年 1 月 1 日間日期的整數差異。
hour(ts)	每天每小時建立一個分割區。分割區值是分鐘和秒數設定為零的時間戳記。
bucket(x, nbuckets)	將資料雜湊處理為指定儲存貯體數目。分割區值為 x 的整數雜湊值，其值介於 0 到 nbuckets - 1 (含) 之間。
truncate(s, nchars)	將分割區值設為 s 的前 nchars 個字元。

範例：

```
WITH (partitioning = ARRAY['month(order_date)',
                           'bucket(account_number, 10)',
                           'country']))
```

### **optimize\_rewrite\_min\_data\_file\_size\_bytes = [long]**

選用。資料最佳化的專用組態。包括小於指定值的檔案以進行最佳化。預設為 write\_target\_data\_file\_size\_bytes 值的 0.75 倍。此屬性僅適用於 Iceberg 資料表。如需詳細資訊，請參閱 [最佳化處理 Iceberg 資料表](#)。

範例：

```
WITH (optimize_rewrite_min_data_file_size_bytes = 402653184)
```

### **optimize\_rewrite\_max\_data\_file\_size\_bytes = [long]**

選用。資料最佳化的專用組態。包括大於指定值的檔案以進行最佳化。預設為 write\_target\_data\_file\_size\_bytes 值的 1.8 倍。此屬性僅適用於 Iceberg 資料表。如需詳細資訊，請參閱 [最佳化處理 Iceberg 資料表](#)。

範例：

```
WITH (optimize_rewrite_max_data_file_size_bytes = 966367641)
```

### **optimize\_rewrite\_data\_file\_threshold = [int]**

選用。資料最佳化的專用組態。如果需要最佳化的資料檔案少於指定的閾值，則不會重寫這些檔案。這允許累積更多的資料檔案，從而產生更接近目標大小的檔案，並略過不必要的運算以節省成本。預設值為 5。此屬性僅適用於 Iceberg 資料表。如需詳細資訊，請參閱 [最佳化處理 Iceberg 資料表](#)。

範例：

```
WITH (optimize_rewrite_data_file_threshold = 5)
```

### **optimize\_rewrite\_delete\_file\_threshold = [int]**

選用。資料最佳化的專用組態。如果與資料檔案關聯的刪除檔案少於閾值，則不會重寫資料檔案。這允許為每個資料檔案累積更多的刪除檔案，從而節省成本。預設為 2。此屬性僅適用於 Iceberg 資料表。如需詳細資訊，請參閱 [最佳化處理 Iceberg 資料表](#)。

範例：

```
WITH (optimize_rewrite_delete_file_threshold = 2)
```

### **vacuum\_min\_snapshots\_to\_keep = [int]**

選用。清空特定組態。要保留之最新快照的數目下限。預設為 1。此屬性僅適用於 Iceberg 資料表。如需詳細資訊，請參閱 [VACUUM](#)。

#### Note

`vacuum_min_snapshots_to_keep` 屬性需要 Athena 引擎版本 3。

範例：

```
WITH (vacuum_min_snapshots_to_keep = 1)
```

### **vacuum\_max\_snapshot\_age\_seconds = [long]**

選用。清空特定組態。以秒為單位的期間，代表要保留之快照的存留期。預設為 432,000 (5 天)。此屬性僅適用於 Iceberg 資料表。如需詳細資訊，請參閱 [VACUUM](#)。

**Note**

`vacuum_max_snapshot_age_seconds` 屬性需要 Athena 引擎版本 3。

範例：

```
WITH (vacuum_max_snapshot_age_seconds = 432000)
```

**write\_compression = [compression\_format]**

用於任何允許指定壓縮的儲存格式的壓縮類型。`compression_format` 值將指定資料寫入資料表時要使用的壓縮。您可以為 TEXTFILE、JSON、PARQUET，以及 ORC 檔案格式指定壓縮。

例如，如果 `format` 屬性指定 PARQUET 作為儲存格式，則 `write_compression` 的值會指定 Parquet 的壓縮格式。在這種情況下，指定 `write_compression` 的值相當於為 `parquet_compression` 指定值。

同樣地，如果 `format` 屬性指定 ORC 作為儲存格式，則 `write_compression` 的值會指定 ORC 的壓縮格式。在這種情況下，指定 `write_compression` 的值相當於為 `orc_compression` 指定值。

無法在相同的 CTAS 查詢中指定多個壓縮格式資料表屬性。例如，您無法在相同查詢中同時指定 `write_compression` 和 `parquet_compression`。同樣適用於 `write_compression` 和 `orc_compression`。如需每種檔案格式支援之壓縮類型的詳細資訊，請參閱 [Athena 壓縮支援](#)。

**orc\_compression = [compression\_format]**

將 ORC 資料寫入資料表時用於 ORC 檔案格式的壓縮類型。例如 `WITH (orc_compression = 'ZLIB')`。ORC 檔案中的區塊 (ORC Postscript 除外) 使用您指定的壓縮進行壓縮。若省略，ORC 預設使用 ZLIB 壓縮。

**Note**

為取得一致，我們建議您使用 `write_compression` 屬性，而不是 `orc_compression`。使用 `format` 屬性將儲存格式指定為 ORC，然後使用 `write_compression` 屬性指定 ORC 將使用的壓縮格式。

## `parquet_compression = [compression_format]`

將 Parquet 資料寫入資料表時，用於 Parquet 檔案格式的壓縮類型。例如 WITH (`parquet_compression = 'SNAPPY'`)。此壓縮會套用至 Prquet 檔案中的資料欄區塊。若省略，Parquet 預設使用 GZIP 壓縮。

### Note

為取得一致，我們建議您使用 `write_compression` 屬性，而不是 `parquet_compression`。使用 `format` 屬性將儲存格式指定為 PARQUET，然後使用 `write_compression` 屬性指定 PARQUET 將使用的壓縮格式。

## `compression_level = [compression_level]`

要使用的壓縮級別。此屬性僅適用於 ZSTD 壓縮。可能的值為 1 到 22。預設值為 3。如需詳細資訊，請參閱 [在 Athena 中使用 ZSTD 壓縮級別](#)。

## 範例

如需 CTAS 查詢的範例，請參閱下列資源。

- [CTAS 查詢的範例](#)
- [使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析](#)
- [使用 CTAS 陳述式搭配 Amazon Athena 來降低成本並改善效能](#)
- [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)

## CREATE VIEW

從指定的 SELECT 查詢建立新的檢視。檢視是可供未來查詢參考的一種邏輯資料表。檢視不會包含任何資料，而且不會寫入資料。相反地，檢視指定的查詢會在每次您透過另一個查詢參考該檢視時執行。

### Note

本主題提供摘要資訊以供參考。如需有關在 Athena 中使用檢視的詳細資訊，請參閱 [使用檢視](#)。如需有關檢視限制的資訊，請參閱 [檢視限制](#)。

## 概要

```
CREATE [ OR REPLACE ] VIEW view_name AS query
```

選用的 OR REPLACE 子句可讓您透過取代來更新現有的檢視。如需詳細資訊，請參閱 [建立檢視](#)。

## 範例

若要從資料表 orders 建立檢視 test，請使用類似以下的查詢：

```
CREATE VIEW test AS
SELECT
orderkey,
orderstatus,
totalprice / 2 AS half
FROM orders;
```

若要從資料表 orders 建立檢視 orders\_by\_date，請使用以下的查詢：

```
CREATE VIEW orders_by_date AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
GROUP BY orderdate;
```

若要更新現有的檢視，請使用類似以下的範例：

```
CREATE OR REPLACE VIEW test AS
SELECT orderkey, orderstatus, totalprice / 4 AS quarter
FROM orders;
```

另請參閱 [SHOW COLUMNS](#)、[SHOW CREATE VIEW](#)、[DESCRIBE VIEW](#) 和 [DROP VIEW](#)。

## DESCRIBE

顯示指定資料表的一個或多個資料欄，包括分割區欄位。此指令有助於檢查複雜資料欄的屬性。

## 概要

```
DESCRIBE [EXTENDED | FORMATTED] [db_name.]table_name [PARTITION partition_spec]
[col_name ( [.field_name] | [.'$elem$'] | [.'$key$'] | [.'$value$'] )]
```

**⚠ Important**

此陳述式的語法是 `DESCRIBE table_name`，而不是 `DESCRIBE TABLE table_name`。使用後一種語法會導致錯誤訊息失敗：SemanticException [錯誤 10001]：找不到資料表。

**參數**

[EXTENDED | FORMATTED]

判斷輸出的格式。省略這些參數會以表格格式顯示資料欄名稱及其對應的資料類型，包括分割區欄位。指定 FORMATTED 不僅以表格格式顯示資料欄名稱和資料類型，而且還顯示詳細的資料表和儲存資訊。EXTENDED 會以表格格式顯示資料欄和資料類型資訊，並以 Thrift 序列化格式顯示資料表的詳細中繼資料。此格式不易讀取，並且主要用於偵錯。

[PARTITION *partition\_spec*]

如果包含，則會列出由 *partition\_spec* 指定之分割區中繼資料，其中 *partition\_spec* 的格式為 (*partition\_column* = *partition\_col\_value*, *partition\_column* = *partition\_col\_value*, ...)。

[*col\_name* ( [*field\_name*] | [ '\$elem\$' ] | [ '\$key\$' ] | [ '\$value\$' ] ) \* ]

指定要檢查的資料欄和屬性。您可以為結構的元素指定 *.field\_name*，為陣列元素指定 '\$elem \$'，為對應金鑰指定 '\$key\$'，以及為對應值指定 '\$value\$'。您可以以遞迴的方式指定，以進一步探索複雜的資料欄。

**範例**

```
DESCRIBE orders
```

```
DESCRIBE FORMATTED mydatabase.mytable PARTITION (part_col = 100) columnA;
```

以下查詢和輸出顯示了基於 Amazon EMR 範例資料的 impressions 資料表的資料欄和資料類型。

```
DESCRIBE impressions
```

requestbegintime	string	from
deserializer		

```

adid                string                from
  deserializer
impressionid       string                from
  deserializer
referrer           string                from
  deserializer
useragent          string                from
  deserializer
usercookie         string                from
  deserializer
ip                 string                from
  deserializer
number             string                from
  deserializer
processid          string                from
  deserializer
browsercookie     string                from
  deserializer
requestendtime     string                from
  deserializer
timers             struct<modelllookup:string,requesttime:string> from
  deserializer
threadid           string                from
  deserializer
hostname           string                from
  deserializer
sessionid          string                from
  deserializer
dt                 string

# Partition Information
# col_name          data_type            comment
dt                 string

```

以下範例查詢和輸出顯示了使用 FORMATTED 選項時相同資料表的結果。

```
DESCRIBE FORMATTED impressions
```

```

requestbeginntime  string                from
  deserializer
adid                string                from
  deserializer

```

```

impressionid      string      from
  deserializer
referrer          string      from
  deserializer
useragent         string      from
  deserializer
usercookie        string      from
  deserializer
ip                string      from
  deserializer
number            string      from
  deserializer
processid         string      from
  deserializer
browsercookie     string      from
  deserializer
requestendtime    string      from
  deserializer
timers            struct<modelllookup:string,requesttime:string> from
  deserializer
threadid          string      from
  deserializer
hostname          string      from
  deserializer
sessionid         string      from
  deserializer
dt                string

```

## # Partition Information

```
# col_name      data_type      comment
```

```
dt              string
```

## # Detailed Table Information

```

Database:      sampledb
Owner:         hadoop
CreateTime:    Thu Apr 23 02:55:21 UTC 2020
LastAccessTime: UNKNOWN
Protect Mode:  None
Retention:     0
Location:      s3://us-east-1.elasticmapreduce/samples/hive-ads/tables/
impressions
Table Type:    EXTERNAL_TABLE
Table Parameters:

```

```

EXTERNAL                                TRUE
transient_lastDdlTime                   1587610521

# Storage Information
SerDe Library:                          org.openx.data.jsonserde.JsonSerDe
InputFormat:                            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:                            org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat
Compressed:                              No
Num Buckets:                             -1
Bucket Columns:                          []
Sort Columns:                             []
Storage Desc Params:
  paths                                  requestbegintime, adid, impressionid,
referrer, useragent, usercookie, ip
  serialization.format                    1

```

以下範例查詢和輸出顯示了使用 EXTENDED 選項時相同資料表的結果。資料表詳細資訊會在單一行上輸出，但在此處進行格式化以增加可讀性。

```
DESCRIBE EXTENDED impressions
```

```

requestbegintime      string      from
  deserializer
adid                  string      from
  deserializer
impressionid         string      from
  deserializer
referrer              string      from
  deserializer
useragent             string      from
  deserializer
usercookie            string      from
  deserializer
ip                    string      from
  deserializer
number                string      from
  deserializer
processid             string      from
  deserializer
browsercookie        string      from
  deserializer

```

```

requestendtime      string      from
  deserializer
timers              struct<modelllookup:string,requesttime:string> from
  deserializer
threadid           string      from
  deserializer
hostname           string      from
  deserializer
sessionid          string      from
  deserializer
dt                 string

```

#### # Partition Information

```
# col_name      data_type      comment
```

```
dt              string
```

```

Detailed Table Information      Table(tableName:impressions, dbName:sampled,
  owner:hadoop, createTime:1587610521,
  lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:
  [FieldSchema(name:requeststarttime, type:string, comment:null),
  FieldSchema(name:adid, type:string, comment:null), FieldSchema(name:impressionid,
  type:string, comment:null),
  FieldSchema(name:referrer, type:string, comment:null), FieldSchema(name:useragent,
  type:string, comment:null),
  FieldSchema(name:usercookie, type:string, comment:null), FieldSchema(name:ip,
  type:string, comment:null),
  FieldSchema(name:number, type:string, comment:null), FieldSchema(name:processid,
  type:string, comment:null),
  FieldSchema(name:browsercookie, type:string, comment:null),
  FieldSchema(name:requestendtime, type:string, comment:null),
  FieldSchema(name:timers, type:struct<modelllookup:string,requesttime:string>,
  comment:null), FieldSchema(name:threadid,
  type:string, comment:null), FieldSchema(name:hostname, type:string, comment:null),
  FieldSchema(name:sessionid,
  type:string, comment:null)], location:s3://us-east-1.elasticmapreduce/samples/hive-ads/
  tables/impressions,
  inputFormat:org.apache.hadoop.mapred.TextInputFormat,
  outputFormat:org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat, compressed:false,
  numBuckets:-1,
  serdeInfo:SerDeInfo(name:null, serializationLib:org.openx.data.jsonserde.JsonSerDe,
  parameters:{serialization.format=1,
  paths=requeststarttime, adid, impressionid, referrer, useragent, usercookie, ip}),
  bucketCols:[], sortCols:[], parameters:{}),

```

```
skewedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[],
  skewedColValueLocationMaps:{}),
storedAsSubDirectories:false), partitionKeys:[FieldSchema(name:dt, type:string,
  comment:null)],
parameters:{EXTERNAL=TRUE, transient_lastDdlTime=1587610521}, viewOriginalText:null,
  viewExpandedText:null,
tableType:EXTERNAL_TABLE)
```

## DESCRIBE VIEW

顯示指定檢視的資料欄清單。這可讓您檢查複雜檢視的屬性。

### 概要

```
DESCRIBE [db_name.]view_name
```

### 範例

```
DESCRIBE orders;
```

另請參閱 [SHOW COLUMNS](#)、[SHOW CREATE VIEW](#)、[SHOW VIEWS](#) 和 [DROP VIEW](#)。

## DROP DATABASE

從目錄中移除指定的資料庫。如果資料庫包含資料表，在執行 DROP DATABASE 之前，您必須捨棄資料表或使用 CASCADE 子句。DATABASE 和 SCHEMA 可互換使用。它們的意義相同。

### 概要

```
DROP {DATABASE | SCHEMA} [IF EXISTS] database_name [RESTRICT | CASCADE]
```

### 參數

#### [IF EXISTS]

如果 database\_name 不存在，則會造成錯誤隱藏。

#### [RESTRICT|CASCADE]

決定在 DROP 操作期間，對待 database\_name 內資料表的方式。如果您指定 RESTRICT，則當資料庫包含資料表時，不會將資料庫捨棄。這是預設行為。指定 CASCADE 會造成將資料庫及其所有資料表捨棄。

## 範例

```
DROP DATABASE clickstreams;
```

```
DROP SCHEMA IF EXISTS clickstreams CASCADE;
```

### Note

當您嘗試刪除名稱包含特殊字元 (例如 my-database) 的資料庫時，可能會收到錯誤訊息。若要解決這個問題，請嘗試將資料庫名稱放在反引號 ( ` ) 字元之間。如需有關 Athena 中命名資料庫的資訊，請參閱[資料表、資料庫和資料欄的名稱](#)。

## DROP TABLE

移除名為 table\_name 資料表的中繼資料資料表定義。捨棄外部資料表時，基礎資料保持不變。

### 概要

```
DROP TABLE [IF EXISTS] table_name
```

### 參數

#### [ IF EXISTS ]

如果 table\_name 不存在，則會造成錯誤隱藏。

### 範例

```
DROP TABLE fulfilled_orders
```

```
DROP TABLE IF EXISTS fulfilled_orders
```

使用 Athena 主控台查詢編輯器來捨棄具有底線 ( \_ ) 以外之特殊字元的資料表時，請使用反引號，如下列範例所示。

```
DROP TABLE `my-athena-database-01.my-athena-table`
```

使用 JDBC 連接器來卸除具有特殊字元的資料表時，不需要反引號字元。

```
DROP TABLE my-athena-database-01.my-athena-table
```

## DROP VIEW

捨棄 (刪除) 現有的檢視。如果檢視不存在，則選用的 IF EXISTS 子句會造成錯誤隱藏。

如需詳細資訊，請參閱 [使用檢視](#)。

### 概要

```
DROP VIEW [ IF EXISTS ] view_name
```

### 範例

```
DROP VIEW orders_by_date
```

```
DROP VIEW IF EXISTS orders_by_date
```

另請參閱 [CREATE VIEW](#)、[SHOW COLUMNS](#)、[SHOW CREATE VIEW](#)、[SHOW VIEWS](#) 和 [DESCRIBE VIEW](#)。

## MSCK REPAIR TABLE

新增 Hive 相容分割區之後，請使用 MSCK REPAIR TABLE 命令來更新目錄中的中繼資料。

MSCK REPAIR TABLE 命令會掃描檔案系統 (例如 Amazon S3)，以查找在建立資料表後新增到檔案系統的 Hive 相容分隔區。MSCK REPAIR TABLE 會比較資料表中繼資料中的分割區和 S3 中的分割區。如果您在建立資料表時指定的 S3 位置有新的分割區，它會將這些分割區新增至中繼資料和 Athena 資料表中。

當您新增實體分割區時，目錄中的中繼資料會變得與檔案系統中的資料配置不一致，而且需要將新分割區的相關資訊新增至目錄。若要更新中繼資料，請執行 MSCK REPAIR TABLE，如此可讓您從 Athena 查詢新分割區中的資料。

### Note

MSCK REPAIR TABLE 只會將分割區新增至中繼資料，而不會刪除它們。若要在 Amazon S3 中手動刪除分割區後從中繼資料移除分割區，請執行 ALTER TABLE *table-name* DROP PARTITION 命令。如需詳細資訊，請參閱 [ALTER TABLE DROP PARTITION](#)。

## 考量與限制

使用 MSCK REPAIR TABLE 時，請謹記以下幾點：

- 新增所有分割區可能需要花一些時間。如果此操作逾時，則會處於不完整狀態，只有幾個分割區新增到目錄。您應該在同一個資料表上執行 MSCK REPAIR TABLE，直到所有分割區都已新增為止。如需詳細資訊，請參閱 [在 Athena 中分割資料](#)。
- 對於與 Hive 不相容的分割區，請使用 [ALTER TABLE ADD PARTITION](#) 載入分割區，以便查詢資料。
- 要搭配 Athena 使用的分割區位置必須使用 s3 通訊協定 (例如，s3://DOC-EXAMPLE-BUCKET/*folder*)。在 Athena 中，當在包含的資料表上執行 MSCK REPAIR TABLE 查詢時，使用其他通訊協定的位置 (例如 s3a://*bucket/folder*) 會導致查詢失敗。
- 由於 MSCK REPAIR TABLE 會同時掃描資料夾及其子資料夾，以尋找相符的分割區配置，請務必將個別資料表的資料留在不同的資料夾階層中。例如，假設您在中表格 1 的資料 s3://DOC-EXAMPLE-BUCKET1，在中表格 2 的資料 s3://DOC-EXAMPLE-BUCKET1/table-2-data。如果兩個資料表都依字串分割，則 MSCK REPAIR TABLE 會將資料表 2 的分割區新增至資料表 1。為了避免這種情況，請 s3://DOC-EXAMPLE-BUCKET2 改用單獨的 s3://DOC-EXAMPLE-BUCKET1 文件夾結構。請注意，此行為與 Amazon EMR 和 Apache Hive 一致。
- 由於已知問題的緣故，只要分割區的值中含有冒號 (:) 字元 (例如分割區的值為時間戳記)，MSCK REPAIR TABLE 就會執行失敗且未出現任何提示。解決方法是使用 [ALTER TABLE ADD PARTITION](#)。
- MSCK REPAIR TABLE 不會新增以底線 (\_) 開頭的分割區資料欄名稱。若要解決此限制，請使用 [ALTER TABLE ADD PARTITION](#)。

## 概要

```
MSCK REPAIR TABLE table_name
```

## 範例

```
MSCK REPAIR TABLE orders;
```

## 故障診斷

執行之後 MSCK REPAIR TABLE，如果 Athena 未將分割區新增至中的表格 AWS Glue Data Catalog，請檢查下列各項：

- AWS Glue 存取 — 確定 AWS Identity and Access Management (IAM) 角色具有允許 `glue:BatchCreatePartition` 動作的政策。如需詳細資訊，請參閱本文件稍後的 [允許膠水：BatchCreatePartition 在 IAM 政策中](#)。
- Amazon S3 存取 – 確定角色具有足夠許可以存取 Amazon S3 的政策，包括 `s3:DescribeJob` 動作。如需允許哪些 Amazon S3 動作的範例，請參閱 [Athena 中的 Amazon S3 儲存貯體跨帳戶存取](#) 中的儲存貯體政策範例。
- Amazon S3 物件金鑰大小寫 – 請確定 Amazon S3 路徑是小寫而不是小駝峰式命名法 (例如，`userid` 而非 `userId`)，或使用 `ALTER TABLE ADD PARTITION` 指定物件金鑰名稱。如需詳細資訊，請參閱本文件稍後的 [變更或重新定義 Amazon S3 路徑](#)。
- 查詢逾時 – `MSCK REPAIR TABLE` 最適合在第一次建立資料表時使用，或者在資料和分割區中繼資料之間存不確定的同位時使用。如果您經常使用 `MSCK REPAIR TABLE` 新增分割區 (例如每天) 並正經歷查詢逾時，請考慮使用 [ALTER TABLE ADD PARTITION](#)。
- 檔案系統中遺失分割區 - 如果您在 Amazon S3 中手動刪除分割區，然後執行 `MSCK REPAIR TABLE`，則您可能會收到錯誤訊息檔案系統中遺失分割區。這是因為 `MSCK REPAIR TABLE` 不會從資料表中繼資料中移除過時的分割區。若要從資料表中繼資料移除已刪除的分割區，請改為執行 [ALTER TABLE DROP PARTITION](#)。請注意，[SHOW PARTITIONS](#) 同樣地只會列出中繼資料中的分割區，而不會列出檔案系統中的分割區。
- 「NullPointerException 名稱為空」錯誤

如果您使用 AWS Glue [CreateTable](#) API 作業或 AWS CloudFormation [AWS::Glue::Table](#) 範本建立要在 Athena 中使用的資料表，但未指定 `TableType` 屬性，然後執行類似 `SHOW CREATE TABLE` 或的 DDL 查詢 `MSCK REPAIR TABLE`，則可能會收到錯誤訊息 `FAILED: NullPointerException 名稱為空值`。

若要解決錯誤，請指定 [TableInput](#) `TableType` 屬性值做為 AWS Glue `CreateTable` API 呼叫或 [AWS CloudFormation 範本](#) 的一部分。`TableType` 可能的值包括 `EXTERNAL_TABLE` 或 `VIRTUAL_VIEW`。

只有當您使用 AWS Glue `CreateTable` API 作業或 [AWS::Glue::Table](#) 範本建立資料表時，才會套用此需求。如果您使用 DDL 陳述式或 AWS Glue 爬蟲程式建立 Athena 資料表，則系統會為您自動定義 `TableType` 屬性。

下列各節提供了一些額外的詳細資訊。

## 允許膠水：BatchCreatePartition 在 IAM 政策中

檢閱[連接至您用來執行 MSCK REPAIR TABLE 之角色的 IAM 政策](#)。當您[AWS Glue Data Catalog 搭配 Athena 使用](#)時，IAM 政策必須允許 `glue:BatchCreatePartition` 動作。如需允許 `glue:BatchCreatePartition` 動作的 IAM 政策範例，請參閱 [AWS 受管理的策略：AmazonAthenaFullAccess](#)。

## 變更或重新定義 Amazon S3 路徑

如果 Amazon S3 路徑中的一或多個物件金鑰是駝峰式大小寫，則 MSCK REPAIR TABLE 可能不會將分割區新增到 AWS Glue Data Catalog。例如，如果您的 Amazon S3 路徑包含物件金鑰名稱 `userId`，則下列分割區可能不會新增到 AWS Glue Data Catalog：

```
s3://DOC-EXAMPLE-BUCKET/path/userId=1/
s3://DOC-EXAMPLE-BUCKET/path/userId=2/
s3://DOC-EXAMPLE-BUCKET/path/userId=3/
```

要解決此問題，請執行下列項目之一：

- 建立 Amazon S3 物件金鑰時，請使用小寫字母而非駝峰式大小寫：

```
s3://DOC-EXAMPLE-BUCKET/path/userid=1/
s3://DOC-EXAMPLE-BUCKET/path/userid=2/
s3://DOC-EXAMPLE-BUCKET/path/userid=3/
```

- 使用 [ALTER TABLE ADD PARTITION](#) 重新定義位置，如下列範例所示：

```
ALTER TABLE table_name ADD [IF NOT EXISTS]
PARTITION (userId=1)
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/userId=1/'
PARTITION (userId=2)
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/userId=2/'
PARTITION (userId=3)
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/userId=3/'
```

請注意，雖然 Amazon S3 物件金鑰名稱可以使用大寫，但 Amazon S3 儲存貯體名稱本身必須始終為小寫。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[物件金鑰命名準則](#)和[儲存貯體命名規則](#)。

## SHOW COLUMNS

僅顯示單一指定資料表或檢視的資料欄名稱。若要取得更詳細的資訊，請 AWS Glue Data Catalog 改為查詢。如需資訊和範例，請參閱 [查詢 AWS Glue Data Catalog](#) 主題的下列章節：

- 若要檢視資料欄的中繼資料 (例如資料類型)，請參閱 [列示或搜尋指定的資料表或檢視的資料欄](#)。
- 若要檢視 AwsDataCatalog 中特定資料庫中的所有資料表的所有資料欄，請參閱 [列示或搜尋指定的資料表或檢視的資料欄](#)。
- 若要檢視 AwsDataCatalog 中所有資料庫的所有資料表的所有資料欄，請參閱 [列出所有資料表的所有資料欄](#)。
- 若要檢視資料庫中特定資料表的共同資料欄，請參閱 [列出特定表格共同的資料欄](#)。

### 概要

```
SHOW COLUMNS {FROM|IN} database_name.table_name
```

```
SHOW COLUMNS {FROM|IN} table_name [{FROM|IN} database_name]
```

FROM 和 IN 關鍵字可互換使用。如果 *table\_name* 或 *database\_name* 具有如連字號的特殊字元，則請使用引號括住名稱 (例如 `my-database`.`my-table`)。請勿使用單引號和雙引號括住 *table\_name* 或 *database\_name*。目前，不支援使用 LIKE 和模式比對表達式。

### 範例

下列相等範例會顯示 customers 資料庫中 orders 資料表的資料欄。前兩個範例假設 customers 為目前的資料庫。

```
SHOW COLUMNS FROM orders
```

```
SHOW COLUMNS IN orders
```

```
SHOW COLUMNS FROM customers.orders
```

```
SHOW COLUMNS IN customers.orders
```

```
SHOW COLUMNS FROM orders FROM customers
```

```
SHOW COLUMNS IN orders IN customers
```

## SHOW CREATE TABLE

分析名為 `table_name` 的現有資料表，以產生建立此資料表的查詢。

### 概要

```
SHOW CREATE TABLE [db_name.]table_name
```

### 參數

TABLE [db\_name.]table\_name

`db_name` 為選用參數。如果省略，內容會預設為目前的資料庫。

#### Note

資料表名稱是必要的。

### 範例

```
SHOW CREATE TABLE orderclickstoday;
```

```
SHOW CREATE TABLE `salesdata.orderclickstoday`;
```

### 故障診斷

如果您使用 AWS Glue [CreateTable](#) API 作業或 AWS CloudFormation [AWS::Glue::Table](#) 範本建立在 Athena 中使用的資料表，但未指定 `TableType` 屬性，然後執行類似 `SHOW CREATE TABLE` 或的 DDL 查詢 `MSCK REPAIR TABLE`，則可能會收到錯誤訊息 `FAILED: NullPointerException` 名稱為空值。

若要解決錯誤，請指定 [TableInput](#) `TableType` 屬性值做為 AWS Glue `CreateTable` API 呼叫或 [AWS CloudFormation 範本](#) 的一部分。`TableType` 可能的值包括 `EXTERNAL_TABLE` 或 `VIRTUAL_VIEW`。

只有當您使用 AWS Glue `CreateTable` API 作業或 `AWS::Glue::Table` 範本建立資料表時，才會套用此需求。如果您使用 DDL 陳述式或 AWS Glue 爬蟲程式建立 Athena 資料表，則系統會為您自動定義 `TableType` 屬性。

## SHOW CREATE VIEW

顯示用於建立指定檢視的 SQL 陳述式。

### 概要

```
SHOW CREATE VIEW view_name
```

### 範例

```
SHOW CREATE VIEW orders_by_date
```

另請參閱 [CREATE VIEW](#) 和 [DROP VIEW](#)。

## SHOW DATABASES

列出中繼存放區中定義的所有資料庫。您可以使用 `DATABASES` 或 `SCHEMAS`。它們的意義相同。

的程式設計相當於 `SHOW DATABASES` 是 [ListDatabases](#) Athena API 動作。中的等效方法 AWS SDK for Python (Boto3) 是 [列表資料庫](#)。

### 概要

```
SHOW {DATABASES | SCHEMAS} [LIKE 'regular_expression']
```

### 參數

[LIKE '*regular\_expression*']

從資料庫清單中篩選符合您指定的 *regular\_expression* 的資料庫。對於萬用字元比對，您可以使用 `.*` 組合，這會比對零到不限次數的任何字元。

## 範例

```
SHOW SCHEMAS;
```

```
SHOW DATABASES LIKE '.*analytics';
```

## SHOW PARTITIONS

以未排序的順序列出 Athena 資料表中的所有分割區。

### 概要

```
SHOW PARTITIONS table_name
```

- 若要在表格中顯示分割區，並以特定順序列出，請參閱 [查詢 AWS Glue Data Catalog](#) 頁面上的 [列出特定資料表的分割區](#) 部分。
- 若要檢視分割區的內容，請參閱 [在 Athena 中分割資料](#) 頁面上的 [查詢資料](#) 部分。
- SHOW PARTITIONS 不會列出 Athena 投影但未在 AWS Glue 目錄中註冊的分割區。如需分割區投影的相關資訊，請參閱 [使用 Amazon Athena 進行分割區投影](#)。
- SHOW PARTITIONS 會列出中繼資料中的分割區，而非實際檔案系統中的分割區。若要在 Amazon S3 中手動刪除分割區後更新中繼資料，請執行 [ALTER TABLE DROP PARTITION](#)。

### 範例

下列範例查詢會顯示 flight\_delays\_csv 資料表的分割區，該資料表顯示來自美國運輸部的航班表資料。如需有關 flight\_delays\_csv 資料表範例的詳細資料，請參閱 [LazySimpleSerDe 適用於 CSV、TSV 和自訂分隔檔案](#)。資料表依年份進行分割。

```
SHOW PARTITIONS flight_delays_csv
```

### 結果

```
year=2007
year=2015
year=1999
year=1993
year=1991
```

```
year=2003
year=1996
year=2014
year=2004
year=2011
...
```

下列範例查詢會顯示 `impressions` 資料表的分割區，該資料表包含網頁瀏覽資料範例。如需有關 `impressions` 資料表範例的詳細資料，請參閱[在 Athena 中分割資料](#)。該資料表依 `dt` (datetime) 資料欄進行分割。

```
SHOW PARTITIONS impressions
```

## 結果

```
dt=2009-04-12-16-00
dt=2009-04-13-18-15
dt=2009-04-14-00-20
dt=2009-04-12-13-00
dt=2009-04-13-02-15
dt=2009-04-14-12-05
dt=2009-04-14-06-15
dt=2009-04-12-21-15
dt=2009-04-13-22-15
...
```

## 以排序順序列出分割區

若要在結果清單中排列分割區的順序，請使用下列 `SELECT` 語法，而非 `SHOW PARTITIONS`。

```
SELECT * FROM database_name."table_name$partitions" ORDER BY column_name
```

下列查詢按排序順序顯示 `flight_delays_csv` 範例的分割區清單。

```
SELECT * FROM "flight_delays_csv$partitions" ORDER BY year
```

## 結果

```
year
1987
```

```
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
...
```

如需詳細資訊，請參閱[查詢 AWS Glue Data Catalog](#)頁面上的[列出特定資料表的分割區](#)區段。

## SHOW TABLES

列出資料庫中的所有基礎資料表和檢視。

### 概要

```
SHOW TABLES [IN database_name] ['regular_expression']
```

### 參數

[IN database\_name]

指定將列出其中資料表的 database\_name。如果省略，則假定為目前內容中的資料庫。

#### Note

如果 database\_name 使用[不支援的字元](#)，例如連字號，則 SHOW TABLES 可能會失敗。解決方法為可嘗試將資料庫名稱放在兩個反引號之間。

['regular\_expression']

從資料表清單中篩選符合您指定的 regular\_expression 的資料表。若要指示 AWSDataCatalog 資料表中的任何字元，您可以使用 \* 或 .\* 萬用字元表達式。對於 Apache Hive 資料庫，請使用 .\* 萬用字元表達式。若要指示字元之間的選擇，請使用 | 字元。

## 範例

Example - 顯示資料庫 **sampledb** 中的所有資料表

```
SHOW TABLES IN sampledb
```

### Results

```
alb_logs  
cloudfront_logs  
elb_logs  
flights_2016  
flights_parquet  
view_2016_flights_dfw
```

Example - 顯示 **sampledb** 中包含「航班」一詞的所有資料表的名稱

```
SHOW TABLES IN sampledb '*flights*'
```

### Results

```
flights_2016  
flights_parquet  
view_2016_flights_dfw
```

Example - 顯示 **sampledb** 中以「日誌」結尾的所有資料表的名稱

```
SHOW TABLES IN sampledb '*logs'
```

### Results

```
alb_logs  
cloudfront_logs  
elb_logs
```

## SHOW TBLPROPERTIES

列出指名的資料表的資料表屬性。

## 概要

```
SHOW TBLPROPERTIES table_name [('property_name')]
```

## 參數

`[('property_name')]`

如果包含，則只會列出名為 `property_name` 的屬性的值。

## 範例

```
SHOW TBLPROPERTIES orders;
```

```
SHOW TBLPROPERTIES orders('comment');
```

## SHOW VIEWS

列出指定的資料庫或目前資料庫 (如果省略資料庫名稱) 中的檢視。使用選用的 LIKE 子句搭配規則表達式，以限制檢視名稱的清單。

Athena 會傳回 STRING 類型值清單，其中每個值都是檢視名稱。

## 概要

```
SHOW VIEWS [IN database_name] [LIKE 'regular_expression']
```

## 參數

`[IN database_name]`

指定將列出其中檢視的 `database_name`。如果省略，則假定為目前內容中的資料庫。

`[LIKE 'regular_expression']`

從檢視清單中篩選符合您指定的 `regular_expression` 的檢視。只能使用萬用字元 \* (表示任何字元) 或 | (表示在字元之間選擇)。

## 範例

```
SHOW VIEWS;
```

```
SHOW VIEWS IN marketing_analytics LIKE 'orders*'
```

另請參閱 [SHOW COLUMNS](#)、[SHOW CREATE VIEW](#)、[DESCRIBE VIEW](#) 和 [DROP VIEW](#)。

## 在 Amazon Athena 中進行 SQL 查詢的考量事項與限制

在 Athena 中執行查詢時，請記住下列考量事項與限制：

- 預存程序 – 不支援預存程序。
- 分割區數量上限 – 您可以使用 CREATE TABLE AS SELECT (CTAS) 陳述式建立的分割區數量上限為 100。如需詳細資訊，請參閱 [CREATE TABLE AS](#)。如需因應措施，請參閱 [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)。
- 不支援的陳述式 – 不支援下列陳述式：
  - 不支援 CREATE TABLE LIKE。
  - 不支援 DESCRIBE INPUT 和 DESCRIBE OUTPUT。
  - MERGE 陳述式僅支援交易型資料表格式。如需詳細資訊，請參閱 [合併為](#)。
  - 不支援 UPDATE 陳述式。
- Trino 和 Presto 連接器 – 不支援 [Trino](#) 和 [Presto](#) 連接器。使用 Amazon Athena 聯合查詢來連接資料來源。如需詳細資訊，請參閱 [使用 Amazon Athena 聯合查詢](#)。
- 具有多個分割區的資料表逾時 – 在查詢具有數千個分割區的資料表時，Athena 可能會逾時。當資料表有許多非類型 string 的分割區時，就會發生這種情況。當您使用類型 string 時，Athena 會在中繼存放區層級剔除分割區。不過，在您使用其他資料類型時，Athena 會剔除伺服器端的分割區。您擁有的分割區越多，此程序所需的時間越長，您的查詢也越有可能逾時。如要解決此問題，請將您的分割區類型設定為 string，以便 Athena 在中繼存放區層級剔除分割區。這可以減少額外負荷，並防止查詢逾時。
- S3 Glacier 支援 – 如需有關查詢還原的 Amazon S3 Glacier 物件的資訊，請參閱 [查詢還原的 Amazon S3 Glacier 物件](#)。
- 檔案視為隱藏 – Athena 會將以底線 (\_) 或點 (.) 開頭的來源檔案視為隱藏。如果要解決這個限制，請將檔案重新命名。
- 資料行或資料欄大小限制 – 單一資料行或其資料欄的大小不可超過 32 MB。例如，當 CSV 或 JSON 檔案中的資料行包含 300 MB 的單一資料欄時，則會超過此限制。超過此限制時也會產生錯誤訊息。

Line too long in text file (文字檔案中的資料行過長)。若要解決此限制，請確定任何資料列中資料欄的資料總和小於 32 MB。

- LIMIT 子句上限 – 可指定給 LIMIT 子句的資料列數上限為

9223372036854775807。使用 ORDER BY 時，LIMIT 子句所支援的資料列數上限為 2147483647。超過此限制會導致顯示錯誤訊息 NOT\_SUPPORTED: ORDER BY LIMIT > 2147483647 is not supported (NOT\_SUPPORTED：不支援 ORDER BY LIMIT > 2147483647)。

- 資訊\_schema — 如果您有少量到中等的中繼資料，查詢information\_schema效能最高。AWS Glue 如果您有大量中繼資料，則可能會出現錯誤。如需有關查詢資料information\_schema庫中 AWS Glue 繼資料的資訊，請參閱[查詢 AWS Glue Data Catalog](#)。
- 陣列初始化 – 由於 Java 中的限制，無法在 Athena 中初始化具有超過 254 個引數的陣列。
- 隱藏的中繼資料欄 – 檢視不支援 Hive 或 Iceberg 隱藏的中繼資料欄 \$bucket、\$file\_modified\_time、\$file\_size 和 \$partition。如需有關在 Athena 中使用 \$path 中繼資料欄的資訊，請參閱 [在 Amazon S3 中取得來源資料的檔案位置](#)。

如需查詢字串長度上限、查詢逾時配額，以及作用中 DML 查詢數量配額的相關資訊，請參閱[Service Quotas](#)。

## Athena 中的疑難排解

Athena 團隊收集了以下客戶問題的疑難排解資訊。雖然不是非常詳盡，但其中包括關於效能、逾時和記憶體不足等一些常見問題的建議。

### 主題

- [CREATE TABLE AS SELECT \(CTAS\)](#)
- [資料檔案問題](#)
- [Linux Foundation Delta Lake 資料表](#)
- [聯合查詢](#)
- [JSON 相關錯誤](#)
- [MSCK REPAIR TABLE](#)
- [輸出問題](#)
- [Parquet 問題](#)
- [分割問題](#)
- [許可](#)

- [查詢語法問題](#)
- [查詢逾時問題](#)
- [調節問題](#)
- [檢視](#)
- [工作群組](#)
- [其他資源](#)
- [Athena 錯誤目錄](#)

## CREATE TABLE AS SELECT (CTAS)

### 並行的 CTAS 陳述式發生重複的資料

Athena 不會維持並行的 CTAS 驗證。請確定在同一時間相同的位置沒有重複的 CTAS 陳述式。即使 CTAS 或 INSERT INTO 陳述式失敗，孤立的資料仍可以留在陳述式中指定的資料位置。

## HIVE\_TOO\_MANY\_OPEN\_PARTITIONS

當您使用 CTAS 陳述式建立具有超過 100 個分割區的資料表時，您可能會收到錯誤

HIVE\_TOO\_MANY\_OPEN\_PARTITIONS: Exceeded limit of 100 open writers for partitions/buckets (HIVE\_TOO\_MANY\_OPEN\_PARTITIONS : 超過分割區 / 儲存貯體 100 個開啟寫入器的限制)。若要避開此限制，您可以使用 CTAS 陳述式和一系列的 INSERT INTO 陳述式，每個陳述式可建立或插入最多 100 個分割區。如需詳細資訊，請參閱 [使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制](#)。

## 資料檔案問題

### Athena 無法讀取隱藏的檔案

Athena 會將以底線 (\_) 或點 (.) 開頭的來源檔案視為隱藏。如果要解決這個限制，請將檔案重新命名。

### Athena 讀取我從 AWS Glue 爬蟲中排除的文件

Athena 無法辨識您指定 AWS Glue 爬行者程式的[排除模式](#)。例如，如果您的 Simple Storage Service (Amazon S3) 儲存貯體同時包含 .csv 和 .json 檔案，並且您將 .json 檔案從爬蟲程式中排除，Athena 會查詢這兩個檔案群組。若要避免這種情況，請將您要排除的檔案放在不同的位置。

## HIVE\_BAD\_DATA : 剖析欄位值時發生錯誤

在下列情況下可能會發生這個錯誤：

- 資料表中定義的資料類型與來源資料不相符，或單一欄位包含不同類型的資料。如需建議的解決方案，請參閱 AWS 知識中心中的[我的 Amazon Athena 查詢失敗，出現錯誤「HIVE\\_BAD\\_DATA：剖析欄位 X 的欄位值時發生錯誤：對於輸入字串："12312845691"」](#)。
- 整數欄位中存在 Null 值。一種解決方法是建立帶有 null 值的資料欄作為 string，然後使用 CAST 來轉換查詢中的欄位，並為 null 提供 0 預設值。如需詳細資訊，請參閱 AWS 知識中心中的[當我在 Athena 中查詢 CSV 資料時，出現錯誤「HIVE\\_BAD\\_DATA：剖析欄位 X 的欄位值時發生錯誤：對於輸入字符串：""」](#)。

## 隱藏 \_ 無法打開 \_ 拆分：打開蜂巢拆分時出錯 s3://DOC-EXAMPLE-BUCKET

當您查詢具有大量物件的 Simple Storage Service (Amazon S3) 儲存貯體字首時，可能會發生此錯誤。如需詳細資訊，請參閱[如何解決 Athena 的「HIVE\\_CANNOT\\_OPEN\\_SPLIT：開啟蜂巢分割時發生錯誤 s3://DOC-EXAMPLE-BUCKET/：放慢速度」錯誤？](#)在 AWS 知識中心。

## HIVE\_CURSOR\_ERROR：

com.amazonaws.services.s3.model.AmazonS3Exception：指定的索引鍵不存在

當查詢正在執行時移除檔案時，通常會發生這個錯誤。重新執行查詢，或檢查您的工作流程，以查看在查詢執行時是否有其他任務或程序正在修改檔案。

## HIVE\_CURSOR\_ERROR：輸入串流的意外結束

此訊息表示檔案已損毀或空白。檢查檔案的完整性，並重新執行查詢。

## HIVE\_FILESYSTEM\_ERROR：檔案的不正確的檔案大小 **1234567**

當檔案已在查詢規劃和查詢執行之間變更時，就會發生此訊息。這通常發生在就地取代 Simple Storage Service (Amazon S3) 上的檔案時 (例如 PUT 在物件已經存在的索引鍵上執行)。Athena 不支援在執行查詢時刪除或取代檔案的內容。若要避免此錯誤，請安排在查詢未執行時覆寫或刪除檔案的任務，或僅將資料寫入新的檔案或分割區。

## HIVE\_UNKNOWN\_ERROR：無法建立輸入格式

此錯誤可能是由以下問題所造成：

- AWS Glue 爬行者程式無法對資料格式進行分類
- 某些 AWS Glue 資料表定義性質為空
- Athena 不支援 Simple Storage Service (Amazon S3) 中檔案的資料格式

如需詳細資訊，請參閱 AWS 知識中心中的[如何在 Athena 中解決「無法建立輸入格式」錯誤？](#)或觀看知識中心的[影片](#)。

提供用於儲存查詢結果的 S3 位置無效。

請確保已為查詢結果指定有效的 S3 位置。如需詳細資訊，請參閱[使用查詢結果、近期查詢和輸出檔案](#)主題中的[指定查詢結果位置](#)。

## Linux Foundation Delta Lake 資料表

### Delta Lake 資料表結構描述不同步

當您查詢的 Delta Lake 資料表中的結構描述 AWS Glue 已過期時，您可能會收到下列錯誤訊息：

```
INVALID_GLUE_SCHEMA: Delta Lake table schema in Glue does not match the most recent schema of the Delta Lake transaction log. Please ensure that you have the correct schema defined in Glue.
```

如果架構在新增至 Athena AWS Glue 之後在中進行修改，則該架構可能會過期。若要更新結構描述，請執行下列其中一個步驟：

- 在中 AWS Glue，執行[AWS Glue 爬行者](#)程式。
- 在 Athena 中，[捨棄資料表](#)並再次[創建](#)它。
- 可以透過使用 Athena 中的 [ALTER TABLE ADD COLUMNS](#) 陳述式，或[編輯 AWS Glue 中的資料表結構描述](#)，手動新增遺失的資料欄。

## 聯合查詢

### 呼叫時逾時 ListTableMetadata

如果資料來源中有很多資料表、資料來源緩慢或網路速度慢，則呼叫 [ListTableMetadata](#) API 可能會逾時。若要對此問題進行疑難排解，請嘗試下列步驟。

- 檢查資料表數量 – 如果您有 1000 個以上的資料表，請嘗試減少資料表數量。為了最快獲得 ListTableMetadata 回應，我們建議每個目錄的資料表數量少於 1000。
- 檢查 Lambda 組態 – 監控 Lambda 函數行為非常重要。使用聯合目錄時，請務必檢查 Lambda 函數的執行日誌。根據結果，相應地調整記憶體和逾時值。若要找出任何潛在的逾時問題，請重新檢視

Lambda 組態。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的[設定 Lambda 函數逾時 \(主控台\)](#)。

- 檢查聯合資料來源日誌 – 檢查來自聯合資料來源的日誌和錯誤訊息，以查看是否有任何問題或錯誤。日誌可以提供有關逾時原因的寶貴見解。
- 使用 **StartQueryExecution** 擷取中繼資料 – 如果您有 1000 個以上的資料表，則使用聯合連接器擷取中繼資料可能需要比預期更長的時間。由於的非同步本質[StartQueryExecution](#)可確保 Athena 以最佳方式執行查詢，因此請考慮使用StartQueryExecution作為ListTableMetadata。下列 AWS CLI 範例顯示如StartQueryExecution何使用，而非取ListTableMetadata得資料目錄中資料表的所有中繼資料。

首先，執行取得所有資料表的查詢，如下列範例所示。

```
aws athena start-query-execution --region us-east-1 \  
--query-string "SELECT table_name FROM information_schema.tables LIMIT 50" \  
--work-group "your-work-group-name"
```

接著，擷取個別資料表的中繼資料，如下列範例所示。

```
aws athena start-query-execution --region us-east-1 \  
--query-string "SELECT * FROM information_schema.columns \  
WHERE table_name = 'your-table-name' AND \  
table_catalog = 'your-catalog-name'" \  
--work-group "your-work-group-name"
```

取得結果所花費的時間取決於目錄中的資料表數量。

[如需疑難排解聯合查詢的詳細資訊，請參閱的 awslabs/ -aws-athena-query-federation 節中的常見問題 GitHub](#)，或參閱個別 Athena 資料來源連接器的說明文件。

## JSON 相關錯誤

### 嘗試讀取 JSON 資料時，發生 NULL 或資料不正確的錯誤

當您嘗試讀取 JSON 資料時，NULL 或資料不正確的錯誤可能是由許多原因引發。若要識別在使用 OpenX 時造成錯誤的行 SerDe，`ignore.malformed.json`請將設定為true。格式錯誤的記錄將以 NULL 的形式傳回。如需詳細資訊，請參閱 AWS 知識中心中的[當我嘗試在 Amazon Athena 中讀取 JSON 資料時出現錯誤](#)或觀看知識中心的[影片](#)。

HIVE\_BAD\_DATA : 剖析欄位 0 的欄位值時發生錯誤 : java.lang.String 無法轉換為 org.openx.data.jsonserde.json.JSONObject

當 [OpenX SerDe](#) 無法剖析 Athena 查詢中的資料欄時，會擲回此錯誤。如果您將資料欄定義為 map 或 struct，但基礎資料實際上是 string、int 或其他基本類型，就會發生此情況。

HIVE\_CURSOR\_ERROR : 資料行不是有效的 JSON 物件 - JSONException : 重複索引鍵

當您使用 Athena 查詢在不同大小寫中具有相同名稱之多個標籤的 AWS Config 資源時，就會發生此錯誤。解決方案是使用 WITH SERDEPROPERTIES 'case.insensitive'='false' 執行 CREATE TABLE 並對應名稱。如需有關 case.insensitive 和映射的資訊，請參閱 [JSON SerDe 程式庫](#)。如需詳細資訊，請參閱從 Athena 讀取檔案時，[如何解決「HIVE\\_CURSOR\\_ERROR : 資料列不是有效的 JSON 物件-JSONException : 重複金鑰」](#)？AWS Config 在 AWS 知識中心。

帶有適合列印 JSON 的 HIVE\_CURSOR\_ERROR 訊息

[蜂巢 SerDe](#) 和 [OpenX SerDe](#) 程式庫預期每筆 JSON 文件都以單行文字表示，而且沒有行終止字元分隔記錄中的欄位。如果 JSON 文本是漂亮的打印格式，您可能會收到一條錯誤消息，如 HIVE\_CURSOR\_ERROR : 行不是有效的 JSON 對象或 HIVE\_CURSOR\_ERROR:: 意外 JsonParseException end-of-input : 當您嘗試在創建表後查詢對象的預期關閉標記。如需詳細資訊，請參閱上 GitHub 的 [OpenX SerDe 文件中的 JSON 資料](#) 檔案。

多個 JSON 記錄傳回 1 的 SELECT COUNT

如果您使用的是 [OpenX SerDe](#)，請確保記錄是以換行符號字元分隔。如需詳細資訊，請參閱 [Amazon Athena 中的 SELECT COUNT 查詢只會傳回一筆記錄，即使輸入 JSON 檔案在 AWS 知識中心中有多筆記錄也一樣](#)。

無法查詢使用自訂 JSON 分類器的 AWS Glue 爬行者程式所建立的資料表

Athena 引擎不支援 [自訂 JSON 分類器](#)。若要解決此問題，請在不使用自訂分類器的情況下建立新的資料表。若要轉換 JSON，您可以使用 CTAS 或建立檢視。例如，如果您正在使用陣列，則可以使用 UNNEST 選項來將 JSON 扁平化。另一種選擇是使用支援自訂分類器的 AWS Glue ETL 任務，將資料轉換為 Amazon S3 中的實木複合地板，然後在 Athena 中進行查詢。

MSCK REPAIR TABLE

如需有關 MSCK REPAIR TABLE 相關問題的資訊，請參閱 [MSCK REPAIR TABLE](#) 頁面的 [考量與限制](#) 和 [故障診斷](#) 章節。

## 輸出問題

### 無法驗證/建立輸出儲存貯體

如果指定的查詢結果位置不存在，或者如果沒有適當的許可，就會發生此錯誤。如需詳細資訊，請參閱[如何解決 Amazon Athena 中的「無法驗證/建立輸出值區」錯誤？](#)在 AWS 知識中心。

### TIMESTAMP 結果為空

Athena 需要 Java TIMESTAMP 格式。如需詳細資訊，請參閱 AWS 知識中心中的[當我在 Amazon Athena 中查詢資料表時，TIMESTAMP 結果為空。](#)

### 以 CSV 以外的格式存放 Athena 查詢輸出

依預設，Athena 僅會以 CSV 格式輸出檔案。若要以不同的格式輸出 SELECT 查詢結果，您可以使用 UNLOAD 陳述式。如需詳細資訊，請參閱[UNLOAD](#)。您也可以使用 CTAS 查詢，其使用 format [資料表屬性](#)以設定輸出格式。不像 UNLOAD，CTAS 技術需要建立資料表。如需詳細資訊，請參閱[如何以 CSV 以外的格式 \(例如壓縮格式\) 儲存 Athena 查詢輸出？](#)在 AWS 知識中心。

### 提供用於儲存查詢結果的 S3 位置無效

如果輸出儲存貯體位置與查詢執行所在的區域不同，則可能會收到此錯誤訊息。若要避免這種情況，請在您執行查詢的區域中指定查詢結果位置。如需這些步驟，請參閱[指定查詢結果位置](#)。

## Parquet 問題

### 組織. 阿帕奇 .io. GroupColumn無法將 IO 轉換為組織。 PrimitiveColumnIO

此錯誤是由 parquet 結構描述不相符造成。具有非基本類型的資料欄 (例如 array) 已在 AWS Glue 中被宣告為基本類型 (例如 string)。若要針對這個問題進行疑難排解，請檢查檔案中的資料結構描述，並將其與 AWS Glue 中宣告的結構描述相比較。

### Parquet 統計資料問題

當您讀取 Parquet 資料時，您可能會收到下列錯誤訊息：

```
HIVE_CANNOT_OPEN_SPLIT: Index x out of bounds for length y
HIVE_CURSOR_ERROR: Failed to read x bytes
HIVE_CURSOR_ERROR: FailureException at Malformed input: offset=x
HIVE_CURSOR_ERROR: FailureException at java.io.IOException:
can not read class org.apache.parquet.format.PageHeader: Socket is closed by peer.
```

若要解決此問題，請使用 [CREATE TABLE](#) 或 [ALTER TABLE SET TBLPROPERTIES](#) 陳述式將 Parquet SerDe `parquet.ignore.statistics` 屬性設定為 `true`，如下列範例所示。

#### CREATE TABLE 範例

```
...  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'  
WITH SERDEPROPERTIES ('parquet.ignore.statistics'='true')  
STORED AS PARQUET  
...
```

#### ALTER TABLE 範例

```
ALTER TABLE ... SET TBLPROPERTIES ('parquet.ignore.statistics'='true')
```

如需有關鑲木地板蜂巢的更多資訊 SerDe，請參閱 [木地板 SerDe](#)。

## 分割問題

### MSCK REPAIR TABLE 不會移除過時的分割區

如果您在 Simple Storage Service (Amazon S3) 中手動刪除分割區，然後執行 `MSCK REPAIR TABLE`，就可能收到錯誤訊息 `Partitions missing from filesystem` (檔案系統中遺失分割區)。這是因為 `MSCK REPAIR TABLE` 不會將過時的分割區從資料表中繼資料中移除。使用 [ALTER TABLE DROP PARTITION](#) 以手動移除過時的分割區。如需詳細資訊，請參閱 [MSCK REPAIR TABLE](#) 主題中的「疑難排解」一節。

### MSCK REPAIR TABLE 失敗

當大量的分割區 (例如，超過 100,000 個) 與特定資料表相關聯時，`MSCK REPAIR TABLE` 可能會因記憶體限制而失敗。若要解決此限制，請改用 [ALTER TABLE ADD PARTITION](#)。

### MSAK 修復表檢測到分區，但不會將它們添加到 AWS Glue

如果 Simple Storage Service (Amazon S3) 路徑使用駝峰式大小寫而不是小寫，或者如果 IAM 政策不允許 `glue:BatchCreatePartition` 動作，就會發生這個問題。如需詳細資訊，請參閱 [MSCL 修復資料表偵測到 Athena 中的分割區，但不會將它們新增至 AWS 知識 AWS Glue Data Catalog](#) 中心中的。

日期格式為 dd-MM-yyyy-HH-mm-ss 或 yyyy-MM-dd 的分割區投影範圍無法運作

若要正常運作，必須將日期格式設定為 yyyy-MM-dd HH:00:00。如需詳細資訊，請參閱《堆疊溢出》帖文：[Athena 分割區投影無法如預期般運作](#)。

## PARTITION BY 不支援 BIGINT 類型

將資料類型轉換為 string，然後重試。

## 沒有可用的有意義分割區

這個錯誤訊息通常表示分割區設定已損毀。若要解決這個問題，請捨棄該資料表，並建立具有新分割區的資料表。

## 分割區投影無法與範圍分割區搭配使用

確認時間範圍單位 `projection.<columnName>.interval.unit` 是否與分割區的分隔符號相符。例如，如果分割區以天分隔，則小時範圍單位將無法運作。

## 由連字號指定範圍時的分割區投影錯誤

使用連字號而不是逗號指定 range 資料表屬性會產生錯誤，例如 INVALID\_TABLE\_PROPERTY：針對輸入字串：`"number-number"`。確定範圍值以逗號分隔，而非連字號。如需詳細資訊，請參閱 [整數類型](#)。

## HIVE\_UNKNOWN\_ERROR：無法建立輸入格式

一或多個 glue 分割區以不同的格式宣告，因為每個 glue 分割區都有其自己的特定輸入格式。請檢查您的分割區是如何在中定義的 AWS Glue。

## HIVE\_PARTITION\_SCHEMA\_MISMATCH

如果分割區的結構描述與資料表的結構描述不同，查詢可能會失敗，並出現錯誤訊息

HIVE\_PARTITION\_SCHEMA\_MISMATCH。如需詳細資訊，請參閱 [同步分割區結構描述以避免 "HIVE\\_PARTITION\\_SCHEMA\\_MISMATCH"](#)。

## SemanticException 資料表未分割，但分割區規格存在

當未在 CREATE TABLE 陳述式中定義分割區時，就可能會發生此錯誤。如需詳細資訊，請參閱 [如何疑難排解 Athena 中的「FAILED：資料 SemanticException 表未分割但分割區規格存在」錯誤？](#) 在 AWS 知識中心。

## 零位元組 `_$folder$` 檔案

如果您執行 `ALTER TABLE ADD PARTITION` 語句並錯誤地指定已存在的分區和不正確的 Simple Storage Service (Amazon S3) 位置，則格式為 `partition_value_$folder$` 的零位元組預留位置檔案會在 Simple Storage Service (Amazon S3) 中建立。您必須手動移除這些檔案。

若要防止這種情況發生，請使用 `ALTER TABLE ADD PARTITION` 陳述式中的 `ADD IF NOT EXISTS` 語法，如下所示：

```
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION [...]
```

## 從分割資料傳回的零筆記錄

基於各種原因，可能會發生此問題。如需可能的原因和解決方案，請參閱[我在 Amazon Athena 中使用定義的分割區建立資料表，但是當我查詢資料表時，AWS 知識中心會傳回零筆記錄。](#)

另請參閱[HIVE\\_TOO\\_MANY\\_OPEN\\_PARTITIONS](#)。

## 許可

### 查詢 Amazon S3 時存取遭拒錯誤

如果您沒有讀取儲存貯體中資料的許可、寫入結果儲存貯體的許可，或是 Simple Storage Service (Amazon S3) 路徑包含 `us-east-1.amazonaws.com` 之類的區域端點，就可能發生此狀況。如需詳細資訊，請參閱 AWS 知識中心中的[當我執行 Athena 查詢時，收到「存取遭拒」錯誤。](#)

在 Amazon S3 中對加密資料執行 DDL 查詢時發生存取遭拒，狀態碼：403 錯誤

當您收到錯誤訊息存取遭拒 (服務：Amazon S3；狀態碼：403；錯誤碼: AccessDenied; 請求識別碼: <request\_id>)時，如果下列情況成立：

1. 您執行 `ALTER TABLE ADD PARTITION` 或 `MSCK REPAIR TABLE` 這類 DDL 查詢。
2. 你有一個儲存貯體，其具有設定為使用 SSE-S3 的[預設加密](#)。
3. 該儲存貯體也有如下所示的儲存貯體政策，其會強制 PutObject 要求來指定 PUT 標頭 `"s3:x-amz-server-side-encryption": "true"` 和 `"s3:x-amz-server-side-encryption": "AES256"`。

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

```
"Statement": [  
  {  
    "Effect": "Deny",  
    "Principal": "*",  
    "Action": "s3:PutObject",  
    "Resource": "arn:aws:s3:::<resource-name>/*",  
    "Condition": {  
      "Null": {  
        "s3:x-amz-server-side-encryption": "true"  
      }  
    }  
  },  
  {  
    "Effect": "Deny",  
    "Principal": "*",  
    "Action": "s3:PutObject",  
    "Resource": "arn:aws:s3:::<resource-name>/*",  
    "Condition": {  
      "StringNotEquals": {  
        "s3:x-amz-server-side-encryption": "AES256"  
      }  
    }  
  }  
]  
}
```

在這種情況下，建議的解決方案是移除像上面那樣的儲存貯體政策，因為儲存貯體的預設加密已經存在。

在另一個帳戶中查詢 Amazon S3 儲存貯體時發生存取遭拒，狀態碼：403

當您嘗試查詢另一個 AWS 服務帳戶寫入的記錄檔，而第二個帳戶是值區擁有者，但不擁有值區中的物件時，就會發生此錯誤。如需詳細資訊，請參閱當[我在知識中心查詢另一個帳戶中的儲存貯體或觀看 AWS 知識中心影片時，在 Amazon Athena 收到 Amazon S3 例外狀況「拒絕存取，且狀態碼為：403」](#)。

### 使用 IAM 角色憑證來連接至 Athena JDBC 驅動程式

您可以擷取角色的臨時憑證，以驗證 [JDBC 與 Athena 的連線](#)。臨時憑證的使用壽命上限為 12 小時。[如需詳細資訊，請參閱使用 JDBC 驅動程式連線至 Athena 時，如何使用 IAM 角色登入資料或切換到其他 IAM 角色？](#) 在 AWS 知識中心。

## 查詢語法問題

### 失敗：NullPointerException 名稱為空

如果您使用 AWS Glue [CreateTable](#) API 作業或 AWS CloudFormation [AWS::Glue::Table](#) 範本建立在 Athena 中使用的資料表，但未指定 `TableType` 屬性，然後執行類似 `SHOW CREATE TABLE` 或的 DDL 查詢 `MSCK REPAIR TABLE`，則可能會收到錯誤訊息 `F AILED: NullPointerException 名稱為空值`。

若要解決錯誤，請指定 [TableInput](#) `TableType` 屬性值做為 AWS Glue `CreateTable` API 呼叫或 [AWS CloudFormation 範本](#) 的一部分。`TableType` 可能的值包括 `EXTERNAL_TABLE` 或 `VIRTUAL_VIEW`。

只有當您使用 AWS Glue `CreateTable` API 作業或 [AWS::Glue::Table](#) 範本建立資料表時，才會套用此需求。如果您使用 DDL 陳述式或 AWS Glue 爬蟲程式建立 Athena 資料表，則系統會為您自動定義 `TableType` 屬性。

### 未註冊函數

當您嘗試使用 Athena 不支援的函數時，就會發生這個錯誤。如需 Athena 支援的函數清單，請參閱 [Amazon Athena 中的函數](#) 或在查詢編輯器中執行 `SHOW FUNCTIONS` 陳述式。您也可撰寫自己的 [使用者定義函數 \(UDF\)](#)。如需詳細資訊，請參閱 AWS 知識中心中的 [如何解決 Athena 中的「未註冊函數」語法錯誤？](#)。

### GENERIC\_INTERNAL\_ERROR 例外狀況

GENERIC\_INTERNAL\_ERROR 例外狀況可能有各式各樣的原因，其中包括下列原因：

- **GENERIC\_INTERNAL\_ERROR: Null** – 您可能會在下列任一種情況下看到此例外狀況：
  - 資料表定義中的資料欄的資料類型與資料集的實際資料類型之間存在結構描述不相符。
  - 您正在執行 `CREATE TABLE AS SELECT (CTAS)` 查詢的語法不正確。
- **GENERIC\_INTERNAL\_ERROR：父項產生器為空值** — 當您查詢具有資料類型資料行的資料表，而您正在使用 `OpenCSV` 程式庫時 `array`，可能會看到此例外狀況。`SerDe OpenCSVSerde` 格式不支援 `array` 資料類型。
- **GENERIC\_INTERNAL\_ERROR：值超過 MAX\_INT** – 當來源資料欄使用資料類型 `INT` 進行定義時，您可能看到此例外狀況，而且具有大於 2,147,483,647 的數值。
- **GENERIC\_INTERNAL\_ERROR：值超過 MAX\_BYTE** – 當來源資料欄的數值超過資料類型 `BYTE` 允許的大小時，您可能看到此例外狀況。資料類型 `BYTE` 相當於 `TINYINT`。`TINYINT` 是以兩的補數格式表示的 8 位元帶正負號的整數，最小值為 -128 以及最大值為 127。

- **GENERIC\_INTERNAL\_ERROR**：分割區的值的數目與篩選的數目不相符 – 如果 Amazon Simple Storage Service(Amazon S3) 資料上有不一致的分割區，您可能會看到此例外狀況。在下列任種情況下，您可能有不一致的分割區：
  - Simple Storage Service (Amazon S3) 上的分割區已變更 (例如：已新增新的分割區)。
  - 資料表中的分割區資料欄數目與分割區中繼資料中的資料欄數目不相符。

如需這些錯誤的詳細資訊，請參閱在 Amazon Athena [查詢資料表時，如何解決錯誤「GENERIC\\_INTERNAL\\_ERROR」](#)？在 AWS 知識中心。

## 相符群組的數目與資料欄數不符

當您在 CREATE TABLE 陳述式中使用 [正則表 SerDe](#)，且 regex 相符群組的數量與您為資料表指定的資料欄數不相符時，就會發生此錯誤。[如需詳細資訊，請參閱 Amazon Athena 如何解決「相符群組數目與欄數不符」RegexSerDe 錯誤？](#) 在 AWS 知識中心。

## queryString 無法滿足限制條件：成員的長度必須小於或等於 262144

Athena 的查詢字串長度上限 (262,144 位元組) 不是可調整的配額。AWS Support 無法為您增加配額，但您可以通過將長查詢拆分為較小的查詢來解決此問題。[如需詳細資訊，請參閱 AWS 知識中心中的如何在 Athena 中增加最大查詢字串長度？](#)

## SYNTAX\_ERROR：無法解析資料欄

當您從具有位元組順序標記 (BOM) 的 UTF-8 編碼 CSV 檔案中查詢 AWS Glue 爬行者程式建立的資料表時，可能會發生此錯誤。AWS Glue 無法識別 BOM 並將其更改為問號，這是 Amazon Athena 無法識別的問號。解決方案是移除 Athena 或 AWS Glue 中的問號。

## 函數呼叫的引數過多

在 Athena 引擎版本 3 中，函數接受的引數不能超過 127 個。此限制由設計所致。如果您使用具有 127 個以上參數的函數，則會出現如下錯誤訊息：

**TOO\_MANY\_ARGUMENTS**：行 *nnn:nn*：函數呼叫 *function\_name()* 的引數過多。

要解決此問題，請在每次函數呼叫中減少參數的使用。

## 查詢逾時問題

如果您的 Athena 查詢發生逾時錯誤，請檢查您的 CloudTrail 記錄。查詢可能會因為節流 AWS Glue 或 Lake Formation API 而逾時。發生這些錯誤時，對應的錯誤訊息可能會指出查詢逾時問題，而不是

限流問題。要解決此問題，您可以在聯繫之前檢查 CloudTrail 日誌 AWS Support。如需詳細資訊，請參閱 [查詢 AWS CloudTrail 記錄檔](#) 及 [記錄 Amazon Athena API 呼叫 AWS CloudTrail](#)。

如需呼叫 ListTableMetadata API 時聯合查詢出現查詢逾時問題的相關資訊，請參閱 [呼叫時逾時 ListTableMetadata](#)。

## 調節問題

如果您的查詢超出 Amazon S3、或 AWS Lambda 等相依服務的限制 AWS KMS AWS Glue，則可能會收到以下訊息。若要解決這些問題，請減少來自相同帳戶的並行呼叫數目。

服務	錯誤訊息
AWS Glue	AWSGlueException：超過比率。
AWS KMS	您已超過可以呼叫 KMS 的費率。減少通話的頻率。
AWS Lambda	已超過費率 TooManyRequestsException
Amazon S3	AmazonS3Exception：請降低請求率。

如需有關在使用 Athena 時防止 Amazon S3 限流的方法的資訊，請參閱 [防止 Amazon S3 限流](#)。

## 檢視

在 Apache Hive Shell 中建立的檢視在 Athena 中無法運作

由於其實作方式完全不同，在 Apache Hive Shell 中建立的檢視與 Athena 不相容。若要解決此問題，請在 Athena 中重新建立檢視。

**檢視已過時；必須重新建立**

如果作為檢視基礎的資料表已變更或遭到捨棄，您可能會收到此錯誤。解決方案是重新建立檢視。如需詳細資訊，請參閱 [如何解決 Athena 中的「檢視已過時；必須重新建立檢視」錯誤？](#) 在 AWS 知識中心。

## 工作群組

如需針對工作群組問題進行疑難排解的相關資訊，請參閱[工作群組疑難排解](#)。

## 其他資源

以下頁面提供針對 Amazon Athena 問題進行疑難排解的其他資訊。

- [Athena 錯誤目錄](#)
- [Service Quotas](#)
- [在 Amazon Athena 中進行 SQL 查詢的考量事項與限制](#)
- [不支援的 DDL](#)
- [資料表、資料庫和資料欄的名稱](#)
- [Amazon Athena 中的資料類型](#)
- [支援的 SerDes 和資料格式](#)
- [Athena 壓縮支援](#)
- [保留的關鍵字](#)
- [工作群組疑難排解](#)

以下 AWS 資源也可能有所幫助：

- [AWS 知識中心中的 Athena 主題](#)
- [關於 Re Amazon Athena 問題：AWS 郵政](#)
- [AWS 大數據部落格中的 Athena 貼文](#)

疑難排解通常需要專家或小幫手社群的反覆查詢和探索。如果您在嘗試此頁面上的建議後仍然遇到問題，請聯絡 AWS Support (在「Support 援」中 AWS Management Console，按一下「Support 中心」中)，或使用 Amazon Athena 標籤在 [AWS Re: POST](#) 上提出問題。

## Athena 錯誤目錄

Athena 提供標準化的錯誤資訊，可協助您瞭解失敗的查詢並在查詢失敗發生後採取步驟。AthenaError 功能包含 ErrorCategory 欄位和 ErrorType 欄位。ErrorCategory 說明查詢失敗的原因是系統錯誤、使用者錯誤還是其他錯誤所引起。ErrorType 提供有關失敗來源的更詳細的資訊。藉由結合這兩個欄位，您可以更清楚地瞭解周圍的情況和發生的具體錯誤的原因。

## 錯誤類別

下表列出 Athena 錯誤類別的值及其意義。

錯誤類別	來源
1	SYSTEM
2	USER
3	OTHER

## 錯誤類型參考

下表列出 Athena 錯誤類型的值及其意義。

錯誤類型	描述
0	在此擴展因數下查詢耗盡的資源
1	在此擴展因數下查詢耗盡的資源
2	在此擴展因數下查詢耗盡的資源
3	在此擴展因數下查詢耗盡的資源
4	在此擴展因數下查詢耗盡的資源
5	在此擴展因數下查詢耗盡的資源
6	在此擴展因數下查詢耗盡的資源
7	在此擴展因數下查詢耗盡的資源
8	在此擴展因數下查詢耗盡的資源
100	內部服務錯誤
200	查詢引擎發生內部錯誤

錯誤類型	描述
201	查詢引擎發生內部錯誤
202	查詢引擎發生內部錯誤
203	驅動程式錯誤
204	中繼存放區發生錯誤
205	查詢引擎發生內部錯誤
206	查詢逾時
207	查詢引擎發生內部錯誤
208	查詢引擎發生內部錯誤
209	無法取消查詢
210	查詢逾時
211	查詢引擎發生內部錯誤
212	查詢引擎發生內部錯誤
213	查詢引擎發生內部錯誤
214	查詢引擎發生內部錯誤
215	查詢引擎發生內部錯誤
216	查詢引擎發生內部錯誤
217	查詢引擎發生內部錯誤
218	查詢引擎發生內部錯誤
219	查詢引擎發生內部錯誤
220	查詢引擎發生內部錯誤

錯誤類型	描述
221	查詢引擎發生內部錯誤
222	查詢引擎發生內部錯誤
223	查詢引擎發生內部錯誤
224	查詢引擎發生內部錯誤
225	查詢引擎發生內部錯誤
226	查詢引擎發生內部錯誤
227	查詢引擎發生內部錯誤
228	查詢引擎發生內部錯誤
229	查詢引擎發生內部錯誤
230	查詢引擎發生內部錯誤
231	查詢引擎發生內部錯誤
232	查詢引擎發生內部錯誤
233	Iceberg 錯誤
234	Lake Formation 錯誤
235	查詢引擎發生內部錯誤
236	查詢引擎發生內部錯誤
237	序列化錯誤
238	無法將中繼資料上傳至 Simple Storage Service (Amazon S3)
239	一般持久性錯誤
240	無法提交查詢

錯誤類型	描述
300	內部服務錯誤
301	內部服務錯誤
302	內部服務錯誤
303	內部服務錯誤
400	內部服務錯誤
401	無法將查詢結果寫入 Simple Storage Service (Amazon S3)
402	無法將查詢結果寫入 Simple Storage Service (Amazon S3)
1000	使用者錯誤
1001	資料錯誤
1002	資料錯誤
1003	DDL 任務失敗
1004	架構錯誤
1005	序列化錯誤
1006	語法錯誤
1007	資料錯誤
1008	查詢被拒絕
1009	查詢失敗
1010	內部服務錯誤
1011	查詢被使用者取消
1012	查詢引擎發生內部錯誤

錯誤類型	描述
1013	查詢引擎發生內部錯誤
1014	查詢被使用者取消
1100	提供的自變數無效
1101	提供的屬性無效
1102	查詢引擎發生內部錯誤
1103	提供的資料表屬性無效
1104	查詢引擎發生內部錯誤
1105	查詢引擎發生內部錯誤
1106	提供的函數自變數無效
1107	無效的檢視
1108	無法註冊函數
1109	找不到提供的 Simple Storage Service (Amazon S3) 路徑
1110	提供的資料表或檢視不存在
1200	不支援的查詢
1201	不支持提供的解碼器
1202	不支援的查詢類型
1300	找不到一般錯誤
1301	找不到一般實體
1302	找不到檔案
1303	找不到提供的函數或函數實施

錯誤類型	描述
1304	查詢引擎發生內部錯誤
1305	查詢引擎發生內部錯誤
1306	找不到 Simple Storage Service (Amazon S3) 儲存貯體
1307	找不到選取的引擎
1308	查詢引擎發生內部錯誤
1400	調節錯誤
1401	由於 AWS Glue 節流，查詢失敗
1402	由於中的表格版本太多，查詢失敗 AWS Glue
1403	查詢失敗，原因是 Simple Storage Service (Amazon S3) 調節
1404	查詢失敗，原因是 Amazon Athena 調節
1405	查詢失敗，原因是 Amazon Athena 調節
1406	查詢失敗，原因是 Amazon Athena 調節
1500	許可錯誤
1501	Simple Storage Service (Amazon S3) 許可錯誤
1602	超過預留容量限制。沒有足夠的能力執行此查詢。
1700	由於 Lake Formation 內部異常，查詢失敗
1701	因為 AWS Glue 內部例外狀況，查詢失敗
9999	內部服務錯誤

# 程式碼範例

此主題中的範例使用 SDK for Java 2.x 做為起點來編寫 Athena 應用程式。

## Note

如需使用其他語言特定 AWS SDK 編寫 Athena 程式設計的相關資訊，請參閱下列資源：

- AWS Command Line Interface ([athena](#))
- AWS SDK for .NET ([Amazon.Athena.Model](#))
- AWS SDK for C++ ([Aws::Athena::AthenaClient](#))
- AWS SDK for Go ([athena](#))
- AWS SDK for JavaScript V3 ( [AthenaClient](#) )
- AWS SDK for PHP 3.x () [Aws\Athena](#)
- AWS SDK for Python (Boto3) ([Athena.Client](#))
- AWS SDK for Ruby V3 ( [Aws::Athena::Client](#) )

如需有關執行本節中 Java 程式碼範例的詳細資訊，請參閱上的[AWS 程式碼範例儲存庫](#)中的 [Amazon Athena Java 讀我檔案](#) GitHub。如需 Athena 的 Java 程式設計參考，請參閱[AthenaClient](#)中的 AWS SDK for Java 2.x。

- Java 程式碼範例
  - [常數](#)
  - [建立用戶端以存取 Athena](#)
  - 使用查詢執行
    - [開始查詢執行](#)
    - [停止查詢執行](#)
    - [列出查詢執行](#)
  - 使用具名查詢
    - [建立具名查詢](#)
    - [刪除具名查詢](#)
    - [列出查詢執行](#)

**Note**

這些範例會對在 `ExampleConstants.java` 類別宣告中定義的字串使用常數 (例如 `ATHENA_SAMPLE_QUERY`)。將這些常數取代為您自己的字串或定義的常數。

## 常數

`ExampleConstants.java` 類別示範如何在 Athena 中查詢由 [開始使用](#) 教學課程建立的資料表。

```
package aws.example.athena;

public class ExampleConstants {

    public static final int CLIENT_EXECUTION_TIMEOUT = 100000;
    public static final String ATHENA_OUTPUT_BUCKET = "s3://bucketscott2"; // change
the Amazon S3 bucket name to match                                     // your
environment
    // Demonstrates how to query a table with a comma-separated value (CSV) table.
    // For information, see
    // https://docs.aws.amazon.com/athena/latest/ug/work-with-data.html
    public static final String ATHENA_SAMPLE_QUERY = "SELECT * FROM scott2;"; // change
the Query statement to match                                         // your
environment
    public static final long SLEEP_AMOUNT_IN_MS = 1000;
    public static final String ATHENA_DEFAULT_DATABASE = "mydatabase"; // change the
database to match your database

}
```

## 建立用戶端以存取 Athena

`AthenaClientFactory.java` 類別說明如何建立和設定 Amazon Athena 用戶端。

```
package aws.example.athena;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.AthenaClientBuilder;
```

```
public class AthenaClientFactory {
    private final AthenaClientBuilder builder = AthenaClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create());

    public AthenaClient createClient() {
        return builder.build();
    }
}
```

## 開始查詢執行

`StartQueryExample` 說明如何將查詢提交至 Athena，請等待結果產生，然後處理結果。

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.QueryExecutionContext;
import software.amazon.awssdk.services.athena.model.ResultConfiguration;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionResponse;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionResponse;
import software.amazon.awssdk.services.athena.model.QueryExecutionState;
import software.amazon.awssdk.services.athena.model.GetQueryResultsRequest;
import software.amazon.awssdk.services.athena.model.GetQueryResultsResponse;
import software.amazon.awssdk.services.athena.model.ColumnInfo;
import software.amazon.awssdk.services.athena.model.Row;
import software.amazon.awssdk.services.athena.model.Datum;
import software.amazon.awssdk.services.athena.paginators.GetQueryResultsIterable;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
```

```
public class StartQueryExample {

    public static void main(String[] args) throws InterruptedException {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        String queryExecutionId = submitAthenaQuery(athenaClient);
        waitForQueryToComplete(athenaClient, queryExecutionId);
        processResultRows(athenaClient, queryExecutionId);
        athenaClient.close();
    }

    // Submits a sample query to Amazon Athena and returns the execution ID of the
    // query.
    public static String submitAthenaQuery(AthenaClient athenaClient) {
        try {
            // The QueryExecutionContext allows us to set the database.
            QueryExecutionContext queryExecutionContext =
                QueryExecutionContext.builder()
                    .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
                    .build();

            // The result configuration specifies where the results of the query should
            go.
            ResultConfiguration resultConfiguration = ResultConfiguration.builder()
                .outputLocation(ExampleConstants.ATHENA_OUTPUT_BUCKET)
                .build();

            StartQueryExecutionRequest startQueryExecutionRequest =
                StartQueryExecutionRequest.builder()
                    .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
                    .queryExecutionContext(queryExecutionContext)
                    .resultConfiguration(resultConfiguration)
                    .build();

            StartQueryExecutionResponse startQueryExecutionResponse = athenaClient
                .startQueryExecution(startQueryExecutionRequest);
            return startQueryExecutionResponse.queryExecutionId();

        } catch (AthenaException e) {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

```
        return "";
    }

    // Wait for an Amazon Athena query to complete, fail or to be cancelled.
    public static void waitForQueryToComplete(AthenaClient athenaClient, String
queryExecutionId)
        throws InterruptedException {
        GetQueryExecutionRequest getQueryExecutionRequest =
GetQueryExecutionRequest.builder()
            .queryExecutionId(queryExecutionId)
            .build();

        GetQueryExecutionResponse getQueryExecutionResponse;
        boolean isQueryStillRunning = true;
        while (isQueryStillRunning) {
            getQueryExecutionResponse =
athenaClient.getQueryExecution(getQueryExecutionRequest);
            String queryState =
getQueryExecutionResponse.queryExecution().status().state().toString();
            if (queryState.equals(QueryExecutionState.FAILED.toString())) {
                throw new RuntimeException(
                    "The Amazon Athena query failed to run with error message: " +
getQueryExecutionResponse
                        .queryExecution().status().stateChangeReason());
            } else if (queryState.equals(QueryExecutionState.CANCELLED.toString())) {
                throw new RuntimeException("The Amazon Athena query was cancelled.");
            } else if (queryState.equals(QueryExecutionState.SUCCEEDED.toString())) {
                isQueryStillRunning = false;
            } else {
                // Sleep an amount of time before retrying again.
                Thread.sleep(ExampleConstants.SLEEP_AMOUNT_IN_MS);
            }
            System.out.println("The current status is: " + queryState);
        }
    }

    // This code retrieves the results of a query
    public static void processResultRows(AthenaClient athenaClient, String
queryExecutionId) {
        try {
            // Max Results can be set but if its not set,
            // it will choose the maximum page size.
            GetQueryResultsRequest getQueryResultsRequest =
GetQueryResultsRequest.builder()
```

```
        .queryExecutionId(queryExecutionId)
        .build();

    GetQueryResultsIterable getQueryResultsResults = athenaClient
        .getQueryResultsPaginator(getQueryResultsRequest);
    for (GetQueryResultsResponse result : getQueryResultsResults) {
        List<ColumnInfo> columnInfoList =
result.resultSet().resultSetMetadata().columnInfo();
        List<Row> results = result.resultSet().rows();
        processRow(results, columnInfoList);
    }

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}

private static void processRow(List<Row> row, List<ColumnInfo> columnInfoList) {
    for (Row myRow : row) {
        List<Datum> allData = myRow.data();
        for (Datum data : allData) {
            System.out.println("The value of the column is " +
data.varCharValue());
        }
    }
}
}
```

## 停止查詢執行

`StopQueryExecutionExample` 會執行範例查詢、立即停止查詢，以及檢查查詢狀態，以確保已取消查詢。

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.StopQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionResponse;
import software.amazon.awssdk.services.athena.model.QueryExecutionState;
import software.amazon.awssdk.services.athena.model.AthenaException;
```

```
import software.amazon.awssdk.services.athena.model.QueryExecutionContext;
import software.amazon.awssdk.services.athena.model.ResultConfiguration;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class StopQueryExecutionExample {
    public static void main(String[] args) {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        String sampleQueryExecutionId = submitAthenaQuery(athenaClient);
        stopAthenaQuery(athenaClient, sampleQueryExecutionId);
        athenaClient.close();
    }

    public static void stopAthenaQuery(AthenaClient athenaClient, String
sampleQueryExecutionId) {
        try {
            StopQueryExecutionRequest stopQueryExecutionRequest =
StopQueryExecutionRequest.builder()
                .queryExecutionId(sampleQueryExecutionId)
                .build();

            athenaClient.stopQueryExecution(stopQueryExecutionRequest);
            GetQueryExecutionRequest getQueryExecutionRequest =
GetQueryExecutionRequest.builder()
                .queryExecutionId(sampleQueryExecutionId)
                .build();

            GetQueryExecutionResponse getQueryExecutionResponse = athenaClient
                .getQueryExecution(getQueryExecutionRequest);
            if (getQueryExecutionResponse.queryExecution()
                .status()
                .state()
                .equals(QueryExecutionState.CANCELLED)) {
```

```
        System.out.println("The Amazon Athena query has been cancelled!");
    }

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}

// Submits an example query and returns a query execution Id value
public static String submitAthenaQuery(AthenaClient athenaClient) {
    try {
        QueryExecutionContext queryExecutionContext =
QueryExecutionContext.builder()
            .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
            .build();

        ResultConfiguration resultConfiguration = ResultConfiguration.builder()
            .outputLocation(ExampleConstants.ATHENA_OUTPUT_BUCKET)
            .build();

        StartQueryExecutionRequest startQueryExecutionRequest =
StartQueryExecutionRequest.builder()
            .queryExecutionContext(queryExecutionContext)
            .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
            .resultConfiguration(resultConfiguration).build();

        StartQueryExecutionResponse startQueryExecutionResponse = athenaClient
            .startQueryExecution(startQueryExecutionRequest);
        return startQueryExecutionResponse.queryExecutionId();

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
    return null;
}
}
```

## 列出查詢執行

ListQueryExecutionsExample 說明如何取得查詢執行 ID 的清單。

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.ListQueryExecutionsRequest;
import software.amazon.awssdk.services.athena.model.ListQueryExecutionsResponse;
import software.amazon.awssdk.services.athena.paginators.ListQueryExecutionsIterable;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListQueryExecutionsExample {
    public static void main(String[] args) {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        listQueryIds(athenaClient);
        athenaClient.close();
    }

    public static void listQueryIds(AthenaClient athenaClient) {
        try {
            ListQueryExecutionsRequest listQueryExecutionsRequest =
ListQueryExecutionsRequest.builder().build();
            ListQueryExecutionsIterable listQueryExecutionResponses = athenaClient
                .listQueryExecutionsPaginator(listQueryExecutionsRequest);
            for (ListQueryExecutionsResponse listQueryExecutionResponse :
listQueryExecutionResponses) {
                List<String> queryExecutionIds =
listQueryExecutionResponse.queryExecutionIds();
                System.out.println("\n" + queryExecutionIds);
            }
        }
    }
}
```

```
        }

        } catch (AthenaException e) {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

## 建立具名查詢

CreateNamedQueryExample 說明如何建立具名查詢。

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.CreateNamedQueryRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class CreateNamedQueryExample {
    public static void main(String[] args) {
        final String USAGE = ""

            Usage:
                <name>

            Where:
                name - the name of the Amazon Athena query.\s
            """;

        if (args.length != 1) {
            System.out.println(USAGE);
            System.exit(1);
        }
    }
}
```

```
    }

    String name = args[0];
    AthenaClient athenaClient = AthenaClient.builder()
        .region(Region.US_WEST_2)
        .build();

    createNamedQuery(athenaClient, name);
    athenaClient.close();
}

public static void createNamedQuery(AthenaClient athenaClient, String name) {
    try {
        // Create the named query request.
        CreateNamedQueryRequest createNamedQueryRequest =
CreateNamedQueryRequest.builder()
        .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
        .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
        .description("Sample Description")
        .name(name)
        .build();

        athenaClient.createNamedQuery(createNamedQueryRequest);
        System.out.println("Done");

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}
}
```

## 刪除具名查詢

DeleteNamedQueryExample 說明如何使用具名查詢 ID 來刪除具名查詢。

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.DeleteNamedQueryRequest;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.CreateNamedQueryRequest;
```

```
import software.amazon.awssdk.services.athena.model.CreateNamedQueryResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteNamedQueryExample {
    public static void main(String[] args) {
        final String USAGE = ""

            Usage:
                <name>

            Where:
                name - the name of the Amazon Athena query.\s
            """;

        if (args.length != 1) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String name = args[0];
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        String sampleNamedQueryId = getNamedQueryId(athenaClient, name);
        deleteQueryName(athenaClient, sampleNamedQueryId);
        athenaClient.close();
    }

    public static void deleteQueryName(AthenaClient athenaClient, String
sampleNamedQueryId) {
        try {
            DeleteNamedQueryRequest deleteNamedQueryRequest =
DeleteNamedQueryRequest.builder()
                .namedQueryId(sampleNamedQueryId)
                .build();
        }
    }
}
```

```
        athenaClient.deleteNamedQuery(deleteNamedQueryRequest);

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}

public static String getNamedQueryId(AthenaClient athenaClient, String name) {
    try {
        CreateNamedQueryRequest createNamedQueryRequest =
CreateNamedQueryRequest.builder()
            .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
            .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
            .name(name)
            .description("Sample description")
            .build();

        CreateNamedQueryResponse createNamedQueryResponse =
athenaClient.createNamedQuery(createNamedQueryRequest);
        return createNamedQueryResponse.namedQueryId();

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
    return null;
}
}
```

## 列出具名查詢

ListNamedQueryExample 說明如何取得具名查詢 ID 的清單。

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.ListNamedQueriesRequest;
import software.amazon.awssdk.services.athena.model.ListNamedQueriesResponse;
import software.amazon.awssdk.services.athena.paginators.ListNamedQueriesIterable;
import java.util.List;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListNamedQueryExample {
    public static void main(String[] args) {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        listNamedQueries(athenaClient);
        athenaClient.close();
    }

    public static void listNamedQueries(AthenaClient athenaClient) {
        try {
            ListNamedQueriesRequest listNamedQueriesRequest =
ListNamedQueriesRequest.builder()
                .build();

            ListNamedQueriesIterable listNamedQueriesResponses = athenaClient
                .listNamedQueriesPaginator(listNamedQueriesRequest);
            for (ListNamedQueriesResponse listNamedQueriesResponse :
listNamedQueriesResponses) {
                List<String> namedQueryIds = listNamedQueriesResponse.namedQueryIds();
                System.out.println(namedQueryIds);
            }

        } catch (AthenaException e) {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

# 在 Amazon Athena 使用 Apache Spark

Amazon Athena 可讓您輕鬆使用 Apache Spark 以互動方式執行資料分析和探勘，而無需規劃、設定或管理資源。在 Athena 上執行 Apache Spark 應用程式表示提交 Spark 程式碼進行處理及直接接收結果，而不需要額外設定。您可以在 Amazon Athena 主控台中使用筆記本的簡化體驗，進而使用 Python 或 Athena 筆記本 API 開發 Apache Spark 應用程式。Amazon Athena 上的 Apache Spark 是無伺服器，並提供自動的隨需擴展，可提供即時啟動運算，以滿足不斷變化的資料量和處理需求。

Amazon Athena 提供如下功能：

- Console usage (主控台用量) – 從 Amazon Athena 主控台提交 Spark 應用程式。
- Scripting (指令碼編寫) – 在 Python 中快速且以互動方式建置和偵錯 Apache Spark 應用程式。
- Dynamic scaling (動態擴展) – Amazon Athena 會自動判斷執行任務所需的運算和記憶體資源，並相應地將這些資源持續擴展到您指定的最大值。這種動態擴展可降低成本，而不影響速度。
- Notebook experience (筆記本體驗) – 使用 Athena 筆記本編輯器，以運用熟悉的介面建立、編輯及執行運算。Athena 筆記本與 Jupyter 筆記本相容，並包含依照計算順序執行的儲存格清單。儲存格內容可以包括程式碼、文字、Markdown、數學、圖和豐富媒體。

如需詳細資訊，請參閱在 [Amazon Athena 星火上執行 Spark SQL](#)，以及在 AWS 大數據部落格中使用 Amazon Athena 進行 Apache Spark 探索您的資料湖。

## 考量與限制

- 目前，以下 AWS 區域提供 Amazon Athena for Apache Spark：
  - 亞太區域 (孟買)
  - 亞太區域 (新加坡)
  - 亞太區域 (雪梨)
  - 亞太區域 (東京)
  - 歐洲 (法蘭克福)
  - 歐洲 (愛爾蘭)
  - 美國東部 (維吉尼亞北部)
  - 美國東部 (俄亥俄)
  - 美國西部 (奧勒岡)
- AWS Lake Formation 不支援。

- 不支援使用分割區投影的資料表。
- 已啟用 Apache Spark 的工作群組可以使用 Athena 筆記本編輯器，但無法使用 Athena 查詢編輯器。只有 Athena SQL 工作群組可以使用 Athena 查詢編輯器。
- 不支援跨引擎檢視查詢。Athena for Spark 無法查詢 Athena SQL 建立的檢視。由於兩個引擎的檢視以不同的方式進行實作，因此他們與跨引擎使用不相容。
- 不支援 MLLib (阿帕奇星火機器學習程式庫) 和 `pyspark.ml` 套件。如需支援的 Python 程式庫清單，請參閱 [預先安裝的 Python 程式庫清單](#)。
- 目前在 Athena 中不支援 Spark 工作階段。 `pip install`
- 每個筆記本僅允許一個作用中的工作階段。
- 當多個使用者使用主控台開啟工作群組中的現有工作階段時，他們會存取同一筆記本。為避免混淆，請僅開啟您自己建立的工作階段。
- 您可能與 Amazon Athena 搭配使用的 Apache Spark 應用程式的託管域 (例如 `analytics-gateway.us-east-1.amazonaws.com`)，會在網際網路 [公用尾碼清單 \(PSL\)](#) 中註冊。如果您需要在域中設定敏感性 Cookie，我們建議您使用帶有 `__Host-` 字首的 Cookie，以協助保護您的域免受跨網站請求偽造 (CSRF) 攻擊的影響。如需更多資訊，請參閱 Mozilla.org 開發人員文件中的 [設定 Cookie](#) 頁面。
- 如需有關對 Athena 中的 Spark 筆記本、工作階段和工作群組進行疑難排解的詳細資訊，請參閱 [對 Athena for Spark 進行疑難排解](#)。

## 開始在 Amazon Athena 上使用 Apache Spark

若要開始在 Amazon Athena 上使用 Apache Spark，您必須先建立一個已啟用 Spark 的工作群組。切換到工作群組後，您可以建立筆記本或開啟現有筆記本。當您在 Athena 中開啟筆記本時，系統會自動啟動新的工作階段，並且您可以直接在 Athena 筆記本編輯器中進行使用。

### Note

在嘗試建立筆記本之前，請確定您建立了已啟用 Spark 的工作群組。

## 在 Athena 中建立已啟用 Spark 的工作群組

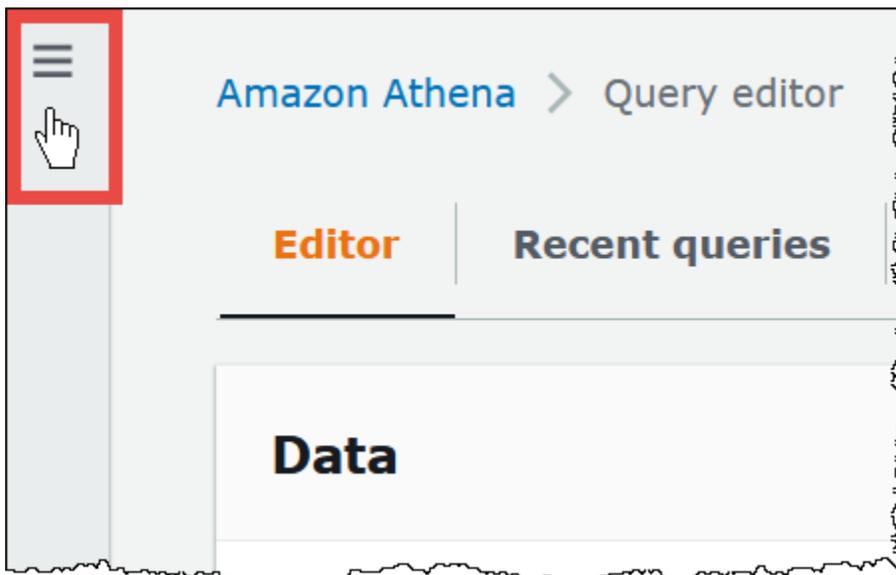
您可以使用 Athena 中的 [工作群組](#) 將使用者、團隊、應用程式或工作負載分組，並追蹤成本。若要在 Amazon Athena 中使用 Apache Spark，您可以建立使用 Spark 引擎的 Amazon Athena 工作群組。

**Note**

已啟用 Apache Spark 的工作群組可以使用 Athena 筆記本編輯器，但無法使用 Athena 查詢編輯器。只有 Athena SQL 工作群組可以使用 Athena 查詢編輯器。

若要在 Athena 中建立已啟用 Spark 的工作群組

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在導覽窗格中，選擇 Workgroups (工作群組)。
4. 在 Workgroups (工作群組) 頁面中，請選擇 Create workgroup (建立工作群組)。
5. 在 Workgroup name (工作群組名稱) 中，輸入 Apache Spark 工作群組的名稱。
6. (選用) 在 Description (說明) 中，輸入工作群組的說明。
7. 對於 Analytics engine (分析引擎)，選擇 Apache Spark。

**Note**

建立工作群組後，便無法變更工作群組的分析引擎類型。例如，Athena 引擎第 3 版工作群組無法變更為 PySpark 引擎第 3 版工作群組。

8. 為了符合本教學的目的，請選取 Turn on example notebook (開啟範例筆記本)。此選用功能會將名稱為名稱的範例筆記本新增 `example-notebook-random_string` 至您的工作群組，並新增筆

記本用來建立、顯示和刪除帳戶中特定資料庫和表格的 AWS Glue 相關權限，以及在 Amazon S3 中針對範例資料集的讀取許可。若要查看已新增許可，請選擇 [View additional permissions details](#) (檢視其他許可詳細資訊)。

#### Note

執行範例筆記本可能會產生一些額外費用。

9. 對於 Additional configurations (其他設定)，請執行下列動作：
  - 使用 Use defaults (使用預設值) 設定。此選項是預設值，可協助您開始使用已啟用 Spark 的工作群組。使用此選項，Athena 會在 Amazon S3 中為您建立 IAM 角色和計算結果位置。要建立的 IAM 角色名稱和 S3 儲存貯體位置會顯示在 Additional configurations (其他組態) 標題下方的方塊中。
  - 停用 Use defaults (使用預設值) 設定，然後繼續執行 [指定您自己的工作群組組態](#) 一節中的步驟，以手動設定您的工作群組。
10. (選用) Tags (標籤) – 使用此選項可將標籤新增至您的工作群組。如需詳細資訊，請參閱 [標記 Athena 資源](#)。
11. 選擇建立工作群組。隨即會出現一則訊息，通知您已成功建立工作群組，並且工作群組會顯示在工作群組清單中。

## 指定您自己的工作群組組態

如果您要為筆記本指定您自己的 IAM 角色和計算結果位置，則請遵循本節中的步驟。如果您為 Additional configurations (其他組態) 選項選擇 Use defaults (使用預設值)，則請跳過此章節並直接前往 [開啟筆記本瀏覽器和切換工作群組](#)。

下列程序假設您已完成上一節中 To create a Spark enabled workgroup in Athena (若要在 Athena 中建立已啟用 Spark 的工作群組) 的步驟 1 到 9。

### 若要指定您自己的工作群組組態

1. 如果您想要建立或使用您自己的 IAM 角色或設定筆記本加密，則請展開 IAM role configuration (IAM 角色組態)。
  - 對於 Service Role (服務角色)，選擇下列其中一項作業：
    - Create a service role (建立服務角色) – 選擇此選項可讓 Athena 為您建立服務角色。若要查看角色授予的許可，請選擇 View permission details (檢視許可詳細資訊)。

- Choose an existing service role (選擇現有服務角色) – 從下拉式選單中，選擇現有的角色。您選擇的角色必須包含第一個選項中的許可。如需有關已啟用筆記本之工作群組的許可的詳細資訊，請參閱 [對已啟用 Spark 的工作群組進行疑難排解](#)。
- 針對 Notebook and calculation code encryption key management (筆記本計算程式碼加密金鑰管理)，請選擇下列其中一種選項：
  - 由 Amazon Athena 擁有- AWS KMS 密鑰由 Amazon Athena 擁有和管理。使用此金鑰不會向您收取額外費用。
  - A symmetric key stored in your account, owned and managed by you (儲存在您的帳戶中，且由您擁有和管理的對稱金鑰。) – 對於此選項，請執行下列其中一項動作：
    - 若要使用現有的金鑰，請使用搜尋方塊選擇 AWS KMS 或輸入金鑰 ARN。
    - 若要在 AWS KMS 主控台中建立金鑰，請選擇 [建立 AWS KMS 金鑰]。您的執行角色必須具備使用所建立之金鑰的許可。

#### Important

當您變更工作群組的 [AWS KMS key](#) 時，更新前的受管筆記本仍會參考舊的 KMS 金鑰。更新後的受管筆記本使用新的 KMS 金鑰。若要更新舊筆記本以參考新的 KMS 金鑰，請將其匯出並匯入每個舊筆記本。如果您在將舊的筆記本參考更新至新的 KMS 金鑰之前刪除舊的 KMS 金鑰，則舊的筆記本將無法再解密且無法復原。

此行為也適用於[別名](#)的更新，別名是 KMS 金鑰的易用名稱。當您更新 KMS 金鑰別名以指向新的 KMS 金鑰時，別名更新前的受管筆記本仍會參考舊的 KMS 金鑰，而別名更新後的受管筆記本則會使用新的 KMS 金鑰。更新 KMS 金鑰或別名之前，請考慮這幾點。

2. 如果您要指定您自己的計算結果設定，則請展開 Calculation result settings (計算結果設定)，然後從下列選項中選擇。
  - Create a new S3 bucket (建立新的 S3 儲存貯體) – 此選項會在您的帳戶中為您的計算結果建立 Amazon S3 儲存貯體。儲存貯體名稱具有格式 `account_id-region-athena-results-bucket-alphanumeric_id`，並使用 ACL 已停用、公有存取已封鎖、版本控制已停用以及儲存貯體擁有者強制執行的設定。
  - Choose an existing S3 location (選擇現有的 S3 位置) – 對於此選項，請執行下列動作：
    - 在搜尋方塊中輸入現有位置的 S3 路徑，或選擇 Browse S3 (瀏覽 S3) 從清單中選擇儲存貯體。

**Note**

當您在 Amazon S3 中選取現有位置時，請不要在該位置附加正斜線 (/)。這麼做會導致 [計算詳細資訊頁面](#) 上的計算結果位置的連結指向不正確的目錄。如果發生此問題，則請編輯工作群組的結果位置，以移除尾隨正斜線。

- (選用) 選擇 View (檢視)，以開啟 Amazon S3 主控台的 Buckets (儲存貯體) 頁面，您可以在其中檢視您選擇之現有儲存貯體的詳細資訊。
  - (選擇性) 針對「預期值區擁有者」，輸入您希望成為查詢結果輸出位置值區擁有者的 AWS 帳戶 ID。建議您盡可能選擇此選項作為新增的安全措施。如果儲存貯體擁有者的帳戶 ID 與您指定的 ID 不相符，則嘗試輸出到儲存貯體的動作會失敗。如需詳細資訊，請參閱 Amazon S3 使用者指南中的 [使用儲存貯體擁有者條件驗證儲存貯體擁有權](#)。
  - (選用) 如果您的計算結果位置係由另一個帳戶所擁有且您想要授予其他帳戶對查詢結果的完整控制，則選取 Assign bucket owner full control over query results (指派儲存貯體擁有者對查詢結果的完整控制)。
3. (選用) 選取 Encrypt calculation results (加密計算結果)，然後選擇以下其中一項：
- SSE\_S3 – 這是一個 S3 受管伺服器端加密金鑰。
  - SSE\_KMS – 您提供的金鑰。對於「選擇 AWS KMS 金鑰」，您可以選擇下列其中一項：
    - 使用 AWS 擁有的金鑰 — 使用為您 AWS 擁有和管理的金鑰。
    - 選擇不同的 AWS KMS 金鑰 (進階) — 選擇或建立金鑰。
      - 若要使用現有的金鑰，請使用搜尋方塊選擇 AWS KMS 或輸入金鑰 ARN。
      - 若要在 KMS 主控台中建立金鑰，請選擇 [建立金 AWS KMS 鑰]。在 KMS 主控台中完成金鑰建立後，請返回 Athena 主控台內的 [建立工作群組] 頁面，然後使用 [選擇金 AWS KMS 鑰] 或輸入 ARN 搜尋方塊，選擇剛才建立的金鑰。
4. (選擇性) 其他設定 — 展開此選項以啟用或停用工作群組的「發佈 CloudWatch 量度」選項。預設會選取此欄位。如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Apache Spark 計算](#)。
5. (選用) Tags (標籤) – 使用此選項可將標籤新增至您的工作群組。如需詳細資訊，請參閱 [標記 Athena 資源](#)。
6. 選擇建立工作群組。隨即會出現一則訊息，通知您已成功建立工作群組，並且工作群組會顯示在工作群組清單中。

## 開啟筆記本瀏覽器和切換工作群組

您必須切換至工作群組，才能使用您剛建立的已啟用 Spark 的工作群組。若要切換已啟用 Spark 的工作群組，您可以使用筆記本瀏覽器或筆記本編輯器中的 Workgroup (工作群組) 選項。

### Note

開始之前，檢查您的瀏覽器是否未封鎖第三方 Cookie。任何因預設或使用者啟用設定而封鎖第三方 Cookie 的瀏覽器，將使筆記本無法啟動。如需管理 Cookie 的詳細資訊，請參閱：

- [Chrome](#)
- [Firefox](#)
- [Safari](#)

### 若要開啟筆記本瀏覽器和切換工作群組

1. 在導覽窗格中，選擇 Notebook explorer (筆記本瀏覽器)。
2. 使用主控台右上角的 Workgroup (工作群組) 選項，以選擇您建立之已啟用 Spark 的工作群組。範例筆記本會顯示在筆記本清單中。

您可用下列方式使用筆記本瀏覽器：

- 選擇筆記本的連結名稱，以在新的工作階段中開啟筆記本。
- 若要重新命名、刪除或匯出筆記本，請使用 Actions (動作) 選單。
- 若要匯入筆記本檔案，請選擇 Import file (匯入檔案)。
- 若要建立筆記本，選擇 Create notebook (建立筆記本)。

## 執行範例筆記本

範例筆記本會查詢公開提供之紐約計程車行程資料集中的資料。筆記本有示例顯示如何使用星火 DataFrames，星火 SQL 和 AWS Glue Data Catalog。

### 若要執行範例筆記本

1. 在筆記本瀏覽器中，選擇範例筆記本的連結名稱。

這樣一來，會使用預設參數啟動筆記本工作階段，並在筆記本編輯器中開啟筆記本。隨即會出現一則訊息，通知您已使用預設參數 (最多 20 個 DPU) 啟動新的 Apache Spark 工作階段。

- 若要依序執行儲存格並觀察結果，請針對筆記本中的每個儲存格選擇 Run (執行) 按鈕一次。
  - 向下滾動，以查看結果並將新儲存格帶入視圖中。
  - 對於包含計算的儲存格，進度列會顯示已完成的百分比、經過時間和剩餘時間。
  - 範例筆記本會在您的帳戶中建立範例資料庫和資料表。最後一個儲存格會移除這些作為清理步驟。

### Note

如果您變更範例筆記本中的資料夾、資料表或資料庫名稱，則請確定這些變更已在您使用的 IAM 角色中顯現。否則，筆記本可能由於許可不足而無法執行。

## 編輯工作階段詳細資訊

啟動筆記本工作階段之後，您可以編輯工作階段詳細資訊，例如資料表格式、加密、工作階段閒置逾時，以及您要使用之資料處理單位 (DPU) 的最大並行數目。DPU 是相對的處理能力，包含 4 個 vCPU 的運算容量和 16 GB 的記憶體。

### 若要編輯工作階段詳細資訊

- 在筆記本編輯器中，從右上角的 Session (工作階段) 選單中，選擇 Edit session (編輯工作階段)。
- 在編輯工作階段詳細資訊對話方塊的 Spark properties 區段中，為下列選項選擇或輸入值：
  - 其他資料表格式 — 選擇 Linux Foundation Delta Lake、Apache Hudi、Apache Iceberg 或自訂。
    - 針對 Delta、Hudi 或 Iceberg 資料表選項，會在在資料表中編輯和在 JSON 中編輯選項中，自動為您提供相應資料表的必要資料表屬性。如需有關使用這些資料表格式的詳細資訊，請參閱 [使用 Amazon Athena for Apache Spark 中的非 Hive 資料表格式](#)。
    - 若要新增或移除自訂或其他資料表類型的資料表屬性，請使用在資料表中編輯和在 JSON 中編輯 選項。
    - 針對在資料表中編輯選項，請選擇新增屬性以新增屬性，或選擇移除以移除屬性。若要輸入屬性名稱及其價值，請使用金鑰和值方塊。

- 針對在 JSON 中編輯選項，請使用 JSON 文字編輯器直接編輯組態。
  - 若要將 JSON 文字複製到剪貼簿，請選擇複製。
  - 若要從 JSON 編輯器移除所有文字，請選擇清除。
  - 若要設定換行或選擇 JSON 編輯器的顏色主題，選擇設定 (齒輪) 圖示。
3. 開啟 Spark 加密 -- 選取此選項可加密寫入磁碟並透過 Spark 網路節點傳送的資料。如需詳細資訊，請參閱 [啟用 Apache Spark 加密](#)。
3. 在工作階段參數區段中，為下列選項選擇或輸入值：
- Session idle timeout (工作階段閒置逾時) - 選擇或輸入介於 1 到 480 分鐘之間的值。預設值為 20。
  - Coordinator size (協調器大小) - 協調器是一種特殊執行器，可用於在筆記本工作階段中協調處理工作並管理其他執行器。目前，1 個 DPU 是預設值，也是唯一可能的值。
  - Executor size (執行器大小) - 執行器是筆記本工作階段可向 Athena 請求的最小運算單位。目前，1 個 DPU 是預設值，也是唯一可能的值。
  - Max concurrent value (最大並行值) - 可同時執行的 DPU 數量上限。預設值為 20，最小值為 3，最大值為 60。增加此值並不會自動配置其他資源，但是當運算負載需要且資源可用時，Athena 將會嘗試配置指定的最大數量。
4. 選擇儲存。
5. 出現 Confirm edit (確認編輯) 提示時，選擇 Confirm (確認)。

Athena 會儲存您的筆記本，並使用您指定之參數啟動新的工作階段。筆記本編輯器中的橫幅會通知您已使用已修改之參數啟動新的工作階段。

#### Note

Athena 會記住您筆記本的工作階段設定。如果您編輯工作階段的參數，然後終止工作階段，則 Athena 會使用您為下次啟動筆記本工作階段所設定的工作階段參數。

## 檢視工作階段和計算詳細資訊

執行筆記本後，您可以檢視工作階段和計算詳細資訊。

若要檢視工作階段和計算詳細資訊

1. 從右上角的 Session (工作階段) 選單中，選擇 View details (檢視詳細資訊)。

- Current session (目前工作階段) 會顯示目前工作階段的相關資訊，包括工作階段 ID、建立時間、狀態和工作群組。
  - History (歷史記錄) 標籤會列出先前工作階段的工作階段 ID。若要檢視先前工作階段的詳細資訊，請選擇 History (歷史記錄) 標籤，然後在清單中選擇工作階段 ID。
  - Calculations (計算) 區段會顯示工作階段中執行的計算清單。
2. 若要檢視計算的詳細資訊，請選擇計算 ID。
  3. 在 Calculation details (計算詳細資訊) 頁面上，您可以執行下列操作：
    - 若要檢視計算的程式碼，請參閱 Code (程式碼) 區段。
    - 若要查看計算的結果，請選擇 Results (結果) 標籤。
    - 若要以文字格式下載您看到的結果，請選擇 Download results (下載結果)。
    - 若要在 Amazon S3 中檢視有關計算結果的資訊，請選擇 View in S3 (在 S3 中檢視)。

## 終止工作階段

若要結束筆記本工作階段

1. 在筆記本編輯器中，從右上角的 Session (工作階段) 選單中，選擇 Terminate (終止)。
2. 出現 Confirm session termination (確認工作階段終止) 提示時，選擇 Confirm (確認)。隨即便會儲存您的筆記本，並返回筆記本編輯器。

### Note

在筆記本編輯器中關閉筆記本標籤本身並不會終止作用中筆記本的工作階段。如果您要確認工作階段已終止，則請使用 Session (工作階段)、Terminate (終止) 選項。

## 建立您自己的筆記本

在您建立已啟用 Spark 的 Athena 工作群組後，您可建立自己的筆記本。

若要建立筆記本

1. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。

2. 在 Athena 主控台導覽窗格中，選擇 Notebook explorer (筆記本瀏覽器) 或 Notebook editor (筆記本編輯器)。
3. 執行以下任意一項：
  - 在 Notebook explorer (筆記本瀏覽器) 中，選擇 Create notebook (建立筆記本)。
  - 在 Notebook editor (筆記本瀏覽器) 中，選擇 Create notebook (建立筆記本)，或選擇加號圖示 (+) 以新增筆記本。
4. 在 Create notebook (建立筆記本) 對話方塊中，針對 Notebook name (筆記本名稱)，輸入名稱。
5. (選用) 展開 Spark 屬性，然後為下列選項選擇或輸入值：
  - 其他資料表格式 — 選擇 Linux Foundation Delta Lake、Apache Hudi、Apache Iceberg 或自訂。
  - 針對 Delta、Hudi 或 Iceberg 資料表選項，會在在資料表中編輯和在 JSON 中編輯選項中，自動為您提供相應資料表的必要資料表屬性。如需有關使用這些資料表格式的詳細資訊，請參閱 [使用 Amazon Athena for Apache Spark 中的非 Hive 資料表格式](#)。
  - 若要新增或移除自訂或其他資料表類型的資料表屬性，請使用在資料表中編輯和在 JSON 中編輯 選項。
  - 針對在資料表中編輯選項，請選擇新增屬性以新增屬性，或選擇移除以移除屬性。若要輸入屬性名稱及其價值，請使用金鑰和值方塊。
  - 針對在 JSON 中編輯選項，請使用 JSON 文字編輯器直接編輯組態。
    - 若要將 JSON 文字複製到剪貼簿，請選擇複製。
    - 若要從 JSON 編輯器移除所有文字，請選擇清除。
    - 若要設定換行或選擇 JSON 編輯器的顏色主題，選擇設定 (齒輪) 圖示。
  - 開啟 Spark 加密 -- 選取此選項可加密寫入磁碟並透過 Spark 網路節點傳送的資料。如需詳細資訊，請參閱 [啟用 Apache Spark 加密](#)。
6. (選用) 展開 Session parameters (工作階段參數)，然後為下列選項選擇或輸入值：
  - Session idle timeout (工作階段閒置逾時) - 選擇或輸入介於 1 到 480 分鐘之間的值。預設值為 20。
  - Coordinator size (協調器大小) - 協調器是一種特殊執行器，可用於在筆記本工作階段中協調處理工作並管理其他執行器。目前，1 個 DPU 是預設值，也是唯一可能的值。DPU (資料處理單位) 是相對的處理能力，包含 4 個 vCPU 的運算容量和 16 GB 的記憶體。
  - Executor size (執行器大小) - 執行器是筆記本工作階段可向 Athena 請求的最小運算單位。目前，1 個 DPU 是預設值，也是唯一可能的值。

- Max concurrent value (最大並行值) - 可同時執行的 DPU 數量上限。預設值為 20，最大值為 60。增加此值並不會自動配置其他資源，但是當運算負載需要且資源可用時，Athena 將會嘗試配置指定的最大數量。

7. 選擇建立。隨即會在筆記本編輯器之新工作階段中開啟您的筆記本。

## 開啟先前建立的筆記本

若要開啟先前建立的筆記本

1. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
2. 在 Athena 主控台導覽窗格中，選擇 Notebook editor (筆記本編輯器) 或 Notebook explorer (筆記本瀏覽器)。
3. 執行以下任意一項：
  - 在 Notebook editor (筆記本編輯器) 中，選擇 Recent notebooks (最近的筆記本) 或 Saved notebooks (儲存的筆記本) 清單中的筆記本。隨即會在新的工作階段中開啟筆記本。
  - 在 Notebook explorer (筆記本瀏覽器) 中，選擇清單中的筆記本的名稱。隨即會在新的工作階段中開啟筆記本。

如需有關管理筆記本檔案的詳細資訊，請參閱 [管理筆記本檔案](#)。

## 使用筆記本

您可以在 Athena 筆記本瀏覽器中管理筆記本，並使用 Athena 筆記本編輯器在工作階段中進行編輯和執行。您可以根據您的需求，設定筆記本工作階段的 DPU 使用情況。

當您停止筆記本時，您會終止相關的工作階段。所有檔案均已儲存，但宣告的變數、函數和類別中正在進行的變更會遺失。當您重新啟動筆記本時，Athena 會重新載入筆記本檔案，並且您可以再次執行程式碼。

## 工作階段和計算

每個筆記本都與單一 Python 核心相關聯，並會執行 Python 程式碼。筆記本可具有一或多個包含命令的儲存格。若要執行筆記本中的儲存格，您必須先為筆記本建立工作階段。工作階段會追蹤變數和筆記本的狀態。

在筆記本中執行儲存格表示在目前工作階段中執行計算。計算會追蹤筆記本的進度狀態，並且可能執行從 Amazon S3 讀取或寫入其他資料存放區等任務。只要工作階段在執行中，計算就會使用並修改為筆記本維護的狀態。

當您不再需要狀態時，可以結束工作階段。當您結束工作階段時，筆記本會保留，但變數和其他狀態資訊會被銷毀。如果您需要同時處理多個專案，則可以為每個專案建立一個工作階段，並且工作階段將彼此獨立。

工作階段具有專屬運算容量，以 DPU 為單位。建立工作階段時，您可以指派工作階段數個 DPU。根據任務的需求，不同的工作階段可以具有不同的容量。

## 使用 Athena 筆記本編輯器

Athena 筆記本編輯器是一種用於編寫和執行程式碼的互動式環境。以下章節描述環境的功能。

### 命令模式與編輯模式

筆記本編輯器具有一個模式使用者介面：用於將文字輸入到儲存格的編輯模式，以及用於向編輯器本身發出命令的命令模式，例如複製、粘貼或執行。

若要使用編輯模式和命令模式，您可以執行下列任務：

- 若要進入編輯模式，請按下 **ENTER**，或選擇儲存格。當儲存格處於編輯模式時，儲存格會有一個綠色的左邊界。
- 若要進入命令模式，請按下 **ESC**，或按一下儲格外側。請注意，命令通常僅適用於當前選定的儲存格，而不適用於所有儲存格。當編輯器處於命令模式時，儲存格會有一個藍色的左邊界。
- 在命令模式中，您可以使用鍵盤快速鍵和編輯器上方的選單，但不能在個別儲存格中輸入文字。
- 若要選取儲存格，請選擇儲存格。
- 若要選取所有儲存格，請按下 **Ctrl+A** (Windows) 或 **Cmd+A** (Mac)

### 筆記本編輯器選單

筆記本編輯器頂端選單中的圖示提供以下選項：

- Save (儲存) – 儲存筆記本的目前狀態。
- Insert cell below (在下方插入儲存格) – 在目前選取的儲存格下方新增新的 (空的) 儲存格。
- Cut selected cells (剪下選取的儲存格) – 從其目前位置移除選取的儲存格，並將儲存格複製到記憶體。

- Copy selected cells (複製選取的儲存格) – 將選取的儲存格複製到記憶體。
- Paste cells below (在下方貼上儲存格) – 在目前儲存格下方貼上複製的儲存格。
- Move selected cells up (上移選取的儲存格) – 將目前儲存格移至上方儲存格的上方。
- Move selected cells up (下移選取的儲存格) – 將目前儲存格移至下方儲存格的下方。
- Run (執行) – 執行目前 (選取的) 儲存格。輸出會立即顯示在目前儲存格的下方。
- Run all (執行所有) – 執行筆記本中的所有儲存格。每個儲存格的輸出會立即顯示在儲存格的下方。
- Stop (Interrupt the kernel) (停止 (中斷核心)) – 透過中斷核心來停止目前的筆記本。
- Format option (格式選項) – 選取儲存格格式，且格式可以是下列其中一種：
  - Code (程式碼) – 用於 Python 程式碼 (預設值)。
  - 降價 — 用於以樣式 [降價GitHub格式](#) 輸入文本。若要轉譯 Markdown，請執行儲存格。
  - Raw NBConvert – 用於以未經修改的形式輸入文字。標記為 Raw NBConvert 的儲存格可透過 Jupyter [nbconvert](#) 命令列工具轉換成不同的格式，例如 HTML。
- Heading (標題) – 用於變更儲存格的標題層級。
- Command palette (命令面板) – 包含 Jupyter 筆記本命令及其鍵盤快速鍵。如需有關鍵盤快速鍵的詳細資訊，請參閱本文件稍後的章節。
- Session (工作階段) – 使用此選單中的選項來 [檢視](#) 工作階段的詳細資訊、[編輯工作階段參數](#) 或 [終止](#) 工作階段。

## 命令模式鍵盤快速鍵

以下是筆記本編輯器命令模式鍵盤快速鍵。按下 **ESC** 進入命令模式後，即可使用這些快速鍵。若要檢視編輯器中可用命令的完整清單，請按下 **ESC + H**。

金鑰	動作
<b>1 - 6</b>	將儲存格類型變更為 Markdown 並將標題層級設定為鍵入的數字
<b>a</b>	在目前儲存格上方建立儲存格
<b>b</b>	在目前儲存格下方建立儲存格
<b>c</b>	將目前儲存格複製到記憶體
<b>d d</b>	刪除目前儲存格

金鑰	動作
<b>h</b>	顯示鍵盤快速鍵協助畫面
<b>j</b>	向下移一個儲存格
<b>k</b>	向上移一個儲存格
<b>m</b>	將目前儲存格格式變更為 Markdown
<b>r</b>	將目前儲存格格式變更為 Raw
<b>s</b>	儲存筆記本
<b>v</b>	在目前儲存格下方貼上記憶體內容
<b>x</b>	剪下選取的一或多個儲存格
<b>y</b>	將儲存格格式變更為程式碼
<b>z</b>	Undo (復原)
<b>Ctrl+Enter</b>	執行目前儲存格並輸入命令模式
<b>Shift+Enter</b> 或 <b>Alt+Enter</b>	執行目前儲存格並在輸出下方建立新的儲存格，然後在編輯模式下輸入新的儲存格
<b>Space</b>	向下翻頁
<b>Shift+Space</b>	向上翻頁
<b>Shift + L</b>	切換儲存格中行號的可見度

## 編輯命令模式快速鍵

筆記本編輯器提供自訂命令模式鍵盤快速鍵的選項。

## 若要編輯命令模式快速鍵

1. 從筆記本編輯器選單中，選擇 Command palette (命令面板)。
2. 從命令面板中，選擇 Edit command mode keyboard shortcuts (編輯命令模式鍵盤快速鍵) 命令。
3. 使用 Edit command mode shortcuts (編輯命令模式快速鍵) 介面可將您想要的命令映射或重新映射至鍵盤。

若要查看編輯命令模式快速鍵的指示，請捲動到 Edit command mode shortcuts (編輯命令模式快速鍵) 畫面的底部。

如需有關在 Athena for Apache Spark 中使用魔術命令的資訊，請參閱 [使用魔術命令](#)。

## 使用魔術命令

魔術命令或魔術是可以在筆記本儲存格中執行的特殊命令。例如，`%env` 會在筆記本工作階段中顯示環境變數。Athena 支援 IPython 6.0.3 中的魔術函數。

本節介紹了 Athena for Apache Spark 的一些關鍵魔術命令。

- 若要在 Athena 中查看這些魔術命令的清單，請在筆記本儲存格中執行命令 `%lsmagic`。
- 如需有關在 Athena 筆記本中使用魔法建立圖形的資訊，請參閱 [用於建立資料圖形的魔術](#)。
- 如需有關其他魔術命令的資訊，則請參閱 IPython 文件中的 [內建魔術命令](#)。

### Note

目前，`%pip` 命令在執行時失敗。這是已知問題。

## 儲存格魔術

編寫在多行上的魔術前面有一個雙百分比符號 (`%%`)，被稱為儲存格魔術函數或儲存格魔術。

```
%%sql
```

此儲存格魔術允許直接執行 SQL 陳述式，而不必用 Spark SQL 語句來進行裝飾。該命令還可隱式呼叫傳回的 Dataframe 上的 `.show()`，進而顯示輸出。

```
In [1]: %%sql
        SELECT 1

Calculation started (calculation_id=dac32df7-e76b-251d-491a-603d755
77bde) in (session=a6c32df6-dc5f-3390-be39-38bd204513be). Checking
calculation status...

Progress: ██████████ elapsed time = 00:06s, DPU counts
100%                active/requested = 0/0

Calculation completed.
+----+
|  1 |
+----+
|  1 |
+----+
```

`%%sql` 命令會自動將資料欄輸出截斷為 20 個字元的寬度。這目前無法設定。若要解決此限制，請使用下列完整語法，並相應地修改 `show` 方法的參數。

```
spark.sql("""YOUR_SQL""").show(n=number, truncate=number, vertical=bool)
```

- `nint`，選用。要顯示的資料列數目。
- 截斷 – `bool` 或 `int`，可選-如為 `true`，則會截斷超過 20 個資源的字串。當設定為大於 1 的數字時，會將長字串截斷為指定長度，並靠右對齊儲存格。
- 垂直 – `bool`，選用。如為 `true`，則會垂直列印輸出資料列 (每個資料欄值一行)。

## 行魔術

單行上的魔術前面有一個百分比符號 (`%`)，被稱為行魔術函數或行魔術。

`%help`

顯示可用魔術命令的描述。

```
In [6]: %help
```

```
Available Magic Commands:
Magic | Input | Description
%session_id | None | Return the session ID for the running session.
%status | None | Describes the current session and display SessionID, State,
WorkGroup, EngineVersion and StartTime
%help | None | Displays list of supported magics
%set_log_level | String | Sets the current log level to the provided log level
ls (ERROR|INFO|WARNING etc)
%list_sessions | None | Lists the most recent sessions associated with the current
workgroup
%%sql | String | Run an SQL command against SparkSQL.
```

### %list\_sessions

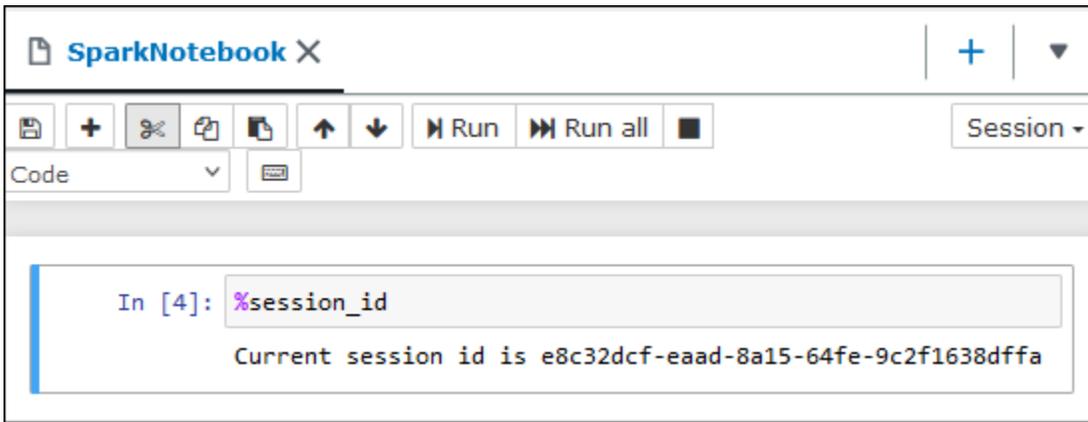
列出與筆記本相關聯的工作階段。每個工作階段的資訊包括工作階段 ID、工作階段狀態，以及工作階段開始和結束的日期和時間。

```
In [12]: %list_sessions
```

SessionId	Status	StartDateTime	EndDateTime
66c32de7-78b9-f2ee-6eb9-d8d9716c6ac8	IDLE	02/16/2023, 19:58:54	
ccc32dda-6dea-6277-d434-5c5da5e1a882	TERMINATED	02/16/2023, 19:30:24	02/16/2023, 19:51:53
e8c32dcf-eaad-8a15-64fe-9c2f1638dffa	TERMINATED	02/16/2023, 19:07:26	02/16/2023, 19:28:53

### %session\_id

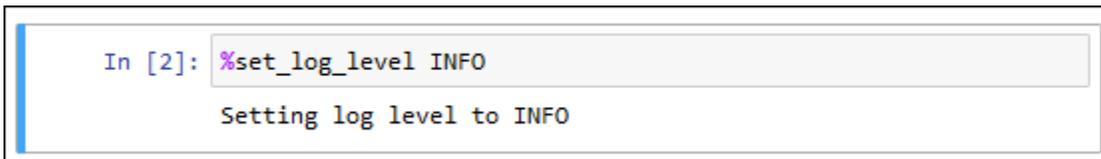
擷取目前的工作階段 ID。



```
SparkNotebook X | + | ▼  
[Icons: Save, Copy, Paste, Undo, Redo, Run, Run all, Stop] Session ▾  
Code ▾ [Icon]  
In [4]: %session_id  
Current session id is e8c32dcf-eaad-8a15-64fe-9c2f1638dffa
```

## %set\_log\_level

設定或重設記錄器，以使用指定的日誌級別。可能值為 DEBUG、ERROR、FATAL、INFO 以及 WARN 或 WARNING。值必須為大寫，不得用單引號或雙引號括住。



```
In [2]: %set_log_level INFO  
Setting log level to INFO
```

## %status

描述目前的工作階段。輸出包括工作階段 ID、工作階段狀態、工作群組名稱、PySpark 引擎版本和工作階段開始時間。此魔術命令需要作用中的工作階段，以擷取工作階段詳細資訊。

以下是狀態的可能值：

正在建立 – 工作階段正在啟動中，包括取得資源。

已建立 – 工作階段已開始。

閒置 – 工作階段能夠接受計算。

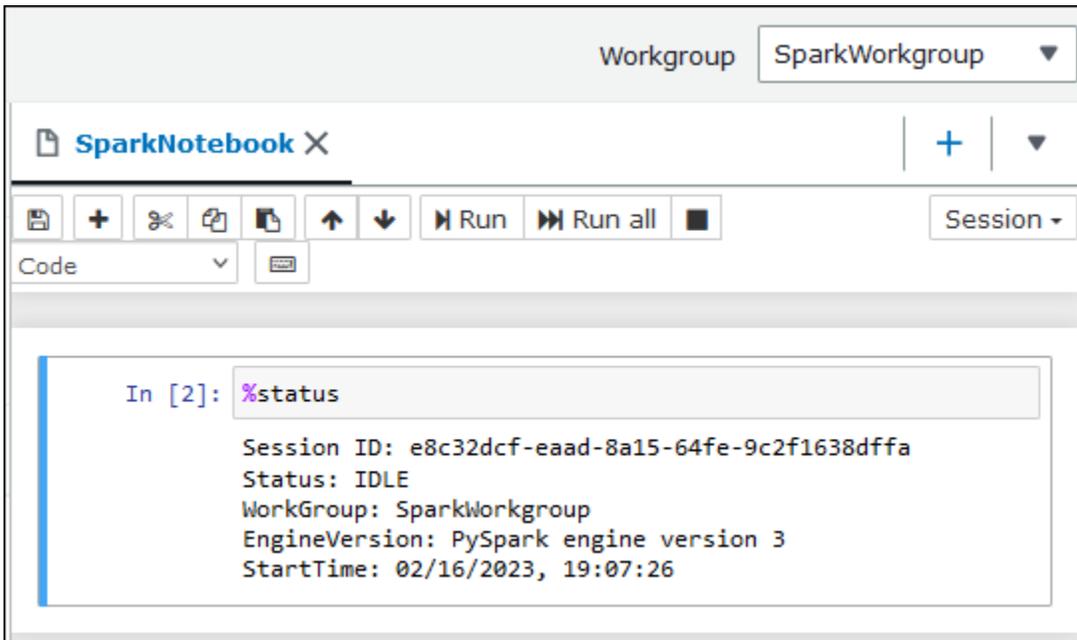
忙碌 – 工作階段正在處理另一項作業，且無法接受計算。

正在終止 – 工作階段正在關閉。

已終止 – 工作階段及其資源不再執行。

降級 – 工作階段沒有健康的協調者。

失敗 – 由於失敗，工作階段及其資源不再執行。



## 用於建立資料圖形的魔術

本節中的行魔術專門用於為特定類型的資料渲染資料或與圖形庫結合使用。

### %table

您可以使用 `%table` 魔術命令，以資料表格式顯示 dataframe 資料。

下下列範例建立了一個包含兩個資料欄和三個資料列資料的 dataframe，然後會以資料表格式顯示資料。

```
In [16]: columns = ["language","users_count"]
data = [("Java", "20000"), ("Python", "100000"), ("Scala", "3000")]
df = spark.createDataFrame(data, columns)
arr = df.collect()
%table arr
```

Calculation started (calculation\_id=12c32e0e-a76e-76e1-a108-707c09599e60) in (session=a6c32df6-dc5f-3390-be39-38bd204513be). Checking calculation status...

Progress:  elapsed time = 00:04s, DPU counts  
100% active/requested = 0/0

Calculation completed.

language	users_count
Java	20000
Python	100000
Scala	3000

`%matplotlib`

[Matplotlib](#) 是用於在 Python 中建立靜態、動畫和交互式可視化的綜合程式庫。在將 matplotlib 程式庫匯入筆記本儲存格之後，您可以使用 `%matplotlib` 魔術指令來建立圖形。

下列範例會匯入 matplotlib 程式庫，建立一組 x 和 y 座標，然後使用 `%matplotlib` 魔術命令建立點的圖形。

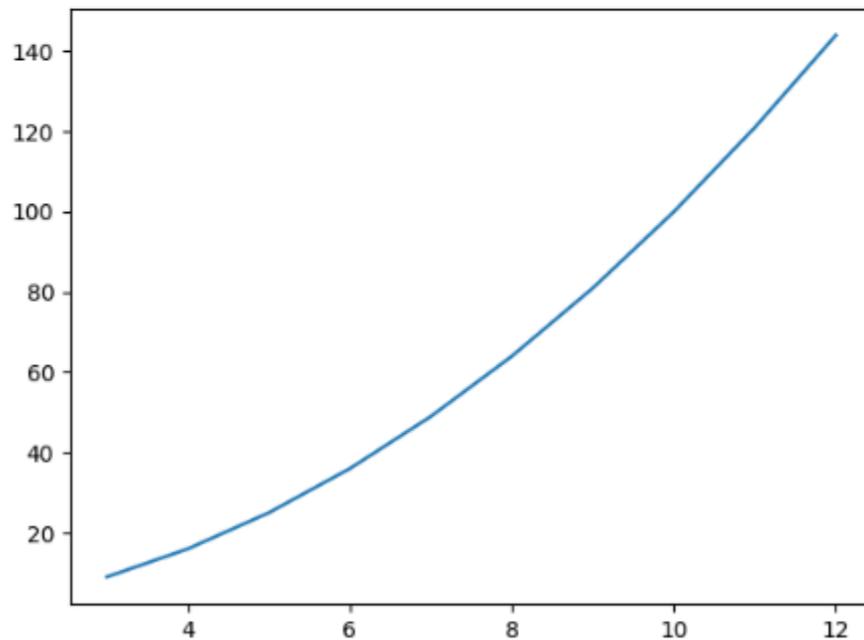
```
import matplotlib.pyplot as plt
x=[3,4,5,6,7,8,9,10,11,12]
y= [9,16,25,36,49,64,81,100,121,144]
plt.plot(x,y)
%matplotlib plt
```

```
In [12]: import matplotlib.pyplot as plt
x=[3,4,5,6,7,8,9,10,11,12]
y= [9,16,25,36,49,64,81,100,121,144]
plt.plot(x,y)
%matplotlib plt
```

Calculation started (calculation\_id=5ac32e04-81b6-9ee7-ce55-539ee2ce383e) in (session=a6c32df6-dc5f-3390-be39-38bd204513be). Checking calculation status...

Progress:  elapsed time =  
100% 00:02s

Calculation completed.



[<matplotlib.lines.Line2D object at 0x7f6e29e580>]

## 同時使用 matplotlib 和 seaborn 程式庫

[Seaborn](#) 是一個用 Python 製作統計圖形程式庫。它以 matplotlib 為建置基礎，並與 [pandas](#) (Python 資料分析) 資料結構緊密整合。您也可以使用 `%matplotlib` 魔術命令來渲染 seaborn 資料。

下列範例同時使用了 matplotlib 和 seaborn 程式庫來建立一個簡單的長條圖。

```
import matplotlib.pyplot as plt
import seaborn as sns

x = ['A', 'B', 'C']
y = [1, 5, 3]

sns.barplot(x, y)
%matplotlib plt
```

```
In [1]: import matplotlib.pyplot as plt
import seaborn as sns

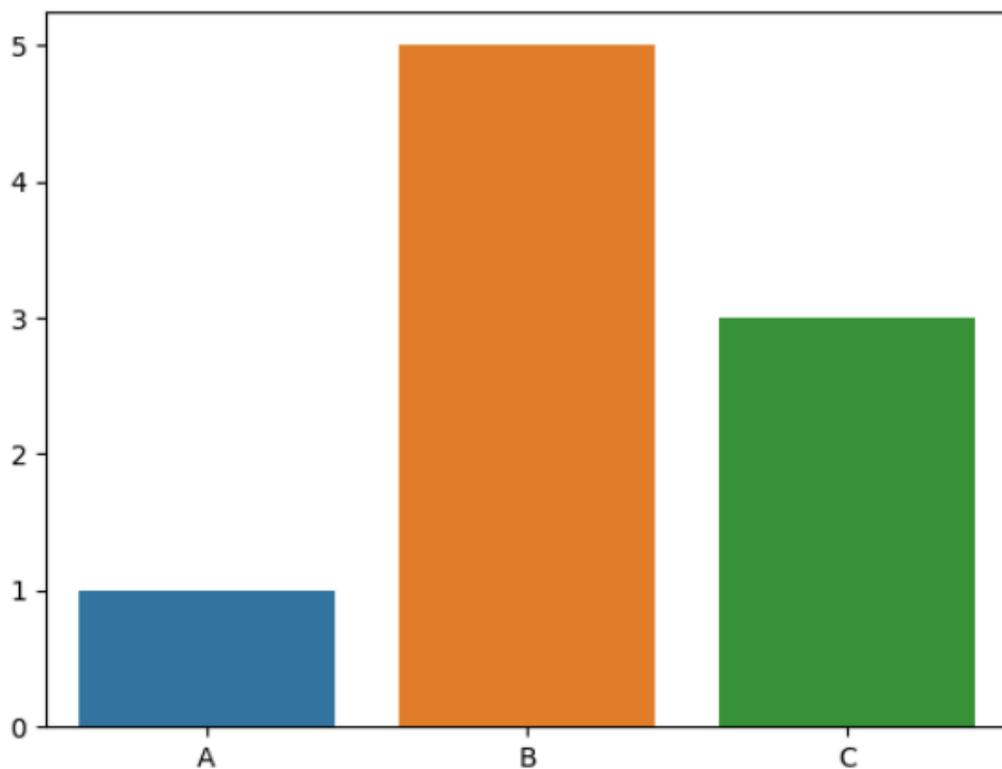
x = ['A', 'B', 'C']
y = [1, 5, 3]

sns.barplot(x, y)
%matplotlib plt
```

Calculation started (calculation\_id=08c32e1b-233b-4a72-6571-1ae7a28a7b78) in (session=64c32e1a-f45e-1d52-b54b-85202e2a9233). Checking calculation status...

Progress: 100%  elapsed time = 00:04s

Calculation completed.



## %plotly

[Plotly](#) 是一個 Python 的開放原始碼圖形程式庫，您可以使用它來製作交互式圖形。您可使用 %plotly 魔法命令來渲染 plotly 資料。

下列範例基於股票價格資料使用 [StringIO](#)、plotly 和 pandas 程式庫來建立 2015 年 2 月和 3 月的股票活動圖表。

```
from io import StringIO
csvString = """
Date,AAPL.Open,AAPL.High,AAPL.Low,AAPL.Close,AAPL.Volume,AAPL.Adjusted,dn,mavg,up,direction
2015-02-17,127.489998,128.880005,126.919998,127.830002,63152400,122.905254,106.7410523,117.9276
2015-02-18,127.629997,128.779999,127.449997,128.720001,44891700,123.760965,107.842423,118.94033
2015-02-19,128.479996,129.029999,128.330002,128.449997,37362400,123.501363,108.8942449,119.8891
2015-02-20,128.619995,129.5,128.050003,129.5,48948400,124.510914,109.7854494,120.7635001,131.74
2015-02-23,130.020004,133,129.660004,133,70974100,127.876074,110.3725162,121.7201668,133.067817
2015-02-24,132.940002,133.600006,131.169998,132.169998,69228100,127.078049,111.0948689,122.6648
2015-02-25,131.559998,131.600006,128.149994,128.789993,74711700,123.828261,113.2119183,123.6296
2015-02-26,128.789993,130.869995,126.610001,130.419998,91287500,125.395469,114.1652991,124.2823
2015-02-27,130,130.570007,128.240005,128.460007,62014800,123.510987,114.9668484,124.8426669,134
2015-03-02,129.25,130.279999,128.300003,129.089996,48096700,124.116706,115.8770904,125.4036668,
2015-03-03,128.960007,129.520004,128.089996,129.360001,37816300,124.376308,116.9535132,125.9551
2015-03-04,129.100006,129.559998,128.320007,128.539993,31666300,123.587892,118.0874253,126.4730
2015-03-05,128.580002,128.75,125.760002,126.410004,56517100,121.539962,119.1048311,126.848667,1
2015-03-06,128.399994,129.369995,126.260002,126.599998,72842100,121.722637,120.190797,127.22883
2015-03-09,127.959999,129.570007,125.059998,127.139999,88528500,122.241834,121.6289771,127.6311
2015-03-10,126.410004,127.220001,123.800003,124.510002,68856600,119.71316,123.1164763,127.92350
"""
csvStringIO = StringIO(csvString)

from io import StringIO
import plotly.graph_objects as go
import pandas as pd
from datetime import datetime
df = pd.read_csv(csvStringIO)
fig = go.Figure(data=[go.Candlestick(x=df['Date'],
open=df['AAPL.Open'],
high=df['AAPL.High'],
low=df['AAPL.Low'],
close=df['AAPL.Close'])])
%plotly fig
```



## 管理筆記本檔案

除了使用筆記本瀏覽器[建立](#)及[開啟](#)筆記本以外，您也可以使用其來重新命名、刪除、匯出或匯入筆記本，或檢視筆記本的工作階段歷史記錄。

### 若要重新命名筆記本

1. [終止](#)任何您要重新命名之筆記本的作用中工作階段。您必須先終止筆記本的作用中工作階段，才能重新命名筆記本。
2. 開啟 Notebook explorer (筆記本瀏覽器)。
3. 在 Notebooks (筆記本) 清單中，為您要重新命名的筆記本選擇選項按鈕。
4. 從 Actions (動作) 選單中，選擇 Rename (重新命名)。

5. 出現 Rename notebook (重新命名筆記本) 提示時，輸入新的名稱，然後選擇 Save (儲存)。新的筆記本名稱會出現在筆記本清單中。

#### 若要刪除筆記本

1. [終止](#)任何您要刪除的筆記本的作用中工作階段。您必須先終止筆記本的作用中工作階段，才能刪除筆記本。
2. 開啟 Notebook explorer (筆記本瀏覽器)。
3. 在 Notebooks (筆記本) 清單中，為您要刪除的筆記本選擇選項按鈕。
4. 在 Actions (動作) 選單中，選擇 Delete (刪除)。
5. 出現 Delete notebook? (刪除筆記本?) 提示時，輸入筆記本的名稱，然後選擇 Delete (刪除) 以確認刪除。筆記本名稱將從筆記本清單中移除。

#### 若要匯出筆記本

1. 開啟 Notebook explorer (筆記本瀏覽器)。
2. 在 Notebooks (筆記本) 清單中，為您要匯出的筆記本選擇選項按鈕。
3. 從 Actions (動作) 選單中，選擇 Export file (匯出檔案)。

#### 若要匯入筆記本

1. 開啟 Notebook explorer (筆記本瀏覽器)。
2. 選擇 Import file (匯入檔案)。
3. 瀏覽至您要匯入之檔案的本機電腦上的位置，然後選擇 Open (開啟)。匯入的筆記本會出現在筆記本清單中。

#### 若要檢視筆記本的工作階段歷史記錄

1. 開啟 Notebook explorer (筆記本瀏覽器)。
2. 在 Notebooks (筆記本) 清單中，為您要檢視之工作階段歷史記錄的筆記本選取選項按鈕。
3. 從 Actions (動作) 選單中，選擇 Session history (工作階段歷史記錄)。
4. 在 History (歷史記錄) 標籤上，選擇 Session ID (工作階段 ID) 以檢視有關工作階段及其計算的資訊。

## 使用 Amazon Athena for Apache Spark 中的非 Hive 資料表格式

當您在 Athena for Spar 中搭配使用工作階段和筆記本時，除了 Apache Hive 資料表，您還可以使用 Linux Foundation Delta Lake、Apache Hudi 和 Apache Iceberg 資料表。

### 考量與限制

當您使用 Apache Hive 與 Athena for Spark 以外的資料表格式時，請考慮下列幾點：

- 除了 Apache Hive 之外，每部筆記型電腦只支援一種資料表格式。若要在 Athena for Spark 中使用多種資料表格式，請為每種資料表格式建立個別的筆記本。如需有關在 Athena for Spark 中建立筆記本的資訊，請參閱 [建立您自己的筆記本](#)。
- 三角洲湖、胡迪和冰山桌格式已在 Athena 的 Spark 上使用 AWS Glue 作為中繼存放區進行測試。您可能可以使用其他中繼存放區，但目前不支援此類用法。
- 若要使用其他資料表格式，請覆寫 Athena 主控台和本文件中所述的預設 `spark_catalog` 屬性。除了其自己的資料表格式之外，這些非 Hive 目錄還可以讀取 Hive 資料表。

### 檔案版本

下表顯示 Amazon Athena for Apache Spark 中支援的非 Hive 資料表版本。

資料表格式	支援的版本
Apache Iceberg	1.2.1
Apache Hudi	0.13
Linux Foundation Delta Lake	2.0.2

在 Athena for Spark 中，這些資料表格式 `.jar` 檔案及其依賴關係會載入到 Spark 驅動程式和執行程式的類路徑中。

如需 AWS 大數據部落格文章，其中說明如何使用 [Amazon Athena 筆記本中的 Spark SQL 使用冰山、Hudi 和三角洲湖表格式](#)，請參閱將 [Amazon Athena 搭配 Spark SQL 搭配開放原始碼交易表格式使用](#)。

### 主題

- [Apache Iceberg](#)
- [Apache Hudi](#)
- [Linux Foundation Delta Lake](#)

## Apache Iceberg

[Apache Iceberg](#) 是 Amazon Simple Storage Service (Amazon S3) 中大型資料集的開放資料表格式。它可針對大型資料表、原子提交、並行寫入和 SQL 相容資料表演化等，為您提供快速查詢效能。

若要使用 Athena for Spark 中的 Apache Iceberg 資料表，請設定以下 Spark 屬性。當您選擇 Apache Iceberg 做為資料表格式時，依預設，通常會在 Athena for Spark 主控台中為您設定這些屬性。如需相關步驟，請參閱 [編輯工作階段詳細資訊](#) 或 [建立您自己的筆記本](#)。

```
"spark.sql.catalog.spark_catalog": "org.apache.iceberg.spark.SparkSessionCatalog",
"spark.sql.catalog.spark_catalog.catalog-impl":
  "org.apache.iceberg.aws.glue.GlueCatalog",
"spark.sql.catalog.spark_catalog.io-impl": "org.apache.iceberg.aws.s3.S3FileIO",
"spark.sql.extensions":
  "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions"
```

下列程序將告訴您如何在 Athena for Spark 筆記本中使用 Apache Iceberg 資料表。在記事本的新儲存格中執行每個步驟。

若要在 Athena for Spark 中使用 Apache Iceberg 資料表

1. 定義要在筆記本中使用的常數。

```
DB_NAME = "NEW_DB_NAME"
TABLE_NAME = "NEW_TABLE_NAME"
TABLE_S3_LOCATION = "s3://DOC-EXAMPLE-BUCKET"
```

2. 創建一個阿帕奇星火 [DataFrame](#)。

```
columns = ["language", "users_count"]
data = [("Golang", 3000)]
df = spark.createDataFrame(data, columns)
```

3. 建立資料庫。

```
spark.sql("CREATE DATABASE {} LOCATION '{}'.format(DB_NAME, TABLE_S3_LOCATION))
```

#### 4. 建立一個空的 Apache Iceberg 資料表。

```
spark.sql("""
CREATE TABLE {}.{} (
  language string,
  users_count int
) USING ICEBERG
""".format(DB_NAME, TABLE_NAME))
```

#### 5. 在資料表中插入一行資料。

```
spark.sql("""INSERT INTO {}.{} VALUES ('Golang',
3000)""").format(DB_NAME, TABLE_NAME))
```

#### 6. 確認您可以查詢新資料表。

```
spark.sql("SELECT * FROM {}.{}".format(DB_NAME, TABLE_NAME)).show()
```

如需使用 Spark 和冰山資料表的詳細資訊 DataFrames 和範例，請參閱 Apache 冰山文件中的 [Spark 查詢](#)。

## Apache Hudi

[Apache Hudi](#) 是一個開源資料管理架構，可簡化增量資料的處理。記錄層級的插入、更新、合併和刪除動作，以更高的精度處理，從而減少了負荷。

若要使用 Athena for Spark 中的 Apache Hudi 資料表，請設定以下 Spark 屬性。當您選擇 Apache Hudi 做為資料表格式時，依預設，通常會在 Athena for Spark 主控台中為您設定這些屬性。如需相關步驟，請參閱 [編輯工作階段詳細資訊](#) 或 [建立您自己的筆記本](#)。

```
"spark.sql.catalog.spark_catalog": "org.apache.spark.sql.hudi.catalog.HoodieCatalog",
"spark.serializer": "org.apache.spark.serializer.KryoSerializer",
"spark.sql.extensions": "org.apache.spark.sql.hudi.HoodieSparkSessionExtension"
```

下列程序將告訴您如何在 Athena for Spark 筆記本中使用 Apache Hudi 資料表。在記事本的新儲存格中執行每個步驟。

若要在 Athena for Spark 中使用 Apache Hudi 資料表

#### 1. 定義要在筆記本中使用的常數。

```
DB_NAME = "NEW_DB_NAME"  
TABLE_NAME = "NEW_TABLE_NAME"  
TABLE_S3_LOCATION = "s3://DOC-EXAMPLE-BUCKET"
```

2. 創建一個阿帕奇星火 [DataFrame](#)。

```
columns = ["language", "users_count"]  
data = [("Golang", 3000)]  
df = spark.createDataFrame(data, columns)
```

3. 建立資料庫。

```
spark.sql("CREATE DATABASE {} LOCATION '{}'.format(DB_NAME, TABLE_S3_LOCATION))
```

4. 建立一個空的 Apache Hudi 資料表。

```
spark.sql("""  
CREATE TABLE {}.{} (  
language string,  
users_count int  
) USING HUDI  
TBLPROPERTIES (  
primaryKey = 'language',  
type = 'mor'  
);  
""".format(DB_NAME, TABLE_NAME))
```

5. 在資料表中插入一列資料。

```
spark.sql("""INSERT INTO {}.{} VALUES ('Golang',  
3000)""").format(DB_NAME, TABLE_NAME))
```

6. 確認您可以查詢新資料表。

```
spark.sql("SELECT * FROM {}.{}".format(DB_NAME, TABLE_NAME)).show()
```

## Linux Foundation Delta Lake

[Linux Foundation Delta Lake](#) 是您可用於大數據分析的資料表格式。您可以使用 Athena for Spark 直接讀取儲存在 Amazon S3 中的 Delta Lake 資料表。

若要使用 Athena for Spark 中的 Delta Lake 資料表，請設定以下 Spark 屬性。當您選擇 Delta Lake 做為資料表格式時，依預設，通常會在 Athena for Spark 主控台中為您設定這些屬性。如需相關步驟，請參閱 [編輯工作階段詳細資訊](#) 或 [建立您自己的筆記本](#)。

```
"spark.sql.catalog.spark_catalog" : "org.apache.spark.sql.delta.catalog.DeltaCatalog",  
"spark.sql.extensions" : "io.delta.sql.DeltaSparkSessionExtension"
```

下列程序將告訴您如何在 Athena for Spark 筆記本中使用 Delta Lake 資料表。在記事本的新儲存格中執行每個步驟。

若要使用 Athena for Spark 中的 Delta Lake 資料表

1. 定義要在筆記本中使用的常數。

```
DB_NAME = "NEW_DB_NAME"  
TABLE_NAME = "NEW_TABLE_NAME"  
TABLE_S3_LOCATION = "s3://DOC-EXAMPLE-BUCKET"
```

2. 創建一個阿帕奇星火 [DataFrame](#)。

```
columns = ["language", "users_count"]  
data = [("Golang", 3000)]  
df = spark.createDataFrame(data, columns)
```

3. 建立資料庫。

```
spark.sql("CREATE DATABASE {} LOCATION '{}'.format(DB_NAME, TABLE_S3_LOCATION))
```

4. 建立一個空的 Delta Lake 資料表。

```
spark.sql("""  
CREATE TABLE {}.{} (  
  language string,  
  users_count int  
) USING DELTA  
""".format(DB_NAME, TABLE_NAME))
```

5. 在資料表中插入一列資料。

```
spark.sql("""INSERT INTO {}.{} VALUES ('Golang',  
3000)""").format(DB_NAME, TABLE_NAME))
```

## 6. 確認您可以查詢新資料表。

```
spark.sql("SELECT * FROM {}.{}".format(DB_NAME, TABLE_NAME)).show()
```

# Amazon Athena for Apache Spark 中的 Python 程式庫支援

本頁面說明 Amazon Athena for Apache Spark 中所用的執行時間、程式庫和套件所使用的術語和需遵循的生命週期管理。

## 定義

- Amazon Athena for Apache Spark 是開源 Apache Spark 的自訂版本。若要查看目前版本，請在記事本儲存格中執行命令 `print(f'{spark.version}')`。
- Athena runtime (Athena 執行時間) 是執行程式碼的環境。該環境包括一個 Python 解釋器和 PySpark 庫。
- external library or package (外部程式庫或套件) 是一種 Java 或 Scala JAR 或 Python 程式庫，其不屬於 Athena 執行時間但會包含在 Athena for Spark 任務中。外部套件可以由 Amazon 建置，也可以由您建置。
- convenience package (便利性套件) 是由 Athena 選取的外部套件集合，您可以選擇將其包含在 Spark 應用程式中。
- bundle (服務包) 結合了 Athena 執行時間和便利性套件。
- user library (使用者程式庫) 是您明確新增至 Athena for Spark 任務的外部程式庫或套件。
  - 使用者程式庫是不屬於便利性套件的外部套件。使用者資料庫需要載入和安裝，就像您寫入一些 .py 檔案，將其壓縮，然後將 .zip 檔案新增至您的應用程式一樣。
- Athena for Spark application (Athena for Spark 應用程式) 是您提交給 Athena for Spark 的任務或查詢。

## 生命週期管理

### 執行時間版本控制和棄用

Athena 執行事件中的主要元件是 Python 解譯器。由於 Python 是一種不斷發展的語言，因此會定期發佈新版本，並移除對舊版本的支援。Athena 不建議您使用已棄用的 Python 解譯器版本來執行程式，強烈建議您盡可能使用最新的 Athena 執行事件。

Athena 執行時間棄用排程如下：

1. Athena 提供新的執行時間後，Athena 將繼續支援上一個執行時間 6 個月。在此期間，Athena 會將安全性修補程式和更新套用至先前的執行時間。
2. 6 個月後，Athena 將終止支援先前的執行時間。Athena 不會再將安全性修補程式和其他更新套用至先前的執行時間。使用先前執行時間的 Spark 應用程式將不再符合技術支援的資格。
3. 12 個月後，您將無法在使用先前執行時間的工作群組中更新或編輯 Spark 應用程式。建議您在此時段結束之前更新 Spark 應用程式。此時段結束後，您仍然可以執行現有的筆記本，但任何仍使用先前執行時間的筆記本將會記錄警告。
4. 18 個月後，您將無法再使用先前的執行時間來執行工作群組中的任務。

## 便利性套件版本控制和棄用

便利性套件的內容會隨時間變化。Athena 偶爾會新增、移除或升級這些便利性套件。

Athena 使用以下適用於便利性套件的準則：

- 便利性套件包含簡單的版本控制結構描述，例如 1、2、3。
- 每個便利性套件版本都包含特定版本的外部套件。Athena 建立便利性套件之後，便利性套件的外部套件集及其對應版本不會變更。
- Athena 會在包含新的外部套件、移除外套件或升級一或多個外部套件的版本時，建立新的便利性套件版本。

Athena 在取代套件使用的 Athena 執行時間時，會棄用便利性套件。Athena 可以更早棄用套件，以限制其支援的服務包數量。

便利性套件棄用排程遵循 Athena 執行時間棄用排程。

## 預先安裝的 Python 程式庫清單

預先安裝的 Python 程式庫包括下列項目。

```
boto3==1.24.31
botocore==1.27.31
certifi==2022.6.15
charset-normalizer==2.1.0
cyclers==0.11.0
cython==0.29.30
docutils==0.19
```

```
fonttools==4.34.4
idna==3.3
jmespath==1.0.1
joblib==1.1.0
kiwisolver==1.4.4
matplotlib==3.5.2
mpmath==1.2.1
numpy==1.23.1
packaging==21.3
pandas==1.4.3
patsy==0.5.2
pillow==9.2.0
plotly==5.9.0
pmdarima==1.8.5
pyathena==2.9.6
pyparsing==3.0.9
python-dateutil==2.8.2
pytz==2022.1
requests==2.28.1
s3transfer==0.6.0
scikit-learn==1.1.1
scipy==1.8.1
seaborn==0.11.2
six==1.16.0
statsmodels==0.13.2
sympy==1.10.1
tenacity==8.0.1
threadpoolctl==3.1.0
urllib3==1.26.10
pyarrow==9.0.0
```

## 備註

- 不支援 MLLib (阿帕奇星火機器學習程式庫) 和 pyspark.ml 套件。
- 目前在 Athena 中不支援 Spark 工作階段。pip install

如需將 Python 程式庫匯入 Amazon Athena 以取得 Apache 星火的資訊，請參閱[將檔案和 Python 程式庫匯入 Amazon Athena for Apache Spark](#)。

## 將檔案和 Python 程式庫匯入 Amazon Athena for Apache Spark

本文件提供了如何將檔案和 Python 程式庫匯入 Amazon Athena for Apache Spark 的範例。

## 考量事項與限制

- Python 版本 – 目前，Athena for Spark 使用 Python 版本 3.9.16。請注意，Python 套件對 Python 次要版本敏感。
- Athena for Spark 架構 – Athena for Spark 在 ARM64 架構上使用 Amazon Linux 2。請注意，某些 Python 程式庫不會為此架構散發二進位檔。
- 二進位共享物件 (SOS) — 由於該 SparkContext [addPyFile](#)方法不會偵測二進位共享物件，因此無法在 Athena for Spark 中使用它來新增依賴於共享物件的 Python 套件。
- 彈性分散式資料集 (RDD) – 不支援 [RDD](#)。
- 資料外匯 — 不支援 `.foreach` 方法。[PySpark DataFrame](#)

## 範例

這些範例使用下列慣例。

- 預留位置 Amazon S3 位置 `s3://DOC-EXAMPLE-BUCKET`。將其替換成您自己的 S3 儲存貯體位置。
- 從 Unix Shell 執行的所有程式碼區塊顯示為 *directory\_name* \$。例如，目錄 `/tmp` 中的命令 `ls` 及其輸出如下所示：

```
/tmp $ ls
```

### 輸出

```
file1 file2
```

- [將檔案寫入本機暫時目錄後，將檔案新增至筆記本](#)
- [從 Amazon S3 匯入檔案](#)
- [新增 Python 檔案並註冊 UDF](#)
- [匯入 Python .zip 檔案](#)
- [將 Python 程式庫的兩個版本匯入為單獨模組](#)
- [從 PyPI 匯入 Python .zip 檔案](#)
- [從具有相依性的 PyPI 匯入 Python .zip 檔案](#)

## 匯入用於計算的文字檔案

本節中的範例說明如何在 Athena for Spark 的筆記本中匯入用於計算的文字檔案。

將檔案寫入本機暫時目錄後，將檔案新增至筆記本

下列範例會顯示如何將檔案寫入本機暫存目錄、將其新增至筆記本，然後進行測試。

```
import os
from pyspark import SparkFiles
tempdir = '/tmp/'
path = os.path.join(tempdir, "test.txt")
with open(path, "w") as testFile:
    _ = testFile.write("5")
sc.addFile(path)

def func(iterator):
    with open(SparkFiles.get("test.txt")) as testFile:
        fileVal = int(testFile.readline())
        return [x * fileVal for x in iterator]

#Test the file
from pyspark.sql.functions import udf
from pyspark.sql.functions import col

udf_with_import = udf(func)
df = spark.createDataFrame([(1, "a"), (2, "b")])
df.withColumn("col", udf_with_import(col('_2'))).show()
```

### 輸出

```
Calculation completed.
+---+---+-----+
| _1| _2|    col|
+---+---+-----+
|  1| a|[aaaaa]|
|  2| b|[bbbbbb]|
+---+---+-----+
```

### 從 Amazon S3 匯入檔案

下列範例會示範如何將檔案從 Amazon S3 匯入筆記本中，然後進行測試。

## 若要將檔案從 Amazon S3 匯入筆記本

1. 建立一個名為 test.txt 的檔案，其中具有包含值 5 的單行。
2. 將檔案新增到 Amazon S3 中的儲存貯體。此範例使用位置 s3://DOC-EXAMPLE-BUCKET。
3. 使用以下程式碼，將檔案匯入您的筆記本並測試檔案。

```
from pyspark import SparkFiles
sc.addFile('s3://DOC-EXAMPLE-BUCKET/test.txt')

def func(iterator):
    with open(SparkFiles.get("test.txt")) as testFile:
        fileVal = int(testFile.readline())
        return [x * fileVal for x in iterator]

#Test the file
from pyspark.sql.functions import udf
from pyspark.sql.functions import col

udf_with_import = udf(func)
df = spark.createDataFrame([(1, "a"), (2, "b")])
df.withColumn("col", udf_with_import(col('_2'))).show()
```

## 輸出

```
Calculation completed.
+---+---+-----+
| _1| _2|    col|
+---+---+-----+
|  1|  a|[aaaaa]|
|  2|  b|[bbbbbb]|
+---+---+-----+
```

## 新增 Python 檔案

本節中的範例說明如何將 Python 檔案和程式庫新增到 Athena 中的 Spark 筆記本。

### 新增 Python 檔案並註冊 UDF

下列範例會示範如何將 Python 檔案從 Amazon S3 匯入您的筆記本，並註冊 UDF。

## 若要將 Python 檔案新增至您的筆記本並註冊 UDF

1. 使用您自己的 Amazon S3 位置，建立包含下列內容的檔案 `s3://DOC-EXAMPLE-BUCKET/file1.py`：

```
def xyz(input):
    return 'xyz - udf ' + str(input);
```

2. 在同一個 S3 位置中，建立包含下列內容的檔案 `s3://DOC-EXAMPLE-BUCKET/file2.py`：

```
from file1 import xyz
def uvw(input):
    return 'uvw -> ' + xyz(input);
```

3. 在 Athena for Spark 筆記本中，執行下列命令。

```
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/file1.py')
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/file2.py')

def func(iterator):
    from file2 import uvw
    return [uvw(x) for x in iterator]

from pyspark.sql.functions import udf
from pyspark.sql.functions import col

udf_with_import = udf(func)

df = spark.createDataFrame([(1, "a"), (2, "b")])

df.withColumn("col", udf_with_import(col('_2'))).show(10)
```

### 輸出

```
Calculation started (calculation_id=1ec09e01-3dec-a096-00ea-57289cdb8ce7) in
(session=c8c09e00-6f20-41e5-98bd-4024913d6cee). Checking calculation status...
Calculation completed.
+---+---+-----+
| _1| _2|          col|
+---+---+-----+
| 1 | a|[uvw -> xyz - ud... |
| 2 | b|[uvw -> xyz - ud... |
```

```
+---+---+-----+-----+
```

## 匯入 Python .zip 檔案

您可以使用 Python `addPyFile` 和 `import` 方法將 Python .zip 檔案匯入您的筆記本。

### Note

您匯入至 Athena Spark 的 .zip 檔案可能只包含 Python 套件。例如，不支援包含 C 型檔案的套件。

若要將 Python **.zip** 檔案匯入您的筆記本

1. 在您的本機電腦上的桌面目錄中 (如 `\tmp`)，建立名為 `moduletest` 的目錄。
2. 在 `moduletest` 目錄中，建立名為 `hello.py` 的檔案，內含下列內容：

```
def hi(input):  
    return 'hi ' + str(input);
```

3. 在同一個目錄中，新增名稱為 `__init__.py` 的空檔案。

如果您列出目錄內容，則其現在應如下所示。

```
/tmp $ ls moduletest  
__init__.py      hello.py
```

4. 使用 `zip` 命令將兩個模組檔案放入名為 `moduletest.zip` 的檔案。

```
moduletest $ zip -r9 ../moduletest.zip *
```

5. 將 .zip 檔案上傳至 Amazon S3 中的儲存貯體。
6. 使用以下程式碼，將 Python.zip 檔案匯入您的筆記本。

```
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/moduletest.zip')  
  
from moduletest.hello import hi  
  
from pyspark.sql.functions import udf  
from pyspark.sql.functions import col
```

```

hi_udf = udf(hi)

df = spark.createDataFrame([(1, "a"), (2, "b")])

df.withColumn("col", hi_udf(col('_2'))).show()

```

## 輸出

```

Calculation started (calculation_id=6ec09e8c-6fe0-4547-5f1b-6b01adb2242c) in
(session=dcc09e8c-3f80-9cdc-bfc5-7effa1686b76). Checking calculation status...
Calculation completed.
+---+---+---+
| _1| _2| col|
+---+---+---+
|  1|  a|hi a|
|  2|  b|hi b|
+---+---+---+

```

## 將 Python 程式庫的兩個版本匯入為單獨模組

下列程式碼範例說明如何從 Amazon S3 中的某個位置新增和匯入兩個不同版本的 Python 程式庫做為兩個獨立的模組。程式碼會從 S3 新增每個程式庫檔案、匯入檔案，然後列印程式庫版本以驗證匯入。

```

sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/python-third-party-libs-test/
simplejson_v3_15.zip')
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/python-third-party-libs-test/
simplejson_v3_17_6.zip')

import simplejson_v3_15
print(simplejson_v3_15.__version__)

```

## 輸出

```
3.15.0
```

```

import simplejson_v3_17_6
print(simplejson_v3_17_6.__version__)

```

## 輸出

## 3.17.6

## 從 PyPI 匯入 Python .zip 檔案

此範例使用 pip 命令從 [Python Package Index \(PyPI\)](#) 下載 [bpabel/piglatin](#) 專案的 Python .zip 檔案。

## 若要從 PyPI 匯入 Python .zip 檔案

1. 在本機桌面上，使用下列命令建立名為 testpiglatin 的目錄並建立虛擬環境。

```
/tmp $ mkdir testpiglatin
/tmp $ cd testpiglatin
testpiglatin $ virtualenv .
```

## 輸出

```
created virtual environment CPython3.9.6.final.0-64 in 410ms
creator CPython3Posix(dest=/private/tmp/testpiglatin, clear=False,
  no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle,
  via=copy, app_data_dir=/Users/user1/Library/Application Support/virtualenv)
added seed packages: pip==22.0.4, setuptools==62.1.0, wheel==0.37.1
activators
  BashActivator, CShellActivator, FishActivator, NushellActivator, PowerShellActivator, PythonAct
```

2. 建立名為 unpacked 的子目錄，以啟動項目。

```
testpiglatin $ mkdir unpacked
```

3. 使用 pip 命令，將專案安裝到 unpacked 目錄。

```
testpiglatin $ bin/pip install -t $PWD/unpacked piglatin
```

## 輸出

```
Collecting piglatin
Using cached piglatin-1.0.6-py2.py3-none-any.whl (3.1 kB)
Installing collected packages: piglatin
Successfully installed piglatin-1.0.6
```

4. 檢查目錄的內容。

```
testpiglating $ ls
```

輸出

```
bin lib pyvenv.cfg unpacked
```

5. 變更為 unpacked 目錄並顯示內容。

```
testpiglating $ cd unpacked
unpacked $ ls
```

輸出

```
piglating piglating-1.0.6.dist-info
```

6. 使用 zip 命令，將 piglating 專案的內容放入名為 library.zip 的檔案。

```
unpacked $ zip -r9 ../library.zip *
```

輸出

```
adding: piglating/ (stored 0%)
adding: piglating/__init__.py (deflated 56%)
adding: piglating/__pycache__/ (stored 0%)
adding: piglating/__pycache__/__init__.cpython-39.pyc (deflated 31%)
adding: piglating-1.0.6.dist-info/ (stored 0%)
adding: piglating-1.0.6.dist-info/RECORD (deflated 39%)
adding: piglating-1.0.6.dist-info/LICENSE (deflated 41%)
adding: piglating-1.0.6.dist-info/WHEEL (deflated 15%)
adding: piglating-1.0.6.dist-info/REQUESTED (stored 0%)
adding: piglating-1.0.6.dist-info/INSTALLER (stored 0%)
adding: piglating-1.0.6.dist-info/METADATA (deflated 48%)
```

7. (選用) 使用以下命令在本機測試匯入。
  - a. 將 Python 路徑設定為 library.zip 檔案位置並啟動 Python。

```
/home $ PYTHONPATH=/tmp/testpiglating/library.zip
/home $ python3
```

## 輸出

```
Python 3.9.6 (default, Jun 29 2021, 06:20:32)
[Clang 12.0.0 (clang-1200.0.32.29)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
```

- b. 匯入程式庫並執行測試命令。

```
>>> import piglatin
>>> piglatin.translate('hello')
```

## 輸出

```
'ello-hay'
```

8. 使用如下命令，從 Amazon S3 新增 .zip 檔案，將檔案匯入 Athena 中的筆記本，然後對其進行測試。

```
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/library.zip')

import piglatin
piglatin.translate('hello')

from pyspark.sql.functions import udf
from pyspark.sql.functions import col

hi_udf = udf(piglatin.translate)

df = spark.createDataFrame([(1, "hello"), (2, "world")])

df.withColumn("col", hi_udf(col('_2'))).show()
```

## 輸出

```
Calculation started (calculation_id=e2c0a06e-f45d-d96d-9b8c-ff6a58b2a525) in
(session=82c0a06d-d60e-8c66-5d12-23bcd55a6457). Checking calculation status...
Calculation completed.
+---+-----+-----+
| _1|  _2|    col|
+---+-----+-----+
```

```
| 1|hello|ello-hay|  
| 2|world|orld-way|  
+--+-----+-----+
```

## 從具有相依性的 PyPI 匯入 Python .zip 檔案

此範例從 PyPI 匯入 [md2gemini](#) 套件，其中該套件會將 Markdown 格式的文字轉換為 [Gemini](#) 文字格式。套件具有以下相依性：

```
cjkrwrap  
mistune  
wcwidth
```

## 若要匯入具有相依性的 Python .zip 檔案

1. 在本機電腦上，使用下列命令建立名為 `testmd2gemini` 的目錄並建立虛擬環境。

```
/tmp $ mkdir testmd2gemini  
/tmp $ cd testmd2gemini  
testmd2gemini$ virtualenv .
```

2. 建立名為 `unpacked` 的子目錄，以啟動項目。

```
testmd2gemini $ mkdir unpacked
```

3. 使用 `pip` 命令，將專案安裝到 `unpacked` 目錄。

```
/testmd2gemini $ bin/pip install -t $PWD/unpacked md2gemini
```

## 輸出

```
Collecting md2gemini  
  Downloading md2gemini-1.9.0-py3-none-any.whl (31 kB)  
Collecting wcwidth  
  Downloading wcwidth-0.2.5-py2.py3-none-any.whl (30 kB)  
Collecting mistune<3,>=2.0.0  
  Downloading mistune-2.0.2-py2.py3-none-any.whl (24 kB)  
Collecting cjkrwrap  
  Downloading CJKwrap-2.2-py2.py3-none-any.whl (4.3 kB)  
Installing collected packages: wcwidth, mistune, cjkrwrap, md2gemini
```

```
Successfully installed cjkwrap-2.2 md2gemini-1.9.0 mistune-2.0.2 wcwidth-0.2.5
...
```

#### 4. 變更為 unpacked 目錄並檢查內容。

```
testmd2gemini $ cd unpacked
unpacked $ ls -lah
```

#### 輸出

```
total 16
drwxr-xr-x 13 user1 wheel 416B Jun 7 18:43 .
drwxr-xr-x 8 user1 wheel 256B Jun 7 18:44 ..
drwxr-xr-x 9 user1 staff 288B Jun 7 18:43 CJKwrap-2.2.dist-info
drwxr-xr-x 3 user1 staff 96B Jun 7 18:43 __pycache__
drwxr-xr-x 3 user1 staff 96B Jun 7 18:43 bin
-rw-r--r-- 1 user1 staff 5.0K Jun 7 18:43 cjkwrap.py
drwxr-xr-x 7 user1 staff 224B Jun 7 18:43 md2gemini
drwxr-xr-x 10 user1 staff 320B Jun 7 18:43 md2gemini-1.9.0.dist-info
drwxr-xr-x 12 user1 staff 384B Jun 7 18:43 mistune
drwxr-xr-x 8 user1 staff 256B Jun 7 18:43 mistune-2.0.2.dist-info
drwxr-xr-x 16 user1 staff 512B Jun 7 18:43 tests
drwxr-xr-x 10 user1 staff 320B Jun 7 18:43 wcwidth
drwxr-xr-x 9 user1 staff 288B Jun 7 18:43 wcwidth-0.2.5.dist-info
```

#### 5. 使用 zip 命令，將 md2gemini 專案的內容放入名為 md2gemini.zip 的檔案。

```
unpacked $ zip -r9 ../md2gemini *
```

#### 輸出

```
adding: CJKwrap-2.2.dist-info/ (stored 0%)
adding: CJKwrap-2.2.dist-info/RECORD (deflated 37%)
....
adding: wcwidth-0.2.5.dist-info/INSTALLER (stored 0%)
adding: wcwidth-0.2.5.dist-info/METADATA (deflated 62%)
```

#### 6. (選用) 使用下列指令來測試程式庫是否可在您的本機電腦上運作。

##### a. 將 Python 路徑設定為 md2gemini.zip 檔案位置並啟動 Python。

```
/home $ PYTHONPATH=/tmp/testmd2gemini/md2gemini.zip
```

```
/home python3
```

b. 匯入程式庫並執行測試。

```
>>> from md2gemini import md2gemini
>>> print(md2gemini('[abc](https://abc.def)'))
```

輸出

```
https://abc.def abc
```

7. 使用以下命令，從 Amazon S3 新增 .zip 檔案，將檔案匯入 Athena 中的筆記本，然後執行 UDF 測試。

```
# (non udf test)
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/md2gemini.zip')
from md2gemini import md2gemini
print(md2gemini('[abc](https://abc.def)'))
```

輸出

```
Calculation started (calculation_id=0ac0a082-6c3f-5a8f-eb6e-f8e9a5f9bc44) in
(session=36c0a082-5338-3755-9f41-0cc954c55b35). Checking calculation status...
Calculation completed.
=> https://abc.def (https://abc.def/) abc
```

8. 使用下列命令執行 UDF 測試。

```
# (udf test)

from pyspark.sql.functions import udf
from pyspark.sql.functions import col
from md2gemini import md2gemini

hi_udf = udf(md2gemini)
df = spark.createDataFrame([(1, "[first website](https://abc.def)"), (2, "[second
website](https://aws.com)")]])
df.withColumn("col", hi_udf(col('_2'))).show()
```

輸出

```

Calculation started (calculation_id=60c0a082-f04d-41c1-a10d-d5d365ef5157) in
(session=36c0a082-5338-3755-9f41-0cc954c55b35). Checking calculation status...
Calculation completed.
+---+-----+-----+-----+
| _1|           _2|           col|
+---+-----+-----+-----+
| 1|[first website](h...|=> https://abc.de...|
| 2|[second website](...|=> https://aws.co...|
+---+-----+-----+-----+

```

## 新增 JAR 檔案和自訂 Spark 組態

當您在 Amazon Athena for Apache Spark 中建立或編輯工作階段時，您可以使用 [Spark 屬性](#) 來指定工作階段的 .jar 檔案、套件或其他自訂組態。若要指定您的 Spark 屬性，您可以使用 Athena 主控台、AWS CLI、或 Athena API。

### 使用 Athena 主控台指定 Spark 屬性

在 Athena 主控台中，您可以在 [建立筆記本](#) 或 [編輯目前工作階段](#) 時指定 Spark 屬性。

若要在建立記事本或編輯工作階段詳細資訊對話方塊中新增屬性

1. 展開 Spark 屬性。
2. 若要新增屬性，請使用在資料表中編輯或在 JSON 中編輯選項。
  - 對於在資料表中編輯選項，請選擇新增屬性以新增屬性，或選擇移除以移除屬性。使用關鍵字和值方塊，以輸入屬性名稱及其值。
    - 若要新增自訂 .jar 檔案，請使用 spark.jars 屬性。
    - 若要指定套件檔案，請使用 spark.jars.packages 屬性。
  - 若要直接輸入和編輯您的設定，請選擇在 JSON 中編輯選項。在 JSON 文字編輯器中，您可以執行以下任務：
    - 選擇複製，以將 JSON 文字複製到剪貼簿。
    - 選擇清除，以從 JSON 編輯器中移除所有文字。
    - 選擇設定 (齒輪) 圖示，以設定換行或選擇 JSON 編輯器的顏色主題。

## 備註

- 您可以在 Athena for Spark 中設定屬性，這與直接在 [SparkConf](#) 物件上設定 [Spark 屬性](#)。
- 所有 Spark 屬性均以 spark. 字首開頭。帶有其他字首的屬性會被忽略。
- 並非所有 Spark 屬性都適用於 Athena 的自訂組態。如果您提交的 StartSession 請求具有限制組態，工作階段將無法啟動。
  - 您無法使用 spark.athena. 字首，因為其已保留。

## 使用 AWS CLI 或 Athena API 提供自訂組態

若要使用 AWS CLI 或 Athena API 提供工作階段組態，請使用 [StartSession](#) API 動作或 [start-session](#) CLI 命令。在您的 StartSession 請求中，使用 [EngineConfiguration](#) 物件的 SparkProperties 欄位，以 JSON 格式傳遞您的組態資訊。這樣，將會使用您指定的組態啟動工作階段。如需請求語法，請參閱《Amazon Athena API 參考》中的 [StartSession](#)。

## 疑難排解工作階段開始錯誤

當工作階段啟動期間發生自訂組態錯誤時，Athena for Spark 主控台會顯示錯誤訊息橫幅。若要對工作階段啟動錯誤進行疑難排解，您可以檢查工作階段狀態變更或記錄資訊。

### 檢視工作階段變更資訊

您可以從 Athena 筆記本編輯器或 Athena API 取得有關工作階段狀態變更的詳細資訊。

在 Athena 主控台中檢視工作階段狀態資訊

1. 在 Athena 筆記本編輯器中，從右上角的工作階段選單中，選擇檢視詳細資訊。
2. 檢視目前工作階段索引標籤。工作階段資訊區段會顯示工作階段 ID、工作群組、狀態和狀態變更原因等資訊。

下列螢幕擷取範例會針對 Athena 中的 Spark 工作階段錯誤，顯示工作階段資訊對話方塊中狀態變更原因區段中的資訊。

Session information		
Session ID [REDACTED]	Status ⚠ Degraded	Run time PySpark engine version 3
Workgroup [REDACTED]	Creation time 2023-05-10T15:58:59.256-07:00	Coordinator size 1 DPU
Description -	Last active 2023-05-10T19:00:54.189-07:00	State change reason Athena experienced a Spark session error. To troubleshoot, look for error messages from AthenaSparkSessionErrorLogger in your CloudWatch log. If no such error is present, contact AWS Support. For information about Spark logging, see <a href="https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-logging.html">https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-logging.html</a> .

## 使用 Athena API 檢視工作階段狀態資訊

- 在 Athena API 中，您可以在 [SessionStatus](#) 物件的 StateChangeReason 欄位中尋找工作階段狀態變更資訊。

### Note

手動停止工作階段之後，或者如果工作階段在閒置逾時 (預設值為 20 分鐘) 之後停止，則 StateChangeReason 的值會變更為根據請求終止工作階段。

## 使用記錄疑難排解工作階段啟動錯誤

[Amazon CloudWatch](#) 會記錄工作階段開始期間發生的自訂組態錯誤。在 CloudWatch Logs 中，從 AthenaSparkSessionErrorLogger 中搜尋錯誤訊息，以便對失敗的工作階段啟動進行疑難排解。

如需有關 Spark 記錄的詳細資訊，請參閱 [在 Athena 中記錄 Spark 應用程式事件](#)。

如需有關在 Athena for Spark 中疑難排解工作階段的詳細資訊，請參閱 [工作階段疑難排解](#)。

## 支援的資料和儲存格式

下表顯示了 Athena for Apache Spark 中原生支援的格式。

資料格式	讀取	寫入	寫入壓縮
parquet	是	是	無、未壓縮、snappy、gzip
orc	是	是	無、snappy、zlib、lzo
json	是	是	bzip2、gzip、deflate
csv	是	是	bzip2、gzip、deflate
text	是	是	無、bzip2、gzip、deflate
二進位檔案	是	N/A	N/A

## 使用 CloudWatch 指標監控 Apache Spark 計算

當為您已啟用 Spark 的工作群組選取 [Publish CloudWatch metrics](#) 選項時，Athena 會將計算相關指標發佈至 Amazon CloudWatch。您可以在 CloudWatch 主控台中建立自訂儀表板，設定指標的警示和觸發。

Athena 會將以下指標發佈至 AmazonAthenaForApacheSpark 命名空間下的 CloudWatch 主控台：

- DPUCount – 工作階段期間用來執行計算所消耗的 DPU 數目。

該指標具有下列兩個維度：

- SessionId – 要提交計算的工作階段 ID。
- WorkGroup – 工作群組的名稱。

若要在 Amazon CloudWatch 主控台中檢視已啟用 Spark 的工作群組的指標

1. 在 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇 Metrics (指標)、All metrics (所有指標)。
3. 選取 AmazonAthenaForApacheSpark 命名空間。

## 使用 CLI 來檢視指標

- 執行下列任意一項：
  - 若要列出已啟用 Athena Spark 的工作群組的指標，請開啟命令提示，然後使用下列命令：

```
aws cloudwatch list-metrics --namespace "AmazonAthenaForApacheSpark"
```

- 若要列出所有可用的指標，請使用以下命令：

```
aws cloudwatch list-metrics
```

## Athena 中的 Apache Spark 計算的 CloudWatch 指標和維度清單

如果您已在已啟用 Spark 的 Athena 工作群組中啟用 CloudWatch 指標，其會按照工作群組將以下指標傳送到 CloudWatch。指標使用 AmazonAthenaForApacheSpark 命名空間。

指標名稱	描述
DPUCount	工作階段期間用來執行計算所消耗的 DPU (資料處理單位) 數目。DPU 是相對的處理能力，包含 4 個 vCPU 的運算容量和 16 GB 的記憶體。

該指標具有下列維度。

維度	描述
SessionId	要提交計算的工作階段 ID。
WorkGroup	工作群組的名稱。

## 啟用請求者支付 Athena for Spark 中的 Amazon S3 儲存貯體費用

將 Amazon S3 儲存貯體設定為請求者付費時，執行查詢的使用者帳戶需支付與查詢相關聯的資料存取和資料傳輸費用。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[使用請求者支付儲存貯體，以進行儲存傳輸和使用](#)。

在 Athena for Spark 中，請求者支付儲存貯體是依工作階段啟用的，而非工作群組。在高層級上，啟用請求者支付時段包含下列步驟：

1. 在 Amazon S3 主控台中，啟用請求者支付儲存貯體的屬性，並新增儲存貯體政策以指定存取權。
2. 在 IAM 主控台中，建立允許存取儲存貯體的 IAM 政策，然後將政策連接至將用於存取請求者支付儲存貯體之 IAM 角色的 IAM 角色。
3. 在 Athena for Spark 中，新增工作階段屬性，以啟用請求者支付功能。

## 1. 在 Amazon S3 儲存貯體上啟用請求者支付並新增儲存貯體政策

若要為 Amazon S3 儲存貯體啟用請求者支付

1. 開啟位於 <https://console.aws.amazon.com/s3/> 的 Amazon S3 主控台。
2. 在儲存貯體清單中，按一下要啟用請求者支付的儲存貯體名稱。
3. 在儲存貯體頁面上，選擇屬性索引標籤。
4. 向下捲動到請求者支付區段，然後選擇編輯。
5. 在編輯請求者支付頁面上，選擇啟用，然後選擇儲存變更。
6. 選擇 許可 標籤。
7. 在儲存貯體政策區段中，選擇編輯。
8. 在編輯儲存貯體政策頁面上，將儲存貯體政策套用至您要確定來源儲存貯體的政策。下列範例政策可讓您存取所有 AWS 主體 ("AWS": "\*")，但您的存取權限可能會更精細。例如，您可能只想在其他帳戶中指定特定 IAM 角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-
        bucket",
```

```

    "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-bucket/"
  "*"
    ]
  }
]
}

```

## 2. 建立 IAM 角色，並將其連接到 IAM 政策

接下來，建立 IAM 政策以允許存取儲存貯體。然後，您將政策連接到將用於存取請求者支付儲存貯體的角色。

為請求者支付儲存貯體建立 IAM 政策，並將該政策連接其中

1. 前往網址 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇政策。
3. 選擇 Create policy (建立政策)。
4. 選擇 JSON。
5. 在政策編輯器中，新增如下政策：

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-
bucket",
        "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-bucket/
*"
      ]
    }
  ]
}

```

6. 選擇 Next (下一步)。
7. 在檢閱和建立頁面上，輸入政策的名稱和可選描述，然後選擇建立政策。

8. 在導覽窗格中，選擇 Roles (角色)。
9. 在角色頁面上，尋找您要使用的角色，然後選擇角色名稱連結。
10. 在許可政策區段中，選擇新增許可、連接政策。
11. 在其他許可政策區段中，選取您建立的政策的核取方塊，然後選擇新增許可。

### 3. 新增 Athena for Spark 工作階段屬性

設定 Amazon S3 儲存貯體和請求者支付的相關許可後，您可以在 Athena for Spark 工作階段中啟用此功能。

若要啟用請求者支付 Athena for Spark 工作階段中的儲存貯體費用

1. 在筆記本編輯器中，從右上角的 Session (工作階段) 選單中，選擇 Edit session (編輯工作階段)。
2. 展開 Spark 屬性。
3. 選擇在 JSON 中編輯。
4. 在 JSON 文字方塊中，輸入下列內容：

```
{
  "spark.hadoop.fs.s3.useRequesterPaysHeader": "true"
}
```

5. 選擇 Save (儲存)。

### 啟用 Apache Spark 加密

您可以在 Athena 上啟用 Apache Spark 加密。這樣做可以加密 Spark 節點之間傳輸中的資料，也可以加密 Spark 本機儲存的靜態資料。若要增強此資料的安全性，Athena 使用下列加密組態：

```
spark.io.encryption.keySizeBits="256"
spark.io.encryption.keygen.algorithm="HmacSHA384"
```

若要啟用 Spark 加密，您可以使用 Athena 主控台、AWS CLI 或 Athena API。

## 使用 Athena 主控台啟用 Spark 加密

若要建立已啟用 Spark 加密的新筆記本

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。
3. 執行下列任意一項：
  - 在 Notebook explorer (筆記本瀏覽器) 中，選擇 Create notebook (建立筆記本)。
  - 在 Notebook editor (筆記本瀏覽器) 中，選擇 Create notebook (建立筆記本)，或選擇加號圖示 (+) 以新增筆記本。
4. 在筆記本名稱中，輸入筆記本的名稱。
5. 展開 Spark 屬性選項。
6. 選取開啟 Spark 加密。
7. 選擇 Create (建立)。

您建立的筆記本工作階段已加密。如常使用新的筆記本。當您稍後啟動使用筆記本的新工作階段時，新的工作階段也會加密。

您也可以使用 Athena 主控台為現有筆記本啟用 Spark 加密。

若要啟用現有筆記本的加密

1. 為先前建立的記事本 [開啟新的工作階段](#)。
2. 在筆記本編輯器中，從右上角的 Session (工作階段) 選單中，選擇 Edit session (編輯工作階段)。
3. 在編輯工作階段詳細資訊對話方塊中，展開 Spark 屬性。
4. 選取開啟 Spark 加密。
5. 選擇 Save (儲存)。

主控台會啟動已啟用加密的新工作階段。您為此筆記本建立的最新工作階段也會啟用加密。

## 使用 AWS CLI 啟用 Spark 加密

您可以在啟動工作階段時指定適當的 Spark 屬性，進而使用 AWS CLI 啟用加密。

## 若要使用 AWS CLI 啟用 Spark 加密

1. 使用類似下列命令來建立指定 Spark 加密屬性的引擎組態 JSON 物件。

```
ENGINE_CONFIGURATION_JSON=$(
  cat <<EOF
{
  "CoordinatorDpuSize": 1,
  "MaxConcurrentDpus": 20,
  "DefaultExecutorDpuSize": 1,
  "SparkProperties": {
    "spark.authenticate": "true",
    "spark.io.encryption.enabled": "true",
    "spark.network.crypto.enabled": "true"
  }
}
EOF
)
```

2. 在 AWS CLI 中，使用 `athena start-session` 命令並將您建立的 JSON 物件傳遞給 `--engine-configuration` 引數，如下列範例所示：

```
aws athena start-session \
  --region "region" \
  --work-group "your-work-group" \
  --engine-configuration "$ENGINE_CONFIGURATION_JSON"
```

## 使用 Athena API 啟用 Spark 加密

若要使用 Athena API 啟用 Spark 加密，請使用 [StartSession](#) 動作及其 [EngineConfiguration](#) `SparkProperties` 參數，以在 `StartSession` 請求中指定加密組態。

## 在 Athena 中設定 Spark 的跨帳戶 AWS Glue 存取

本主題將說明如何將消費者帳戶 `666666666666` 和擁有者帳戶 `999999999999` 設定為跨帳戶 AWS Glue 存取權。設定帳戶後，消費者帳戶可以在擁有者的 AWS Glue 資料庫和資料表上從 Athena for Spark 執行查詢。

## 1. 在中 AWS Glue，提供對消費者角色的存取

在中 AWS Glue，擁有者會建立原則，提供取用者角色存取擁有者 AWS Glue 資料目錄的權限。

新增允許取用者角色存取擁有者資料目錄的 AWS Glue 原則

1. 使用目錄擁有者的帳戶，登入 AWS Management Console。
2. [請在以下位置開啟 AWS Glue 主控台。](https://console.aws.amazon.com/glue/) <https://console.aws.amazon.com/glue/>
3. 在導覽窗格中，展開資料目錄，然後選擇目錄設定。
4. 在資料目錄設定頁面的許可區段中，新增如下政策。此政策為消費者帳戶 **666666666666** 提供存取所有者帳戶 **999999999999** 中的資料目錄的角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Cataloguers",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::666666666666:role/Admin",
          "arn:aws:iam::666666666666:role/AWSAthenaSparkExecutionRole"
        ]
      },
      "Action": "glue:*",
      "Resource": [
        "arn:aws:glue:us-west-2:999999999999:catalog",
        "arn:aws:glue:us-west-2:999999999999:database/*",
        "arn:aws:glue:us-west-2:999999999999:table/*"
      ]
    }
  ]
}
```

## 2. 設定消費者帳戶，以用於存取

在取用者帳戶中，建立允許存取擁有者 AWS Glue Data Catalog、資料庫和資料表的原則，並將原則附加至角色。下列範例會使用消費者帳戶 **666666666666**。

## 若要建立存取擁有者的 AWS Glue 原則 AWS Glue Data Catalog

1. 使用消費者帳戶，登入 AWS Management Console。
2. 前往 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
3. 在導覽窗格中，展開存取管理，然後選擇政策。
4. 選擇建立政策。
5. 在指定許可頁面上，選擇 JSON。
6. 在政策編輯器中，輸入如下所示的 JSON 陳述式，允許對擁有者帳戶的資料目錄 AWS Glue 執行動作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "glue:*",
      "Resource": [
        "arn:aws:glue:us-east-1:999999999999:catalog",
        "arn:aws:glue:us-east-1:999999999999:database/*",
        "arn:aws:glue:us-east-1:999999999999:table/*"
      ]
    }
  ]
}
```

7. 選擇下一步。
8. 在檢閱和建立頁面上，針對政策名稱，輸入該政策的名稱。
9. 選擇建立政策。

接下來，您可以在消費者帳戶中使用 IAM 主控台，將剛建立的政策連接到消費者帳戶用來存取擁有者資料目錄的 IAM 角色或角色。

### 將 AWS Glue 策略附加到用戶帳戶中的角色

1. 在消費者帳戶 IAM 主控台導覽窗格中，選擇角色。
2. 在角色頁面上，尋找您要連接政策的角色。
3. 選擇新增許可，然後選擇連接政策。
4. 尋找您剛建立的政策。

5. 選取政策的核取方塊，然後選擇新增許可。
6. 重複這些步驟，將政策新增至您要使用的其他角色。

### 3. 設定工作階段並建立查詢

在 Athena Spark 的請求者帳戶中，使用指定的角色，藉由[建立筆記本](#)或[編輯目前工作階段](#)來建立工作階段，以測試存取。[設定工作階段屬性](#)時，請指定下列其中一個：

- Glue 目錄分隔符號 – 使用這種方法，您可以在查詢中包含擁有者帳戶 ID。如果您要使用工作階段來查詢不同擁有者的資料目錄，請使用此方法。
- Glue 目錄 ID – 使用這種方法，您可以直接查詢資料庫。如果您要使用工作階段來僅查詢單一擁有者的資料目錄，則此方法比較方便。

#### 使用 AWS Glue 目錄分隔符號方法

當您編輯工作階段屬性時，請新增下列項目：

```
{
  "spark.hadoop.aws.glue.catalog.separator": "/"
}
```

當您在儲存格中執行查詢時，請使用類似下列範例中的語法。請注意，在 FROM 子句中，資料庫名稱前需要目錄 ID 和分隔符號。

```
df = spark.sql('SELECT requestip, uri, method, status FROM `999999999999/
mydatabase`.cloudfront_logs LIMIT 5')
df.show()
```

#### 使用目 AWS Glue 錄 ID 方法

當您編輯工作階段屬性時，請輸入下列屬性。將 **999999999999** 取代為擁有者帳戶 ID。

```
{
  "spark.hadoop.hive.metastore.glue.catalogid": "999999999999"
}
```

當您在儲存格中執行查詢時，請使用如下語法。請注意，在 FROM 子句中，資料庫名稱前不需要目錄 ID 和分隔符號。

```
df = spark.sql('SELECT * FROM mydatabase.cloudfront_logs LIMIT 10')
df.show()
```

## 其他資源

[跨帳戶存取 AWS Glue 資料目錄](#)

[使用AWS Lake Formation 開發人員指南中的 Lake Formation 管理跨帳戶權限。](#) AWS Glue

在AWS 規範性指導模式中[AWS Glue Data Catalog 使用 Amazon Athena 設定對共用的跨帳戶存取。](#)

## Amazon Athena for Apache Spark 的服務配額

服務配額 (也稱為限制) 是您的 AWS 帳戶 可使用的服務資源或操作數目最大值。如需有關您可以與 Amazon Athena for Spark 搭配使用的其他 AWS 服務的服務配額的詳細資訊，請參閱 Amazon Web Services 一般參考 中的 [AWS 服務配額](#)。

### Note

新的 AWS 帳戶 可能具有較低的初始配額，而配額可能會隨著時間增加。Amazon Athena for Apache Spark 會監控每個 AWS 區域 內的帳戶使用情況，再根據您的用量自動增加配額。如果您的要求超過規定的限制，請聯絡客戶支援。

下表列有 Amazon Athena for Apache Spark 的服務配額。

名稱	預設	可調整	描述
Apache Spark DPU 並行	160	否	您在目前 AWS 區域 的單一帳戶中可同時使用 Apache Spark 計算的資料處理單位 (DPU) 數目上限。DPU 是相對的處理能力，包含 4 個 vCPU 的運算容量和 16 GB 的記憶體。
Apache Spark session DPU 並行	60	否	在工作階段中，您可以同時使用 Apache Spark 計算的 DPU 數目上限。

# Athena 筆記本 API

下列清單包含 Athena 筆記本 API 動作的參考連結。如需資料結構和其他 Athena API 動作，請參閱 [Amazon Athena API 參考](#)。

- [CreateNotebook](#)
- [CreatePresignedNotebookUrl](#)
- [DeleteNotebook](#)
- [ExportNotebook](#)
- [GetCalculationExecution](#)
- [GetCalculationExecutionCode](#)
- [GetCalculationExecutionStatus](#)
- [GetNotebookMetadata](#)
- [GetSession](#)
- [GetSessionStatus](#)
- [ImportNotebook](#)
- [ListApplicationDPUSizes](#)
- [ListCalculationExecutions](#)
- [ListExecutors](#)
- [ListNotebookMetadata](#)
- [ListNotebookSessions](#)
- [ListSessions](#)
- [StartCalculationExecution](#)
- [StartSession](#)
- [StopCalculationExecution](#)
- [TerminateSession](#)
- [UpdateNotebook](#)
- [UpdateNotebookMetadata](#)

## Athena for Spark 中的已知問題

本頁說明了 Athena for Apache Spark 中的一些已知問題。

## 建立資料表時的非法引數例外狀況

雖然 Spark 不允許使用空位置屬性來建立資料庫，但中的資料庫 AWS Glue 可以有空 LOCATION 屬性，如果它們是在 Spark 之外建立的。

如果您建立資料表並指定具有空白 LOCATION 欄位的 AWS Glue 資料庫，則可能會發生如下的例外狀況 `IllegalArgumentException`：無法從空字串建立路徑。

例如，如果 AWS Glue 中的預設資料庫包含空 LOCATION 欄位，則下列命令會擲出例外狀況：

```
spark.sql("create table testTable (firstName STRING)")
```

建議的解決方案 A — 用 AWS Glue 來將位置新增至您正在使用的資料庫。

若要將位置新增至資 AWS Glue 料庫

1. 請登入 AWS Management Console 並開啟 AWS Glue 主控台，網址為 <https://console.aws.amazon.com/glue/>。
2. 在導覽窗格中，選擇 Databases (資料庫)。
3. 在資料庫清單中，選擇您要編輯的資料庫。
4. 在資料庫的詳細資訊頁面上，選擇 Edit (編輯)。
5. 在 Update a database (更新資料庫) 頁面上，針對 Location (位置)，輸入 Amazon S3 位置。
6. 選擇 Update Database (更新資料庫)。

Suggested solution B (建議的解決方案 B) – 使用在 Amazon S3 中具有現有有效位置的不同 AWS Glue 資料庫。例如，如果您有名為 `dbWithLocation` 的資料庫，請使用命令 `spark.sql("use dbWithLocation")` 切換至該資料庫。

Suggested solution C (建議的解決方案 C) – 當您使用 Spark SQL 建立資料表時，請指定 `location` 的值，如下列範例所示。

```
spark.sql("create table testTable (firstName STRING)
          location 's3://DOC-EXAMPLE-BUCKET/').
```

Suggested solution D (建議的解決方案 D) – 如果您在建立資料表時指定了位置，但仍然發生該問題，則請確定您提供的 Amazon S3 路徑具有尾隨正斜線。例如，下列命令會擲出非法引數例外狀況：

```
spark.sql("create table testTable (firstName STRING)
          location 's3://DOC-EXAMPLE-BUCKET'")
```

若要更正此問題，請在位置新增尾隨斜線 (例如，'s3:// DOC-EXAMPLE-BUCKET/')。

## 在工作群組位置中建立的資料庫

如果您使用類似 `spark.sql('create database db')` 的命令建立資料庫，但未指定資料庫的位置，則 Athena 會在您的工作群組位置中建立子目錄，並將該位置用於新建立的資料庫。

## 預設資料庫中 Hive 管理資 AWS Glue 料表的問題

如果中的預設資料庫 Location 屬性 AWS Glue 非空，並在 Amazon S3 中指定了有效位置，而且您使用 Athena for Spark 在 AWS Glue 預設資料庫中建立 Hive 受管資料表，則資料會寫入 Athena Spark 工作群組中指定的 Amazon S3 位置，而不是資料 AWS Glue 庫指定的位置。

因為 Apache Hive 如何處理其預設資料庫，就會發生這個問題。Apache Hive 在 Hive 倉庫根位置中建立資料表，該位置可能與實際的預設資料庫位置不同。

當您使用 Athena for Spark 在中的預設資料庫下建立 Hive 管理的資料表時 AWS Glue，資料 AWS Glue 表中繼資料可以指向兩個不同的位置。當您嘗試 INSERT 或 DROP TABLE 作業時，這可能會導致非預期的行為。

重現問題的步驟如下：

1. 在 Athena for Spark 中，您可以使用下列其中一種方法來建立或儲存 Hive 管理的資料表：
  - 一個 SQL 陳述式 `CREATE TABLE $tableName`
  - `df.write.mode("overwrite").saveAsTable($tableName)` 這樣的 PySpark 命令不會在數據幀 API 中指定 path 選項。

此時，AWS Glue 主控台可能會在 Amazon S3 中針對表格顯示不正確的位置。

2. 在 Athena for Spark 中，您可以使用 `DROP TABLE $table_name` 陳述式刪除您所建立的資料表。
3. 執行 `DROP TABLE` 陳述式之後，您會注意到 Amazon S3 中的基礎檔案仍然存在。

要解決此問題，請執行下列項目之一：

解決方案 A-當您創建 Hive 託管的表使用不同的 AWS Glue 數據庫。

解決方案 B – 在 AWS Glue 中指定預設資料庫的空白位置。然後，在預設資料庫中建立受管理的資料表。

## Athena for Spark 和 Athena SQL 之間的 CSV 和 JSON 檔案格式不相容

由於開放原始碼 Spark 的已知問題，當您在 Athena for Spark 中就 CSV 或 JSON 資料建立資料表時，該資料表可能無法從 Athena SQL 讀取，反之亦然。

例如，您可使用下列其中一種方式在 Athena for Spark 中建立資料表：

- 使用下列 USING csv 語法：

```
spark.sql('''CREATE EXTERNAL TABLE $tableName (  
  $colName1 $colType1,  
  $colName2 $colType2,  
  $colName3 $colType3)  
  USING csv  
  PARTITIONED BY ($colName1)  
  LOCATION $s3_location''')
```

- 使用以下 [DataFrame](#) API 語法：

```
df.write.format('csv').saveAsTable($table_name)
```

由於開放原始碼 Spark 的已知問題，Athena SQL 對結果資料表的查詢可能無法成功。

建議的解決方案 – 嘗試使用 Apache Hive 語法在 Athena for Spark 中建立資料表。如需詳細資訊，請參閱 Apache Spark 文件中的 [建立 HIVEFORMAT 資料表](#)。

## 對 Athena for Spark 進行疑難排解

請使用以下資訊，對在 Athena 上使用筆記本和工作階段時可能遇到的問題進行疑難排解。

### 主題

- [對已啟用 Spark 的工作群組進行疑難排解](#)
- [使用 Spark EXPLAIN 陳述式，以對 Spark SQL 進行疑難排解](#)
- [記錄 Athena 中的 Spark 應用程式事件](#)
- [使用 CloudTrail 對 Athena 筆記本 API 呼叫進行疑難排解](#)

- [克服 68k 程式碼區塊大小限制](#)
- [工作階段疑難排解](#)
- [資料表疑難排解](#)
- [取得支援](#)

## 對已啟用 Spark 的工作群組進行疑難排解

使用以下資訊，以對 Athena 中已啟用 Spark 的工作群組進行疑難排解。

### 使用現有 IAM 角色時，工作階段會停止回應

如果您沒有為已啟用 Spark 的工作群組建立新的 `AWSAthenaSparkExecutionRole`，而是更新或選擇現有 IAM 角色，則您的工作階段可能會停止回應。在此情況下，您可能需要將下列信任和許可政策新增至已啟用 Spark 的工作群組執行角色。

新增下列信任政策範例。此政策包含執行角色的混淆代理人檢查。將 `111122223333aws-region`、和 `workgroup-name` 的值取代為您正在使用的 AWS 帳戶 ID AWS 區域、和工作群組。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "athena.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:athena:aws-  
region:111122223333:workgroup/workgroup-name"
        }
      }
    }
  ]
}
```

新增許可政策，如下列已啟用筆記本之工作群組的預設政策。修改預留位置 Amazon S3 位置 and AWS 帳戶 ID，使其對應於您正在使用的位置。將 `DOC-EXAMPLE-BUCKET`、`aws-region`、`111122223333` 和 `workgroup-name` 的值取代為您正在使用的 Amazon S3 儲存貯體、AWS 區域、AWS 帳戶 ID 和工作群組。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "athena:CreatePresignedNotebookUrl",
        "athena:TerminateSession",
        "athena:GetSession",
        "athena:GetSessionStatus",
        "athena:ListSessions",
        "athena:StartCalculationExecution",
        "athena:GetCalculationExecutionCode",
        "athena:StopCalculationExecution",
        "athena:ListCalculationExecutions",
        "athena:GetCalculationExecution",
        "athena:GetCalculationExecutionStatus",
        "athena:ListExecutors",
        "athena:ExportNotebook",
        "athena:UpdateNotebook"
      ],
      "Resource": "arn:aws:athena:aws-region:111122223333:workgroup/workgroup-name"
    }
  ]
}
```

```

    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:CreateLogGroup",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:aws-region:111122223333:log-group:/aws-athena:*",
        "arn:aws:logs:aws-region:111122223333:log-group:/aws-athena*:log-
stream:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "logs:DescribeLogGroups",
      "Resource": "arn:aws:logs:aws-region:111122223333:log-group:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "AmazonAthenaForApacheSpark"
        }
      }
    }
  ]
}

```

## 使用 Spark EXPLAIN 陳述式，以對 Spark SQL 進行疑難排解

您可以使用含有 Spark SQL 的 Spark EXPLAIN 陳述式，以對您的 Spark 程式碼進行疑難排解。下列程式碼和輸出範例說明了這種用法。

### Example – Spark SELECT 陳述式

```
spark.sql("select * from select_taxi_table").explain(True)
```

## 輸出

```

Calculation started (calculation_id=20c1ebd0-1ccf-ef14-db35-7c1844876a7e) in
(session=24c1ebcb-57a8-861e-1023-736f5ae55386).
Checking calculation status...

Calculation completed.
== Parsed Logical Plan ==
'Project [*]
+- 'UnresolvedRelation [select_taxi_table], [], false

== Analyzed Logical Plan ==
VendorID: bigint, passenger_count: bigint, count: bigint
Project [VendorID#202L, passenger_count#203L, count#204L]
+- SubqueryAlias spark_catalog.spark_demo_database.select_taxi_table
  +- Relation spark_demo_database.select_taxi_table[VendorID#202L,
    passenger_count#203L,count#204L] csv

== Optimized Logical Plan ==
Relation spark_demo_database.select_taxi_table[VendorID#202L,
passenger_count#203L,count#204L] csv

== Physical Plan ==
FileScan csv spark_demo_database.select_taxi_table[VendorID#202L,
passenger_count#203L,count#204L]
Batched: false, DataFilters: [], Format: CSV,
Location: InMemoryFileIndex(1 paths)
[s3://DOC-EXAMPLE-BUCKET/select_taxi],
PartitionFilters: [], PushedFilters: [],
ReadSchema: struct<VendorID:bigint,passenger_count:bigint,count:bigint>

```

## Example – Spark 資料框架

以下範例顯示如何搭配使用 EXPLAIN 與 Spark 資料框架。

```

taxi1_df=taxi_df.groupBy("VendorID", "passenger_count").count()
taxi1_df.explain("extended")

```

## 輸出

```

Calculation started (calculation_id=d2c1ebd1-f9f0-db25-8477-3effc001b309) in
(session=24c1ebcb-57a8-861e-1023-736f5ae55386).
Checking calculation status...

```

```

Calculation completed.
== Parsed Logical Plan ==
'Aggregate ['VendorID, 'passenger_count],
['VendorID, 'passenger_count, count(1) AS count#321L]
+- Relation [VendorID#49L,tpep_pickup_datetime#50,tpep_dropoff_datetime#51,
passenger_count#52L,trip_distance#53,RatecodeID#54L,store_and_fwd_flag#55,
PULocationID#56L,DOLocationID#57L,payment_type#58L,fare_amount#59,
extra#60,mta_tax#61,tip_amount#62,tolls_amount#63,improvement_surcharge#64,
total_amount#65,congestion_surcharge#66,airport_fee#67] parquet

== Analyzed Logical Plan ==
VendorID: bigint, passenger_count: bigint, count: bigint
Aggregate [VendorID#49L, passenger_count#52L],
[VendorID#49L, passenger_count#52L, count(1) AS count#321L]
+- Relation [VendorID#49L,tpep_pickup_datetime#50,tpep_dropoff_datetime#51,
passenger_count#52L,trip_distance#53,RatecodeID#54L,store_and_fwd_flag#55,
PULocationID#56L,DOLocationID#57L,payment_type#58L,fare_amount#59,extra#60,
mta_tax#61,tip_amount#62,tolls_amount#63,improvement_surcharge#64,
total_amount#65,congestion_surcharge#66,airport_fee#67] parquet

== Optimized Logical Plan ==
Aggregate [VendorID#49L, passenger_count#52L],
[VendorID#49L, passenger_count#52L, count(1) AS count#321L]
+- Project [VendorID#49L, passenger_count#52L]
  +- Relation [VendorID#49L,tpep_pickup_datetime#50,tpep_dropoff_datetime#51,
passenger_count#52L,trip_distance#53,RatecodeID#54L,store_and_fwd_flag#55,
PULocationID#56L,DOLocationID#57L,payment_type#58L,fare_amount#59,extra#60,
mta_tax#61,tip_amount#62,tolls_amount#63,improvement_surcharge#64,
total_amount#65,congestion_surcharge#66,airport_fee#67] parquet

== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- HashAggregate(keys=[VendorID#49L, passenger_count#52L], functions=[count(1)],
output=[VendorID#49L, passenger_count#52L, count#321L])
  +- Exchange hashpartitioning(VendorID#49L, passenger_count#52L, 1000),
    ENSURE_REQUIREMENTS, [id=#531]
    +- HashAggregate(keys=[VendorID#49L, passenger_count#52L],
      functions=[partial_count(1)], output=[VendorID#49L,
        passenger_count#52L, count#326L])
      +- FileScan parquet [VendorID#49L,passenger_count#52L] Batched: true,
        DataFilters: [], Format: Parquet,
        Location: InMemoryFileIndex(1 paths)[s3://DOC-EXAMPLE-BUCKET/
          notebooks/yellow_tripdata_2016-01.parquet], PartitionFilters: [],

```

```
PushedFilters: [],  
ReadSchema: struct<VendorID:bigint,passenger_count:bigint>
```

## 記錄 Athena 中的 Spark 應用程式事件

Athena 筆記本編輯器允許標準的 Jupyter、Spark 和 Python 記錄。您可以使 `df.show()` 用顯示 PySpark DataFrame 內容或用 `print("Output")` 於在儲存格輸出中顯示值。計算的 `stdout`、`stderr` 和 `results` 輸出會寫入 Amazon S3 中的查詢結果儲存貯體位置。

### 將 Spark 應用程式事件記錄到 Amazon CloudWatch

您的 Athena 會話也可以在您使用的帳戶 CloudWatch 中將日誌寫入 [Amazon](#)。

#### 了解日誌串流和日誌群組

CloudWatch 將記錄活動組織到記錄串流和記錄群組中。

**防護記錄串流** — CloudWatch 記錄資料流是共用相同來源的一系列記錄事件。記錄檔中的每個個別記錄來源都會組成個別的 CloudWatch 記錄資料流。

**防護記錄群組** — 在 CloudWatch 記錄檔中，記錄群組是共用相同保留、監視和存取控制設定的一組記錄資料流。

可以屬於一個日誌群組的日誌串流數量並沒有限制。

在 Athena 中，當您第一次啟動筆記本工作階段時，Athena 會在中建立一個使用 CloudWatch 已啟用 Spark 之工作群組名稱的記錄群組，如下列範例所示。

```
/aws-athena/workgroup-name
```

此日誌群組會為工作階段中的每個執行器接收日誌串流，其中該日誌串流會產生至少一個日誌事件。執行器是筆記本工作階段可向 Athena 請求的最小運算單位。在中 CloudWatch，記錄資料流的名稱以工作階段 ID 和執行程式 ID 開頭。

如需有關 [CloudWatch 日誌群組和日誌串流的詳細資訊](#)，請參閱 [Amazon CloudWatch Logs 使用者指南](#) 中的 [使用日誌群組和日誌串流](#)。

在 Athena for Spark 中，使用標準記錄器物件

在 Athena for Spark 工作階段中，您可以使用下列兩個全域標準記錄器物件將日誌寫入 Amazon CloudWatch：

- 使用者記錄器 — 僅將記錄檔傳送給。CloudWatch 當您想要將 Spark 應用程式直接記錄到的資訊時，請使用此物件 CloudWatch，如下列範例所示。

```
athena_user_logger.info("CloudWatch log line.")
```

此範例會寫入記錄事件，CloudWatch 如下所示：

```
AthenaForApacheSpark: 2022-01-01 12:00:00,000 INFO builtins: CloudWatch log line.
```

- athena\_shared\_logger — 將相同的防護記錄檔傳送至或傳送至以供支援之用。CloudWatch AWS 您可以使用此物件與 AWS 服務團隊共用記錄檔以進行疑難排解，如下列範例所示。

```
athena_shared_logger.info("Customer debug line.")  
var = [...some variable holding customer data...]  
athena_shared_logger.info(var)
```

此範例會將 debug 行和 var 變數值 CloudWatch 記錄到 Logs，並將每行的副本傳送至 AWS Support。

#### Note

為了您的隱私，您的計算代碼和結果不會與之共享 AWS。確定您的 athena\_shared\_logger 呼叫僅寫入您想要 AWS Support 顯示的資訊。

提供的記錄器會透過 [Apache Log4j](#) 寫入事件，並繼承此介面的日誌記錄層級。可能的日誌層級值為 DEBUG、ERROR、FATAL、INFO 和 WARN 或 WARNING。您可以使用記錄器上相應的具名函數來產生這些值。

#### Note

請勿重新繫結名稱 athena\_user\_logger 或 athena\_shared\_logger。這樣做會使日誌對象無法在會話 CloudWatch 的其餘部分寫入。

範例：將筆記本事件記錄到 CloudWatch

下列程序說明如何將 Athena 筆記本事件記錄到 Amazon CloudWatch 日誌。

## 將 Athena 筆記本事件記錄到 Amazon CloudWatch 日誌

1. 請遵循 [開始在 Amazon Athena 上使用 Apache Spark](#)，以使用唯一名稱在 Athena 中建立已啟用 Spark 的工作群組。本教學課程使用工作群組名稱 `athena-spark-example`。
2. 請遵循 [建立您自己的筆記本](#) 中的步驟，以建立筆記本並啟動新的工作階段。
3. 在 Athena 筆記本編輯器的新筆記本儲存格中，輸入下列命令：

```
athena_user_logger.info("Hello world.")
```

4. 執行儲存格。
5. 執行下列其中一項動作，擷取目前工作階段 ID：
  - 檢視儲存格輸出 (例如，`... session=72c24e73-2c24-8b22-14bd-443bdcd72de4`)。
  - 在新的儲存格中，執行[魔術](#)命令 `%session_id`。
6. 儲存工作階段 ID。
7. 使用與 AWS 帳戶執行筆記本工作階段相同的方式，請在 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
8. 在 CloudWatch 主控台瀏覽窗格中，選擇 [記錄群組]。
9. 在日誌群組清單中，選擇具有已啟用 Spark 之 Athena 工作群組名稱的日誌群組，如下列範例所示。

```
/aws-athena/athena-spark-example
```

Log streams (日誌串流) 區段包含工作群組的一或多個日誌串流連結清單。每個日誌串流名稱都包含工作階段 ID、執行器 ID 和由正斜線字元分隔的唯一 UUID。

例如，如果工作階段 ID 為 `5ac22d11-9fd8-ded7-6542-0412133d3177` 且執行器 ID 為 `f8c22d11-9fd8-ab13-8aba-c4100bfba7e2`，日誌串流的名稱會類似下列範例。

```
5ac22d11-9fd8-ded7-6542-0412133d3177/f8c22d11-9fd8-ab13-8aba-c4100bfba7e2/f012d7cb-cefd-40b1-90b9-67358f003d0b
```

10. 選擇工作階段的日誌串流連結。
11. 在 Log events (日誌事件) 頁面上，檢視 Message (訊息) 資料欄。

您執行之儲存格的日誌事件類似下列：

```
AthenaForApacheSpark: 2022-01-01 12:00:00,000 INFO builtins: Hello world.
```

12. 返回 Athena 筆記本編輯器。
13. 在新的儲存格中，輸入下列程式碼。程式碼會將變數記錄到 CloudWatch：

```
x = 6  
athena_user_logger.warn(x)
```

14. 執行儲存格。
15. 返回相同記錄資料流的 CloudWatch 主控台 [記錄事件] 頁面。
16. 日誌串流現在包含日誌事件項目，其中包含類似下列訊息：

```
AthenaForApacheSpark: 2022-01-01 12:00:00,000 WARN builtins: 6
```

## 使用 CloudTrail 對 Athena 筆記本 API 呼叫進行疑難排解

若要對筆記本 API 呼叫進行疑難排解，您可以檢查 Athena CloudTrail 日誌，以調查異常狀況或探索使用者發起的動作。如需有關搭配使用 CloudTrail 與 Athena 的詳細資訊，請參閱 [記錄 Amazon Athena API 呼叫 AWS CloudTrail](#)。

以下範例會示範 Athena 筆記本 API 的 CloudTrail 日誌項目：

- [StartSession](#)
- [TerminateSession](#)
- [ImportNotebook](#)
- [UpdateNotebook](#)
- [StartCalculationExecution](#)

### StartSession

下列範例會顯示筆記本的 CloudTrail 日誌 [StartSession](#) 事件。

```
{  
  "eventVersion": "1.08",  
  "userIdentity": {  
    "type": "AssumedRole",
```

```
"principalId": "EXAMPLE_PRINCIPAL_ID:alias",
"arn": "arn:aws:sts::123456789012:assumed-role/Admin/alias",
"accountId": "123456789012",
"accessKeyId": "EXAMPLE_KEY_ID",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "creationDate": "2022-10-14T16:41:51Z",
    "mfaAuthenticated": "false"
  }
}
},
"eventTime": "2022-10-14T17:05:36Z",
"eventSource": "athena.amazonaws.com",
"eventName": "StartSession",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.10",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36",
"requestParameters": {
  "workGroup": "notebook-workgroup",
  "engineConfiguration": {
    "coordinatorDpuSize": 1,
    "maxConcurrentDpus": 20,
    "defaultExecutorDpuSize": 1,
    "additionalConfigs": {
      "NotebookId": "b8f5854b-1042-4b90-9d82-51d3c2fd5c04",
      "NotebookIframeParentUrl": "https://us-east-1.console.aws.amazon.com"
    }
  }
},
"notebookVersion": "KeplerJupyter-1.x",
"sessionIdleTimeoutInMinutes": 20,
"clientRequestToken": "d646ff46-32d2-42f0-94d1-d060ec3e5d78"
},
"responseElements": {
  "sessionId": "a2c1ebba-ad01-865f-ed2d-a142b7451f7e",
  "state": "CREATED"
}
```

```

},
"requestID": "d646ff46-32d2-42f0-94d1-d060ec3e5d78",
"eventID": "b58ce998-eb89-43e9-8d67-d3d8e30561c9",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
}

```

## TerminateSession

下列範例會顯示筆記本的 CloudTrail 日誌 [TerminateSession](#) 事件。

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:alias",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/alias",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-14T16:41:51Z",
        "mfaAuthenticated": "false"
      }
    }
  }
},

```

```

"eventTime": "2022-10-14T17:21:03Z",
"eventSource": "athena.amazonaws.com",
"eventName": "TerminateSession",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.11",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36",
"requestParameters": {
  "sessionId": "a2c1ebba-ad01-865f-ed2d-a142b7451f7e"
},
"responseElements": {
  "state": "TERMINATING"
},
"requestID": "438ea37e-b704-4cb3-9a76-391997cf42ee",
"eventID": "49026c5a-bf58-4cdb-86ca-978e711ad238",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
}

```

## ImportNotebook

下列範例會顯示筆記本的 CloudTrail 日誌 [ImportNotebook](#) 事件。為了安全起見，某些內容會隱藏。

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:alias",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/alias",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",

```

```
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
        "creationDate": "2022-10-14T16:41:51Z",
        "mfaAuthenticated": "false"
    }
}
},
"eventTime": "2022-10-14T17:08:54Z",
"eventSource": "athena.amazonaws.com",
"eventName": "ImportNotebook",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.12",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36",
"requestParameters": {
    "workGroup": "notebook-workgroup",
    "name": "example-notebook-name",
    "payload": "HIDDEN_FOR_SECURITY_REASONS",
    "type": "IPYNB",
    "contentMD5": "HIDDEN_FOR_SECURITY_REASONS"
},
"responseElements": {
    "notebookId": "05f6225d-bdcc-4935-bc25-a8e19434652d"
},
"requestID": "813e777f-6dac-41f4-82a7-e99b7b33f319",
"eventID": "4abec837-143b-4458-9c1f-fa9fb88ab69b",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
}
```

## UpdateNotebook

下列範例會顯示筆記本的 CloudTrail 日誌 [UpdateNotebook](#) 事件。為了安全起見，某些內容會隱藏。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:AthenaExecutor-9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "arn": "arn:aws:sts::123456789012:assumed-role/AWSAthenaSparkExecutionRole-om0yj71w5l/AthenaExecutor-9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/service-role/AWSAthenaSparkExecutionRole-om0yj71w5l",
        "accountId": "123456789012",
        "userName": "AWSAthenaSparkExecutionRole-om0yj71w5l"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-14T16:48:06Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-10-14T16:52:22Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "UpdateNotebook",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.13",
  "userAgent": "Boto3/1.24.84 Python/3.8.14 Linux/4.14.225-175.364.amzn2.aarch64 Botocore/1.27.84",
  "requestParameters": {
    "notebookId": "c87553ff-e740-44b5-884f-a70e575e08b9",
    "payload": "HIDDEN_FOR_SECURITY_REASONS",
    "type": "IPYNB",
    "contentMD5": "HIDDEN_FOR_SECURITY_REASONS",
    "sessionId": "9cc1ebb2-aac5-b1ca-8247-5d827bd8232f"
  },
}
```

```

"responseElements": null,
"requestID": "baaba1d2-f73d-4df1-a82b-71501e7374f1",
"eventID": "745cdd6f-645d-4250-8831-d0ffd2fe3847",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
}
}

```

## StartCalculationExecution

下列範例會顯示筆記本的 CloudTrail 日誌 [StartCalculationExecution](#) 事件。為了安全起見，某些內容會隱藏。

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:AthenaExecutor-9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "arn": "arn:aws:sts::123456789012:assumed-role/AWSAthenaSparkExecutionRole-om0yj71w5l/AthenaExecutor-9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/service-role/AWSAthenaSparkExecutionRole-om0yj71w5l",
        "accountId": "123456789012",
        "userName": "AWSAthenaSparkExecutionRole-om0yj71w5l"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-14T16:48:06Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}

```

```
    }
  },
  "eventTime": "2022-10-14T16:52:37Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "StartCalculationExecution",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.14",
  "userAgent": "Boto3/1.24.84 Python/3.8.14 Linux/4.14.225-175.364.amzn2.aarch64
BotoCore/1.27.84",
  "requestParameters": {
    "sessionId": "9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "description": "Calculation started via Jupyter notebook",
    "codeBlock": "HIDDEN_FOR_SECURITY_REASONS",
    "clientRequestToken": "0111cd63-4fd0-4ad8-a738-fd350115fc21"
  },
  "responseElements": {
    "calculationExecutionId": "82c1ebb4-bd08-e4c3-5631-a662fb2ff2c5",
    "state": "CREATING"
  },
  "requestID": "1a107461-3f1b-481e-b8a2-7fbd524e2373",
  "eventID": "b74dbd00-e839-4bd1-a1da-b75fbc70ab9a",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
  }
}
```

## 克服 68k 程式碼區塊大小限制

Athena for Spark 的已知計算程式碼區塊大小限制為 6.8 萬個字元。當您使用超過此限制的程式碼區塊執行計算時，您可能會收到下列錯誤訊息：

「程式碼區塊」的 '...' 無法滿足限制條件：成員的長度必須小於或等於 68000

下圖顯示了 Athena 主控台筆記本編輯器中的此錯誤。

```
In [2]: JE00cOV0sNEDP7PUCpHePE5vni6nZZtVvKREuSoIJt0Vrnw901acjRcnKUtZT1Xuw2vIumRRMnHxuEKsDFV0wT3PQcu5pU3sbQlIMDzatGO5M9sjKr4WV1N
JE00cOV0sNEDP7PUCpHePE5vni6nZZtVvKREuSoIJt0Vrnw901acjRcnKUtZT1Xuw2vIumRRMnHxuEKsDFV0wT3PQcu5pU3sbQlIMDzatGO5M9sjKr4WV1N
b8cy2HjUP08VpSc80tNVO825bDhaV4iu78JhrZRro6W9j1zDnitgKk6piR617jzQoG8uSW4fbigSKdChr2vlhW7XVRhsEWTT12Xu7PHxjEr1DE1H0Xv4M
gHrWz...
5UGIMHnMGUld2k2AstjHvpKICKtxcQgEMoK7hTDPGivfYZgai2YUXhmxwWof6flkEeVzpBBsUNCEDKrOo9rFkGbpJfAAKbpBbNxjpiVwIrmennQX9iQ7a
ZksYvu0150hdYwGEX2i6cLO' at 'codeBlock' failed to satisfy constraint: Member must have length less than or equal to 6
8000
```

當您使用 AWS CLI 執行具有大型程式碼區塊的計算時，可能會發生相同的錯誤，如下列範例所示。

```
aws athena start-calculation-execution \
  --session-id "{SESSION_ID}" \
  --description "{SESSION_DESCRIPTION}" \
  --code-block "{LARGE_CODE_BLOCK}"
```

該命令會提供以下錯誤訊息：

「程式碼區塊」的 `{LARGE_CODE_BLOCK}` 無法滿足限制條件：成員的長度必須小於或等於 68000

## 解決方法

若要解決此問題，請將具有您的查詢或計算程式碼的檔案上傳至 Amazon S3。然後，使用 boto3 讀取文件並執行 SQL 或程式碼。

下列範例假設，您已將具有 SQL 查詢或 Python 程式碼的檔案上傳至 Amazon S3。

### SQL 範例

下列範例程式碼會從 Amazon S3 儲存貯體讀取 `large_sql_query.sql` 檔案，然後執行檔案包含的大型查詢。

```
s3 = boto3.resource('s3')
def read_s3_content(bucket_name, key):
    response = s3.Object(bucket_name, key).get()
    return response['Body'].read()

# SQL
sql = read_s3_content('bucket_name', 'large_sql_query.sql')
df = spark.sql(sql)
```

### PySpark 範例

下列程式碼範例會從 Amazon S3 讀取 `large_py_spark.py` 檔案，然後執行檔案中的大型程式碼區塊。

```
s3 = boto3.resource('s3')

def read_s3_content(bucket_name, key):
    response = s3.Object(bucket_name, key).get()
    return response['Body'].read()

# PySpark
py_spark_code = read_s3_content('bucket_name', 'large_py_spark.py')
exec(py_spark_code)
```

## 工作階段疑難排解

使用本主題中的資訊來疑難排解工作階段問題。

### 工作階段處於狀況不良狀態

如果您收到錯誤訊息 `Session in unhealthy state` (工作階段處於狀況不良狀態)。請建立新的工作階段，終止現有的工作階段並建立新的工作階段。

### 無法建立與筆記本伺服器的連線

當您開啟筆記本時，可能會看到下列錯誤訊息：

```
A connection to the notebook server could not be established.
The notebook will continue trying to reconnect.
Check your network connection or notebook server configuration.
```

#### 原因

Athena 開啟筆記本時，Athena 會建立工作階段，並使用預先簽章的筆記本 URL 連線至筆記本。與筆記型電腦的連線使用 WSS ([WebSocket安全](#)) 通訊協定。

錯誤的發生原因如下：

- 本機防火牆 (例如，適用於全公司範圍的防火牆) 會封鎖 WSS 流量。
- 本機電腦上的代理或防毒軟體會封鎖 WSS 連線。

#### 解決方案

假設您在 `us-east-1` 區域中有一個 WSS 連線，如下所示：

```
wss://94c2bcdf-66f9-4d17-9da6-7e7338060183.analytics-gateway.us-east-1.amazonaws.com/
api/kernels/33c78c82-b8d2-4631-bd22-1565dc6ec152/channels?session_id=
7f96a3a048ab4917b6376895ea8d7535
```

若要解決錯誤，請使用以下其中一種策略。

- 使用萬用字元模式語法允許在 AWS 區域 和 443 AWS 帳戶之間的連接埠上列出 WSS 流量。

```
wss://*amazonaws.com
```

- 使用萬用字元模式語法，允許在您指定的一個 AWS 區域 和對面 443 AWS 帳戶 的連接埠上列 AWS 區域 出 WSS 流量。以下範例使用 us-east-1。

```
wss://*analytics-gateway.us-east-1.amazonaws.com
```

## 資料表疑難排解

### 建立資料表時無法建立路徑錯誤

錯誤訊息 `IllegalArgumentException::` 無法從空字串建立路徑。

原因：當您在 Athena 中使用 Apache Spark 在資料庫中建立資料表，且 AWS Glue 資料庫具有空 `LOCATION` 屬性時，可能會發生此錯誤。

建議的解決方案：如需詳細資訊和解決方案，則請參閱 [建立資料表時的非法引數例外狀況](#)

### AccessDeniedException 查詢 AWS Glue 表格時

錯誤信息：錯誤信息。 `AnalysisException: 無法驗證預設資料庫是否存在:是否存在。`

```
AccessDeniedException: ###:ARN: ss: sts::: aws-role/aws-account-id-####/-
##AWSAthenaSparkExecutionRole#####GetDatabase #####arn#aws###
aws-#####aws-account-id#####AthenaExecutor###; #####400; ##
###; ID ##### ID###GetDatabase AWSGlue AccessDeniedException
```

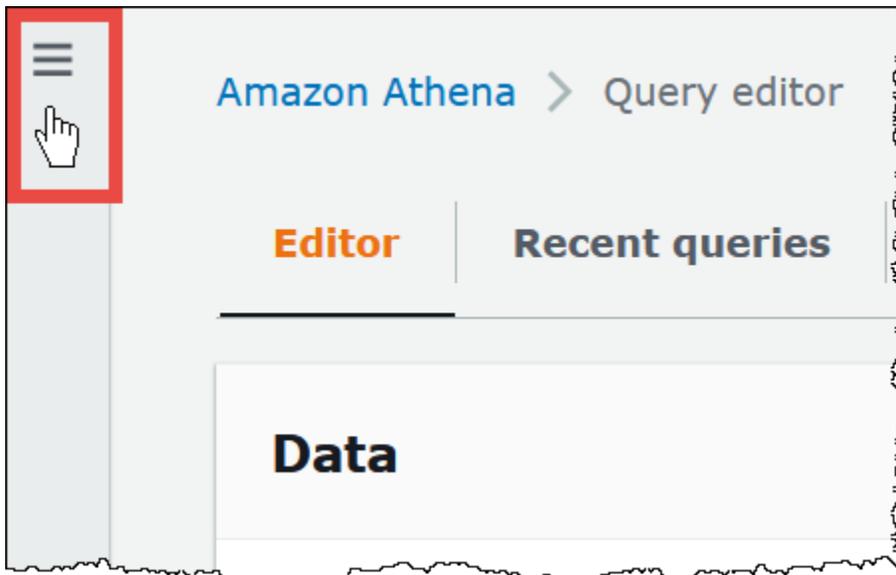
原因：啟用 Spark 的工作群組的執行角色缺少存取資源的權 AWS Glue 限。

建議的解決方案：若要解決此問題，請授予您的執行角色 AWS Glue 資源存取權，然後編輯 Amazon S3 儲存貯體政策以授與執行角色的存取權。

下列程序更詳細地說明了這些步驟。

## 若要授與您的執行角色 AWS Glue 資源存取權

1. 前往 <https://console.aws.amazon.com/athena/> 開啟 Athena 主控台。
2. 如果未顯示主控台的導覽窗格，請選擇左側的展開選單。



3. 在 Athena 主控台導覽窗格中，選擇 Workgroups (工作群組)。
4. 在 Workgroups (工作群組) 頁面上，請選擇您要檢視的工作群組連結。
5. 在工作群組的 Overview Details (概觀詳細資訊) 頁面上，選擇 Role ARN (角色 ARN) 連結。此連結會在 IAM 主控台中開啟 Spark 執行角色。
6. 在 Permissions policies (許可政策) 區段中，選擇連結角色政策名稱。
7. 選擇 Edit policy (編輯政策)，然後選擇 JSON。
8. 新增角色的 AWS Glue 存取權。通常，您可以為 `glue:GetDatabase` 和 `glue:GetTable` 動作新增許可。如需有關設定 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的 [新增和移除 IAM 身分許可](#)。
9. 選擇 Review policy (檢閱政策)，然後選擇 Save changes (儲存變更)。
10. 編輯 Amazon S3 儲存貯體政策，以授予執行角色存取權。請注意，您必須授予角色儲存貯體和儲存貯體中的物件存取權。如需相關步驟，請參閱《Amazon Simple Storage Service 使用者指南》中的 [使用 Amazon S3 主控台新增儲存貯體政策](#)。

## 取得支援

如需協助 AWS，請從中選擇「Support 援」、「支援中心」AWS Management Console。為增強您的體驗，請準備以下資訊：

- Athena 查詢 ID
- 工作階段 ID
- 計算 ID

# 版本備註

依發行日期描述 Amazon Athena 功能、改善項目和錯誤修正。

## 主題

- [Athena 2024 年發行公告](#)
- [2023 年 Athena 版本備註](#)
- [2022 年 Athena 版本備註](#)
- [2021 年 Athena 版本備註](#)
- [2020 年 Athena 版本備註](#)
- [2019 年 Athena 版本備註](#)
- [2018 年 Athena 版本備註](#)
- [2017 年 Athena 版本備註](#)

## Athena 2024 年發行公告

2024年6月26日

### 發佈於

佈建容量現已在南美洲 (聖保羅) 和歐洲 (西班牙) 區域正式推出。佈建的容量可讓您在完全受管的運算容量上執行 SQL 查詢，並提供工作負載管理功能，協助您排定優先順序、控制和調整最重要的互動式工作負載。您可以隨時新增容量以增加可同時執行的查詢數量、控制哪些工作負載使用該容量，以及在工作負載之間共用容量。

如需詳細資訊，請參閱 [管理查詢處理容量](#)。如需定價資訊，請造訪 [Amazon Athena 定價](#) 頁面。

2024年4月26日

### 發佈於二零四年四月二十六日

Athena 發布 JDBC 驅動程序版本 3.2.0。如需此驅動程式版本的詳細資訊，請參閱 [Amazon Athena JDBC 3.x 版本資訊](#)。若要下載 JDBC 3.x 驅動程式，請參閱 [JDBC 3.x 驅動程式下載](#)。

2024年4月24日

### 發佈於 4 月 24 日

Athena 發布下列修復和改善。

- 實木複合地板 — Athena 現在支援在 Parquet 中向後相容讀取，適用於未包含在清單或地圖群組中的未註解、重複的原始欄位。此變更可防止以無訊息方式傳回不正確的結果，並改善結構描述不相符的錯誤訊息。

有關更多信息，請參閱在 .com 上 GitHub 的 [Parquet 中 Support 向後兼容讀取未註釋的重複原始字](#) [段](#)。

- Iceberg OPTIMIZE — 解決在子句中使用非分割區索引鍵篩選器時造成資料遺失的 OPTIMIZE 查詢問題。WHERE 如需詳細資訊，請參閱 [OPTIMIZE](#)。

## 2024年4月16日

發佈於 4 月 16 日

使用全新的 Amazon Athena 聯合查詢傳遞功能，直接在基礎資料來源上執行整個查詢。聯合傳遞查詢可協助您充分利用原始資料來源的獨特功能、查詢語言和效能功能。例如，您可以使用 [Parti QL](#) 語言在 DynamoDB 上執行 Athena 查詢。當您想要執行彙總、聯結或叫用 Athena 無法使用的資料來源函數的 SELECT 查詢時，同盟傳遞查詢也很有用。使用直通查詢可減少 Athena 處理的資料量，並加快查詢時間。

如需詳細資訊，請參閱 [執行同盟傳遞查詢](#)。若要將您今天使用的連接器升級到最新版本，請參閱 [更新資料來源連接器](#)。

## 2024年4月10日

發佈於 2 月 4 日

Athena 發布了下列功能以及強化功能。

### 十二月一千驅動程式

適用於 Athena 的驅動程式版本。

解決的問題：

- Proxy 伺服器連線問題 — 在沒有根憑證的情況下使用 Proxy 伺服器時，連接器無法建立連線。

如需詳細資訊，並下載 ODBC 1.x 驅動程式、版本說明和文件，請參閱 [Athena ODBC 1.x 驅動程式](#)

## 驅動程式

JDBC 2.1.5 Athena 的驅動程序發布。

更新和增強功能：

- 更新了 AWS Java 開發套件以使用 1.12.687 版本。
- 更新了傑克遜庫使用 2.16.0 版本。
- 更新記錄庫以使用 1.3.14 版本。

如需詳細資訊，並下載 JDBC 2.x 驅動程式、版本說明和文件，請參閱 [Athena JDBC 2.x 驅動程式](#)

## 2024年4月8日

發佈於二零一四年四月零八日

Athena 宣布 ODBC 驅動程式版本 2.0.3.0。如需詳細資訊，請參閱 [2.0.3.0](#) 版本備註。若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## 2024年3月15日

發佈於

Amazon Athena 宣布在加拿大西部 (卡爾加里) 地區推出 Athena SQL。

如需每個項目中 AWS 服務 可用項目的完整清單 AWS 區域，請參閱 [按地區分類的AWS 服務](#)。

## 2024年2月15日

發佈於 2 月 15 日

Athena 發布 JDBC 驅動程序版本 3.1.0。

Amazon Athena JDBC 驅動程序 3.1.0 版增加了對 Microsoft 活動目錄聯合服務 (AD FS) 視窗集成身份驗證和基於表單的身份驗證的支持。3.1.0 版本還包括其他小改進和錯誤修復。

若要下載 JDBC v3 驅動程式，請參閱 [JDBC 3.x 驅動程式下載](#)。

## 2024 年 1 月 31 日

發佈於二零二一年一月三十一

Athena 發布了下列功能以及強化功能。

- 胡迪升級 — 您現在可以使用 Athena SQL 查詢胡迪 0.14.0 表。如需使用 Athena SQL 查詢 Hudi 資料表的相關資訊，請參閱[使用 Athena 查詢 Apache Hudi 資料集](#)。

## 2023 年 Athena 版本備註

2023 年 12 月 14 日

2023-12-14 發布者

Athena 發布下列修復和改善。

Athena 發布了 JDBC 驅動程式 2.1.3 版。驅動程式可解決下列問題：

- 改進了日誌記錄，以避免與 Spring Boot 和 Gradle 應用程式日誌記錄發生衝突。
- 當使用 `executeBatch()` JDBC 方法插入記錄時，驅動程式錯誤地只插入一則記錄。由於 Athena 不支援批次執行查詢，驅動程式現在會在您使用 `executeBatch()` 時報告錯誤。若要解決該限制，您可在迴圈中提交單一查詢。

若要下載新的 JDBC 驅動程式、版本備註和文件，請參閱 [Athena JDBC 2.x 驅動程式](#)。

2023 年 12 月 9 日

2023-12-09 發布

發布了適用於 Athena 的 ODBC 1.2.1.1000 驅動程式。

功能和增強功能：

- 更新了 RStudio 支援 – ODBC 驅動程式現在支援 macOS 上的 RStudio。
- 單一型錄和結構描述支援 – 連接器現在可以傳回單一型錄和結構描述。如需詳細資訊，請參閱可下載的安裝和設定指南。

解決的問題：

- 準備好的陳述式 – 執行含有參數陣列 (使用欄式結構描述) 的準備好的陳述式時，連接器會傳回不正確的查詢結果。
- 資料欄大小 – 選取 `$file_modified_time` 系統資料欄時，連接器傳回不正確的資料欄大小。

- SQL 準備 – 繫結與 SELECT 查詢中 SQLPrepare 相關的參數時，連接器會傳回錯誤。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱[Athena ODBC 1.x 驅動程式](#)。

## 2023 年 12 月 7 日

2023-12-07 發布

Athena 宣佈推出 ODBC 驅動程式 2.0.2.1 版。如需詳細資訊，請參閱 [2.0.2.1 版本備註](#)。若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## 2023 年 12 月 5 日

2023-12-05 發布

您現在可以建立使用 AWS IAM Identity Center 驗證模式的 Athena SQL 工作群組。這些工作群組支援 IAM Identity Center 的受信任身分傳播功能。受信任的身分傳播允許在 Amazon 雅典娜和亞馬遜 EMR 工作室等 AWS 分析服務中使用身分。

如需詳細資訊，請參閱 [使用啟用 IAM Identity Center 的 Athena 工作群組](#)。

## 2023 年 11 月 28 日

2023-11-28 發布

您現在可在 [Amazon S3 Express 單區域儲存類別](#) 中查詢資料，以獲取快速查詢結果。S3 Express One Zone 是一種高效能的單一可用區域儲存類別，專門針對最常存取的資料和延遲敏感的應用程式，提供一致、延遲不到 10 毫秒的資料存取。若要開始使用，請使用 [AWS Glue Data Catalog](#) 將您的資料移至 S3 Express 單區域儲存並編目資料，以便在 Athena 中提供無縫的查詢體驗。

如需詳細資訊，請參閱 [查詢 S3 Express 單區域資料](#)。

## 2023 年 11 月 27 日

2023-11-27 發布

Athena 發布了下列功能以及強化功能。

- Glue 資料目錄檢視 — Glue 資料目錄檢視提供跨 AWS 服務的單一通用檢視，例如 Amazon Athena 和 Amazon Redshift。在 Glue Data Catalog 檢視中，存取許可由建立檢視的使用者定義，而不是查

詢檢視的使用者。這些檢視提供更大的存取控制權、協助確保完整的記錄、提供增強的安全性，以及防止存取基礎資料表。

如需詳細資訊，請參閱 [使用 AWS Glue Data Catalog 視圖](#)。

- CloudTrail 湖泊支援 — 您現在可以使用 Amazon Athena 分析 [AWS CloudTrail 湖泊](#) 中的資料。AWS CloudTrail Lake 是受管理的資料湖，您 CloudTrail 可以用來彙總、不變化的儲存和分析活動記錄，以進行稽核、安全性和作業調查。若要從 Athena 查詢 CloudTrail Lake 活動記錄，您不需要移動資料或建立個別的資料處理管道。無需 ETL 操作。

若要開始使用，請在 CloudTrail Lake 中啟用資料聯合。當您與之共用 CloudTrail Lake 事件資料存放區中繼資料時 AWS Glue Data Catalog，CloudTrail 會建立必要的 AWS Glue Data Catalog 資源並向其註冊資料 AWS Lake Formation。在 Lake Formation 中，您可以指定可使用 Athena 查詢事件資料儲存的使用者和角色。

如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的 [啟用 Lake 查詢聯合](#)。

## 2023 年 11 月 17 日

2023-11-17 發布

Athena 發布了下列功能以及強化功能。

### 功能

- 以成本為基礎的最佳化工具 — Athena 宣布使用來自的統計資料正式推出成本 AWS Glue 若要最佳化 Athena SQL 中的查詢，您可以請求 Athena 為 AWS Glue 中的資料表收集資料表或資料欄層級的統計資訊。如果查詢中的所有資料表都有統計資訊，Athena 會使用這些統計資訊來檢查替代的執行計畫，並選取可能速度最快的執行計畫。

如需詳細資訊，請參閱 [使用成本型最佳化工具](#)。

- Amazon EMR Studio 整合 – 您現在可以在 Amazon EMR Studio 中直接使用 Athena，而無需使用 Athena 主控台。您可以使用 Amazon EMR 中的 Athena 整合進行下列任務：
  - 進行 Athena SQL 查詢
  - 檢視查詢結果
  - 檢視查詢歷史記錄
  - 檢視儲存的查詢
  - 進行參數化查詢

- 檢視資料目錄的資料庫、資料表和檢視

如需詳細資訊，請參閱 [AWS 服務與 Athena 的集成](#) 主題中的 [Amazon EMR Studio](#)。

- 巢狀存取控制 – Athena 宣布支援針對巢狀資料的 Lake Formation 存取控制。在 Lake Formation 中，您可以在具有 struct 資料類型的巢狀資料欄上定義和套用資料篩選條件。您可以使用資料篩選來限制使用者對巢狀資料欄子結構的存取。如需有關如何為巢狀資料建立資料篩選條件的資訊，請參閱《AWS Lake Formation 開發人員指南》中的 [建立資料篩選條件](#)。
- 佈建的容量使用量指標 — Athena 宣布容量保留的新 CloudWatch 指標。您可以使用新指標來追蹤已佈建的 DPU 數目以及查詢所使用的 DPU 數目。查詢完成後，您也可以檢視查詢所使用的 DPU 數目。

如需詳細資訊，請參閱 [使用 CloudWatch 指標監控 Athena 查詢](#)。

## 改善項目

- 錯誤訊息變更 – Insufficient Lake Formation permissions 錯誤訊息現在會讀取 Table not found 或 Schema not found。此變更是為了防止惡意行為者從錯誤訊息推斷出資料表或資料庫資源的存在。

## 2023 年 11 月 16 日

2023-11-16 發布

Athena 發布了新的 JDBC 驅動程式，可改善從相容 SQL 開發和商業智慧應用程式連線、查詢和視覺化資料的體驗。新驅動程式很容易升級。該驅動程式可以直接從 Amazon S3 讀取查詢結果，讓您更快取得查詢結果。

如需詳細資訊，請參閱 [Athena JDBC 3.x 驅動程式](#)。

## 2023 年 10 月 31 日

2023-10-31 發布

Amazon Athena 宣布佈建容量的保留時間為 1 小時。從今天開始，您可以保留佈建的容量並在一小時後釋放。這項變更可讓您更輕鬆地針對需求隨時間變化的工作負載，實現成本最佳化。

佈建的容量是 Athena 的一項功能，可提供工作負載管理功能，協助您優先處理、控制和擴展最重要的互動式工作負載。您可以隨時新增容量以增加可同時執行的查詢數量、控制哪些工作負載使用該容量，以及在工作負載之間共用容量。

如需詳細資訊，請參閱 [管理查詢處理容量](#)。如需定價資訊，請造訪 [Amazon Athena 定價](#) 頁面。

## 2023 年 10 月 25 日

2023-10-26 發布

Athena 發布了下列修復和改善。

jackson-core 套件 – 數值大於 1000 個字元的 JSON 文字現在會失敗。此項修復解決了安全性問題 [sonatype-2022-6438](#)。

## 2023 年 10 月 17 日

2023-10-17 發布

Athena 宣布了 ODBC 驅動程式 2.0.2.0 版。如需詳細資訊，請參閱 [2.0.2.0](#) 版本備註。若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## 2023 年 9 月 26 日

2023-09-26 發布

Athena 發布了下列功能以及改善項目。

- 適用於 Delta Lake 資料表的 Lake Formation 讀取支援。如需有關將 Delta Lake 資料表與 Athena 配合使用的詳細資訊，請參閱 [查詢 Linux Foundation Delta Lake 資料表](#)。

## 2023 年 8 月 23 日

2023-08-23 發布

Amazon Athena 宣佈在以色列 (特拉維夫) 區域推出 Athena SQL。

如需每個項目中 AWS 服務 可用項目的完整清單 AWS 區域，請參閱 [按地區分類的AWS 服務](#)。

## 2023 年 8 月 10 日

2023-08-10 發布

Athena 發布下列修復和改善。

## ODBC 驅動程式 2.0.1.1 版

Athena 宣佈了 ODBC 驅動程式 2.0.1.1 版。如需詳細資訊，請參閱 [2.0.1.1 版本備註](#)。若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

## JDBC 驅動程式 2.1.1 版

Athena 發布了 JDBC 驅動程式 2.1.1 版。驅動程式可解決下列問題：

- 使用包含規則表達式的陳述式建立資料表時發生的錯誤。
- 造成 ApplicationName 連線參數套用不正確的問題。

若要下載新的 JDBC 驅動程式、版本備註和文件，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。

## 2023 年 7 月 31 日

2023-07-31 發布

Amazon Athena 宣佈在其他 AWS 區域推出 Athena SQL。

此版本擴展了 Athena SQL 的可用性，包括亞太區域 (海德拉巴)、亞太區域 (墨爾本)、歐洲 (西班牙) 和歐洲 (蘇黎世)。

如需每個項目中 AWS 服務 可用項目的完整清單 AWS 區域，請參閱 [按地區分類的 AWS 服務](#)。

## 2023 年 7 月 27 日

2023-07-27 發布

Athena 發布谷歌 BigQuery 連接器版本 2023.30.1。此版本的連接器可減少查詢執行時間，並新增針對 BigQuery 私有端點進行查詢的支援。

如需 Google BigQuery 連接器的相關資訊，請參閱 [Amazon Athena 谷歌 BigQuery 連](#)。如需有關更新現有資料來源連接器的資訊，請參閱 [更新資料來源連接器](#)。

## 2023 年 7 月 24 日

2023-07-24 發布

Athena 發布下列修復和改善。

- 含聯集的查詢 – 改善含聯集的特定查詢的效能。
- 聯結類型比較 – 修正了包含兩種不同類型之間比較的 JOIN 陳述式的潛在查詢失敗。
- 巢狀資料欄上的子查詢 – 修正了當子查詢與巢狀資料欄相關聯時，與查詢失敗相關的問題。
- Iceberg 檢視 – 修正了 Apache Iceberg 檢視中時間戳記欄精度的相容性問題。現在，無論資料欄是在 Athena 引擎版本 2 還是 Athena 引擎版本 3 上建立的，具有時間戳記欄的 Iceberg 檢視皆可讀取。

## 2023 年 7 月 20 日

2023-07-20 發布

Athena 發布了 JDBC 驅動程式 2.1.0 版。該驅動程式包含新的增強功能並解決了問題。

### 增強功能

下列 [Jackson](#) JSON 剖析器程式庫已升級：

- jackson-annotations 2.15.2 (先前為 2.14.0)
- jackson-core 2.15.2 (先前為 2.14.0)
- jackson-databind 2.15.2 (先前為 2.14.0)

### 已解決問題

- 修正使用 [sql2o](#) 程式庫時傳遞陣列參數的問題。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

## 2023 年 7 月 13 日

2023-09-19 發布

Athena 發布了下列功能以及強化功能。

- EXPLAIN ANALYZE – 新增了對佇列、分析、規劃和執行期的支援至 EXPLAIN ANALYZE 輸出中。
- EXPLAIN – 當查詢包含彙總時，EXPLAIN 輸出現在會顯示統計資訊。
- 實木複合地板 SerDe-添加了該parquet.ignore.statistics屬性，以便在讀取 Parquet 數據時忽略處理統計信息。如需相關資訊，請參閱[忽略 Parquet 統計資料](#)。

如需 EXPLAIN 和 EXPLAIN ANALYZE 的更多相關資訊，請參閱在 [Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)。如需有關鑲木地板蜂巢的更多資訊 SerDe，請參閱 [木地板 SerDe](#)。

## 2023 年 7 月 3 日

2023-07-25 發布

從 2023 年 7 月 3 日起，Athena 開始編輯記錄中 CloudTrail 的查詢字串。查詢字串現在的值為 `***OMITTED***`。進行此變更是為了防止意外揭露可能包含敏感資訊的資料表名稱或篩選條件值。如果您先前依賴 CloudTrail 記錄檔來存取完整的查詢字串，建議您使用 `Athena::GetQueryExecution` API 並 `responseElements.queryExecutionId` 從記錄傳 CloudTrail 入的值。如需詳細資訊，請參閱 Amazon Athena API 參考中的 [GetQueryExecution](#) 動作。

## 2023 年 6 月 30 日

2023-06-30 發布

Athena 查詢編輯器現在支援預先輸入的程式碼建議，以獲得更快的查詢編寫體驗。您現在可以使用下列功能來編寫 SQL 查詢，提高精度和效率：

- 當您輸入時，將會即時顯示關鍵字、本機變數、程式碼片段和目錄項目。
- 當您輸入資料庫名稱或資料表名稱，並在其後加上一個點時，編輯器會輕鬆顯示可供選擇的資料表和資料欄清單。
- 將滑鼠游標移至程式碼片段建議上時，該概要會顯示程式碼片段語法和用法的簡要概觀。
- 為提高程式碼可讀性，關鍵字及其醒目提示規則也已更新，符合 Trino 和 Hive 的最新語法。

此功能預設為啟用。您可以在程式碼編輯器偏好設定中啟用或停用功能。

若要在 Athena 查詢編輯器中嘗試預先輸入的程式碼建議，請造訪 Athena 主控台：<https://console.aws.amazon.com/athena/>。

## 2023 年 6 月 29 日

2023-06-29 發布

- Athena 宣佈了 ODBC 驅動程式 2.0.1.0 版。如需詳細資訊，請參閱 [2.0.1.0](#) 版本備註。若要下載新的 ODBC v2 驅動程式，請參閱 [ODBC 2.x 驅動程式下載](#)。如需連線資訊，請參閱 [Amazon Athena ODBC 2.x](#)。

- Athena 及其[功能](#)現已在中東 (阿拉伯聯合大公國) 區域提供。如需每個項目中 AWS 服務 可用項目的完整清單 AWS 區域，請參閱[按地區分類的AWS 服務](#)。

## 2023 年 6 月 28 日

2023-06-28 發布

您現在可以使用 Amazon Athena 查詢從 S3 Glacier Flexible Retrieval (舊稱 Glacier) 和 S3 Glacier Deep Archive [Amazon S3 儲存類別](#)中還原的物件。您可基於每個資料表設定此功能。此功能僅支援 Athena 引擎版本 3 上的 Apache Hive 資料表。

如需詳細資訊，請參閱 [查詢還原的 Amazon S3 Glacier 物件](#)。

## 2023 年 6 月 12 日

2023-06-12 發布

Athena 發布下列修復和改善。

- Parquet Reader 時間戳記 – 新增了對 [Parquet Reader](#) 讀取時間戳記為 bigint (millis) 的支援 此更新提供與 Athena 引擎版本 2 中相同的支援。
- EXPLAIN ANALYZE – 新增實體輸入讀取時間至查詢統計資訊和 EXPLAIN ANALYZE 的輸出。如需 EXPLAIN ANALYZE 的資訊，請參閱「[在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)」。
- INSERT – 改善使用 INSERT 寫入的資料表的查詢效能。如需 INSERT 的資訊，請參閱「[INSERT INTO](#)」。
- Delta Lake 資料表 – 已修正 Delta Lake 資料表上 DROP TABLE 的問題，此問題可防止其在並行修改時完全被刪除。

## 2023 年 6 月 8 日

2023-06-08 發布

Amazon Athena for Apache Spark 發布了下列新版功能。

- 支援自訂 Java 程式庫和組態 – 您現在可以在 Athena 中為 Apache Spark 工作階段使用您自己的 Java 套件和自訂組態。使用 Spark 屬性，透過 Athena 主控台、或 Athena API 指定 .jar 檔案、套件 AWS CLI 或其他自訂組態。如需詳細資訊，請參閱 [新增 JAR 檔案和自訂 Spark 組態](#)。

- 支援 Apache Hudi、Apache Iceberg 和 Delta Lake 資料表 – Athena for Spark 現在支援 Apache Iceberg、Apache Hudi 和 Linux Foundation Delta Lake 開放原始碼資料湖儲存資料表格式。如需詳細資訊，請參閱 [使用 Amazon Athena for Apache Spark 中的非 Hive 資料表格式](#) 及在 Athena for Spark 中使用 [Apache Iceberg](#)、[Apache Hudi](#) 和 [Linux Foundation Delta Lake](#) 資料表的個別主題。
- Apache Spark 的加密支援 – 在 Athena for Spark 中，您現在可以對 Spark 節點之間傳輸中的資料以及 Spark 儲存在磁碟上的本機靜態資料啟用加密功能。若要啟用 Spark 加密，您可以使用 Athena 主控台 AWS CLI、或 Athena API。如需詳細資訊，請參閱 [啟用 Apache Spark 加密](#)。

如需有關 Amazon Athena for Apache Spark 的詳細資訊，請參閱 [在 Amazon Athena 使用 Apache Spark](#)。

## 2023 年 6 月 2 日

2023-06-02 發布

您現在可以刪除 Athena 中的容量保留，並使用 AWS CloudFormation 範本來指定 Athena 容量保留。

- 刪除容量保留 – 您現在可以刪除 Athena 中已取消的容量保留。必須先取消保留，然後才可將其刪除。刪除容量保留會立即從您的帳戶中移除保留。已刪除的保留無法再參考，包括其 ARN。若要刪除保留，您可以使用 Athena 主控台或 Athena API。如需詳細資訊，請參閱 [刪除容量保留](#) 亞 Amazon Athena 使用者指南和 [DeleteCapacityReservation](#) 亞 Amazon Athena API 參考中的。
- 使用容量保留 AWS CloudFormation 範本 — 您現在可以使用 AWS CloudFormation 範本，使用 `AWS::Athena::CapacityReservation` 資源來指定 Athena 容量保留。如需詳細資訊，請參閱 AWS CloudFormation 使用者指南 CapacityReservation 中的 [AWS::Athena::](#)。

如需有關使用容量保留在 Athena 佈建容量的詳細資訊，請參閱 [管理查詢處理容量](#)。

## 2023 年 5 月 25 日

2023-05-25 發布

Athena 已發布可改善聯合查詢效能的資料來源連接器更新。全新的下推最佳化和動態篩選功能，可在來源資料庫 (而非 Athena) 中執行更多作業。這些最佳化可降低查詢執行期和減少掃描的資料量。這些改進需要 Athena 引擎版本 3。

下列連接器已更新：

- [Azure Data Lake Storage](#)

- [Azure Synapse](#)
- [Cloudera Hive](#)
- [Cloudera Impala](#)
- [Db2](#)
- [DynamoDB](#)
- [谷歌 BigQuery](#)
- [Hortonworks](#)
- [MySQL](#)
- [Oracle](#)
- [PostgreSQL](#)
- [Redshift](#)
- [SAP HANA](#)
- [Snowflake](#)
- [SQL Server](#)
- [Teradata](#)

如需有關升級資料來源連接器的資訊，請參閱 [更新資料來源連接器](#)。

## 2023 年 5 月 18 日

2023-05-18 發布

您現在可以用 AWS PrivateLink 於與 Amazon Athena 的 IPv6 入站連線。

Amazon Athena 已透過網際網路通訊協定第 6 版 (IPv6) 端點擴展對傳入連線的支援，以包括 [AWS PrivateLink](#)。AWS PrivateLink 從今天開始，除了[先前](#)可用的公有 IPv6 端點之外，您還可以使用 [Amazon Virtual Private Cloud \(Amazon VPC\)](#) 安全且私密地連線到 Athena。

網際網路的快速成長正在耗盡網際網路通訊協定第 4 版 (IPv4) 地址的可用性。IPv6 數次增加可用地址數目，因此您不再需要管理 VPC 中重疊位址空間。在此版本中，您現在可以結合使用 IPv6 定址的優點與 AWS PrivateLink 的安全與效能優勢。

若要以程式設計方式連線至 AWS 服務，您可以使用 [AWS CLI](#) 或 [AWS SDK](#) 來指定端點。如需服務端點和 Athena 服務端點的詳細資訊，請參閱 Amazon Web Services 一般參考 中的 [AWS 服務端點和 Amazon Athena 端點和配額](#)。

## 2023 年 5 月 15 日

2023-05-15 發布

Athena 宣布針對 DynamoDB、CloudWatch 日誌、CloudWatch 指標和 CMDB 推出阿帕奇星火 DataSource V2 (DSV2) 連接器。AWS 使用新的 DSV2 連接器，以使用 Spark 查詢這些資料來源。DSV2 連接器使用的參數與其對應的 Athena 聯合連接器使用的相同。DSV2 連接器會直接在 Spark 工作者上執行，且不需要您部署 Lambda 函數即可予以使用。

如需詳細資訊，請參閱 [適用於 Apache Spark 的 Athena 資料來源連接器](#)。

## 2023 年 5 月 10 日

2023-05-10 發布

發行了 Athena 的 ODBC 1.1.20 驅動程式。

功能和增強功能：

- Lake Formation 端點覆寫支援。
- ADFS 身分驗證外掛程式具有用於設定依賴方值的新參數 (LoginToRP)。
- AWS 程式庫更新。

錯誤修正：

- 當 SQLPrepare() 方法提交失敗時，準備好的陳述式解除配置失敗。
- 將 C 類型轉換為 SQL 類型時，繫結預備陳述式參數時發生錯誤。
- 當 EXPLAIN 和 EXPLAIN ANALYZE 查詢使用 SQLPrepare() 和 SQLExecute() 時，無法傳回資料。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱 [使用 ODBC 連接到 Amazon Athena](#)。

## 2023 年 5 月 8 日

2023-05-08 發布

Athena 發布下列修復和改善。

- 更新了 Hudi 整合 – Athena 更新了其與 Apache Hudi 的整合。您現在可以使用 Athena 查詢 Hudi 0.12.2 資料表，且現在支援 Hudi 資料表的 Hudi 中繼資料清單。如需詳細資訊，請參閱 [使用 Athena 查詢 Apache Hudi 資料集](#) 及 [Hudi 中繼資料清單](#)。
- 時間戳記轉換修正 – 將時間戳記轉換的處理修正為精確度較低的資料類型。先前，Athena 引擎第 3 版錯誤地將值四捨五入為目標類型，而不是在轉換期間予以截斷。

下列範例說明了修正之前的不正確處理。

範例 1：從時間戳記 (以微秒為單位) 轉換為毫秒

範例資料

```
A, 2020-06-10 15:55:23.383
B, 2020-06-10 15:55:23.382
C, 2020-06-10 15:55:23.383345
D, 2020-06-10 15:55:23.383945
E, 2020-06-10 15:55:23.383345734
F, 2020-06-10 15:55:23.383945278
```

下列查詢會嘗試擷取符合特定值的時間戳記。

```
SELECT *
FROM table
WHERE timestamps.col = timestamp'2020-06-10 15:55:23.383'
```

此查詢已傳回下列結果。

```
A, 2020-06-10 15:55:23.383
C, 2020-06-10 15:55:23.383
E, 2020-06-10 15:55:23.383
```

在修正之前，Athena 沒有包含值 2020-06-10 15:55:23.383945 或 2020-06-10 15:55:23.383945278，因為它們已四捨五入為 2020-06-10 15:55:23.384。

範例 2：從時間戳記轉換至最新

下列查詢已傳回錯誤的結果。

```
SELECT date(timestamp '2020-12-31 23:59:59.999')
```

## 結果

2021-01-01

在修正之前，Athena 將值四捨五入，因此這一天有所進展。這些值現被截斷，而不是四捨五入。

## 2023 年 4 月 28 日

2023-04-28 發布

您現在可以在 Amazon Athena 上的容量保留，以在完全受管的運算容量上執行 SQL 查詢。

佈建的容量可聽工作負載管理功能，協助您優先處理序、控制和擴展最重要的互動式工作負載。您可以隨時新增容量以增加可同時執行的查詢數量、控制哪些工作負載使用該容量，以及在工作負載之間共用容量。

如需詳細資訊，請參閱 [管理查詢處理容量](#)。如需定價資訊，請造訪 [Amazon Athena 定價](#) 頁面。

## 2023 年 4 月 17 日

2023-04-17 發布

Athena 發佈 JDBC 驅動程式 2.0.36 版。該驅動程式包含新的功能並解決了問題。

### 新功能

- 您現在可以搭配使用可自訂的依賴方識別符與 AD FS 身分驗證。
- 您現在可以將使用連接器的應用程式名稱新增至使用者代理程式字串。

### 已解決問題

- 修正使用 `getSchema()` 擷取不存在的結構描述時發生的錯誤。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。

## 2023 年 4 月 14 日

2023-06-20 發布

Athena 發布下列修復和改善。

- 當您將字串轉換為時間戳時，日期和時間或時區之間需要一個空格。如需詳細資訊，請參閱 [從字串轉換為時間戳記時，日期和時間值之間需要的空格](#)。
- 移除時間戳記精確度處理方式的重大變更。為了維持 Athena 引擎版本 2 和 Athena 引擎版本 3 之間的一致性，時間戳記精確度現在預設為毫秒 (而非微秒)。
- Athena 現在會在執行查詢時持續強制執行查詢輸出儲存貯體的存取權。請確定執行 [StartQueryExecution](#) 動作的所有 IAM 主體都具有查詢輸出儲存貯體的 [S3: GetBucketLocation](#) 權限。

## 2023 年 4 月 4 日

2023-04-04 發布

您現已可用 Amazon Athena 來建立和查詢聯合資料來源上的檢視。使用單一聯合檢視來查詢多個外部資料表或資料子集。這樣可以簡化所需的 SQL，並讓您靈活地模糊化必須使用 SQL 來查詢資料的最終使用者的資料來源。

如需詳細資訊，請參閱 [使用檢視](#) 及 [執行聯合查詢](#)。

## 2023 年 3 月 30 日

2023-03-30 發布

Amazon Athena 宣佈在其他 AWS 區域推出 Amazon Athena for Apache Spark。

此版本擴展了 Amazon Athena for Apache Spark 的可用性，包括亞太區域 (孟買)、亞太區域 (新加坡)、亞太區域 (雪梨) 和歐洲 (法蘭克福)。

如需有關 Amazon Athena for Apache Spark 的詳細資訊，請參閱 [在 Amazon Athena 使用 Apache Spark](#)。

## 2023 年 3 月 28 日

2023-03-28 發布

Athena 發布下列修復和改善。

- 在 GetQueryExecution 和 BatchGetQueryExecution Athena API 動作的回應中，新 subStatementType 欄位會顯示執行的查詢類型 (例如 SELECT、INSERT、UNLOAD、CREATE\_TABLE 或 CREATE\_TABLE\_AS\_SELECT)。
- 已修正 Apache Hive 寫入作業未正確加密資訊清單檔案的錯誤。

- Athena 引擎版本 3 現在可以正確處理 `approx_percentile` 函數中的 NaN 和 Infinity 值。`approx_percentile` 函數會傳回給定百分比的資料集的近似百分位數。

Athena 引擎版本 2 錯誤地將 NaN 視為大於 Infinity 的值。Athena 引擎版本 3 現在會根據其他分析和統計函數中度這些值的處理方式來處理 NaN 和 Infinity。以下幾點更詳細地說明了新行為。

- 如果 NaN 在資料集中存在，則 Athena 會傳回 NaN。
- 如果 NaN 不存在，但 Infinity 存在，Athena 會將 Infinity 視為一個非常大的數值。
- 如果存在多個 Infinity 值，Athena 會將它們視為相同的非常大的數值。如有必要，Athena 會輸出 Infinity。
- 如果單一資料集同時具有 Infinity 和 `-Double.MAX_VALUE`，且百分位數結果為 `-Double.MAX_VALUE`，Athena 便會傳回 `-Infinity`。
- 如果單一資料集同時具有 Infinity 和 `Double.MAX_VALUE`，且百分位數結果為 `Double.MAX_VALUE`，Athena 便會傳回 Infinity。
- 若要從計算中排除 Infinity 和 NaN，請使用 `is_finite()` 函數，如下列範例所示。

```
approx_percentile(x, 0.5) FILTER (WHERE is_finite(x))
```

## 2023 年 3 月 27 日

2023-03-27 發布

您現在可以為 Amazon Athena 的 Athena SQL 工作群組指定最低加密層級。此功能可確保 Athena SQL 工作群組中所有查詢的結果在您指定的加密層級或以上加密。您可以選擇多種層級的加密強度來保護您的資料。若要設定所需的最低加密層級，您可以使用 Athena 主控台 AWS CLI、API 或 SDK。

最低加密功能不適用於已啟用 Apache Spark 的工作群組。如需詳細資訊，請參閱 [為工作群組設定最低加密](#)。

## 2023 年 3 月 17 日

2023-03-17 發布

Athena 發布下列修復和改善。

- 修正 Amazon Athena DynamoDB 連接器造成查詢失敗且錯誤訊息 `KeyConditionExpressions` 每個金鑰只能包含一個條件的問題。

此問題是因為 Athena 引擎版本 3 能夠比 Athena 引擎版本 2 下推更多類型的述詞。在 Athena 引擎版本 3 中，類似 `some_column LIKE 'someprefix%'` 的子句會被下推為在指定資料欄上套用下限和上限的篩選條件述詞。Athena 引擎版本 2 沒有下推這些述詞。在 Athena 引擎版本 3 中，如果 `some_column` 是排序索引鍵資料欄，引擎會將篩選條件述詞下推至 DynamoDB 連接器。接著，篩選條件述詞會進一步下推至 DynamoDB 服務。由於 DynamoDB 在排序索引鍵上不支援多個篩選條件，因此 DynamoDB 會傳回錯誤。

若要修正此問題，請將您的 Amazon Athena DynamoDB 連接器更新至 2023.11.1 版。如需有關更新連接器的說明，請參閱 [更新資料來源連接器](#)。

## 2023 年 3 月 8 日

2023-03-08 發布

Athena 發布下列修復和改善。

- 修正聯合查詢的問題，該問題會造成時間戳記述詞值以微秒而非毫秒傳送。

## 2023 年 2 月 15 日

2023-02-15 發布

Athena 發布下列修復和改善。

- 您現在可以使用 [用戶端加密](#) 來加密 Amazon S3 中的資料，以進行 Iceberg 寫入操作。
- 修正會影響 Amazon S3 中 Iceberg 寫入操作的 [伺服器端加密](#) 的問題。

## 2023 年 1 月 31 日

2023 年 1 月 31 日發布

您現在可使用 Amazon Athena 在 Google Cloud Storage 中查詢資料。與 Amazon S3 一樣，Google Cloud Storage 是一項受管服務，可將資料存放在儲存貯體中。使用適用於 Google Cloud Storage 的 Athena 連接器，對外部資料執行互動式聯合查詢。

如需詳細資訊，請參閱 [Amazon Athena Google Cloud Storage 連接器](#)。

## 2023 年 1 月 20 日

2023 年 1 月 20 日發布

您現在可以查看 Athena 壓縮支援的擴充文件。已針對 [Hive 資料表壓縮](#)、[Iceberg 資料表壓縮](#) 和 [ZSTD 壓縮級別](#) 新增單獨主題。

如需詳細資訊，請參閱 [Athena 壓縮支援](#)。

## 2023 年 1 月 3 日

2023-01-03 發布

Athena 發布下列更新：

- Hive 中繼存放區的其他命令 – 您可以使用 Athena 連線到自我管理的 Apache Hive 中繼存放區作為中繼資料目錄，並查詢存放在 Amazon S3 中的資料。在此版本中，您可以使用 CREATE TABLE AS (CTAS)、INSERT INTO 和 12 個其他資料定義語言 (DDL) 命令，以與 Apache Hive 中繼存放區進行互動。您可以使用這組擴充的 SQL 功能，直接從 Athena 管理 Hive 中繼存放區結構描述。

如需詳細資訊，請參閱 [將 Athena 資料連接器用於外部 Hive 中繼存放區](#)。

- JDBC 驅動程式 2.0.35 版 – Athena 發布 JDBC 驅動程式 2.0.35 版 JDBC 2.0.35 驅動程式包含下列更新：
  - 該驅動程式現在為 Jackson JSON 剖析器使用以下程式庫。
    - jackson-annotations 2.14.0 (先前為 2.13.2)
    - jackson-core 2.14.0 (先前為 2.13.2)
    - jackson-databind 2.14.0 (先前為 2.13.2.2)
  - 對 JDBC 4.1 版的 Support 已經停止。

如需詳細資訊並且要下載新的驅動程式、版本備註和文件，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。

## 2022 年 Athena 版本備註

### 2022 年 12 月 14 日

2022-12-14 發布

您現在可以使用適用於 Kafka 的 Amazon Athena 連接器，對串流資料執行 SQL 查詢。例如，您可以在 Amazon Managed Streaming for Apache Kafka (Amazon MSK) 中對即時串流資料執行分析查詢，並將其與 Amazon S3 資料湖中的歷史資料聯結在一起。

適用於 Kafka 的 Amazon Athena 連接器支援多個串流引擎的查詢。您可以使用 Athena 在 Amazon MSK 佈建的無伺服器叢集、自我管理的 Kafka 部署，以及在 Confluent 雲端中的串流資料上執行 SQL 查詢。

如需詳細資訊，請參閱 [Amazon Athena MSK 連接器](#)。

## 2022 年 12 月 2 日

2022-12-02 發布

Athena 發佈 JDBC 驅動程式 2.0.34 版。JDBC 2.0.34 驅動程式包括下列新功能和已解決的問題：

- 查詢結果重複使用支援 – 您現在可以在指定的時間限制內重複使用先前執行的查詢的結果，而不是讓 Athena 在每次執行查詢時重新計算結果。如需詳細資訊，請參閱 JDBC 下載頁面中的安裝和設定指南和 [重複使用查詢結果](#)。
- EC2 InstanceMetadata 支援 — JDBC 驅動程式現在支援使用 IAM [執行個體設定檔](#) 的 [EC2 InstanceMetadata 身份驗證方法](#)。
- 基於字元的里外狀況修正 – 修正了查詢包含某些語言字元時發生的例外狀況。
- 弱點修正 — 已更正與連接器一起封裝的相 AWS 依性相關的弱點。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。

## 2022 年 11 月 30 日

2022-11-30 發布

您現在可以在 Athena 上以互動方式建立和執行 Apache Spark 應用程式和與 Jupyter 相容的筆記本。使用 Spark 在 Athena 上執行資料分析，而不必規劃、設定或管理資源。提交 Spark 程式碼進行處理並直接接收結果。在 Amazon Athena 主控台中使用筆記本的簡化體驗，以使用 Python 或 [Athena 筆記本 API](#) 開發 Apache Spark 應用程式。

Amazon Athena 上的 Apache Spark 是無伺服器，並提供自動的隨需擴展，可提供即時啟動運算，以滿足不斷變化的資料量和處理需求。

如需詳細資訊，請參閱 [在 Amazon Athena 使用 Apache Spark](#)。

## 2022 年 11 月 18 日

2022-11-18 發布

您現在可以使用適用於 IBM Db2 的 Amazon Athena 連接器，查詢來自 Athena 的 Db2。例如，您可以在 Db2 上的資料倉儲和 Amazon S3 中的資料湖上執行分析查詢。

Amazon Athena Db2 連接器透過 Lambda 環境變數公開數個組態選項。如需有關組態選項、參數、連接字串、部署和限制的詳細資訊，請參閱 [Amazon Athena IBM Db2 連接器](#)。

## 2022 年 11 月 17 日

2022-11-17 發布

Athena 引擎版本 3 中的 Apache Iceberg 支援現提供下列增強型 ACID 交易功能：

- ORC 和 Avro 支援 – 使用 [Apache Avro](#) 和 [Apache ORC](#) 以資料列和資料欄為基礎的檔案格式，建立 Iceberg 資料表。對這些格式的支援即為對 Parquet 現有支援的補充。
- 合併為 – 使用 MERGE INTO 命令有效率地大規模合併資料。MERGE INTO 將 INSERT、UPDATE 和 DELETE 運算合併為一個交易。這樣可以減少資料管道中的處理負擔，並減少寫入的 SQL。如需詳細資訊，請參閱 [更新 Iceberg 資料表資料](#) 及 [合併為](#)。
- CTAS 和 VIEW 支援 – 搭配使用 CREATE TABLE AS SELECT (CTAS) 和 CREATE VIEW 陳述式與 Iceberg 資料表。如需詳細資訊，請參閱 [CREATE TABLE AS](#) 及 [CREATE VIEW](#)。
- VACUUM 支援 – 您可以使用 VACUUM 陳述式刪除不再需要的快照和資料，進而最佳化資料湖。您可以使用此功能來改善讀取效能並符合 [GDPR](#) 等法規要求。如需詳細資訊，請參閱 [最佳化處理 Iceberg 資料表](#) 及 [VACUUM](#)。

這些新功能需要 Athena 引擎版本 3，並且適用於支援 Athena 的所有區域。您可以搭配 [Athena 主控台](#)、[驅動程式](#) 或 [API](#) 使用它們。

如需有關在 Athena 中使用 Iceberg 的詳細資訊，請參閱 [使用 Apache Iceberg 資料表](#)。

## 2022 年 11 月 14 日

2022-11-14 發布

Amazon Athena 現在支援用於傳入連線的 IPv6 端點，而您可以使用這些端點透過 IPv6 叫用 Athena 函數。您可以使用此功能來滿足 IPv6 合規要求。它也不需要額外的聯網設備來處理 IPv4 和 IPv6 之間的位址轉譯。

若要使用此功能，請將應用程式設定為使用支援 IPv4 和 IPv6 的全新 Athena 雙堆疊端點。雙堆疊端點使用該格式 `athena.region.api.aws`。例如，美國東部 (維吉尼亞北部) 區域的雙堆疊端點是 `athena.us-east-1.api.aws`。

當您請求雙堆疊 Athena 端點時，端點會解析為 IPv6 或 IPv4 地址，具體視您的網路和用戶端使用的通訊協定而異。若要以程式設計方式連線至 AWS 服務，您可以使用 [AWS CLI](#) 或 [AWS SDK](#) 來指定端點。

如需有關服務端點的詳細資訊，請參閱 [AWS 服務端點](#)。若要進一步了解 Athena 的服務端點，請參閱 AWS 文件中的 [Amazon Athena 端點和配額](#)。

您可以使用新的 Athena 雙堆疊端點進行傳入連線，且無需額外付費。雙堆疊端點通常可用於所有 AWS 區域。

## 2022 年 11 月 11 日

2022-11-11 發布

Athena 發布下列修復和改善。

- 擴充的 Lake Formation 精細存取控制 – 您現在可以在 Athena 查詢中對以任何支援檔案或資料表格式儲存的資料使用 [AWS Lake Formation](#) 精細存取控制政策。您可以在 Lake Formation 中使用精細存取控制，使用資料篩選條件來限制對查詢結果中資料的存取，從而達到資料欄層級、資料列層級和儲存格層級安全性。在 Athena 中支援的資料表格式包括 Apache Iceberg、Apache Hudi 和 Apache Hive。Athena 支援的所有區域均提供擴充的精細存取控制功能。擴充的資料表和檔案格式支援需要 [Athena 引擎版本 3](#)，這可 [提供新功能和改進的查詢效能](#)，但不會變更您在 Lake Formation 中設定精細存取控制政策的方式。

在 Athena 中使用此擴充的精細存取控制有下列考量：

- EXPLAIN – 在 Lake Formation 中定義的資料列或儲存格篩選資訊及查詢統計資料資訊未在 EXPLAIN 和 EXPLAIN ANALYZE 的輸出中顯示。如需有關 Athena 中的 EXPLAIN 的詳細資訊，請參閱 [在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)。
- 外部 Hive 中繼存放區 – Apache Hive 隱藏資料欄，無法用於精細存取控制篩選，而且精細存取控制不支援 Apache Hive 隱藏系統資料表。如需詳細資訊，請參閱 [將 Athena 資料連接器用於外部 Hive 中繼存放區](#) 主題中的 [考量與限制](#)。
- 查詢統計資料 – 當查詢具有 Lake Formation 中定義的資料列層級篩選條件時，Athena 查詢統計資料不會顯示階段層級輸入和輸出資料列計數和資料大小資訊。如需查看 Athena 查詢統計資料的詳細資訊，請參閱 [檢視已完成查詢的統計資料和執行詳細資訊](#) 和 [GetQueryRuntimeStatistics](#)。

- 工作群組 – 同一 Athena 工作群組中的使用者可以看到 Lake Formation 精細存取控制設定為可供工作群組存取的資料。如需有關使用 Athena 查詢向 Lake Formation 註冊的資料的詳細資訊，請參閱 [使用 Athena 查詢向 AWS Lake Formation 註冊的資料](#)。

如需有關在 Lake Formation 中使用精細定義存取控制的詳細資訊，請參閱 AWS 大數據部落格中的 [使用 AWS Lake Formation 管理精細定義存取控制](#)。

- Athena 聯合查詢 – Athena 聯合查詢現在會保留 struct 物件中欄位名稱的原始大小寫。先前，struct 欄位名稱會自動變成小寫。

## 2022 年 11 月 8 日

2022-11-08 發布

您現在可以使用查詢結果重複使用快取功能，進而加速 Athena 中的重複查詢。重複查詢是與最近提交的 SQL 查詢相同的查詢，會產生相同的結果。當您需要執行相同的多個查詢時，結果重複使用快取可以減少產生結果所需的時間。結果重複使用快取也會減少掃描的位元組數，進而降低成本。

如需詳細資訊，請參閱 [重複使用查詢結果](#)。

## 2022 年 10 月 13 日

2022 年 10 月 13 日發布

Athena 宣布推出 Athena 引擎版本 3。

Athena 已升級其 SQL 查詢引擎，以納入 [Trino](#) 開放原始碼專案的最新功能。Athena 引擎版本 3 不僅支援 Athena 引擎版本 2 的所有功能，還包含 50 多個新 SQL 函數、30 項新功能，以及 90 多項查詢效能改進。今天推出的版本中，Athena 也仍然採用持續整合方法，管理開放原始碼軟體，確保 Trino 和 [Presto](#) 專案與時俱進，讓您可以更快速地使用到各種改善後的社群功能，並在 Athena 引擎中執行整合和調整。

如需詳細資訊，請參閱 [Athena 引擎版本 3](#)。

## 2022 年 10 月 10 日

2022 年 10 月 10 日發布

Athena 發佈 JDBC 驅動程式 2.0.33 版。JDBC 2.0.33 驅動程式有以下變更：

- 新的驅動程式版本、JDBC 版本和外掛程式名稱屬性已新增至憑證供應商類別中的 user-agent 字串。
- 錯誤訊息已更正且新增了必要的資訊。
- 現在，如果連線關閉或 Athena 的預備陳述式執行失敗，則系統會取消配置預備陳述式。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

## 2022 年 9 月 23 日

2022 年 9 月 26 日發布

Amazon Athena Neptune 連接器現在支援不區分大小寫的資料欄和資料表名稱比對。

- Neptune 資料來源連接器可以解析 Neptune 資料表上使用大小寫的資料欄名稱，即使 AWS Glue 中資料表的資料欄名稱都是小寫也一樣。若要啟用此行為，請在 Neptune 連接器 Lambda 函數上將 `enable_caseinsensitivematch` 環境變數設定為 `true`。
- 由於僅 AWS Glue 支援小寫資料表名稱，因此當您為 Neptune 建立資 AWS Glue 料表時，請指定 AWS Glue table 參數 `"glabel" = table_name`。

如需有關 Neptune 連接器的詳細資訊，請參閱[Amazon Athena Neptune 連接器](#)。

## 2022 年 9 月 13 日

2022 年 9 月 13 日發布

Athena 發布下列修復和改善。

- 外部 Hive 中繼存放區 - 當 WHERE 子句包含[外部 Hive 中繼存放區](#) (EHMS) 中不存在的分割區時，Athena 現在會傳回 NULL 而不是擲回例外狀況。新行為符合 AWS Glue Data Catalog 的行為。
- 參數化查詢 - [參數化查詢](#) 中的值現在可轉換為 DOUBLE 資料類型。
- Apache Iceberg - 在 Amazon S3 儲存貯體中啟用 [Object Lock](#) (物件鎖定) 時，對 [Iceberg 資料表](#) 的寫入操作現在會成功。

## 2022 年 8 月 31 日

2022 年 8 月 31 日發布

Amazon Athena 宣佈 Athena 及其[功能](#)可在亞太區域 (雅加達) 使用。

此版本擴展了 Athena 在亞太區域的可用性，以包含亞太區域 (香港)、亞太區域 (雅加達)、亞太區域 (孟買)、亞太區域 (大阪)、亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (新加坡)、亞太區域 (雪梨) 和亞太區域 (東京)。如需在這些和其他區域中的可用 AWS 服務完整清單，請參閱 [AWS 區域區域服務清單](#)。

## 2022 年 8 月 23 日

2022-08-23 發布

Athena Query Federation 軟體開發套件 [2022.32.1 版](#) 包含下列變更：

- Amazon Athena Oracle 資料來源連接器新增以 SSL 連線 Amazon RDS 執行個體的相關支援。支援範圍僅限於 Transport Layer Security (TLS) 通訊協定，以及用戶端對伺服器執行的身分驗證作業。由於 Amazon RDS 不支援交互身分驗證，因此該更新不包含對交互身分驗證的支援。

如需詳細資訊，請參閱 [Amazon Athena Oracle 連接器](#)。

## 2022 年 8 月 3 日

2022-08-03 發布

Athena 發佈 JDBC 驅動程式 2.0.32 版。JDBC 2.0.32 驅動程式包括以下變更：

- 傳送至 Athena 軟體開發套件的 User-Agent 字串已有所擴充，內含驅動程式版本、JDBC 規格版本，以及身分驗證外掛程式的名稱。
- 修正未提供任何 CheckNonProxyHost 參數值時，系統擲回的 NullPointerException。
- 修正 BrowserSaml 驗證外掛程式中 login\_url 剖析的問題。
- 修正 UseProxyforIdp 參數設為 true 時所發生的代理主機問題。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。

## 2022 年 8 月 1 日

2022-08-01 發布

Athena 宣布針對 Athena Query Federation 軟體開發套件和 Athena 預先建置資料來源連接器的改善改善項目包括：

- 結構剖析 – 修正了 Athena Query Federation 軟體開發套件中的 GlueFieldLexer 剖析問題，會導致某些複雜的結構無法顯示其所有資料。此問題會影響在 Athena Query Federation 軟體開發套件上建置的連接器。
- AWS Glue 表格 — 已新增對 AWS Glue 表格中 set 和 decimal 欄類型的其他支援。
- DynamoDB 連接器 – 新增了忽略 DynamoDB 屬性名稱上大小寫的功能。如需詳細資訊，請參閱 [Amazon Athena DynamoDB 連接器](#) 頁面之 [參數](#) 區段的 `disable_projection_and_casing`。

如需詳細資訊，請參閱 [《Athena 查詢聯盟》v2022.30.2](#) 版。GitHub

## 2022 年 7 月 21 日

2022-07-21 發布

您現在可以在 Athena 主控台中使用效能指標和互動式視覺化查詢分析工具來對您的查詢進行分析和除錯。查詢效能資料和執行詳細資訊可協助您找出查詢中的瓶頸、檢查每個查詢階段的運算子和統計資料、追蹤階段之間流動的資料量，以及驗證查詢述詞的影響。您現在可以：

- 只要按一下，即可存取您查詢的分散式和邏輯執行計劃。
- 在階段執行之前，探索每個階段的操作。
- 使用在佇列、規劃和執行階段所花費時間的指標，將已完成查詢的效能視覺化。
- 針對您的查詢所處理和輸出的資料列數和來源資料數量，取得相關資訊。
- 查看以情境呈現並採用互動式圖形格式的查詢的精細執行詳細資訊。
- 使用精確的階段層級執行詳細資訊，來了解查詢過程中的資料流。
- 使用亦於今日發布的新的 API，以程式設計方式分析查詢效能資料，以 [《取得查詢執行期資料》](#)。

若要了解如何在查詢中使用這些功能，請觀看影片教學課程在頻 AWS YouTube 道上使用 [新的查詢分析工具優化 Amazon Athena](#) 查詢。

如需說明文件，請參閱 [檢視 SQL 查詢的執行計劃](#) 和 [檢視已完成查詢的統計資料和執行詳細資訊](#)。

## 2022 年 7 月 11 日

2022-07-11 發布

您現在可以直接從 Athena 主控台或 API 執行參數化查詢，而不需要事先準備 SQL 陳述式。

當您在 Athena 主控台中執行具有問號形式參數的查詢時，使用者介面現在會提示您直接輸入參數值。如此一來，每次您想要執行查詢時，就不需要在查詢編輯器中修改字面值。

如果您使用增強的 [《查詢執行》](#) API，您現在可以在單一呼叫中提供執行參數及其值。

如需詳細資訊，請參閱本使用者指南中的 [使用參數化查詢](#) 以及 AWS 大數據部落格文章 [Use Amazon Athena parameterized queries to provide data as a service](#) (使用 Amazon Athena 參數化查詢提供資料即服務)。

## 2022 年 7 月 8 日

2022-07-08 發布

Athena 發布下列修復和改善。

- 修正導致查詢失敗之 SageMaker 端點 (UDF) 的 DATE 資料行轉換處理問題。

## 2022 年 6 月 6 日

2022-06-06 發布

Athena 發佈了 JDBC 驅動程式 2.0.31 版。JDBC 2.0.31 驅動程式包括以下變更：

- log4j 相依性問題 – 已解決因 log4j 相依性所引起的找不到驅動程式類別錯誤訊息。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。

## 2022 年 5 月 25 日

2022 年 5 月 25 日發布

Athena 發布下列修復和改善。

- Iceberg 支援
  - 推出跨區域查詢支援。現在，您可以查詢與您正在使用的不同 AWS 區域的 Iceberg 表。AWS 區域 中國地區不支援跨區域查詢。
  - 推出對伺服器端加密組態的支援。現在，您可以使用 [SSE-S3/SSE-KMS](#)，來加密 Amazon S3 中的 Iceberg 寫入操作。

如需有關在 Athena 中使用 Apache Iceberg 的詳細資訊，請參閱 [使用 Apache Iceberg 資料表](#)。

- JDBC 2.0.30 驅動程式發布

Athena 的 JDBC 2.0.30 Athena 驅動程式進行了以下改進：

- 修復了影響參數化準備陳述式的資料爭用問題。
- 修復了 Gradle 建置環境中發生的應用程式啟動問題。

若要下載 JDBC 2.0.30 驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

## 2022 年 5 月 6 日

2022-05-06 發布

推出了 Athena JDBC 2.0.29 和 ODBC 1.1.17 驅動程式。

這些驅動程式包括下列變更：

- 更新了 SAML 外掛程式瀏覽器啟動程序。

如需有關這些變更以及下載新驅動程式、版本備註和文件的詳細資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)和[使用 ODBC 連接到 Amazon Athena](#)。

## 2022 年 4 月 22 日

2022-04-22 發布

Athena 發布下列修復和改善。

- 修正[分割區索引和篩選功能](#)中的問題，其中滿足以下條件時會發生分割快取：
  - 在資料表的 AWS Glue 資料表屬性true中，partition\_filtering.enabled索引鍵已設定為。
  - 可以採用不同的分割區篩選條件值多次使用相同的資料表。

## 2022 年 4 月 21 日

2022-04-21 發布

您現在可以使用 Amazon Athena 在新的數據源上運行聯合查詢，包括谷歌 BigQuery，Azure 突觸和雪花。新的資料來源連接器包括：

- [Azure Data Lake Storage \(ADLS\) Gen2](#)
- [Azure Synapse](#)

- [Cloudera Hive](#)
- [Cloudera Impala](#)
- [谷歌 BigQuery](#)
- [Hortonworks](#)
- [Microsoft SQL Server](#)
- [Oracle](#)
- [SAP HANA \(Express 版本\)](#)
- [Snowflake](#)
- [Teradata](#)

如需 Athena 支援的資料來源完整清單，請參閱[可用資料來源連接器](#)。

為了更輕鬆地瀏覽可用來源並連接到資料，您現在可以從 Athena 主控台中更新的 Data Sources (資料來源) 螢幕搜尋、排序及篩選可用的連接器。

若要瞭解有關查詢聯合來源的資訊，請參閱[使用 Amazon Athena 聯合查詢](#) 和 [執行聯合查詢](#)。

## 2022 年 4 月 13 日

2022-04-13 發布

Athena 發佈了 JDBC 驅動程式 2.0.28 版。JDBC 2.0.28 驅動程式包括以下變更：

- JWT 支援 – 驅動程式現已支援 JSON Web 字符 (JWT) 進行身分驗證。如需搭配使用 JWT 與 JDBC 驅動程式的資訊，請參閱安裝和設定指南，可從[JDBC 驅動程式頁面](#)下載該指南。
- 更新 Log4j 程式庫 – JDBC 驅動程式現使用以下 Log4j 程式庫：
  - Log4j-api 2.17.1 (先前為 2.17.0)
  - Log4j-core 2.17.1 (先前為 2.17.0)
  - Log4j-jcl 2.17.2
- 其他改進 – 新驅動程式還包含以下改進和錯誤修正：
  - Athena 預備陳述式功能現可透過 JDBC 使用。如需預備陳述式的資訊，請參閱[使用參數化查詢](#)。
  - Athena JDBC SAML 聯合現可在中國地區正常運作。
  - 其他次要改進。

如需詳細資訊並下載新的驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

## 2022 年 3 月 30 日

2022-03-30 發布

Athena 發布下列修復和改善。

- 跨區域查詢 — 您現在可以使用 Athena 查詢位於 Amazon S3 儲存貯體中的資料，AWS 區域 包括亞太區域 (香港)、中東 (巴林)、非洲 (開普敦) 和歐洲 (米蘭)。中國地區不支援跨區域查詢。
  - 如需可使用雅典娜 AWS 區域 的清單，請參閱 [Amazon Athena 端點和配額](#)。
  - 如需啟用預設為停 AWS 區域 用的相關資訊，請參閱 [啟用區域](#)。
  - 如需有關跨區域查詢的資訊，請參閱 [跨區域查詢](#)。

## 2022 年 3 月 18 日

2022-03-18 發布

Athena 發布下列修復和改善。

- 動態篩選 – 透過有效地將篩選條件套用於相應資料的每條記錄，改善了整數欄的 [動態篩選](#)。
- Iceberg – 修復寫入大於 2GB 的 Iceberg Parquet 檔案時導致故障的問題。
- 未壓縮輸出 – [CREATE TABLE](#) 陳述式現在支援編寫未壓縮的檔案。若要編寫未壓縮檔案，請使用下列語法：
  - CREATE TABLE (文字檔案或 JSON) – 在 TBLPROPERTIES 中，指定 `write.compression = NONE`。
  - CREATE TABLE (Parquet) – 在 TBLPROPERTIES 中，指定 `parquet.compression = UNCOMPRESSED`。
  - CREATE TABLE (ORC) – 在 TBLPROPERTIES 中，指定 `orc.compress = NONE`。
- 壓縮 – 修復了使用非預設壓縮方法時以一種格式建立壓縮檔案，但使用另一種壓縮格式檔案副檔名的文字檔案資料表插入問題。
- Avro – 修復了從 Avro 檔案讀取固定類型的小數時發生的問題。

## 2022 年 3 月 2 日

2022-03-02 發布

Athena 發布了下列功能以及強化功能。

- 針對查詢結果儲存貯體，當 [ACL 已啟用](#) 時，您現在可以授予 Simple Storage Service (Amazon S3) 儲存貯體擁有者對查詢結果的完整控制存取。如需詳細資訊，請參閱 [指定查詢結果位置](#)。
- 您現在可以更新現有已命名的查詢。如需詳細資訊，請參閱 [使用已儲存的查詢](#)。

## 2022 年 2 月 23 日

2022-02-23 發布

Athena 發布下列修復和效能改善。

- 記憶體處理改善，以強化效能並減少記憶體錯誤。
- Athena 現在讀取 ORC 時間戳記欄以及存放在 Stripe 註腳中的時區資訊，並連同註腳中的時區 (UTC) 寫入 ORC 檔案。如果要讀取的 ORC 檔案是在非 UTC 時區環境中建立的，則這只會影響 ORC 時間戳記讀取的行為。
- 修復導致次優查詢計畫的錯誤的符號連結資料表大小估算。
- 現在可以在 Athena 主控台中從 Hive 中繼存放區資料來源查詢橫向展開檢視。
- 改善 Simple Storage Service (Amazon S3) 讀取錯誤訊息，以包含更詳細的 [Simple Storage Service \(Amazon S3\) 錯誤代碼](#) 資訊。
- 修復導致 ORC 格式輸出檔案與 Apache Hive 3.1 不相容的問題。
- 修復導致在某些 DML 和 DDL 查詢中具有引號的資料名稱失敗的問題。

## 2022 年 2 月 15 日

2022-02-15 發布

Amazon Athena 已增加所有 AWS 區域的使用中 DML 查詢配額。作用中查詢包括執行中和佇列中的查詢。借由本次變更，您現在可以擁有比以前更多的作用中狀態的 DML 查詢。

如需 Athena 服務配額的相關資訊，請參閱 [Service Quotas](#)。有關使用 Athena 的區域中的查詢配額，請參閱 AWS 一般參考 中的 [Amazon Athena 端點和配額](#)。

若要監控配額使用量，您可以使用 CloudWatch 用量度。Athena 在 AWS/Usage 命名空間中發布 ActiveQueryCount 指標。如需詳細資訊，請參閱 [監控 Athena 用量指標](#)。

查看您的用量後，您可以使用 [Service Quotas](#) 主控台請求提高配額。如果您之前已請求提高帳戶的配額，則如果您請求的配額超過新的預設作用中 DML 查詢配額，則請求的配額仍適用。否則，所有帳戶都會使用新的預設值。

## 2022 年 2 月 14 日

2022-02-14 發布

此版本將 `ErrorType` 分欄新增至 Athena [GetQueryExecution](#) API 動作中的 [AthenaError](#) 回應物件。

雖然現有的 `ErrorCategory` 欄位顯示失敗查詢的一般來源 (系統、使用者或其他)，但新的 `ErrorType` 欄位會提供發生的錯誤的更詳細資訊。結合這兩個欄位中的資訊，以取得查詢失敗原因的洞察。

如需詳細資訊，請參閱 [Athena 錯誤目錄](#)。

## 2022 年 2 月 9 日

2022-02-09 發布

舊版 Athena 主控台無法再使用。Athena 的新版主控台可支援舊版主控台的所有功能，但具有更好用的新型介面，內含的新功能可改進查詢開發、分析資料和管理使用情況等體驗。若要使用新版的 Athena 主控台，請造訪 <https://console.aws.amazon.com/athena/>。

## 2022 年 2 月 8 日

2022-02-08 發布

預期值區擁有者 — 作為一項額外的安全性措施，您現在可以選擇性地指定您希望成為 Athena 中查詢結果輸出位置值區擁有者的 AWS 帳戶 ID。如果查詢結果儲存貯體擁有者的帳戶 ID 與您在此處指定的 ID 不符，則嘗試輸出到儲存貯體的動作會失敗並產生 Simple Storage Service (Amazon S3) 許可錯誤。您可在客戶端或工作群組層級進行此設定。

如需詳細資訊，請參閱 [指定查詢結果位置](#)。

## 2022 年 1 月 28 日

2022-01-28 發布

Athena 推出下列引擎強化功能。

- Apache Hudi – 在 Hudi 讀取時合併 (MOR) 資料表上的快照查詢現在可以讀取有 INT64 資料類型的時間戳記資料欄。
- UNION 查詢 – 效能改進並減少某些 UNION 查詢 (會掃描同一個資料表多次) 的掃描次數。
- 分區查詢 – 篩選條件只使用各分割區資料欄的分區值時提高查詢效能。

- 分割區投影強化功能
  - 現在篩選條件已可使用分區值來查詢 injected 類型的資料欄。如需詳細資訊，請參閱 [注入類型](#)。
  - 篩選條件只使用諸如 CHAR 或 VARCHAR 等字串型資料欄分區值來進行查詢的效能已經改進。

## 2022 年 1 月 13 日

2022-01-13 發布

推出了 Athena JDBC 2.0.27 和 ODBC 1.1.15 驅動程式。

JDBC 2.0.27 驅動程式包括以下變更：

- 驅動程式已更新，可檢索外部目錄。
- 延伸的驅動程式版本號現已包含在 Athena API 呼叫的 user-agent 字串中。

ODBC 1.1.15 驅動程式包括以下變更：

- 透過第二次呼叫 SqlParameterData() 來修正問題。

如需有關這些變更以及下載新驅動程式、版本備註和文件的詳細資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)和[使用 ODBC 連接到 Amazon Athena](#)。

## 2021 年 Athena 版本備註

### 2021 年 11 月 26 日

2021-11-26 發布

Athena 推出的 Athena ACID 交易公開預覽版，可在 Athena 的 SQL 資料處理語言 (DML) 中新增寫入、刪除、更新和時間歷程查詢等操作。Athena ACID 交易可讓多名使用者同時對 Simple Storage Service (Amazon S3) 資料進行可靠的資料列級修改。以 [Apache Iceberg](#) 資料表格式建立的 Athena ACID 交易與 [Amazon EMR](#) 和 [Apache Spark](#) 等其他也支援 Iceberg 資料表格式的服務相容。

Athena ACID 交易和熟悉的 SQL 語法可簡化您的業務和法規資料的更新。例如，若要回應資料清除請求，您可以執行 SQL DELETE 操作。若要進行手動記錄更正，您可以使用單一 UPDATE 陳述式。若要復原最近刪除的資料，您可以使用 SELECT 陳述式發布時間歷程查詢。Athena 交易可透過 Athena 主控台、API 操作以及 ODBC 和 JDBC 驅動程式取得。

如需詳細資訊，請參閱 [使用 Athena ACID 交易](#)。

## 2021 年 11 月 24 日

2021-11-24 發布

Athena 宣佈支援讀取和寫入 [ZStandard](#) 壓縮 ORC、Parquet 和文字檔案資料。寫入 ZStandard 壓縮資料時，Athena 會使用 ZStandard 壓縮第 3 級。

如需有關 Athena 中資料壓縮的資訊，請參閱 [Athena 壓縮支援](#)。

## 2021 年 11 月 22 日

2021-11-22 發布

您現在可以從 Amazon Athena 主控台管理 AWS Step Functions 工作流程，輕鬆建置可擴展的資料處理管道、根據自訂商業邏輯執行查詢、自動化管理和警示任務等等。

Step Functions 現已與 Athena 的升級主控台整合在一起，您可以使用它來查看調用 Athena 的狀態機器的互動式工作流程圖。若要開始使用，請在左邊的導覽窗格中選擇工作流程。如果現有的狀態機器具有 Athena 查詢功能，請選擇該狀態機器以查看互動式工作流程圖。如果您是 Step Functions 的新手，一開始可使用 Athena 主控台啟動範例專案並自訂該專案，使其配合您的使用案例的需求。

如需詳細資訊，請參閱 [使用 Amazon Athena 建置和協調 ETL 管道 AWS Step Functions](#)，或參閱 [Step Functions](#) 文件。

## 2021 年 11 月 18 日

2021-11-18 發布

Athena 發布了新版功能和強化功能。

- Support 包含 DISTINCT、ORDER BY 或兩者的彙總查詢，如下列範例所示：spill-to-disk

```
SELECT array_agg(orderstatus ORDER BY orderstatus)
FROM orders
GROUP BY orderpriority, custkey
```

- 解決了使用 DISTINCT 的查詢的記憶體處理問題。為了避免在使用 DISTINCT 查詢時出現這個規模的查詢用盡了資源等錯誤訊息，請選擇低基數的 DISTINCT，或是減少查詢的資料大小。

- 不指定特定資料欄的 `SELECT COUNT(*)` 查詢透過僅保留計數而不使用資料列緩衝提高了效能並改進記憶體使用率。
- 採用了下列字串函數。
  - `translate(source, from, to)` – 傳回 `source` 字串，其中包含在 `from` 字串中的字元會替換成 `to` 字串中的對應字元。如果 `from` 字串包含重複項目，則只使用第一個項目。如果 `from` 字串中沒有 `source` 字元，則會複製 `source` 字元而不轉譯。如果檢索出 `from` 字串中的字元大於 `to` 字串的長度，則最後產生的字串會省略該字元。
  - `concat_ws(string0, array(varchar))` – 傳回串連的數列，使用 `string0` 作為分隔符號。如果 `string0` 為 `NULL`，則傳回值為 `NULL`。會略過數列中的任何 `NULL` 值。
- 修復了因嘗試存取 `struct` 中缺少的子欄位而發生查詢失敗的錯誤。查詢現在會在缺少子欄位時傳回 `NULL` 值。
- 修復了小數資料類型雜湊不一致的問題。
- 修復了分割區中資料欄過多而導致資源用盡的問題。

## 2021 年 11 月 17 日

2021-11-17 發布

[Amazon Athena](#) 現在支援分割區檢索以加速 [AWS Glue Data Catalog](#) 中已分割的資料表的查詢。

查詢已分割的資料表時，Athena 會檢索可用的資料表分割區，並篩選出與您的查詢相關的一小部分資料。隨著新資料和分割區的增加，處理分割區需要更長的時間，執行查詢的時間可能會增加。為了最佳化分割區的處理並提高對高度已分割的資料表的查詢效能，Athena 現已支援 [AWS Glue 分割區檢索](#)。

如需詳細資訊，請參閱 [AWS Glue 分割區索引和篩選](#)。

## 2021 年 11 月 16 日

2021-11-16 發布

全新且改良的 [Amazon Athena](#) 主控台現已在提供 [Athena 服務的 AWS 商業和 GovCloud 地區](#) 正式推出。Athena 的新版主控台可支援舊版主控台的所有功能，但具有更好用的新型介面，內含的新功能可改進查詢開發、分析資料和管理使用情況等體驗。您現在可以：

- 使用新版的分頁列來重新排列、導覽或關閉多個查詢索引標籤。
- 透過改進後的 SQL 和文字設定格式，更輕鬆地讀取和編輯查詢。
- 除了下載完整的結果集之外，還可將查詢結果複製到剪貼簿。

- 將查詢的歷史記錄、已儲存的查詢和工作群組加以排序，然後選擇要顯示或隱藏的資料欄。
- 使用簡化的界面，設定資料來源和工作群組所需的點選次數更少。
- 設定用於顯示查詢結果、查詢歷史記錄、換行等功能的偏好設定。
- 利用新的改良版鍵盤快速鍵和嵌入式產品文件提高您的工作效率。

隨著今天的宣佈，[新版主控台](#)現已成為預設設定。若要跟我們分享您的使用體驗，請選擇主控台左下角的意見回饋。

如有需要，您可以登入您的主控台 AWS 帳戶，選擇 Amazon Athena，然後從左側導覽面板取消選取新 Athena 體驗，以使用舊版主控台。

## 2021 年 11 月 12 日

2021-11-12 發布

您現已可用 Amazon Athena 對位於 AWS 帳戶 (非您自己所有) 的資料來源執行聯合查詢。直到今天，查詢此資料時，資料來源及其連接器必須使用與查詢資料的使用者相同 AWS 帳戶 的連接器。

若您是資料管理員，可透過與資料分析師的帳戶共用資料連接器來啟用跨帳戶的聯合查詢。若您是資料分析師，可將資料管理員與您共用的資料連接器加到您的帳戶中。對原始帳戶中連接器的組態變更會自動套用於共用連接器。

如需有關啟用跨帳戶聯合查詢的資訊，請參閱[啟用跨帳戶聯合查詢](#)。若要瞭解有關查詢聯合來源的資訊，請參閱[使用 Amazon Athena 聯合查詢](#) 和 [執行聯合查詢](#)。

## 2021 年 11 月 2 日

2021-11-02 發布

現在已可使用 Athena 中的 EXPLAIN ANALYZE 陳述式來查看 SQL 查詢的分散式執行計劃，以及每項操作的成本。

如需詳細資訊，請參閱 [在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)。

## 2021 年 10 月 29 日

2021-10-29 發布

Athena 發佈了 JDBC 2.0.25 和 ODBC 1.1.13 驅動程式，並宣佈改良過後的功能。

## JDBC 和 ODBC 驅動程式

發布了 Athena 的 JDBC 2.0.25 和 ODBC 1.1.13 驅動程式。這兩個驅動程式都支持瀏覽器 SAML 多重要素驗證，可設定此驗證法，使其用於任何 SAML 2.0 提供者。

JDBC 2.0.25 驅動程式包括以下變更：

- 支援瀏覽器 SAML 身分驗證。此驅動程式所含的瀏覽器 SAML 外掛程式可設定為用於任何 SAML 2.0 提供者。
- Support AWS Glue API 呼叫。您可使用 `GlueEndpointOverride` 參數來覆寫 AWS Glue 端點。
- `com.simba.athena.amazonaws` 類路徑已改為 `com.amazonaws`。

ODBC 1.1.13 驅動程式包括以下變更：

- 支援瀏覽器 SAML 身分驗證。此驅動程式所含的瀏覽器 SAML 外掛程式可設定為用於任何 SAML 2.0 提供者。有關如何將瀏覽器 SAML 外掛程式與 ODBC 驅動程式一起使用的範例，請參閱 [使用 ODBC、SAML 2.0 和 Okta 身分提供者設定單一登入](#)。
- 現已可在使用 ADFS、Azure AD 或瀏覽器 Azure AD 進行身分驗證時，設定角色工作階段的持續時間。

如需有關這些變更以及其他變更和下載新驅動程式、版本備註和文件的詳細資訊，請參閱 [使用 JDBC 連接到 Amazon Athena](#) 和 [使用 ODBC 連接到 Amazon Athena](#)。

## 功能和改進

Athena 發布了下列功能以及強化功能。

- 已推出新版最佳化規則，可避免在某些情況下重複掃描資料表。

## 2021 年 10 月 4 日

2021-10-04 發布

Athena 發布了下列功能以及強化功能。

- SQL 位移 – SELECT 陳述式現已支援 SQL OFFSET 子句。如需詳細資訊，請參閱 [SELECT](#)。
- CloudWatch 使用量度 — Athena 現在會在 AWS/Usage 命名空間中發佈 `ActiveQueryCount` 量度。如需詳細資訊，請參閱 [監控 Athena 用量指標](#)。

- 查詢規劃 – 已修正在極少數情況下可能導致查詢規劃逾時的錯誤。

## 2021 年 9 月 16 日

2021-09-16 發布

Athena 發布了下列新版功能以及強化功能。

### 功能

- 利用 `write_compression` 資料表屬性提高了對在 CTAS 中指定文字檔案和 JSON 壓縮時的支援。您還可在 CTAS 中指定用於 Parquet 和 ORC 格式的 `write_compression` 屬性。如需詳細資訊，請參閱 [CTAS 資料表屬性](#)。
- 現已支援編寫文字檔案和 JSON 檔案時使用 BZIP2 壓縮格式。如需有關 Athena 中壓縮格式的資訊，請參閱 [Athena 壓縮支援](#)。

### 改善項目

- 修復了身分資訊無法發送給 UDF Lambda 函數的錯誤。
- 修復了帶有分離篩選條件的述詞下推問題。
- 修復了小數類型的雜湊問題。
- 修復了不必要的統計數字收集問題。
- 刪除了不一致的錯誤訊息。
- 藉由在工作節點中套用動態分割區剪清除來改善廣播聯結效能。
- 針對聯合查詢：
  - 已更改組態以減少聯合查詢時發生 `CONSTRAINT_VIOLATION` 錯誤。

## 2021 年 9 月 15 日

2021-09-15 發布

您現已可使用新版的 Amazon Athena 主控台 (預覽版)。已推出了新版 Athena JDBC 驅動程式。

### Athena 主控台預覽版

您現在可以在任何可用 [Athena 的 AWS 區域 地方使用重新設計的 Amazon Athena 主控台 \(預覽版\)](#)。新版主控台支援現有主控台的所有功能，但其新型介面更加好用。

若要切換到新主控台，請登入您的主控台，AWS 帳戶 然後選擇 Amazon Athena。在 AWS 主控台導覽列中，選擇 [切換至新主控台]。若要恢復成預設的主控台，請取消選取左側導覽窗格中的 New Athena experience (新版 Athena 體驗)。

立即開始使用新版[主控台](#)。請選擇左下角的意見回饋，跟我們分享您的使用體驗。

## Athena JDBC 2.0.24 版驅動程式

Athena 宣佈推出 Athena 的 JDBC 2.0.24 版驅動程式。此版本更新了所有憑據提供者的代理支援。驅動程式現已支援 NonProxyHosts 連接屬性不支援的所有主機的代理身分驗證。

為了方便起見，此版本包括 JDBC 驅動程式的下載，無論是否使用 AWS SDK。此 JDBC 驅動程式版本讓您的專案可同時使用內嵌的 AWS 開發套件和 Athena JDBC 驅動程式。

如需詳細資訊並且要下載新的驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

## 2021 年 8 月 31 日

2021-08-31 發布

Athena 宣佈下列功能增強和錯誤修正。

- Athena 聯合增強功能 – Athena 已經增加對映射類型的支援，以及對複雜類型的更好支援，作為 [Athena Query Federation 軟體開發套件](#) 的一部分。此版本也包含一些記憶體增強功能和效能最佳化。
- 新的錯誤類別 – 引入了 USER 和 SYSTEM 錯誤訊息中的錯誤類別。這些類別可協助您區分可自行修正的錯誤 (USER)，以及可能需要 Athena 支援協助的錯誤 (SYSTEM)。
- 聯合查詢錯誤訊息 – 更新了聯合查詢相關錯誤的 USER\_ERROR 分類。
- JOIN — 修正 spill-to-disk 相關錯誤和記憶體問題，以增強效能並減少 JOIN 作業中的記憶體錯誤。

## 2021 年 8 月 12 日

2021-08-12 發布

發行了 Athena 的 ODBC 1.1.12 驅動程式。此版本修正了 SQLPrepare()、SQLGetInfo() 以及 EndpointOverride 的相關問題。

若要下載新的驅動程式、版本備註和文件，請參閱[使用 ODBC 連接到 Amazon Athena](#)。

## 2021 年 8 月 6 日

2021-08-06 發布

Amazon Athena 宣佈 Athena 及其[功能](#)可在亞太區域 (大阪) 區域使用。

此版本擴展了 Athena 在亞太區域的可用性，以包含亞太區域 (香港)、亞太區域 (孟買)、亞太區域 (大阪)、亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (新加坡)、亞太區域 (雪梨) 和亞太區域 (東京)。如需這些地區和其他區域 AWS 服務 可用的完整清單，請參閱 [AWS 區域 al 服務清單](#)。

## 2021 年 8 月 5 日

2021-08-05 發布

您可以使用 UNLOAD 陳述式，將 SELECT 查詢的輸出寫入 PARQUET、ORC、AVRO 和 JSON 格式。

如需詳細資訊，請參閱[UNLOAD](#)。

## 2021 年 7 月 30 日

2021-07-30 發布

Athena 宣佈下列功能增強和錯誤修正。

- 動態篩選和分割區剔除 – 改善項目可提高效能並降低某些查詢中的掃描資料量，如下列範例所示。

此範例假設 Table\_B 是未分割的資料表，其檔案大小加起來小於 20 MB。針對這樣的查詢，從 Table\_A 讀取的資料更少且查詢完成更快。

```
SELECT *
FROM Table_A
JOIN Table_B ON Table_A.date = Table_B.date
WHERE Table_B.column_A = "value"
```

- ORDER BY with LIMIT, DISTINCT with LIMIT – 針對後面接著 LIMIT 子句且使用 ORDER BY 或 DISTINCT 的查詢進行效能改善。
- S3 Glacier Deep Archive 檔案 – 當 Athena 查詢一個混合 [S3 Glacier Deep Archive 檔案](#) 和非 S3 Glacier 檔案的資料表時，Athena 現在會替您略過 S3 Glacier Deep Archive 檔案。先前，您需手動將這些檔案從查詢位置移動，否則查詢會失敗。如果您想要使用 Athena 查詢 S3 Glacier Deep Archive 儲存空間中的物件，則必須將其還原。如需詳細資訊，請參閱《Simple Storage Service (Amazon S3) 使用者指南》中的[還原已封存的物件](#)。

- 修正了 CTAS bucketed\_by [資料表屬性](#) 建立的空檔案未能正確加密的錯誤。

## 2021 年 7 月 21 日

2021-07-21 發布

透過 2021 年 7 月發行的 [Microsoft Power BI 桌面版](#)，您可以使用 Amazon Athena 的原生資料來源連接器來建立報告和儀表板。Amazon Athena 的連接器可作為 Power BI 中的標準連接器使用 [DirectQuery](#)，並可透過 [Power BI 閘道](#) 對大型資料集進行分析和內容重新整理。

因為連接器使用您現有的 ODBC 資料來源名稱 (DSN) 來連接至 Athena 並在上面執行查詢，所以它需要 Athena ODBC 驅動程式。若要下載最新的 ODBC 驅動程式，請參閱 [使用 ODBC 連接到 Amazon Athena](#)。

如需詳細資訊，請參閱 [使用 Amazon Athena Power BI 連接器](#)。

## 2021 年 7 月 16 日

2021-07-16 發布

Amazon Athena 已經更新了與 Apache Hudi 的整合。Hudi 是一個開源資料管理架構，用於簡化 Simple Storage Service (Amazon S3) 資料湖中的增量資料處理。更新後的整合讓您能夠使用 Athena 查詢透過 Amazon EMR、Apache Spark、Apache Hive 或其他兼容服務管理的 Hudi 0.8.0 資料表。此外，Athena 現在支援兩項其他功能：讀取時合併 (MoR) 資料表上的快照查詢，以及讀取引導資料表的支援。

Apache Hudi 提供記錄層級的資料處理，可協助您簡化變更資料擷取 (CDC) 管道的開發、符合歐盟《一般資料保護規範》(GDPR) 驅動的更新和刪除，以及更好地管理串流資料，這些資料來自需要資料插入和事件更新的感應器或裝置。0.8.0 版本可以更容易地將大型 Parquet 資料表遷移至 Hudi，而無需複製資料，因此您可以透過 Athena 查詢和分析它們。您可以使用 Athena 適用於快照查詢的新支援，以近乎即時的方式檢視串流資料表更新。

若要進一步了解如何搭配 Athena 使用 Hudi，請參閱 [使用 Athena 查詢 Apache Hudi 資料集](#)。

## 2021 年 7 月 8 日

2021-07-08 發布

發行了 Athena 的 ODBC 1.1.11 驅動程式。ODBC 驅動程式現在可以使用 JSON Web Token (JWT) 驗證連線。在 Linux 上，Workgroup (工作群組) 屬性的預設值已設定為 Primary (主要)。

如需詳細資訊並且要下載新的驅動程式、版本備註和文件，請參閱[使用 ODBC 連接到 Amazon Athena](#)。

## 2021 年 7 月 1 日

2021-07-01 發布

2021 年 7 月 1 日，預覽工作群組的特殊處理已結束。雖然 AmazonAthenaPreviewFunctionality 工作群組會保留其名稱，但是不再具有特殊狀態。您可以繼續使用 AmazonAthenaPreviewFunctionality 工作群組來檢視、修改、組織和執行查詢。不過，使用先前預覽版功能的查詢現在受標準 Athena 帳單條款與條件的約束。如需帳單資訊，請參閱[Amazon Athena 定價](#)。

## 2021 年 6 月 23 日

2021-06-23 發布

發行了 Athena 的 JDBC 2.0.23 和 ODBC 1.1.10 驅動程式。這兩種驅動程式都能提供改善的讀取效能，並支援 [EXPLAIN](#) 陳述式和 [參數化查詢](#)。

EXPLAIN 陳述式會顯示 SQL 查詢的邏輯或分散式執行計劃。透過參數化查詢，可以採用執行時提供的不同值多次使用相同的查詢。

JDBC 版本也會新增對 Active Directory Federation Services 2019 的支援，以及適用於 AWS STS 的自訂端點覆寫選項。ODBC 版本修正了 IAM 設定檔憑證的問題。

如需詳細資訊並且要下載新的驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)和[使用 ODBC 連接到 Amazon Athena](#)。

## 2021 年 5 月 12 日

2021-05-12 發布

您現在可以使用 Amazon Athena 從您自己的帳戶以外的帳戶註冊 AWS Glue 目錄。設定所需的 IAM 許可後 AWS Glue，您可以使用 Athena 執行跨帳戶查詢。

如需詳細資訊，請參閱[AWS Glue Data Catalog 從另一個帳戶註冊](#)和[跨帳戶存取 AWS Glue 資料目錄](#)。

## 2021 年 5 月 10 日

2021-05-10 發布

發行了 Athena 的 ODBC 驅動程式版本 1.1.9.1001。此版本修正了使用 Azure Active Directory (AD) 時所產生的 BrowserAzureAD 身分驗證類型問題。

若要下載新的驅動程式、版本備註和文件，請參閱[使用 ODBC 連接到 Amazon Athena](#)。

## 2021 年 5 月 5 日

2021-05-05 發布

您現在可以在聯合查詢中使用 Amazon Athena Vertica 連接器，從 Athena 查詢 Vertica 資料來源。例如，您可以在 Vertica 上的資料倉儲和 Simple Storage Service (Amazon S3) 中的資料湖上執行分析查詢。

若要部署 Athena Vertica 連接器，請造訪中的[AthenaVerticaConnector](#)頁面。AWS Serverless Application Repository

Amazon Athena Vertica 連接器透過 Lambda 環境變數公開數個組態選項。如需有關組態選項、參數、連接字串、部署和限制的詳細資訊，請參閱 [Amazon Athena Vertica 連接器](#)。

如需有關使用 Vertica 連接器的深入資訊，請參閱 AWS 大數據部落格中的[使用 Athena 聯合查詢軟體開發套件查詢 Amazon Athena 中的 Vertica 資料來源](#)。

## 2021 年 4 月 30 日

2021-04-30 發布

發行了 Athena 的驅動程式 JDBC 2.0.21 和 ODBC 1.1.9。這兩個版本都支援使用 Azure 作用中目錄 (AD) 和 SAML 驗證的 SAML 驗證。PingFederateJDBC 版本也支援參數化查詢。如需有關 Athena 中參數化查詢的資訊，請參閱[使用參數化查詢](#)。

若要下載新的驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)和[使用 ODBC 連接到 Amazon Athena](#)。

## 2021 年 4 月 29 日

2021-04-29 發布

Amazon Athena 宣佈在中國 (北京) 和中國 (寧夏) 區域提供 Athena 引擎版本 2。

如需有關 Athena 引擎版本 2 的資訊，請參閱[Athena 引擎版本 2](#)。

## 2021 年 4 月 26 日

2021-04-26 發布

Athena 引擎版本 2 中的視窗值函數現在支援 IGNORE NULLS 和 RESPECT NULLS。

如需詳細資訊，請參閱 Presto 文件中的[值函數](#)。

## 2021 年 4 月 21 日

2021-04-21 發布

Amazon Athena 宣佈在歐洲 (米蘭) 和非洲 (開普敦) 區域提供 Athena 引擎版本 2。

如需有關 Athena 引擎版本 2 的資訊，請參閱[Athena 引擎版本 2](#)。

## 2021 年 4 月 5 日

2021-04-05 發布

### EXPLAIN 陳述式

您現在可以在 Athena 中使用 EXPLAIN 陳述式檢視 SQL 查詢的執行計劃。

如需詳細資訊，請參閱[在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE](#)和[了解 Athena EXPLAIN 陳述式結果](#)。

### SageMaker SQL 查詢中的 Machine Learning 模型

Amazon 的機器學習模型推論現已 SageMaker 正式提供給 Amazon Athena 使用。在 SQL 查詢中使用機器學習模型來簡化複雜的任務，例如異常偵測、客群分析及時間序列預測，只要在 SQL 查詢中叫用函數即可。

如需詳細資訊，請參閱[搭配 Amazon Athena 使用機器學習 \(ML\)](#)。

### 使用者定義函數 (UDF)

使用者定義函數 (UDF) 現在已普遍適用於 Athena。使用 UDF 來利用在單一 SQL 查詢中處理記錄或記錄群組的自訂函數。

如需詳細資訊，請參閱[利用使用者定義函數來查詢](#)。

## 2021 年 3 月 30 日

2021-03-30 發布

Amazon Athena 宣佈在亞太區域 (香港) 和中東 (巴林) 區域中提供 Athena 引擎版本 2。

如需有關 Athena 引擎版本 2 的資訊，請參閱[Athena 引擎版本 2](#)。

## 2021 年 3 月 25 日

2021-03-25 發布

Amazon Athena 宣佈在歐洲 (斯德哥爾摩) 區域提供 Athena 引擎版本 2。

如需有關 Athena 引擎版本 2 的資訊，請參閱[Athena 引擎版本 2](#)。

## 2021 年 3 月 5 日

2021-03-05 發布

Amazon Athena 宣佈在加拿大 (中部)、歐洲 (法蘭克福) 和南美洲 (聖保羅) 區域中提供 Athena 引擎版本 2。

如需有關 Athena 引擎版本 2 的資訊，請參閱[Athena 引擎版本 2](#)。

## 2021 年 2 月 25 日

2021-02-25 發布

Amazon Athena 宣佈在亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (雪梨)、歐洲 (倫敦) 和歐洲 (巴黎) 區域全面提供 Athena 引擎版本 2。

如需有關 Athena 引擎版本 2 的資訊，請參閱[Athena 引擎版本 2](#)。

## 2020 年 Athena 版本備註

### 2020 年 12 月 16 日

2020-12-16 發布

Amazon Athena 宣布推出 Athena 引擎第 2 版、Athena 聯合查詢和其他 AWS PrivateLink 區域。

## Athena 引擎版本 2 和 Athena 聯合查詢

Amazon Athena 宣佈在亞太區域 (孟買)、亞太區域 (東京)、歐洲 (愛爾蘭) 和美國西部 (加利佛尼亞北部) 區域全面提供 Athena 引擎版本 2 和 Athena 聯合查詢。Athena 引擎版本 2 和聯合查詢已在美國東部 (維吉尼亞北部)、美國東部 (俄亥俄州) 和美國西部 (奧勒岡) 區域提供。

如需詳細資訊，請參閱[Athena 引擎版本 2](#)和[使用 Amazon Athena 聯合查詢](#)。

## AWS PrivateLink

AWS PrivateLink 現在，歐洲 (斯德哥爾摩) 地區支援 Athena。如需 Athena AWS PrivateLink 的相關資訊，請參閱[使用界面 VPC 端點連接至 Amazon Athena](#)。

## 2020 年 11 月 24 日

2020-11-24 發布

發行了 Athena 的驅動程式 JDBC 2.0.16 和 ODBC 1.1.6。這些版本在帳戶層級支援 Okta Verify 多重要素驗證 (MFA)。您也可以使用 Okta MFA 來設定 SMS 身分驗證和 Google Authenticator 身分驗證作為因素。

若要下載新的驅動程式、版本備註和文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)和[使用 ODBC 連接到 Amazon Athena](#)。

## 2020 年 11 月 11 日

2020-11-11 發布

Amazon Athena 宣佈在美國東部 (維吉尼亞北部)、美國東部 (俄亥俄) 和美國西部 (奧勒岡) 區域全面提供 Athena 引擎版本 2 和聯合查詢。

## Athena 引擎版本 2

Amazon Athena 宣佈在美國東部 (維吉尼亞北部)、美國東部 (俄亥俄) 和美國西部 (奧勒岡) 區域全面提供新的查詢引擎版本，即 Athena 引擎版本 2。

Athena 引擎版本 2 包含效能增強功能和新功能，例如 Parquet 格式資料的結構描述演變支援、其他地理空間函數、降低成本的讀取巢狀結構描述之支援，以及 JOIN 和 AGGREGATE 操作的效能增強功能。

- 如需有關改善功能、重大變更和錯誤修正的資訊，請參閱[Athena 引擎版本 2](#)。
- 如需有關如何升級的資訊，請參閱[變更 Athena 引擎版本](#)。

- 如需有關測試查詢的資訊，請參閱[在引擎版本升級之前測試查詢](#)。

## 聯合 SQL 查詢

現在，您可以在美國東部 (維吉尼亞北部)、美國東部 (俄亥俄) 和美國西部 (奧勒岡) 區域使用 Athena 的聯合查詢，不需要使用 AmazonAthenaPreviewFunctionality 工作群組。

使用聯合 SQL 查詢來針對關聯式、非關聯式、物件和自訂資料來源執行 SQL 查詢。透過聯合查詢，您可以提交單一 SQL 查詢，以掃描內部部署執行或在雲端託管的多個來源的資料。

要跨應用程式對資料執行分析極為複雜，而且耗費時間，原因如下：

- 分析所需的資料通常分佈在關聯式、金鑰值、文件、記憶體內、搜尋、圖形、物件、時間序列和分類帳資料儲存中。
- 為了分析這些來源中的資料，分析師建置了複雜的通道來擷取、轉換和載入至資料倉儲中，以便可以查詢資料。
- 若要從各種來源存取資料，需要學習新的程式設計語言和資料存取結構。

Athena 中的聯合 SQL 查詢允許使用者從資料所在的任何位置就地查詢，從而消除了這種複雜性。分析師可以使用熟悉的 SQL 建構子來 JOIN (聯結) 多個資料來源的資料以進行快速分析，以及將結果存放在 Simple Storage Service (Amazon S3) 中供後續使用。

### 資料來源連接器

若要處理聯合查詢，Athena 會使用在 [AWS Lambda](#) 上執行的 Athena 資料來源連接器。下列開源的、預先建置的連接器由 Athena 寫入和測試。使用它們在 Athena 中針對其對應的資料來源執行 SQL 查詢。

- [CloudWatch](#)
- [CloudWatch](#) 指標
- [DocumentDB](#)
- [DynamoDB](#)
- [OpenSearch](#)
- [HBase](#)
- [Neptune](#)
- [Redis](#)
- [Timestream](#)

- [TPC Benchmark DS \(TPC-DS\)](#)

### 自訂資料來源連接器

使用 [Athena Query Federation 軟體開發套件](#)，開發人員可以建立任何資料來源的連接器，讓 Athena 能夠針對該資料來源執行 SQL 查詢。Athena 查詢同盟連接器將聯合查詢的優點延伸到 AWS 提供的連接器之外。由於連接器在上執行 AWS Lambda，因此您不需要管理基礎結構或規劃擴充至尖峰需求。

### 後續步驟

- 若要進一步了解聯合查詢功能，請參閱[使用 Amazon Athena 聯合查詢](#)。
- 若要開始使用現有的連接器，請參閱[部署連接器並連接至資料來源](#)。
- 若要了解如何使用 Athena 查詢同盟 SDK 建立自己的資料來源連接器，請參閱上的[範例 Athena 連接器](#) GitHub。

## 2020 年 10 月 22 日

2020-10-22 發布

您現在可以打電話給 Athena AWS Step Functions。AWS Step Functions 可以 AWS 服務 直接使用 [Amazon 國家語言](#) 控制某些。您可以搭配 Athena 使用 Step Functions 來啟動和停止查詢執行、取得查詢結果、執行隨機操作或排程的資料查詢，以及從 Simple Storage Service (Amazon S3) 中的資料湖擷取結果。

如需詳細資訊，請參閱《AWS Step Functions 開發人員指南》中的[使用 Step Functions 呼叫 Athena](#)。

## 2020 年 7 月 29 日

2020-07-29 發布

發行了 JDBC 驅動程式版本 2.0.13。此版本支援使用多個[在 Athena 註冊的資料目錄](#)、用於身分驗證的 Okta 服務以及連線至 VPC 端點。

若要下載並使用新版驅動程式，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

## 2020 年 7 月 9 日

2020-07-09 發布

Amazon Athena 新增查詢壓縮的 Hudi 資料集的支援，並新增 AWS CloudFormation `AWS::Athena::DataCatalog` 資源以建立、更新或刪除您在 Athena 註冊的資料目錄。

## 查詢 Apache Hudi 資料集

Apache Hudi 是一個開源資料管理架構，可簡化增量資料的處理。Amazon Athena 現在支援在以 Simple Storage Service (Amazon S3) 為基礎的資料湖中查詢 Apache Hudi 資料集的讀取最佳化檢視。

如需詳細資訊，請參閱 [使用 Athena 查詢 Apache Hudi 資料集](#)。

## AWS CloudFormation 資料目錄資源

若要使用 Amazon Athena 的 [聯合查詢功能](#) 查詢任何資料來源，您必須先在 Athena 中註冊資料目錄。您現在可以使用 AWS CloudFormation `AWS::Athena::DataCatalog` 資源來建立、更新或刪除在 Athena 註冊的資料目錄。

若要取得更多資訊，請參閱《AWS CloudFormation 使用指南》[AWS::Athena::DataCatalog](#) 中的。

2020 年 6 月 1 日

2020-06-01 發布

## 使用 Apache Hive 中繼存放區作為中繼目錄搭配 Amazon Athena

除了使用 Athena 搭配 AWS Glue Data Catalog 之外，您現在還可以將 Apache 連接至一或多個 Apache Hive 中繼存放區。

若要連接至自我託管的 Hive 中繼存放區，您需要一個 Athena Hive 中繼存放區連接器。Athena 提供 [參考實作](#) 連接器供您使用。連接器在您的帳戶中會作為 AWS Lambda 函數執行。

如需詳細資訊，請參閱 [將 Athena 資料連接器用於外部 Hive 中繼存放區](#)。

2020 年 5 月 21 日

2020-05-21 發布

Amazon Athena 新增了對分割區投影的支援。使用分割區投影可加速高度分割資料表的查詢處理，並自動化分割區管理。如需詳細資訊，請參閱 [使用 Amazon Athena 進行分割區投影](#)。

2020 年 4 月 1 日

2020-04-01 發布

除了美國東部 (維吉尼亞北部) 區域，Amazon Athena [聯合查詢](#)、[使用者定義函數 \(UDF\)](#)、[機器學習推論](#)以及[外部 Hive 中繼存放區](#)功能現已在亞太區域 (孟買)、歐洲 (愛爾蘭) 和美國西部 (奧勒岡) 區域以預覽模式提供。

## 2020 年 3 月 11 日

2020-03-11 發布

Amazon Athena 現在發佈 Amazon EventBridge 事件，以進行查詢狀態 例如，當查詢在狀態之間切換至終端機狀態 (例如「成功」或「已取消-」) 時，Athena 會將查詢狀態變更事件發佈至 EventBridge。該事件包含有關查詢狀態轉換的資訊。如需詳細資訊，請參閱 [使用 Amazon EventBridge 事件監控 Athena 查](#)。

## 2020 年 3 月 6 日

2020-03-06 發布

您現在可以使用 AWS CloudFormation `AWS::Athena::WorkGroup` 資源建立和更新 Amazon Athena 工作群組。若要取得更多資訊，請參閱《AWS CloudFormation 使用指南》[AWS::Athena::WorkGroup](#) 中的。

## 2019 年 Athena 版本備註

### 2019 年 11 月 26 日

2019-12-17 發布

Amazon Athena 新增了對以下項目的支援：跨關聯式、非關聯式、物件和自訂資料來源執行 SQL 查詢、在 SQL 查詢中叫用機器學習模型、使用者定義函數 (UDF) (預覽)、使用 Apache Hive 中繼存放區作為 Amazon Athena 的中繼資料目錄 (預覽)，以及另外四個查詢相關的指標。

### 聯合 SQL 查詢

使用聯合 SQL 查詢來針對關聯式、非關聯式、物件和自訂資料來源執行 SQL 查詢。

您現在可以使用 Athena 的聯合查詢來掃描儲存在關聯式、非關聯式、物件和自訂資料來源中的資料。透過聯合查詢，您可以提交單一 SQL 查詢，以掃描內部部署執行或在雲端託管的多個來源的資料。

要跨應用程式對資料執行分析極為複雜，而且耗費時間，原因如下：

- 分析所需的資料通常分佈在關聯式、金鑰值、文件、記憶體內、搜尋、圖形、物件、時間序列和分類帳資料儲存中。
- 為了分析這些來源中的資料，分析師建置了複雜的通道來擷取、轉換和載入至資料倉儲中，以便可以查詢資料。
- 若要從各種來源存取資料，需要學習新的程式設計語言和資料存取結構。

Athena 中的聯合 SQL 查詢允許使用者從資料所在的任何位置就地查詢，從而消除了這種複雜性。分析師可以使用熟悉的 SQL 建構子來 JOIN (聯結) 多個資料來源的資料以進行快速分析，以及將結果存放在 Simple Storage Service (Amazon S3) 中供後續使用。

### 資料來源連接器

Athena 使用在 [AWS Lambda](#) 上執行的 Athena 資料來源連接器來處理聯合查詢。使用這些開放原始碼資料來源連接器，在 [Amazon DynamoDB](#)、[Apache HBase](#)、[Amazon 文件資料庫](#)、[Amazon CloudWatch](#)、[亞馬遜 CloudWatch 指標](#) 以及與 [JDBC](#) 相容的關聯式資料庫 (例如 MySQL) 以及 Apache 2.0 授權下的 PostgreSQL 執行聯合 SQL 查詢。

### 自訂資料來源連接器

使用 [Athena Query Federation 軟體開發套件](#)，開發人員可以建立任何資料來源的連接器，讓 Athena 能夠針對該資料來源執行 SQL 查詢。Athena 查詢同盟連接器將聯合查詢的優點延伸到 AWS 提供的連接器之外。由於連接器在上執行 AWS Lambda，因此您不需要管理基礎結構或規劃擴充至尖峰需求。

### 預覽可用性

Athena 聯合查詢在美國東部 (維吉尼亞北部) 區域以預覽模式提供。

### 後續步驟

- 若要開始預覽，請遵循 [Athena 預覽功能常見問答集](#) 中的指示。
- 若要進一步了解聯合查詢功能，請參閱 [使用 Amazon Athena 聯合查詢 \(預覽\)](#)。
- 若要開始使用現有的連接器，請參閱 [部署連接器並連接至資料來源](#)。
- 若要了解如何使用 Athena 查詢同盟 SDK 建立自己的資料來源連接器，請參閱上的 [範例 Athena 連接器](#) GitHub。

## 在 SQL 查詢中呼叫機器學習模型

您現在可以叫用機器學習模型，直接從您的 Athena 查詢進行推論。在 SQL 查詢使用機器學習，只要在 SQL 查詢叫用函數就能執行複雜的任務，例如異常偵測、客群分析及銷售預測。

## ML 模型

您可以使用 [Amazon](#) 提供的十多種內建機器學習演算法 SageMaker、訓練您自己的模型，或尋找並訂閱 [Amazon SageMaker 託管服務AWS Marketplace](#) 和部署的模型套件。不需要額外的設定。您可以從 Athena 主控台、[Athena API](#)，以及 Athena 的 [預覽 JDBC 驅動程式](#)，在 SQL 查詢中叫用這些機器學習 (ML) 模型。

### 預覽可用性

Athena 的機器學習 (ML) 功能目前已在美國東部 (維吉尼亞北部) 區域以預覽模式提供。

### 後續步驟

- 若要開始預覽，請遵循 [Athena 預覽功能常見問答集](#) 中的指示。
- 若要進一步了解機器學習功能，請參閱 [搭配 Amazon Athena 使用機器學習 \(ML\) \(預覽\)](#)。

## 使用者定義函數 (UDF) (預覽)

您現在可以撰寫自訂純量函數，並在 Athena 查詢中叫用它們。您可以使用 [Athena Query Federation 軟體開發套件](#) 以 Java 編寫 UDF。在提交至 Athena 的 SQL 查詢中使用 UDF 時，會在 [AWS Lambda](#) 上叫用並執行 UDF。UDF 可用於 SQL 查詢的 SELECT 與 FILTER 子句。您可以在相同查詢中叫用多個 UDF。

### 預覽可用性

Athena UDF 功能在美國東部 (維吉尼亞北部) 區域以預覽模式提供。

### 後續步驟

- 若要開始預覽，請遵循 [Athena 預覽功能常見問答集](#) 中的指示。
- 若要進一步了解，請參閱 [利用使用者定義函數來查詢 \(預覽\)](#)。
- 例如 UDF 實作，請參閱上的 [Amazon Athena UDF 連接器](#)。GitHub
- 若要了解如何使用 Athena Query Federation 軟體開發套件撰寫您的函數，請參閱 [使用 Lambda 建立和部署 UDF](#)。

## 使用 Apache Hive 中繼存放區作為中繼目錄搭配 Amazon Athena (預覽)

除了使用 Athena 搭配 AWS Glue Data Catalog 之外，您現在還可以將 Apache 連接至一或多個 Apache Hive 中繼存放區。

## 中繼存放區連接器

若要連接至自我託管的 Hive 中繼存放區，您需要一個 Athena Hive 中繼存放區連接器。Athena 提供[參考實作](#)連接器供您使用。連接器會以您帳戶中的 AWS Lambda 函式執行。如需詳細資訊，請參閱[針對外部 Hive 中繼存放區使用 Athena 資料連接器 \(預覽\)](#)。

### 預覽可用性

Hive 中繼存放區功能在美國東部 (維吉尼亞北部) 區域以預覽模式提供。

### 後續步驟

- 若要開始預覽，請遵循 [Athena 預覽功能常見問答集](#) 中的指示。
- 若要進一步了解此功能，請造訪我們的[針對外部 Hive 中繼存放區使用 Athena 資料連接器 \(預覽\)](#)。

## 新的查詢相關指標

Athena 現在發布可協助您了解 [Amazon Athena](#) 效能的其他查詢指標。[Athena 向 Amazon 發布與查詢相關的指標](#)。CloudWatch 在此版本中，Athena 發布下列額外的查詢指標：

- Query Planning Time (查詢規劃時間) – 規劃查詢所花費的時間。這包括從資料來源擷取資料表分割區所花費的時間。
- Query Queuing Time (查詢佇列時間) – 查詢在佇列中等待資源的時間。
- Service Processing Time (服務處理時間) – 查詢引擎完成處理後寫入結果所需的時間。
- Total Execution Time (總執行期) – Athena 執行查詢所花費的時間。

若要使用這些新的查詢指標，您可以建立自訂儀表板、在中設定指標的警示和觸發器 CloudWatch，或直接從 Athena 主控台使用預先填入的儀表板。

### 後續步驟

如需詳細資訊，請參閱[使用 CloudWatch 指標監控 Athena 查詢](#)。

## 2019 年 11 月 12 日

2019-12-17 發布

Amazon Athena 現已在中東 (巴林) 區域提供。

## 2019 年 11 月 8 日

2019-12-17 發布

Amazon Athena 現已在美國西部 (加利佛尼亞北部) 區域和歐洲 (巴黎) 區域提供。

## 2019 年 10 月 8 日

2019-12-17 發布

[Amazon Athena](#) 現在可讓您透過 Virtual Private Cloud (VPC) 中的介面 VPC 端點直接連接至 Athena。使用這項功能，您可以安全地將查詢提交到 Athena，無需在 VPC 中使用網際網路閘道。

若要建立連線至 Athena 的介面 VPC 端點，您可以使用 AWS Management Console 或 AWS Command Line Interface (AWS CLI)。如需建立介面端點的詳細資訊，請參閱[建立介面端點](#)。

當您使用介面虛擬私人雲端端點時，VPC 和 Athena API 之間的通訊是安全的，並保持在網路中 AWS。使用此功能無需支付額外的 Athena 費用。必須支付介面 VPC 端點[費用](#)。

若要進一步了解此功能，請參閱[使用介面 VPC 端點連接至 Amazon Athena](#)。

## 2019 年 9 月 19 日

2019-12-17 發布

Amazon Athena 新增了對使用 INSERT INTO 陳述式將新資料插入現有資料表的支援。您可以根據在來源表格上執行的 SELECT 查詢陳述式，或根據提供作為查詢陳述式一部分的一組值，將新資料行插入目的地表格。支援的資料格式包含 Avro、JSON、ORC、Parquet 和文字檔案。

INSERT INTO 陳述式也能協助您簡化 ETL 程序。例如，您可以在單一查詢中使用 INSERT INTO 從 JSON 格式的來源表格選取資料，再將資料寫入 Parquet 格式的目的地表格。

INSERT INTO 陳述式會根據 SELECT 階段中掃描的位元組數收費，類似於 Athena 針對 SELECT 查詢的收費方式。如需詳細資訊，請參閱[Amazon Athena 定價](#)。

如需使用 INSERT INTO 的詳細資訊 (包括支援的格式 SerDes 和範例)，請參閱 Athena 使用手冊中的[插入](#)。

## 2019 年 9 月 12 日

2019-12-17 發布

Amazon Athena 現已在亞太區域 (香港) 區域提供。

## 2019 年 8 月 16 日

2019-12-17 發布

[Amazon Athena](#) 新增了對在 Simple Storage Service (Amazon S3) 申請者付款儲存貯體中查詢資料的支援。

當 Simple Storage Service (Amazon S3) 儲存貯體設定為 Requester Pays (申請者付款) 時，請求者 (而非儲存貯體擁有者) 會支付 Simple Storage Service (Amazon S3) 請求與資料傳輸的費用。在 Athena 中，工作群組管理員現在可以設定工作群組設定，以允許工作群組成員查詢 S3 申請者付款儲存貯體。

如需有關如何為您的工作群組設定 Requester Pays (申請者付款) 設定的資訊，請參閱《Amazon Athena 使用者指南》中的[建立工作群組](#)。如需有關申請者付款儲存貯體的詳細資訊，請參閱《Amazon Simple Storage Service 開發人員指南》中的[申請者付款儲存貯體](#)。

## 2019 年 8 月 9 日

2019-12-17 發布

Amazon Athena 現在支援強制執行 [AWS Lake Formation](#) 政策，這些政策針對存放在 Simple Storage Service (Amazon S3) 中的資料在新的或現有的資料庫、資料表和 [AWS Glue Data Catalog](#) 中定義的資料欄進行精細存取控制。

您可以在下列項目中使用此功能 AWS 區域：美國東部 (俄亥俄)、美國東部 (維吉尼亞北部)、美國西部 (奧勒岡)、亞太區域 (東京) 和歐洲 (愛爾蘭)。使用此功能無須額外付費。

如需使用此功能的詳細資訊，請參閱 [使用 Athena 查詢向 AWS Lake Formation 註冊的資料](#) 一文。如需有關 AWS Lake Formation 的詳細資訊，請參閱 [《AWS Lake Formation》](#)。

## 2019 年 6 月 26 日

Amazon Athena 現在已在歐洲 (斯德哥爾摩) 區域提供。如需支援區域的清單，請參閱 [AWS 區域 和節點](#)。

## 2019 年 5 月 24 日

2019-05-24 發布

Amazon Athena 現在可在 AWS GovCloud (美國東部) 和 AWS GovCloud (美國西部) 區域使用。如需支援區域的清單，請參閱 [AWS 區域 和節點](#)。

## 2019 年 3 月 5 日

2019-03-05 發布

Amazon Athena 現已在加拿大 (中部) 區域提供。如需支援區域的清單，請參閱 [AWS 區域 和節點](#)。發布了支援 Athena 工作群組的新版 ODBC 驅動程式。如需詳細資訊，請參閱 [ODBC 驅動程式版本備註](#)。

若要下載 ODBC 驅動程式 1.0.5 版及其文件，請參閱 [使用 ODBC 連接到 Amazon Athena](#)。如需此版本的相關資訊，請參閱 [ODBC 驅動程式版本備註](#)。

若要使用工作群組搭配 ODBC 驅動程式，請在連線字串中設定新的連線屬性 Workgroup，如下所示：

```
Driver=Simba Athena ODBC
Driver;AwsRegion=[Region];S3OutputLocation=[S3Path];AuthenticationType=IAM
Credentials;UID=[YourAccessKey];PWD=[YourSecretKey];Workgroup=[WorkgroupName]
```

如需詳細資訊，請在 [ODBC 驅動程式安裝和組態指南 1.0.5 版](#) 中搜尋「工作群組」。當您在工作群組上使用標籤時，ODBC 驅動程式連線字串沒有變更。若要使用標籤，請升級到最新版本的 ODBC 驅動程式 (目前的版本)。

此驅動程式版本可讓您使用 [Athena API 工作群組動作](#) 建立和管理工作群組，並使用 [Athena API 標籤動作](#) 在工作群組上新增、列出或移除標籤。開始之前，請確定您在 IAM 中有資源層級許可，而能夠對工作群組和標籤執行動作。

如需詳細資訊，請參閱：

- [使用工作群組來執行查詢](#) 與 [工作群組政策範例](#)。
- [標記 Athena 資源](#) 與 [標籤型 IAM 存取控制政策](#)。

如果您使用 JDBC 驅動程式或 AWS SDK，請升級至最新版本的驅動程式和 SDK，這兩個版本都已包含對 Athena 中工作群組和標籤的支援。如需詳細資訊，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。

## 2019 年 2 月 22 日

2019-02-22 發布

對 Amazon Athena 中的工作群組新增了標籤支援。每個標記皆包含由您定義的金鑰和值。當您標記工作群組時，您可以指派自訂中繼資料給它。您可以使 AWS 用標籤 [最佳實踐](#)，將標籤加入至工作群組以

協助對其進行分類。您可以使用標籤來限制存取工作群組和追蹤成本。例如，為每個成本中心建立工作群組。然後，您可以將標籤新增到工作群組，以追蹤每個成本中心的 Athena 花費。如需詳細資訊，請參閱《AWS Billing and Cost Management 使用者指南》中的[使用標籤計費](#)。

您可以使用 Athena 主控台或 API 操作來處理標籤。如需詳細資訊，請參閱[標記 Athena 資源](#)。

在 Athena 主控台中，您可以將一或多個標籤新增到每個工作群組，並依標籤來搜尋。工作群組在 Athena 中是 IAM 控制的資源。在 IAM 中，您可以限制誰可以在您建立的工作群組上新增、移除或列出標籤。您也可以使用具有選用標籤參數的 CreateWorkGroup API 操作，將一或多個標籤新增到工作群組。若要新增、移除或列出標籤，請使用 TagResource、UntagResource 和 ListTagsForResource。如需詳細資訊，請參閱[使用標籤操作](#)。

若要允許使用者在建立工作群組時新增標籤，請確保將 TagResource 和 CreateWorkGroup API 動作的 IAM 許可給予每個使用者。如需詳細資訊和範例，請參閱[標籤型 IAM 存取控制政策](#)。

當您在工作群組上使用標籤時，JDBC 驅動程式沒有變更。如果您建立新的工作群組並使用 JDBC 驅動程式或 AWS SDK，請升級至最新版本的驅動程式和 SDK。如需相關資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

## 2019 年 2 月 18 日

2019-02-18 發布

新增能夠在工作群組中執行查詢來控制查詢成本。如需相關資訊，請參閱[使用工作群組來控制查詢存取和成本](#)。改善 Athena 中 SerDe 使用的 JSON OpenX，修正 Athena 未忽略轉換至 GLACIER 儲存區類別的物件的問題，並新增查詢 Network Load Balancer 記錄的範例。

進行下列變更：

- 新增支援工作群組。使用工作群組來區隔使用者、團隊、應用程式或工作負載，以及對每個查詢或整個工作群組可以處理的資料量設定限制。因為工作群組扮演 IAM 資源的角色，所以您可以使用資源層級許可來控制對特定工作群組的存取。您也可以 Amazon 中檢視查詢相關指標 CloudWatch、透過設定掃描的資料量限制來控制查詢成本、建立閾值，以及在超出這些閾值時觸發動作 (例如 Amazon SNS 警示)。如需詳細資訊，請參閱[使用工作群組來執行查詢](#)和[使用指標和事件控制成本並 CloudWatch 監控查詢](#)。

工作群組是 IAM 資源。如需 IAM 中與工作群組相關的動作、資源和條件的完整清單，請參閱《服務授權參考》中的[適用於 Amazon Athena 的動作、資源及條件索引鍵](#)。建立新的工作群組之前，請確保使用[工作群組 IAM 政策](#)和[AWS 受管理的策略：AmazonAthenaFullAccess](#)。

您可以在主控台、透過[工作群組 API 操作](#)或透過 JDBC 驅動程式來開始使用工作群組。如需高階程序，請參閱[設定工作群組](#)。若要下載具有工作群組支援的 JDBC 驅動程式，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

如果您使用工作群組搭配 JDBC 驅動程式，您必須在連線字串中使用 Workgroup 組態參數來設定工作群組名稱，如下列範例所示：

```
jdbc:awsathena://AwsRegion=<AWSREGION>;UID=<ACCESSKEY>;
PWD=<SECRETKEY>;S3outputLocation=s3://DOC-EXAMPLE-BUCKET/<athena-
output>-<AWSREGION>;
Workgroup=<WORKGROUPNAME>;
```

您對驅動程式執行 SQL 陳述式或進行 JDBC API 呼叫的方式沒有變更。驅動程式會將工作群組名稱傳遞給 Athena。

如需工作群組所造成差異的相關資訊，請參閱[Athena 工作群組 API](#)和[工作群組疑難排解](#)。

- 改進了在 Athena SerDe 使用的 JSON OpenX。改善功能包括 (但不限於) 下列各項：
  - 支援 ConvertDotsInJsonKeysToUnderscores 屬性。當設定為時 TRUE，它允許 SerDe 用底線取代關鍵名稱中的點。例如，如果 JSON 資料集包含名為 "a.b" 的索引鍵，您可以使用此屬性在 Athena 中將資料欄名稱定義為 "a\_b"。預設值為 FALSE。在預設情況下，Athena 在資料欄名稱中不允許點。
  - 支援 case.insensitive 屬性。在預設情況下，Athena 規定 JSON 資料集裡的所有索引鍵必須使用小寫。使用 WITH SERDE PROPERTIES ("case.insensitive"= FALSE;) 可讓您在資料中使用區分大小寫的索引鍵名稱。預設值為 TRUE。設定為時 TRUE，會 SerDe 將所有大寫欄轉換為小寫。

如需詳細資訊，請參閱[OpenX SerDe](#)。

- 修正了 Athena 在處理依 Simple Storage Service (Amazon S3) 生命週期政策而存檔到 Glacier 的 Simple Storage Service (Amazon S3) 物件時傳回 "access denied" 錯誤訊息這個問題。修正此問題後，Athena 會忽略轉換到 GLACIER 儲存類別的物件。Athena 不支援從 GLACIER 儲存類別來查詢資料。

如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[the section called "Athena 資料表和 Amazon S3 資料的要求"](#)和[轉換為 GLACIER 儲存類別 \(物件存檔\)](#)。

- 新增了如何查詢 Network Load Balancer 存取日誌的範例，這些日誌接收 Transport Layer Security (TLS) 請求的相關資訊。如需詳細資訊，請參閱[the section called "Network Load Balancer"](#)。

## 2018 年 Athena 版本備註

### 2018 年 11 月 20 日

2018-11-20 發布

發行了新版本的 JDBC 和 ODBC 驅動程式，支援透過 AD FS 和 SAML 2.0 (安全性聲明標記語言 2.0) 來聯合存取 Athena API。如需詳細資訊，請參閱 [JDBC 驅動程式版本備註](#) 和 [ODBC 驅動程式版本備註](#)。

在此版本中，支援 Active Directory Federation Service (AD FS 3.0) 聯合存取 Athena。透過支援 SAML 2.0 的 JDBC 或 ODBC 驅動程式版本建立存取權。如需有關設定聯合存取 Athena API 的資訊，請參閱 [the section called “啟用對 Athena API 的聯合存取”](#)。

若要下載 JDBC 驅動程式 2.0.6 版及其文件，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。如需此版本的詳細資訊，請參閱 [JDBC 驅動程式版本備註](#)。

若要下載 ODBC 驅動程式 1.0.4 版及其文件，請參閱 [使用 ODBC 連接到 Amazon Athena](#)。如需此版本的詳細資訊，請參閱 [ODBC 驅動程式版本備註](#)。

如需有關中的 SAML 2.0 支援的詳細資訊 AWS，請參閱《IAM 使用者指南》中的 [關於 SAML 2.0 同盟](#)。

### 2018 年 10 月 15 日

2018-10-15 發布

如果您已升級至 AWS Glue Data Catalog，則有兩個新功能可提供以下支援：

- 資料目錄中繼資料的加密。如果您選擇加密資料目錄中的中繼資料，您必須將特定的政策新增至 Athena。如需詳細資訊，請參閱 [存取 AWS Glue Data Catalog 中的加密中繼資料](#)。
- 存取中資源的 AWS Glue Data Catalog 精細權限。您現在可以定義身分型 (IAM) 政策，以限制或允許從 Athena 中使用的資料目錄存取特定的資料庫和資料表。如需詳細資訊，請參閱 [對於中的資料庫和資料表的精細存取 AWS Glue Data Catalog](#)。

#### Note

資料位於 Amazon S3 儲存貯體中，而其存取權取決於 [從 Athena 訪問 Amazon S3](#)。若要存取資料庫和資料表中的資料，請繼續對存放資料的 Simple Storage Service (Amazon S3) 儲存貯體使用存取控制政策。

## 2018 年 10 月 10 日

2018-10-10 發布

Athena 支援 CREATE TABLE AS SELECT，它會根據 SELECT 查詢陳述式的結果建立資料表。如需詳細資訊，請參閱[從查詢結果建立資料表 \(CTAS\)](#)。

建立 CTAS 查詢之前，請務必在 Athena 文件中了解其行為。其中包含有關將查詢結果儲存在 Simple Storage Service (Amazon S3) 中的位置、支援存放 CTAS 查詢結果的格式清單、您可以建立的分割區數目，以及支援的壓縮格式等資訊。如需詳細資訊，請參閱[CTAS 查詢的考量事項與限制](#)。

使用 CTAS 查詢來：

- 在一個步驟中[從查詢結果建立資料表](#)。
- 使用範例在 [Athena 主控台中建立 CTAS 查詢](#)。如需語法的相關資訊，請參閱 [CREATE TABLE AS](#)。
- 將查詢結果轉換為其他儲存格式，例如 PARQUET、ORC、AVRO、JSON 和 TEXTFILE。如需詳細資訊，請參閱 [CTAS 查詢的考量事項與限制](#) 及 [單欄式儲存格式](#)。

## 2018 年 9 月 6 日

2018-09-06 發布

發行新版本的 ODBC 驅動程式 (1.0.3 版)。新版本的 ODBC 驅動程式預設會串流結果，而不會翻頁瀏覽，這樣可讓商業智慧工具更快擷取大型資料集。此版本還包括改善功能、錯誤修正，以及使用 SSL 搭配代理伺服器的更新文件。如需詳細資訊，請參閱驅動程式的[版本備註](#)。

若要下載 ODBC 驅動程式 1.0.3 版及其文件，請參閱[使用 ODBC 連接到 Amazon Athena](#)。

串流結果功能適用於此新版本的 ODBC 驅動程式。也適用於 JDBC 驅動程式。如需串流結果的相關資訊，請參閱《[ODBC 驅動程式安裝和設定指南](#)》，然後搜尋 UseResultsetStreaming。

ODBC 驅動程式 1.0.3 版直接取代舊版的驅動程式。建議您遷移到最新的驅動程式。

### Important

若要使用 ODBC 驅動程式 1.0.3 版，請依照以下要求：

- 連接埠 444 保持開放給傳出流量。

- 在 Athena 的政策清單中新增 `athena:GetQueryResultsStream` 政策動作。此政策動作不會直接透過 API 公開，只能用於 ODBC 和 JDBC 驅動程式，做為串流結果支援的一部分。如需政策範例，請參閱 [AWS 受管理的策略：AWSQuicksightAthenaAccess](#)。

## 2018 年 8 月 23 日

2018-08-23 發布

增加支援這些 DDL 相關功能並修正幾個錯誤，如下所示：

- 對於 Parquet 的資料，增加支援 BINARY 和 DATE 資料類型，對於 Avro 的資料，增加支援 DATE 和 TIMESTAMP 資料類型。
- 在 DDL 查詢中增加支援 INT 和 DOUBLE。INTEGER 是 INT 的別名，DOUBLE PRECISION 是 DOUBLE 的別名。
- 已改進 DROP DATABASE 和 DROP TABLE 查詢的效能。
- 當資料儲存貯體是空的時，不再於 Simple Storage Service (Amazon S3) 中建立 `_$folder$` 物件。
- 修正在未提供分割區值時 ALTER TABLE ADD PARTITION 擲出錯誤的問題。
- 修正在陳述式中指定合格名稱之後 DROP TABLE 檢查分割區時忽略資料庫名稱的問題。

如需有關 Athena 中支援的資料類型的詳細資訊，請參閱 [Amazon Athena 中的資料類型](#)。

如需有關 Athena、JDBC 驅動程式和 Java 資料類型之間支援之資料類型映射的資訊，請參閱《JDBC 驅動程式安裝和設定指南》[https://downloads.athena.us-east-1.amazonaws.com/drivers/JDBC/SimbaAthenaJDBC\\_2.0.5/docs/Simba+Athena+JDBC+Driver+Install+and+Configuration+Guide.pdf](https://downloads.athena.us-east-1.amazonaws.com/drivers/JDBC/SimbaAthenaJDBC_2.0.5/docs/Simba+Athena+JDBC+Driver+Install+and+Configuration+Guide.pdf) 中的「資料類型」一節。

## 2018 年 8 月 16 日

2018-08-16 發布

發行 JDBC 驅動程式 2.0.5 版。新版本的 JDBC 驅動程式預設會串流結果，而不會翻頁瀏覽，這樣可讓商業智慧工具更快擷取大型資料集。相較於舊版的 JDBC 驅動程式，有以下效能改良：

- 在擷取不超過 10K 個資料行時，效能增加大約 2 倍。

- 在擷取超過 10K 個資料行時，效能增加大約 5-6 倍。

串流結果功能僅適用於 JDBC 驅動程式。不適用於 ODBC 驅動程式。不能搭配 Athena API 一起使用。如需有關串流結果的資訊，請參閱《[JDBC 驅動程式安裝和組態指南](#)》，然後搜尋 UseResultSetStreaming。

若要下載 JDBC 驅動程式 2.0.5 版及其文件，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

JDBC 驅動程式 2.0.5 版直接取代舊版的驅動程式 (2.0.2)。為確保您可以使用 JDBC 驅動程式 2.0.5 版，請將 athena:GetQueryResultsStream 政策動作新增至 Athena 的政策清單。此政策動作不會直接透過 API 公開，只能用於 JDBC 驅動程式，做為串流結果支援的一部分。如需政策範例，請參閱 [AWS 受管理的策略：AWSQuicksightAthenaAccess](#)。如需從驅動程式 2.0.2 版遷移至 2.0.5 版的詳細資訊，請參閱 [JDBC 驅動程式遷移指南](#)。

如果您是從 1.x 驅動程式遷移到 2.x 驅動程式，您需要將現有的設定遷移到新的設定。強烈建議您遷移到最新版的驅動程式。如需詳細資訊，請參閱 [JDBC 驅動程式遷移指南](#)。

## 2018 年 8 月 7 日

2018-08-07 發布

您現在可以將 Amazon Virtual Private Cloud 流程日誌以 GZIP 格式直接存放在 Simple Storage Service (Amazon S3) 中，然後在 Athena 中查詢它們。如需相關資訊，請參閱[查詢 Amazon VPC 流程日誌](#)和 [Amazon VPC Flow Logs 現在可以傳遞至 S3](#)。

## 2018 年 6 月 5 日

2018-06-05 發布

主題

- [支援檢視](#)
- [改善和更新錯誤訊息](#)
- [錯誤修正](#)

### 支援檢視

新增了對檢視的支援。您現在可以在 Athena 中使用 [CREATE VIEW](#)、[DESCRIBE VIEW](#)、[DROP VIEW](#)、[SHOW CREATE VIEW](#) 和 [SHOW VIEWS](#)。您在查詢中每次參考檢視時，定義該檢視的查詢就會執行。如需詳細資訊，請參閱[使用檢視](#)。

## 改善和更新錯誤訊息

- 在中包含 GSON 2.8.0 程式庫 CloudTrail SerDe，以解決 JSON 字串的問題 CloudTrail SerDe 並啟用剖析。
- 在 Athena 中增強了對於 Parquet 的分割區結構描述驗證，在某些情況下允許重新排序資料欄，以適用於 ORC。這使 Athena 能夠更好地處理結構描述演進隨著時間的變化，以及 AWS Glue 爬行者程式新增的表格。如需詳細資訊，請參閱 [處理結構描述更新](#)。
- 增加對 SHOW VIEWS 的剖析支援。
- 對最常見的錯誤訊息做出以下改進：
  - 在 Athena 查詢中剖析資料行 SerDe 失敗時，將內部錯誤訊息取代為描述性錯誤訊息。以前，Athena 在剖析錯誤的情況下會發出內部錯誤。新的錯誤訊息是：  
「HIVE\_BAD\_DATA：剖析欄位 0 的欄位值時發生錯誤：java.lang.String 無法轉換為 org.openx.data.jsonserde.json.JSONObject」。
  - 新增更多詳細資訊以改善有關許可不足的錯誤訊息。

## 錯誤修正

修正以下錯誤：

- 修正內部可能將 REAL 轉譯為 FLOAT 資料類型的問題。這可改善與 AWS Glue 爬蟲程式 (傳回 FLOAT 資料類型) 之間的整合。
- 修正了 Athena 未將 AVRO DECIMAL (邏輯類型) 轉換為 DECIMAL 類型的問題。
- 修正了以 WHERE 子句參考 TIMESTAMP 資料類型的值來查詢 Parquet 資料時，Athena 未傳回結果的問題。

## 2018 年 5 月 17 日

2018-05-17 發布

將 Athena 中的查詢並行配額從 5 提高到了 20。這表示您一次可以提交和執行最多 20 個 DDL 查詢和 20 個 SELECT 查詢。請注意，DDL 和 SELECT 查詢有個別的並行配額。

Athena 中的並行配額定義為可同時提交到服務的查詢次數。您一次最多可以提交 20 個相同類型 (DDL 或 SELECT) 的查詢。如果您提交的查詢超過並行查詢配額，則 Athena API 會顯示錯誤訊息。

當您將查詢提交到 Athena 之後，它會根據整體服務負載和傳入的請求數量，指派資源來處理查詢。我們會持續監控並調整服務，讓您的查詢能夠儘快處理。

如需相關資訊，請參閱[Service Quotas](#)。這是可調整的配額。您可以使用 [Service Quotas 主控台](#) 來要求提高並行查詢的配額。

## 2018 年 4 月 19 日

2018-04-19 發布

發行新版本的 JDBC 驅動程式 (2.0.2 版)，支援以陣列資料類型傳回 ResultSet 資料、改進功能和錯誤修正。如需詳細資訊，請參閱驅動程式的[版本備註](#)。

如需有關下載新的 JDBC 驅動程式 2.0.2 版及其文件的資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)。

JDBC 驅動程式的最新版本為 2.0.2。如果您是從 1.x 驅動程式遷移到 2.x 驅動程式，您需要將現有的設定遷移到新的設定。強烈建議您遷移到最新的驅動程式。

如需有關新版本的驅動程式中引進的變更、版本差異和範例的資訊，請參閱[JDBC 驅動程式遷移指南](#)。

## 2018 年 4 月 6 日

2018-04-06 發布

在 Athena 主控台中使用自動完成輸入查詢。

## 2018 年 3 月 15 日

2018-03-15 發布

新增直接從 CloudTrail 主控台為 CloudTrail 記錄檔自動建立 Athena 資料表的功能。如需相關資訊，請參閱[使用主 CloudTrail 控制台為 CloudTrail 記錄建立 Athena 資料表](#)。

## 2018 年 2 月 2 日

2018-02-12 發布

新增針對使用 GROUP BY 子句的記憶體密集型查詢，安全地將中繼資料卸載到磁碟的支援。這可以改善這類查詢的可靠性，防止 "Query resource exhausted" (「查詢資源耗盡」) 錯誤。

## 2018 年 1 月 19 日

2018-01-19 發布

Athena 使用開源分散式查詢引擎 Presto 來執行查詢。

使用 Athena 時，不需要管理版本。我們以透明方式將 Athena 中的基礎引擎升級到基於 Presto 0.172 版的版本。您不需要進行任何動作。

透過這次升級，您現在可以使用 Presto 0.172 函數和運算子，包括 Athena 中的 Presto 0.172 lambda 表達式。

此版本的主要更新 (包括社群貢獻的修正) 包括：

- 支援忽略標頭。您在定義資料表時可以使用 `skip.header.line.count` 屬性，以允許 Athena 忽略標頭。這適用於使用 [LazySimpleSerDe](#) 和 [OpenCSV](#) 的查詢 SerDe，而不適用於 Grok 或正則表達式。SerDes
- 在 STRING 函數中支援 CHAR(n) 資料類型。CHAR(n) 的範圍是 [1,255]，而 VARCHAR(n) 的範圍是 [1,65535]。
- 支援相互關聯的子查詢。
- 支援 Presto Lambda 表達式和函數。
- 提升 DECIMAL 類型和運算子的效能。
- 支援篩選的彙總，例如 `SELECT sum(col_name) FILTER`，其中 `id > 0`。
- 降低 DECIMAL、TINYINT、SMALLINT 和 REAL 資料類型的述詞。
- 支援量化比較述詞：ALL、ANY 和 SOME。
- 新增函數：[arrays\\_overlap\(\)](#)、[array\\_except\(\)](#)、[levenshtein\\_distance\(\)](#)、[codepoint\(\)](#)、[skewness\(\)](#) 和 [typeof\(\)](#)。
- 新增另一版的 [from\\_unixtime\(\)](#) 函數，可接受時區引數。
- 新增 [bitwise\\_and\\_agg\(\)](#) 和 [bitwise\\_or\\_agg\(\)](#) 彙總函數。
- 新增了 [xxhash64\(\)](#) 和 [to\\_big\\_endian\\_64\(\)](#) 函數。
- 對 [json\\_extract\(\)](#) 和 [json\\_extract\\_scalar\(\)](#) 函數，使用反斜線加上 JSON 路徑上標，以增加支援逸出雙引號或雙反斜線。對於任何使用反斜線的叫用，這使得語意改變，因為反斜線在以前視為一般字元。

如需有關函數和運算子的詳細資訊，請參閱本指南中的 [DML 查詢、函數和運算子](#)，以及 Presto 文件中的 [函數和運算子](#)。

Athena 不支援 Presto 的所有功能。如需詳細資訊，請參閱 [限制](#)。

## 2017 年 Athena 版本備註

2017 年 11 月 13 日

2017-11-13 發布

已新增將 Athena 連接到 ODBC 驅動程式的支援。如需相關資訊，請參閱[使用 ODBC 連接到 Amazon Athena](#)。

2017 年 11 月 1 日

2017-11-01 發布

增加支援查詢地理空間資料，也增加支援亞太區域 (首爾)、亞太區域 (孟買) 和歐洲 (倫敦) 區域。如需相關資訊，請參閱[查詢地理空間資料](#)和 [AWS 區域 與端點](#)。

2017 年 10 月 19 日

2017-10-19 發布

增加支援歐洲 (法蘭克福)。如需支援區域的清單，請參閱 [AWS 區域 與端點](#)。

2017 年 10 月 3 日

2017-10-03 發布

使用建立命名的 Athena 查詢 AWS CloudFormation。若要取得更多資訊，請參閱《AWS CloudFormation 使用指南》[AWS::Athena::NamedQuery](#)中的。

2017 年 9 月 25 日

2017-09-25 發布

新增了對亞太區域 (雪梨) 的支援。如需支援區域的清單，請參閱 [AWS 區域 與端點](#)。

2017 年 8 月 14 日

2017-08-14 發布

已新增與 AWS Glue Data Catalog 和移轉精靈的整合，可將 Athena 受管理的資料目錄更新至 AWS Glue Data Catalog。如需詳細資訊，請參閱 [與整合 AWS Glue](#)。

## 2017 年 8 月 4 日

2017-08-04 發布

增加了對 Grok 的支持 SerDe，該功能為非結構化文本文件（例如日誌）中的記錄提供了更容易的模式匹配。如需詳細資訊，請參閱 [GROK SerDe](#)。新增鍵盤捷徑，可使用主控台 (Windows 使用 CTRL+↑/↓，Mac 使用 CMD+↑/↓) 來捲動查詢歷史記錄。

## 2017 年 6 月 22 日

2017-06-22 發布

新增了對亞太區域 (東京) 和亞太區域 (新加坡) 的支援。如需支援區域的清單，請參閱 [AWS 區域與端點](#)。

## 2017 年 6 月 8 日

2017-06-08 發布

新增了對歐洲 (愛爾蘭) 的支援。如需詳細資訊，請參閱 [AWS 區域和端點](#)。

## 2017 年 5 月 19 日

2017-05-19 發布

添加了亞馬遜 Athena API 和對雅典娜的 AWS CLI 支持；更新 JDBC 驅動程序到 1.1.0 版本；修復了各種問題。

- Amazon Athena 支援為 Athena 設計應用程式。如需詳細資訊，請參閱《[Amazon Athena API 參考](#)》。最新的 AWS 開發套件包括對 Athena API 的支援。如需文件和下載的連結，請參閱[適用於 Amazon Web Services 的工具](#)中的軟體開發套件一節。
- 包 AWS CLI 括 Athena 的新命令。如需詳細資訊，請參閱《[Amazon Athena API 參考](#)》。
- 有新的 JDBC 驅動程式 1.1.0 可用，支援新的 Athena API 及最新的功能和錯誤修正。請到此處下載驅動程式：<https://downloads.athena.us-east-1.amazonaws.com/drivers/AthenaJDBC41-1.1.0.jar>。建議升級到最新的 Athena JDBC 驅動程式；不過，您仍可繼續使用舊的驅動程式版本。舊的驅動程式版本不支援 Athena API。如需詳細資訊，請參閱[使用 JDBC 連接到 Amazon Athena](#)。
- 舊版 Athena 中專用於政策陳述式的動作已淘汰。如果您升級到 JDBC 驅動程式 1.1.0 版，並將客戶受管或內嵌 IAM 政策連接到 JDBC 使用者，則必須更新 IAM 政策。反之，舊版的 JDBC 驅動程式不

支援 Athena API，因此，在連接到舊版 JDBC 使用者的政策中，您只能指定已淘汰的動作。因此，您應該不需要更新客戶受管或內嵌 IAM 政策。

- 在 Athena API 發行之之前，Athena 中使用這些政策特定動作。政策中這些已淘汰的動作，只適用於 1.1.0 版以前的 JDBC 驅動程式。如果您正在升級 JDBC 驅動程式，請將允許或拒絕淘汰動作的政策陳述式，取代為列出的適當 API 動作，否則會發生錯誤：

#### 已淘汰的政策特定動作

#### 對應的 Athena API 動作

athena:RunQuery

athena:StartQueryExecution

athena:CancelQueryExecution

athena:StopQueryExecution

athena:GetQueryExecutions

athena:ListQueryExecutions

## 改善項目

- 查詢字串長度限制提高到 256 KB。

## 錯誤修正

- 修正在主控台上捲動結果時導致查詢結果看似格式不正確的問題。
- 修正 Simple Storage Service (Amazon S3) 資料檔案中的 `\u0000` 字元字串會造成錯誤的問題。
- 修正透過 JDBC 驅動程式來請求取消查詢時導致失敗的問題。
- 修正美國東部 (俄亥俄) 的 Amazon S3 資料失敗的問題。AWS CloudTrail SerDe
- 修正導致 DROP TABLE 在分割資料表上失敗的問題。

## 2017 年 4 月 4 日

2017-04-04 發布

增加支援 Simple Storage Service (Amazon S3) 資料加密，並發行 JDBC 驅動程式更新 (1.0.1 版) 來提供加密支援、改進功能和錯誤修正。

## 功能

- 新增以下加密功能：
  - 支援在 Simple Storage Service (Amazon S3) 中查詢加密資料。
  - 支援加密 Athena 查詢結果。
- 新版本的驅動程式支援新的加密功能、新增改進功能並修正問題。
- 新增使用 ALTER TABLE 來新增、取代和變更資料欄的支援。如需詳細資訊，請參閱 Hive 文件中的[更改資料欄](#)。
- 增加支援查詢 LZO 壓縮資料。

如需詳細資訊，請參閱 [靜態加密](#)。

## 改善項目

- 經由改善分頁大小以提高 JDBC 查詢效能，可傳回 1,000 個資料行而不是 100 個資料行。
- 增加支援使用 JDBC 驅動程式界面來取消查詢。
- 增加支援在 JDBC 連線 URL 中指定 JDBC 選項。關於最新的 JDBC 驅動程式，請參閱 [使用 JDBC 連接到 Amazon Athena](#)。
- 在驅動程序中添加了代理設置，現在可以在 Java 的 AWS SDK [ClientConfiguration](#) 中進行設置。

## 錯誤修正

修正以下錯誤：

- 使用 JDBC 驅動程式界面發出多個查詢會發生節流錯誤。
- 投射小數點資料類型時，JDBC 驅動程式會停止。
- 無論資料表中如何定義資料類型，JDBC 驅動程式會以字串傳回每個資料類型。例如，使用 INT 選取定義為 resultSet.GetObject() 資料類型的資料欄會傳回 STRING 資料類型，而不是 INT。
- JDBC 驅動程式會在建立連線時驗證登入資料，而不是在執行查詢時。
- 當結構描述連同 URL 一起指定時，透過 JDBC 驅動程式提出的查詢會失敗。

2017 年 3 月 24 日

2017-03-24 發布

添加了 AWS CloudTrail SerDe，改進的性能，固定的分區問題。

## 功能

- 已新增 AWS CloudTrail SerDe，此後已被讀取 CloudTrail 記錄所取[蜂巢 SerDe](#)代。如需查詢 CloudTrail 記錄檔的詳細資訊，請參閱[查詢 AWS CloudTrail 記錄檔](#)。

## 改善項目

- 改進了掃描大量分割區時的效能。
- 改進了 MSCK Repair Table 操作的效能。
- 增加了查詢主要區域以外的區域中存放的 Simple Storage Service (Amazon S3) 資料的功能。除了標準 Athena 費用，也要支付 Simple Storage Service (Amazon S3) 的標準跨區域資料傳輸費用。

## 錯誤修正

- 修正未載入分割區時可能發生「找不到資料表錯誤」的錯誤。
- 修正錯誤以避免 ALTER TABLE ADD PARTITION IF NOT EXISTS 查詢擲回例外狀況。
- 修正 DROP PARTITIONS 中的錯誤。

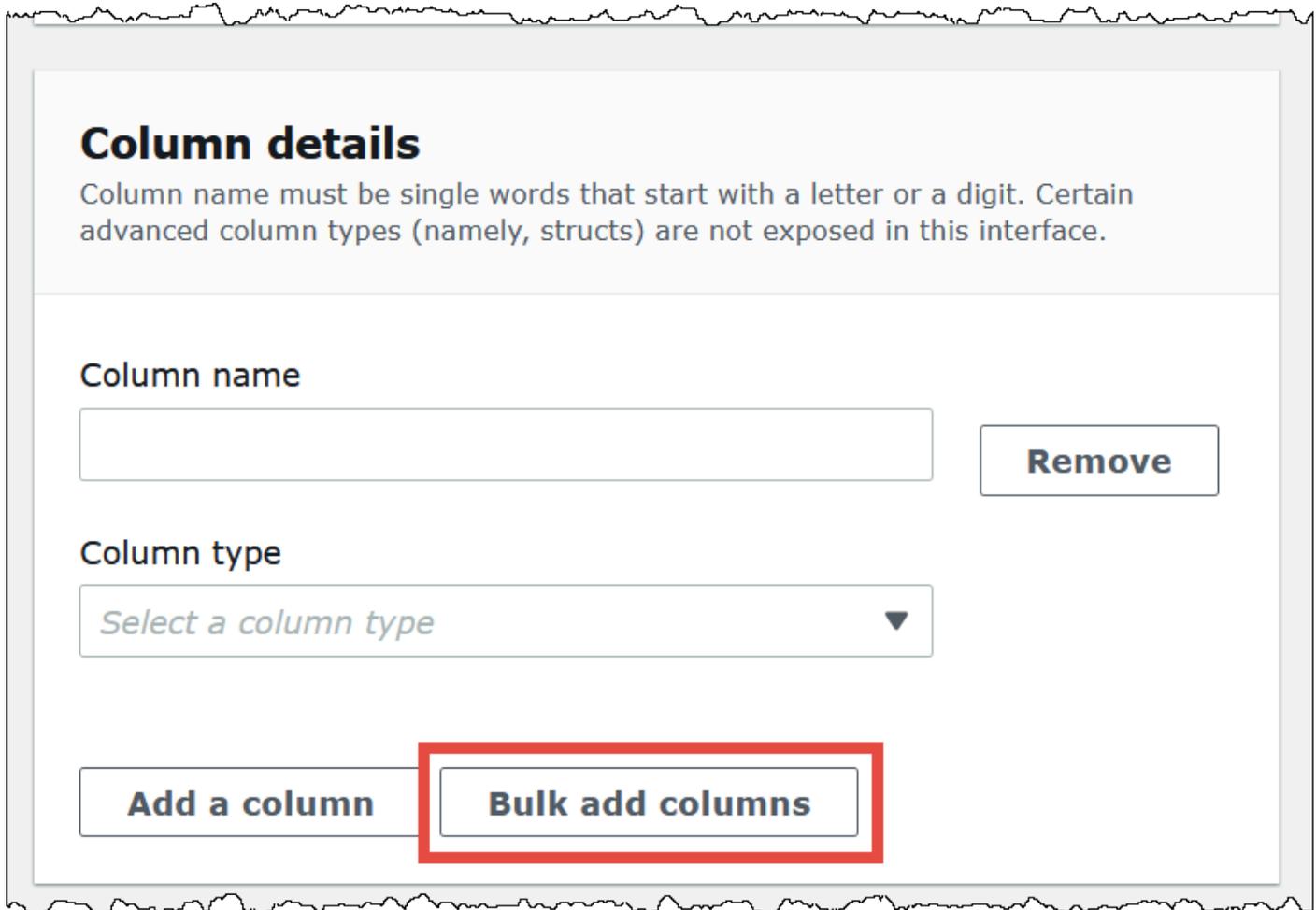
## 2017 年 2 月 20 日

2017-02-20 發布

已新增對 AvroSerDe 和 OpenCSV SerDe、美國東部 (俄亥俄) 區域的支援，以及主控台精靈中的大量編輯欄。改進大型 Parquet 資料表的效能。

## 功能

- 推出了對新的支持 SerDes：
  - [Avro SerDe](#)
  - [SerDe 用於處理 CSV 的開啟 CSV](#)
- 美國東部 (俄亥俄) 區域 (us-east-2) 啟動。您現在可以在此區域中執行查詢。
- 現已可使用從 S3 儲存貯體資料建立資料表方式來大量定義資料表的結構描述。在查詢編輯器中，選擇 Create (建立)、S3 bucket data (S3 儲存貯體資料)，然後選擇 Column details (資料欄詳細資訊) 部分的 Bulk add columns (大量新增欄)。



**Column details**

Column name must be single words that start with a letter or a digit. Certain advanced column types (namely, structs) are not exposed in this interface.

Column name

**Remove**

Column type

Select a column type ▼

**Add a column** **Bulk add columns**

在文字方塊中輸入名稱/值對，然後選擇 Add (新增)。

## Bulk add columns ×

Define columns in name value pairs, using commas to separate definitions (col1\_name data\_type, col2\_name data\_type, ...). Certain advanced data types (namely, structs) are not supported in this interface, but are supported using DDL statements.

```
id int, name string
```

## 改善項目

- 改進大型 Parquet 資料表的效能。

## 文件歷史紀錄

最新文件更新：2024 年 6 月 28 日。

我們會經常更新文件，以處理您的意見回饋。下表說明 Amazon Athena 文件的重要增補項目。並非所有更新都會顯示。

變更	描述	發行日期
更新了 AmazonAthenaFullAccess 受管政策。	已新增 <code>glue:GetCatalogImportStatus</code> 至受 <a href="#">AmazonAthenaFullAccess</a> 管理原則的 <code>BaseGluePPermissions</code> 區段。新增的動作可讓 Athena 使用公開記錄的 AWS Glue API 來擷取目錄匯入狀態。	2024年6月18日
更新了 AmazonAthenaFullAccess 受管政策。	<code>datazone:ListDomains</code> 、 <code>datazone:ListProjects</code> 和 <code>datazone:ListAccountEnvironments</code> 權限已新增至 <a href="#">AmazonAthenaFullAccess</a> 受管理的策略。新增的動作可讓 Athena 使用者使用 Amazon DataZone 網域、專案和環境。如需詳細資訊，請參閱 <a href="#">在 Athena 中使用 Amazon DataZone</a> 。	2024 年 1 月 3 日
更新了 AmazonAthenaFullAccess 受管政策。	已新增 <code>glue:StartColumnStatisticsTaskRun</code> 、 <code>glue:GetColumnStatisticsTaskRun</code> 、和 <code>glue:GetColumnStatisticsTaskRuns</code> 權限至 <a href="#">AmazonAthenaFullAccess</a> 受管理的策略。新增的動作可讓 Athena 呼叫，擷取 AWS Glue 以成本為基礎的最佳化程式功能的統計資料。如需詳細資訊，請參閱 <a href="#">使用成本型最佳化工具</a> 。	2024 年 1 月 3 日
新增了啟用 IAM Identity Center 的 Athena 工作群組的文件。	您可建立使用 IAM Identity Center 身分驗證方式的 Athena SQL 工作群組。這些工作群組支援跨 Amazon 雅典娜和亞馬遜 EMR 工作室等 AWS 服務使用相同的身分。如需詳細資訊，請參閱 <a href="#">使用啟用 IAM Identity Center 的 Athena 工作群組</a> 。	2023 年 12 月 5 日

變更	描述	發行日期
新增了查詢 S3 Express 單區域資料的文件	您可使用 Athena 在 Amazon S3 Express 單區域儲存類別中查詢資料。如需詳細資訊，請參閱 <a href="#">查詢 S3 Express 單區域資料</a> 。	2023 年 11 月 28 日
新增了 Glue Data Catalog 檢視的文件。	您可使用 Glue Data Catalog 檢視，來提供跨 Amazon Athena 和 Amazon Redshift 等 AWS 服務的單一通用檢視。如需詳細資訊，請參閱 <a href="#">使用 AWS Glue Data Catalog 視圖</a> 。	2023 年 11 月 27 日
新增了以成本為基礎的最佳化工具的文件。	您可以使用來自的統計資料 AWS Glue 來最佳化您在 Athena SQL 中的查詢。如需詳細資訊，請參閱 <a href="#">使用成本型最佳化工具</a> 。	2023 年 11 月 17 日
新增了 Athena JDBC 3.x 驅動程式的文件	您可以使用 Athena JDBC 3.x 驅動程式，直接從 Amazon S3 讀取查詢結果。JDBC 3.x 驅動程式幾乎支援 JDBC 2.x 驅動程式支援的所有身分驗證方法。如需詳細資訊，請參閱 <a href="#">Athena JDBC 3.x 驅動程式</a> 。	2023 年 11 月 16 日
增加了 DataZone 在 Athena 使用的文檔。	您可以使用 DataZone 來簡化 Athena 和 Lake Formation 等 AWS 分析服務的體驗。AWS Glue 如需詳細資訊，請參閱 <a href="#">在 Athena 中使用 Amazon DataZone</a> 。	2023 年 10 月 4 日
新增容量保留的文件。	您現在可以在 Amazon Athena 上的容量保留，以在完全受管的運算容量上執行 SQL 查詢。如需詳細資訊，請參閱 <a href="#">管理查詢處理容量</a> 。	2023 年 4 月 28 日
新增查詢聯合檢視的文件。	您現在可以在 Athena 中建立和查詢聯合資料來源上的檢視。如需詳細資訊，請參閱 <a href="#">查詢聯合檢視</a> 。	2023 年 4 月 4 日
新增關於防止 Amazon S3 限流的文件。	如需詳細資訊，請參閱 <a href="#">防止 Amazon S3 限流</a> 。	2023 年 3 月 24 日

變更	描述	發行日期
更新了 AmazonAthenaFullAccess 受管政策。	已新增 <code>pricing:GetProducts</code> 至受 <a href="#">AmazonAthenaFullAccess</a> 管理的策略。新增的動作可讓您存取 AWS Billing and Cost Management。如需詳細資訊，請參閱 AWS Billing and Cost Management API 參考 <a href="#">GetProducts</a> 中的。	2023 年 1 月 25 日
已擴充 Athena 壓縮支援的文件。	為 <a href="#">Hive 資料表壓縮</a> 、 <a href="#">Iceberg 資料表壓縮</a> 和 <a href="#">ZSTD 壓縮級別</a> 新增的單獨主題。如需詳細資訊，請參閱 <a href="#">Athena 壓縮支援</a> 。	2023 年 1 月 20 日
新增了 Amazon Athena for Apache Spark 的文件。	您現在可以在 Amazon Athena 上以互動方式建立和執行 Apache Spark 應用程式和與 Jupyter 相容的筆記本。如需詳細資訊，請參閱 <a href="#">在 Amazon Athena 使用 Apache Spark</a> 。	2022 年 11 月 30 日
新增了 Athena IBM Db2 連接器的文件。	您可以使用適用於 IBM Db2 的 Amazon Athena 連接器，查詢來自 Athena 的 Db2。如需更多資訊，請參閱 <a href="#">Amazon Athena IBM Db2 連接器</a>	2022 年 11 月 18 日
新增了查詢結果重複使用的文件。	在 Athena 中重新執行查詢時，您可以選擇性地選擇重複使用最近儲存的查詢結果。這可以提高效能並降低已掃描之位元組數的成本。如需詳細資訊，請參閱 <a href="#">重複使用查詢結果</a> 。	2022 年 11 月 8 日
更新的 CloudTrail 日誌文檔。	用於查詢 CloudTrail 記錄檔的 CREATE TABLE DDL 已更新為使用 JSON SerDe 而非 CloudTrail SerDe 如需詳細資訊，請參閱 <a href="#">查詢 AWS CloudTrail 記錄檔</a> 。	2022 年 11 月 3 日
新增了 Athena 引擎版本 3 的文件。	如需有關 Athena 引擎版本 3 的詳細資訊，請參閱 <a href="#">Athena 引擎版本 3</a> 。	2022 年 10 月 13 日
新增使用 Okta 外掛程式為 ODBC 設定 SSO 的教學課程。	設定 Amazon Athena ODBC 驅動程式和 Okta 外掛程式，以使用 Okta 身分提供者新增單一登入 (SSO) 功能。如需詳細資訊，請參閱 <a href="#">使用 Okta 外掛程式和 Okta 身分提供者設定 ODBC 的 SSO</a> 。	2022 年 8 月 23 日

變更	描述	發行日期
新增了 Athena 主控台中檢視查詢計劃和統計資料的說明文件。	您可以使用 Athena 查詢編輯器來查看查詢執行方式的圖形呈現，以及已完成查詢之執行方式的圖形、詳細資料和統計資料。如需詳細資訊，請參閱 <a href="#">檢視 SQL 查詢的執行計劃</a> 及 <a href="#">檢視已完成查詢的統計資料和執行詳細資訊</a> 。	2022 年 7 月 21 日
已新增查詢外部 Hive 中繼存放區中的 Apache Hive 視圖的文件。	您可以使用 Athena 來查詢在外部 Hive 中繼存放區中建立的 Apache 視圖。某些 Hive 功能不受支援或需要特殊處理。如需詳細資訊，請參閱 <a href="#">使用 Hive 檢視</a> 。	2022 年 4 月 22 日
為已儲存的查詢新增文件。	您可以使用 Athena 中的已儲存的查詢的功能來儲存、重新呼叫、編輯和重新命名您的查詢。如需詳細資訊，請參閱 <a href="#">使用已儲存的查詢</a> 本指南和 Amazon Athena API 參考 <a href="#">UpdateNamedQuery</a> 中的。	2022 年 2 月 28 日
針對 Apache Iceberg 支援新增了預覽文件。	Athena 支援 Apache Iceberg 資料表的讀取、時間旅行和寫入查詢，這些資料表使用 Apache 拼花格式來處理資料，以及其中繼存放區的 AWS Glue 目錄。如需詳細資訊，請參閱 <a href="#">使用 Apache Iceberg 資料表</a> 。	2021 年 11 月 26 日
新增跨帳戶聯合查詢的文件。	您可以使用跨帳戶聯合查詢功能來查詢另一個帳戶中的資料來源。如需設定許可以啟用這項功能的相關資訊，請參閱 <a href="#">啟用跨帳戶聯合查詢</a> 。	2021 年 11 月 12 日
新增了 Athena UNLOAD 陳述式的文件。	使用要撰寫的 UNLOAD 陳述式來查詢對 Apache Parquet、ORC，Apache Avro 和 JSON 格式之 SELECT 陳述式結果。如需詳細資訊，請參閱 <a href="#">UNLOAD</a> 。	2021 年 8 月 5 日
新增了 Athena EXPLAIN 陳述式功能的文件。	如需詳細資訊，請參閱 <a href="#">在 Athena 使用 EXPLAIN 和 EXPLAIN ANALYZE</a> 及 <a href="#">了解 Athena EXPLAIN 陳述式結果</a> 。	2021 年 4 月 5 日
在 Athena 中新增了疑難排解和效能調校。	如需詳細資訊，請參閱 <a href="#">Athena 中的疑難排解</a> 及 <a href="#">Athena 中的效能調校</a> 。	2020 年 12 月 30 日

變更	描述	發行日期
新增了 Athena 引擎版本控制和 Athena 引擎版本 2 的文件。	如需詳細資訊，請參閱 <a href="#">Athena 引擎版本控制</a> 。	2020 年 11 月 11 日
更新了一般可用性發行版本的聯合查詢文件。	如需詳細資訊，請參閱 <a href="#">使用 Amazon Athena 聯合查詢</a> 及 <a href="#">搭配 CalledVia 內容金鑰使用 Athena</a> 。	2020 年 11 月 11 日
新增了搭配 Lake Formation 使用 JDBC 驅動程式以聯合存取 Athena 的文件。	如需詳細資訊，請參閱 <a href="#">使用 Lake Formation 和 Athena JDBC 和 ODBC 驅動程式對 Athena 進行聯合存取及教學課程：使用 Lake Formation 和 JDBC 設定 Okta 使用者對 Athena 的聯合存取</a> 。	2020 年 9 月 25 日
已新增 Amazon Athena OpenSearch 資料連接器的文件。	如需詳細資訊，請參閱 <a href="#">Amazon Athena OpenSearch 連</a> 。	2020 年 7 月 21 日
新增了查詢 Hudi 資料集的文件。	如需詳細資訊，請參閱 <a href="#">使用 Athena 查詢 Apache Hudi 資料集</a> 。	2020 年 7 月 9 日
新增了有關查詢存放在 Simple Storage Service (Amazon S3) 中的 Apache web 伺服器日誌和 IIS web 伺服器日誌的文件。	如需詳細資訊，請參閱 <a href="#">查詢存放在 Amazon S3 中的 Apache 日誌</a> 及 <a href="#">查詢存放在 Amazon S3 中的網際網路資訊伺服器 (IIS) 日誌</a> 。	2020 年 7 月 8 日
新增了外部 Hive 中繼存放區中一般發行版本 Athena 資料連接器的文件。	如需詳細資訊，請參閱 <a href="#">將 Athena 資料連接器用於外部 Hive 中繼存放區</a> 。	2020 年 6 月 1 日

變更	描述	發行日期
新增了用於標記資料目錄資源的文件。	如需詳細資訊，請參閱 <a href="#">標記 Athena 資源</a> 。	2020 年 6 月 1 日
新增了分割區投影的文件。	如需詳細資訊，請參閱 <a href="#">使用 Amazon Athena 進行分割區投影</a> 。	2020 年 5 月 21 日
更新了 Athena 的 Java 程式碼範例。	如需詳細資訊，請參閱 <a href="#">程式碼範例</a> 。	2020 年 5 月 11 日
添加了有關查詢 Amazon GuardDuty 發現的主題。	如需詳細資訊，請參閱 <a href="#">查詢 Amazon GuardDuty 發現</a> 。	2020 年 3 月 19 日
已新增使用 CloudWatch 事件監視 Athena 查詢狀態轉換的主題。	如需詳細資訊，請參閱 <a href="#">使用 Amazon EventBridge 事件監控 Athena 查</a> 。	2020 年 3 月 11 日
已新增有關使用 Athena 查詢 AWS Global Accelerator 流程記錄的主題。	如需詳細資訊，請參閱 <a href="#">查詢 AWS Global Accelerator 流程記錄</a> 。	2020 年 2 月 6 日

變更	描述	發行日期
<ul style="list-style-type: none"> <li>• 新增了有關使用 CTAS 搭配 INSERT INTO 將未分割來源的資料新增至分割目的地的說明文件。</li> <li>• 新增了 Athena 的 ODBC 驅動程式 1.1.0 預覽版本的下載連結。</li> <li>• 已更正 SHOW DATABASES LIKE regex 的描述。</li> <li>• 已更正 CTA 主題中的 partitioned_by 語法。</li> <li>• 其他次要修正。</li> </ul>	<p>文件更新包括但不限於下列主題：</p> <ul style="list-style-type: none"> <li>• <a href="#">使用 CTAS 和 INSERT INTO 以進行 ETL 和資料分析</a></li> <li>• <a href="#">使用 ODBC 連接到 Amazon Athena</a> (1.1.0 預覽功能現已包含在 1.1.2 ODBC 驅動程式中。)</li> <li>• <a href="#">SHOW DATABASES</a></li> <li>• <a href="#">CREATE TABLE AS</a></li> </ul>	2020 年 2 月 4 日
<p>新增有關使用 CTAS 搭配 INSERT INTO 將已分割來源的資料新增至分割目的地的說明文件。</p>	<p>如需詳細資訊，請參閱<a href="#">使用 CTAS 和 INSERT INTO 來解決 100 個分割區限制</a>。</p>	2020 年 1 月 22 日
<p>已更新查詢結果位置資訊。</p>	<p>Athena 不再建立「預設」查詢結果位置。如需詳細資訊，請參閱<a href="#">指定查詢結果位置</a>。</p>	2020 年 1 月 20 日

變更	描述	發行日期
已新增有關查詢 AWS Glue Data Catalog. 更新 Athena 中服務配額 (舊稱「服務限制」) 的相關資訊。	<p>如需詳細資訊，請參閱下列主題：</p> <ul style="list-style-type: none"> <li>• <a href="#">查詢 AWS Glue Data Catalog</a></li> <li>• <a href="#">Service Quotas</a></li> </ul>	2020 年 1 月 17 日
已更正 OpenCSV 的主題，SerDe 以注意應該以 UNIX 數字格式指定TIMESTAMP 類型。	如需詳細資訊，請參閱 <a href="#">SerDe 用於處理 CSV 的開啟 CSV</a> 。	2020 年 1 月 15 日
更新了有關加密的安全性主題，以註明 Athena 不支援非對稱金鑰。	<p>Athena 僅支援用於讀取和寫入資料的對稱金鑰。</p> <p>如需詳細資訊，請參閱<a href="#">支援的 Amazon S3 加密選項</a>。</p>	2020 年 1 月 8 日
已新增使用自訂 AWS KMS 金鑰加密之 Amazon S3 儲存貯體的跨帳戶存取相關資訊。	如需詳細資訊，請參閱 <a href="#">跨帳戶存取使用自訂 AWS KMS 金鑰加密的儲存貯體</a> 。	2019 年 12 月 13 日
新增了聯合查詢、外部 Hive 中繼存放區、機器學習和使用者定義函數的文件。新增 CloudWatch 指標。	<p>如需詳細資訊，請參閱下列主題：</p> <ul style="list-style-type: none"> <li>• <a href="#">使用 Amazon Athena 聯合查詢</a></li> <li>• <a href="#">可用資料來源連接器</a></li> <li>• <a href="#">將 Athena 資料連接器用於外部 Hive 中繼存放區</a></li> <li>• <a href="#">搭配 Amazon Athena 使用機器學習 (ML)</a></li> <li>• <a href="#">利用使用者定義函數來查詢</a></li> <li>• <a href="#">Athena 的 CloudWatch 度量和維度清單</a></li> </ul>	2019 年 11 月 26 日

變更	描述	發行日期
新增了 INSERT INTO 命令章節，以及更新支援資料資訊清單檔案的查詢結果位置資訊。	如需詳細資訊，請參閱 <a href="#">INSERT INTO</a> 及 <a href="#">使用查詢結果、近期查詢和輸出檔案</a> 。	2019 年 9 月 18 日
已新增介面 VPC 端點 (PrivateLink) 支援的區段。更新了 JDBC 驅動程式。更新豐富型 VPC 流程日誌的相關資訊。	如需詳細資訊，請參閱 <a href="#">使用界面 VPC 端點連接至 Amazon Athena</a> 、 <a href="#">查詢 Amazon VPC 流程日誌</a> 及 <a href="#">使用 JDBC 連接到 Amazon Athena</a> 。	2019 年 9 月 11 日
增加了與集成部分 AWS Lake Formation。	如需詳細資訊，請參閱 <a href="#">使用 Athena 查詢向 AWS Lake Formation 註冊的資料</a> 。	2019 年 6 月 26 日
更新了安全性章節以保持與其他 AWS 服務的一致性。	如需詳細資訊，請參閱 <a href="#">Amazon Athena 安全性</a> 。	2019 年 6 月 26 日
已新增查詢記 AWS WAF 錄檔的區段。	如需詳細資訊，請參閱 <a href="#">查詢 AWS WAF 記錄檔</a> 。	2019 年 5 月 31 日
發布了支援 Athena 工作群組的新版 ODBC 驅動程式。	<p>若要下載 ODBC 驅動程式 1.0.5 版及其文件，請參閱 <a href="#">使用 ODBC 連接到 Amazon Athena</a>。當您在工作群組上使用標籤時，ODBC 驅動程式連線字串沒有變更。若要使用標籤，請升級到最新版本的 ODBC 驅動程式 (目前的版本)。</p> <p>此驅動程式版本可讓您使用 <a href="#">Athena API 工作群組動作</a> 建立和管理工作群組，並使用 <a href="#">Athena API 標籤動作</a> 在工作群組上新增、列出或移除標籤。開始之前，請確定您在 IAM 中有資源層級許可，而能夠對工作群組和標籤執行動作。</p>	2019 年 3 月 5 日

變更	描述	發行日期
對 Amazon Athena 中的工作群組新增了標籤支援。	每個標記皆包含由您定義的金鑰和值。當您標記工作群組時，您可以指派自訂中繼資料給它。例如，為每個成本中心建立工作群組。然後，您可以將標籤新增到工作群組，以追蹤每個成本中心的 Athena 花費。如需詳細資訊，請參閱《AWS Billing and Cost Management 使用者指南》中的 <a href="#">使用標籤計費</a> 。	2019 年 2 月 22 日
改進了在 Athena SerDe 使用的 JSON OpenX。	<p>改善功能包括 (但不限於) 下列各項：</p> <ul style="list-style-type: none"> <li>• 支援 <code>ConvertDotsInJsonKeysToUnderscores</code> 屬性。當設定為時 <code>TRUE</code>，它允許 SerDe 用底線取代關鍵名稱中的點。例如，如果 JSON 資料集包含名為 "a.b" 的索引鍵，您可以使用此屬性在 Athena 中將資料欄名稱定義為 "a_b"。預設值為 <code>FALSE</code>。在預設情況下，Athena 在資料欄名稱中不允許點。</li> <li>• 支援 <code>case.insensitive</code> 屬性。在預設情況下，Athena 規定 JSON 資料集裡的所有索引鍵必須使用小寫。使用 <code>WITH SERDE PROPERTIES ("case.insensitive"= FALSE;)</code> 可讓您在資料中使用區分大小寫的索引鍵名稱。預設值為 <code>TRUE</code>。設定為時 <code>TRUE</code>，會 SerDe 將所有大寫欄轉換為小寫。</li> </ul> <p>如需詳細資訊，請參閱 <a href="#">OpenX SerDe</a>。</p>	2019 年 2 月 18 日
新增支援工作群組。	使用工作群組來區隔使用者、團隊、應用程式或工作負載，以及對每個查詢或整個工作群組可以處理的資料量設定限制。因為工作群組扮演 IAM 資源的角色，所以您可以使用資源層級許可來控制對特定工作群組的存取。您也可以可以在 Amazon 中檢視查詢相關指標 CloudWatch、透過設定掃描的資料量限制來控制查詢成本、建立閾值，以及在超出這些閾值時觸發動作 (例如 Amazon SNS 警示)。如需詳細資訊，請參閱 <a href="#">使用工作群組來執行查詢</a> 及 <a href="#">使用指標和事件控制成本並 CloudWatch 監控查詢</a> 。	2019 年 2 月 18 日

變更	描述	發行日期
<p>新增了對分析 Network Load Balancer 日誌的支援。</p>	<p>新增了 Athena 查詢範例，用於分析 Network Load Balancer 日誌。這些日誌會接收傳送至 Network Load Balancer 之 Transport Layer Security (TLS) 的詳細資訊。您可以使用這些存取日誌來分析流量模式和排除問題。如需相關資訊，請參閱<a href="#">the section called “Network Load Balancer”</a>。</p>	<p>2019 年 1 月 24 日</p>
<p>發行了新版本的 JDBC 和 ODBC 驅動程式，支援透過 AD FS 和 SAML 2.0 (安全性聲明標記語言 2.0) 來聯合存取 Athena API。</p>	<p>此版本驅動程式支援 Active Directory Federation Service (AD FS 3.0) 聯合存取 Athena。透過支援 SAML 2.0 的 JDBC 或 ODBC 驅動程式版本建立存取權。如需有關設定聯合存取 Athena API 的資訊，請參閱<a href="#">the section called “啟用對 Athena API 的聯合存取”</a>。</p>	<p>2018 年 11 月 10 日</p>
<p>新增了對 Athena 中資料庫和資料表的精細存取控制的支援。此外，在 Athena 中新增的政策允許您加密資料目錄中的資料庫和資料表中繼資料。</p>	<p>已新增建立身分型 (IAM) 政策的支援 AWS Glue Data Catalog，這些政策可對中的資源 (例如 Athena 中使用的資料庫和表格) 提供更精細的存取控制。</p> <p>此外，您可以將特定政策新增至 Athena，以加密資料目錄中的資料庫和資料表中繼資料。</p> <p>如需詳細資訊，請參閱 <a href="#">對於中的資料庫和資料表的精細存取 AWS Glue Data Catalog</a>。</p>	<p>2018 年 10 月 15 日</p>
<p>新增對 CREATE TABLE AS SELECT 陳述式的支援。</p> <p>在文件中做了其他改進。</p>	<p>新增對 CREATE TABLE AS SELECT 陳述式的支援。請參閱<a href="#">從查詢結果建立資料表 (CTAS)</a>、<a href="#">CTAS 查詢的考量事項與限制</a>和<a href="#">CTAS 查詢的範例</a>。</p>	<p>2018 年 10 月 10 日</p>

變更	描述	發行日期
<p>發行 ODBC 驅動程式 1.0.3 版，此版本支援串流結果功能，而非在頁面中擷取結果。</p> <p>在文件中做了其他改進。</p>	<p>ODBC 驅動程式 1.0.3 版支援串流結果，也包含改進功能、錯誤修正以及說明使用 SSL 搭配代理伺服器的更新文件。</p> <p>若要下載 ODBC 驅動程式 1.0.3 版及其文件，請參閱<a href="#">使用 ODBC 連接到 Amazon Athena</a>。</p>	<p>2018 年 9 月 6 日</p>
<p>發行 JDBC 驅動程式 2.0.5 版，此版本預設支援串流結果功能，而非在頁面中擷取結果。</p> <p>在文件中做了其他改進。</p>	<p>發行 JDBC 驅動程式 2.0.5 版，此版本預設支援串流結果功能，而非在頁面中擷取結果。如需相關資訊，請參閱<a href="#">使用 JDBC 連接到 Amazon Athena</a>。</p>	<p>2018 年 8 月 16 日</p>
<p>更新了文件中查詢 Amazon Virtual Private Cloud 流程日誌的部分；這種日誌可在 Simple Storage Service (Amazon S3) 中以 GZIP 格式直接存放。</p> <p>更新了查詢 ALB 日誌的範例。</p>	<p>更新了文件中查詢 Amazon Virtual Private Cloud 流程日誌的部分；這種日誌可在 Simple Storage Service (Amazon S3) 中以 GZIP 格式直接存放。如需相關資訊，請參閱<a href="#">查詢 Amazon VPC 流程日誌</a>。</p> <p>更新了查詢 ALB 日誌的範例。如需相關資訊，請參閱<a href="#">查詢 Application Load Balancer 日誌</a>。</p>	<p>2018 年 8 月 7 日</p>
<p>新增了對檢視的支援。新增了各種資料儲存體格式的結構描述操作指導方針。</p>	<p>新增了對檢視的支援。如需相關資訊，請參閱<a href="#">使用檢視</a>。</p> <p>更新了本指南，說明對各種資料儲存體格式之結構描述的處理指導更新。如需相關資訊，請參閱<a href="#">處理結構描述更新</a>。</p>	<p>2018 年 6 月 5 日</p>

變更	描述	發行日期
預設查詢並行限制從 5 提高到 20。	您一次可以提交和執行最多 20 個 DDL 查詢和 20 個 SELECT 查詢。如需相關資訊，請參閱 <a href="#">Service Quotas</a> 。	2018 年 5 月 17 日
新增了查詢標籤，以及在查詢編輯器中設定自動完成的功能。	新增了查詢標籤，以及在查詢編輯器中設定自動完成的功能。如需相關資訊，請參閱 <a href="#">開始使用</a> 。	2018 年 5 月 8 日
發行了 JDBC 驅動程式 2.0.2 版。	發行了新版本的 JDBC 驅動程式 (2.0.2 版)。如需相關資訊，請參閱 <a href="#">使用 JDBC 連接到 Amazon Athena</a> 。	2018 年 4 月 19 日
新增了在 Athena 主控台中輸入查詢會自動完成的功能。	新增了在 Athena 主控台中輸入查詢會自動完成的功能。	2018 年 4 月 6 日
新增直接從 CloudTrail 主控台為 CloudTrail 記錄檔建立 Athena 資料表的功能。	新增直接從 CloudTrail 主控台為 CloudTrail 記錄檔自動建立 Athena 資料表的功能。如需相關資訊，請參閱 <a href="#">使用主 CloudTrail 控制台為 CloudTrail 記錄建立 Athena 資料表</a> 。	2018 年 3 月 15 日
新增針對使用 GROUP BY 的查詢，安全地將中繼資料卸載到磁碟的支援。	新增針對使用 GROUP BY 子句的記憶體密集型查詢，安全地將中繼資料卸載到磁碟的支援。這可以改善這類查詢的可靠性，防止 "Query resource exhausted" (「查詢資源耗盡」) 錯誤。如需詳細資訊，請參閱 <a href="#">2018 年 2 月 2 日</a> 的版本備註。	2018 年 2 月 2 日
新增對 Presto 0.172 版的支援。	將 Amazon Athena 中的基礎引擎升級到以 Presto 0.172 版為基礎的版本。如需詳細資訊，請參閱 <a href="#">2018 年 1 月 19 日</a> 的版本備註。	2018 年 1 月 19 日
新增對 ODBC 驅動程式的支援。	已新增將 Athena 連接到 ODBC 驅動程式的支援。如需相關資訊，請參閱 <a href="#">使用 ODBC 連接到 Amazon Athena</a> 。	2017 年 11 月 13 日

變更	描述	發行日期
<p>新增了對亞太區域 (首爾)、亞太區域 (孟買) 和歐洲 (倫敦) 區域的支援。</p> <p>新增了查詢地理空間資料的支援。</p>	<p>新增了對查詢地理空間資料，以及亞太區域 (首爾)、亞太區域 (孟買) 和歐洲 (倫敦) 區域的支援。如需相關資訊，請參閱<a href="#">查詢地理空間資料</a>和 <a href="#">AWS 區域 與端點</a>。</p>	2017 年 11 月 1 日
<p>新增了對歐洲 (法蘭克福) 的支援。</p>	<p>新增了對歐洲 (法蘭克福) 的支援。如需支援區域的清單，請參閱 <a href="#">AWS 區域 與端點</a>。</p>	2017 年 10 月 19 日
<p>增加了對命名 Athena 查詢的支持 AWS CloudFormation。</p>	<p>已新增支援使用建立命名的 Athena 查詢 AWS CloudFormation。若要取得更多資訊，請參閱《AWS CloudFormation 使用指南》<a href="#">AWS::Athena::Named Query</a>中的。</p>	2017 年 10 月 3 日
<p>新增了對亞太區域 (雪梨) 的支援。</p>	<p>新增了對亞太區域 (雪梨) 的支援。如需支援區域的清單，請參閱 <a href="#">AWS 區域 與端點</a>。</p>	2017 年 9 月 25 日
<p>在本指南中新增一個章節，以查詢 AWS 服務記錄檔和不同類型的資料，包括地圖、陣列、巢狀資料和包含 JSON 的資料。</p>	<p>新增了 <a href="#">查詢 AWS 服務記錄檔</a> 的範例以及在 Athena 中查詢不同類型資料的範例。如需相關資訊，請參閱<a href="#">使用 Amazon Athena 執行 SQL 查詢</a>。</p>	2017 年 9 月 5 日
<p>增加了對 AWS Glue Data Catalog.</p>	<p>已新增與 AWS Glue Data Catalog 和移轉精靈的整合，可將 Athena 受管理的資料目錄更新至 AWS Glue Data Catalog. 若需詳細資訊，請參閱<a href="#">與 AWS Glue整合</a>以及 <a href="#">AWS Glue</a>。</p>	2017 年 8 月 14 日
<p>增加了對格羅克 SerDe的支持。</p>	<p>增加了對 Grok 的支持 SerDe，該功能為非結構化文本文件 (例如日誌) 中的記錄提供了更容易的模式匹配。如需詳細資訊，請參閱 <a href="#">Grok SerDe</a>。新增了鍵盤快速鍵，可使用主控台捲動查詢歷程記錄。</p>	2017 年 8 月 4 日

變更	描述	發行日期
新增了對亞太區域 (東京) 的支援。	新增了對亞太區域 (東京) 和亞太區域 (新加坡) 的支援。如需支援區域的清單，請參閱 <a href="#">AWS 區域 與端點</a> 。	2017 年 6 月 22 日
新增了對歐洲 (愛爾蘭) 的支援。	新增了對歐洲 (愛爾蘭) 的支援。如需詳細資訊，請參閱 <a href="#">AWS 區域 和端點</a> 。	2017 年 6 月 8 日
添加了 Amazon Athena API 和 AWS CLI 支持。	添加了 Amazon Athena API 和對 Athena 的 AWS CLI 支持。將 JDBC 驅動程式更新至 1.1.0 版。	2017 年 5 月 19 日
新增了對 Simple Storage Service (Amazon S3) 資料加密的支援。	新增了對 Simple Storage Service (Amazon S3) 資料加密的支援，並發行了一項 JDBC 驅動程式更新 (1.0.1 版) 來提供加密支援、改進功能和錯誤修正。如需詳細資訊，請參閱 <a href="#">靜態加密</a> 。	2017 年 4 月 4 日
添加了 AWS CloudTrail SerDe.	<p>添加了 AWS CloudTrail SerDe，改進的性能，固定的分區問題。</p> <ul style="list-style-type: none"> <li>• AWS CloudTrail SerDe 已由讀取記錄檔所取 <a href="#">蜂巢 SerDe</a> 代。CloudTrail 如需查詢 CloudTrail 記錄檔的詳細資訊，請參閱 <a href="#">查詢 AWS CloudTrail 記錄檔</a>。</li> <li>• 改進了掃描大量分割區時的效能。</li> <li>• 改進了 MSCK Repair Table 操作的效能。</li> <li>• 增加了查詢主要區域以外的區域中存放的 Simple Storage Service (Amazon S3) 資料的功能。除了標準 Athena 費用，也要支付 Simple Storage Service (Amazon S3) 的標準跨區域資料傳輸費用。</li> </ul>	2017 年 3 月 24 日
新增了對美國東部 (俄亥俄) 的支援。	新增了對 <a href="#">Avro SerDe</a> 和 <a href="#">SerDe 用於處理 CSV 的開啟 CSV</a> 、美國東部 (俄亥俄) 和在主控台精靈中大量編輯資料欄的支援。改進大型 Parquet 資料表的效能。	2017 年 2 月 20 日
	《Amazon Athena 使用者指南》的初始版本。	2016 年 11 月

# AWS 詞彙表

有關最新 AWS 術語，請參閱AWS 詞彙表 參考文獻中的[AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。