aws

開發人員指南

Amazon OpenSearch Service

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

Amazon OpenSearch Service: 開發人員指南

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

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

Table of Contents

什麼是 Amazon OpenSearch Service？ 1
Amazon Opensearch Service 的功能 1
使用情況 3
支援的 Elasticsearch 和 OpenSearch 版本 3
標準和延伸支援 4
標準支援和擴充版本 6
計算延伸支援費用 6
定價 7
相關服務 7
設定 10
授予許可 10
授與程式設計存取權 10
設定 AWS CLI 12
開啟 主控台 13
開始使用 14
建立網域 14
上傳資料以編製索引 16
選項 1：上傳單一文件 16
選項 2：上傳多個文件 16
搜尋文件 17
從命令列搜尋文件 17
使用 OpenSearch Dashboards 搜尋文件 19
刪除網域 19
Amazon OpenSearch Ingestion 20
重要概念 20
優勢 21
限制 22
支援的資料準備版本 22
擴展管道 23
定價 25
支援的 AWS 區域 25
設定角色和使用者 25
管道角色 26
擷取角色 29

授予管道對網域的存取權 31
授予管道對集合的存取權 34
OpenSearch Ingestion 入門 39
教學課程：將資料擷取至網域 39
教學課程：將資料擷取至集合 47
管道功能 53
持久性緩衝 54
分割 55
鏈接 56
無效信件佇列 57
索引管理 59
End-to-end認可 62
來源背壓 62
建立管道 63
先決條件和必要的 IAM 角色 64
所需的 IAM 許可 64
指定管道版本 66
指定擷取路徑 66
建立管道 67
追蹤管道建立的狀態 71
使用藍圖 72
檢視管道 74
更新管道 76
考量事項 77
必要許可 77
更新管道 78
管道更新的藍/綠部署 79
管理管道成本 79
停止管道 80
啟動管道 81
刪除管道 82
支援的外掛程式和選項 83
支援的外掛程式 83
無狀態與有狀態處理器 85
組態需求和限制條件 86
整合管道 91

建構擷取端點 91
建立擷取角色 92
Atlassian Services 94
Amazon Aurora 110
Amazon DynamoDB 137
Amazon DocumentDB 149
Confluent Cloud Kafka 164
Amazon MSK 174
Amazon RDS 181
Amazon S3 207
Amazon Security Lake 216
Fluent Bit 220
Fluentd 222
ML 離線批次推論 224
OpenTelemetry Collector 239
Amazon Managed Service for Prometheus 245
自我管理的 Kafka 252
自我管理的 OpenSearch 叢集 259
Amazon Kinesis Data Streams 267
後續步驟 278
AWS Lambda 278
在網域和集合之間遷移資料 282
限制 282
OpenSearch Service 做為來源 283
指定多個 OpenSearch Service 網域目的地 285
將資料遷移至 OpenSearch Serverless VPC 集合 285
使用 AWS SDKs 管理管道 286
Python 286
OpenSearch 擷取的安全性 290
設定管道的 VPC 存取 291
設定跨帳戶擷取的管道 295
身分和存取權管理 301
使用 CloudTrail 進行監控 308
VPC 端點 (AWS PrivateLink) 311
標記管道 314
必要許可 314

處理標籤 (主控台) 315
處理標籤 (AWS CLI) 315
日誌記錄和監控 316
監控管道日誌 316
監控管道指標 318
最佳實務 345
一般最佳實務 345
建議 CloudWatch 警示 346
Amazon OpenSearch Serverless 352
優勢 352
什麼是 Amazon OpenSearch Serverless？ 352
OpenSearch Serverless 使用案例 353
運作方式 353
選擇集合類型 355
定價 356
支援的 AWS 區域 357
限制 357
比較 OpenSearch Service 和 OpenSearch Serverless 358
教學課程：開始使用 OpenSearch Serverless 360
步驟 1：設定許可 361
步驟 2：建立集合 362
步驟 3：上傳並搜尋資料 363
處理錯誤 364
步驟 4：刪除集合 365
後續步驟 365
集合 365
管理集合 366
使用向量搜尋集合 387
使用資料生命週期政策 394
使用 AWS SDKs 管理集合 402
使用 CloudFormation 建立集合 413
使用快照備份集合 415
Amazon OpenSearch Serverless 中的 Zstandard Codec 支援 421
使用衍生來源儲存儲存體 423
集合群組 424
重要概念 425

集合群組容量限制 425
集合群組中的加密和 KMS 金鑰 428
建立集合群組 429
管理集合群組 432
管理容量限制 434
進行容量設定 436
容量限制上限 437
監控容量用量 437
將資料擷取至集合 438
所需的最低許可 438
OpenSearch 擷取 439
Fluent Bit 439
Amazon Data Firehose 440
Go 441
Java 443
JavaScript 444
Logstash 447
Python 449
Ruby 451
其他用戶端 452
設定Machine Learning 453
機器學習 453
連接器 453
模型 454
設定Machine Learning的許可 454
不支援APIs 和功能 455
設定神經和混合搜尋 456
神經搜尋 456
混合搜尋 457
神經和混合查詢 457
設定 許可 458
設定工作流程 461
工作流程 461
設定 許可 461
OpenSearch Serverless 的安全功能 462
加密政策 464

網路政策 465
資料存取政策 465
IAM 和 SAML 身分驗證 466
基礎設施安全性 467
開始使用安全功能 467
身分和存取權管理 481
加密 497
網路存取 506
FIPS 合規 517
資料存取控制 521
資料平面 VPC 端點 531
控制平面 VPC 端點 541
SAML 身分驗證 544
法規遵循驗證 553
標記集合 553
必要許可 553
標記集合 （主控台） 553
標記集合 (AWS CLI) 554
支援的操作和外掛程式 555
支援的 OpenSearch API 操作與許可 555
支援的 OpenSearch 外掛程式 564
監控 OpenSearch Serverless 565
使用 CloudWatch 進行監控 566
使用 CloudTrail 進行監控 571
使用 EventBridge 進行監控 576
建立和管理網域 580
建立 OpenSearch Service 網域 580
建立 OpenSearch Service 網域 (主控台) 580
建立 OpenSearch Service 網域 (AWS CLI) 586
建立 OpenSearch Service 網域 (AWS SDKs) 587
建立 OpenSearch Service 網域 (AWS CloudFormation) 588
設定存取政策 588
使用遷移助理遷移至 OpenSearch Service 588
概觀 588
遷移案例 589
遷移助理入門 590

解決方案架構 590
進階叢集設定 591
自動語意擴充 591
簡介 591
模型詳細資訊和效能基準 592
支援的語言 592
設定網域的自動語意擴充索引 593
資料擷取和搜尋 594
設定自動語意擴充的許可 595
查詢重寫 598
自動語意擴充的限制 598
定價 598
組態變更 599
通常會導致藍/綠部署的變更 599
通常不會導致藍/綠部署的變更 600
藍/綠部署選項 601
判斷變更是否會導致藍/綠部署 603
追蹤組態變更 607
組態變更的階段 609
藍/綠部署的效能影響 610
組態變更的費用 611
對驗證錯誤進行疑難排解 611
服務軟體更新 615
選用與必要的更新 616
修補程式更新 617
考量事項 617
開始更新 617
離峰時段 620
監控更新 621
網域不符合更新條件時， 622
離峰時段 622
離峰服務軟體更新 623
離峰自動調整最佳化 624
啟用離峰時段 624
設定自訂離峰時段 625
檢視排程動作 625

重新排程動作 ... 627
從自動調整維護時段遷移 ... 629
通知 ... 630
開始使用通知 ... 630
通知嚴重性 ... 631
EventBridge 事件範例 ... 631
設定多可用區域網域 ... 632
具有待命的異地同步備份 ... 632
無待命的異地同步備份 ... 634
可用區域中斷 ... 637
VPC 支援 ... 639
VPC 與公有網域 ... 639
限制 ... 639
Architecture ... 640
建立索引快照 ... 646
先決條件 ... 646
註冊手動快照儲存庫 ... 650
手動拍攝快照 ... 655
還原快照 ... 656
刪除手動快照 ... 658
使用快照管理自動化快照 ... 659
使用索引狀態管理自動化快照 ... 661
使用 Curator 進行快照 ... 661
升級網域 ... 661
支援的升級路徑 ... 662
升級網域 （主控台） ... 665
升級網域 (CLI) ... 665
升級網域 (SDK) ... 666
對驗證失敗進行故障排除 ... 668
升級疑難排解 ... 668
使用快照來遷移資料 ... 670
建立自訂端點 ... 675
新網域的自訂端點 ... 676
現有網域的自訂端點 ... 676
CNAME 映射 ... 677
自動調校 ... 677

變更類型 677
啟用或停用自動調整 678
排程自動調整增強功能 679
監控自動調整變更 680
標記網域 680
標記範例 681
標記網域 （主控台） 681
標記網域 (AWS CLI) 682
標記網域 (AWS SDKs) 683
執行管理動作 685
在資料節點上重新啟動 OpenSearch 程序 685
重新啟動資料節點 685
重新啟動儀表板程序 686
限制 687
使用直接查詢 688
定價 688
限制 689
一般限制 689
Amazon S3 的限制 689
Amazon CloudWatch Logs 的限制 689
Amazon Security Lake 的限制 690
建議 690
一般建議 690
Amazon S3 建議 691
CloudWatch Logs 建議 692
Security Lake 建議 692
配額 693
Amazon S3 的配額 693
CloudWatch Logs 的配額 694
Security Lake 的配額 695
支援的 AWS 區域 695
AWS 區域 適用於 Amazon S3 695
AWS 區域 適用於 CloudWatch Logs 696
AWS 區域 適用於 Security Lake 697
S3 中的直接查詢 697
建立 S3 資料來源 698

設定 S3 資料來源 705
CloudWatch Logs 中的直接查詢 708
建立 CloudWatch Logs 資料來源 708
設定 CloudWatch Logs 資料來源 713
Security Lake 中的直接查詢 715
建立 Security Lake 資料來源 715
設定 Security Lake 資料來源 721
管理資料來源 723
使用 CloudWatch 指標資料來源進行監控 723
啟用和停用資料來源 726
使用 AWS Budget 監控 727
刪除資料來源 727
最佳化查詢效能 728
略過索引 728
具體化視觀表 729
涵蓋索引 729
支援的 SQL 和 PPL 命令 729
支援的 SQL 命令 730
支援的 PPL 命令 918
監控網域 1092
監控叢集指標 1093
在 CloudWatch 中檢視指標 1094
解讀 OpenSearch Service 中的運作狀態圖表 1094
叢集指標 1095
專用主節點指標 1101
專用協調器節點指標 1102
EBS 磁碟區指標 1103
執行個體指標 1105
暖指標 1116
冷儲存指標 1122
OpenSearch Optimized Instances (OR1) 指標 1123
提醒指標 1124
異常偵測指標 1125
非同步搜尋指標 1126
自動調整指標 1128
具有待命指標的異地同步備份 1129

時間點指標 1131
SQL 指標 1132
k-NN 指標 1133
跨叢集搜尋指標 1136
跨叢集複寫指標 1136
Learning to Rank 指標 1138
Piped Processing Language 指標 1139
監控日誌 1139
啟用日誌發佈 (主控台) 1141
啟用日誌發佈 (AWS CLI) 1142
啟用日誌發佈 (AWS SDKs) 1144
啟用日誌發佈 (CloudFormation) 1145
設定搜尋請求慢速日誌閾值 1147
設定碎片慢速日誌閾值 1147
測試慢速日誌 1148
檢視日誌 1148
監控稽核日誌 1149
限制 1149
啟用稽核日誌 1150
使用 啟用稽核記錄 AWS CLI 1151
使用組態 API 啟用稽核日誌記錄 1152
稽核日誌層和類別 1152
稽核日誌設定 1154
稽核日誌範例 1157
使用 REST API 設定稽核日誌 1160
監控事件 1161
服務軟體更新事件 1162
自動調整事件 1169
叢集運作狀態事件 1173
VPC 端點事件 1186
節點淘汰事件 1188
已降級的節點淘汰事件 1190
網域錯誤事件 1192
教學課程：接聽 OpenSearch Service 事件 1194
教學課程：傳送可用更新的 SNS 提醒 1196
使用 CloudTrail 進行監控 1198

CloudTrail 中的 Amazon OpenSearch Service 資訊 572
了解 Amazon OpenSearch Service 日誌檔案項目 575
叢集洞見 1202
優勢 1202
透過主控台存取 Cluster Insights 1202
透過 OpenSearch UI 存取 Cluster Insights 和詳細指標 1203
建立和設定應用程式以檢視 Cluster Insights 1203
了解 Cluster Insights 1205
概觀儀表板 1206
目前叢集狀態 1206
Insights 趨勢 1206
目前開放的洞見 1207
OpenSearch Service 叢集 1207
依嚴重性列出的最佳洞見 1207
洞見詳細資訊 1207
叢集詳細資訊 1207
概觀 1207
設定最佳實務 1207
洞見 1207
指標區段 1209
節點、索引和碎片檢視 1209
查詢檢視 1210
透過 Amazon EventBridge 事件存取 Insights 1212
Insights 目錄 1212
安全 1215
資料保護 1216
靜態加密 1216
節點對節點加密 1220
身分和存取權管理 1220
政策的類型 1221
提出並簽署 OpenSearch Service 請求 1229
當政策衝突時 1230
政策元素參考 1231
進階選項和 API 考量 1236
設定存取政策 1239
其他範例政策 1239

API 許可參考 1239
AWS 受管政策 1240
預防跨服務混淆代理人 1248
精細定義存取控制 1249
宏觀：精細存取控制和 OpenSearch Service 安全 1250
重要概念 1253
關於主要使用者 1253
啟用精細存取控制 1254
以主要使用者身分存取 OpenSearch Dashboards 1257
管理許可 1259
建議的組態 1264
限制 1267
修改主要使用者 1268
其他主要使用者 1269
手動快照 1271
整合 1271
REST API 差異 1272
教學課程：使用 Cognito 身分驗證進行精細存取控制 1273
教學課程：內部使用者資料庫和基本身分驗證 1278
法規遵循驗證 1281
恢復能力 1281
JSON Web 權杖 1282
考量事項 1282
修改網域存取政策 1282
設定 JWT 身分驗證和授權 1283
使用 JWT 傳送測試請求 1284
基礎設施安全性 1285
使用 OpenSearch Service 受管 VPC 端點 1286
OpenSearch Dashboards 的 SAML 身分驗證 1289
SAML 組態概觀 1290
考量事項 1290
VPC 網域的 SAML 身分驗證 1291
修改網域存取政策 1291
設定 SP 或 IdP 啟動的身分驗證 1292
同時設定 SP 和 IdP 啟動的身分驗證 1298
設定 SAML 身分驗證 (AWS CLI) 1299

設定 SAML 身分驗證 (組態 API) 1299
SAML 疑難排解 1300
停用 SAML 身分驗證 1302
OpenSearch 的 IAM Identity Center 支援 1303
考量事項 1303
修改網域存取政策 1303
設定 IAM Identity Center 身分驗證和授權 （主控台） 1304
設定精細存取控制 1305
設定 IAM Identity Center 身分驗證和授權 (CLI) 1305
在網域上停用 IAM Identity Center 身分驗證 1305
OpenSearch Dashboards 的 Amazon Cognito 身分驗證 1306
先決條件 1307
設定網域以使用 Amazon Cognito 身分驗證 1309
允許已經過身分驗證的角色 1313
設定身分提供者 1314
(選用) 設定精細分級的存取 1315
(選用) 自訂登入頁面 1316
(選用) 設定進階安全性 1316
測試 1316
配額 1316
常見的設定問題 1317
停用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證 1320
刪除使用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證的網域 1321
使用服務連結角色 1321
VPC 網域和資料來源建立角色 1321
集合建立角色 1324
管道建立角色 1326
範本程式碼 1330
Elasticsearch 用戶端相容性 1330
壓縮 HTTP 請求 1331
啟用 gzip 壓縮 1331
必要標頭 1332
範本程式碼 (Python 3) 1332
使用 AWS SDKs 1333
Java 1333
Python 1345

節點 1347
建立資料索引 1351
索引的命名限制 1351
縮減回應大小 1352
索引轉碼器 1353
將資料串流載入至 OpenSearch Service 1354
從 OpenSearch Ingestion 載入串流資料 1354
從 Amazon S3 載入串流資料 1354
從 Amazon Kinesis Data Streams 中載入串流資料 1360
從 Amazon DynamoDB 中載入串流資料 1364
從 Amazon Data Firehose 載入串流資料 1368
從 Amazon CloudWatch 中載入串流資料 1368
從 載入串流資料 AWS IoT 1368
使用 Logstash 載入資料 1368
Configuration 1368
搜尋資料 1372
URI 搜尋 1372
要求主體搜尋 1374
增加欄位 1375
搜尋結果反白呈現 1376
計數 API 1378
對搜尋結果進行分頁 1378
時間點 1378
from 和 size 參數 1379
儀表板查詢語言 1379
套件 1381
所需的許可 1382
將套件上傳至 Amazon S3 1382
匯入和關聯套件 1383
搭配 OpenSearch 使用套件 1383
更新套件 1388
使用新的字典手動更新索引 1391
解除關聯並移除套件 1393
自訂外掛程式 1394
第三方外掛程式 1409
SQL 支援 1412

範例呼叫 1414
備註和差異 1414
SQL Workbench 1415
SQL CLI 1284
JDBC 驅動程式 1415
ODBC 驅動程式 1415
跨叢集搜尋 1415
限制 1416
跨叢集搜尋先決條件 1417
跨叢集搜尋定價 1417
設定連線 1417
移除連線 1419
設定安全性和範例演練 1419
OpenSearch Dashboards 1425
Learning to Rank 1425
Learning to Rank 入門 1426
Learning to Rank API 1447
非同步搜尋 1453
搜尋呼叫範例 1453
非同步搜尋許可 1455
非同步搜尋設定 1455
跨叢集搜尋 1455
UltraWarm 1457
時間點 1458
考量事項 1458
建立 PIT 1458
時間點許可 1460
PIT 設定 1461
跨叢集搜尋 1461
UltraWarm 1461
代理搜尋 1461
語意搜尋 1462
並行區段搜尋 1462
產生自然語言查詢 1463
先決條件 1463
開始使用 1463

設定 許可 1463
組態自動化 1464
向量搜尋 1465
從 Amazon S3 向量匯入 OpenSearch Serverless 1466
整合架構 1467
所需的 IAM 許可 1468
設定 Amazon S3 Vectors 與 OpenSearch 的整合 1471
使用 Amazon S3 向量引擎的進階搜尋功能 1473
啟用 S3 向量 1473
使用 S3 向量引擎建立索引 1473
功能限制 1476
索引文件 1477
搜尋文件 1477
支援的映射參數 1479
計量和計費 1480
停用 s3vector 引擎 1480
k-NN 搜尋 1480
k-NN 入門 1480
k-NN 差異、調校和限制 1483
向量擷取 1483
運作方式 1483
OpenSearch Vector 擷取定價 1484
先決條件 1484
建立向量資料庫 1484
相關功能 1485
將 Amazon S3 向量索引匯出至 OpenSearch Service 向量引擎 1485
將 Amazon S3 向量命名空間匯入 OpenSearch Service 向量引擎 1486
檢視向量擷取任務和匯入歷史記錄 1487
自動最佳化 1489
優勢 1489
運作方式 1489
先決條件 1490
自動最佳化的使用案例 1491
限制 1491
帳單與成本 1492
支援的資料格式 1493

在主控台中使用自動最佳化 1495
相關功能 1501
向量索引的 GPU 加速 1501
先決條件 1501
運作方式 1501
GPU 加速定價 1502
GPU 加速和寫入操作 1502
支援的索引組態 1503
支援的 AWS 區域 1503
啟用 GPU 加速 1503
建立 GPU 加速向量索引 1506
索引和強制合併 1508
最佳實務 1508
OpenSearch Dashboards 1509
控制對 Dashboards 的存取 1509
使用代理伺服器從 Dashboards 中存取 OpenSearch Service 1510
設定儀表板以使用 WMS 地圖伺服器 1514
將本機 Dashboards 伺服器連線至 OpenSearch Service 1515
在儀表板中管理索引 1516
額外功能 1517
OpenSearch UI 1518
版本歷史記錄 1519
開始使用 1520
建立 Amazon OpenSearch Service 應用程式所需的許可 1521
建立應用程式 1524
管理應用程式管理員 1531
代理式 AI 1534
關鍵功能 1535
定價和用量限制 1535
安全 1536
啟用和停用代理式 AI 功能 1536
區域可用性 1537
客服人員聊天 1537
調查代理程式 1541
代理程式記憶體 1544
加密和客戶受管金鑰 1545

使用客戶受管金鑰的先決條件 1546
使用 主控台建立具有客戶受管金鑰加密的應用程式 1547
使用 建立具有客戶受管金鑰加密的應用程式 AWS CLI 1548
監控客戶受管金鑰用量 1548
更新加密設定 1549
使用 IAM 啟用 SAML 聯合 1549
步驟 1：設定身分提供者應用程式 (Okta) 1550
步驟 2：設定 Okta 的 AWS 組態 1553
步驟 3：在 IAM 中建立 Amazon OpenSearch Service 存取政策 1555
步驟 4：使用 SAML 驗證身分提供者起始的單一登入體驗 1557
步驟 5：設定以 SAML 屬性為基礎的精細存取控制 1561
管理資料來源關聯和 VPC 存取許可 1564
將資料來源與 OpenSearch UI 應用程式建立關聯 1564
管理對 VPC 中網域的存取 1565
設定對 VPC 中 OpenSearch Serverless 集合的存取 1567
使用 Amazon OpenSearch Service 工作區 1570
建立 OpenSearch UI 應用程式工作區 1570
工作區隱私權和協作者 1571
工作區類型 1571
跨區域和跨帳戶資料存取 1572
OpenSearch 網域的跨帳戶資料存取 1573
跨叢集搜尋 1594
從 VPC 端點管理對 OpenSearch UI 的存取 1603
在 VPC 和 OpenSearch UI 之間建立私有連線 1603
更新 VPC 端點政策以允許存取 OpenSearch UI 應用程式 1605
在 VPC 端點政策中撤銷對 OpenSearch UI 的存取 1605
從 OpenSearch Dashboard 遷移 1606
遷移概觀 1606
先決條件 1607
步驟 1：從 OpenSearch Dashboards 匯出儲存的物件 1607
步驟 2：將儲存的物件匯入 OpenSearch UI 1610
端點和配額 1611
OpenSearch UI 端點 1611
OpenSearch UI 服務配額 1614
管理索引 1616
OpenSearch 最佳化執行個體 1616

限制 1617
調校以獲得更好的擷取輸送量 1617
OpenSearch 最佳化執行個體與其他執行個體的差異 1617
OpenSearch 最佳化執行個體與 UltraWarm 執行個體有何不同 1618
使用 OpenSearch 最佳化執行個體佈建網域 1618
多層儲存 1620
多層儲存架構 1621
UltraWarm 儲存 1627
先決條件 1627
UltraWarm 儲存要求和效能考量 1629
UltraWarm 定價 1630
啟用 UltraWarm 1630
將索引遷移到 UltraWarm 儲存 1632
自動化遷移 1635
遷移調整 1635
取消遷移 1636
列出熱索引和暖索引 1636
將暖索引傳回熱儲存區 1636
從快照還原暖索引 1637
暖索引的手動快照 1638
將暖索引遷移到冷儲存 1639
KNN 索引的最佳實務 1639
停用 UltraWarm 1640
冷儲存 1640
先決條件 1641
冷儲存要求和效能考量 1642
冷儲存定價 1642
啟用冷儲存 1643
管理 OpenSearch Dashboards 中的冷索引 1645
將索引遷移至冷儲存 1645
自動移轉至冷儲存 1646
取消遷移至冷儲存 1646
列出冷索引 1647
將冷索引遷移至暖儲存 1650
從快照中還原冷索引 1652
取消從冷儲存遷移至暖儲存 1652

更新冷索引中繼資料 1653
刪除冷索引 1653
停用冷儲存 1653
索引狀態管理 1654
建立 ISM 政策。 1654
範例政策 1655
ISM 範本 1659
差異 1660
教學課程：自動化 ISM 程序 1661
索引彙總 1665
建立索引彙總任務 1666
索引轉換 1667
建立索引轉換任務 1667
跨叢集複寫 1668
限制 1669
先決條件 1670
許可需求 1670
設定跨叢集連線 1672
開始複寫 1672
確認複寫 1673
暫停和繼續複寫 1674
停止複寫 1675
自動追蹤 1675
升級連線的網域 1677
遠端重新索引 1677
先決條件 1677
在 OpenSearch Service 網際網路網域之間重新索引資料 1678
遠端網域位於 VPC 時重新索引資料 1679
在非 OpenSearch Service 網域之間重新建立資料索引 1683
重新索引大型資料集 1684
遠端重新索引設定 1685
資料串流 1686
資料串流入門 1686
監控資料 1690
提醒 1690
提醒許可 1691

提醒入門 1691
通知 1692
差異 1692
異常偵測 1694
先決條件 1694
開始使用異常偵測 1695
教學課程：使用異常偵測來偵測高 CPU 使用率 1697
AI 助理 1700
支援的 AWS 區域 1700
設定適用於 OpenSearch Service 的 AI 助理 1701
使用自然語言產生視覺化效果 1701
檢視提醒摘要和洞見 1702
開始之前 1705
檢視提醒摘要和洞見 1706
在探索頁面上檢視 AI 助理產生的查詢結果摘要 1706
檢視建議的異常偵測器 1708
存取適用於 OpenSearch Service 問題的 AI 助理聊天 1708
機器學習 1711
的連接器 AWS 服務 1711
先決條件 1711
建立 OpenSearch Service 連接器 1715
外部平台的連接器 1717
先決條件 1717
建立 OpenSearch Service 連接器 1720
CloudFormation 範本整合 1722
可用的 CloudFormation 範本 1723
先決條件 1724
Amazon Bedrock 範本 1725
使用 Bedrock Claude 設定代理程式搜尋 1726
MCP 伺服器整合範本 1727
Amazon SageMaker 範本 1730
語意反白範本的遠端推論 1731
不支援的 ML Commons 設定 1732
流程架構外掛程式 1732
在 OpenSearch Service 中建立 ML 連接器 1733
設定 許可 1740

安全分析 1741
安全分析元件和概念 1741
日誌類型 1741
偵測器 1742
Rules 1742
調查結果 1742
Alerts (提醒) 1742
探索安全分析 1742
設定 許可 1744
疑難排解 1746
沒有此類索引錯誤 1746
可觀測性 1747
OpenSearch Service 中的可觀測性工作流程 1747
OpenSearch UI 和 OpenSearch Dashboards 1748
日誌 1749
OpenSearch UI 和可觀測性工作區 1750
使用 PPL 查詢日誌 1750
使用 AI 查詢您的日誌 1751
使用 SQL 查詢您的日誌 1751
使用 DQL 查詢您的日誌 1752
日誌的儀表板和提醒 1752
追蹤 1753
追蹤資料處理和擷取 1753
OpenSearch UI 和可觀測性工作區 1750
主要功能 1753
最佳實務 1757
監控和提醒 1757
設定 CloudWatch 警示 1757
啟用日誌發佈 1757
碎片策略 1758
確定碎片和資料節點計數 1758
避免儲存扭曲 1759
穩定性 1759
與 OpenSearch 保持同步 1759
改善快照效能 1760
啟用專用主節點 1760

在多個可用區域中進行部署 1760
控制擷取流量和緩衝 1761
建立搜尋工作負載的映射 1761
使用索引範本 1762
使用索引狀態管理功能來管理索引 1763
移除未使用的索引 1763
使用多個網域實現高可用性 1763
效能 1763
最佳化批量請求大小和壓縮 1763
減少批量請求回應的大小 1764
調校重新整理間隔 1764
啟用自動調整 1764
安全 1765
啟用精細存取控制 1765
在 VPC 內部署網域 1765
套用限制性存取政策 1765
啟用靜態加密 1765
啟用節點對節點加密 1766
使用 監控 AWS Security Hub CSPM 1766
成本最佳化 1766
使用最新一代執行個體類型 1766
使用最新的 Amazon EBS gp3 磁碟區 1766
使用 UltraWarm 和冷儲存來處理時間序列日誌資料 1767
檢閱預留執行個體的建議 1767
建議 CloudWatch 警示 1767
您可能會考慮的其他警示 1771
調整網域大小 1773
計算儲存需求 1774
選擇碎片數 1775
選擇執行個體類型並進行測試 1776
PB 規模 1778
專用協調器節點 1779
何時使用專用協調器節點 1779
架構和行為 1779
要求與限制 1780
佈建專用協調器節點 1780

最佳實務 1781
專用主節點 1782
選擇專用主節點數目 1783
選擇專用主節點的執行個體類型 1784
一般參考 1786
支援的執行個體類型 1786
最新一代執行個體類型 1786
上一代執行個體類型 1804
各引擎版本的功能 1815
依引擎版本分類的外掛程式 1819
選用的外掛程式 1822
受支援的操作 1823
值得注意的 API 差異 1823
配額 1874
暖節點儲存配額 1874
每個 AZ 的資料節點數量 1784
執行個體系列的總節點限制 1875
EBS 磁碟區大小配額 1876
網路配額 1888
碎片大小配額 1898
碎片計數配額 1899
Java 處理序配額 1899
網域政策配額 1900
預留執行個體 1900
購買預留執行個體 (主控台) 1900
購買預留執行個體 (AWS CLI) 1901
購買預留執行個體 (AWS SDKs) 1904
檢查成本 1905
成本最佳化 1906
Amazon OpenSearch Service 受管叢集的成本最佳化 1906
Amazon OpenSearch Service Serverless 的成本最佳化 1910
其他受支援的資源 1913
教學 1915
建立和搜尋文件 1915
先決條件 1915
將文件新增至索引 1916

建立自動產生的 ID 1917
使用 POST 命令更新文件 1918
執行大量動作 1919
搜尋文件 1920
相關資源 1921
遷移至 OpenSearch Service 1922
建立並上傳快照 1922
建立網域 1923
提供許可以存取 S3 儲存貯體。 1924
還原快照 1926
建立搜尋應用程式 1929
先決條件 1930
步驟 1：索引範例資料 1930
步驟 2：建立和部署 Lambda 函數 1931
步驟 3：在 API Gateway 中建立 API 1933
步驟 4：(選用) 修改網域存取政策 1936
映射 Lambda 角色 (如果使用精細存取控制) 1938
步驟 5：測試 Web 應用程式 1939
後續步驟 1941
視覺化支援呼叫 1941
步驟 1：設定先決條件 1942
步驟 2：複製範本程式碼 1943
(選用) 步驟 3：索引範例資料 1948
步驟 4：分析和視覺化您的資料 1949
步驟 5：清除資源和後續步驟 1953
Amazon OpenSearch Service 重新命名 1954
新的 API 版本 1954
已重新命名的執行個體類型 1954
存取政策變更 1954
IAM 政策 1955
SCP 政策 1955
新資源類型 1955
Kibana 已更名為 OpenSearch Dashboards 1956
已重新命名的 CloudWatch 指標 1957
「帳單和成本管理」主控台的變更 1958
新事件格式 1959

什麼保持不變？ 1959
開始使用：將您的網域升級至 OpenSearch 1.x 1959
疑難排解 1961
無法存取 OpenSearch Dashboards 1961
無法存取 VPC 網域 1961
叢集處於唯讀狀態 1961
紅色叢集狀態 1962
紅色叢集的自動修復 1964
從持續繁重的處理負載中復原 1964
黃色叢集狀態 1966
ClusterBlockException 1966
缺少可用儲存空間 1966
高 JVM 記憶體壓力 1966
使用待命遷移至異地同步備份時發生錯誤 1967
從沒有待命的網域遷移到有待命的網域期間建立索引、索引範本或 ISM 政策 1746
不正確的資料副本數量 1967
JVM OutOfMemoryError 1967
叢集節點失敗 1968
超出最大碎片限制 1969
網域卡在處理狀態 1969
低 EBS 爆量餘額 1969
在磁碟區調整大小期間增加 EBS 指標 1970
無法啟用稽核日誌 1970
無法關閉索引 1970
用戶端授權檢查 1971
請求調節 1971
無法對節點執行 SSH 1971
「無效的物件存放區類別」快照錯誤 1971
無效主機標頭 1971
無效的 M3 執行個體類型 1972
熱查詢在啟用 UltraWarm 後停止運作 1972
升級後無法降級 1972
需要所有 的網域摘要 AWS 區域 1972
使用 OpenSearch Dashboards 時發生瀏覽器錯誤 1973
節點碎片和儲存扭曲 1973
索引碎片和儲存扭曲 1974

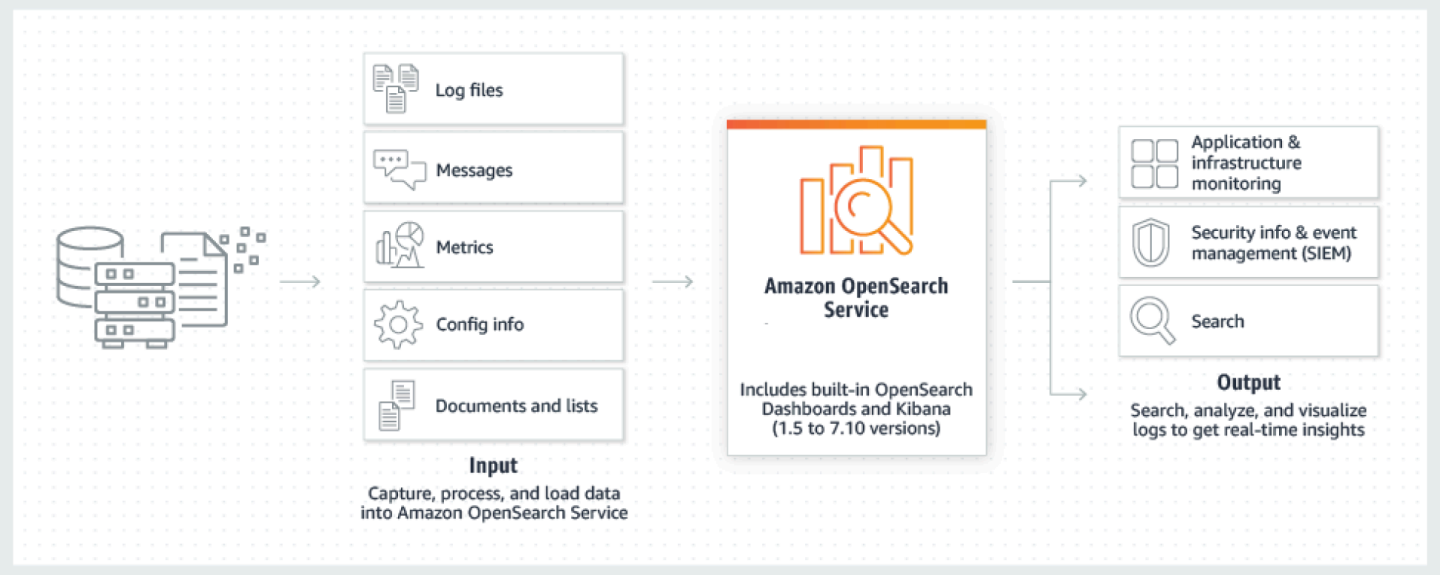
在選取 VPC 存取後未經授權的操作 1975
在建立 VPC 網域後載入停滯 1975
對 OpenSearch API 的請求遭拒 1975
無法從 Alpine Linux 連線 1976
搜尋背壓的請求過多 1976
使用開發套件時發生憑證錯誤 1977
由於版本相容性，自訂外掛程式安裝失敗 1978
文件歷史紀錄 1980
舊版更新 2022
AWS 詞彙表 2025
..... mmxxvi

什麼是 Amazon OpenSearch Service？

Amazon OpenSearch Service 是一項受管服務，可讓您在 AWS 雲端中輕鬆部署、操作和擴展 OpenSearch 叢集。OpenSearch Service 網域與 OpenSearch 叢集同義。網域是指具有您指定之設定、執行個體類型、執行個體計數和儲存資源的叢集。Amazon OpenSearch Service 支援 OpenSearch 和舊版 Elasticsearch OSS (最高為 7.10，該軟體的最終開放原始碼版本)。建立網域時，您可以選擇要使用哪個搜尋引擎。

OpenSearch 是一個完全開源的搜尋和分析引擎，適用於例如日誌分析、即時應用程式監控及點擊流分析等使用案例。如需詳細資訊，請參閱 OpenSearch 文件。

Amazon OpenSearch Service 會為您的 OpenSearch 叢集佈建所有資源並啟動它。它還會自動偵測和更換發生故障的 OpenSearch Service 節點，可降低與自我管理的基礎設施相關的營運成本。您只需單一的 API 呼叫或在主控台中按幾下，就可以擴展您的叢集。



若要開始使用 OpenSearch Service，請建立一個 OpenSearch Service 網域，它等同於 OpenSearch 叢集。叢集中的每個 EC2 執行個體都可作為一個 OpenSearch Service 節點。

您可以使用 OpenSearch Service 主控台在幾分鐘內設定和配置您的網域。如果您偏好程式設計存取，您可以使用 AWS CLI、 AWS SDKs 或 Terraform。

Amazon Opensearch Service 的功能

OpenSearch Service 包含下列功能：

擴展

- CPU、記憶體和儲存容量的許多組態，稱為執行個體類型，包括符合成本效益的 Graviton 執行個體
- 支援最多 1002 個資料節點
- 最多 25 PB 的連接儲存
- 符合成本效益的 UltraWarm 和冷儲存，適用於唯讀資料

安全性

- AWS Identity and Access Management (IAM) 存取控制
- 輕鬆整合 Amazon VPC 和 VPC 安全群組
- 靜態資料加密和節點對節點加密
- OpenSearch Dashboards 的 Amazon Cognito、HTTP 基本或 SAML 身分驗證
- 索引層級、文件層級，以及欄位層級安全
- 稽核日誌
- Dashboards 多租用

穩定性

- 適用於您資源的多個地理位置，也稱為區域和可用區域
- 在相同 AWS 區域跨兩個或三個可用區域的節點分配，也稱為多可用區域
- 卸載叢集管理任務用的專用主節點
- 已自動化快照以備份和還原 OpenSearch Service 網域

彈性

- SQL 支援與商業智慧 (BI) 應用程式整合
- 自訂套件以改善搜尋結果

與熱門服務整合

- 使用 OpenSearch Dashboards 的資料視覺化
- 與 Amazon CloudWatch 整合以監控 OpenSearch Service 網域指標和設定警示
- 與 整合 AWS CloudTrail ，以稽核對 OpenSearch Service 網域的組態 API 呼叫

- 與 Simple Storage Service (Amazon S3)、Amazon Kinesis 和 Amazon DynamoDB 整合，將串流資料載入到 OpenSearch Service
- 當您的資料超過特定閾值時 Amazon SNS 發出的提醒

何時使用 OpenSearch 與 Amazon OpenSearch Service

使用下表協助您決定佈建的 Amazon OpenSearch Service 或自我管理的 OpenSearch 是否為適合您的選擇。

OpenSearch	Amazon OpenSearch Service
• 您的組織願意且讓具備正確技能的人員手動監控和維護自行佈建的叢集。 • 您想要完整、編譯層級的程式碼控制。 • 您的組織偏好或唯一使用開放原始碼軟體。 • 您有多雲端策略，需要非廠商特定的技術。 • 您的團隊能夠解決任何關鍵的生產問題。 • 您想要彈性使用、修改和擴展您想要的產品。 • 您想要在新功能發佈後立即存取新功能。	• 您不想手動管理、監控和維護您的基礎設施。 • 您想要透過跨儲存層分層資料來管理不斷增長的分析成本的簡單方法，並利用 Amazon S3 的耐用性和低成本。 • 您想要利用與其他 的整合， AWS 服務 例如 DynamoDB、Amazon DocumentDB （具有 MongoDB 相容性）、IAM、CloudWatch 和 CloudFormation。 • 您想要在生產問題期間輕鬆存取 的協助， 支援 以進行預防性維護。 • 您想要利用自我修復、主動維護、彈性和備份等功能。

支援的 Elasticsearch 和 OpenSearch 版本

OpenSearch Service 支援下列版本的 OpenSearch：

- 3.5、3.3、3.1、2.19、2.17、2.15、2.13、2.11、2.9、2.7、2.5、2.3、1.3、1.2、1.1 和 1.0

OpenSearch Service 支援下列版本的舊版 Elasticsearch：

- 7.10、7.9、7.8、7.7、7.4、7.1、6.8、6.7、6.5、6.4、6.3、6.2、6.0、5.6、5.5、5.3、5.1、2.3 和 1.5

我們建議您升級到最新的可用 OpenSearch 版本，以充分利用 OpenSearch Service，包括價格效能、功能豐富性和安全性改進。

標準和延伸支援

AWS 在標準支援下， 為 版本提供錯誤修正和安全性更新。對於延伸支援中的版本， AWS 會在標準支援結束後提供至少 12 個月的關鍵安全性修正，每個標準化執行個體小時 (NIH) 按固定費用計費。NIH 是以執行個體大小和使用時數為基礎。

當網域執行不再受標準支援的版本時，會自動收取延長支援費用。若要避免這些費用，請升級至支援的版本。

下表顯示 OpenSearch 和舊版 Elasticsearch 版本的終止支援排程。

OpenSearch Service 支援多個版本的 OpenSearch 和舊版開放原始碼 Elasticsearch 版本。對於某些版本，我們已發佈結束標準支援和延長支援日期。我們建議您升級到最新的可用 OpenSearch 版本，以在價格效能、功能豐富性和安全性改進方面充分利用 OpenSearch Service。下表提供 Elasticsearch 和 OpenSearch 版本及其支援排程的清單。

Elasticsearch 版本的終止支援排程如下：

軟體版本	標準支援結束	延長支援結束
Elasticsearch 1.5 和 2.3 版	2025 年 11 月 7 日	2026 年 11 月 7 日
Elasticsearch 5.1 到 5.5 版	2025 年 11 月 7 日	2026 年 11 月 7 日
Elasticsearch 5.6 版	2025 年 11 月 7 日	2028 年 11 月 7 日

軟體版本	標準支援結束	延長支援結束
Elasticsearch 6.0 到 6.7 版	2025 年 11 月 7 日	2026 年 11 月 7 日
Elasticsearch 6.8 版	未宣布	未宣布
Elasticsearch 7.1 到 7.8 版	2025 年 11 月 7 日	2026 年 11 月 7 日
Elasticsearch 7.9 版	未宣布	未宣布
Elasticsearch 7.10 版	未宣布	未宣布

OpenSearch 版本的終止支援排程如下：

軟體版本	標準支援結束	延長支援結束
OpenSearch 1.0 到 1.2 版	2025 年 11 月 7 日	2026 年 11 月 7 日
OpenSearch 1.3 版	未宣布	未宣布
OpenSearch 2.3 到 2.9 版	2025 年 11 月 7 日	2026 年 11 月 7 日
OpenSearch 2.11 版和更新版本	未宣布	未宣布

OpenSearch 和 Elasticsearch 的標準支援和延伸支援

AWS 為標準支援涵蓋的版本提供定期錯誤修正和安全性更新。對於延伸支援下的版本， AWS 會在標準支援結束後提供至少 12 個月的重大安全性修正，每個標準化執行個體小時 (NIH) 需支付額外的固定費用。NIH 的計算方式為執行個體大小（例如中型、大型）和執行個體時數的因素（如需範例，請參閱計算延伸支援費用一節）。當網域執行的標準支援已結束的版本時，會自動套用延長支援費用。您可以升級至標準支援仍涵蓋的最新版本，以避免延長支援費用。如需延長支援費用的詳細資訊，請參閱定價頁面。如需延伸支援的一般相關資訊，請參閱延伸支援常見問答集。

計算延伸支援費用

在延伸支援下執行版本的網域需支付固定額外費用/標準化執行個體小時 (NIH)，例如美國東部（維吉尼亞北部）區域中的 0.0065 美元。NIH 的計算方式為執行個體大小（例如，中型、大型）和執行個體時數的因素。例如，如果您在美國東部（維吉尼亞北部）區域執行 m7g.medium.search 執行個體 24 小時，其定價為每執行個體小時（隨需）0.068 USD，則您通常會支付 1.632 USD (0.068 x 24 USD)。如果您執行的版本處於擴充支援中，您將額外支付 0.0065 USD/NIH，計算方式為 0.0065 USD x 24（執行個體時數）x 2（大小標準化因素；中型執行個體為 2)，而 24 小時的擴充支援將增加至 0.312 USD。您將為 24 小時支付的總費用將是標準執行個體使用成本和延伸支援成本的總和，即 1.944 美元 (1.632 美元 + 0.312 美元）。下表顯示 OpenSearch Service 中各種執行個體大小的標準化因素。

執行個體大小	正規化因素
超微型	0.25
微型	0.5
小型	1
中型	2
大型	4
xlarge	8
2xlarge	16
4xlarge	32

執行個體大小	正規化因素	
8xlarge	64	
9xlarge	72	
10xlarge	80	
12xlarge	96	
16xlarge	128	
18xlarge	144	
24xlarge	192	
32xlarge	256	

Amazon OpenSearch Service 定價

若是 OpenSearch Service，您需對於每小時 EC2 執行個體的使用，以及連接到您的執行個體的任何 EBS 儲存磁碟區的累積大小支付費用。也需支付標準 AWS 資料傳輸費用。

不過，存在值得注意的某些資料傳輸例外狀況。如果網域使用多個可用區域，則 OpenSearch Service 不會針對可用區域間的流量計費。在碎片分配和重新平衡期間網域內發生大量資料傳輸。OpenSearch Service 對此流量不會計量也不會收費。同樣的，OpenSearch Service 不會針對 UltraWarm/冷節點和 Simple Storage Service (Amazon S3) 之間的資料傳輸收取費用。

如需完整定價資訊，請參閱 Amazon OpenSearch Service 定價。如需組態變更所產生費用的變動的資訊，請參閱the section called “組態變更的費用”。

相關服務

OpenSearch Service 通常搭配使用以下服務：

Amazon CloudWatch

OpenSearch Service 網域會自動傳送指標至 CloudWatch，讓您得以監控網域的運作狀態和效能。如需詳細資訊，請參閱使用 Amazon CloudWatch 監控 OpenSearch 叢集指標。

CloudWatch Logs 也可以前往其他方向。您可以設定 CloudWatch Logs，將資料串流到 OpenSearch Service 以進行分析。如需詳細資訊，請參閱 the section called “從 Amazon CloudWatch 中載入串流資料”。

AWS CloudTrail

使用 AWS CloudTrail 取得您帳戶的 OpenSearch Service 組態 API 呼叫和相關事件的歷史記錄。如需詳細資訊，請參閱使用 監控 Amazon OpenSearch Service API 呼叫 AWS CloudTrail。

Amazon Kinesis

Kinesis 是一項受管服務，可即時處理大規模的串流資料。如需詳細資訊，請參閱 the section called “從 Amazon Kinesis Data Streams 中載入串流資料” 和 the section called “從 Amazon Data Firehose 載入串流資料”。

Amazon S3

Amazon Simple Storage Service (Amazon S3) 為網際網路提供儲存服務。本指南提供與 Simple Storage Service (Amazon S3) 整合的 Lambda 範本程式碼。如需詳細資訊，請參閱the section called “從 Amazon S3 載入串流資料”。

AWS IAM

AWS Identity and Access Management (IAM) 是一種 Web 服務，可用來管理對 OpenSearch Service 網域的存取。如需詳細資訊，請參閱the section called “身分和存取權管理”。

AWS Lambda

AWS Lambda 是一種運算服務，可讓您執行程式碼，而無需佈建或管理伺服器。本指南提供從 DynamoDB、Simple Storage Service (Amazon S3) 和 Kinesis 中串流資料的 Lambda 範本程式碼。如需詳細資訊，請參閱the section called “將資料串流載入至 OpenSearch Service”。

Amazon DynamoDB

Amazon DynamoDB 是一項完全受管的 NoSQL 資料庫服務，可提供快速且可預期的效能及無縫的可擴展性。如需進一步了解有關將資料串流到 OpenSearch Service 的資訊，請參閱the section called “從 Amazon DynamoDB 中載入串流資料”。

Amazon Quick

您可以使用快速儀表板視覺化 OpenSearch Service 中的資料。如需詳細資訊，請參閱《快速使用者指南》中的將 Amazon OpenSearch Service 與 Quick 搭配使用。

> **Note**
>
> OpenSearch 包括來自 Elasticsearch B.V. 的某些 Apache 授權的 Elasticsearch 程式碼和其他原始程式碼。Elasticsearch B.V. 不是該其他原始程式碼的來源。ELASTICSEARCH 是 Elasticsearch B.V. 的註冊商標。

設定 Amazon OpenSearch Service

授予許可

在生產環境中，我們建議您使用更精細的政策。若要進一步了解存取管理，請參閱《IAM 使用者指南》中的AWS 資源的存取管理。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 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。遵循《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。
- 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。
  - (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。

授與程式設計存取權

如果使用者想要與 AWS 外部互動，則需要程式設計存取 AWS 管理主控台。授予程式設計存取權的方式取決於正在存取的使用者類型 AWS。

若要授予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	根據
IAM	（建議）使用主控台登入資料做為臨時登入資料，以簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的登入以進行 AWS 本機開發。

哪個使用者需要程式設計存取權？	到	根據
		• AWS SDKs，請參閱 AWS SDKs 和工具參考指南中的登入以進行 AWS 本機開發。
人力資源身分 (IAM Identity Center 中管理的使用者)	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。 • AWS SDKs、工具和 AWS APIs，請參閱 AWS SDK 和工具參考指南中的 SDKsIAM Identity Center 身分驗證。
IAM	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	遵循《IAM 使用者指南》中將臨時登入資料與 AWS 資源搭配使用的指示。

哪個使用者需要程式設計存取權？	到	根據
IAM	(不建議使用) 使用長期憑證簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的使用 IAM 使用者憑證進行身分驗證。 • AWS SDKs和工具，請參閱 AWS SDKs和工具參考指南中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱《 IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

安裝和設定 AWS CLI

如果您想要使用 OpenSearch Service APIs，您必須安裝最新版本的 AWS Command Line Interface (AWS CLI)。您不需要從主控台 AWS CLI 使用 OpenSearch Service，而且可以按照中的步驟在沒有 CLI 的情況下開始使用Amazon OpenSearch Service 入門。

若要設定 AWS CLI

1. 若要 AWS CLI 為 macOS、Linux 或 Windows 安裝最新版本的 ，請參閱安裝或更新最新版本的 AWS CLI。
2. 若要設定存取的 AWS CLI 和安全設定 AWS 服務，包括 OpenSearch Service，請參閱使用 的快速組態`aws configure`。
3. 若要驗證設定，請在命令提示中輸入下列 DataBrew 命令。

```
aws opensearch help
```

AWS CLI 命令會使用 AWS 區域 組態中的預設值，除非您使用參數或設定檔來設定。若要 AWS 區域 使用 參數設定 ，您可以將 `--region` 參數新增至每個命令。

若要 AWS 區域 使用設定檔設定您的 ，請先在 `~/.aws/config` 檔案或 `%UserProfile%/.aws/config` 檔案中新增具名設定檔 （適用於 Microsoft Windows)。請遵循 的具名設定檔 AWS CLI中的步驟。接著，使用類似下列範例中的命令來設定您的 AWS 區域 和其他設定。

```
[profile opensearch]
aws_access_key_id = ACCESS-KEY-ID-OF-IAM-USER
aws_secret_access_key = SECRET-ACCESS-KEY-ID-OF-IAM-USER
region = us-east-1
output = text
```

開啟 主控台

本節中的大多數主控台導向主題都從 OpenSearch Service 主控台開始。如果您尚未登入 AWS 帳戶，請登入，然後開啟 OpenSearch Service 主控台並繼續前往下一節以繼續開始使用 OpenSearch Service。

Amazon OpenSearch Service 入門

若要開始使用，而您尚未擁有帳戶，請註冊一個 AWS 帳戶。在設定好帳戶之後，請完成 Amazon OpenSearch Service 的入門教學課程。如果您需要更多資訊，同時要了解該服務，請參閱以下簡介主題：

- 建立網域。
- 為您的工作負載適當調整網域大小。
- 使用網域存取政策或精細存取控制來控制對網域的存取。 ???
- 手動或從其他 AWS 服務為資料編製索引。
- 使用 OpenSearch Dashboards 搜尋您的資料並建立視覺化效果。
- 進一步了解建立網域的進階選項。如需詳細資訊，請參閱建立和管理網域。
- 探索如何管理網域中的索引。如需詳細資訊，請參閱管理索引。
- 嘗試使用 Amazon OpenSearch Service 的教程之一。如需詳細資訊，請參閱教學。

如需有關從自我管理的 OpenSearch 叢集遷移至 OpenSearch Service 的資訊，請參閱the section called “遷移至 OpenSearch Service”。

如需更詳細的資訊，請參閱建立和管理網域和本指南內的其他主題。如需有關從自我管理的 OpenSearch 叢集遷移至 OpenSearch Service 的資訊，請參閱the section called “遷移至 OpenSearch Service”。

您可以使用 OpenSearch Service 主控台、 AWS CLI或 AWS SDK 來完成下列步驟。如需安裝和設定的詳細資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》。

建立 Amazon OpenSearch Service 網域

> **Important**
>
> 這是設定測試版 Amazon OpenSearch Service 網域的簡明教程。請勿使用此程序來建立生產網域。如需相同程序的完整版本，請參閱建立和管理網域。

OpenSearch Service 網域與 OpenSearch 叢集同義。網域是指具有您指定之設定、執行個體類型、執行個體計數和儲存資源的叢集。您可以使用 主控台 AWS CLI、 或 AWS SDKs 來建立 OpenSearch Service 網域。

若要使用主控台來建立 OpenSearch Service 網域

1. 前往 https://aws.amazon.com 並選擇 Sign In to the Console (登入主控台)。
2. 在 Analytics (分析) 下，選擇 Amazon OpenSearch Service。
3. 選擇 Create domain (建立網域)。
4. 提供網域名稱。本教學課程中的範例使用名稱 movies。
5. 針對網域建立方法，選擇標準建立。

> **Note**
>
> 若要使用最佳實務快速設定生產網域，您可以選擇輕鬆建立。為了本教學課程的開發和測試目的，我們將使用標準建立。

6. 針對範本，選擇開發/測試。
7. 針對部署選項，選擇具有待命的網域。
8. 對於 Version (版本)，選擇最新版本。
9. 現在，請忽略資料節點、暖和冷資料儲存、專用主節點、快照組態和自訂端點區段。
10. 為了保持本教學課程的簡易性，請使用公有存取網域。在 Network (網路) 中，選擇 Public access (公有存取)。
11. 在精細存取控制設定中，保持選取啟用精細存取控制核取方塊。選取建立主要使用者，並提供使用者名稱和密碼。
12. 現在，請忽略 SAML authentication (SAML 身分驗證) 和 Amazon Cognito authentication (Amazon Cognito 身分驗證) 區段。
13. 對於 Access policy (存取政策)，選擇 Only use fine-grained access control (僅使用精細存取控制)。在本教學課程中，精細存取控制會處理驗證，而非網域存取政策。
14. 忽略其他設定，並選擇 Create (建立)。新網域通常需要 15-30 分鐘才能初始化，但視組態而定，可能需要更長的時間。網域初始化後，選擇網域以打開其組態窗格。記下 General information (一般資訊) 下的網域端點 (例如，`https://search-my-domain.us-east-1.es.amazonaws.com`)，下一步會用到。

下一步：將資料上傳到 OpenSearch Service 網域以編製索引

將資料上傳至 Amazon OpenSearch Service 以編製索引

> **Important**
>
> 這是將少量測試資料上傳到 Amazon OpenSearch Service 的簡明教程。如需有關在生產網域中上傳資料的詳細資訊，請參閱建立資料索引。

您可以使用命令列或大部分程式設計語言將資料上傳至 OpenSearch Service 網域。

為了簡潔和方便起見，下列範例請求使用 curl (常見的 HTTP 用戶端)。像 curl 一樣的用戶端無法執行請求簽署，但如果您的存取政策指定 IAM 使用者或角色，便需要執行請求簽署。若要成功完成此程序，您必須使用精細存取控制搭配主要使用者名稱和密碼，如您在步驟 1 中所設定。

您可以在 Windows 中安裝 curl，並在命令提示中使用它，但我們建議使用像 Cygwin 或 Windows Subsystem for Linux 之類的工具。macOS 和大部分 Linux 發行版本均預先安裝有 curl。

選項 1：上傳單一文件

執行以下命令，將單一文件加入到 movies 網域：

```
curl -XPUT -u 'master-user:master-user-password' 'domain-endpoint/movies/_doc/1' -d
 '{"director": "Burton, Tim", "genre": ["Comedy","Sci-Fi"], "year": 1996, "actor":
 ["Jack Nicholson","Pierce Brosnan","Sarah Jessica Parker"], "title": "Mars Attacks!"}'
 -H 'Content-Type: application/json'
```

在 命令中，提供您在步驟 1 中建立的使用者名稱和密碼。

如需有關此命令以及如何向 OpenSearch Service 提出簽署請求的詳細說明，請參閱建立資料索引。

選項 2：上傳多個文件

若要將包含多個文件的 JSON 檔案上傳到 OpenSearch Service 網域

1. 建立稱為 `bulk_movies.json` 的本機檔案。將以下內容貼到檔案中，並新增結尾新行：

```
{ "index" : { "_index": "movies", "_id" : "2" } }
{"director": "Frankenheimer, John", "genre": ["Drama", "Mystery", "Thriller",
 "Crime"], "year": 1962, "actor": ["Lansbury, Angela", "Sinatra, Frank", "Leigh,
 Janet", "Harvey, Laurence", "Silva, Henry", "Frees, Paul", "Gregory, James",
```

```
 "Bissell, Whit", "McGiver, John", "Parrish, Leslie", "Edwards, James", "Flowers,
 Bess", "Dhiegh, Khigh", "Payne, Julie", "Kleeb, Helen", "Gray, Joe", "Nalder,
 Reggie", "Stevens, Bert", "Masters, Michael", "Lowell, Tom"], "title": "The
 Manchurian Candidate"}
{ "index" : { "_index": "movies", "_id" : "3" } }
{"director": "Baird, Stuart", "genre": ["Action", "Crime", "Thriller"], "year":
 1998, "actor": ["Downey Jr., Robert", "Jones, Tommy Lee", "Snipes, Wesley",
 "Pantoliano, Joe", "Jacob, Ir\u00e8ne", "Nelligan, Kate", "Roebuck, Daniel",
 "Malahide, Patrick", "Richardson, LaTanya", "Wood, Tom", "Kosik, Thomas",
 "Stellate, Nick", "Minkoff, Robert", "Brown, Spitfire", "Foster, Reese",
 "Spielbauer, Bruce", "Mukherji, Kevin", "Cray, Ed", "Fordham, David", "Jett,
 Charlie"], "title": "U.S. Marshals"}
{ "index" : { "_index": "movies", "_id" : "4" } }
{"director": "Ray, Nicholas", "genre": ["Drama", "Romance"], "year": 1955, "actor":
 ["Hopper, Dennis", "Wood, Natalie", "Dean, James", "Mineo, Sal", "Backus, Jim",
 "Platt, Edward", "Ray, Nicholas", "Hopper, William", "Allen, Corey", "Birch,
 Paul", "Hudson, Rochelle", "Doran, Ann", "Hicks, Chuck", "Leigh, Nelson",
 "Williams, Robert", "Wessel, Dick", "Bryar, Paul", "Sessions, Almira", "McMahon,
 David", "Peters Jr., House"], "title": "Rebel Without a Cause"}
```

2. 在儲存檔案的本機目錄中執行下列命令，以將其上傳到 movies 網域：

```
curl -XPOST -u 'master-user:master-user-password' 'domain-endpoint/movies/_bulk' --
data-binary @bulk_movies.json -H 'Content-Type: application/x-ndjson'
```

如需有關大量檔案格式的詳細資訊，請參閱建立資料索引。

下一步：搜尋文件

在 Amazon OpenSearch Service 中搜尋文件

若要在 Amazon OpenSearch Service 網域中搜尋文件，請使用 OpenSearch 搜尋 API。或者，您可以使用 OpenSearch Dashboards 搜尋網域中的文件。

從命令列搜尋文件

執行以下命令搜尋影片網域的 mars 這個字：

```
curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/movies/_search?
q=mars&pretty=true'
```

如果已在先前頁面中使用大量資料，則請嘗試改成搜尋 rebel。

您應該會看到類似以下的回應：


```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.2876821,
    "hits" : [
      {
        "_index" : "movies",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.2876821,
        "_source" : {
          "director" : "Burton, Tim",
          "genre" : [
            "Comedy",
            "Sci-Fi"
          ],
          "year" : 1996,
          "actor" : [
            "Jack Nicholson",
            "Pierce Brosnan",
            "Sarah Jessica Parker"
          ],
          "title" : "Mars Attacks!"
        }
      }
    ]
  }
}
```

使用 OpenSearch Dashboards 搜尋文件

OpenSearch Dashboards 是一種常見的開源視覺化工具，專為與 OpenSearch 一起使用而設計。它提供了一個很有幫助的使用者介面，供您搜尋和監控您的索引。

若要使用 Dashboards 從 OpenSearch Service 網域中搜尋文件

1. 導覽至您網域的 OpenSearch Dashboards URL。您可以在 OpenSearch Service 主控台網域的儀表板上找到 URL。URL 遵循以下格式：

   ```
   domain-endpoint/_dashboards/
   ```

2. 使用您的主要使用者名稱和密碼登入。
3. 若要使用 Dashboards，您需要建立至少一個索引模式。Dashboards 使用這些模式來識別您要分析的索引。開啟左側導覽面板，選擇 Stack Management (堆疊管理)，選擇 Index Patterns (索引模式)，然後選擇 Create index pattern (建立索引模式)。對於本教學課程，輸入 movies。
4. 選擇 Next step (下一步)，然後選擇 Create index pattern (建立索引模式)。建立模式之後，您可以檢視各種文件欄位，例如 `actor` 和 `director`。
5. 返回 Index Patterns (索引模式) 頁面，並確認將 `movies` 設為預設值。如果不是，請選擇模式並選擇星形圖示以使其成為預設值。
6. 若要開始搜尋您的資料，請再次開啟左側導覽面板，然後選擇 Discover (探索)。
7. 如果上傳了單一文件，則在搜尋列中輸入 mars，或者如果上傳了多份文件，則輸入 rebel，然後按 Enter 鍵。您可以嘗試搜尋其他用語，例如演員或導演姓名。

下一步：刪除網域

刪除 Amazon OpenSearch Service 網域

由於本教學課程的 movies 網域是用於測試目的，因此請確保在完成試驗後將其刪除，以避免產生任何費用。

若要從主控台中刪除 OpenSearch Service 網域

1. 登入 Amazon OpenSearch Service 主控台。
2. 在 Domains (網域) 中，選取 movies 網域。
3. 選擇 Delete (刪除)，並確認刪除。

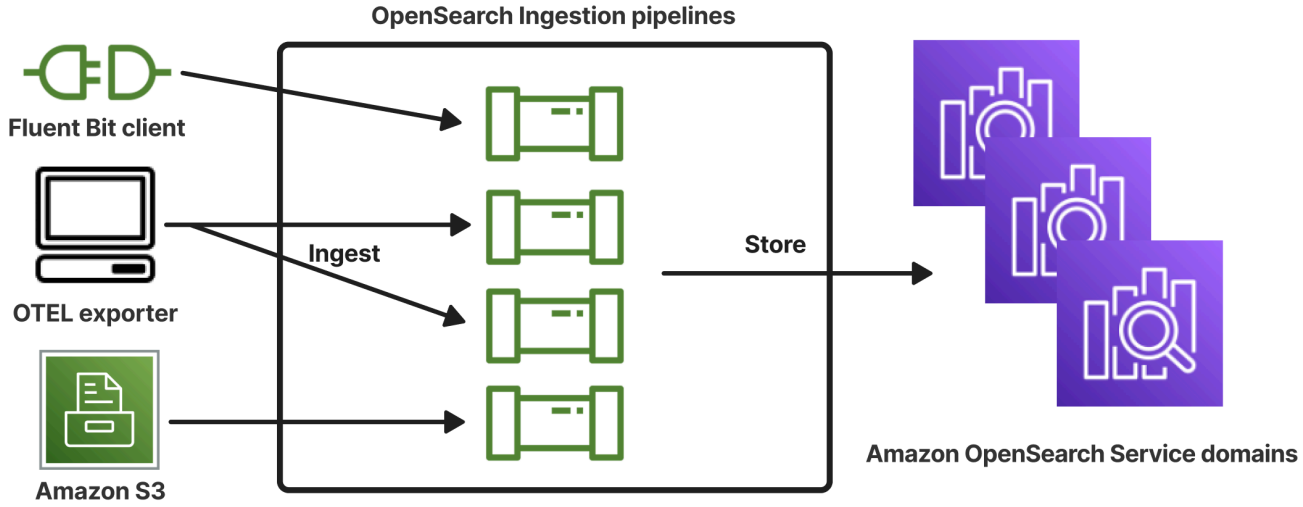
Amazon OpenSearch 擷取概觀

Amazon OpenSearch Ingestion 是全受管的無伺服器資料收集器，可將即時日誌、指標和追蹤資料串流至 Amazon OpenSearch Service 網域和 OpenSearch Serverless 集合。

使用 OpenSearch Ingestion，您不再需要 Logstash 或 Jaeger 等第三方工具來擷取資料。您可以將資料生產者設定為將資料傳送至 OpenSearch Ingestion，並自動將其交付至您指定的網域或集合。您也可以在交付之前轉換資料。

由於 OpenSearch Ingestion 是無伺服器，因此您不需要手動管理基礎設施、修補程式軟體或擴展叢集。您可以直接在 中佈建擷取管道 AWS 管理主控台，而 OpenSearch Ingestion 會處理其餘項目。

OpenSearch Ingestion 是 Amazon OpenSearch Service 的元件，採用 Data Prepper 技術，這是一種開放原始碼資料收集器，可篩選、擴充、轉換、標準化和彙總資料，以進行下游分析和視覺化。



Amazon OpenSearch Ingestion 中的重要概念

在您開始使用 OpenSearch Ingestion 之前，了解這些關鍵概念會很有幫助。

管道

從 OpenSearch Ingestion 的角度來看，管道是指您在 OpenSearch Service 中建立的單一佈建資料收集器。您可以將其視為整個 YAML 組態檔案，其中包含一或多個子管道。如需建立擷取管道的步驟，請參閱the section called “建立管道”。

子管道

您可以在 YAML 組態檔案中定義子管道。每個子管道是來源、緩衝區、零或多個處理器和一或多個接收器的組合。您可以在單一 YAML 檔案中定義多個子管道，每個都具有唯一的來源、處理器和接收器。為了協助監控 CloudWatch 和其他 服務，建議您指定與其所有子管道不同的管道名稱。

您可以在單一 YAML 檔案中將多個子管道串連在一起，以便一個子管道的來源是另一個子管道，而其接收器是第三個子管道。如需範例，請參閱 the section called “OpenTelemetry Collector”。

來源

子管道的輸入元件。它定義管道使用記錄的機制。來源可以透過 HTTPS 接收事件，或從 Amazon S3 等外部端點讀取來取用事件。來源有兩種類型：推送型和提取型。以推送為基礎的來源，例如 HTTP 和 OTel 日誌，將記錄串流到擷取端點。提取型來源，例如 OTel 追蹤和 S3，從來源提取資料。

處理器

中繼處理單元，可在將記錄發佈到接收器之前，先篩選、轉換記錄，並將記錄擴充為所需的格式。處理器是管道的選用元件。如果您未定義處理器，記錄會以來源中定義的格式發佈。您可以有多個處理器。管道會依您定義的順序執行處理器。

接收

子管道的輸出元件。它定義了子管道會將記錄發佈至其中的一或多個目的地。OpenSearch Ingestion 支援 OpenSearch Service 網域做為接收器。它也支援子管道做為接收器。這表示您可以在單一 OpenSearch Ingestion 管道 (YAML 檔案） 中將多個子管道串連在一起。不支援自我管理 OpenSearch 叢集做為接收器。

緩衝區

處理器的一部分，可做為來源和接收器之間的圖層。您無法在管道中手動設定緩衝區。OpenSearch Ingestion 使用預設緩衝區組態。

路由

處理器的一部分，允許管道作者僅將符合特定條件的事件傳送至不同的接收器。

有效的子管道定義必須包含來源和接收器。如需這些管道元素的詳細資訊，請參閱組態參考。

Amazon OpenSearch 擷取的優點

OpenSearch Ingestion 具有下列主要優點：

- 您不需要手動管理自行佈建的管道。
- 根據您定義的容量限制自動擴展管道。
- 使用安全性和錯誤修補程式讓您的管道保持最新狀態。
- 提供將管道連接到虛擬私有雲端 (VPC) 的選項，以增加一層安全性。
- 可讓您停止和啟動管道，以控制成本。
- 提供熱門使用案例的管道組態藍圖，協助您更快地啟動和執行。
- 可讓您透過各種 AWS SDKs 和 OpenSearch Ingestion API，以程式設計方式與管道互動。
- 支援 Amazon CloudWatch 中的效能監控和 CloudWatch Logs 中的錯誤記錄。

Amazon OpenSearch 擷取的限制

OpenSearch Ingestion 有下列限制：

- 您只能將資料擷取至執行 OpenSearch 1.0 或更新版本的網域，或 Elasticsearch 6.8 或更新版本。如果您使用的是 OTel 追蹤來源，我們建議您使用 Elasticsearch 7.9 或更新版本，以便您可以使用 OpenSearch Dashboards 外掛程式。
- 如果管道正在寫入 VPC 內的 OpenSearch Service 網域，則必須在 AWS 區域 與網域相同的 中建立管道。
- 您只能在管道定義中設定單一資料來源。
- 您無法將自我管理 OpenSearch 叢集指定為接收器。
- 您無法將自訂端點指定為接收器。您仍然可以寫入已啟用自訂端點的網域，但必須指定其標準端點。
- 您無法將選擇加入區域中的資源指定為來源或目的地。
- 您可以在管道組態中包含的參數有一些限制。如需詳細資訊，請參閱the section called “組態需求和限制條件”。

支援的資料準備版本

OpenSearch Ingestion 目前支援下列主要版本的 Data Prepper：

- 2.x

當您使用程式碼編輯器建立管道時，請使用必要的`version`選項來指定要使用的主要 Data Prepper 版本。例如 `version: "2"`。OpenSearch Ingestion 會擷取該主要版本的最新支援的次要版本，並使用該版本佈建管道。

如果您不使用程式碼編輯器來建立管道，OpenSearch Ingestion 會自動為您的管道佈建最新的支援版本。

目前，OpenSearch Ingestion 會佈建具有 Data Prepper 2.7 版的管道。如需詳細資訊，請參閱 2.7 版本備註。OpenSearch Ingestion 不支援特定主要版本的每個次要版本。

當您更新管道的組態時，如果支援新的 Data Prepper 次要版本，OpenSearch Ingestion 會自動將管道升級至管道組態中指定之主要版本的最新支援次要版本。例如，您可能在管道組態`version: "2"`中具有 ，而 OpenSearch Ingestion 最初使用 2.6.0 版佈建管道。當新增對 2.7.0 版的支援，且您對管道組態進行變更時，OpenSearch Ingestion 會將管道升級至 2.7.0 版。此程序可讓您的管道隨時掌握最新的錯誤修正和效能改善。除非您在管道組態中手動變更 `version`選項，否則 OpenSearch Ingestion 無法更新管道的主要版本。如需詳細資訊，請參閱the section called “更新管道”。

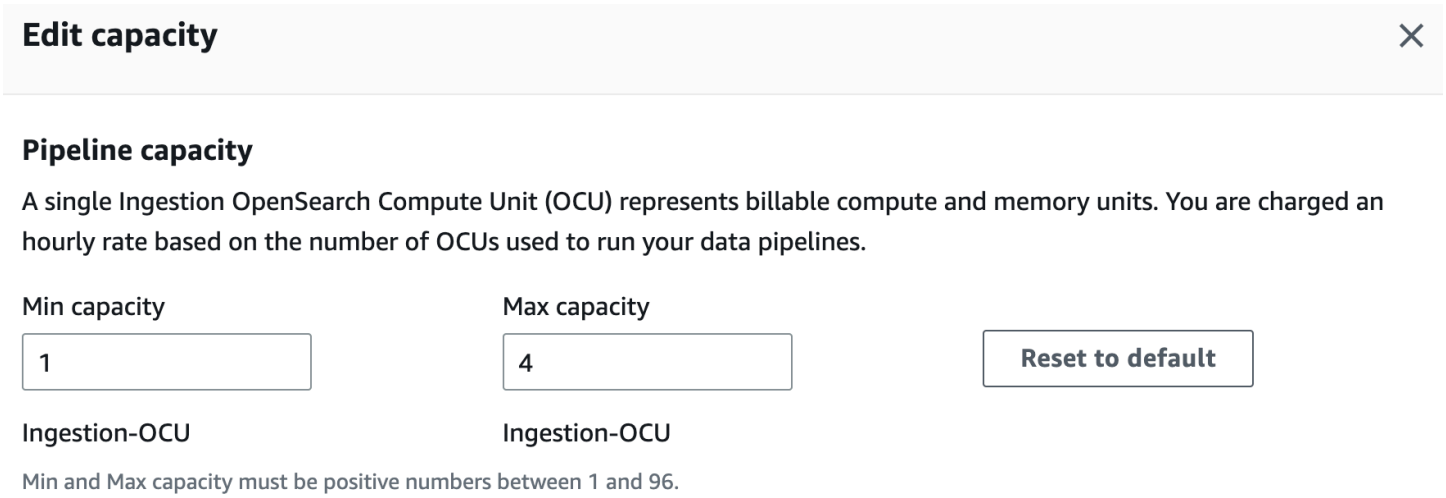
在 Amazon OpenSearch Ingestion 中擴展管道

OpenSearch Ingestion 會根據您指定的最小和最大擷取 OpenSearch 運算單位 （擷取 OCUs) 自動擴展管道容量。這樣就不需要手動佈建和管理。

每個擷取 OCU 是大約 15 GiB 記憶體和 2 個 vCPUs的組合。您可以指定管道的最小和最大 OCU 值，OpenSearch Ingestion 會根據這些限制自動擴展管道容量。

您可以在建立管道時指定下列值：

- 最小容量 – 管道可以將容量減少到此數量的擷取 OCUs。指定的最小容量也是管道的起始容量。
- 最大容量 – 管道可以將容量增加到此數量的擷取 OCUs。



請確定管道的最大容量足以處理工作負載激增，且最小容量足夠低，以便在管道不忙碌時將成本降至最低。根據您的設定，OpenSearch Ingestion 會自動擴展管道的擷取 OCUs 數量，以處理擷取工作負載。在任何特定時間，您只需為管道正在使用的擷取 OCUs 付費。

分配給 OpenSearch Ingestion 管道的容量會根據管道的處理需求以及用戶端應用程式產生的負載進行擴展和縮減。當容量受到限制時，OpenSearch Ingestion 會透過配置更多運算單位 (GiB 的記憶體) 來擴展。當您的管道正在處理較小的工作負載，或完全不處理資料時，可以縮減至設定的擷取 OCUs 下限。

您可以指定至少 1 個擷取 OCU、無狀態管道最多 96 個擷取 OCUs，以及有狀態管道最多 48 個擷取 OCUs。對於推送型來源，我們建議至少 2 個擷取 OCUs。啟用持久性緩衝時，您可以指定最少 2 個，最多 384 個擷取 OCUs。

鑑於具有單一來源、簡單 grok 模式和接收器的標準日誌管道，每個運算單位每秒最多可支援 2 MiB。對於具有多個處理器的更複雜日誌管道，每個運算單位可能支援較少的擷取負載。根據管道容量和資源使用率，OpenSearch Ingestion 擴展程序會開始。

為了確保高可用性，擷取 OCUs 會分散到可用區域 (AZs)。AZs 的數量取決於您指定的最小容量。

例如，如果您指定至少 2 個運算單位，則在任何指定時間使用的擷取 OCUs會平均分散到 2 AZs。如果您指定至少 3 個或更多運算單位，則擷取 OCUs會平均分散到 3 AZs。我們建議您佈建至少兩個擷取 OCUs，以確保擷取管道有 99.9% 的可用性。

當管道處於 `Create failed`、`Deleting`、 和 `Stopped` 狀態時`Creating`，您不需要支付擷取 OCUs 的費用。

如需設定和擷取管道容量設定的指示，請參閱 the section called “建立管道”。

OpenSearch 擷取定價

在任何特定時間，您只需支付配置給管道的擷取 OCUs 數量，無論是否有資料流經管道。OpenSearch Ingestion 會根據用量向上或向下擴展管道容量，立即滿足您的工作負載。

如需完整定價資訊，請參閱 Amazon OpenSearch Service 定價。

支援的 AWS 區域

OpenSearch 擷取可在 AWS 區域 OpenSearch Service 提供的子集中使用。如需支援區域的清單，請參閱《》中的 Amazon OpenSearch Service 端點和配額AWS 一般參考。

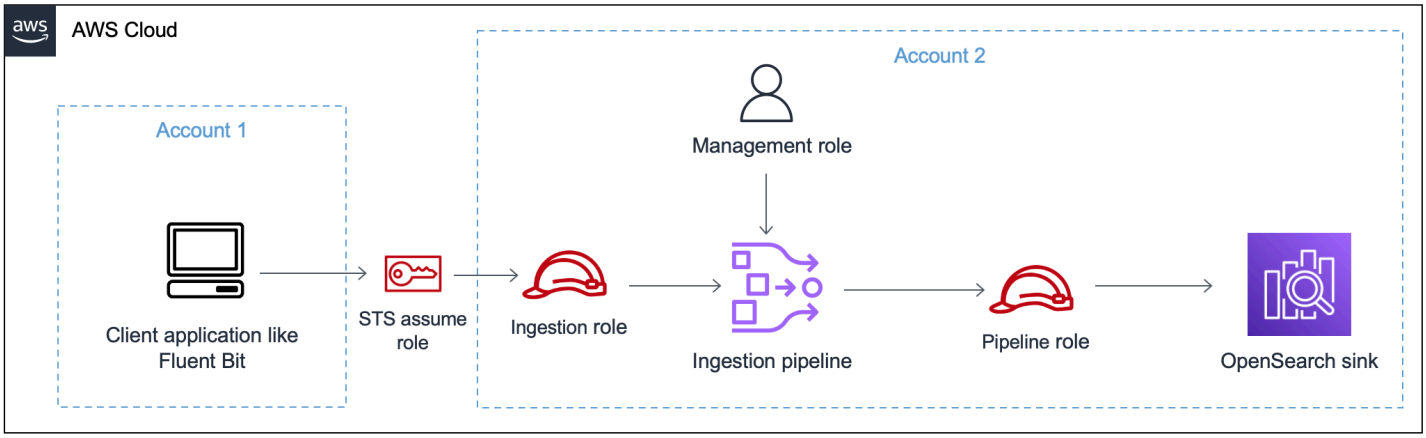
在 Amazon OpenSearch 擷取中設定角色和使用者

Amazon OpenSearch Ingestion 使用各種許可模型和 IAM 角色，以允許來源應用程式寫入管道，並允許管道寫入接收器。您必須先根據您的使用案例建立具有特定許可的一或多個 IAM 角色，才能開始擷取資料。

至少需要下列角色才能設定成功的管道。

名稱	描述
管道角色	管道角色提供管道從來源讀取和寫入網域或集合目的地所需的許可。您可以手動建立管道角色，也可以讓 OpenSearch Ingestion 為您建立管道角色。
擷取角色	擷取角色包含管道資源的 `osis:Ingest` 許可。此許可允許以推送為基礎的來源將資料擷取至管道。

下圖示範典型管道設定，例如 Amazon S3 或 Fluent Bit 的資料來源正在寫入不同帳戶中的管道。在此情況下，用戶端需要擔任擷取角色才能存取管道。如需詳細資訊，請參閱the section called “跨帳戶擷取”。



如需簡單的設定指南，請參閱 the section called “教學課程：將資料擷取至網域”。

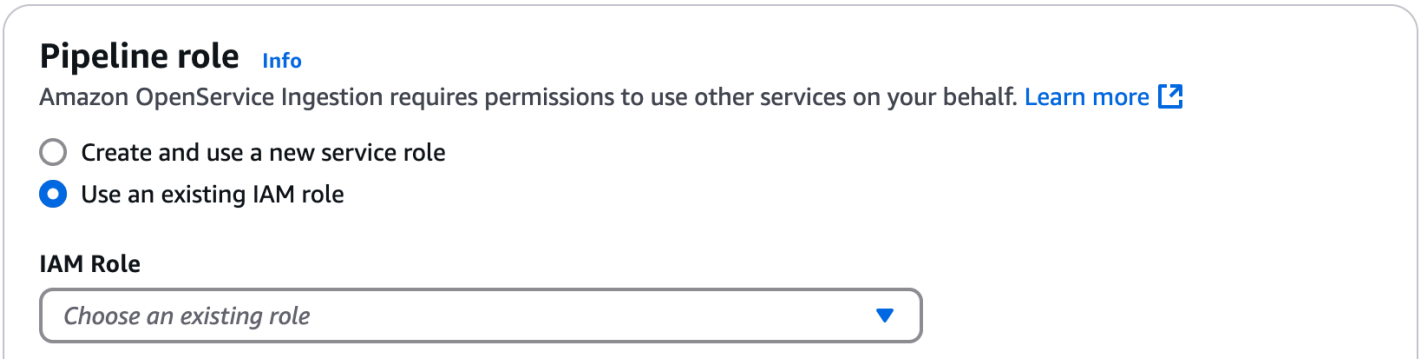
主題

- the section called “管道角色”
- the section called “擷取角色”
- the section called “跨帳戶擷取”

管道角色

管道需要特定許可，才能從其來源讀取和寫入其接收器。這些許可取決於用戶端應用程式或寫入管道 AWS 服務 的 ，以及目的地是 OpenSearch Service 網域、OpenSearch Serverless 集合或 Amazon S3。此外，管道可能需要實際從來源應用程式提取資料的許可 （ 如果來源是提取型外掛程式 ），以及寫入 S3 無效字母佇列的許可，如果啟用的話。

當您建立管道時，您可以選擇指定您手動建立的現有 IAM 角色，或讓 OpenSearch Ingestion 根據您選擇的來源和接收器自動建立管道角色。下圖顯示如何在 中指定管道角色 AWS 管理主控台。



主題

- 自動化管道角色建立
- 手動建立管道角色

自動化管道角色建立

您可以選擇讓 OpenSearch Ingestion 為您建立管道角色。它會根據設定的來源和目的地，自動識別角色所需的許可。它使用您輸入的字首 和 OpenSearchIngestion-建立 IAM 角色。例如，如果您輸入 PipelineRole做為尾碼，OpenSearch Ingestion 會建立名為 的角色OpenSearchIngestion-PipelineRole。

自動建立管道角色可簡化設定程序，並降低組態錯誤的可能性。透過自動化角色建立，您可以避免手動指派許可，確保套用正確的政策，而不會造成安全設定錯誤的風險。這也可透過強制執行最佳實務來節省時間並增強安全合規性，同時確保多個管道部署之間的一致性。

您只能讓 OpenSearch Ingestion 在 中自動建立管道角色 AWS 管理主控台。如果您使用的是 AWS CLI、OpenSearch Ingestion API 或其中一個 SDKs，則必須指定手動建立的管道角色。

若要讓 OpenSearch Ingestion 為您建立角色，請選取建立並使用新的服務角色。

> **Important**
>
> 您仍然需要手動修改網域或集合存取政策，以授予管道角色的存取權。對於使用精細存取控制的網域，您還必須將管道角色映射到後端角色。您可以在建立管道之前或之後執行這些步驟。如需說明，請參閱下列主題：
>
> - 設定網域的資料存取
> - 設定集合的資料和網路存取

手動建立管道角色

如果您需要更多控制許可以符合特定安全或合規要求，建議您手動建立管道角色。手動建立可讓您量身打造角色以符合現有的基礎設施或存取管理策略。您也可以選擇手動設定，將角色與其他 整合， AWS 服務 或確保其符合您的獨特操作需求。

若要選擇手動建立的管道角色，請選取使用現有的 IAM 角色，然後選擇現有的角色。角色必須擁有從所選來源接收資料並寫入所選接收器所需的所有許可。下列各節概述如何手動建立管道角色。

主題

- 從來源讀取的許可
- 寫入網域目的地的許可
- 寫入集合目的地的許可
- 寫入 Amazon S3 或無效字母佇列的許可

從來源讀取的許可

OpenSearch 擷取管道需要從指定來源讀取和接收資料的許可。例如，對於 Amazon DynamoDB 來源，它需要許可，例如 `dynamodb:DescribeTable`和 `dynamodb:DescribeStream`。如需常見來源的範例管道角色存取政策，例如 Amazon S3、Fluent Bit 和 OpenTelemetry Collector，請參閱 the section called “整合管道”。

寫入網域目的地的許可

OpenSearch Ingestion 管道需要許可，才能寫入設定為其目的地的 OpenSearch Service 網域。這些許可包括描述網域並向其傳送 HTTP 請求的能力。公有和 VPC 網域的這些許可相同。如需建立管道角色並在網域存取政策中指定該角色的說明，請參閱允許管道存取網域。

寫入集合目的地的許可

OpenSearch Ingestion 管道需要許可，才能寫入設定為其接收器的 OpenSearch Serverless 集合。這些許可包括描述集合並向其傳送 HTTP 請求的能力。

首先，請確定您的管道角色存取政策授予必要的許可。然後，在資料存取政策中包含此角色，並提供其在集合中建立索引、更新索引、描述索引和撰寫文件的許可。如需完成每個步驟的指示，請參閱允許管道存取集合。

寫入 Amazon S3 或無效字母佇列的許可

如果您將 Amazon S3 指定為管道的目的地，或啟用無效字母佇列 (DLQ)，則管道角色必須允許它存取您指定為目的地的 S3 儲存貯體。

將單獨的許可政策連接到提供 DLQ 存取的管道角色。至少必須授予角色對儲存貯體資源`S3:PutObject`的動作：

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "WriteToS3DLQ",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-dlq-bucket/*"
    }
  ]
}
```

擷取角色

擷取角色是一種 IAM 角色，可讓外部服務安全地與 OpenSearch Ingestion 管道互動和傳送資料。對於推送型來源，例如 Amazon Security Lake，此角色必須授予將資料推送到管道的許可，包括 `osis:Ingest`。對於提取型來源，例如 Amazon S3，該角色必須啟用 OpenSearch Ingestion 來取得它，並使用必要的許可存取資料。

主題

- 以推送為基礎的來源的擷取角色
- 提取型來源的擷取角色
- 跨帳戶擷取

以推送為基礎的來源的擷取角色

對於以推送為基礎的來源，資料會從其他服務傳送或推送至擷取管道，例如 Amazon Security Lake 或 Amazon DynamoDB。在此案例中，擷取角色至少需要與管道互動的`osis:Ingest`許可。

下列 IAM 存取政策示範如何將此許可授予擷取角色：

JSON

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

```
        "osis:Ingest"
      ],
      "Resource": "arn:aws:osis:us-east-1:111122223333:pipeline/pipeline-name/*"
    }
  ]
}
```

提取型來源的擷取角色

對於提取型來源，OpenSearch Ingestion 管道會主動從外部來源提取或擷取資料，例如 Amazon S3。在此情況下，管道必須擔任 IAM 管道角色，授予存取資料來源的必要許可。在這些案例中，擷取角色與管道角色同義。

角色必須包含允許 OpenSearch Ingestion 擔任該角色的信任關係，以及資料來源特定的許可。如需詳細資訊，請參閱the section called “從來源讀取的許可”。

跨帳戶擷取

您可能需要從不同的管道擷取資料 AWS 帳戶，例如應用程式帳戶。若要設定跨帳戶擷取，請在與管道相同的帳戶中定義擷取角色，並在擷取角色與應用程式帳戶之間建立信任關係：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::444455556666:root"
      },
     "Action": "sts:AssumeRole"
  }]
}
```

然後，將應用程式設定為擔任擷取角色。應用程式帳戶必須授予管道帳戶中擷取角色的應用程式角色 AssumeRole 許可。

如需詳細步驟和範例 IAM 政策，請參閱 the section called “提供跨帳戶擷取存取權”。

授予 Amazon OpenSearch Ingestion 管道對網域的存取權

Amazon OpenSearch Ingestion 管道需要許可，才能寫入設定為其目的地的 OpenSearch Service 網域。若要提供存取權，您可以使用限制性許可政策來設定 AWS Identity and Access Management (IAM) 角色，以限制對管道傳送資料的網域的存取。例如，您可能想要將擷取管道限制為僅支援其使用案例所需的網域和索引。

> **Important**
>
> 您可以選擇手動建立管道角色，也可以讓 OpenSearch Ingestion 在管道建立期間為您建立。如果您選擇自動建立角色，OpenSearch Ingestion 會根據您選擇的來源和目的地，將所有必要的許可新增至管道角色存取政策。它會在 IAM 中以您輸入的字首`OpenSearchIngestion-`和尾碼建立管道角色。如需詳細資訊，請參閱the section called “管道角色”。
>
> 如果您讓 OpenSearch Ingestion 為您建立管道角色，則在建立管道之前或之後，您仍需要在網域存取政策中包含該角色，並將其對應至後端角色（如果網域使用精細分級的存取控制）。如需說明，請參閱步驟 2。

主題

- 步驟 1：建立管道角色
- 步驟 2：設定網域的資料存取

步驟 1：建立管道角色

管道角色必須具有連接的許可政策，允許其將資料傳送至網域目的地。它還必須具有信任關係，允許 OpenSearch Ingestion 擔任該角色。如需如何將政策連接至角色的指示，請參閱《IAM 使用者指南》中的新增 IAM 身分許可。

下列範例政策示範您可以在管道角色中提供的最低權限，供它寫入單一網域：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "es:DescribeDomain",
```

```
            "Resource": "arn:aws:es:*:111122223333:domain/*"
        },
        {
            "Effect": "Allow",
            "Action": "es:ESHttp*",
            "Resource": "arn:aws:es:*:111122223333:domain/domain-name/*"
        }
    ]
}
```

如果您打算重複使用角色來寫入多個網域，您可以使用萬用字元 () 取代網域名稱，讓政策更為廣泛*。

角色必須具有下列信任關係，允許 OpenSearch Ingestion 擔任管道角色：

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"osis-pipelines.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

步驟 2：設定網域的資料存取

為了讓管道將資料寫入網域，網域必須具有允許管道角色存取它的網域層級存取政策。

下列範例網域存取政策允許名為 的管道角色`pipeline-role`將資料寫入名為 的網域`ingestion-domain`：

JSON

```
{
```

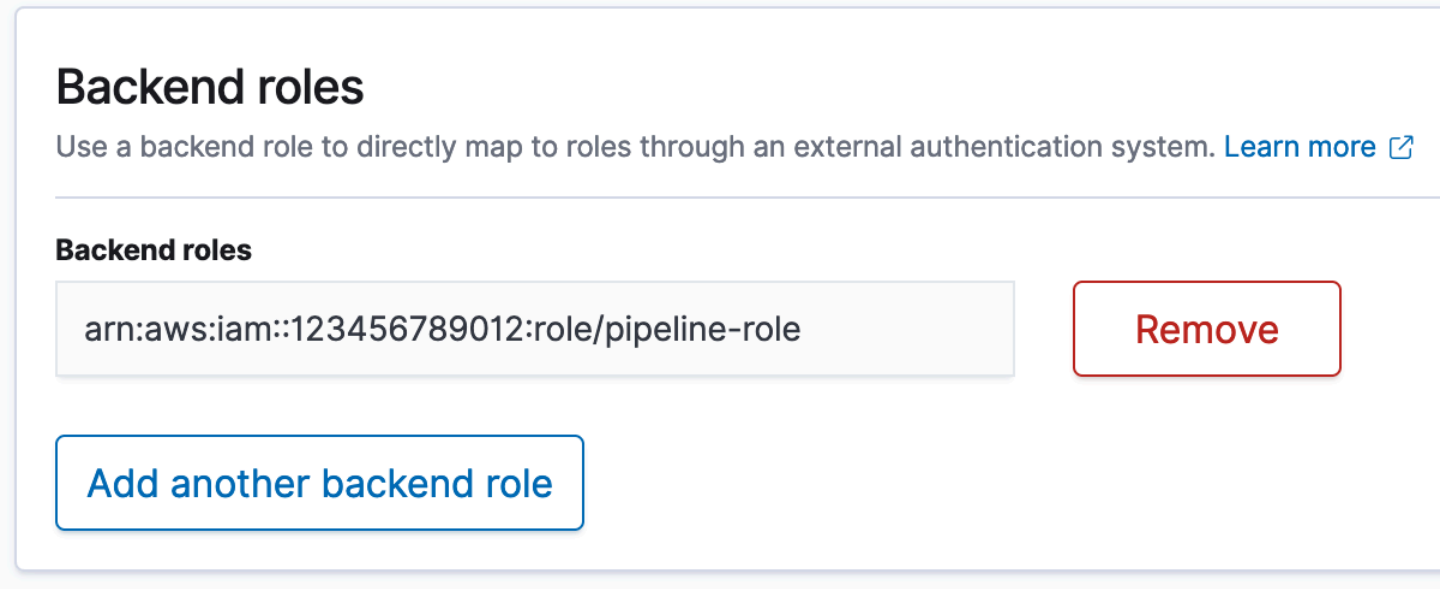
```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/pipeline-role"
            },
            "Action": [
                "es:DescribeDomain",
                "es:ESHttp*"
            ],
            "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
        }
    ]
}
```

映射管道角色（僅適用於使用精細存取控制的網域）

如果您的網域使用精細存取控制進行身分驗證，您需要採取額外的步驟來提供對網域的管道存取權。步驟會根據您的網域組態而有所不同：

- 案例 1：不同的主角色和管道角色 – 如果您使用 IAM Amazon Resource Name (ARN) 做為主使用者，且它與管道角色不同，則需要將管道角色映射至 OpenSearch `all_access` 後端角色。這會將管道角色新增為額外的主要使用者。如需詳細資訊，請參閱其他主要使用者。
- 案例 2：內部使用者資料庫中的主要使用者 – 如果您的網域使用內部使用者資料庫中的主要使用者，以及 OpenSearch Dashboards 的 HTTP 基本身分驗證，則無法直接將主要使用者名稱和密碼傳遞至管道組態。反之，請將管道角色映射至 OpenSearch `all_access` 後端角色。這會將管道角色新增為額外的主要使用者。如需詳細資訊，請參閱其他主要使用者。
- 案例 3：相同的主角色和管道角色（不常見）– 如果您使用 IAM ARN 做為主使用者，而且它與您用作管道角色的 ARN 相同，則不需要採取任何進一步的動作。管道具有寫入網域所需的許可。此案例並不常見，因為大多數環境使用管理員角色或其他角色做為主角色。

下圖顯示如何將管道角色映射至後端角色：

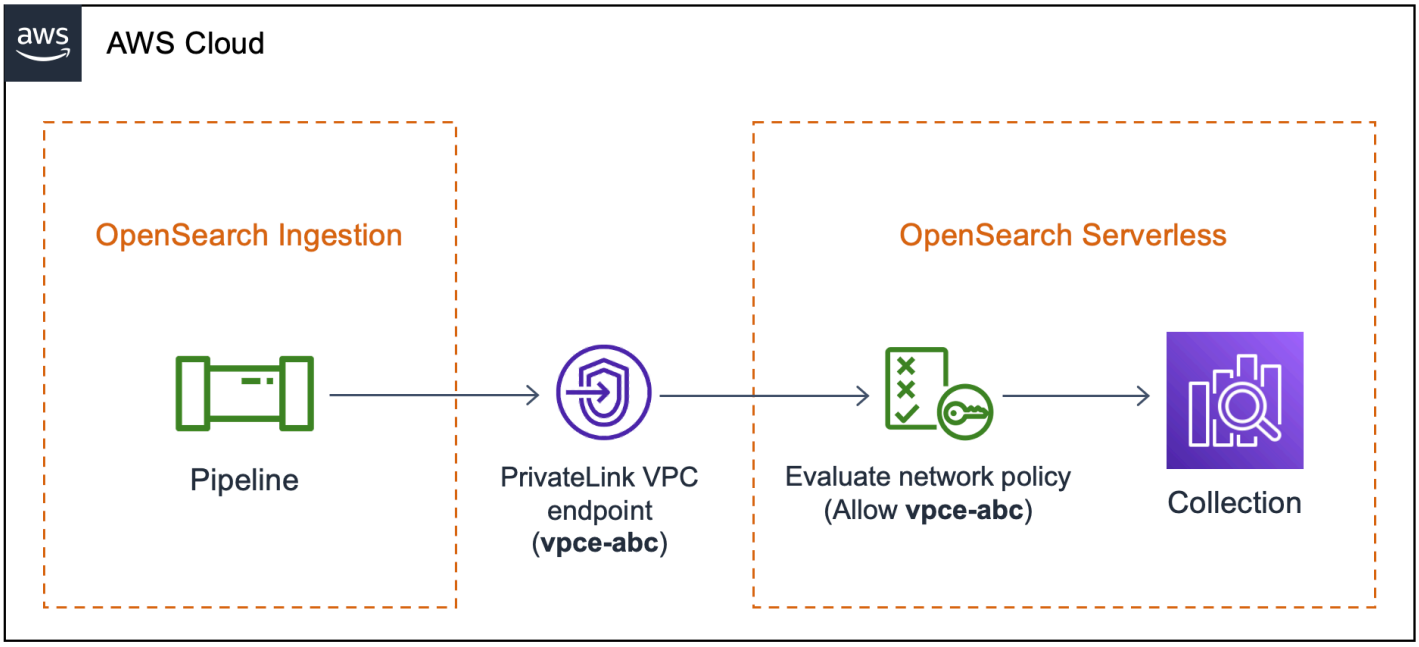


授予 Amazon OpenSearch Ingestion 管道對集合的存取權

Amazon OpenSearch Ingestion 管道可以寫入 OpenSearch Serverless 公有集合或 VPC 集合。若要提供集合的存取權，您可以使用授予集合存取權的許可政策來設定 AWS Identity and Access Management (IAM) 管道角色。管道會擔任此角色，以簽署 OpenSearch Serverless 集合目的地的請求。

> **Important**
>
> 您可以選擇手動建立管道角色，也可以讓 OpenSearch Ingestion 在管道建立期間為您建立。如果您選擇自動建立角色，OpenSearch Ingestion 會根據您選擇的來源和目的地，將所有必要的許可新增至管道角色存取政策。它會在 IAM 中以您輸入的字首`OpenSearchIngestion-`和尾碼建立管道角色。如需詳細資訊，請參閱the section called “管道角色”。
>
> 如果您讓 OpenSearch Ingestion 為您建立管道角色，則在建立管道之前或之後，您仍需要在集合的資料存取政策中包含該角色。如需說明，請參閱步驟 2。

在管道建立期間，OpenSearch Ingestion 會在管道與 OpenSearch Serverless 集合之間建立 AWS PrivateLink 連線。來自管道的所有流量都會經過此 VPC 端點，並路由至集合。若要存取集合，必須透過網路存取政策授予端點對集合的存取權。



主題

- 步驟 1：建立管道角色
- 步驟 2：設定集合的資料和網路存取

步驟 1：建立管道角色

管道角色必須具有連接的許可政策，允許其將資料傳送至集合目的地。它還必須具有信任關係，允許 OpenSearch Ingestion 擔任該角色。如需如何將政策連接至角色的指示，請參閱《IAM 使用者指南》中的新增 IAM 身分許可。

下列範例政策示範您可以在管道角色存取政策中提供的最小權限，供它寫入集合：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll",
                "aoss:BatchGetCollection",
```

```
                "aoss:CreateSecurityPolicy",
                "aoss:GetSecurityPolicy",
                "aoss:UpdateSecurityPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

角色必須具有下列信任關係，允許 OpenSearch Ingestion 擔任該角色：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "osis-pipelines.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

步驟 2：設定集合的資料和網路存取

使用下列設定建立 OpenSearch Serverless 集合。如需建立集合的指示，請參閱 the section called “建立集合”。

資料存取政策

為 集合建立資料存取政策，以授予管道角色所需的許可。例如：

```
[
  {
    "Rules": [
      {
        "Resource": [
```

```
          "index/collection-name/*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:UpdateIndex",
          "aoss:DescribeIndex",
          "aoss:WriteDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::account-id:role/pipeline-role"
    ],
    "Description": "Pipeline role access"
  }
]
```

> **Note**
>
> 在 `Principal`元素中，指定管道角色的 Amazon Resource Name (ARN)。

網路存取政策

您在 OpenSearch Serverless 中建立的每個集合至少有一個與其相關聯的網路存取政策。網路存取政策會判斷是否可透過網際網路從公有網路存取集合，或是否必須私下存取集合。如需網路政策的詳細資訊，請參閱 the section called “網路存取”。

在網路存取政策中，您只能指定 OpenSearch Serverless 受管 VPC 端點。如需詳細資訊，請參閱the section called “資料平面 VPC 端點”。不過，為了讓管道寫入集合，政策也必須授予對 OpenSearch Ingestion 在管道與集合之間自動建立的 VPC 端點的存取權。因此，如果您選擇 OpenSearch Serverless 集合做為管道的目的地目的地目的地，則必須在網路政策名稱欄位中輸入相關聯的網路政策名稱。

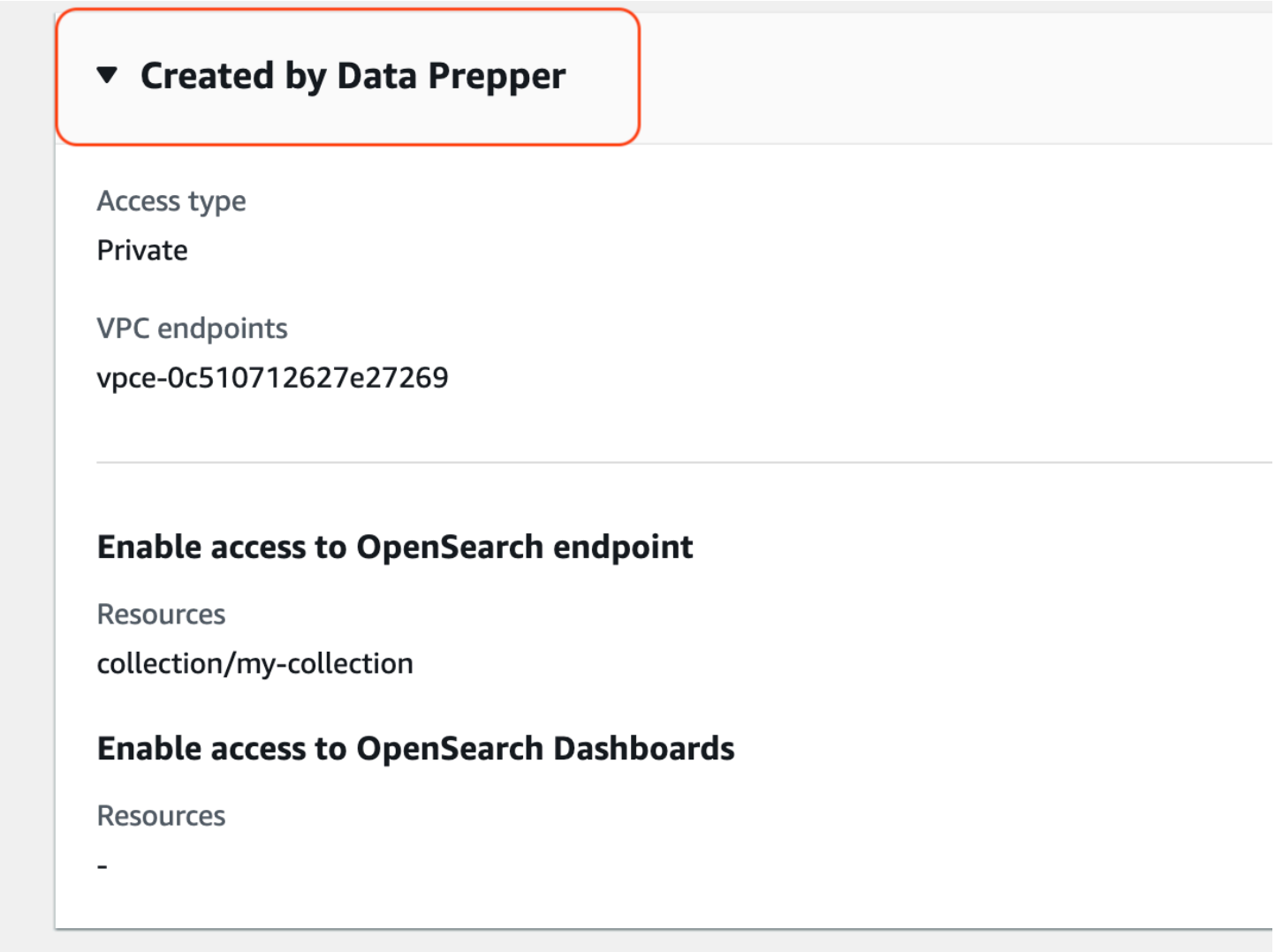
在管道建立期間，OpenSearch Ingestion 會檢查指定網路政策是否存在。如果不存在，OpenSearch Ingestion 會建立它。如果它確實存在，OpenSearch Ingestion 會透過新增規則來更新它。此規則會授予連線管道和集合的 VPC 端點存取權。

例如：

```
{
```

```
   "Rules":[
      {
         "Resource":[
            "collection/my-collection"
         ],
         "ResourceType":"collection"
      }
   ],
   "SourceVPCEs":[
      "vpce-0c510712627e27269" # The ID of the VPC endpoint that OpenSearch Ingestion
 creates between the pipeline and collection
   ],
   "Description":"Created by Data Prepper"
}
```

在 主控台中，OpenSearch Ingestion 新增至網路政策的任何規則都名為 Created by Data Prepper：



> **Note**
>
> 一般而言，指定集合公有存取權的規則會覆寫指定私有存取權的規則。因此，如果政策已設定公開存取，則 OpenSearch Ingestion 新增的這個新規則實際上不會變更政策的行為。如需詳細資訊，請參閱the section called “政策優先順序”。

如果您停止或刪除管道，OpenSearch Ingestion 會刪除管道與集合之間的 VPC 端點。它也會修改網路政策，從允許的端點清單中移除 VPC 端點。如果您重新啟動管道，它會重新建立 VPC 端點，並使用端點 ID 重新更新網路政策。

Amazon OpenSearch Ingestion 入門

Amazon OpenSearch Ingestion 支援將資料擷取至受管 OpenSearch Service 網域和 OpenSearch Serverless 集合。下列教學課程會逐步引導您完成啟動和執行管道的基本步驟。

第一個教學課程說明如何使用 Amazon OpenSearch Ingestion 設定簡單的管道，並將資料擷取至 Amazon OpenSearch Service 網域。

第二個教學課程說明如何使用 Amazon OpenSearch Ingestion 設定簡單的管道，並將資料擷取至 Amazon OpenSearch Serverless 集合。

> **Note**
>
> 如果您未設定正確的許可，管道建立將會失敗。在建立管道之前，the section called “設定角色和使用者”請參閱 以進一步了解所需的角色。

主題

- 教學課程：使用 Amazon OpenSearch Ingestion 將資料擷取至網域
- 教學課程：使用 Amazon OpenSearch Ingestion 將資料擷取至集合

教學課程：使用 Amazon OpenSearch Ingestion 將資料擷取至網域

本教學課程說明如何使用 Amazon OpenSearch Ingestion 設定簡單的管道，並將資料擷取至 Amazon OpenSearch Service 網域。管道是 OpenSearch Ingestion 佈建和管理的資源。您可以使用管道來篩選、擴充、轉換、標準化和彙總資料，以在 OpenSearch Service 中進行下游分析和視覺化。

本教學課程會逐步引導您快速啟動和執行管道的基本步驟。如需更完整的說明，請參閱 the section called “建立管道”。

在本教學課程中，您會完成下列步驟：

1. 建立網域。
2. 建立管道。
3. 擷取一些範例資料。

在教學課程中，您將建立下列資源：

- 管道寫入ingestion-domain的名為 的網域
- 名為 的管道 ingestion-pipeline

所需的許可

若要完成本教學課程，您的使用者或角色必須具有具有下列最低許可的連接身分型政策。這些許可可讓您建立管道角色並連接政策 (iam:Create* 和)iam:Attach*、建立或修改網域 (es:*)，以及使用管道 ()osis:*。

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:*",
            "iam:Create*",
            "iam:Attach*",
            "es:*"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::111122223333:role/OpenSearchIngestion-PipelineRole"
         ],
```

```
         "Effect":"Allow",
         "Action":[
            "iam:CreateRole",
            "iam:AttachRolePolicy",
            "iam:PassRole"
         ]
      }
   ]
}
```

步驟 1：建立管道角色

首先，建立管道將擔任的角色，以存取 OpenSearch Service 網域目的地。您將在本教學稍後的管道組態中包含此角色。

建立管道角色

1. 在 https：//https://console.aws.amazon.com/iamv2/ 開啟 AWS Identity and Access Management 主控台。
2. 選擇政策，然後選擇建立政策。
3. 在本教學課程中，您會將資料擷取至名為 的網域`ingestion-domain`，您將在下一個步驟中建立該網域。選取 JSON，並將下列政策貼到編輯器中。將 取代*`your-account-id`*為您的帳戶 ID，並視需要修改區域。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "es:DescribeDomain",
            "Resource": "arn:aws:es:us-east-1:111122223333:domain/ingestion-
domain"
        },
        {
            "Effect": "Allow",
            "Action": "es:ESHttp*",
            "Resource": "arn:aws:es:us-east-1:111122223333:domain/ingestion-
domain/*"
```

```
        }
    ]
}
```

如果您想要將資料寫入現有網域，請將 取代`ingestion-domain`為您的網域名稱。

> **Note**
>
> 為了簡化本教學課程，我們使用廣泛的存取政策。不過，在生產環境中，建議您將更嚴格的存取政策套用至管道角色。如需提供最低必要許可的範例政策，請參閱 the section called “授予管道對網域的存取權”。

4. 選擇下一步，選擇下一步，並命名您的政策管道政策。
5. 選擇建立政策。
6. 接著，建立角色並將政策連接到該角色。選擇 Roles (角色)，然後選擇 Create role (建立角色)。
7. 選擇自訂信任政策，並將下列政策貼到編輯器中：

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"osis-pipelines.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

8. 選擇下一步。然後搜尋並選取管道政策 （您剛建立）。
9. 選擇下一步並命名角色 PipelineRole。
10. 選擇建立角色。

記住角色的 Amazon Resource Name (ARN)（例如 `arn:aws:iam::your-account-id:role/PipelineRole`）。建立管道時，您會需要它。

步驟 2：建立網域

首先，建立名為 的網域`ingestion-domain`以擷取資料。

導覽至位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台，並建立符合下列要求的網域：

- 正在執行 OpenSearch 1.0 或更新版本，或 Elasticsearch 7.4 或更新版本
- 使用公有存取
- 不使用精細存取控制

> **Note**
>
> 這些要求旨在確保本教學課程中的簡單性。在生產環境中，您可以使用 VPC 存取和/或使用精細存取控制來設定網域。若要使用精細存取控制，請參閱映射管道角色。

網域必須具有授予 IAM `OpenSearchIngestion-PipelineRole` 角色許可的存取政策，OpenSearch Service 將在下一個步驟中為您建立該角色。管道將擔任此角色，以便將資料傳送至網域目的地。

請確定網域具有下列網域層級存取政策，這會授予管道角色對網域的存取權。將區域和帳戶 ID 取代為您自己的 ID：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/OpenSearchIngestion-PipelineRole"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/ingestion-domain/*"
```

```
    }
  ]
}
```

如需建立網域層級存取政策的詳細資訊，請參閱 the section called “資源型政策”。

如果您已建立網域，請修改其現有的存取政策，以提供上述許可給 `OpenSearchIngestion-PipelineRole`。

步驟 3：建立管道

現在您已擁有網域，您可以建立管道。

建立管道

1. 在 Amazon OpenSearch Service 主控台中，從左側導覽窗格中選擇管道。
2. 選擇 Create pipeline (建立管道)。
3. 選取空白管道，然後選擇選取藍圖。
4. 在本教學課程中，我們將建立使用 HTTP 來源外掛程式的簡單管道。外掛程式接受 JSON 陣列格式的日誌資料。我們將指定單一 OpenSearch Service 網域做為接收器，並將所有資料擷取至`application_logs`索引。

   在來源功能表中，選擇 HTTP。針對路徑，輸入 /logs。
5. 為了簡化本教學課程，我們將設定管道的公有存取。針對來源網路選項，選擇公開存取。如需設定 VPC 存取的資訊，請參閱 the section called “設定管道的 VPC 存取”。
6. 選擇下一步。
7. 針對處理器，輸入日期，然後選擇新增。
8. 啟用從接收到的時間。將所有其他設定保留為預設值。
9. 選擇下一步。
10. 設定接收器詳細資訊。針對 OpenSearch 資源類型，選擇受管叢集。然後選擇您在上一節中建立的 OpenSearch Service 網域。

    針對索引名稱，輸入 application_logs。如果網域中尚不存在，OpenSearch Ingestion 會自動建立此索引。
11. 選擇下一步。
12. 命名管道擷取管道。將容量設定保留為預設值。

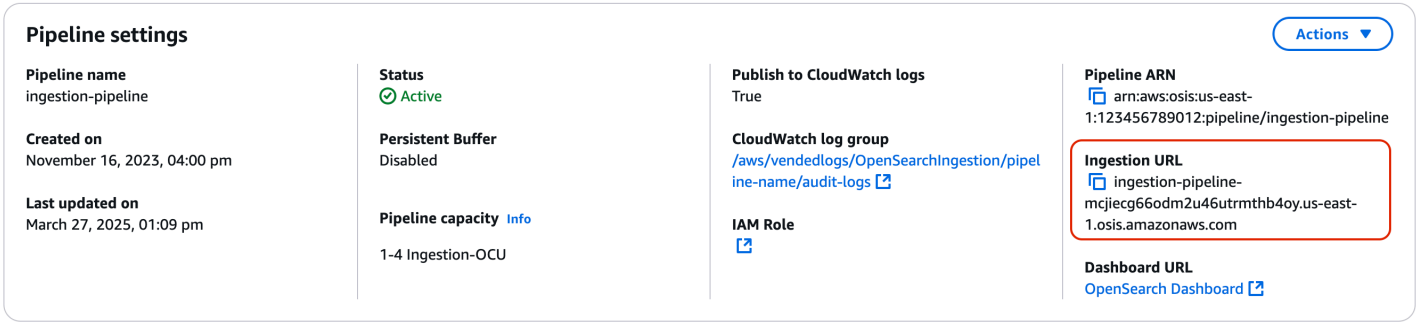
13. 針對管道角色，選取建立並使用新的服務角色。管道角色為管道提供寫入網域目的地並從提取型來源讀取所需的許可。透過選取此選項，您可以允許 OpenSearch Ingestion 為您建立角色，而不是在 IAM 中手動建立角色。如需詳細資訊，請參閱the section called “設定角色和使用者”。
14. 針對服務角色名稱尾碼，輸入 PipelineRole。在 IAM 中，角色的格式為 `arn:aws:iam::`*`your-account-id`*`:role/OpenSearchIngestion-`**`PipelineRole`**。
15. 選擇下一步。檢閱您的管道組態，然後選擇建立管道。管道需要 5-10 分鐘才會變成作用中。

步驟 4：擷取一些範例資料

當管道狀態為 時`Active`，您可以開始將資料導入其中。您必須使用 Signature 第 4 版簽署管道的所有 HTTP 請求。使用 Postman 或 awscurl 等 HTTP 工具，將一些資料傳送至管道。如同將資料直接索引至網域一樣，將資料擷取至管道一律需要 IAM 角色或 IAM 存取金鑰和私密金鑰。

> **Note**
>
> 簽署請求的委託人必須具有 `osis:Ingest` IAM 許可。

首先，從管道設定頁面取得擷取 URL：



然後，擷取一些範例資料。下列請求使用 awscurl 將單一日誌檔案傳送至管道：

```
awscurl --service osis --region us-east-1 \
    -X POST \
    -H "Content-Type: application/json" \
    -d
 '[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","request":
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]' \
    https://pipeline-endpoint.us-east-1.osis.amazonaws.com/logs
```

您應該會看到200 OK回應。如果您收到身分驗證錯誤，可能是因為您從獨立帳戶擷取資料，而不是管道所在的資料。請參閱 the section called “修正許可問題”。

現在，請查詢application_logs索引，以確保您的日誌項目已成功擷取：

```
awscurl --service es --region us-east-1 \
     -X GET \
     https://search-ingestion-domain.us-east-1.es.amazonaws.com/application_logs/
_search | json_pp
```

回應範例：

```
{
   "took":984,
   "timed_out":false,
   "_shards":{
      "total":1,
      "successful":5,
      "skipped":0,
      "failed":0
   },
   "hits":{
      "total":{
         "value":1,
         "relation":"eq"
      },
      "max_score":1.0,
      "hits":[
         {
            "_index":"application_logs",
            "_type":"_doc",
            "_id":"z6VY_IMBRpceX-DU6V4O",
            "_score":1.0,
            "_source":{
               "time":"2014-08-11T11:40:13+00:00",
               "remote_addr":"122.226.223.69",
               "status":"404",
               "request":"GET http://www.k2proxy.com//hello.html HTTP/1.1",
               "http_user_agent":"Mozilla/4.0 (compatible; WOW64; SLCC2;)",
               "@timestamp":"2022-10-21T21:00:25.502Z"
            }
         }
      ]
```

```
    }
}
```

修正許可問題

如果您遵循教學課程中的步驟，並且在嘗試擷取資料時仍看到身分驗證錯誤，可能是因為寫入管道的角色與管道本身 AWS 帳戶 不同。在這種情況下，您需要建立並擔任特別可讓您擷取資料的角色。如需說明，請參閱the section called “提供跨帳戶擷取存取權”。

相關資源

本教學課程提供透過 HTTP 擷取單一文件的簡單使用案例。在生產案例中，您將設定用戶端應用程式（例如 Fluent Bit、Kubernetes 或 OpenTelemetry Collector) 將資料傳送至一或多個管道。您的管道可能比本教學課程中的簡單範例更複雜。

若要開始設定用戶端和擷取資料，請參閱下列資源：

- 建立和管理管道
- 設定您的用戶端將資料傳送至 OpenSearch Ingestion
- Data Prepper 文件

教學課程：使用 Amazon OpenSearch Ingestion 將資料擷取至集合

本教學課程說明如何使用 Amazon OpenSearch Ingestion 設定簡單的管道，並將資料擷取至 Amazon OpenSearch Serverless 集合。管道是 OpenSearch Ingestion 佈建和管理的資源。您可以使用管道來篩選、擴充、轉換、標準化和彙總資料，以在 OpenSearch Service 中進行下游分析和視覺化。

如需示範如何將資料擷取到佈建 OpenSearch Service 網域的教學課程，請參閱 the section called “教學課程：將資料擷取至網域”。

您將完成本教學課程中的下列步驟：。

1. 建立集合。
2. 建立管道。
3. 擷取一些範例資料。

在教學課程中，您將建立下列資源：

- 管道將寫入ingestion-collection的名為 的集合
- 名為 的管道 ingestion-pipeline-serverless

所需的許可

若要完成本教學課程，您的使用者或角色必須具有具有下列最低許可的連接身分型政策。這些許可可讓您建立管道角色並連接政策 (iam:Create* 和)iam:Attach*、建立或修改集合 (aoss:*)，以及使用管道 ()osis:*。

此外，需要數個 IAM 許可，才能自動建立管道角色並將其傳遞給 OpenSearch Ingestion，以便將資料寫入集合。

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:*",
            "iam:Create*",
            "iam:Attach*",
            "aoss:*"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::111122223333:role/OpenSearchIngestion-PipelineRole"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:CreateRole",
            "iam:AttachRolePolicy",
            "iam:PassRole"
         ]
      }
   ]
}
```

步驟 1：建立集合

首先，建立要擷取資料的集合。我們將集合命名為 `ingestion-collection`。

1. 導覽至 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 從左側導覽中選擇集合，然後選擇建立集合。
3. 命名集合擷取集合。
4. 針對安全性，選擇標準建立。
5. 在網路存取設定下，將存取類型變更為公有。
6. 將其他所有設定保留為預設值，然後選擇 Next (下一步)。
7. 現在，為集合設定資料存取政策。取消選取自動比對存取政策設定。
8. 針對定義方法，選擇 JSON，並將下列政策貼到編輯器中。此政策會執行兩個動作：

   - 允許管道角色寫入集合。
   - 可讓您從集合讀取 。稍後，在您將一些範例資料擷取至管道後，您將查詢集合，以確保資料已成功擷取並寫入索引。

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/ingestion-collection/*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:UpdateIndex",
          "aoss:DescribeIndex",
          "aoss:ReadDocument",
          "aoss:WriteDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::your-account-id:role/OpenSearchIngestion-PipelineRole",
      "arn:aws:iam::your-account-id:role/Admin"
    ],
```

```
    "Description": "Rule 1"
  }
]
```

9. 修改Principal元素以包含您的 AWS 帳戶 ID。針對第二個委託人，指定可用於稍後查詢集合的使用者或角色。
10. 選擇下一步。命名存取政策pipeline-collection-access，然後再次選擇下一步。
11. 檢閱集合組態，然後選擇 Submit (提交)。

步驟 2：建立管道

現在您已擁有集合，您可以建立管道。

建立管道

1. 在 Amazon OpenSearch Service 主控台中，從左側導覽窗格中選擇管道。
2. 選擇 Create pipeline (建立管道)。
3. 選取空白管道，然後選擇選取藍圖。
4. 在本教學課程中，我們將建立使用 HTTP 來源外掛程式的簡單管道。外掛程式接受 JSON 陣列格式的日誌資料。我們將指定單一 OpenSearch Serverless 集合做為接收器，並將所有資料擷取至my_logs索引。

   在來源功能表中，選擇 HTTP。針對路徑，輸入 /logs。
5. 為了簡化本教學課程，我們將設定管道的公有存取。針對來源網路選項，選擇公有存取。如需設定 VPC 存取的資訊，請參閱 the section called “設定管道的 VPC 存取”。
6. 選擇下一步。
7. 針對處理器，輸入日期，然後選擇新增。
8. 啟用從接收到的時間。將所有其他設定保留為預設值。
9. 選擇下一步。
10. 設定接收器詳細資訊。針對 OpenSearch 資源類型，選擇集合 （無伺服器）。然後選擇您在上一節中建立的 OpenSearch Service 集合。

    將網路政策名稱保留為預設值。針對索引名稱，輸入 my_logs。如果集合中尚不存在，OpenSearch Ingestion 會自動建立此索引。
11. 選擇下一步。

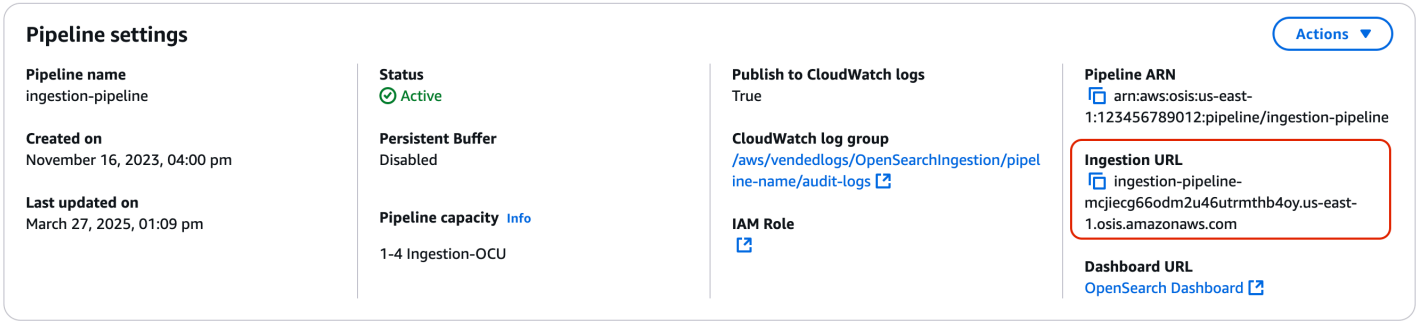
12. 命名管道 ingestion-pipeline-serverless。將容量設定保留為預設值。
13. 對於管道角色，選取建立並使用新的服務角色。管道角色為管道提供寫入集合目的地並從提取型來源讀取所需的許可。透過選取此選項，您可以允許 OpenSearch Ingestion 為您建立角色，而不是在 IAM 中手動建立角色。如需詳細資訊，請參閱the section called “設定角色和使用者”。
14. 針對服務角色名稱尾碼，輸入 PipelineRole。在 IAM 中，角色的格式為 `arn:aws:iam::`*`your-account-id`*`:role/OpenSearchIngestion-`**`PipelineRole`**。
15. 選擇下一步。檢閱您的管道組態，然後選擇建立管道。管道需要 5-10 分鐘才會變成作用中。

步驟 3：擷取一些範例資料

當管道狀態為 時`Active`，您可以開始將資料導入其中。您必須使用 Signature 第 4 版簽署管道的所有 HTTP 請求。使用 Postman 或 awscurl 等 HTTP 工具，將一些資料傳送至管道。如同將資料直接索引至集合一樣，將資料擷取至管道一律需要 IAM 角色或 IAM 存取金鑰和私密金鑰。

> **Note**
>
> 簽署請求的委託人必須具有 `osis:Ingest` IAM 許可。

首先，從管道設定頁面取得擷取 URL：



然後，將一些範例資料傳送至擷取路徑。下列範例請求使用 awscurl 將單一日誌檔案傳送至管道：

```
awscurl --service osis --region us-east-1 \
    -X POST \
    -H "Content-Type: application/json" \
    -d
 '[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","request":
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]' \
```

```
    https://pipeline-endpoint.us-east-1.osis.amazonaws.com/logs
```

您應該會看到200 OK回應。

現在，請查詢my_logs索引，以確保已成功擷取日誌項目：

```
awscurl --service aoss --region us-east-1 \
     -X GET \
     https://collection-id.us-east-1.aoss.amazonaws.com/my_logs/_search | json_pp
```

回應範例：

```
{
   "took":348,
   "timed_out":false,
   "_shards":{
      "total":0,
      "successful":0,
      "skipped":0,
      "failed":0
   },
   "hits":{
      "total":{
         "value":1,
         "relation":"eq"
      },
      "max_score":1.0,
      "hits":[
         {
            "_index":"my_logs",
            "_id":"1%3A0%3ARJgDvIcBTy5m12xrKE-y",
            "_score":1.0,
            "_source":{
               "time":"2014-08-11T11:40:13+00:00",
               "remote_addr":"122.226.223.69",
               "status":"404",
               "request":"GET http://www.k2proxy.com//hello.html HTTP/1.1",
               "http_user_agent":"Mozilla/4.0 (compatible; WOW64; SLCC2;)",
               "@timestamp":"2023-04-26T05:22:16.204Z"
            }
         }
      ]
   }
```

```
}
```

相關資源

本教學課程提供透過 HTTP 擷取單一文件的簡單使用案例。在生產案例中，您將設定用戶端應用程式（例如 Fluent Bit、Kubernetes 或 OpenTelemetry Collector) 將資料傳送至一或多個管道。您的管道可能比本教學課程中的簡單範例更複雜。

若要開始設定用戶端和擷取資料，請參閱下列資源：

- 建立和管理管道
- 設定您的用戶端將資料傳送至 OpenSearch Ingestion
- Data Prepper 文件

Amazon OpenSearch Ingestion 中的管道功能概觀

Amazon OpenSearch Ingestion 佈建管道，其中包含來源、緩衝區、零或多個處理器，以及一或多個接收器。擷取管道採用 Data Prepper 做為資料引擎。如需管道各種元件的概觀，請參閱 the section called “重要概念”。

下列各節提供 Amazon OpenSearch Ingestion 中一些最常用功能的概觀。

> **Note**
>
> 這不是可供管道使用的完整功能清單。如需所有可用管道功能的完整文件，請參閱 Data Prepper 文件。請注意，OpenSearch Ingestion 會限制您可以使用的外掛程式和選項。如需詳細資訊，請參閱the section called “支援的外掛程式和選項”。

主題

- 持久性緩衝
- 分割
- 鏈接
- 無效信件佇列
- 索引管理
- End-to-end認可

- 來源背壓

持久性緩衝

持久性緩衝會將您的資料儲存在跨多個可用區域的磁碟型緩衝區中，以增強資料耐久性。您可以使用持久性緩衝，從所有支援的推送型來源擷取資料，而無需設定獨立的緩衝。這些來源包括日誌、追蹤和指標的 HTTP 和 OpenTelemetry。若要啟用持久性緩衝，請在建立或更新管道時選擇啟用持久性緩衝。如需詳細資訊，請參閱the section called “建立管道”。

OpenSearch Ingestion 會動態決定用於持久性緩衝、考量資料來源、串流轉換和目的地的 OCUs 數量。由於它將一些 OCUs 配置為緩衝，您可能需要增加最小和最大 OCUs以維持相同的擷取輸送量。管道會在緩衝區中保留資料長達 72 小時。

如果您為管道啟用持久性緩衝，預設請求承載大小上限如下：

- HTTP 來源 – 10 MB
- OpenTelemetry 來源 – 4 MB

對於 HTTP 來源，您可以將承載大小上限提高到 20 MB。請求承載大小包含整個 HTTP 請求，通常包含多個事件。每個事件不能超過 3.5 MB。

具有持久性緩衝的管道會在運算和緩衝單位之間分割設定的管道單位。如果管道使用 grok、key-value 或 split string 等 CPU 密集型處理器，則會以 1：1 buffer-to-compute比率配置單位。否則，它會以 3：1 的比例配置它們，一律偏好運算單位。

例如：

- 具有 grok 和最多 2 個單位的管道 – 1 個運算單位和 1 個緩衝單位
- 具有 grok 和最多 5 個單位的管道 – 3 個運算單位和 2 個緩衝單位
- 沒有處理器和最多 2 個單位的管道 – 1 個運算單位和 1 個緩衝單位
- 沒有處理器和最多 4 個單位的管道 – 1 個運算單位和 3 個緩衝單位
- 具有 grok 和最多 5 個單位的管道 – 2 個運算單位和 3 個緩衝單位

根據預設，管道會使用 AWS 擁有的金鑰 來加密緩衝區資料。這些管道不需要管道角色的任何其他許可。

或者，您可以指定客戶受管金鑰，並將下列 IAM 許可新增至管道角色：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "KeyAccess",
            "Effect": "Allow",
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "arn:aws:kms:us-
east-1:111122223333:key/ASIAIOSFODNN7EXAMPLE"
        }
    ]
}
```

如需更多資訊，請參閱 AWS Key Management Service 開發人員指南中的客戶受管金鑰。

> **Note**
>
> 如果您停用持久性緩衝，您的管道會開始完全在記憶體內緩衝上執行。

分割

您可以設定 OpenSearch Ingestion 管道，將傳入的事件分割為子管道，讓您在相同的傳入事件上執行不同類型的處理。

下列範例管道會將傳入的事件分割成兩個子管道。每個子管道都會使用自己的處理器來充實和操作資料，然後將資料傳送至不同的 OpenSearch 索引。

```
version: "2"
log-pipeline:
  source:
    http:
    ...
  sink:
    - pipeline:
```

```
        name: "logs_enriched_one_pipeline"
    - pipeline:
        name: "logs_enriched_two_pipeline"

logs_enriched_one_pipeline:
  source:
    pipeline:
      name: "log-pipeline"
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
        index: "enriched_one_logs"

logs_enriched_two_pipeline:
  source:
    pipeline:
      name: "log-pipeline"
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_two_logs"
```

鏈接

您可以將多個子管道鏈結在一起，以便在區塊中執行資料處理和擴充。換言之，您可以在一個子管道中使用特定處理功能來豐富傳入事件，然後將其傳送至另一個子管道，以使用不同的處理器進行額外擴充，最後將其傳送至其 OpenSearch 接收器。

在下列範例中，`log_pipeline`子管道會使用一組處理器來充實傳入日誌事件，然後將事件傳送至名為 的 OpenSearch 索引`enriched_logs`。管道會將相同的事件傳送至`log_advanced_pipeline`子管道，以處理該事件並將其傳送至名為 的不同 OpenSearch 索引`enriched_advanced_logs`。

```
version: "2"
log-pipeline:
  source:
    http:
    ...
  processor:
    ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_logs"
    - pipeline:
        name: "log_advanced_pipeline"

log_advanced_pipeline:
  source:
    pipeline:
      name: "log-pipeline"
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_advanced_logs"
```

無效信件佇列

無效字母佇列 (DLQs) 是管道無法寫入目的地的事件目的地。在 OpenSearch Ingestion 中，您必須指定具有適當寫入許可的 Amazon S3 儲存貯體，以用作 DLQ。您可以將 DLQ 組態新增至管道中的每個接收器。當管道遇到寫入錯誤時，它會在設定的 S3 儲存貯體中建立 DLQ 物件。DLQ 物件存在於 JSON 檔案中，做為失敗事件的陣列。

當符合下列任一條件時，管道會將事件寫入 DLQ：

- OpenSearch 接收器的重試次數上限已用盡。此設定至少需要 16 個 OpenSearch Ingestion。
- 目的地因為錯誤條件而拒絕事件。

Configuration

若要設定子管道的無效字母佇列，請在設定目的地時選擇啟用 S3 DLQ。然後，指定佇列的必要設定。如需詳細資訊，請參閱 Data Prepper DLQ 文件中的組態。

寫入此 S3 DLQ 的檔案具有下列命名模式：

```
dlq-v${version}-${pipelineName}-${pluginId}-${timestampIso8601}-${uniqueId}
```

如需手動設定管道角色以允許存取 DLQ 寫入的 S3 儲存貯體的說明，請參閱 the section called “寫入 Amazon S3 或無效字母佇列的許可”。

範例

請考慮下列範例 DLQ 檔案：

```
dlq-v2-apache-log-pipeline-opensearch-2023-04-05T15:26:19.152938Z-e7eb675a-
f558-4048-8566-dac15a4f8343
```

以下是無法寫入目的地，並傳送至 DLQ S3 儲存貯體以進行進一步分析的資料範例：

```
Record_0
pluginId            "opensearch"
pluginName          "opensearch"
pipelineName        "apache-log-pipeline"
failedData
index    "logs"
indexId   null
status    0
message  "Number of retries reached the limit of max retries (configured value 15)"
document
log      "sample log"
timestamp     "2023-04-14T10:36:01.070Z"

Record_1
pluginId            "opensearch"
pluginName          "opensearch"
pipelineName        "apache-log-pipeline"
```

```
failedData
index               "logs"
indexId   null
status    0
message  "Number of retries reached the limit of max retries (configured value 15)"
document
log                 "another sample log"
timestamp           "2023-04-14T10:36:01.071Z"
```

索引管理

Amazon OpenSearch Ingestion 有許多索引管理功能，包括下列項目。

建立索引

您可以在管道接收器中指定索引名稱，OpenSearch Ingestion 會在佈建管道時建立索引。如果索引已存在，管道會使用它來為傳入事件編製索引。如果您停止並重新啟動管道，或更新其 YAML 組態，管道會嘗試在尚未存在時建立新索引。管道永遠無法刪除索引。

佈建管道時，下列範例接收器會建立兩個索引：

```
sink:
  - opensearch:
      index: apache_logs
  - opensearch:
      index: nginx_logs
```

產生索引名稱和模式

您可以使用傳入事件欄位中的變數來產生動態索引名稱。在接收器組態中，使用 格式`string ${}`來發出字串插補訊號，並使用 JSON 指標從事件擷取欄位。的選項`index_type`為`custom`或 `management_disabled`。由於 `custom` OpenSearch 網域`index_type`預設為 `management_disabled` ，OpenSearch Serverless 集合預設為 ，因此可以保持取消設定。

例如，下列管道會從傳入事件中選取 `metadataType` 欄位，以產生索引名稱。

```
pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}"
```

下列組態會持續每天或每小時產生新的索引。

```
pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}-%{yyyy.MM.dd}"

pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}-%{yyyy.MM.dd.HH}"
```

索引名稱也可以是純字串，並以日期時間模式做為尾碼，例如 `my-index-%{yyyy.MM.dd}`。當接收器將資料傳送至 OpenSearch 時，它會將日期時間模式取代為 UTC 時間，並為每天建立新的索引，例如 `my-index-2022.01.25`。如需詳細資訊，請參閱 DateTimeFormatter 類別。

此索引名稱也可以是格式化字串（有或沒有日期時間模式尾碼），例如 `my-${index}-name`。當接收將資料傳送至 OpenSearch 時，它會將`"${index}"`部分取代為正在處理的事件中的 值。如果格式為 `"${index1/index2/index3}"`，則會將 欄位取代`index1/index2/index3`為事件中的值。

產生文件 IDs

將文件編製索引至 OpenSearch 時，管道可以產生文件 ID。它可以從傳入事件中的欄位推斷這些文件 IDs。

此範例使用來自傳入事件的 `uuid` 欄位來產生文件 ID。

```
pipeline:
  ...
  sink:
    opensearch:
      index_type: custom
      index: "metadata-${metadataType}-%{yyyy.MM.dd}"
      "document_id": "uuid"
```

在下列範例中，新增項目處理器會合併`other_field`傳入事件中的欄位 `uuid`和 ，以產生文件 ID。

`create` 動作可確保具有相同 IDs的文件不會遭到覆寫。管道會捨棄重複的文件，而沒有任何重試或 DLQ 事件。對於使用此動作的管道作者來說，這是合理的預期，因為目標是避免更新現有的文件。

```
pipeline:
  ...
  processor:
   - add_entries:
      entries:
        - key: "my_doc_id_field"
          format: "${uuid}-${other_field}"
  sink:
    - opensearch:
       ...
       action: "create"
       document_id: "my_doc_id"
```

您可能想要將事件的文件 ID 設定為子物件中的欄位。在下列範例中，OpenSearch 接收器外掛程式會使用 子物件`info/id`來產生文件 ID。

```
sink:
  - opensearch:
       ...
       document_id: info/id
```

鑑於下列事件，管道將產生 `_id` 欄位設定為 的文件`json001`：

```
{
   "fieldA":"arbitrary value",
   "info":{
      "id":"json001",
      "fieldA":"xyz",
      "fieldB":"def"
   }
}
```

產生路由 IDs

您可以使用 OpenSearch 接收器外掛程式中的 `routing_field`選項，將文件路由屬性 (`_routing`) 的值設定為來自傳入事件的值。

路由支援 JSON 指標語法，因此也提供巢狀欄位，而不只是最上層欄位。

```
sink:
  - opensearch:
```

```
        ...
        routing_field: metadata/id
        document_id: id
```

鑑於下列事件，外掛程式會產生將 _routing 欄位設定為 的文件abcd：

```
{
   "id":"123",
   "metadata":{
      "id":"abcd",
      "fieldA":"valueA"
   },
   "fieldB":"valueB"
}
```

如需建立管道可在索引建立期間使用的索引範本的說明，請參閱索引範本。

End-to-end認可

OpenSearch Ingestion 使用end-to-end確認，追蹤其從來源到無狀態管道中接收器的交付，以確保資料的耐用性和可靠性。

> **Note**
>
> 目前，只有 S3 來源外掛程式支援end-to-end確認。

透過end-to-end確認，管道來源外掛程式會建立確認集來監控一批事件。當這些事件成功傳送到其目的地時，會收到正面的確認，或當任何事件無法傳送到其目的地時，會收到負面的確認。

如果管道元件發生故障或當機，或來源無法接收確認，則來源會逾時，並採取必要動作，例如重試或記錄故障。如果管道已設定多個接收器或多個子管道，則只有在事件傳送至所有子管道中的所有接收器之後，才會傳送事件層級確認。如果接收器已設定 DLQ，end-to-end確認也會追蹤寫入 DLQ 的事件。

若要啟用end-to-end確認，請展開 Amazon S3 來源組態中的其他選項，然後選擇啟用end-to-end訊息確認。

來源背壓

當管道處理資料忙碌，或其接收器暫時停機或擷取資料的速度變慢時，管道可能會遇到背壓。OpenSearch Ingestion 有不同的處理背壓方式，取決於管道使用的來源外掛程式。

HTTP 來源

使用 HTTP 來源外掛程式的管道會根據擁塞的管道元件，以不同的方式處理背壓：

- 緩衝區 – 當緩衝區已滿時，管道會開始將錯誤碼REQUEST_TIMEOUT為 408 的 HTTP 狀態傳回至來源端點。當緩衝區釋放時，管道會再次開始處理 HTTP 事件。
- 來源執行緒 – 當所有 HTTP 來源執行緒都忙於執行請求，且未處理的請求佇列大小已超過允許的最大請求數時，管道會開始將錯誤碼為 429 TOO_MANY_REQUESTS的 HTTP 狀態傳回來源端點。當請求佇列低於允許的佇列大小上限時，管道會再次開始處理請求。

OTel 來源

當使用 OpenTelemetry 來源 (OTel 日誌、OTel 指標和 OTel 追蹤) 的管道緩衝區已滿時，管道會開始將REQUEST_TIMEOUT錯誤碼為 408 的 HTTP 狀態傳回來源端點。當緩衝區釋放時，管道會再次開始處理事件。

S3 來源

當具有 S3 來源的管道緩衝區已滿時，管道會停止處理 SQS 通知。當緩衝區釋放時，管道會再次開始處理通知。

如果目的地關閉或無法擷取資料，且來源已啟用end-to-end確認，管道會停止處理 SQS 通知，直到收到來自所有目的地的成功確認為止。

建立 Amazon OpenSearch Ingestion 管道

管道是 Amazon OpenSearch Ingestion 用來將資料從其來源 （資料來源） 移動到其目的地 （資料來源） 的機制。在 OpenSearch Ingestion 中，接收器一律是單一 Amazon OpenSearch Service 網域，而資料來源可能是 Amazon S3、Fluent Bit 或 OpenTelemetry Collector 等用戶端。

如需詳細資訊，請參閱 OpenSearch 文件中的管道。

主題

- 先決條件和必要的 IAM 角色
- 所需的 IAM 許可
- 指定管道版本
- 指定擷取路徑
- 建立管道

- 追蹤管道建立的狀態
- 使用藍圖

先決條件和必要的 IAM 角色

若要建立 OpenSearch Ingestion 管道，您必須擁有下列資源：

- OpenSearch Ingestion 擔任的 IAM 角色稱為管道角色，以便寫入目的地。您可以事先建立此角色，也可以讓 OpenSearch Ingestion 在您建立管道時自動建立角色。
- 做為接收器的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如果您要寫入網域，它必須執行 OpenSearch 1.0 或更新版本，或 Elasticsearch 7.4 或更新版本。目的地必須具有存取政策，將適當的許可授予您的 IAM 管道角色。

如需建立這些資源的說明，請參閱下列主題：

- the section called “授予管道對網域的存取權”
- the section called “授予管道對集合的存取權”

> **Note**
>
> 如果您要寫入使用精細存取控制的網域，則需要完成額外的步驟。請參閱 the section called “映射管道角色（僅適用於使用精細存取控制的網域）”。

所需的 IAM 許可

OpenSearch Ingestion 使用下列 IAM 許可來建立管道：

- `osis:CreatePipeline` – 建立管道。
- `osis:ValidatePipeline` – 檢查管道組態是否有效。
- `iam:CreateRole` 和 `iam:AttachPolicy` – 讓 OpenSearch Ingestion 自動為您建立管道角色。
- `iam:PassRole` – 將管道角色傳遞至 OpenSearch Ingestion，以便將資料寫入網域。此許可必須位於管道角色資源上，或者*如果您計劃在每個管道中使用不同的角色。

例如，下列政策會授予建立管道的許可：

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:CreatePipeline",
            "osis:ListPipelineBlueprints",
            "osis:ValidatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::111122223333:role/pipeline-role"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:CreateRole",
            "iam:AttachRolePolicy",
            "iam:PassRole"
         ]
      }
   ]
}
```

OpenSearch Ingestion 也包含名為 的許可`osis:Ingest`，這是使用 Signature 第 4 版將已簽署的請求傳送至管道的必要許可。如需詳細資訊，請參閱the section called “建立擷取角色”。

> **Note**
>
> 此外，第一個在帳戶中建立管道的使用者必須具有 `iam:CreateServiceLinkedRole`動作的許可。如需詳細資訊，請參閱管道角色資源。

如需每個許可的詳細資訊，請參閱《服務授權參考》中的 OpenSearch Ingestion 的動作、資源和條件索引鍵。

指定管道版本

當您使用組態編輯器建立管道時，您必須指定管道將執行的主要 Data Prepper 版本。若要指定 版本，請在管道組態中包含 `version`選項：

```
version: "2"
log-pipeline:
  source:
    ...
```

當您選擇建立時，OpenSearch Ingestion 會決定您指定之主要版本的最新可用次要版本，並使用該版本佈建管道。例如，如果您指定 `version: "2"`，且 Data Prepper 的最新支援版本為 2.1.1，則 OpenSearch Ingestion 會將您的管道佈建為 2.1.1 版。我們不會公開顯示管道正在執行的次要版本。

若要在 Data Prepper 的新主要版本可用時升級管道，請編輯管道組態並指定新版本。您無法將管道降級至舊版。

> **Note**
>
> OpenSearch Ingestion 不會立即支援新版本的 Data Prepper。當新版本可公開使用時，以及 OpenSearch Ingestion 中支援時，會有一些延遲。此外，OpenSearch Ingestion 可能完全不支援某些主要或次要版本。如需完整清單，請參閱the section called “支援的資料準備版本”。

每當您變更啟動藍/綠部署的管道時，OpenSearch Ingestion 都可以將其升級至目前為管道設定之主要版本的最新次要版本。如需詳細資訊，請參閱the section called “管道更新的藍/綠部署”。除非您在管道組態中明確更新 `version`選項，否則 OpenSearch Ingestion 無法變更管道的主要版本。

指定擷取路徑

對於提取型來源，例如 OTel 追蹤和 OTel 指標，OpenSearch Ingestion 需要來源組態中的其他`path`選項。路徑是字串，例如 `/log/ingest`，代表擷取的 URI 路徑。此路徑會定義您用來將資料傳送至管道的 URI。

例如，假設您為具有 HTTP 來源的管道指定下列路徑：

當您將資料擷取至管道時，您必須在用戶端組態中指定下列端點：`https://pipeline-name-abc123.us-west-2.osis.amazonaws.com/my/test_path`。

路徑必須以斜線 (/) 開頭，可包含特殊字元 '-'、'_'、'.' 和 '/'，以及`${pipelineName}`預留位置。如果您使用 `${pipelineName}` (例如 `/${pipelineName}/test_path`)，OpenSearch Ingestion 會將變數取代為相關聯的子管道名稱。

建立管道

本節說明如何使用 OpenSearch Service 主控台和 建立 OpenSearch Ingestion 管道 AWS CLI。

主控台

若要建立管道，請登入位於 https：//https://console.aws.amazon.com/aos/osis/home 的 Amazon OpenSearch Service 主控台，然後選擇建立管道。

選擇空白管道，或選擇組態藍圖。藍圖包含適用於各種常見使用案例的預先設定管道。如需詳細資訊，請參閱the section called “使用藍圖”。

選擇選取藍圖。

設定來源

1. 如果您是從空白管道開始，請從下拉式選單中選取來源。可用的來源可能包括其他 AWS 服務、OpenTelemetry 或 HTTP。如需詳細資訊，請參閱the section called “整合管道”。
2. 根據您選擇的來源，為來源設定其他設定。例如，若要使用 Amazon S3 做為來源，您必須從管道指定 Amazon SQS 佇列的 URL 來接收訊息。如需支援的來源外掛程式清單及其文件的連結，請參閱 the section called “支援的外掛程式和選項”。
3. 對於某些來源，您必須指定來源網路選項。選擇 VPC 存取或公有存取。如果選擇 Public access (公開存取)，請跳到下一步驟。如果您選擇 VPC 存取，請設定下列設定：

設定	Description
端點管理	選擇您要自行建立虛擬私有雲端 (VPC) 端點，還是讓 OpenSearch Ingestion 為您建立。端點管理預設為 OpenSearch Ingestion 管理的端點。
VPC	選擇您要使用的 VPC ID。VPC 和管道必須位於相同的 中 AWS 區域。
子網路	選擇一或多個子網路。OpenSearch Service 將會在子網路中放置 VPC 端點和彈性網路介面。
安全群組	選擇一或多個 VPC 安全群組，允許您所需的應用程式在連接埠 (80 或 443) 和管道公開的通訊協定 (HTTP 或 HTTPs) 上連接 OpenSearch 擷取管道。
VPC 連接選項	如果您的來源是自我管理的端點，請將管道連接到 VPC。選擇其中一個提供的預設 CIDR 選項，或使用自訂 CIDR。

如需詳細資訊，請參閱the section called “設定管道的 VPC 存取”。

4. 選擇下一步。

設定處理器

將一或多個處理器新增至您的管道。處理器是子管道中的元件，可讓您在將記錄發佈至網域或集合目的地之前篩選、轉換和豐富事件。如需支援的處理器清單及其文件的連結，請參閱 the section called “支援的外掛程式和選項”。

您可以選擇動作並新增下列項目：

- 條件式路由 – 根據特定條件將事件路由到不同的目的地。如需詳細資訊，請參閱條件式路由。
- 子管道 – 每個子管道是單一來源、零或多個處理器和單一接收器的組合。只有一個子管道可以有外部來源。所有其他 必須具有整體管道組態中其他子管道的來源。單一管道組態可以包含 1-10 個子管道。

選擇下一步。

設定接收器

選取管道發佈記錄的目的地。每個子管道必須至少包含一個接收器。您最多可以將 10 個接收器新增至管道。

對於 OpenSearch 接收器，請設定下列欄位：

設定	Description
網路政策名稱 （僅限無伺服器接收器）	如果您選取 OpenSearch Serverless 集合，請輸入網路政策名稱。OpenSearch Ingestion 會在政策不存在時建立政策，或使用規則更新政策，以授予連線至管道和集合的 VPC 端點存取權。如需詳細資訊，請參閱the section called “授予管道對集合的存取權”。
索引名稱	管道傳送資料的索引名稱。如果不存在，OpenSearch Ingestion 會建立此索引。
索引映射選項	選擇管道如何將文件及其欄位存放和編製索引到 OpenSearch 接收器。如果您選取動態映射，OpenSearch 會在您為文件編製索引時自動新增欄位。如果您選取自訂映射，請輸入索引映射範本。如需詳細資訊，請參閱索引範本。
啟用 DLQ	設定管道的 Amazon S3 無效字母佇列 (DLQ)。如需詳細資訊，請參閱the section called “無效信件佇列”。
其他設定	設定 OpenSearch 接收器的進階選項。如需詳細資訊，請參閱 Data Prepper 文件中的組態選項。

若要新增 Amazon S3 接收器，請選擇新增接收器和 Amazon S3。如需詳細資訊，請參閱the section called “Amazon S3 做為目的地”。

選擇下一步。

設定管道

設定下列其他管道設定：

設定	Description
管道名稱	管道的唯一名稱。

設定	Description
持久性緩衝區	持久性緩衝會將您的資料儲存在跨多個可用區域的磁碟型緩衝區中。如需詳細資訊，請參閱the section called “持久性緩衝”。 如果您啟用持久性緩衝，請選取 AWS Key Management Service 金鑰來加密緩衝資料。
管道容量	最小和最大管道容量，以擷取 OpenSearch 運算單位 (OCUs。如需詳細資訊，請參閱the section called “擴展管道”。
管道角色	IAM 角色，提供管道寫入目的地並從提取型來源讀取所需的許可。您可以自行建立角色，或讓 OpenSearch Ingestion 根據您選擇的使用案例為您建立角色。 如需詳細資訊，請參閱the section called “設定角色和使用者”。
Tags (標籤)	將一或多個標籤新增至管道。如需詳細資訊，請參閱the section called “標記管道”。
日誌發佈選項	啟用管道日誌發佈至 Amazon CloudWatch Logs。我們建議您啟用日誌發佈，以便更輕鬆地對管道問題進行故障診斷。如需詳細資訊，請參閱the section called “監控管道日誌”。

選擇下一步，然後檢閱管道組態，然後選擇建立管道。

OpenSearch Ingestion 會執行非同步程序來建置管道。一旦管道狀態為 `Active`，您就可以開始擷取資料。

AWS CLI

create-pipeline 命令接受管道組態作為字串或在 .yaml 或 .json 檔案中。如果您提供組態做為字串，則必須使用 逸出每行新行\n。例如 `"log-pipeline:\n source:\n http:\n processor:\n - grok:\n ...`

下列範例命令會使用下列組態建立管道：

- 最少 4 個擷取 OCUs，最多 10 個擷取 OCUs
- 在虛擬私有雲端 (VPC) 中佈建

- 已啟用日誌發佈

```
aws osis create-pipeline \
  --pipeline-name my-pipeline \
  --min-units 4 \
  --max-units 10 \
  --log-publishing-options
 IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="MyLogGroup"} \
  --vpc-options
 SecurityGroupIds={sg-12345678,sg-9012345},SubnetIds=subnet-1212234567834asdf \
  --pipeline-configuration-body "file://pipeline-config.yaml" \
  --pipeline-role-arn  arn:aws:iam::1234456789012:role/pipeline-role
```

OpenSearch Ingestion 會執行非同步程序來建置管道。一旦管道狀態為 `Active`，您就可以開始擷取資料。若要檢查管道的狀態，請使用 GetPipeline 命令。

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 建立 OpenSearch Ingestion 管道，請呼叫 CreatePipeline 操作。

成功建立管道後，您可以設定用戶端，並開始將資料擷取到您的 OpenSearch Service 網域。如需詳細資訊，請參閱the section called “整合管道”。

追蹤管道建立的狀態

您可以追蹤管道的狀態，因為 OpenSearch Ingestion 會佈建管道，並準備擷取資料。

主控台

最初建立管道之後，它會經歷多個階段，因為 OpenSearch Ingestion 會準備擷取資料。若要檢視管道建立的各個階段，請選擇管道名稱以查看其管道設定頁面。在狀態下，選擇檢視詳細資訊。

管道在可用於擷取資料之前會經歷下列階段：

- 驗證 – 驗證管道組態。當此階段完成時，所有驗證都已成功。
- 建立環境 – 準備和佈建資源。當此階段完成時，就會建立新的管道環境。
- 部署管道 – 部署管道。當此階段完成時，管道已成功部署。
- 檢查管道運作狀態 – 檢查管道的運作狀態。當此階段完成時，所有運作狀態檢查都已通過。
- 啟用流量 – 啟用管道以擷取資料。當此階段完成時，您可以開始將資料擷取至管道。

CLI

使用 get-pipeline-change-progress 命令來檢查管道的狀態。下列 AWS CLI 請求會檢查名為 之管道的狀態my-pipeline：

```
aws osis get-pipeline-change-progress \
    --pipeline-name my-pipeline
```

回應：

```
{
   "ChangeProgressStatuses": {
      "ChangeProgressStages": [
         {
            "Description": "Validating pipeline configuration",
            "LastUpdated": 1.671055851E9,
            "Name": "VALIDATION",
            "Status": "PENDING"
         }
      ],
      "StartTime": 1.671055851E9,
      "Status": "PROCESSING",
      "TotalNumberOfStages": 5
   }
}
```

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 追蹤管道建立的狀態，請呼叫 GetPipelineChangeProgress 操作。

使用藍圖

與其從頭開始建立管道定義，您可以使用組態藍圖，這些藍圖是預先設定的範本，用於常見的擷取案例，例如追蹤分析或 Apache 日誌。組態藍圖可協助您輕鬆佈建管道，而無需從頭開始撰寫組態。

主控台

使用管道藍圖

1. 登入 OpenSearch Ingestion 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。

2. 選擇 Create pipeline (建立管道)。
3. 從使用案例清單中選擇藍圖，然後選擇選取藍圖。管道組態會為您選取的使用案例填入子管道。

   管道藍圖無效。您需要根據選取的來源指定其他設定。

CLI

若要使用 取得所有可用藍圖的清單 AWS CLI，請傳送 list-pipeline-blueprints 請求。

```
aws osis list-pipeline-blueprints
```

請求會傳回所有可用藍圖的清單。

若要取得特定藍圖的詳細資訊，請使用 get-pipeline-blueprint 命令：

```
aws osis get-pipeline-blueprint --blueprint-name AWS-ApacheLogPipeline
```

此請求會傳回 Apache 日誌管道藍圖的內容：

```
{
   "Blueprint":{
      "PipelineConfigurationBody":"###\n  # Limitations: https://docs.aws.amazon.com/
opensearch-service/latest/ingestion/ingestion.html#ingestion-limitations\n###\n###\n
 # apache-log-pipeline:\n    # This pipeline receives logs via http (e.g. FluentBit),
 extracts important values from the logs by matching\n    # the value in the 'log' key
 against the grok common Apache log pattern. The grokked logs are then sent\n    # to
 OpenSearch to an index named 'logs'\n###\n\nversion: \"2\"\napache-log-pipeline:\n
 source:\n    http:\n      # Provide the path for ingestion. ${pipelineName} will be
 replaced with pipeline name configured for this pipeline.\n      # In this case it
 would be \"/apache-log-pipeline/logs\". This will be the FluentBit output URI value.
\n      path: \"/${pipelineName}/logs\"\n  processor:\n    - grok:\n        match:\n
         log: [ \"%{COMMONAPACHELOG_DATATYPED}\" ]\n  sink:\n    - opensearch:\n
   # Provide an AWS OpenSearch Service domain endpoint\n        # hosts: [ \"https://
search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-east-1.es.amazonaws.com\" ]\n        aws:
\n          # Provide the region of the domain.\n          # region: \"us-east-1\"\n
        # Enable the 'serverless' flag if the sink is an Amazon OpenSearch Serverless
 collection\n          # serverless: true\n        index: \"logs\"\n        # Enable
 the S3 DLQ to capture any failed requests in an S3 bucket\n        # dlq:\n
  # s3:\n            # Provide an S3 bucket\n            # bucket: \"your-dlq-bucket-
name\"\n            # Provide a key path prefix for the failed requests\n
  # key_path_prefix: \"${pipelineName}/logs/dlq\"\n            # Provide the region
 of the bucket.\n            # region: \"us-east-1\"\n            # Provide a Role
```

```
 ARN with access to the bucket. This role should have a trust relationship with osis-
pipelines.amazonaws.com\n"
      "BlueprintName":"AWS-ApacheLogPipeline"
   }
}
```

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 取得管道藍圖的相關資訊，請使用 ListPipelineBlueprints 和 GetPipelineBlueprint 操作。

檢視 Amazon OpenSearch 擷取管道

您可以使用 AWS 管理主控台、 AWS CLI或 OpenSearch Ingestion API 檢視 Amazon OpenSearch Ingestion 管道的詳細資訊。

主控台

檢視管道

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。
2. （選用）若要檢視具有特定狀態的管道，請選擇任何狀態，然後選取要篩選的狀態。

   管道可以有下列狀態：

   - `Active` – 管道處於作用中狀態，並準備好擷取資料。
   - `Creating` – 正在建立管道。
   - `Updating` – 管道正在更新。
   - `Deleting` – 正在刪除管道。
   - `Create failed` – 無法建立管道。
   - `Update failed` – 無法更新管道。
   - `Stop failed` – 管道無法停止。
   - `Start failed` – 無法啟動管道。
   - `Stopping` – 管道正在停止。
   - `Stopped` – 管道已停止，可隨時重新啟動。
   - `Starting` – 管道正在啟動。

當管道處於 Create failed、Deleting、 和 Stopped 狀態時Creating，您不需要支付擷取 OCUs 的費用。

CLI

若要使用 檢視管道 AWS CLI，請傳送清單管道請求：

```
aws osis list-pipelines
```

請求會傳回所有現有管道的清單：

```
{
    "NextToken": null,
    "Pipelines": [
        {,
            "CreatedAt": 1.671055851E9,
            "LastUpdatedAt": 1.671055851E9,
            "MaxUnits": 4,
            "MinUnits": 2,
            "PipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/log-pipeline",
            "PipelineName": "log-pipeline",
            "Status": "ACTIVE",
            "StatusReason": {
                "Description": "The pipeline is ready to ingest data."
            }
        },
            "CreatedAt": 1.671055851E9,
            "LastUpdatedAt": 1.671055851E9,
            "MaxUnits": 2,
            "MinUnits": 8,
            "PipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/another-
pipeline",
            "PipelineName": "another-pipeline",
            "Status": "CREATING",
            "StatusReason": {
                "Description": "The pipeline is being created. It is not able to ingest
 data."
            }
        }
    ]
}
```

若要取得單一管道的相關資訊，請使用 get-pipeline 命令：

```
aws osis get-pipeline --pipeline-name "my-pipeline"
```

請求會傳回指定管道的組態資訊：

```
{
    "Pipeline": {
        "PipelineName": "my-pipeline",
        "PipelineArn": "arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline",
        "MinUnits": 9,
        "MaxUnits": 10,
        "Status": "ACTIVE",
        "StatusReason": {
            "Description": "The pipeline is ready to ingest data."
        },
        "PipelineConfigurationBody": "log-pipeline:\n source:\n http:\n processor:\n
 - grok:\n match:\nlog: [ '%{COMMONAPACHELOG}' ]\n - date:\n from_time_received: true
\n destination: \"@timestamp\"\n  sink:\n - opensearch:\n hosts: [ \"https://search-
mdp-performance-test-duxkb4qnycd63rpy6svmvyvfpi.us-east-1.es.amazonaws.com\" ]\n index:
 \"apache_logs\"\n aws_sts_role_arn: \"arn:aws:iam::123456789012:role/my-domain-role
\"\n  aws_region: \"us-east-1\"\n  aws_sigv4: true",,
        "CreatedAt": "2022-10-01T15:28:05+00:00",
        "LastUpdatedAt": "2022-10-21T21:41:08+00:00",
        "IngestEndpointUrls": [
            "my-pipeline-123456789012.us-east-1.osis.amazonaws.com"
        ]
    }
}
```

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 檢視 OpenSearch Ingestion 管道，請呼叫 ListPipelines 和 GetPipeline 操作。

更新 Amazon OpenSearch 擷取管道

您可以使用 AWS 管理主控台、 AWS CLI或 OpenSearch Ingestion API 更新 Amazon OpenSearch Ingestion 管道。OpenSearch Ingestion 會啟動藍/綠部署。如需詳細資訊，請參閱the section called “管道更新的藍/綠部署”。

主題

- 考量事項

- 必要許可
- 更新管道
- 管道更新的藍/綠部署

考量事項

當您更新管道時，請考慮下列事項：

- 您無法更新管道的名稱或網路設定。
- 如果您的管道寫入 VPC 網域接收器，則在管道建立後，您無法返回並將接收器變更為不同的 VPC 網域。您必須使用新接收器刪除並重新建立管道。您仍然可以將接收器從 VPC 網域切換到公有網域、從公有網域切換到 VPC 網域，或從公有網域切換到另一個公有網域。
- 您可以隨時在公有 OpenSearch Service 網域和 OpenSearch Serverless 集合之間切換管道接收器。
- 當您更新管道的來源、處理器或接收器組態時，OpenSearch Ingestion 會啟動藍/綠部署。如需詳細資訊，請參閱the section called “管道更新的藍/綠部署”。
- 當您更新管道的來源、處理器或接收器組態時，OpenSearch Ingestion 會自動將您的管道升級至管道正在執行之主要 Data Prepper 版本的最新支援次要版本。此程序可讓您的管道隨時掌握最新的錯誤修正和效能改善。
- 您仍然可以在管道停止時更新管道。

必要許可

OpenSearch Ingestion 使用下列 IAM 許可來更新管道：

- `osis:UpdatePipeline` – 更新管道。
- `osis:ValidatePipeline` – 檢查管道組態是否有效。
- `iam:PassRole` – 將管道角色傳遞至 OpenSearch Ingestion，以便將資料寫入網域。只有在您更新管道組態時，才需要此許可，而不需要修改其他設定，例如日誌發佈或容量限制。

例如，下列政策會授予更新管道的許可：

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "osis:UpdatePipeline",
                "osis:ValidatePipeline"
            ]
        },
        {
            "Resource": [
                "arn:aws:iam::111122223333:role/pipeline-role"
            ],
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ]
        }
    ]
}
```

更新管道

您可以使用 AWS 管理主控台、 AWS CLI或 OpenSearch Ingestion API 更新 Amazon OpenSearch Ingestion 管道。

主控台

更新管道

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。
2. 選擇管道以開啟其設定。然後，選擇其中一個編輯選項。
3. 修改完成後，請選擇 Save (儲存)。

CLI

若要使用 更新管道 AWS CLI，請傳送更新管道請求。下列範例請求會上傳新的組態檔案，並更新容量值下限和上限：

```
aws osis update-pipeline \
  --pipeline-name "my-pipeline" \
  --pipline-configuration-body "file://new-pipeline-config.yaml" \
  --min-units 11 \
  --max-units 18
```

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 更新 OpenSearch Ingestion 管道，請呼叫 UpdatePipeline 操作。

管道更新的藍/綠部署

OpenSearch Ingestion 會啟動藍/綠部署程序。

藍/綠是指為管道更新建立新的環境，並在這些更新完成後將流量路由到新環境的做法。實務可在萬一部署到新環境不成功時將停機時間減至最小並維護原始環境。藍/綠部署本身沒有任何效能影響，但如果管道組態以改變效能的方式變更，效能可能會變更。

OpenSearch Ingestion 會在藍/綠部署期間封鎖自動擴展。您只會繼續支付舊管道的流量費用，直到流量重新導向至新管道為止。一旦流量重新導向，您只需支付新管道的費用。您永遠不會同時支付兩個管道的費用。

當您更新管道的來源、處理器或接收器組態時，OpenSearch Ingestion 會自動將您的管道升級至管道執行中主要版本的最新支援次要版本。例如，您可能在管道組態`version: "2"`中具有 ，而 OpenSearch Ingestion 最初使用 2.1.0 版佈建管道。新增對 2.1.1 版的支援，並變更管道組態時，OpenSearch Ingestion 會將管道升級至 2.1.1 版。

此程序可讓您的管道隨時掌握最新的錯誤修正和效能改善。OpenSearch Ingestion 無法更新管道的主要版本，除非您手動變更管道組態中的 `version`選項。

管理 Amazon OpenSearch Ingestion 管道成本

您可以在 Amazon OpenSearch Ingestion 中啟動和停止擷取管道，以根據您的需求控制資料流程。停止管道會在保留組態時停止資料處理，因此您可以重新啟動它，而無需重新設定它。這有助於最佳化成本、管理資源用量或疑難排解問題。當您停止管道時，OpenSearch Ingestion 不會處理傳入資料，但先前擷取的資料仍可在 OpenSearch 中使用。

啟動和停止可簡化管道的設定和縮減程序，讓您用於不需要持續可用性的開發、測試或類似活動。當您的管道停止時，您不需要支付任何擷取 OCU 時數的費用。您仍然可以更新已停止的管道，它們會收到自動次要版本更新和安全性修補程式。

停止和啟動管道將導致從頭開始重新處理提取型管道 (DDB、S3、DocDB 等） 的所有資料。當您停止管道時，管道建立的任何服務受管 VPC 端點都會遭到移除。對於具有自我管理 VPC 端點的管道，您必須在重新啟動管道時，在帳戶中重新建立 VPC 端點。如需詳細資訊，請參閱the section called “自我管理 VPC 端點”。

> **Note**
>
> 如果您的管道有多餘的容量，但需要保持運作，請考慮調整其最大容量限制，而不是停止和重新啟動它。這有助於管理成本，同時確保管道繼續有效率地處理資料。如需詳細資訊，請參閱the section called “擴展管道”。

下列主題說明如何使用 AWS 管理主控台 AWS CLI和 OpenSearch Ingestion API 來啟動和停止管道。

主題

- 停止 Amazon OpenSearch 擷取管道
- 啟動 Amazon OpenSearch 擷取管道

停止 Amazon OpenSearch 擷取管道

若要使用 OpenSearch Ingestion 管道或執行管理，請務必從作用中管道開始，然後停止管道，然後再次啟動管道。當您的管道停止時，您不需要支付擷取 OCU 時數的費用。

主控台

停止管道

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。
2. 選擇管道。您可以從此頁面執行停止操作，或導覽至您要停止之管道的詳細資訊頁面。
3. 針對動作，選擇停止管道。

   如果無法停止和啟動管道，則無法使用停止管道動作。

AWS CLI

若要使用 停止管道 AWS CLI，請使用下列參數呼叫 stop-pipeline 命令：

- `--pipeline-name` – 管道的名稱。

Example

```
aws osis stop-pipeline --pipeline-name my-pipeline
```

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 停止管道，請使用下列參數呼叫 StopPipeline 操作：

- `PipelineName` – 管道的名稱。

啟動 Amazon OpenSearch 擷取管道

您一律會從已處於停止狀態的管道開始啟動 OpenSearch Ingestion 管道。管道會保留其組態設定，例如容量限制、網路設定和日誌發佈選項。

重新啟動管道通常需要幾分鐘的時間。

主控台

啟動管道

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。
2. 選擇管道。您可以從此頁面執行啟動操作，或導覽至您要啟動之管道的詳細資訊頁面。
3. 針對動作，選擇啟動管道。

AWS CLI

若要使用 啟動管道 AWS CLI，請使用下列參數呼叫 start-pipeline 命令：

- `--pipeline-name` – 管道的名稱。

Example

```
aws osis start-pipeline --pipeline-name my-pipeline
```

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 啟動 OpenSearch Ingestion 管道，請使用下列參數呼叫 StartPipeline 操作：

- `PipelineName` – 管道的名稱。

刪除 Amazon OpenSearch 擷取管道

您可以使用 AWS 管理主控台、 AWS CLI或 OpenSearch Ingestion API 刪除 Amazon OpenSearch Ingestion 管道。當 狀態為 `Creating`或 時，您無法刪除管道`Updating`。

主控台

刪除管道

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。
2. 選取您要刪除的管道，然後選擇動作、刪除。
3. 確認刪除，然後選擇 Delete (刪除)。

CLI

若要使用 刪除管道 AWS CLI，請傳送刪除管道請求：

```
aws osis delete-pipeline --pipeline-name "my-pipeline"
```

OpenSearch 擷取 API

若要使用 OpenSearch Ingestion API 刪除 OpenSearch Ingestion 管道，請使用下列參數呼叫 DeletePipeline 操作：

- `PipelineName` – 管道的名稱。

Amazon OpenSearch Ingestion 管道支援的外掛程式和選項

Amazon OpenSearch Ingestion 支援開放原始碼 OpenSearch Data Prepper 中的來源、處理器和接收器子集。此外，OpenSearch Ingestion 對每個支援的外掛程式的可用選項有一些限制。下列各節說明 OpenSearch Ingestion 支援的外掛程式和相關選項。

> **Note**
>
> OpenSearch Ingestion 不支援任何緩衝外掛程式，因為它會自動設定預設緩衝區。如果您在管道組態中包含緩衝區，則會收到驗證錯誤。

主題

- 支援的外掛程式
- 無狀態與有狀態處理器
- 組態需求和限制條件

支援的外掛程式

OpenSearch Ingestion 支援下列 Data Prepper 外掛程式：

來源：

- DocumentDB
- DynamoDB

- HTTP
- Kafka
- Kinesis
- OpenSearch
- OTel 日誌
- OTel 指標
- OTel 追蹤
- S3

處理器：

- 新增項目
- Aggregate
- 異常偵測器
- AWS Lambda
- 轉換項目類型
- 複製值
- CSV
- 日期
- 延遲
- 解壓縮
- 刪除項目
- 剖析
- 捨棄事件
- 壓平
- 地理 IP
- Grok
- 索引鍵值
- 要映射的清單
- 小寫字串
- 映射至清單
- 變動事件（處理器系列）
- Mutate 字串（處理器系列）
- 混淆
- OTel 指標
- OTel 追蹤群組
- OTel 追蹤
- 剖析 Ion
- 剖析 JSON
- 剖析 XML

- 重新命名金鑰
- 選取項目
- 服務地圖
- 分割事件
- 分割字串
- 字串轉換器
- 替代字串
- 追蹤對等轉送器
- Translate (轉譯)
- Trim 字串
- 截斷
- 大寫字串
- 使用者代理程式
- 寫入 JSON

接收器：

- OpenSearch （支援 OpenSearch Service、OpenSearch Serverless 和 Elasticsearch 6.8 或更新版本）
- S3

接收器轉碼器：

- Avro
- NDJSON
- JSON
- Parquet

無狀態與有狀態處理器

無狀態處理器會執行轉換和篩選等操作，而有狀態處理器則會執行彙總等操作，以記住先前執行的結果。OpenSearch Ingestion 支援具狀態處理器彙總和服務映射。所有其他支援的處理器都是無狀態處理器。

對於僅包含無狀態處理器的管道，最大容量限制為 96 個擷取 OCUs。如果管道包含任何具狀態處理器，則最大容量限制為 48 個擷取 OCUs。不過，如果管道已啟用持久性緩衝，則最多可以有 384 個僅具有無狀態處理器的擷取 OCUs，如果包含任何有狀態處理器，則可以有 192 個擷取 OCUs。如需詳細資訊，請參閱the section called “擴展管道”。

只有無狀態處理器才支援End-to-end確認。如需詳細資訊，請參閱the section called “End-to-end認可”。

組態需求和限制條件

除非以下另有說明，否則在 OpenSearch Ingestion 管道中允許上述支援外掛程式的 Data Prepper 組態參考中所述的所有選項。下列各節說明 OpenSearch Ingestion 對特定外掛程式選項的限制。

> **Note**
>
> OpenSearch Ingestion 不支援任何緩衝外掛程式，因為它會自動設定預設緩衝區。如果您在管道組態中包含緩衝區，則會收到驗證錯誤。

許多選項是由 OpenSearch Ingestion 在內部設定和管理，例如 `authentication`和 `acm_certificate_arn`。如果手動變更`request_timeout`， `thread_count`和 等其他選項會影響效能。因此，這些值會在內部設定，以確保管道的最佳效能。

最後，某些選項無法傳遞至 OpenSearch Ingestion，例如 `ism_policy_file`和 `sink_template`，因為在開放原始碼 Data Prepper 中執行時，它們是本機檔案。不支援這些值。

主題

- 一般管道選項
- Grok 處理器
- HTTP 來源
- OpenSearch 接收器
- OTel 指標來源、OTel 追蹤來源和 OTel 日誌來源
- OTel 追蹤群組處理器
- OTel 追蹤處理器
- 服務映射處理器
- S3 來源

一般管道選項

下列一般管道選項是由 OpenSearch Ingestion 設定，管道組態不支援：

- `workers`
- `delay`

Grok 處理器

不支援下列 Grok 處理器選項：

- `patterns_directories`
- `patterns_files_glob`

HTTP 來源

HTTP 來源外掛程式有下列需求和限制：

- 選項為必要`path`項目。路徑是字串，例如 `/log/ingest`，代表日誌擷取的 URI 路徑。此路徑會定義您用來將資料傳送至管道的 URI。例如 `https://log-pipeline.us-west-2.osis.amazonaws.com/log/ingest`。路徑必須以斜線 (/) 開頭，可包含特殊字元 '-'、'_'、'.' 和 '/'，以及`${pipelineName}`預留位置。
- 下列 HTTP 來源選項是由 OpenSearch Ingestion 設定，管道組態不支援：
  - `port`
  - `ssl`
  - `ssl_key_file`
  - `ssl_certificate_file`
  - `aws_region`
  - `authentication`
  - `unauthenticated_health_check`
  - `use_acm_certificate_for_ssl`
  - `thread_count`
  - `request_timeout`
  - `max_connection_count`
  - `max_pending_requests`

- health_check_service
- acm_private_key_password
- acm_certificate_timeout_millis
- acm_certificate_arn

OpenSearch 接收器

OpenSearch 接收器外掛程式有下列需求和限制。

- aws 選項為必要，且必須包含下列選項：
  - sts_role_arn
  - region
  - hosts
  - serverless （如果接收器是 OpenSearch Serverless 集合）
- sts_role_arn 選項必須指向 YAML 定義檔案中每個接收器的相同角色。
- hosts 選項必須指定 OpenSearch Service 網域端點或 OpenSearch Serverless 集合端點。您無法指定網域的自訂端點；它必須是標準端點。
- 如果 hosts選項是無伺服器集合端點，您必須將 serverless選項設定為 true。此外，如果您的 YAML 定義檔案包含 index_type選項，則必須將其設定為 management_disabled，否則驗證會失敗。
- 不支援下列選項：
  - username
  - password
  - cert
  - proxy
  - dlq_file - 如果您想要將失敗的事件卸載至無效字母佇列 (DLQ)，您必須使用 dlq選項並指定 S3 儲存貯體。
  - ism_policy_file
  - socket_timeout
  - template_file
  - insecure

OTel 指標來源、OTel 追蹤來源和 OTel 日誌來源

OTel 指標來源、OTel 追蹤來源和 OTel 日誌來源外掛程式具有下列需求和限制：

- path 選項為必要項目。路徑是字串，例如 /log/ingest，代表日誌擷取的 URI 路徑。此路徑會定義您用來將資料傳送至管道的 URI。例如 https://log-pipeline.us-west-2.osis.amazonaws.com*/log/ingest*。路徑必須以斜線 (/) 開頭，可包含特殊字元 '-'、'_'、'.' 和 '/'，以及${pipelineName}預留位置。
- 下列選項是由 OpenSearch Ingestion 設定，管道組態不支援：
  - port
  - ssl
  - sslKeyFile
  - sslKeyCertChainFile
  - authentication
  - unauthenticated_health_check
  - useAcmCertForSSL
  - unframed_requests
  - proto_reflection_service
  - thread_count
  - request_timeout
  - max_connection_count
  - acmPrivateKeyPassword
  - acmCertIssueTimeOutMillis
  - health_check_service
  - acmCertificateArn
  - awsRegion

OTel 追蹤群組處理器

OTel 追蹤群組處理器有下列需求和限制：

- aws 選項為必要，且必須包含下列選項：
  - sts_role_arn

  - region
  - hosts
- sts_role_arn 選項指定與您在 OpenSearch 接收器組態中指定的管道角色相同的角色。
- 不支援 username、cert、 password和 insecure選項。
- aws_sigv4 選項為必要項目，且必須設定為 true。
- 不支援 OpenSearch 接收器外掛程式中的 serverless選項。Otel 追蹤群組處理器目前不適用於 OpenSearch Serverless 集合。
- 管道組態內文中的otel_trace_group處理器數量不能超過 8。

OTel 追蹤處理器

OTel 追蹤處理器有下列需求和限制：

- trace_flush_interval 選項的值不能超過 300 秒。

服務映射處理器

Service-map 處理器有下列需求和限制：

- window_duration 選項的值不能超過 300 秒。

S3 來源

S3 來源外掛程式有下列需求和限制：

- aws 選項為必要，且必須包含 region 和 sts_role_arn選項。
- records_to_accumulate 選項的值不能超過 200。
- maximum_messages 選項的值不能超過 10。
- 如果指定，則 disable_bucket_ownership_validation選項必須設定為 false。
- 如果指定，則必須將 input_serialization選項設定為 parquet。

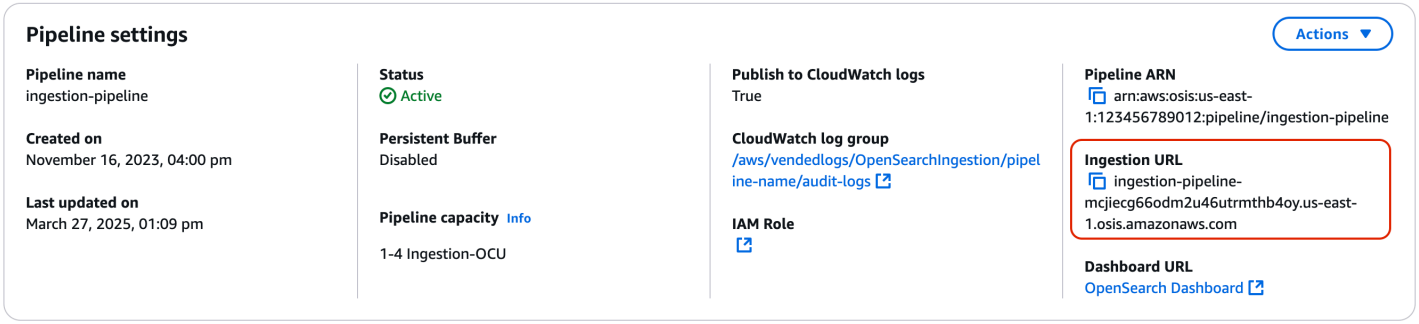
將 Amazon OpenSearch Ingestion 管道與其他 服務和應用程式整合

若要成功將資料擷取至 Amazon OpenSearch Ingestion 管道，您必須設定用戶端應用程式 (來源) 將資料傳送至管道端點。您的來源可能是 Fluent Bit 日誌、OpenTelemetry Collector 或簡單 S3 儲存貯體等用戶端。每個用戶端的確切組態會有所不同。

來源組態期間的重要差異 （ 相較於直接將資料傳送至 OpenSearch Service 網域或 OpenSearch Serverless 集合 ） 是 AWS 服務名稱 (`osis`) 和主機端點，其必須是管道端點。

建構擷取端點

若要將資料擷取至管道，請將其傳送至擷取端點。若要尋找擷取 URL，請導覽至管道設定頁面並複製擷取 URL。



若要建構提取型來源的完整擷取端點，例如 OTel 追蹤和 OTel 指標，請將管道組態的擷取路徑新增至擷取 URL。

例如，假設您的管道組態具有下列擷取路徑：

您在用戶端組態中指定的完整擷取端點將採用下列格式：`https://`*`ingestion-pipeline-abcdefg`*`.us-east-1.osis.amazonaws.com`*`/my/test_path`*。

建立擷取角色

對 OpenSearch Ingestion 的所有請求都必須使用 Signature 第 4 版簽署。至少，必須授予簽署請求的角色 `osis:Ingest`動作的許可，以允許它將資料傳送至 OpenSearch Ingestion 管道。

例如，下列 AWS Identity and Access Management (IAM) 政策允許對應的角色將資料傳送至單一管道：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:us-east-1:111122223333:pipeline/pipeline-name"
    }
  ]
}
```

> **Note**
>
> 若要將 角色用於所有管道，請以萬用字元 (*) 取代 `Resource`元素中的 ARN。

提供跨帳戶擷取存取權

> **Note**
>
> 您只能為公有管道提供跨帳戶擷取存取，而不是 VPC 管道。

您可能需要從不同的管道擷取資料 AWS 帳戶，例如存放來源應用程式的 帳戶。如果寫入管道的委託人位於與管道本身不同的帳戶中，您需要設定委託人以信任另一個 IAM 角色將資料擷取至管道。

設定跨帳戶擷取許可

1. 在 AWS 帳戶 與管道相同的 內建立具有`osis:Ingest`許可 （ 如上一節所述 ） 的擷取角色。如需說明，請參閱建立 IAM 角色。
2. 將信任政策連接至 擷取角色，允許另一個帳戶中的委託人擔任該角色：

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [{
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::111122223333:root"
         },
        "Action": "sts:AssumeRole"
     }]
   }
   ```

3. 在另一個帳戶中，設定您的用戶端應用程式 （ 例如 Fluent Bit) 以擔任擷取角色。為了讓此項目正常運作，應用程式帳戶必須將許可授予應用程式使用者或角色，以擔任擷取角色。

   下列以身分為基礎的政策範例允許連接的委託人`ingestion-role`從管道帳戶擔任：

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": "sts:AssumeRole",
         "Resource": "arn:aws:iam::111122223333:role/ingestion-role"
       }
     ]
   }
   ```

然後，用戶端應用程式可以使用 AssumeRole 操作來假設`ingestion-role`並將資料擷取至相關聯的管道。

搭配 Atlassian 服務使用 OpenSearch Ingestion 管道

您可以使用 Atlassian Jira 和 Confluence 來源外掛程式，將資料從 Atlassian 服務擷取到您的 OpenSearch Ingestion 管道。這些整合可讓您透過同步完整的 Jira 專案和 Confluence 空間來建立統一的可搜尋知識庫，同時透過持續監控和自動同步更新來維持即時相關性。

Integrating with Jira

將您的 Jira 內容整合到 OpenSearch，以強大的內容搜尋功能轉換您的 Jira 體驗。Data Prepper Atlassian Jira 來源外掛程式可讓您透過同步完整的 Jira 專案來建立統一的可搜尋知識庫，同時透過持續監控和自動同步更新來維持即時相關性。此整合可讓特定專案、問題類型和狀態的彈性篩選選項進行資料同步，確保僅匯入您需要的資訊。

為了確保安全可靠的連線， 外掛程式支援多種身分驗證方法，包括基本 API 金鑰身分驗證和 OAuth2 身分驗證，並增加了使用存放於 中的秘密管理憑證的安全性 AWS Secrets Manager。它還具有自動權杖續約功能，實現不間斷的存取，確保持續操作。此整合以 Atlassian 的 API 第 2 版為基礎，可讓團隊透過 OpenSearch 的進階搜尋功能，從其 Jira 資料釋放寶貴的洞見。

Integrating with Confluence

透過 Data Prepper 的 Confluence 來源外掛程式，將 Atlassian Confluence 內容整合至 OpenSearch，以增強您團隊的知識管理和協同合作功能。此整合可讓您建立集中且可搜尋的集體知識儲存庫，進而改善資訊探索和團隊生產力。透過同步 Confluence 內容並持續監控更新，外掛程式可確保 OpenSearch 索引保持up-to-date且全面。

整合提供靈活的篩選選項，可讓您選擇性地從特定空間或頁面類型匯入內容，根據組織的需求量身打造同步內容。外掛程式同時支援基本 API 金鑰和 OAuth2 身分驗證方法，並可選擇透過 安全地管理登入資料 AWS Secrets Manager。外掛程式的自動權杖續約功能可確保不間斷的存取和無縫的操作。此整合以 Atlassian 的 Confluence API 為基礎，可讓團隊在其 Confluence 內容中利用 OpenSearch 的進階搜尋功能，進而增強組織內的資訊可存取性和使用率。

主題

- 先決條件
- 設定管道角色
- Jira 連接器管道組態
- Confluence 連接器管道組態
- 資料一致性

- 限制
- 適用於 Atlassian 連接器的 CloudWatch 指標
- 使用 OAuth 2.0 將 Amazon OpenSearch Ingestion 管道連線到 Atlassian Jira 或 Confluence

先決條件

建立 OpenSearch Ingestion 管道之前，請完成下列步驟：

1. 選擇下列其中一個選項，為您的 Jira 網站準備登入資料。OpenSearch Ingestion 只需要對內容`ReadOnly`的授權。
   a. 選項 1：API 金鑰 – 登入您的 Atlassian 帳戶，並使用下列主題中的資訊來產生您的 API 金鑰：
      - 管理 Atlassian 帳戶的 API 字符
   b. 選項 2：OAuth2 – 登入您的 Atlassian 帳戶，並使用 中的資訊the section called “使用 OAuth 2.0 將 Amazon OpenSearch Ingestion 管道連線到 Atlassian Jira 或 Confluence”。
2. 在 中建立秘密 AWS Secrets Manager，以存放上一個步驟中建立的登入資料。遵循程序進行下列選擇：
   - 針對機密類型，選擇其他類型的機密。
   - 對於鍵/值對，根據您選取的授權類型建立下列對：

   API key

   ```
   {
      "username": user-name-usualy-email-id,
      "password": api-key
   }
   ```

   OAuth 2.0

   ```
   {
      "clientId": client-id
      "clientSecret": client-secret
      "accessKey": access-key
      "refreshKey": refresh-key
   }
   ```

   建立秘密之後，請複製秘密的 Amazon Resource Name (ARN)。您會將其包含在管道角色許可政策中。

設定管道角色

在管道中傳遞的角色必須附加下列政策，才能讀取和寫入先決條件區段中建立的秘密。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SecretReadWrite",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:PutSecretValue",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:secret-name-random-6-characters"
        }
    ]
}
```

此角色也應該連接政策，以存取和寫入您選擇的接收器。例如，如果您選擇 OpenSearch 做為目的地，政策看起來會類似以下內容：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "OpenSearchWritePolicy",
            "Effect": "Allow",
            "Action": "aoss:*",
            "Resource": "arn:aws:aoss:us-
east-1:111122223333:collection/collection-id"
        }
```

```
    ]
}
```

Jira 連接器管道組態

您可以使用預先設定的 Atlassian Jira 藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

使用您的資訊取代*######*。

```
version: "2"
extension:
  aws:
    secrets:
      jira-account-credentials:
        secret_id: "secret-arn"
        region: "secret-region"
        sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
atlassian-jira-pipeline:
  source:
    jira:
      # We only support one host url for now
      hosts: ["jira-host-url"]
      acknowledgments: true
      authentication:
        # Provide one of the authentication method to use. Supported methods are
 'basic' and 'oauth2'.
        # For basic authentication, password is the API key that you generate using
 your jira account
        basic:
          username: ${{aws_secrets:jira-account-credentials:username}}
          password: ${{aws_secrets:jira-account-credentials:password}}
        # For OAuth2 based authentication, we require the following 4 key values stored
 in the secret
        # Follow atlassian instructions at the below link to generate these keys.
        # https://developer.atlassian.com/cloud/confluence/oauth-2-3lo-apps/
        # If you are using OAuth2 authentication, we also require, write permission to
 your AWS secret to
        # be able to write the renewed tokens back into the secret.
        # oauth2:
          # client_id: ${{aws_secrets:jira-account-credentials:clientId}}
          # client_secret: ${{aws_secrets:jira-account-credentials:clientSecret}}
```

```
          # access_token: ${{aws_secrets:jira-account-credentials:accessToken}}
          # refresh_token: ${{aws_secrets:jira-account-credentials:refreshToken}}
      filter:
        project:
          key:
            include:
              # This is not project name.
              # It is an alphanumeric project key that you can find under project
 details in Jira.
              - "project-key"
              - "project-key"
            # exclude:
              # - "project-key"
              # - "project-key"
        issue_type:
          include:
            - "issue-type"
            # - "Story"
            # - "Bug"
            # - "Task"
         # exclude:
             # - "Epic"
        status:
          include:
            - "ticket-status"
            # - "To Do"
            # - "In Progress"
            # - "Done"
         # exclude:
           # - "Backlog"

  sink:
    - opensearch:
        # Provide an Amazon OpenSearch Service domain endpoint
        hosts: [ "https://search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-
east-1.es.amazonaws.com" ]
        index: "index_${getMetadata(\"project\")}"
        # Ensure adding unique document id which is the unique ticket id in this case
        document_id: '${/id}'
        aws:
          # Provide a Role ARN with access to the domain. This role should have a trust
 relationship with osis-pipelines.amazonaws.com
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          # Provide the region of the domain.
```

```
          region: "us-east-1"
          # Enable the 'serverless' flag if the sink is an Amazon OpenSearch Serverless
collection
          serverless: false
          # serverless_options:
            # Specify a name here to create or update network policy for the serverless
collection
            # network_policy_name: "network-policy-name"
       # Enable the 'distribution_version' setting if the Amazon OpenSearch Service
domain is of version Elasticsearch 6.x
       # distribution_version: "es6"
       # Enable and switch the 'enable_request_compression' flag if the default
compression setting is changed in the domain.
       # See the section called "## HTTP ##"
       # enable_request_compression: true/false
       # Optional: Enable the S3 DLQ to capture any failed requests in an S3 bucket.
Delete this entire block if you don't want a DLQ.
       dlq:
         s3:
           # Provide an S3 bucket
           bucket: "your-dlq-bucket-name"
           # Provide a key path prefix for the failed requests
           # key_path_prefix: "kinesis-pipeline/logs/dlq"
           # Provide the region of the bucket.
           region: "us-east-1"
           # Provide a Role ARN with access to the bucket. This role should have a
trust relationship with osis-pipelines.amazonaws.com
           sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
```

Jira 來源中屬性的索引鍵：

1. 主機：您的 Jira 雲端或內部部署 URL。一般而言，它看起來像 `https://your-domain-name.atlassian.net/`。
2. 確認：保證將資料交付至目的地。
3. 身分驗證：描述您希望管道如何存取 Jira 執行個體。選擇 `Basic`或 ，`OAuth2`並指定參考 AWS 秘密中金鑰的對應金鑰屬性。
4. 篩選條件：本節可協助您選取要擷取和同步的 Jira 資料部分。
   a. 專案：在 `include`區段中列出您要同步的專案金鑰。否則，請在 `exclude`區段下列出您要排除的專案。在任何指定時間僅提供其中一個包含或排除選項。

b. issue_type：您要同步的特定問題類型。遵循符合您需求的類似 include或 exclude 模式。請注意，附件會顯示為原始附件的錨點連結，但不會擷取附件內容。

c. 狀態：您想要套用至資料擷取查詢的特定狀態篩選條件。如果您指定 include，則只會同步具有這些狀態的票證。如果您指定 exclude，則會同步列出排除狀態的所有票證以外的所有票證。

Confluence 連接器管道組態

您可以使用預先設定的 Atlassian Confluence 藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

```
version: "2"
extension:
  aws:
    secrets:
      confluence-account-credentials:
        secret_id: "secret-arn"
        region: "secret-region"
        sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
atlassian-confluence-pipeline:
  source:
    confluence:
      # We currently support only one host URL.
      hosts: ["confluence-host-url"]
      acknowledgments: true
      authentication:
        # Provide one of the authentication method to use. Supported methods are
 'basic' and 'oauth2'.
        # For basic authentication, password is the API key that you generate using
 your Confluence account
        basic:
          username: ${{aws_secrets:confluence-account-credentials:confluenceId}}
          password: ${{aws_secrets:confluence-account-
credentials:confluenceCredential}}
        # For OAuth2 based authentication, we require the following 4 key values stored
 in the secret
        # Follow atlassian instructions at the following link to generate these keys:
        # https://developer.atlassian.com/cloud/confluence/oauth-2-3lo-apps/
        # If you are using OAuth2 authentication, we also require write permission to
 your AWS secret to
        # be able to write the renewed tokens back into the secret.
        # oauth2:
```

```
          # client_id: ${{aws_secrets:confluence-account-credentials:clientId}}
          # client_secret: ${{aws_secrets:confluence-account-credentials:clientSecret}}
          # access_token: ${{aws_secrets:confluence-account-credentials:accessToken}}
          # refresh_token: ${{aws_secrets:confluence-account-credentials:refreshToken}}
      filter:
        space:
          key:
            include:
              # This is not space name.
              # It is a space key that you can find under space details in Confluence.
              - "space key"
              - "space key"
           # exclude:
             #  - "space key"
             #  - "space key"
        page_type:
          include:
            - "content type"
            # - "page"
            # - "blogpost"
            # - "comment"
         # exclude:
            # - "attachment"

  sink:
    - opensearch:
        # Provide an Amazon OpenSearch Service domain endpoint
        hosts: [ "https://search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-
east-1.es.amazonaws.com" ]
         index: "index_${getMetadata(\"space\")}"
        # Ensure adding unique document id which is the unique ticket ID in this case.
        document_id: '${/id}'
        aws:
          # Provide the Amazon Resource Name (ARN) for a role with access to the
 domain. This role should have a trust relationship with osis-pipelines.amazonaws.com.
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          # Provide the Region of the domain.
          region: "us-east-1"
          # Enable the 'serverless' flag if the sink is an Amazon OpenSearch Serverless
 collection
          serverless: false
          # serverless_options:
            # Specify a name here to create or update network policy for the serverless
 collection.
```

```
            # network_policy_name: "network-policy-name"
        # Enable the 'distribution_version' setting if the Amazon OpenSearch Service
domain is of version Elasticsearch 6.x
        # distribution_version: "es6"
        # Enable and switch the 'enable_request_compression' flag if the default
compression setting is changed in the domain.
        # For more information, see the section called “## HTTP ##”.
        # enable_request_compression: true/false
        # Optional: Enable the S3 DLQ to capture any failed requests in an S3 bucket.
Delete this entire block if you don't want a DLQ.
        dlq:
          s3:
            # Provide an S3 bucket
            bucket: "your-dlq-bucket-name"
            # Provide a key path prefix for the failed requests
            # key_path_prefix: "kinesis-pipeline/logs/dlq"
            # Provide the Rregion of the bucket.
            region: "us-east-1"
            # Provide the Amazon Resource Name (ARN) for a role with access to the
bucket. This role should have a trust relationship with osis-pipelines.amazonaws.com
            sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
```

Confluence 來源中的關鍵屬性：

1. 主機：您的 Confluence 雲端或內部部署 URL。一般而言，它看起來像 `https://your-domain-name.atlassian.net/`
2. 確認：保證將資料交付至目的地。
3. 身分驗證：描述您希望管道如何存取 Confluence 執行個體。選擇 `Basic`或 ，`OAuth2`並指定參考 AWS 秘密中金鑰的對應金鑰屬性。
4. 篩選條件：本節可協助您選取要擷取和同步 Confluence 資料的哪個部分。
   a. space：列出您要在 `include`區段中同步的空間鍵。否則，請在 `exclude`區段下列出您要排除的空格。在任何指定時間僅提供其中一個包含或排除選項。
   b. page_type：您想要同步的特定頁面類型（例如頁面、部落格文章或附件）。遵循符合您需求的類似 `include`或 `exclude` 模式。請注意，附件會顯示為原始附件的錨點連結，但不會擷取附件內容。

資料一致性

根據管道 YAML 中指定的篩選條件，選取的專案 （或空格） 將擷取一次，並完全同步到目標目的地。然後，持續變更監控會在變更發生時擷取變更，並更新目的地中的資料。其中一個例外是變更監控只會同步 `create`和 `update`動作，而不是 `delete`動作。

限制

- 使用者刪除動作不會同步。記錄在接收器中的資料會保留在接收器中。如果在接收器設定中指定 ID 映射，更新會以新的變更覆寫現有內容。
- 使用舊版 Atlassian 軟體不支援下列 APIs 的現場部署執行個體與此來源不相容：
  - Jira Search API 第 3 版
    - `rest/api/3/search`
    - `rest/api/3/issue`
  - Confluence
    - `wiki/rest/api/content/search`
    - `wiki/rest/api/content`
    - `wiki/rest/api/settings/systemInfo`

適用於 Atlassian 連接器的 CloudWatch 指標

類型：Jira 連接器指標

來源	指標	指標類型
acknowledgementSetSuccesses.count	計數器	如果啟用確認，此指標會提供已成功同步的票證數量。
acknowledgementSetFailures.count	計數器	如果啟用確認，此指標會提供無法同步的票證數量。

來源	指標	指標類型
crawlingTime.avg	Timer	爬取所有新變更所需的時間。
ticketFetchLatency.avg	Timer	票證擷取 API 延遲平均值。
ticketFetchLatency.max	Timer	票證擷取 API 延遲上限。
ticketsRequested.count	計數器	提出的票證擷取請求數量。
ticketRequestedFailed.count	計數器	票證擷取請求數目失敗。
ticketRequestedSuccess.count	計數器	票證擷取請求成功的數量。
searchCallLatency.avg	Timer	搜尋 API 呼叫延遲平均值。
searchCallLatency.max	Timer	搜尋 API 呼叫延遲上限。
searchResultsFound.count	計數器	在指定的搜尋呼叫中找到的項目數量。

來源	指標	指標類型
searchRequestFailed.count	計數器	搜尋 API 呼叫失敗計數。
authFailures.count	計數器	身分驗證失敗計數。

類型：Confluence 連接器指標

來源	指標	指標類型
acknowledgementSetSuccesses.count	計數器	如果啟用確認，此指標會提供已成功同步的頁面數。
acknowledgementSetFailures.count	計數器	如果啟用確認，此指標會提供無法同步的頁面數。
crawlingTime.avg	Timer	爬取所有新變更所需的時間。
pageFetchLatency.avg	Timer	內容擷取 API 延遲（平均值）。
pageFetchLatency.max	Timer	內容擷取 API 延遲（上限）。
pagesRequested.count	計數器	擷取 API 的內容叫用次數。

來源	指標	指標類型
pageRequestFailed.count	計數器	內容擷取 API 的失敗請求數量。
pageRequestedSuccess.count	計數器	內容擷取 API 的成功請求數量。
searchCallLatency.avg	Timer	搜尋 API 呼叫延遲平均值。
searchCallLatency.max	Timer	搜尋 API 呼叫延遲上限。
searchResultsFound.count	計數器	在指定的搜尋呼叫中找到的項目數量。
searchRequestsFailed.count	計數器	搜尋 API 呼叫失敗計數。
authFailures.count	計數器	身分驗證失敗計數。

使用 OAuth 2.0 將 Amazon OpenSearch Ingestion 管道連線到 Atlassian Jira 或 Confluence

使用本主題中的資訊來協助您設定 Amazon OpenSearch Ingestion 管道，並使用 OAuth 2.0 身分驗證連線至 Jira 或 Confluence 帳戶。當 正在透過 Atlassian Services 完成使用 OpenSearch Ingestion 管道the section called “先決條件”的 ，但選擇不使用 API 金鑰登入資料時，請執行此任務。

主題

- 建立 OAuth 2.0 整合應用程式

- 產生和重新整理 Atlassian Developer 存取權杖

建立 OAuth 2.0 整合應用程式

使用下列程序協助您在 Atlassian 開發人員網站上建立 OAuth 2.0 整合應用程式。

建立 OAuth 2.0 整合應用程式

1. 在 https：//https://developer.atlassian.com/console/myapps/ 登入您的 Atlassian Developer 帳戶。
2. 選擇建立、OAuth 2.0 整合。
3. 針對名稱，輸入名稱以識別應用程式的目的。
4. 選取我同意受 Atlassian 開發人員條款約束核取方塊，然後選擇建立。
5. 在左側導覽中，選擇授權，然後選擇新增。
6. 針對回呼 URL，輸入任何 URL，例如 **`https://www.amazon.com`**或 **`https://www.example.com`**，然後選擇儲存變更。
7. 在左側導覽中，選擇許可頁面，然後在 Jira API 的資料列中，選擇新增，然後選擇設定。然後選取所有傳統範圍讀取許可 （ 下方提供的清單），然後選取儲存
8. 選擇精細範圍索引標籤，然後選擇編輯範圍以開啟編輯 Jira API 對話方塊。
9. 選取您正在使用的來源外掛程式許可：

   Jira

```
read:audit-log:jira
read:issue:jira
read:issue-meta:jira
read:attachment:jira
read:comment:jira
read:comment.property:jira
read:field:jira
read:field.default-value:jira
read:field.option:jira
read:field-configuration-scheme:jira
read:field-configuration:jira
read:issue-link:jira
read:issue-link-type:jira
read:issue-link-type:jira
read:issue.remote-link:jira
```

```
read:issue.property:jira
read:resolution:jira
read:issue-details:jira
read:issue-type:jira
read:issue-worklog:jira
read:issue-field-values:jira
read:issue.changelog:jira
read:issue.transition:jira
read:issue.vote:jira
read:jira-expressions:jira
```

Confluence

```
read:content:confluence
read:content-details:confluence
read:space-details:confluence
read:audit-log:confluence
read:page:confluence
read:blogpost:confluence
read:custom-content:confluence
read:comment:confluence
read:space:confluence
read:space.property:confluence
read:space.setting:confluence
read:content.property:confluence
read:content.metadata:confluence
read:task:confluence
read:whiteboard:confluence
read:app-data:confluence
manage:confluence-configuration
```

10. 選擇儲存。

如需相關資訊，請參閱 Atlassian Developer 網站上的實作 OAuth 2.0 (3LO) 和判斷 操作所需的範圍。

產生和重新整理 Atlassian Developer 存取權杖

使用下列程序，協助您在 Atlassian Developer 網站上產生和重新整理 Atlassian Developer 存取字符。

產生和重新整理 Jira 存取權杖

1. 在 https：//https://developer.atlassian.com/console/myapps/ 登入您的 Atlassian Developer 帳戶。

2. 選擇您在 中建立的應用程式the section called “建立 OAuth 2.0 整合應用程式”。
3. 在左側導覽中，選擇授權。
4. 從頁面底部複製精細的 Atlassian API 授權 URL 值，並將其貼到您選擇的文字編輯器。

   URL 的格式如下：

   ```
   https://auth.atlassian.com/authorize?
   audience=api.atlassian.com
   &client_id=YOUR_CLIENT_ID
   &scope=REQUESTED_SCOPE%20REQUESTED_SCOPE_TWO
   &redirect_uri=https://YOUR_APP_CALLBACK_URL
   &state=YOUR_USER_BOUND_VALUE
   &response_type=code
   &prompt=consent
   ```

5. 對於 `state=YOUR_USER_BOUND_VALUE`，將參數值變更為您選擇的任何項目，例如 state="**`sample_text`**"。

   如需詳細資訊，請參閱 Atlassian Developer 網站上的什麼是 使用的狀態參數？。

6. 請注意， `scope`區段列出您在先前任務中選取的精細範圍。例如：`scope=read%3Ajira-work%20read%3Ajira-user%20offline_access`

   `offline_access` 表示您想要產生 `refresh_token`。

7. 開啟 Web 瀏覽器視窗，並輸入您複製到瀏覽器視窗地址列的授權 URL。
8. 當目標頁面開啟時，請確認資訊正確無誤，然後選擇接受以重新導向至您的 Jira 或 Confluence 首頁。
9. 載入首頁後，複製此頁面的 URL。它包含應用程式的授權碼。您可以使用此程式碼來產生存取權杖。之後的整個區段`code=`是授權碼。
10. 使用下列 cURL 命令來產生存取權杖。使用您的資訊取代`######`。

    > **Tip**
    >
    > 您也可以使用第三方服務，例如 Postman。

    ```
    curl --request POST --url 'https://auth.atlassian.com/oauth/token' \
    --header 'Content-Type: application/json' \
    --data '{"grant_type": "authorization_code",
    ```

```
"client_id": "YOUR_CLIENT_ID",
"client_secret": "YOUR_CLIENT_SECRET",
"code": "AUTHORIZATION_CODE",
"redirect_uri": "YOUR_CALLBACK_URL"}'
```

此命令的回應包含 `access_code`和 的值`refresh_token`。

搭配 Amazon Aurora 使用 OpenSearch Ingestion 管道

您可以使用 OpenSearch Ingestion 管道搭配 Amazon Aurora，將現有的資料和串流變更（例如建立、更新和刪除）匯出至 Amazon OpenSearch Service 網域和集合。OpenSearch Ingestion 管道整合了變更資料擷取 (CDC) 基礎設施，以提供從 Amazon Aurora 持續串流資料的高規模、低延遲方式。支援 Aurora MySQL 和 Aurora PostgreSQL。

有兩種方式可以使用 Amazon Aurora 作為來源來處理資料，無論是否有完整的初始快照。完整初始快照是指定資料表的快照，此快照會匯出至 Amazon S3。從那裡，OpenSearch Ingestion 管道會將其傳送至網域中的一個索引，或將其分割至網域中的多個索引。為了讓 Amazon Aurora 和 OpenSearch 中的資料保持一致，管道會將 Amazon Aurora 叢集中資料表中的所有建立、更新和刪除事件與儲存在 OpenSearch 索引或索引中的文件同步。

當您使用完整初始快照時，您的 OpenSearch 擷取管道會先擷取快照，然後開始從 Amazon Aurora 變更串流讀取資料。它最終會追上並維持 Amazon Aurora 和 OpenSearch 之間的近乎即時的資料一致性。

您也可以使用 OpenSearch Ingestion 與 Amazon Aurora 整合來追蹤變更資料擷取，並將 Aurora 中的所有更新擷取至 OpenSearch。如果您已經擁有其他機制的完整快照，或者只想要擷取 Amazon Aurora 叢集中資料的所有變更，請選擇此選項。

選擇此選項時，您需要為 Aurora MySQL 設定二進位記錄，或在叢集上為 Aurora PostgreSQL 設定邏輯複寫。

主題

- Aurora MySQL
- Aurora PostgreSQL

Aurora MySQL

完成下列步驟，以使用 Amazon Aurora for Aurora MySQL 設定 OpenSearch 擷取管道。

主題

- Aurora MySQL 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道
- 資料一致性
- 映射資料類型
- 限制
- 建議的 CloudWatch 警示

Aurora MySQL 先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 在 Amazon Aurora 中建立自訂 Aurora 資料庫叢集參數群組，以設定二進位記錄。

   ```
   aurora_enhanced_binlog=1
   binlog_backup=0
   binlog_format=ROW
   binlog_replication_globaldb=0
   binlog_row_image=full
   binlog_row_metadata=full
   ```

   此外，請確定 `binlog_transaction_compression` 參數未設定為 `ON`，且 `binlog_row_value_options` 參數未設定為 `PARTIAL_JSON`。
2. 選取或建立 Aurora MySQL 資料庫叢集，並將上一個步驟中建立的參數群組與資料庫叢集建立關聯。
3. 將二進位日誌保留期設定為 24 小時或更久。
4. 使用具有 Aurora 和 的密碼管理，在您的 Amazon Aurora AWS Secrets Manager 叢集上設定使用者名稱和密碼身分驗證。您也可以建立 Secrets Manager 秘密來建立使用者名稱/密碼組合。
5. 如果您使用完整的初始快照功能，請建立 AWS KMS key 和 IAM 角色，以將資料從 Amazon Aurora 匯出到 Amazon S3。

   IAM 角色應具有下列許可政策：

   JSON

   ```
   {
   ```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ExportPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::s3-bucket-used-in-pipeline",
                "arn:aws:s3:::s3-bucket-used-in-pipeline/*"
            ]
        }
    ]
}
```

此角色也應該具有下列信任關係：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "export.rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

6. 選取或建立 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱建立 OpenSearch Service 網域和建立集合。
7. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從 Amazon Aurora 資料庫叢集寫入您的網域或集合。

步驟 1：設定管道角色

在您設定 Amazon Aurora 管道先決條件之後，請將管道角色設定為在管道組態中使用。同時將 Amazon Aurora 來源的下列許可新增至角色：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
    "Sid": "allowReadingFromS3Buckets",
    "Effect": "Allow",
    "Action": [
    "s3:GetObject",
    "s3:DeleteObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:PutObject"
    ],
    "Resource": [
    "arn:aws:s3:::s3_bucket",
    "arn:aws:s3:::s3_bucket/*"
    ]
    },
    {
    "Sid": "allowNetworkInterfacesActions",
    "Effect": "Allow",
    "Action": [
    "ec2:AttachNetworkInterface",
    "ec2:CreateNetworkInterface",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:DetachNetworkInterface",
    "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": [
    "arn:aws:ec2:*:111122223333:network-interface/*",
    "arn:aws:ec2:*:111122223333:subnet/*",
    "arn:aws:ec2:*:111122223333:security-group/*"
    ]
    },
```

```
    {
    "Sid": "allowDescribeEC2",
    "Effect": "Allow",
    "Action": [
    "ec2:Describe*"
    ],
    "Resource": "*"
    },
    {
    "Sid": "allowTagCreation",
    "Effect": "Allow",
    "Action": [
    "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:111122223333:network-interface/*",
    "Condition": {
    "StringEquals": {
    "aws:RequestTag/OSISManaged": "true"
    }
    }
    },
    {
    "Sid": "AllowDescribeInstances",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBInstances"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:db:*"
    ]
    },
    {
    "Sid": "AllowDescribeClusters",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBClusters"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:cluster:DB-id"
    ]
    },
    {
    "Sid": "AllowSnapshots",
    "Effect": "Allow",
```

```
    "Action": [
    "rds:DescribeDBClusterSnapshots",
    "rds:CreateDBClusterSnapshot",
    "rds:AddTagsToResource"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:cluster:DB-id",
    "arn:aws:rds:us-east-2:111122223333:cluster-snapshot:DB-id*"
    ]
    },
    {
    "Sid": "AllowExport",
    "Effect": "Allow",
    "Action": [
    "rds:StartExportTask"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:cluster:DB-id",
    "arn:aws:rds:us-east-2:111122223333:cluster-snapshot:DB-id*"
    ]
    },
    {
    "Sid": "AllowDescribeExports",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeExportTasks"
    ],
    "Resource": "*",
    "Condition": {
    "StringEquals": {
    "aws:RequestedRegion": "us-east-2",
    "aws:ResourceAccount": "111122223333"
    }
    }
    },
    {
    "Sid": "AllowAccessToKmsForExport",
    "Effect": "Allow",
    "Action": [
    "kms:Decrypt",
    "kms:Encrypt",
    "kms:DescribeKey",
    "kms:RetireGrant",
    "kms:CreateGrant",
```

```
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
    ],
    "Resource": [
    "arn:aws:kms:us-east-2:111122223333:key/export-key-id"
    ]
    },
    {
    "Sid": "AllowPassingExportRole",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
    "arn:aws:iam::111122223333:role/export-role"
    ]
    },
    {
    "Sid": "SecretsManagerReadAccess",
    "Effect": "Allow",
    "Action": [
    "secretsmanager:GetSecretValue"
    ],
    "Resource": [
    "arn:aws:secretsmanager:*:111122223333:secret:*"
    ]
    }
    ]
    }
```

步驟 2：建立管道

設定類似下列的 OpenSearch Ingestion 管道。範例管道指定 Amazon Aurora 叢集做為來源。

```
version: "2"
aurora-mysql-pipeline:
  source:
    rds:
      db_identifier: "cluster-id"
      engine: aurora-mysql
      database: "database-name"
      tables:
        include:
          - "table1"
          - "table2"
```

```
      s3_bucket: "bucket-name"
      s3_region: "bucket-region"
      s3_prefix: "prefix-name"
      export:
        kms_key_id: "kms-key-id"
        iam_role_arn: "export-role-arn"
      stream: true
      aws:
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        region: "us-east-1"
      authentication:
        username: ${{aws_secrets:secret:username}}
        password: ${{aws_secrets:secret:password}}
  sink:
    - opensearch:
        hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
        index: "${getMetadata(\"table_name\")}"
        index_type: custom
        document_id: "${getMetadata(\"primary_key\")}"
        action: "${getMetadata(\"opensearch_action\")}"
        document_version: "${getMetadata(\"document_version\")}"
        document_version_type: "external"
        aws:
          sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
          region: "us-east-1"
extension:
  aws:
    secrets:
      secret:
        secret_id: "rds-secret-id"
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        refresh_interval: PT1H
```

您可以使用預先設定的 Amazon Aurora 藍圖來建立此管道。如需詳細資訊，請參閱使用藍圖。

若要使用 Amazon Aurora 做為來源，您需要設定管道的 VPC 存取。您選擇的 VPC 應與 Amazon Aurora 來源使用的 VPC 相同。然後選擇一或多個子網路和一或多個 VPC 安全群組。請注意，管道需要網路存取 Aurora MySQL 資料庫，因此您也應該確認您的 Aurora 叢集已設定 VPC 安全群組，允許從管道的 VPC 安全群組到資料庫連接埠的傳入流量。如需詳細資訊，請參閱使用安全群組控制存取。

如果您使用 AWS 管理主控台 建立管道，也必須將管道連接至 VPC，才能使用 Amazon Aurora 做為來源。若要這樣做，請尋找網路組態區段，選取連接至 VPC 核取方塊，然後從其中一個提供的預設選

項中選擇 CIDR，或選取您自己的選項。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。

若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 和 Amazon Aurora 之間的 IP 地址發生衝突，請確定 Amazon Aurora VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

如需詳細資訊，請參閱設定管道的 VPC 存取。

資料一致性

管道會持續輪詢或接收來自 Amazon Aurora 叢集的變更，並更新 OpenSearch 索引中的對應文件，以確保資料一致性。

OpenSearch Ingestion end-to-end確認，以確保資料耐久性。當管道讀取快照或串流時，它會動態建立分割區以進行平行處理。在擷取 OpenSearch 網域或集合中的所有記錄之後，管道收到確認時，會將分割區標記為完成。如果您想要擷取到 OpenSearch Serverless 搜尋集合，您可以在管道中產生文件 ID。如果您想要擷取 OpenSearch Serverless 時間序列集合，請注意管道不會產生文件 ID，因此您必須在管道接收器組態`document_id: "${getMetadata(\"primary_key\")}"`中省略 。

OpenSearch 擷取管道也會將傳入的事件動作映射至對應的大量索引動作，以協助擷取文件。這可讓資料保持一致，以便 Amazon Aurora 中的每個資料變更都與 OpenSearch 中的對應文件變更進行協調。

映射資料類型

OpenSearch Ingestion 管道會將 MySQL 資料類型映射至適合 OpenSearch Service 網域或集合使用的表示法。如果 OpenSearch 中未定義映射範本，則 OpenSearch 會根據第一個傳送的文件自動判斷具有動態映射的欄位類型。您也可以透過映射範本，在 OpenSearch 中明確定義最適合您的欄位類型。

下表列出 MySQL 資料類型和對應的 OpenSearch 欄位類型。如果未定義明確的映射，預設 OpenSearch 欄位類型欄會顯示 OpenSearch 中的對應欄位類型。在此情況下，OpenSearch 會自動判斷具有動態映射的欄位類型。建議的 OpenSearch 欄位類型欄是建議在映射範本中明確指定的對應欄位類型。這些欄位類型與 MySQL 中的資料類型更緊密一致，通常可以在 OpenSearch 中啟用更好的搜尋功能。

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
BIGINT	long	long

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
BIGINT UNSIGNED	long	未簽署的長
BIT	long	位元組、短、整數或長，取決於位元數
DECIMAL	text	雙 或 關鍵字
DOUBLE	float	double
FLOAT	float	float
INT	long	integer
INT UNSIGNED	long	long
MEDIUMIN	long	integer
MEDIUMIN UNSIGNED	long	integer
NUMERIC	text	雙 或 關鍵字
SMALLINT	long	short
SMALLINT UNSIGNED	long	integer
TINYINT	long	byte
TINYINT UNSIGNED	long	short
BINARY	text	binary
BLOB	text	binary
CHAR	text	text

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
ENUM	text	關鍵字
LONGBLOB	text	binary
LONGTEXT	text	text
MEDIUMBLOB	text	binary
MEDIUMTEXT	text	text
SET	text	關鍵字
TEXT	text	text
TINYBLOB	text	binary
TINYTEXT	text	text
VARBINARY	text	binary
VARCHAR	text	text
DATE	長（以 epoch 毫秒為單位）	date
DATETIME	長（以 epoch 毫秒為單位）	date
TIME	長（以 epoch 毫秒為單位）	date
TIMESTAMP	長（以 epoch 毫秒為單位）	date
YEAR	長（以 epoch 毫秒為單位）	date
GEOMETRY	text (WKT 格式）	geo_shape
GEOMETRYCOLLECTION	text (WKT 格式）	geo_shape

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
LINESTRING	text (WKT 格式)	geo_shape
MULTILINESTRING	text (WKT 格式)	geo_shape
MULTIPOINT	text (WKT 格式)	geo_shape
MULTIPOLYGON	text (WKT 格式)	geo_shape
POINT	text (WKT 格式)	geo_point 或 geo_shape
POLYGON	text (WKT 格式)	geo_shape
JSON	text	object

建議您在 OpenSearch Ingestion 管道中設定無效字母佇列 (DLQ)。如果您已設定佇列，OpenSearch Service 會將因動態映射失敗而無法擷取的所有失敗文件傳送至佇列。

如果自動映射失敗，您可以在管道組態`template_content`中使用 `template_type`和 來定義明確的映射規則。或者，您可以在啟動管道之前，直接在搜尋網域或集合中建立映射範本。

限制

當您為 Aurora MySQL 設定 OpenSearch 擷取管道時，請考慮下列限制：

- 整合每個管道僅支援一個 MySQL 資料庫。
- 整合目前不支援跨區域資料擷取；您的 Amazon Aurora 叢集和 OpenSearch 網域必須位於相同的 中 AWS 區域。
- 整合目前不支援跨帳戶資料擷取；您的 Amazon Aurora 叢集和 OpenSearch Ingestion 管道必須位於相同的 中 AWS 帳戶。
- 確保 Amazon Aurora 叢集已使用 Secrets Manager 啟用身分驗證，Secrets Manager 是唯一支援的身分驗證機制。

- 現有的管道組態無法更新，無法從不同的資料庫和/或不同的資料表擷取資料。若要更新管道的資料庫和/或資料表名稱，您必須停止管道，並使用更新的組態重新啟動管道，或建立新的管道。
- 通常不支援資料定義語言 (DDL) 陳述式。如果符合下列條件，將不會維持資料一致性：
  - 主索引鍵已變更 add/delete/rename)。
  - 資料表遭到捨棄/截斷。
  - 資料欄名稱或資料類型已變更。
- 如果要同步的 MySQL 資料表未定義主索引鍵，則不保證資料一致性。您需要在 OpenSearch 接收器組態中正確定義自訂`document_id`選項，才能將更新/刪除同步至 OpenSearch。
- 不支援具有串聯刪除動作的外部金鑰參考，這可能會導致 Aurora MySQL 和 OpenSearch 之間的資料不一致。
- 支援的版本：Aurora MySQL 3.05.2 版及更新版本。

建議的 CloudWatch 警示

建議使用下列 CloudWatch 指標來監控擷取管道的效能。這些指標可協助您識別從匯出處理的資料量、從串流處理的事件數、處理匯出和串流事件的錯誤，以及寫入目的地的文件數。您可以設定 CloudWatch 警示，在其中一個指標在指定的時間內超過指定的值時執行動作。

指標	Description
pipeline-name .rds.credentialsChanged	此指標表示 AWS 秘密輪換的頻率。
pipeline-name .rds.executorRefreshErrors	此指標表示重新整理 AWS 秘密失敗。
pipeline-name .rds.exportRecordsTotal	此指標表示從 Amazon Aurora 匯出的記錄數目。
pipeline-name .rds.exportRecordsProcessed	此指標表示 OpenSearch Ingestion 管道處理的記錄數量。

指標	Description
pipeline-name .rds.exportRecordProcessingErrors	此指標表示從 Amazon Aurora 叢集讀取資料時OpenSearch 擷取管道中的處理錯誤數目。
pipeline-name .rds.exportRecordsSuccessTotal	此指標表示成功處理的匯出記錄總數。
pipeline-name .rds.exportRecordsFailedTotal	此指標表示無法處理的匯出記錄總數。
pipeline-name .rds.bytesReceived	此指標表示 OpenSearch Ingestion 管道收到的位元組總數。
pipeline-name .rds.bytesProcessed	此指標表示 OpenSearch Ingestion 管道處理的位元組總數。
pipeline-name .rds.streamRecordsSuccessTotal	此指標表示從串流成功處理的記錄數。
pipeline-name .rds.streamRecordsFailedTotal	此指標表示無法從串流處理的記錄總數。

Aurora PostgreSQL

完成下列步驟，以使用 Amazon Aurora for Aurora PostgreSQL 設定 OpenSearch 擷取管道。

主題

- Aurora PostgreSQL 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道
- 資料一致性
- 映射資料類型
- 限制
- 建議的 CloudWatch 警示

Aurora PostgreSQL 先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 在 Amazon Aurora 中建立自訂資料庫叢集參數群組，以設定邏輯複寫。

```
rds.logical_replication=1
    aurora.enhanced_logical_replication=1
    aurora.logical_replication_backup=0
    aurora.logical_replication_globaldb=0
```

2. 選取或建立 Aurora PostgreSQL 資料庫叢集，並將步驟 1 中建立的參數群組與資料庫叢集建立關聯。
3. 使用密碼管理搭配 Aurora 和 在 Amazon Aurora AWS Secrets Manager 叢集上設定使用者名稱和密碼身分驗證。您也可以建立 Secrets Manager 秘密來建立使用者名稱/密碼組合。
4. 如果您使用完整的初始快照功能，請建立 AWS KMS key 和 IAM 角色，將資料從 Amazon Aurora 匯出到 Amazon S3。

   IAM 角色應具有下列許可政策：

   JSON

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

```
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject*",
                    "s3:ListBucket",
                    "s3:GetObject*",
                    "s3:DeleteObject*",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::s3-bucket-used-in-pipeline",
                    "arn:aws:s3:::s3-bucket-used-in-pipeline/*"
                ]
            }
        ]
    }
```

此角色也應該具有下列信任關係：

JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "export.rds.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    }
```

5. 選取或建立 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱建立 OpenSearch Service 網域和建立集合。
6. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從 Amazon Aurora 資料庫叢集寫入您的網域或集合。

步驟 1：設定管道角色

設定 Amazon Aurora 管道先決條件之後，請將管道角色設定為在管道組態中使用。同時將 Amazon Aurora 來源的下列許可新增至角色：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
    "Sid": "allowReadingFromS3Buckets",
    "Effect": "Allow",
    "Action": [
    "s3:GetObject",
    "s3:DeleteObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:PutObject"
    ],
    "Resource": [
    "arn:aws:s3:::s3_bucket",
    "arn:aws:s3:::s3_bucket/*"
    ]
    },
    {
    "Sid": "allowNetworkInterfacesActions",
    "Effect": "Allow",
    "Action": [
    "ec2:AttachNetworkInterface",
    "ec2:CreateNetworkInterface",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:DetachNetworkInterface",
    "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": [
    "arn:aws:ec2:*:111122223333:network-interface/*",
    "arn:aws:ec2:*:111122223333:subnet/*",
    "arn:aws:ec2:*:111122223333:security-group/*"
    ]
    },
    {
    "Sid": "allowDescribeEC2",
    "Effect": "Allow",
    "Action": [
    "ec2:Describe*"
```

```
    ],
    "Resource": "*"
    },
    {
    "Sid": "allowTagCreation",
    "Effect": "Allow",
    "Action": [
    "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:111122223333:network-interface/*",
    "Condition": {
    "StringEquals": {
    "aws:RequestTag/OSISManaged": "true"
    }
    }
    },
    {
    "Sid": "AllowDescribeInstances",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBInstances"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:db:*"
    ]
    },
    {
    "Sid": "AllowDescribeClusters",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBClusters"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:cluster:DB-id"
    ]
    },
    {
    "Sid": "AllowSnapshots",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBClusterSnapshots",
    "rds:CreateDBClusterSnapshot",
    "rds:AddTagsToResource"
    ],
```

```
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:cluster:DB-id",
    "arn:aws:rds:us-east-2:111122223333:cluster-snapshot:DB-id*"
    ]
    },
    {
    "Sid": "AllowExport",
    "Effect": "Allow",
    "Action": [
    "rds:StartExportTask"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:cluster:DB-id",
    "arn:aws:rds:us-east-2:111122223333:cluster-snapshot:DB-id*"
    ]
    },
    {
    "Sid": "AllowDescribeExports",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeExportTasks"
    ],
    "Resource": "*",
    "Condition": {
    "StringEquals": {
    "aws:RequestedRegion": "us-east-2",
    "aws:ResourceAccount": "111122223333"
    }
    }
    },
    {
    "Sid": "AllowAccessToKmsForExport",
    "Effect": "Allow",
    "Action": [
    "kms:Decrypt",
    "kms:Encrypt",
    "kms:DescribeKey",
    "kms:RetireGrant",
    "kms:CreateGrant",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
    ],
    "Resource": [
    "arn:aws:kms:us-east-2:111122223333:key/export-key-id"
```

```
    ]
    },
    {
    "Sid": "AllowPassingExportRole",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
    "arn:aws:iam::111122223333:role/export-role"
    ]
    },
    {
    "Sid": "SecretsManagerReadAccess",
    "Effect": "Allow",
    "Action": [
    "secretsmanager:GetSecretValue"
    ],
    "Resource": [
    "arn:aws:secretsmanager:*:111122223333:secret:*"
    ]
    }
    ]
    }
```

步驟 2：建立管道

如下所示設定 OpenSearch Ingestion 管道，指定 Aurora PostgreSQL 叢集做為來源。

```
version: "2"
aurora-postgres-pipeline:
  source:
    rds:
      db_identifier: "cluster-id"
      engine: aurora-postgresql
      database: "database-name"
      tables:
        include:
          - "schema1.table1"
          - "schema2.table2"
      s3_bucket: "bucket-name"
      s3_region: "bucket-region"
      s3_prefix: "prefix-name"
      export:
        kms_key_id: "kms-key-id"
```

```
        iam_role_arn: "export-role-arn"
      stream: true
      aws:
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        region: "us-east-1"
      authentication:
        username: ${{aws_secrets:secret:username}}
        password: ${{aws_secrets:secret:password}}
  sink:
    - opensearch:
        hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
        index: "${getMetadata(\"table_name\")}"
        index_type: custom
        document_id: "${getMetadata(\"primary_key\")}"
        action: "${getMetadata(\"opensearch_action\")}"
        document_version: "${getMetadata(\"document_version\")}"
        document_version_type: "external"
        aws:
          sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
          region: "us-east-1"
extension:
  aws:
    secrets:
      secret:
        secret_id: "rds-secret-id"
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        refresh_interval: PT1H
```

> **Note**
>
> 您可以使用預先設定的 Amazon Aurora 藍圖來建立此管道。如需詳細資訊，請參閱使用藍圖。

若要使用 Amazon Aurora 做為來源，您需要設定管道的 VPC 存取。您選擇的 VPC 應與 Amazon Aurora 來源使用的 VPC 相同。然後選擇一或多個子網路和一或多個 VPC 安全群組。請注意，管道需要網路存取 Aurora MySQL 資料庫，因此您也應該確認您的 Aurora 叢集已設定 VPC 安全群組，允許從管道的 VPC 安全群組到資料庫連接埠的傳入流量。如需詳細資訊，請參閱使用安全群組控制存取。

如果您使用 AWS 管理主控台 建立管道，也必須將管道連接至 VPC，才能使用 Amazon Aurora 做為來源。若要執行此作業，請尋找網路組態區段，選擇連接至 VPC，然後從提供的預設選項中選擇

CIDR，或選取您自己的 CIDR。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。

若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 和 Amazon Aurora 之間的 IP 地址發生衝突，請確保 Amazon Aurora VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

如需詳細資訊，請參閱設定管道的 VPC 存取。

資料一致性

管道會持續輪詢或接收來自 Amazon Aurora 叢集的變更，並更新 OpenSearch 索引中的對應文件，以確保資料一致性。

OpenSearch Ingestion end-to-end確認，以確保資料耐久性。當管道讀取快照或串流時，它會動態建立分割區以進行平行處理。在擷取 OpenSearch 網域或集合中的所有記錄之後，管道收到確認時，會將分割區標記為完成。如果您想要擷取到 OpenSearch Serverless 搜尋集合，您可以在管道中產生文件 ID。如果您想要擷取 OpenSearch Serverless 時間序列集合，請注意管道不會產生文件 ID，因此您必須在管道接收器組態`document_id: "${getMetadata(\"primary_key\")}"`中省略 。

OpenSearch 擷取管道也會將傳入的事件動作映射至對應的大量索引動作，以協助擷取文件。這可讓資料保持一致，以便 Amazon Aurora 中的每個資料變更都與 OpenSearch 中的對應文件變更進行協調。

映射資料類型

OpenSearch Ingestion 管道會將 Aurora PostgreSQL 資料類型映射至適合 OpenSearch Service 網域或集合使用的表示法。如果 OpenSearch 中未定義映射範本，則 OpenSearch 會根據第一個傳送的文件自動判斷具有動態映射的欄位類型。您也可以透過映射範本，明確定義最適合您在 OpenSearch 中的欄位類型。

下表列出 Aurora PostgreSQL 資料類型和對應的 OpenSearch 欄位類型。如果未定義明確的映射，預設 OpenSearch 欄位類型欄會顯示 OpenSearch 中的對應欄位類型。在此情況下，OpenSearch 會自動判斷具有動態映射的欄位類型。建議的 OpenSearch 欄位類型欄是對應範本中明確指定的對應建議欄位類型。這些欄位類型與 Aurora PostgreSQL 中的資料類型更緊密一致，通常可以在 OpenSearch 中啟用更好的搜尋功能。

Aurora PostgreSQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
smallint	long	short
integer	long	integer
bigint	long	long
decimal	text	雙 或 關鍵字
numeric【(	text	雙 或 關鍵字
real	float	float
double precision	float	double
smallserial	long	short
serial	long	integer
bigserial	long	long
money	object	object
字元變體 (n)	text	text
varchar(n)	text	text
character (n)	text	text
char(n)	text	text
bpchar(n)	text	text

Aurora PostgreSQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
bpchar	text	text
text	text	text
enum	text	text
bytea	text	binary
時間戳記【(p)】【不含時區】	長（以 epoch 毫秒為單位）	date
具有時區的時間戳記【(p)】	長（以 epoch 毫秒為單位）	date
date	長（以 epoch 毫秒為單位）	date
time【(p)】【無時區】	長（以 epoch 毫秒為單位）	date
time【(p)】與時區	長（以 epoch 毫秒為單位）	date
間隔【欄位】【(p)】	text (ISO8601 格式）	text
boolean	boolean	boolean

Aurora PostgreSQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
point	text (WKT 格式)	geo_shape
線條	text (WKT 格式)	geo_shape
lseg	text (WKT 格式)	geo_shape
方塊	text (WKT 格式)	geo_shape
路徑	text (WKT 格式)	geo_shape
多邊形	text (WKT 格式)	geo_shape
圓圈	object	object
cidr	text	text
inet	text	text
macaddr	text	text
macaddr8	text	text
bit(n)	long	位元組、短、整數或長 (取決於位元數)
bit varying(n)	long	位元組、短、整數或長 (取決於位元數)
json	object	object
jsonb	object	object
jsonpath	text	text

我們建議您在 OpenSearch Ingestion 管道中設定無效字母佇列 (DLQ)。如果您已設定佇列，OpenSearch Service 會將因動態映射失敗而無法擷取的所有失敗文件傳送至佇列。

如果自動映射失敗，您可以在管道組態`template_content`中使用 `template_type`和 來定義明確的映射規則。或者，您可以在啟動管道之前，直接在搜尋網域或集合中建立映射範本。

限制

當您為 Aurora PostgreSQL 設定 OpenSearch 擷取管道時，請考慮下列限制：

- 整合每個管道僅支援一個 Aurora PostgreSQL 資料庫。
- 整合目前不支援跨區域資料擷取；您的 Amazon Aurora 叢集和 OpenSearch 網域必須位於相同的 中 AWS 區域。
- 整合目前不支援跨帳戶資料擷取；您的 Amazon Aurora 叢集和 OpenSearch Ingestion 管道必須位於相同的 中 AWS 帳戶。
- 確保 Amazon Aurora 叢集已使用 啟用身分驗證 AWS Secrets Manager，這是唯一支援的身分驗證機制。
- 現有的管道組態無法更新，無法從不同的資料庫和/或不同的資料表擷取資料。若要更新管道的資料庫和/或資料表名稱，您必須停止管道，並使用更新的組態重新啟動管道，或建立新的管道。
- 通常不支援資料定義語言 (DDL) 陳述式。如果符合下列條件，將不會維持資料一致性：
  - 主索引鍵已變更 add/delete/rename)。
  - 資料表遭到捨棄/截斷。
  - 資料欄名稱或資料類型已變更。
- 如果要同步的 Aurora PostgreSQL 資料表未定義主索引鍵，則不保證資料一致性。您需要在 OpenSearch 中正確定義自訂 `document_id` 選項，才能將更新/刪除同步至 OpenSearch。
- 支援的版本：Aurora PostgreSQL 16.4 版及更新版本。

建議的 CloudWatch 警示

建議使用下列 CloudWatch 指標來監控擷取管道的效能。這些指標可協助您識別從匯出處理的資料量、從串流處理的事件數、處理匯出和串流事件的錯誤，以及寫入目的地的文件數。您可以設定 CloudWatch 警示，在其中一個指標在指定的時間內超過指定的值時執行動作。

指標	Description
pipeline-name .rds.credentialsChanged	此指標表示 AWS 秘密輪換的頻率。
pipeline-name .rds.executorRefreshErrors	此指標表示重新整理 AWS 秘密失敗。
pipeline-name .rds.exportRecordsTotal	此指標表示從 Amazon Aurora 匯出的記錄數目。
pipeline-name .rds.exportRecordsProcessed	此指標表示 OpenSearch Ingestion 管道處理的記錄數量。
pipeline-name .rds.exportRecordProcessingErrors	此指標表示從 Amazon Aurora 叢集讀取資料時OpenSearch Ingestion 管道中的處理錯誤數目。
pipeline-name .rds.exportRecordsSuccessTotal	此指標表示成功處理的匯出記錄總數。
pipeline-name .rds.exportRecordsFailedTotal	此指標表示無法處理的匯出記錄總數。
pipeline-name .rds.bytesReceived	此指標表示 OpenSearch Ingestion 管道收到的位元組總數。

指標	Description
pipeline-name .rds.bytesProcessed	此指標表示 OpenSearch Ingestion 管道處理的位元組總數。
pipeline-name .rds.streamRecordsSuccessTotal	此指標表示從串流成功處理的記錄數。
pipeline-name .rds.streamRecordsFailedTotal	此指標表示無法從串流處理的記錄總數。

搭配 Amazon DynamoDB 使用 OpenSearch 擷取管道

您可以使用 DynamoDB 外掛程式，將建立、更新和刪除等資料表事件串流至 Amazon OpenSearch Service 網域和 Amazon OpenSearch Serverless 集合。管道使用變更資料擷取 (CDC) 進行大規模、低延遲串流。

您可以使用或不使用完整初始快照來處理 DynamoDB 資料。

- 使用完整快照 – DynamoDB point-in-time復原 (PITR) 來建立備份並將其上傳至 Amazon S3。OpenSearch Ingestion 接著會在一或多個 OpenSearch 索引中為快照編製索引。為了保持一致性，管道會同步所有 DynamoDB 變更與 OpenSearch。此選項需要您同時啟用 PITR 和 DynamoDB Streams。
- 沒有快照 – OpenSearch Ingestion 只會串流新的 DynamoDB 事件。如果您已經擁有快照或需要沒有歷史資料的即時串流，請選擇此選項。此選項要求您僅啟用 DynamoDB Streams。

如需詳細資訊，請參閱《 Amazon DynamoDB 開發人員指南》中的 DynamoDB 零 ETL 與 Amazon OpenSearch Service 整合。

主題

- 先決條件
- 步驟 1：設定管道角色

- 步驟 2：建立管道
- 資料一致性
- 映射資料類型
- 限制
- DynamoDB 的建議 CloudWatch 警示

先決條件

若要設定管道，您必須啟用 DynamoDB Streams 的 DynamoDB 資料表。您的串流應使用 NEW_IMAGE 串流檢視類型。不過，NEW_AND_OLD_IMAGES如果此串流檢視類型符合您的使用案例，OpenSearch Ingestion 管道也可以使用 串流事件。

如果您使用的是快照，您還必須在資料表上啟用point-in-time復原。如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的建立資料表、啟用point-in-time復原和啟用串流。

步驟 1：設定管道角色

設定 DynamoDB 資料表之後，請設定要在管道組態中使用的管道角色，並在角色中新增下列 DynamoDB 許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "allowRunExportJob",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:ExportTableToPointInTime"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:111122223333:table/my-table"
            ]
        },
        {
            "Sid": "allowCheckExportjob",
```

```
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeExport"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:111122223333:table/my-table/export/*"
            ]
        },
        {
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeStream",
                "dynamodb:GetRecords",
                "dynamodb:GetShardIterator"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:111122223333:table/my-table/stream/*"
            ]
        },
        {
            "Sid": "allowReadAndWriteToS3ForExport",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/export-folder/*"
            ]
        }
    ]
}
```

您也可以使用 AWS KMS 客戶受管金鑰來加密匯出資料檔案。若要解密匯出的物件，請在管道的匯出組態中s3_sse_kms_key_id指定 金鑰 ID，格式如下：arn:aws:kms:*region*:*account-id*:key/*my-key-id*。下列政策包含使用客戶受管金鑰的必要許可：

```
{
    "Sid": "allowUseOfCustomManagedKey",
```

```
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
    ],
    "Resource": arn:aws:kms:region:account-id:key/my-key-id
}
```

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 DynamoDB 作為來源。此範例管道`table-a`會使用 PITR 快照從 擷取資料，接著從 DynamoDB Streams 擷取事件。的開始位置`LATEST`表示管道應從 DynamoDB Streams 讀取最新資料。

```
version: "2"
cdc-pipeline:
  source:
    dynamodb:
      tables:
      - table_arn: "arn:aws:dynamodb:region:account-id:table/table-a"
        export:
          s3_bucket: "my-bucket"
          s3_prefix: "export/"
        stream:
          start_position: "LATEST"
      aws:
        region: "us-east-1"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.region.es.amazonaws.com"]
      index: "${getMetadata(\"table-name\")}"
      index_type: custom
      normalize_index: true
      document_id: "${getMetadata(\"primary_key\")}"
      action: "${getMetadata(\"opensearch_action\")}"
      document_version: "${getMetadata(\"document_version\")}"
      document_version_type: "external"
```

您可以使用預先設定的 DynamoDB 藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

資料一致性

OpenSearch Ingestion end-to-end確認，以確保資料耐久性。當管道讀取快照或串流時，它會動態建立分割區以進行平行處理。在擷取 OpenSearch 網域或集合中的所有記錄之後，管道收到確認時，會將分割區標記為完成。

如果您想要擷取到 OpenSearch Serverless 搜尋集合，您可以在管道中產生文件 ID。如果您想要擷取至 OpenSearch Serverless 時間序列集合，請注意管道不會產生文件 ID。

OpenSearch 擷取管道也會將傳入的事件動作映射至對應的大量索引動作，以協助擷取文件。這可讓資料保持一致，以便 DynamoDB 中的每個資料變更都與 OpenSearch 中的對應文件變更進行協調。

映射資料類型

OpenSearch Service 會將每個傳入文件中的資料類型動態映射至 DynamoDB 中對應的資料類型。下表顯示 OpenSearch Service 如何自動映射各種資料類型。

資料類型	OpenSearch	DynamoDB
Number	OpenSearch 會自動映射數值資料。如果數字是整數，則 OpenSearch 會將該數字對應為長值。如果數字是小數，則 OpenSearch 會將其映射為浮點值。 OpenSearch 會根據第一個傳送的文件動態映射各種屬性。如果您在 DynamoDB 中具有相同屬性的混合資料類型，例如整數和小數，映射可能會失敗。 例如，如果您的第一個文件具有整數的屬性，而稍後的文件具有與小數相同的屬性，則 OpenSearch 無法擷取第二個文件。在這些情況下，您應該提供明確的映射範本，如下所示： `{ "template": { "mappings": { "properties": { "MixedNumberAttribute": {`	DynamoDB 支援號碼。

資料類型	OpenSearch	DynamoDB

	<pre>"type": "float"
	}
	}
	}
	}
	}</pre>
	如果您需要雙精度，請使用字串類型欄位映射。沒有支援 OpenSearch 中 38 位數精確度的同等數字類型。	

號碼集	OpenSearch 會自動將設定的數字映射至長值或浮點值的陣列。如同純量數字，這取決於擷取的第一個數字是整數還是小數。您可以提供數字集的映射，方式與映射純量字串相同。	DynamoDB 支援代表一組數字的類型。

String	OpenSearch 會自動將字串值映射為文字。在某些情況下，例如列舉值，您可以映射到關鍵字類型。
	
	下列範例示範如何將名為 的 DynamoDB 屬性映射`PartType`至 OpenSearch 關鍵字。
	
	<pre>{
	"template": {
	"mappings": {
	"properties": {
	"PartType": {
	"type": "keyword"
	}
	}
	}
	}
	}</pre>
		DynamoDB 支援字串。

資料類型	OpenSearch	DynamoDB
字串集	OpenSearch 會自動將字串集映射至字串陣列。您可以提供字串集的映射，方法與映射純量字串相同。	DynamoDB 支援代表字串集的類型。
二進位	OpenSearch 會自動將二進位資料映射為文字。您可以提供映射，以將這些項目寫入 OpenSearch 中的二進位欄位。 下列範例示範如何將名為 的 DynamoDB 屬性映射`ImageData` 至 OpenSearch 二進位欄位。 `{ "template": { "mappings": { "properties": { "ImageData": { "type": "binary" } } } } }`	DynamoDB 支援二進位類型屬性。
二進位集	OpenSearch 會自動將二進位集映射到二進位資料陣列中做為文字。您可以提供數字集的映射，方式與映射純量二進位檔的方式相同。	DynamoDB 支援代表二進位值集的類型。
Boolean	OpenSearch 會將 DynamoDB 布林類型映射至 OpenSearch 布林類型。	DynamoDB 支援布林值類型屬性。

資料類型	OpenSearch	DynamoDB
Null	OpenSearch 可以使用 DynamoDB null 類型擷取文件。它會將值儲存為文件中的 null 值。沒有此類型的映射，而且此欄位無法編製索引或搜尋。 如果將相同的屬性名稱用於 null 類型，然後變更為不同類型的，例如字串，則 OpenSearch 會為第一個非 Null 值建立動態映射。後續值仍然可以是 DynamoDB null 值。	DynamoDB 支援 null 類型屬性。
Map	OpenSearch 會將 DynamoDB 映射屬性映射至巢狀欄位。相同的映射適用於巢狀欄位。 下列範例會將巢狀欄位中的字串映射至 OpenSearch 中的關鍵字類型： `{ "template": { "mappings": { "properties": { "AdditionalDescriptions": { "properties": { "PartType": { "type": "keyword" } } } } } } }`	DynamoDB 支援映射類型屬性。

資料類型	OpenSearch	DynamoDB
清單	OpenSearch 會根據清單中的內容，為 DynamoDB 清單提供不同的結果。 當清單包含所有相同類型的純量類型（例如，所有字串的清單）時，OpenSearch 會將清單擷取為該類型的陣列。這適用於字串、數字、布林值和 null 類型。每種類型的限制都與該類型的純量限制相同。 您也可以使用與映射相同的映射來提供映射清單的映射。 您無法提供混合類型的清單。	DynamoDB 支援清單類型屬性。
設定	OpenSearch 會根據集合中的內容，為 DynamoDB 集提供不同的結果。 當集合包含所有相同類型的純量類型（例如，一組所有字串）時，OpenSearch 會將集合擷取為該類型的陣列。這適用於字串、數字、布林值和 null 類型。每種類型的限制都與該類型的純量限制相同。 您也可以使用與映射相同的映射來提供映射集。 您無法提供一組混合類型。	DynamoDB 支援代表集合的類型。

我們建議您在 OpenSearch Ingestion 管道中設定無效字母佇列 (DLQ)。如果您已設定佇列，OpenSearch Service 會將因動態映射失敗而無法擷取的所有失敗文件傳送至佇列。

如果自動映射失敗，您可以在管道組態`template_content`中使用 `template_type`和 來定義明確的映射規則。或者，您可以在啟動管道之前，直接在搜尋網域或集合中建立映射範本。

限制

當您為 DynamoDB 設定 OpenSearch Ingestion 管道時，請考慮下列限制：

- OpenSearch Ingestion 與 DynamoDB 整合目前不支援跨區域擷取。您的 DynamoDB 資料表和 OpenSearch 擷取管道必須位於相同的 中 AWS 區域。
- 您的 DynamoDB 資料表和 OpenSearch Ingestion 管道必須位於相同的 中 AWS 帳戶。
- OpenSearch Ingestion 管道僅支援一個 DynamoDB 資料表作為其來源。
- DynamoDB Streams 最多只會將資料存放在日誌中 24 小時。如果從大型資料表的初始快照擷取需要 24 小時或更長時間，則會有一些初始資料遺失。若要緩解此資料遺失，請估計資料表的大小，並設定 OpenSearch Ingestion 管道的適當運算單位。

DynamoDB 的建議 CloudWatch 警示

建議使用以下 CloudWatch 指標來監控擷取管道的效能。這些指標可協助您識別從匯出處理的資料量、從串流處理的事件量、處理匯出和串流事件的錯誤，以及寫入目的地的文件數量。您可以設定 CloudWatch 警示，在其中一個指標在指定的時間內超過指定的值時執行動作。

指標	Description
`dynamodb-pipeline.BlockingBuffer.bufferUsage.value`	指出使用了多少緩衝區。
`dynamodb-pipeline.dynamodb.activeExportS3ObjectConsumers.value`	顯示正在為匯出處理 Amazon S3 物件OCUs 總數。
`dynamodb-pipeline.dynamodb.bytesProcessed.count`	從 DynamoDB 來源處理的位元組數。
`dynamodb-pipeline.dynamodb.changeEventsProcessed.count`	從 DynamoDB 串流處理的變更事件數目。
`dynamodb-pipeline.dynamodb.changeEventsProcessingErrors.count`	從 DynamoDB 處理之變更事件的錯誤數目。

指標	Description
dynamodb-pipeline.dynamodb.exportJobFailure.count	Number of export job submission attempts that have failed.
dynamodb-pipeline.dynamodb.exportJobSuccess.count	Number of export jobs that have been submitted successfully.
dynamodb-pipeline.dynamodb.exportRecordsProcessed.count	從匯出處理的記錄總數。
dynamodb-pipeline.dynamodb.exportRecordsTotal.count	從 DynamoDB 匯出的記錄總數，對於追蹤資料匯出磁碟區至關重要。
dynamodb-pipeline.dynamodb.exportS3ObjectsProcessed.count	Total number of export data files that have been processed successfully from Amazon S3.
dynamodb-pipeline.opensearch.bulkBadRequestErrors.count	Count of errors during bulk requests due to malformed request.
dynamodb-pipeline.opensearch.bulkRequestLatency.avg	Average latency for bulk write requests made to OpenSearch.
dynamodb-pipeline.opensearch.bulkRequestNotFoundErrors.count	Number of bulk requests that failed because the target data could not be found.
dynamodb-pipeline.opensearch.bulkRequestNumberOfRetries.count	Number of retries by OpenSearch Ingestion pipelines to write OpenSearch cluster.
dynamodb-pipeline.opensearch.bulkRequestSizeBytes.sum	Total size in bytes of all bulk requests made to OpenSearch.
dynamodb-pipeline.opensearch.documentErrors.count	Number of errors when sending documents to OpenSearch. The documents causing the errors witll be sent to DLQ.
dynamodb-pipeline.opensearch.documentsSuccess.count	Number of documents successfully written to an OpenSearch cluster or collection.

指標	Description
dynamodb-pipeline.opensearch.documentsSuccessFirstAttempt.count	Number of documents successfully indexed in OpenSearch on the first attempt.
dynamodb-pipeline.opensearch.documentsVersionConflictErrors.count	Count of errors due to version conflicts in documents during processing.
dynamodb-pipeline.opensearch.PipelineLatency.avg	Average latency of OpenSearch Ingestion pipeline to process the data by reading from the source to writint to the destination.
dynamodb-pipeline.opensearch.PipelineLatency.max	Maximum latency of OpenSearch Ingestion pipeline to process the data by reading from the source to writing the destination.
dynamodb-pipeline.opensearch.recordsIn.count	Count of records successfully ingested into OpenSearch. This metric is essential for tracking the volume of data being processed and stored.
dynamodb-pipeline.opensearch.s3.dlqS3RecordsFailed.count	Number of records that failed to write to DLQ.
dynamodb-pipeline.opensearch.s3.dlqS3RecordsSuccess.count	Number of records that are written to DLQ.
dynamodb-pipeline.opensearch.s3.dlqS3RequestLatency.count	Count of latency measurements for requests to the Amazon S3 dead-letter queue.
dynamodb-pipeline.opensearch.s3.dlqS3RequestLatency.sum	Total latency for all requests to the Amazon S3 dead-letter queue
dynamodb-pipeline.opensearch.s3.dlqS3RequestSizeBytes.sum	Total size in bytes of all requests made to the Amazon S3 dead-letter queue.

指標	Description
dynamodb-pipeline.recordsProcessed.count	Total number of records processed in the pipeline, a key metric for overal throughput.
dynamodb.changeEventsProcessed.count	No records are being gathered from DynamoDB streams. This could be due to no activitiy on the table, an export being in progress, or an issue accessing the DynamoDB streams.
dynamodb.exportJobFailure.count	The attempt to trigger an export to S3 failed.
dynamodb-pipeline.opensearch.bulkRequestInvalidInputErrors.count	Count of bulk request errors in OpenSearch due to invalid input, crucial for monitoring data quality and operational issues.
opensearch.EndToEndLatency.avg	The end to end latnecy is higher than desired for reading from DynamoDB streams. This could be due to an underscaled OpenSearch cluster or a maximum pipeline OCU capacity that is too low for the WCU throughput on the DynamoDB table. This end to end latency will be high after an export and should decrease over time as it catches up to the latest DynamoDB streams.

搭配 Amazon DocumentDB 使用 OpenSearch 擷取管道

您可以使用 DocumentDB 外掛程式將文件變更串流至 Amazon OpenSearch Service，例如建立、更新和刪除。管道支援變更資料擷取 (CDC)，如果可用，或 API 輪詢以進行大規模、低延遲串流。

您可以使用或不使用完整初始快照來處理資料。完整快照會擷取整個 Amazon DocumentDB 集合，並將其上傳至 Amazon S3。然後，管道會將資料傳送至一或多個 OpenSearch 索引。擷取快照後，管道會同步進行中的變更以維持一致性，並最終擷取近乎即時的更新。

如果您已經擁有來自其他來源的完整快照，或只需要處理新事件，則可以在沒有快照的情況下串流。在此情況下，管道會直接從 Amazon DocumentDB 變更串流讀取，而無需初始大量載入。

如果您啟用串流，則必須在 Amazon DocumentDB 集合上啟用變更串流。 Amazon DocumentDB 不過，如果您只執行完全載入或匯出，則不需要變更串流。

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 遵循《Amazon DocumentDB 開發人員指南》中的建立 Amazon DocumentDB 叢集中的步驟，建立具有讀取資料許可的 Amazon DocumentDB 叢集。 Amazon DocumentDB 如果您使用 CDC 基礎設施，請將 Amazon DocumentDB 叢集設定為發佈變更串流。
2. 在您的 Amazon DocumentDB 叢集上啟用 TLS。
3. 設定私有地址空間的 VPC CIDR，以便與 OpenSearch Ingestion 搭配使用。
4. 使用 在 Amazon DocumentDB 叢集上設定身分驗證 AWS Secrets Manager。依照 Amazon DocumentDB 自動輪換密碼中的步驟啟用秘密輪換。如需詳細資訊，請參閱 Amazon DocumentDB 中使用角色型存取控制和安全性進行資料庫存取。 Amazon DocumentDB
5. 如果您使用變更串流來訂閱 Amazon DocumentDB 集合上的資料變更，請使用 `change_stream_log_retention_duration` 參數將保留期間延長至最多 7 天，以避免資料遺失。根據預設，變更串流事件會在記錄事件後存放 3 小時，這不足以容納大型集合。若要修改變更串流保留期間，請參閱修改變更串流日誌保留期間。
6. 建立 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱the section called “建立 OpenSearch Service 網域”及the section called “建立集合”。
7. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從 Amazon DocumentDB 叢集寫入您的網域或集合。

   下列範例網域存取政策允許您在下一個步驟中建立的管道角色將資料寫入網域。請務必`resource`使用自己的 ARN 更新 。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
```

```
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-east-1:111122223333:domain/domain-name"
      ]
    }
  ]
}
```

若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱 the section called “設定角色和使用者”。

步驟 1：設定管道角色

在您設定 Amazon DocumentDB 管道先決條件之後，請設定您要在管道組態中使用的管道角色，並在角色中新增下列 Amazon DocumentDB 許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "allowS3ListObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::s3-bucket"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": "s3-prefix/*"
                }
            }
        },
        {
            "Sid": "allowReadAndWriteToS3ForExportStream",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::s3-bucket/s3-prefix/*"
            ]
        },
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-
name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:network-interface/*",
                "arn:aws:ec2:*:111122223333:subnet/*",
                "arn:aws:ec2:*:111122223333:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
```

```
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/OSISManaged": "true"
                }
            }
        }
    ]
}
```

您必須對您用來建立 OpenSearch Ingestion 管道的 IAM 角色提供上述 Amazon EC2 許可，因為管道使用這些許可來建立和刪除 VPC 中的網路介面。管道只能透過此網路界面存取 Amazon DocumentDB 叢集。

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 Amazon DocumentDB 作為來源。請注意，若要填入索引名稱，`getMetadata`函數會使用 *`documentdb_collection`*做為中繼資料金鑰。如果您想要在沒有 `getMetadata`方法的情況下使用不同的索引名稱，您可以使用組態 `index: "`*`my_index_name`*`"`。

```
version: "2"
documentdb-pipeline:
  source:
    documentdb:
      acknowledgments: true
      host: "https://docdb-cluster-id.us-east-1.docdb.amazonaws.com"
      port: 27017
      authentication:
        username: ${aws_secrets:secret:username}
        password: ${aws_secrets:secret:password}
      aws:
      s3_bucket: "bucket-name"
```

```
      s3_region: "bucket-region"
      s3_prefix: "path" #optional path for storing the temporary data
      collections:
        - collection: "dbname.collection"
          export: true
          stream: true
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      index: "${getMetadata(\"documentdb_collection\")}"
      index_type: custom
      document_id: "${getMetadata(\"primary_key\")}"
      action: "${getMetadata(\"opensearch_action\")}"
      document_version: "${getMetadata(\"document_version\")}"
      document_version_type: "external"
extension:
  aws:
    secrets:
      secret:
        secret_id: "my-docdb-secret"
        region: "us-east-1"
        refresh_interval: PT1H
```

您可以使用預先設定的 Amazon DocumentDB 藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

如果您使用 AWS 管理主控台 建立管道，也必須將管道連接至 VPC，才能使用 Amazon DocumentDB 做為來源。若要這樣做，請尋找來源網路選項區段，選取連接至 VPC 核取方塊，然後從其中一個提供的預設選項中選擇 CIDR。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。

若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 和 Amazon DocumentDB 之間的 IP 地址發生衝突，請確定 Amazon DocumentDB VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

如需詳細資訊，請參閱設定管道的 VPC 存取。

資料一致性

管道會持續輪詢或接收來自 Amazon DocumentDB 叢集的變更，並更新 OpenSearch 索引中的對應文件，以確保資料一致性。

OpenSearch Ingestion end-to-end確認，以確保資料耐久性。當管道讀取快照或串流時，它會動態建立分割區以進行平行處理。在擷取 OpenSearch 網域或集合中的所有記錄之後，管道收到確認時，會將分割區標記為完成。

如果您想要擷取至 OpenSearch Serverless 搜尋集合，您可以在管道中產生文件 ID。如果您想要擷取 OpenSearch Serverless 時間序列集合，請注意管道不會產生文件 ID，因此您必須在管道接收器組態`document_id: "${getMetadata(\"primary_key\")}"`中省略 。

OpenSearch 擷取管道也會將傳入的事件動作映射至對應的大量索引動作，以協助擷取文件。這可讓資料保持一致，以便 Amazon DocumentDB 中的每個資料變更都與 OpenSearch 中的對應文件變更進行協調。

映射資料類型

OpenSearch Service 會將每個傳入文件中的資料類型動態映射至 Amazon DocumentDB 中的對應資料類型。下表顯示 OpenSearch Service 如何自動映射各種資料類型。

資料類型	OpenSearch	Amazon DocumentDB
Integer	OpenSearch 會自動將 Amazon DocumentDB 整數值映射至 OpenSearch 整數。 OpenSearch 會根據第一個傳送的文件動態對應 欄位。如果您在 Amazon DocumentDB 中具有相同屬性的混合資料類型，則自動映射可能會失敗。 例如，如果您的第一個文件具有很長的屬性，而稍後的文件具有與整數相同的屬性，則 OpenSearch 無法擷取第二個文件。在這些情況下，您應該提供明確的映射範本，選擇最靈活的數字類型，如下所示： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float"`	Amazon DocumentDB 支援整數。

資料類型	OpenSearch	Amazon DocumentDB

	<pre>}
	}
	}
	}
	}</pre>	

Long	OpenSearch 會自動將 Amazon DocumentDB 長值映射至 OpenSearch 長。
	
	OpenSearch 會根據第一個傳送的文件動態對應 欄位。如果您在 Amazon DocumentDB 中具有相同屬性的混合資料類型，則自動映射可能會失敗。
	
	例如，如果您的第一個文件具有很長的屬性，而稍後的文件具有與整數相同的屬性，則 OpenSearch 無法擷取第二個文件。在這些情況下，您應該提供明確的映射範本，選擇最靈活的數字類型，如下所示：
	
	<pre>{
	"template": {
	"mappings": {
	"properties": {
	"MixedNumberField": {
	"type": "float"
	}
	}
	}
	}
	}</pre>
		Amazon DocumentDB 支援長時間。

資料類型	OpenSearch	Amazon DocumentDB
String	OpenSearch 會自動將字串值映射為文字。在某些情況下，例如列舉值，您可以映射到關鍵字類型。 下列範例示範如何將名為 的 Amazon DocumentDB 屬性映射`PartType`至 OpenSearch 關鍵字。 `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	Amazon DocumentDB 支援字串。

資料類型	OpenSearch	Amazon DocumentDB
Double	OpenSearch 會自動將 Amazon DocumentDB 雙值映射至 OpenSearch 雙值。 OpenSearch 會根據第一個傳送的文件動態對應 欄位。如果您在 Amazon DocumentDB 中具有相同屬性的混合資料類型，則自動映射可能會失敗。 例如，如果您的第一個文件具有很長的屬性，而稍後的文件具有與整數相同的屬性，則 OpenSearch 無法擷取第二個文件。在這些情況下，您應該提供明確的映射範本，選擇最靈活的數字類型，如下所示： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB 支援雙工。

資料類型	OpenSearch	Amazon DocumentDB
Date	根據預設，日期會映射到 OpenSearch 中的整數。您可以定義自訂映射範本，將日期映射至 OpenSearch 日期。 `{ "template": { "mappings": { "properties": { "myDateField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB 支援日期。
時間戳記	根據預設，時間戳記會映射至 OpenSearch 中的整數。您可以定義自訂映射範本，將日期映射至 OpenSearch 日期。 `{ "template": { "mappings": { "properties": { "myTimestampField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB 支援時間戳記。
Boolean	OpenSearch 會將 Amazon DocumentDB 布林類型映射至 OpenSearch 布林類型。	Amazon DocumentDB 支援布林值類型屬性。

資料類型	OpenSearch	Amazon DocumentDB
Decimal (小數)	OpenSearch 會將 Amazon DocumentDB 映射屬性映射至巢狀欄位。相同的映射適用於巢狀欄位。 下列範例會將巢狀欄位中的字串映射至 OpenSearch 中的關鍵字類型： `{ "template": { "mappings": { "properties": { "myDecimalField": { "type": "double" } } } } }` 使用此自訂映射，您可以使用雙階精確度查詢和彙總 欄位。原始值會在 OpenSearch 文件的 `_source` 屬性中保留完整精確度。如果沒有此映射，OpenSearch 預設會使用文字。	Amazon DocumentDB 支援小數。
規則表達式	regex 類型會建立巢狀欄位。這些包括 *<myFieldName>* .pattern和 *<myFieldName>* .options。	Amazon DocumentDB 支援規則表達式。

資料類型	OpenSearch	Amazon DocumentDB
二進位資料	OpenSearch 會自動將 Amazon DocumentDB 二進位資料映射至 OpenSearch 文字。您可以提供映射，將這些項目寫入 OpenSearch 中的二進位欄位。 下列範例示範如何將名為 的 Amazon DocumentDB 欄位映射`imageData` 至 OpenSearch 二進位欄位。 `{ "template": { "mappings": { "properties": { "imageData": { "type": "binary" } } } } }`	Amazon DocumentDB 支援二進位資料欄位。
ObjectId	具有類型 objectId 的欄位會映射至 OpenSearch 文字欄位。值將是 objectId 的字串表示法。	Amazon DocumentDB 支援 objectIds。
Null	OpenSearch 可以使用 Amazon DocumentDB null 類型擷取文件。它會將值儲存為文件中的 null 值。沒有此類型的映射，且此欄位無法編製索引或搜尋。 如果相同屬性名稱用於 null 類型，然後變更為不同類型，例如字串，則 OpenSearch 會為第一個非 Null 值建立動態映射。後續值仍然可以是 Amazon DocumentDB null 值。	Amazon DocumentDB 支援 null 類型欄位。

資料類型	OpenSearch	Amazon DocumentDB
未定義	OpenSearch 可以擷取 Amazon DocumentDB 未定義類型的文件。它會將值儲存為文件中的 null 值。沒有此類型的映射，且此欄位無法編製索引或搜尋。 如果未定義類型使用相同的欄位名稱，然後變更為不同的類型，例如字串，則 OpenSearch 會為第一個未定義值建立動態映射。後續值仍然可以是 Amazon DocumentDB 未定義的值。	Amazon DocumentDB 支援未定義的類型欄位。
MinKey	OpenSearch 可以使用 Amazon DocumentDB minKey 類型擷取文件。它會將值儲存為文件中的 null 值。沒有此類型的映射，且此欄位無法編製索引或搜尋。 如果 minKey 類型使用相同的欄位名稱，然後變更為不同的類型，例如字串，則 OpenSearch 會為第一個非 minKey 值建立動態映射。後續值仍然可以是 Amazon DocumentDB minKey 值。	Amazon DocumentDB 支援 minKey 類型欄位。
MaxKey	OpenSearch 可以使用 Amazon DocumentDB maxKey 類型擷取文件。它會將值儲存為文件中的 null 值。沒有此類型的映射，且此欄位無法編製索引或搜尋。 如果 maxKey 類型使用相同的欄位名稱，然後變更為不同的類型，例如字串，則 OpenSearch 會為第一個非maxKey 值建立動態映射。後續值仍然可以是 Amazon DocumentDB maxKey 值。	Amazon DocumentDB 支援 maxKey 類型欄位。

建議您在 OpenSearch Ingestion 管道中設定無效字母佇列 (DLQ)。如果您已設定佇列，OpenSearch Service 會將因動態映射失敗而無法擷取的所有失敗文件傳送至佇列。

如果自動映射失敗，您可以在管道組態`template_content`中使用 `template_type`和 來定義明確的映射規則。或者，您可以在啟動管道之前，直接在搜尋網域或集合中建立映射範本。

限制

當您為 Amazon DocumentDB 設定 OpenSearch 擷取管道時，請考慮下列限制：

- OpenSearch Ingestion 與 Amazon DocumentDB 整合目前不支援跨區域擷取。您的 Amazon DocumentDB 叢集和 OpenSearch Ingestion 管道必須位於相同的 中 AWS 區域。
- OpenSearch Ingestion 與 Amazon DocumentDB 整合目前不支援跨帳戶擷取。您的 Amazon DocumentDB 叢集和 OpenSearch Ingestion 管道必須位於相同的 中 AWS 帳戶。
- OpenSearch Ingestion 管道僅支援一個 Amazon DocumentDB 叢集做為其來源。
- OpenSearch Ingestion 與 Amazon DocumentDB 整合特別支援以 Amazon DocumentDB 執行個體為基礎的叢集。它不支援 Amazon DocumentDB 彈性叢集。
- OpenSearch Ingestion 整合僅支援 AWS Secrets Manager 做為 Amazon DocumentDB 叢集的身分驗證機制。
- 您無法更新現有的管道組態，以從不同的資料庫或集合擷取資料。您必須改為建立新的管道。

建議 CloudWatch 警示

為了獲得最佳效能，我們建議您在建立 OpenSearch Ingestion 管道以存取 Amazon DocumentDB 叢集做為來源時，使用下列 CloudWatch 警示。

CloudWatch 警示	Description
<pipeline-name> .doucmentdb.credentialsChanged	此指標表示 AWS 秘密輪換的頻率。
<pipeline-name> .doucmentdb.executorRefreshErrors	此指標表示重新整理 AWS 秘密失敗。
<pipeline-name> .doucmentdb.exportRecordsTotal	此指標表示從 Amazon DocumentDB 匯出的記錄數。

CloudWatch 警示	Description
<pipeline-name> .doucmentdb.exportRecordsProcessed	此指標表示 OpenSearch Ingestion 管道處理的記錄數目。
<pipeline-name> .doucmentdb.exportRecordProcessingErrors	此指標表示從 Amazon DocumentDB 叢集讀取資料時OpenSearch 擷取管道中的處理錯誤數目。
<pipeline-name> .doucmentdb.exportRecordsSuccessTotal	此指標表示成功處理的匯出記錄總數。
<pipeline-name> .doucmentdb.exportRecordsFailedTotal	此指標表示無法處理的匯出記錄總數。
<pipeline-name> .doucmentdb.bytesReceived	此指標表示 OpenSearch Ingestion 管道收到的位元組總數。
<pipeline-name> .doucmentdb.bytesProcessed	此指標表示 OpenSearch Ingestion 管道處理的位元組總數。
<pipeline-name> .doucmentdb.exportPartitionQueryTotal	此指標表示匯出分割區總計。
<pipeline-name> .doucmentdb.streamRecordsSuccessTotal	此指標表示從串流成功處理的記錄數量。
<pipeline-name> .doucmentdb.streamRecordsFailedTotal	此指標表示無法從串流處理的記錄總數。

搭配 Confluent Cloud Kafka 使用 OpenSearch 擷取管道

您可以使用 OpenSearch Ingestion 管道，將資料從 Confluent Cloud Kafka 叢集串流至 Amazon OpenSearch Service 網域和 OpenSearch Serverless 集合。OpenSearch Ingestion 支援公有和私有網路組態，可將資料從 Confluent Cloud Kafka 叢集串流至 OpenSearch Service 或 OpenSearch Serverless 管理的網域或集合。

Confluent Cloud 公有 Kafka 叢集的連線能力

您可以使用 OpenSearch Ingestion 管道從具有公有組態的 Confluent Cloud Kafka 叢集遷移資料，這表示網域 DNS 名稱可以公開解析。若要這樣做，請使用 Confluent Cloud 公有 Kafka 叢集做為來源，以及 OpenSearch 設定 OpenSearch Service 或 OpenSearch Serverless 做為目的地。這會處理從自我管理來源叢集到受 AWS管目的地網域或集合的串流資料。

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 建立充當來源的 Confluent Cloud Kafka 叢集。叢集應包含您要擷取至 OpenSearch Service 的資料。
2. 建立您要將資料遷移至其中的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱the section called “建立 OpenSearch Service 網域”及the section called “建立集合”。
3. 使用 在 Confluent Cloud Kafka 叢集上設定身分驗證 AWS Secrets Manager。依照輪換秘密中的步驟啟用AWS Secrets Manager 秘密輪換。
4. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從自我管理的叢集寫入您的網域或集合。

   下列範例網域存取政策允許您在下一個步驟中建立的管道角色將資料寫入網域。請務必`resource`使用自己的 ARN 更新 。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-east-1:111122223333:domain/domain-name"
      ]
```

```
    }
  ]
}
```

若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱 the section called “設定角色和使用者”。

步驟 1：設定管道角色

設定 Confluent Cloud Kafka 叢集管道先決條件之後，請設定管道組態中要使用的管道角色，並新增寫入 OpenSearch Service 網域或 OpenSearch Serverless 集合的許可，以及從 Secrets Manager 讀取秘密的許可。

需要下列許可才能管理網路介面：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
                "arn:aws:ec2:us-east-1:111122223333:subnet/*",
                "arn:aws:ec2:us-east-1:111122223333:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
```

```
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:subnet/*"
        },
        {
            "Effect": "Allow",
            "Action": [ "ec2:CreateTags" ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:network-interface/*",
            "Condition": {
               "StringEquals": { "aws:RequestTag/OSISManaged": "true" }
            }
        }
    ]
}
```

以下是從 AWS Secrets Manager 服務讀取秘密所需的許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": ["secretsmanager:GetSecretValue"],
            "Resource": ["arn:aws:secretsmanager:us-
east-1:111122223333:secret:,secret-name"]
        }
    ]
}
```

需要下列許可才能寫入 Amazon OpenSearch Service 網域：

```
{
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-id:role/pipeline-role"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，將 Confluent Cloud Kafka 指定為來源。

您可以指定多個 OpenSearch Service 網域做為資料的目的地。此功能可將傳入資料的條件式路由或複寫到多個 OpenSearch Service 網域。

您也可以將資料從來源 Confluent Kafka 叢集遷移至 OpenSearch Serverless VPC 集合。請確定您在管道組態中提供網路存取政策。您可以使用 Confluent 結構描述登錄檔來定義 Confluent 結構描述。

```
version: "2"
kafka-pipeline:
  source:
    kafka:
      encryption:
        type: "ssl"
      topics:
        - name: "topic-name"
          group_id: "group-id"
      bootstrap_servers:
        - "bootstrap-server.us-east-1.aws.private.confluent.cloud:9092"
      authentication:
        sasl:
          plain:
            username: ${aws_secrets:confluent-kafka-secret:username}
            password: ${aws_secrets:confluent-kafka-secret:password}
      schema:
        type: confluent
        registry_url: https://my-registry.us-east-1.aws.confluent.cloud
        api_key: "${{aws_secrets:schema-secret:schema_registry_api_key}}"
        api_secret: "${{aws_secrets:schema-secret:schema_registry_api_secret}}"
        basic_auth_credentials_source: "USER_INFO"
```

```
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
  aws:
    secrets:
      confluent-kafka-secret:
        secret_id: "my-kafka-secret"
        region: "us-east-1"
      schema-secret:
        secret_id: "my-self-managed-kafka-schema"
        region: "us-east-1"
```

您可以使用預先設定的藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

在 VPC 中連線至 Confluent Cloud Kafka 叢集

您也可以使用 OpenSearch Ingestion 管道，從 VPC 中執行的 Confluent Cloud Kafka 叢集遷移資料。若要這麼做，請設定 OpenSearch Ingestion 管道，並將 Confluent Cloud Kafka 叢集做為來源，並將 OpenSearch Service 或 OpenSearch Serverless 做為目的地。這會處理從 Confluent Cloud Kafka 來源叢集到 AWS受管目的地網域或集合的串流資料。

OpenSearch Ingestion 支援 Confluent 中所有支援的網路模式中設定的 Confluent Cloud Kafka 叢集。OpenSearch Ingestion 中支援下列網路組態模式做為來源：

- AWS VPC 對等互連
- AWS PrivateLink 專用叢集的
- AWS PrivateLink for Enterprise 叢集
- AWS Transit Gateway

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 使用包含您要擷取至 OpenSearch Service 之資料的 VPC 網路組態建立 Confluent Cloud Kafka 叢集。
2. 建立您要將資料遷移至其中的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱 和 the section called “建立 OpenSearch Service 網域” the section called “建立集合”。

3. 使用 在 Confluent Cloud Kafka 叢集上設定身分驗證 AWS Secrets Manager。依照輪換秘密中的步驟啟用AWS Secrets Manager 秘密輪換。
4. 取得可存取 Confluent Cloud Kafka 叢集的 VPC ID。選擇要由 OpenSearch Ingestion 使用的 VPC CIDR。

> **Note**
>
> 如果您使用 AWS 管理主控台 建立管道，您還必須將 OpenSearch Ingestion 管道連接至 VPC，才能使用 Confluent Cloud Kafka 叢集。若要這樣做，請尋找網路組態區段，選取連接至 VPC 核取方塊，然後從其中一個提供的預設選項中選擇 CIDR，或選取您自己的選項。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。
> 若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 與自我管理 OpenSearch 之間的 IP 地址發生衝突，請確定自我管理的 OpenSearch VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

5. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從自我管理的叢集寫入您的網域或集合。

> **Note**
>
> 如果您使用 AWS PrivateLink 來連接 Confluent Cloud Kafka，則需要設定 VPC DHCP 選項。應啟用 DNS 主機名稱和 DNS 解析。
> 具體而言，請使用下列選項集值：
> 企業叢集：
>
> ```
> domain-name: aws.private.confluent.cloud
> domain-name-servers: AmazonProvidedDNS
> ```
>
> 專用叢集：
>
> ```
> domain-name: aws.confluent.cloud
> domain-name-servers: AmazonProvidedDNS
> ```
>
> 此變更可確保 Confluent PrivateLink 端點的 DNS 解析在 VPC 內正常運作。

下列範例網域存取政策允許您在下一個步驟中建立的管道角色將資料寫入網域。請務必`resource`使用自己的 ARN 更新 。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-east-1:111122223333:domain/domain-name"
      ]
    }
  ]
}
```

若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱 the section called “設定角色和使用者”。

步驟 1：設定管道角色

在您設定管道先決條件之後，請設定您要在管道組態中使用的管道角色，並在角色中新增下列許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": ["arn:aws:secretsmanager:us-
east-1:111122223333:secret:secret-name"]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals":
                    {
                        "aws:RequestTag/OSISManaged": "true"
                    }
            }
        }
    ]
```

```
}
```

您必須對您用來建立 OpenSearch Ingestion 管道的 IAM 角色提供上述 Amazon EC2 許可，因為管道使用這些許可來建立和刪除 VPC 中的網路介面。管道只能透過此網路界面存取 Kafka 叢集。

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 Kafka 作為來源。

您可以指定多個 OpenSearch Service 網域做為資料的目的地。此功能可將傳入資料的條件式路由或複寫到多個 OpenSearch Service 網域。

您也可以將資料從來源 Confluent Kafka 叢集遷移至 OpenSearch Serverless VPC 集合。請確定您在管道組態中提供網路存取政策。您可以使用 Confluent 結構描述登錄檔來定義 Confluent 結構描述。

```
 version: "2"
kafka-pipeline:
  source:
    kafka:
      encryption:
        type: "ssl"
      topics:
        - name: "topic-name"
          group_id: "group-id"
      bootstrap_servers:
        - "bootstrap-server.us-east-1.aws.private.confluent.cloud:9092"
      authentication:
        sasl:
          plain:
            username: ${aws_secrets:confluent-kafka-secret:username}
            password: ${aws_secrets:confluent-kafka-secret:password}
      schema:
        type: confluent
        registry_url: https://my-registry.us-east-1.aws.confluent.cloud
        api_key: "${{aws_secrets:schema-secret:schema_registry_api_key}}"
        api_secret: "${{aws_secrets:schema-secret:schema_registry_api_secret}}"
        basic_auth_credentials_source: "USER_INFO"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
```

```
      index: "confluent-index"
extension:
  aws:
    secrets:
      confluent-kafka-secret:
        secret_id: "my-kafka-secret"
        region: "us-east-1"
      schema-secret:
        secret_id: "my-self-managed-kafka-schema"
        region: "us-east-2"
```

搭配 使用 OpenSearch 擷取管道 Amazon Managed Streaming for Apache Kafka

您可以使用 Kafka 外掛程式，將資料從 Amazon Managed Streaming for Apache Kafka (Amazon MSK) 擷取到您的 OpenSearch Ingestion 管道。透過 Amazon MSK，您可以建置和執行使用 Apache Kafka 來處理串流資料的應用程式。OpenSearch Ingestion 使用 AWS PrivateLink 連線到 Amazon MSK。您可以從 Amazon MSK 和 Amazon MSK Serverless 叢集擷取資料。這兩個程序的唯一區別是設定管道之前必須採取的先決條件。

主題

- 佈建的 Amazon MSK 先決條件
- Amazon MSK Serverless 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道
- 步驟 3：（選用）使用 AWS Glue 結構描述登錄檔
- 步驟 4：（選用）為 Amazon MSK 管道設定建議的運算單位 (OCUs)

佈建的 Amazon MSK 先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 按照 Amazon Managed Streaming for Apache Kafka 開發人員指南中的建立叢集中的步驟建立 Amazon MSK 佈建叢集。 對於中介裝置類型，請選擇`t3`類型以外的任何選項，因為 OpenSearch Ingestion 不支援這些選項。
2. 叢集處於作用中狀態後，請遵循開啟多 VPC 連線中的步驟。

3. 根據您的叢集和管道是否位於相同位置，遵循將叢集政策連接至 MSK 叢集中的步驟，以連接下列其中一個政策 AWS 帳戶。此政策允許 OpenSearch Ingestion 建立 Amazon MSK 叢集的 AWS PrivateLink 連線，並從 Kafka 主題讀取資料。請務必`resource`使用自己的 ARN 更新 。

   當您的叢集和管道位於相同的 時，適用下列政策 AWS 帳戶：

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-
name/cluster-id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-
name/cluster-id"
    }
  ]
}
```

   如果您的 Amazon MSK 叢集與管道 AWS 帳戶 位於不同的 中，請改為連接下列政策。請注意，跨帳戶存取僅適用於佈建的 Amazon MSK 叢集，而不適用於 Amazon MSK Serverless 叢集。的 AWS `principal` ARN 應該是您提供給管道組態的相同管道角色的 ARN：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-
name/cluster-id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-
name/cluster-id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
      },
      "Action": [
        "kafka-cluster:*",
        "kafka:*"
      ],
      "Resource": [
        "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-name/cluster-id",
        "arn:aws:kafka:us-east-1:111122223333:topic/cluster-name/cluster-id/*",
        "arn:aws:kafka:us-east-1:111122223333:group/cluster-name/*"
```

```
      ]
    }
  ]
}
```

4. 依照建立主題中的步驟建立 Kafka 主題。確定 *BootstrapServerString*是其中一個私有端點（單一 VPC) 引導 URLs。根據 Amazon MSK 叢集擁有的區域數量3，的值`--replication-factor`應為 2或 。的值`--partitions`應至少為 10。
5. 遵循生產和使用資料中的步驟來生產和使用資料。同樣地，請確定 *BootstrapServerString*是您的私有端點（單一 VPC) 引導 URLs之一。

Amazon MSK Serverless 先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 按照 Amazon Managed Streaming for Apache Kafka 開發人員指南中的建立 MSK Serverless 叢集中的步驟建立 Amazon MSK Serverless 叢集。
2. 叢集處於作用中狀態後，請遵循將叢集政策連接至 MSK 叢集中的步驟，以連接下列政策。請務必`resource`使用自己的 ARN 更新 。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-name/cluster-id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
```

```
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-
name/cluster-id"
    }
  ]
}
```

此政策允許 OpenSearch Ingestion 建立與 Amazon MSK Serverless 叢集的 AWS PrivateLink 連線，並從 Kafka 主題讀取資料。當您的叢集和管道位於相同的 時，此政策即適用 AWS 帳戶，因為 Amazon MSK Serverless 不支援跨帳戶存取，因此此政策必須是 true。

3. 依照建立主題中的步驟建立 Kafka 主題。請確定 *BootstrapServerString*是您的簡易身分驗證和安全層 (SASL) IAM 引導 URLs之一。根據 Amazon MSK Serverless 叢集擁有的區域數量3， 的值`--replication-factor`應為 2或 。的值`--partitions`應至少為 10。
4. 遵循生產和使用資料中的步驟來生產和使用資料。同樣地，請確定 *BootstrapServerString*是您的簡易身分驗證和安全層 (SASL) IAM 引導 URLs之一。

步驟 1：設定管道角色

在您設定 Amazon MSK 提供或無伺服器叢集之後，請在管道組態中要使用的管道角色中新增下列 Kafka 許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster",
                "kafka:DescribeClusterV2",
                "kafka:GetBootstrapBrokers"
```

```
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:111122223333:cluster/cluster-
name/cluster-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:111122223333:topic/cluster-name/cluster-
id/topic-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:111122223333:group/cluster-name/*"
            ]
        }
    ]
}
```

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，將 Kafka 指定為來源：

```
version: "2"
log-pipeline:
  source:
    kafka:
      acknowledgements: true
      topics:
      - name: "topic-name"
        group_id: "grouplambd-id"
```

```
      aws:
        msk:
          arn: "arn:aws:kafka:region:account-id:cluster/cluster-name/cluster-id"
        region: "us-west-2"
  processor:
  - grok:
      match:
        message:
        - "%{COMMONAPACHELOG}"
  - date:
      destination: "@timestamp"
      from_time_received: true
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
      index: "index_name"
      aws_region: "region"
      aws_sigv4: true
```

您可以使用預先設定的 Amazon MSK 藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

步驟 3：（選用）使用 AWS Glue 結構描述登錄檔

當您將 OpenSearch Ingestion 與 Amazon MSK 搭配使用時，您可以將 AVRO 資料格式用於 AWS Glue 結構描述登錄檔中託管的結構描述。使用AWS Glue 結構描述登錄檔，您可以集中探索、控制和發展資料串流結構描述。

若要使用此選項，請在管道組態type中啟用結構描述：

```
schema:
  type: "aws_glue"
```

您還必須在管道角色中提供 AWS Glue 讀取存取許可。您可以使用稱為 AWSGlueSchemaRegistryReadonlyAccess 的 AWS 受管政策。此外，您的登錄檔必須與 OpenSearch Ingestion 管道位於相同的 AWS 帳戶 和 區域。

步驟 4：（選用）為 Amazon MSK 管道設定建議的運算單位 (OCUs)

每個運算單位每個主題都有一個取用者。中介裝置在指定主題的這些取用者之間平衡分割區。不過，當分割區數量大於取用者數量時，Amazon MSK 會在每個取用者上託管多個分割區。OpenSearch Ingestion 具有內建的自動擴展功能，可根據 CPU 用量或管道中待定記錄的數量來擴展或縮減規模。

為了獲得最佳效能，請將分割區分散到許多運算單位以進行平行處理。如果主題具有大量分割區（例如，超過 96 個，也就是每個管道的最大 OCUs)，建議您使用 1–96 OCUs 設定管道。這是因為它將視需要自動擴展。如果主題具有少量分割區（例如，小於 96)，請保持最大運算單位與分割區數量相同。

當管道有多個主題時，請選擇分割區數量最高的主題做為設定最大運算單位的參考。透過將具有一組新OCUs的另一個管道新增至相同的主題和取用者群組，您可以幾乎線性地擴展輸送量。

搭配 Amazon RDS 使用 OpenSearch Ingestion 管道

您可以使用 OpenSearch Ingestion 管道搭配 Amazon RDS，將現有的資料和串流變更（例如建立、更新和刪除）匯出至 Amazon OpenSearch Service 網域和集合。OpenSearch Ingestion 管道整合了變更資料擷取 (CDC) 基礎設施，以提供從 Amazon RDS 持續串流資料的高規模、低延遲方式。支援 RDS for MySQL 和 RDS for PostgreSQL。

有兩種方式可以使用 Amazon RDS 作為來源來處理資料，無論是否有完整的初始快照。完整初始快照是指定資料表的快照，此快照會匯出至 Amazon S3。從那裡，OpenSearch Ingestion 管道會將其傳送至網域中的一個索引，或將其分割至網域中的多個索引。為了讓 Amazon RDS 和 OpenSearch 中的資料保持一致，管道會將 Amazon RDS 執行個體中資料表中的所有建立、更新和刪除事件與儲存在 OpenSearch 索引或索引中的文件同步。

當您使用完整初始快照時，您的 OpenSearch 擷取管道會先擷取快照，然後開始從 Amazon RDS 變更串流讀取資料。它最終會追上並維持 Amazon RDS 和 OpenSearch 之間的近乎即時的資料一致性。

您也可以使用 OpenSearch Ingestion 與 Amazon RDS 整合來追蹤變更資料擷取，並將 Aurora 中的所有更新擷取至 OpenSearch。如果您已經擁有其他機制的完整快照，或者只想要擷取 Amazon RDS 執行個體中資料的所有變更，請選擇此選項。

選擇此選項時，您需要設定 Amazon RDS for MySQL 二進位記錄或設定 Amazon RDS for PostgresSQL 資料庫執行個體的邏輯複寫。

主題

- RDS for MySQL
- RDS for PostgreSQL

RDS for MySQL

完成下列步驟，以使用 Amazon RDS for RDS for MySQL 設定 OpenSearch 擷取管道。

主題

- RDS for MySQL 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道
- 資料一致性
- 映射資料類型
- 限制
- 建議的 CloudWatch 警示

RDS for MySQL 先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 在 Amazon RDS 中建立自訂資料庫參數群組，以設定二進位記錄並設定下列參數。

   ```
   binlog_format=ROW
   binlog_row_image=full
   binlog_row_metadata=FULL
   ```

   此外，請確定 `binlog_row_value_options` 參數未設定為 `PARTIAL_JSON`。

   如需詳細資訊，請參閱設定 RDS for MySQL 二進位記錄。
2. 選取或建立 RDS for MySQL 資料庫執行個體，並將上一個步驟中建立的參數群組與資料庫執行個體建立關聯。
3. 確認資料庫上已啟用自動備份。如需詳細資訊，請參閱啟用自動備份。
4. 設定具有足夠時間進行複寫的二進位日誌保留，例如 24 小時。如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的設定和顯示二進位日誌組態。
5. 使用 Amazon RDS 和 的密碼管理，在您的 Amazon RDS AWS Secrets Manager執行個體上設定使用者名稱和密碼身分驗證。您也可以建立 Secrets Manager 秘密來建立使用者名稱/密碼組合。
6. 如果您使用完整的初始快照功能，請建立 AWS KMS key 和 IAM 角色，以將資料從 Amazon RDS 匯出至 Amazon S3。

   IAM 角色應具有下列許可政策：

   JSON

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

```
    "Statement": [
        {
            "Sid": "ExportPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::s3-bucket-used-in-pipeline",
                "arn:aws:s3:::s3-bucket-used-in-pipeline/*"
            ]
        }
    ]
}
```

此角色也應該具有下列信任關係：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "export.rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

7. 選取或建立 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱建立 OpenSearch Service 網域和建立集合。
8. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從 Amazon RDS 資料庫執行個體寫入您的網域或集合。

步驟 1：設定管道角色

設定 Amazon RDS 管道先決條件之後，請將管道角色設定為在管道組態中使用。同時將 Amazon RDS 來源的下列許可新增至角色：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
    "Sid": "allowReadingFromS3Buckets",
    "Effect": "Allow",
    "Action": [
    "s3:GetObject",
    "s3:DeleteObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:PutObject"
    ],
    "Resource": [
    "arn:aws:s3:::s3_bucket",
    "arn:aws:s3:::s3_bucket/*"
    ]
    },
    {
    "Sid": "allowNetworkInterfacesActions",
    "Effect": "Allow",
    "Action": [
    "ec2:AttachNetworkInterface",
    "ec2:CreateNetworkInterface",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:DetachNetworkInterface",
    "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": [
    "arn:aws:ec2:*:111122223333:network-interface/*",
    "arn:aws:ec2:*:111122223333:subnet/*",
    "arn:aws:ec2:*:111122223333:security-group/*"
    ]
    },
```

```
    {
    "Sid": "allowDescribeEC2",
    "Effect": "Allow",
    "Action": [
    "ec2:Describe*"
    ],
    "Resource": "*"
    },
    {
    "Sid": "allowTagCreation",
    "Effect": "Allow",
    "Action": [
    "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:111122223333:network-interface/*",
    "Condition": {
    "StringEquals": {
    "aws:RequestTag/OSISManaged": "true"
    }
    }
    },
    {
    "Sid": "AllowDescribeInstances",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBInstances"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:db:*"
    ]
    },
    {
    "Sid": "AllowSnapshots",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBSnapshots",
    "rds:CreateDBSnapshot",
    "rds:AddTagsToResource"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:db:DB-id",
    "arn:aws:rds:us-east-2:111122223333:snapshot:DB-id*"
    ]
    },
```

```
    {
    "Sid": "AllowExport",
    "Effect": "Allow",
    "Action": [
    "rds:StartExportTask"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:snapshot:DB-id*"
    ]
    },
    {
    "Sid": "AllowDescribeExports",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeExportTasks"
    ],
    "Resource": "*",
    "Condition": {
    "StringEquals": {
    "aws:RequestedRegion": "us-east-2",
    "aws:ResourceAccount": "111122223333"
    }
    }
    },
    {
    "Sid": "AllowAccessToKmsForExport",
    "Effect": "Allow",
    "Action": [
    "kms:Decrypt",
    "kms:Encrypt",
    "kms:DescribeKey",
    "kms:RetireGrant",
    "kms:CreateGrant",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
    ],
    "Resource": [
    "arn:aws:kms:us-east-2:111122223333:key/export-key-id"
    ]
    },
    {
    "Sid": "AllowPassingExportRole",
    "Effect": "Allow",
    "Action": "iam:PassRole",
```

```
    "Resource": [
    "arn:aws:iam::111122223333:role/export-role"
    ]
    },
    {
    "Sid": "SecretsManagerReadAccess",
    "Effect": "Allow",
    "Action": [
    "secretsmanager:GetSecretValue"
    ],
    "Resource": [
    "arn:aws:secretsmanager:*:111122223333:secret:*"
    ]
    }
    ]
    }
```

步驟 2：建立管道

設定類似下列的 OpenSearch Ingestion 管道。範例管道指定 Amazon RDS 執行個體做為來源。

```
version: "2"
rds-mysql-pipeline:
  source:
    rds:
      db_identifier: "instance-id"
      engine: mysql
      database: "database-name"
      tables:
        include:
          - "table1"
          - "table2"
      s3_bucket: "bucket-name"
      s3_region: "bucket-region"
      s3_prefix: "prefix-name"
      export:
        kms_key_id: "kms-key-id"
        iam_role_arn: "export-role-arn"
      stream: true
      aws:
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        region: "us-east-1"
      authentication:
```

```
        username: ${{aws_secrets:secret:username}}
        password: ${{aws_secrets:secret:password}}
  sink:
    - opensearch:
        hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
        index: "${getMetadata(\"table_name\")}"
        index_type: custom
        document_id: "${getMetadata(\"primary_key\")}"
        action: "${getMetadata(\"opensearch_action\")}"
        document_version: "${getMetadata(\"document_version\")}"
        document_version_type: "external"
        aws:
          sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
          region: "us-east-1"
extension:
  aws:
    secrets:
      secret:
        secret_id: "rds-secret-id"
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        refresh_interval: PT1H
```

您可以使用預先設定的 Amazon RDS 藍圖來建立此管道。如需詳細資訊，請參閱使用藍圖。

若要使用 Amazon Aurora 做為來源，您需要設定管道的 VPC 存取。您選擇的 VPC 應與 Amazon Aurora 來源使用的 VPC 相同。然後選擇一或多個子網路和一或多個 VPC 安全群組。請注意，管道需要網路存取 Aurora MySQL 資料庫，因此您也應該確認您的 Aurora 叢集已設定 VPC 安全群組，允許從管道的 VPC 安全群組到資料庫連接埠的傳入流量。如需詳細資訊，請參閱使用安全群組控制存取。

如果您使用 AWS 管理主控台 建立管道，也必須將管道連接至 VPC，才能使用 Amazon Aurora 做為來源。若要這樣做，請尋找網路組態區段，選擇連接至 VPC，然後從其中一個提供的預設選項中選擇 CIDR，或選取您自己的選項。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。

若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 和 Amazon RDS 之間的 IP 地址發生衝突，請確定 Amazon RDS VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

如需詳細資訊，請參閱設定管道的 VPC 存取。

資料一致性

管道會持續輪詢或接收來自 Amazon RDS 執行個體的變更，並更新 OpenSearch 索引中的對應文件，以確保資料一致性。

OpenSearch Ingestion end-to-end確認，以確保資料耐久性。當管道讀取快照或串流時，它會動態建立分割區以進行平行處理。在擷取 OpenSearch 網域或集合中的所有記錄之後，管道收到確認時，會將分割區標記為完成。如果您想要擷取到 OpenSearch Serverless 搜尋集合，您可以在管道中產生文件 ID。如果您想要擷取 OpenSearch Serverless 時間序列集合，請注意管道不會產生文件 ID，因此您必須在管道接收器組態`document_id: "${getMetadata(\"primary_key\")}"`中省略 。

OpenSearch 擷取管道也會將傳入的事件動作映射至對應的大量索引動作，以協助擷取文件。這可讓資料保持一致，以便 Amazon RDS 中的每個資料變更都與 OpenSearch 中的對應文件變更進行協調。

映射資料類型

OpenSearch Ingestion 管道會將 MySQL 資料類型映射至適合 OpenSearch Service 網域或集合使用的表示法。如果 OpenSearch 中未定義映射範本，則 OpenSearch 會根據第一個傳送的文件自動判斷具有動態映射的欄位類型。您也可以透過映射範本，明確定義最適合您在 OpenSearch 中的欄位類型。

下表列出 MySQL 資料類型和對應的 OpenSearch 欄位類型。如果未定義明確的映射，預設 OpenSearch 欄位類型欄會顯示 OpenSearch 中的對應欄位類型。在此情況下，OpenSearch 會自動判斷具有動態映射的欄位類型。建議的 OpenSearch 欄位類型欄是建議在映射範本中明確指定的對應欄位類型。這些欄位類型與 MySQL 中的資料類型更緊密一致，通常可以在 OpenSearch 中啟用更好的搜尋功能。

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
BIGINT	long	long
BIGINT UNSIGNED	long	未簽署的長
BIT	long	位元組、短、整數或長，取決於位元數
DECIMAL	text	雙 或 關鍵字
DOUBLE	float	double
FLOAT	float	float

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
INT	long	integer
INT UNSIGNED	long	long
MEDIUMIN	long	integer
MEDIUMIN UNSIGNED	long	integer
NUMERIC	text	雙 或 關鍵字
SMALLINT	long	short
SMALLINT UNSIGNED	long	integer
TINYINT	long	byte
TINYINT UNSIGNED	long	short
BINARY	text	binary
BLOB	text	binary
CHAR	text	text
ENUM	text	關鍵字
LONGBLOE	text	binary
LONGTEXT	text	text
MEDIUMBL B	text	binary

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
MEDIUMTEXT	text	text
SET	text	關鍵字
TEXT	text	text
TINYBLOB	text	binary
TINYTEXT	text	text
VARBINARY	text	binary
VARCHAR	text	text
DATE	長（以 epoch 毫秒為單位）	date
DATETIME	長（以 epoch 毫秒為單位）	date
TIME	長（以 epoch 毫秒為單位）	date
TIMESTAMP	長（以 epoch 毫秒為單位）	date
YEAR	長（以 epoch 毫秒為單位）	date
GEOMETRY	text (WKT 格式）	geo_shape
GEOMETRYCOLLECTION	text (WKT 格式）	geo_shape
LINESTRING	text (WKT 格式）	geo_shape
MULTILINESTRING	text (WKT 格式）	geo_shape
MULTIPOINT	text (WKT 格式）	geo_shape

MySQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
MULTIPOLYGON	text (WKT 格式)	geo_shape
POINT	text (WKT 格式)	geo_point 或 geo_shape
POLYGON	text (WKT 格式)	geo_shape
JSON	text	object

建議您在 OpenSearch Ingestion 管道中設定無效字母佇列 (DLQ)。如果您已設定佇列，OpenSearch Service 會將因動態映射失敗而無法擷取的所有失敗文件傳送至佇列。

如果自動映射失敗，您可以在管道組態`template_content`中使用 `template_type`和 來定義明確的映射規則。或者，您可以在啟動管道之前，直接在搜尋網域或集合中建立映射範本。

限制

當您為 RDS for MySQL 設定 OpenSearch 擷取管道時，請考慮下列限制：

- 整合每個管道僅支援一個 MySQL 資料庫。
- 整合目前不支援跨區域資料擷取；您的 Amazon RDS 執行個體和 OpenSearch 網域必須位於相同的中 AWS 區域。
- 整合目前不支援跨帳戶資料擷取；您的 Amazon RDS 執行個體和 OpenSearch Ingestion 管道必須位於相同的 中 AWS 帳戶。
- 確保 Amazon RDS 執行個體已使用 Secrets Manager 啟用身分驗證，Secrets Manager 是唯一支援的身分驗證機制。
- 現有的管道組態無法更新，無法從不同的資料庫和/或不同的資料表擷取資料。若要更新管道的資料庫和/或資料表名稱，您必須建立新的管道。
- 通常不支援資料定義語言 (DDL) 陳述式。如果符合下列條件，將不會維持資料一致性：
  - 主索引鍵已變更 add/delete/rename)。
  - 資料表遭到捨棄/截斷。
  - 資料欄名稱或資料類型已變更。
- 如果要同步的 MySQL 資料表未定義主索引鍵，則不保證資料一致性。您需要在 OpenSearch 接收器組態中正確定義自訂`document_id`選項，才能將更新/刪除同步至 OpenSearch。

- 不支援包含串聯刪除動作的外部金鑰參考，這可能會導致 RDS for MySQL 和 OpenSearch 之間的資料不一致。
- 不支援 Amazon RDS 多可用區域資料庫叢集。
- 支援的版本：MySQL 8.0 版和更新版本。

建議的 CloudWatch 警示

建議使用下列 CloudWatch 指標來監控擷取管道的效能。這些指標可協助您識別從匯出處理的資料量、從串流處理的事件數、處理匯出和串流事件的錯誤，以及寫入目的地的文件數。您可以設定 CloudWatch 警示，在其中一個指標在指定的時間內超過指定的值時執行動作。

指標	Description
pipeline-name .rds.credentialsChanged	此指標表示 AWS 秘密輪換的頻率。
pipeline-name .rds.executorRefreshErrors	此指標表示重新整理 AWS 秘密失敗。
pipeline-name .rds.exportRecordsTotal	此指標表示從 Amazon Aurora 匯出的記錄數目。
pipeline-name .rds.exportRecordsProcessed	此指標表示 OpenSearch Ingestion 管道處理的記錄數目。
pipeline-name .rds.exportRecordProcessingErrors	此指標表示從 Amazon Aurora 叢集讀取資料時OpenSearch 擷取管道中的處理錯誤數目。
pipeline-name .rds.expo	此指標表示成功處理的匯出記錄總數。

指標	Description
rtRecords SuccessTotal	
pipeline-name .rds.expo rtRecordsFailedTot al	此指標表示無法處理的匯出記錄總數。
pipeline-name .rds.byte sReceived	此指標表示 OpenSearch Ingestion 管道收到的位元組總數。
pipeline-name .rds.byte sProcessed	此指標表示 OpenSearch Ingestion 管道處理的位元組總數。
pipeline-name .rds.stre amRecords SuccessTotal	此指標表示從串流成功處理的記錄數量。
pipeline-name .rds.stre amRecords FailedTotal	此指標表示無法從串流處理的記錄總數。

RDS for PostgreSQL

完成下列步驟，使用 Amazon RDS for RDS for PostgreSQL 設定 OpenSearch 擷取管道。

主題

- RDS for PostgreSQL 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道
- 資料一致性

- 映射資料類型
- 限制
- 建議的 CloudWatch 警示

RDS for PostgreSQL 先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 在 Amazon RDS 中建立自訂資料庫參數群組，以設定邏輯複寫。

   ```
   rds.logical_replication=1
   ```

   如需詳細資訊，請參閱執行 Amazon RDS for PostgreSQL 的邏輯複寫。
2. 選取或建立 RDS for PostgreSQL 資料庫執行個體，並將步驟 1 中建立的參數群組與資料庫執行個體建立關聯。
3. 使用具有 Aurora 和 的密碼管理 AWS Secrets Manager，在您的 Amazon RDS 執行個體上設定使用者名稱和密碼身分驗證。您也可以建立 Secrets Manager 秘密來建立使用者名稱/密碼組合。
4. 如果您使用完整的初始快照功能，請建立 AWS KMS key 和 IAM 角色，以將資料從 Amazon RDS 匯出到 Amazon S3。

   IAM 角色應具有下列許可政策：

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Sid": "ExportPolicy",
               "Effect": "Allow",
               "Action": [
                   "s3:PutObject*",
                   "s3:ListBucket",
                   "s3:GetObject*",
                   "s3:DeleteObject*",
                   "s3:GetBucketLocation"
               ],
               "Resource": [
                   "arn:aws:s3:::s3-bucket-used-in-pipeline",
                   "arn:aws:s3:::s3-bucket-used-in-pipeline/*"
   ```

```
            ]
        }
    ]
}
```

此角色也應該具有下列信任關係：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "export.rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

5. 選取或建立 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱建立 OpenSearch Service 網域和建立集合。
6. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從 Amazon RDS 資料庫執行個體寫入您的網域或集合。

步驟 1：設定管道角色

設定 Amazon RDS 管道先決條件之後，請將管道角色設定為在管道組態中使用。同時將 Amazon RDS 來源的下列許可新增至角色：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
    "Sid": "allowReadingFromS3Buckets",
    "Effect": "Allow",
    "Action": [
```

```
    "s3:GetObject",
    "s3:DeleteObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:PutObject"
    ],
    "Resource": [
    "arn:aws:s3:::s3_bucket",
    "arn:aws:s3:::s3_bucket/*"
    ]
    },
    {
    "Sid": "allowNetworkInterfacesActions",
    "Effect": "Allow",
    "Action": [
    "ec2:AttachNetworkInterface",
    "ec2:CreateNetworkInterface",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:DetachNetworkInterface",
    "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": [
    "arn:aws:ec2:*:111122223333:network-interface/*",
    "arn:aws:ec2:*:111122223333:subnet/*",
    "arn:aws:ec2:*:111122223333:security-group/*"
    ]
    },
    {
    "Sid": "allowDescribeEC2",
    "Effect": "Allow",
    "Action": [
    "ec2:Describe*"
    ],
    "Resource": "*"
    },
    {
    "Sid": "allowTagCreation",
    "Effect": "Allow",
    "Action": [
    "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:111122223333:network-interface/*",
```

```
    "Condition": {
    "StringEquals": {
    "aws:RequestTag/OSISManaged": "true"
    }
    }
    },
    {
    "Sid": "AllowDescribeInstances",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBInstances"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:db:*"
    ]
    },
    {
    "Sid": "AllowSnapshots",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeDBSnapshots",
    "rds:CreateDBSnapshot",
    "rds:AddTagsToResource"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:db:DB-id",
    "arn:aws:rds:us-east-2:111122223333:snapshot:DB-id*"
    ]
    },
    {
    "Sid": "AllowExport",
    "Effect": "Allow",
    "Action": [
    "rds:StartExportTask"
    ],
    "Resource": [
    "arn:aws:rds:us-east-2:111122223333:snapshot:DB-id*"
    ]
    },
    {
    "Sid": "AllowDescribeExports",
    "Effect": "Allow",
    "Action": [
    "rds:DescribeExportTasks"
```

```
    ],
    "Resource": "*",
    "Condition": {
    "StringEquals": {
    "aws:RequestedRegion": "us-east-2",
    "aws:ResourceAccount": "111122223333"
    }
    }
    },
    {
    "Sid": "AllowAccessToKmsForExport",
    "Effect": "Allow",
    "Action": [
    "kms:Decrypt",
    "kms:Encrypt",
    "kms:DescribeKey",
    "kms:RetireGrant",
    "kms:CreateGrant",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
    ],
    "Resource": [
    "arn:aws:kms:us-east-2:111122223333:key/export-key-id"
    ]
    },
    {
    "Sid": "AllowPassingExportRole",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
    "arn:aws:iam::111122223333:role/export-role"
    ]
    },
    {
    "Sid": "SecretsManagerReadAccess",
    "Effect": "Allow",
    "Action": [
    "secretsmanager:GetSecretValue"
    ],
    "Resource": [
    "arn:aws:secretsmanager:*:111122223333:secret:*"
    ]
    }
    ]
```

```
    }
```

步驟 2：建立管道

如下所示設定 OpenSearch Ingestion 管道，指定 RDS for PostgreSQL 執行個體做為來源。

```
version: "2"
rds-postgres-pipeline:
  source:
    rds:
      db_identifier: "instance-id"
      engine: postgresql
      database: "database-name"
      tables:
        include:
          - "schema1.table1"
          - "schema2.table2"
      s3_bucket: "bucket-name"
      s3_region: "bucket-region"
      s3_prefix: "prefix-name"
      export:
        kms_key_id: "kms-key-id"
        iam_role_arn: "export-role-arn"
      stream: true
      aws:
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        region: "us-east-1"
      authentication:
        username: ${{aws_secrets:secret:username}}
        password: ${{aws_secrets:secret:password}}
  sink:
    - opensearch:
        hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
        index: "${getMetadata(\"table_name\")}"
        index_type: custom
        document_id: "${getMetadata(\"primary_key\")}"
        action: "${getMetadata(\"opensearch_action\")}"
        document_version: "${getMetadata(\"document_version\")}"
        document_version_type: "external"
        aws:
          sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
          region: "us-east-1"
extension:
```

```
  aws:
    secrets:
      secret:
        secret_id: "rds-secret-id"
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::account-id:role/pipeline-role"
        refresh_interval: PT1H
```

> **Note**
>
> 您可以使用預先設定的 Amazon RDS 藍圖來建立此管道。如需詳細資訊，請參閱使用藍圖。

若要使用 Amazon Aurora 做為來源，您需要設定管道的 VPC 存取。您選擇的 VPC 應與 Amazon Aurora 來源使用的 VPC 相同。然後選擇一或多個子網路和一或多個 VPC 安全群組。請注意，管道需要網路存取 Aurora MySQL 資料庫，因此您也應該確認您的 Aurora 叢集已設定 VPC 安全群組，允許從管道的 VPC 安全群組到資料庫連接埠的傳入流量。如需詳細資訊，請參閱使用安全群組控制存取。

如果您使用 AWS 管理主控台 建立管道，也必須將管道連接至 VPC，才能使用 Amazon Aurora 做為來源。若要這樣做，請尋找網路組態區段，選擇連接至 VPC，然後從其中一個提供的預設選項中選擇 CIDR，或選取您自己的選項。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。

若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 和 Amazon RDS 之間的 IP 地址發生衝突，請確保 Amazon Aurora VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

如需詳細資訊，請參閱設定管道的 VPC 存取。

資料一致性

管道會持續輪詢或接收來自 Amazon RDS 執行個體的變更，並更新 OpenSearch 索引中的對應文件，以確保資料一致性。

OpenSearch Ingestion end-to-end確認，以確保資料耐久性。當管道讀取快照或串流時，它會動態建立分割區以進行平行處理。在擷取 OpenSearch 網域或集合中的所有記錄之後，管道收到確認時，會將分割區標記為完成。如果您想要擷取到 OpenSearch Serverless 搜尋集合，您可以在管道中產生文件 ID。如果您想要擷取 OpenSearch Serverless 時間序列集合，請注意管道不會產生文件 ID，因此您必須在管道接收器組態`document_id: "${getMetadata(\"primary_key\")}"`中省略 。

OpenSearch 擷取管道也會將傳入的事件動作映射至對應的大量索引動作，以協助擷取文件。這可讓資料保持一致，以便 Amazon RDS 中的每個資料變更都會與 OpenSearch 中的對應文件變更進行協調。

映射資料類型

OpenSearch Ingestion 管道會將 PostgreSQL 資料類型映射至適合 OpenSearch Service 網域或集合使用的表示法。如果 OpenSearch 中未定義映射範本，則 OpenSearch 會根據第一個傳送的文件自動判斷具有動態映射的欄位類型。您也可以透過映射範本，在 OpenSearch 中明確定義最適合您的欄位類型。

下表列出 RDS for PostgreSQL 資料類型和對應的 OpenSearch 欄位類型。如果未定義明確的映射，預設 OpenSearch 欄位類型欄會顯示 OpenSearch 中的對應欄位類型。在此情況下，OpenSearch 會自動判斷具有動態映射的欄位類型。建議的 OpenSearch 欄位類型欄是對應範本中明確指定的對應建議欄位類型。這些欄位類型與 RDS for PostgreSQL 中的資料類型更加一致，通常可以啟用 OpenSearch 中可用的更佳搜尋功能。

RDS for PostgreSQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
smallint	long	short
integer	long	integer
bigint	long	long
decimal	text	雙 或 關鍵字
numeric【(	text	雙 或 關鍵字
real	float	float
double precision	float	double
smallserial	long	short
serial	long	integer

RDS for PostgreSQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
bigserial	long	long
money	object	object
字元變體(n)	text	text
varchar(n)	text	text
character(n)	text	text
char(n)	text	text
bpchar(n)	text	text
bpchar	text	text
text	text	text
enum	text	text
bytea	text	binary
時間戳記【(p)】【不含時區】	長（以 epoch 毫秒為單位）	date
具有時區的時間戳記【(p)】	長（以 epoch 毫秒為單位）	date
date	長（以 epoch 毫秒為單位）	date

RDS for PostgreSQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
time【(p)】【無時區】	長（以 epoch 毫秒為單位）	date
time【(p)】與時區	長（以 epoch 毫秒為單位）	date
間隔【欄位】【(p)】	text (ISO8601 格式）	text
boolean	boolean	boolean
point	text (WKT 格式）	geo_shape
線條	text (WKT 格式）	geo_shape
lseg	text (WKT 格式）	geo_shape
方塊	text (WKT 格式）	geo_shape
路徑	text (WKT 格式）	geo_shape
多邊形	text (WKT 格式）	geo_shape
圓圈	object	object
cidr	text	text
inet	text	text
macaddr	text	text

RDS for PostgreSQL 資料類型	預設 OpenSearch 欄位類型	建議的 OpenSearch 欄位類型
macaddr8	text	text
bit(n)	long	位元組、短、整數或長（取決於位元數）
bit varying(n)	long	位元組、短、整數或長（取決於位元數）
json	object	object
jsonb	object	object
jsonpath	text	text

我們建議您在 OpenSearch Ingestion 管道中設定無效字母佇列 (DLQ)。如果您已設定佇列，OpenSearch Service 會將因動態映射失敗而無法擷取的所有失敗文件傳送至佇列。

如果自動映射失敗，您可以在管道組態`template_content`中使用 `template_type`和 來定義明確的映射規則。或者，您可以在啟動管道之前，直接在搜尋網域或集合中建立映射範本。

限制

當您為 RDS for PostgreSQL 設定 OpenSearch Ingestion 管道時，請考慮下列限制：

- 整合每個管道僅支援一個 PostgreSQL 資料庫。
- 整合目前不支援跨區域資料擷取；您的 Amazon RDS 執行個體和 OpenSearch 網域必須位於相同的 中 AWS 區域。
- 整合目前不支援跨帳戶資料擷取；您的 Amazon RDS 執行個體和 OpenSearch Ingestion 管道必須位於相同的 中 AWS 帳戶。
- 確保 Amazon RDS 執行個體已使用 啟用身分驗證 AWS Secrets Manager，這是唯一支援的身分驗證機制。
- 現有的管道組態無法更新，無法從不同的資料庫和/或不同的資料表擷取資料。若要更新管道的資料庫和/或資料表名稱，您必須停止管道，並使用更新的組態重新啟動管道，或建立新的管道。

- 通常不支援資料定義語言 (DDL) 陳述式。如果符合下列條件，將不會維持資料一致性：
  - 主索引鍵已變更 add/delete/rename)。
  - 資料表遭到捨棄/截斷。
  - 資料欄名稱或資料類型已變更。
- 如果要同步的 PostgreSQL 資料表未定義主索引鍵，則不保證資料一致性。您需要在 OpenSearch 中正確定義自訂 `document_id` 選項，才能將更新/刪除同步至 OpenSearch。
- 不支援 RDS 多可用區域資料庫叢集。
- 支援的版本：PostgreSQL 16 和更新版本。

建議的 CloudWatch 警示

建議使用下列 CloudWatch 指標來監控擷取管道的效能。這些指標可協助您識別從匯出處理的資料量、從串流處理的事件數、處理匯出和串流事件的錯誤，以及寫入目的地的文件數。您可以設定 CloudWatch 警示，在其中一個指標在指定的時間內超過指定的值時執行動作。

指標	Description
pipeline-name .rds.credentialsChanged	此指標表示 AWS 秘密輪換的頻率。
pipeline-name .rds.executorRefreshErrors	此指標表示重新整理 AWS 秘密失敗。
pipeline-name .rds.exportRecordsTotal	此指標表示從 Amazon Aurora 匯出的記錄數目。
pipeline-name .rds.exportRecordsProcessed	此指標表示 OpenSearch Ingestion 管道處理的記錄數量。
pipeline-name .rds.expo	此指標表示從 Amazon Aurora 叢集讀取資料時OpenSearch Ingestion 管道中的處理錯誤數目。

指標	Description
rtRecordProcessing Errors	
pipeline-name .rds.expo rtRecords SuccessTotal	此指標表示成功處理的匯出記錄總數。
pipeline-name .rds.expo rtRecordsFailedTot al	此指標表示無法處理的匯出記錄總數。
pipeline-name .rds.byte sReceived	此指標表示 OpenSearch Ingestion 管道收到的位元組總數。
pipeline-name .rds.byte sProcessed	此指標表示 OpenSearch Ingestion 管道處理的位元組總數。
pipeline-name .rds.stre amRecords SuccessTotal	此指標表示從串流成功處理的記錄數。
pipeline-name .rds.stre amRecords FailedTotal	此指標表示無法從串流處理的記錄總數。

搭配 Amazon S3 使用 OpenSearch 擷取管道

使用 OpenSearch Ingestion，您可以使用 Amazon S3 做為來源或目的地。當您使用 Amazon S3 做為來源時，會將資料傳送至 OpenSearch Ingestion 管道。當您使用 Amazon S3 做為目的地時，您可以將資料從 OpenSearch 擷取管道寫入一或多個 S3 儲存貯體。

主題

- Amazon S3 做為來源
- Amazon S3 做為目的地
- Amazon S3 跨帳戶做為來源

Amazon S3 做為來源

有兩種方式可以使用 Amazon S3 作為來源來處理資料：透過 S3-SQS 處理和排程掃描。

當您在檔案寫入 S3 後需要近乎即時的檔案掃描時，請使用 S3-SQS 處理。 S3 您可以將 Amazon S3 儲存貯體設定為在儲存貯體中存放或修改物件時引發事件。使用一次性或重複排程掃描來批次處理 S3 儲存貯體中的資料。

主題

- 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道

先決條件

若要使用 Amazon S3 作為排程掃描或 S3-SQS 處理之 OpenSearch Ingestion 管道的來源，請先建立 S3 儲存貯體。

> **Note**
>
> 如果做為 OpenSearch Ingestion 管道來源的 S3 儲存貯體位於不同的 中 AWS 帳戶，您也需要在儲存貯體上啟用跨帳戶讀取許可。這可讓管道讀取和處理資料。若要啟用跨帳戶許可，請參閱《Amazon S3 使用者指南》中的授予跨帳戶儲存貯體許可的儲存貯體擁有者。
> 如果您的 S3 儲存貯體位於多個帳戶中，請使用`bucket_owners`映射。如需範例，請參閱 OpenSearch 文件中的跨帳戶 S3 存取。

若要設定 S3-SQS 處理，您也需要執行下列步驟：

1. 建立 Amazon SQS 佇列。
2. 在目的地為 SQS 佇列的 S3 儲存貯體上啟用事件通知。

步驟 1：設定管道角色

與其他將資料推送至管道的來源外掛程式不同，S3 來源外掛程式具有讀取型架構，其中管道會從來源提取資料。

因此，若要從 S3 讀取管道，您必須在管道的 S3 來源組態中指定可存取 S3 儲存貯體和 Amazon SQS 佇列的角色。管道將擔任此角色，以便從佇列讀取資料。

> **Note**
>
> 您在 S3 來源組態中指定的角色必須是管道角色。因此，您的管道角色必須包含兩個單獨的許可政策，一個用於寫入接收器，另一個用於從 S3 來源提取。您必須在所有管道元件`sts_role_arn`中使用相同的 。

下列範例政策顯示使用 S3 做為來源的必要許可：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action":[
          "s3:ListBucket",
          "s3:GetBucketLocation",
          "s3:GetObject"
       ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    },
    {
       "Effect":"Allow",
       "Action":"s3:ListAllMyBuckets",
       "Resource":"arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
        "sqs:ReceiveMessage",
```

```
        "sqs:ChangeMessageVisibility"
      ],
      "Resource": "arn:aws:sqs:us-east-1:111122223333:MyS3EventSqsQueue"
    }
  ]
}
```

您必須將這些許可連接到您在 S3 來源外掛程式組態中的 `sts_role_arn`選項中指定的 IAM 角色：

```
version: "2"
source:
  s3:
    ...
    aws:
      ...
processor:
  ...
sink:
  - opensearch:
      ...
```

步驟 2：建立管道

設定許可後，您可以根據 Amazon S3 使用案例設定 OpenSearch Ingestion 管道。

S3-SQS 處理

若要設定 S3-SQS 處理，請設定管道以指定 S3 做為來源，並設定 Amazon SQS 通知：

```
version: "2"
s3-pipeline:
  source:
    s3:
      notification_type: "sqs"
      codec:
        newline: null
      sqs:
        queue_url: "https://sqs.us-east-1amazonaws.com/account-id/ingestion-queue"
      compression: "none"
      aws:
        region: "region"
  processor:
```

```
  - grok:
      match:
        message:
        - "%{COMMONAPACHELOG}"
  - date:
      destination: "@timestamp"
      from_time_received: true
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
      index: "index-name"
      aws:
        region: "region"
```

如果您在 Amazon S3 上處理小型檔案時觀察到低 CPU 使用率，請考慮透過修改 workers選項的值來增加輸送量。如需詳細資訊，請參閱 S3 外掛程式組態選項。

排程掃描

若要設定排程掃描，請在套用至所有 S3 儲存貯體的掃描層級或儲存貯體層級使用排程來設定管道。儲存貯體層級排程或掃描間隔組態一律會覆寫掃描層級組態。

您可以使用適合資料遷移的一次性掃描，或適合批次處理的週期性掃描來設定排程掃描。

若要將管道設定為從 Amazon S3 讀取，請使用預先設定的 Amazon S3 藍圖。您可以編輯管道組態scan的部分，以符合排程需求。如需詳細資訊，請參閱the section called “使用藍圖”。

一次性掃描

一次性排程掃描執行一次。在管道組態中，您可以使用 start_time和 end_time 來指定何時掃描儲存貯體中的物件。或者，您可以使用 range 來指定相對於您希望掃描儲存貯體中物件之目前時間的時間間隔。

例如，設定為PT4H掃描過去四小時內建立的所有檔案的範圍。若要設定一次性掃描以執行第二次，您必須停止並重新啟動管道。如果您沒有設定範圍，您還必須更新開始和結束時間。

下列組態會為這些儲存貯體中的所有儲存貯體和所有物件設定一次性掃描：

```
version: "2"
log-pipeline:
  source:
    s3:
      codec:
```

```
      csv:
    compression: "none"
    aws:
      region: "region"
    acknowledgments: true
    scan:
      buckets:
        - bucket:
            name: my-bucket
            filter:
              include_prefix:
                - Objects1/
              exclude_suffix:
                - .jpeg
                - .png
        - bucket:
            name: my-bucket-2
            key_prefix:
              include:
                - Objects2/
              exclude_suffix:
                - .jpeg
                - .png
    delete_s3_objects_on_read: false
processor:
  - date:
      destination: "@timestamp"
      from_time_received: true
sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
      index: "index-name"
      aws:
        region: "region"
      dlq:
        s3:
          bucket: "dlq-bucket"
          region: "us-east-1"
```

下列組態會為指定時段中的所有儲存貯體設定一次性掃描。這表示 S3 只會處理建立時間落在此時段內的物件。

```
scan:
```

```
  start_time: 2023-01-21T18:00:00.000Z
  end_time: 2023-04-21T18:00:00.000Z
  buckets:
    - bucket:
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

下列組態會在掃描層級和儲存貯體層級設定一次性掃描。儲存貯體層級的開始和結束時間覆寫掃描層級的開始和結束時間。

```
scan:
  start_time: 2023-01-21T18:00:00.000Z
  end_time: 2023-04-21T18:00:00.000Z
  buckets:
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-2
        filter:
          include:
```

```
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

停止管道會移除管道在停止之前已掃描哪些物件的任何預先存在參考。如果單一掃描管道停止，它將在啟動之後重新掃描所有物件，即使它們已經掃描。如果您需要停止單一掃描管道，建議您在再次啟動管道之前變更時間範圍。

如果您需要依開始時間和結束時間篩選物件，則停止和啟動管道是唯一的選項。如果您不需要依開始時間和結束時間篩選，您可以依名稱篩選物件。依名稱轉譯不需要您停止和啟動管道。若要這樣做，請使用 `include_prefix`和 `exclude_suffix`。

重複掃描

定期排程掃描會以定期排程的間隔執行指定 S3 儲存貯體的掃描。您只能在掃描層級設定這些間隔，因為不支援個別儲存貯體層級組態。

在您的管道組態中， `interval`會指定重複掃描的頻率，而且可以介於 30 秒到 365 天之間。這些掃描的第一個一律會在您建立管道時發生。`count` 定義掃描執行個體的總數。

下列組態會設定重複掃描，掃描之間延遲 12 小時：

```
scan:
  scheduling:
    interval: PT12H
    count: 4
  buckets:
    - bucket:
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
```

```
            - .png
```

Amazon S3 做為目的地

若要將 OpenSearch Ingestion 管道的資料寫入 S3 儲存貯體，請使用預先設定的 S3 藍圖來建立具有 S3 接收器的管道。此管道會將選擇性資料路由至 OpenSearch 接收器，並同時傳送所有資料以供 S3 中封存。如需詳細資訊，請參閱the section called “使用藍圖”。

建立 S3 接收器時，您可以從各種接收器轉碼器指定偏好的格式。例如，如果您想要以單欄式格式寫入資料，請選擇 Parquet 或 Avro 轉碼器。如果您偏好以資料列為基礎的格式，請選擇 JSON 或 NDJSON。若要在指定的結構描述中將資料寫入 S3，您也可以使用 Avro 格式在接收器轉碼器中定義內嵌結構描述。

下列範例定義 S3 接收器中的內嵌結構描述：

```
- s3:
  codec:
    parquet:
      schema: >
        {
           "type" : "record",
           "namespace" : "org.vpcFlowLog.examples",
           "name" : "VpcFlowLog",
           "fields" : [
             { "name" : "version", "type" : "string"},
             { "name" : "srcport", "type": "int"},
             { "name" : "dstport", "type": "int"},
             { "name" : "start", "type": "int"},
             { "name" : "end", "type": "int"},
             { "name" : "protocol", "type": "int"},
             { "name" : "packets", "type": "int"},
             { "name" : "bytes", "type": "int"},
             { "name" : "action", "type": "string"},
             { "name" : "logStatus", "type" : "string"}
           ]
         }
```

當您定義此結構描述時，請指定可能存在於管道交付至接收器之不同類型事件的所有金鑰超級集。

例如，如果事件可能遺失索引鍵，請在結構描述中將該索引鍵加上 null值。Null 值宣告允許結構描述處理不均勻的資料（其中某些事件具有這些索引鍵，而其他則否）。當傳入事件確實存在這些索引鍵時，其值會寫入目的地。

此結構描述定義可做為篩選條件，僅允許將定義的金鑰傳送至目的地，並從傳入事件捨棄未定義的金鑰。

您也可以在目的地exclude_keys中使用 include_keys和 來篩選路由到其他目的地的資料。這兩個篩選條件是互斥的，因此您一次只能在結構描述中使用一個。此外，您無法在使用者定義的結構描述中使用它們。

若要使用此類篩選條件建立管道，請使用預先設定的目的地篩選條件藍圖。如需詳細資訊，請參閱the section called “使用藍圖”。

Amazon S3 跨帳戶做為來源

您可以使用 Amazon S3 跨帳戶授予存取權，以便 OpenSearch Ingestion 管道可以存取另一個帳戶中的 S3 儲存貯體作為來源。若要啟用跨帳戶存取，請參閱《Amazon S3 使用者指南》中的授予跨帳戶儲存貯體許可的儲存貯體擁有者。授予存取權之後，請確定您的管道角色具有必要的許可。

然後，您可以使用 建立管道bucket_owners，以啟用 Amazon S3 儲存貯體的跨帳戶存取做為來源：

```
s3-pipeline:
 source:
  s3:
   notification_type: "sqs"
   codec:
    csv:
     delimiter: ","
     quote_character: "\""
     detect_header: True
   sqs:
    queue_url: "https://sqs.ap-northeast-1.amazonaws.com/401447383613/test-s3-queue"
   bucket_owners:
    my-bucket-01: 123456789012
    my-bucket-02: 999999999999
   compression: "gzip"
```

搭配 Amazon Security Lake 使用 OpenSearch 擷取管道

您可以使用 S3 來源外掛程式，將資料從 Amazon Security Lake 擷取到 OpenSearch Ingestion 管道。Security Lake 會自動將來自 AWS 環境、內部部署環境和 SaaS 供應商的安全資料集中到專用資料湖中。您可以建立訂閱，將資料從 Security Lake 複寫到 OpenSearch Ingestion 管道，然後將其寫入 OpenSearch Service 網域或 OpenSearch Serverless 集合。

若要將管道設定為從 Security Lake 讀取，請使用預先設定的 Security Lake 藍圖。藍圖包含從 Security Lake 擷取開放網路安全結構描述架構 (OCSF) parquet 檔案的預設組態。如需詳細資訊，請參閱the section called “使用藍圖”。

主題

- 使用 OpenSearch 擷取管道搭配 Amazon Security Lake 做為來源
- 使用 OpenSearch 擷取管道搭配 Amazon Security Lake 做為接收器

使用 OpenSearch 擷取管道搭配 Amazon Security Lake 做為來源

您可以在 OpenSearch 擷取管道中使用 Amazon S3 來源外掛程式，從 Amazon Security Lake 擷取資料。Security Lake 會自動將來自 AWS 環境、內部部署系統和 SaaS 供應商的安全資料集中到專用資料湖中。

Amazon Security Lake 在管道中具有下列中繼資料屬性：

- `bucket_name`：Security Lake 為存放安全資料而建立的 Amazon S3 儲存貯體名稱。
- `path_prefix`：Security Lake IAM 角色政策中定義的自訂來源名稱。
- `region`： AWS 區域 Security Lake S3 儲存貯體所在的 。
- `accountID`：啟用 Security Lake 的 AWS 帳戶 ID。
- `sts_role_arn`：旨在與 Security Lake 搭配使用之 IAM 角色的 ARN。

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

- 啟用 Security Lake。
- 在 Security Lake 中建立訂閱者。
  - 選擇您要擷取至管道的來源。
  - 針對訂閱者登入資料，新增您要建立管道之 AWS 帳戶 的 ID。針對外部 ID，指定 `OpenSearchIngestion-{accountid}`。
  - 針對資料存取方法，選擇 S3。
  - 如需通知詳細資訊，請選擇 SQS 佇列。

當您建立訂閱者時，Security Lake 會自動建立兩個內嵌許可政策，一個用於 S3，另一個用於 SQS。這些政策採用下列格式： AmazonSecurityLake-*amzn-s3-demo-bucket*-S3和 AmazonSecurityLake-*AWS Demo*-SQS。若要允許管道存取訂閱者來源，您必須將必要的許可與管道角色建立關聯。

設定管道角色

在 IAM 中建立新的許可政策，只結合 Security Lake 自動建立的兩個政策所需的許可。下列範例政策顯示 OpenSearch 擷取管道從多個 Security Lake 來源讀取資料所需的最低權限：

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3:::aws-security-data-lake-us-east-1-abcde/aws/
LAMBDA_EXECUTION/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-us-east-1-abcde/aws/S3_DATA/1.0/
*",
            "arn:aws:s3:::aws-security-data-lake-us-east-1-abcde/aws/
VPC_FLOW/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-us-east-1-abcde/aws/ROUTE53/1.0/
*",
            "arn:aws:s3:::aws-security-data-lake-us-east-1-abcde/aws/
SH_FINDINGS/1.0/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sqs:ReceiveMessage",
            "sqs:DeleteMessage"
         ],
         "Resource":[
            "arn:aws:sqs:us-east-1:111122223333:AmazonSecurityLake-abcde-Main-
Queue"
```

```
            ]
        }
    ]
}
```

> **Important**
>
> Security Lake 不會為您管理管道角色政策。如果您從 Security Lake 訂閱新增或移除來源，則必須手動更新政策。Security Lake 會為每個日誌來源建立分割區，因此您需要在管道角色中手動新增或移除許可。

您必須將這些許可連接到您在 S3 來源外掛程式組態中的 `sts_role_arn`選項中指定的 IAM 角色，位於 下`sqs`。

```
version: "2"
source:
  s3:
    ...
    sqs:
      queue_url: "https://sqs.us-east-1amazonaws.com/account-id/
AmazonSecurityLake-amzn-s3-demo-bucket-Main-Queue"
    aws:
      ...
processor:
  ...
sink:
  - opensearch:
      ...
```

建立管道

將許可新增至管道角色之後，請使用預先設定的 Security Lake 藍圖來建立管道。如需詳細資訊，請參閱the section called “使用藍圖”。

您必須在`s3`來源組態中指定 `queue_url`選項，這是要讀取的 Amazon SQS 佇列 URL。若要格式化 URL，請在訂閱者組態中尋找訂閱端點，並`arn:aws:`變更為 `https://`。例如 `https://sqs.us-east-1amazonaws.com/account-id/AmazonSecurityLake-AWS Demo-Main-Queue`。

`sts_role_arn` 您在 S3 來源組態中指定的 必須是管道角色的 ARN。

使用 OpenSearch 擷取管道搭配 Amazon Security Lake 做為接收器

使用 OpenSearch Ingestion 中的 Amazon S3 接收器外掛程式，將資料從任何支援的來源傳送至 Amazon Security Lake。Security Lake 會從專用資料湖 AWS、內部部署環境和 SaaS 供應商收集並儲存安全資料。

若要設定管道將日誌資料寫入 Security Lake，請使用預先設定的防火牆流量日誌藍圖。藍圖包含預設組態，用於擷取存放在 Amazon S3 儲存貯體中的原始安全日誌或其他資料、處理記錄並標準化記錄。然後，它會將資料映射至開放網路安全結構描述架構 (OCSF)，並將轉換後的 OCSF 相容資料傳送至 Security Lake。

管道具有下列中繼資料屬性：

- `bucket_name`：Security Lake 為存放安全資料而建立的 Amazon S3 儲存貯體名稱。
- `path_prefix`：Security Lake IAM 角色政策中定義的自訂來源名稱。
- `region`： AWS 區域 Security Lake S3 儲存貯體所在的 。
- `accountID`：啟用 Security Lake 的 AWS 帳戶 ID。
- `sts_role_arn`：旨在與 Security Lake 搭配使用之 IAM 角色的 ARN。

先決條件

在建立管道以將資料傳送至 Security Lake 之前，請執行下列步驟：

- 啟用和設定 Amazon Security Lake：設定 Amazon Security Lake 以集中來自各種來源的安全資料。如需說明，請參閱使用主控台啟用 Security Lake。

  當您選取來源時，請選擇擷取特定 AWS 來源，然後選取您要擷取的一或多個日誌和事件來源。
- 設定許可：使用將資料寫入 Security Lake 所需的許可來設定管道角色。如需詳細資訊，請參閱管道角色。

建立管道

使用預先設定的 Security Lake 藍圖來建立管道。如需詳細資訊，請參閱使用藍圖建立管道。

搭配 Fluent Bit 使用 OpenSearch 擷取管道

此範例 Fluent Bit 組態檔案會將日誌資料從 Fluent Bit 傳送至 OpenSearch Ingestion 管道。如需擷取日誌資料的詳細資訊，請參閱 Data Prepper 文件中的 Log Analytics。

注意下列事項：

- host 值必須是您的管道端點。例如 *pipeline-endpoint*.*us-east-1*osis.amazonaws.com。
- aws_service 值必須為 osis。
- 此aws_role_arn值是用戶端擔任和用於 Signature AWS 第 4 版身分驗證的 IAM 角色的 ARN。

```
[INPUT]
  name                  tail
  refresh_interval      5
  path                  /var/log/test.log
  read_from_head        true

[OUTPUT]
  Name http
  Match *
  Host pipeline-endpoint.us-east-1osis.amazonaws.com
  Port 443
  URI /log/ingest
  Format json
  aws_auth true
  aws_region region
  aws_service osis
  aws_role_arn arn:aws:iam::account-id:role/ingestion-role
  Log_Level trace
  tls On
```

然後，您可以設定 OpenSearch Ingestion 管道，如下所示，其具有 HTTP 作為來源：

```
version: "2"
unaggregated-log-pipeline:
  source:
    http:
      path: "/log/ingest"
  processor:
    - grok:
        match:
          log:
            - "%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:network_node}
 %{NOTSPACE:network_host} %{IPORHOST:source_ip}:%{NUMBER:source_port:int} ->
 %{IPORHOST:destination_ip}:%{NUMBER:destination_port:int} %{GREEDYDATA:details}"
    - grok:
```

```
        match:
          details:
            - "'%{NOTSPACE:http_method} %{NOTSPACE:http_uri}' %{NOTSPACE:protocol}"
            - "TLS%{NOTSPACE:tls_version} %{GREEDYDATA:encryption}"
            - "%{NUMBER:status_code:int} %{NUMBER:response_size:int}"
    - delete_entries:
        with_keys: ["details", "log"]

  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
        index: "index_name"
        index_type: custom
        bulk_size: 20
        aws:
          region: "region"
```

搭配 Fluentd 使用 OpenSearch 擷取管道

Fluentd 是一種開放原始碼資料收集生態系統，可為 Fluent Bit 等不同語言和子專案提供 SDKs。此範例 Fluentd 組態檔案會將日誌資料從 Fluentd 傳送至 OpenSearch Ingestion 管道。如需擷取日誌資料的詳細資訊，請參閱 Data Prepper 文件中的 Log Analytics。

注意下列事項：

- endpoint 值必須是您的管道端點。例如 *pipeline-endpoint.us-east-1*osis.amazonaws.com/apache-log-pipeline/logs。
- aws_service 值必須為 osis。
- 此aws_role_arn值是用戶端擔任和用於 Signature AWS 第 4 版身分驗證的 IAM 角色的 ARN。

```
<source>
  @type tail
  path logs/sample.log
  path_key log
  tag apache
  <parse>
    @type none
  </parse>
</source>

<filter apache>
```

```
  @type record_transformer
  <record>
    log ${record["message"]}
  </record>
</filter>

<filter apache>
  @type record_transformer
  remove_keys message
</filter>

<match apache>
  @type http
  endpoint pipeline-endpoint.us-east-1osis.amazonaws.com/apache-log-pipeline/logs
  json_array true

  <auth>
    method aws_sigv4
    aws_service osis
    aws_region region
    aws_role_arn arn:aws:iam::account-id:role/ingestion-role
  </auth>

  <format>
    @type json
  </format>

  <buffer>
    flush_interval 1s
  </buffer>
</match>
```

然後，您可以設定 OpenSearch Ingestion 管道，如下所示，其具有 HTTP 作為來源：

```
version: "2"
apache-log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log:
```

```
            - "%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:network_node}
%{NOTSPACE:network_host} %{IPORHOST:source_ip}:%{NUMBER:source_port:int} ->
%{IPORHOST:destination_ip}:%{NUMBER:destination_port:int} %{GREEDYDATA:details}"
 sink:
   - opensearch:
       hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
       index: "index_name"
       aws_region: "region"
       aws_sigv4: true
```

搭配機器學習離線批次推論使用 OpenSearch 擷取管道

Amazon OpenSearch Ingestion (OSI) 管道支援機器學習 (ML) 離線批次推論處理，以低成本有效率地充實大量資料。每當您有可非同步處理的大型資料集時，請使用離線批次推論。離線批次推論適用於 Amazon Bedrock 和 SageMaker 模型。此功能適用於所有支援 OpenSearch Service 2 AWS 區域 .17+ 網域的 OpenSearch Ingestion。

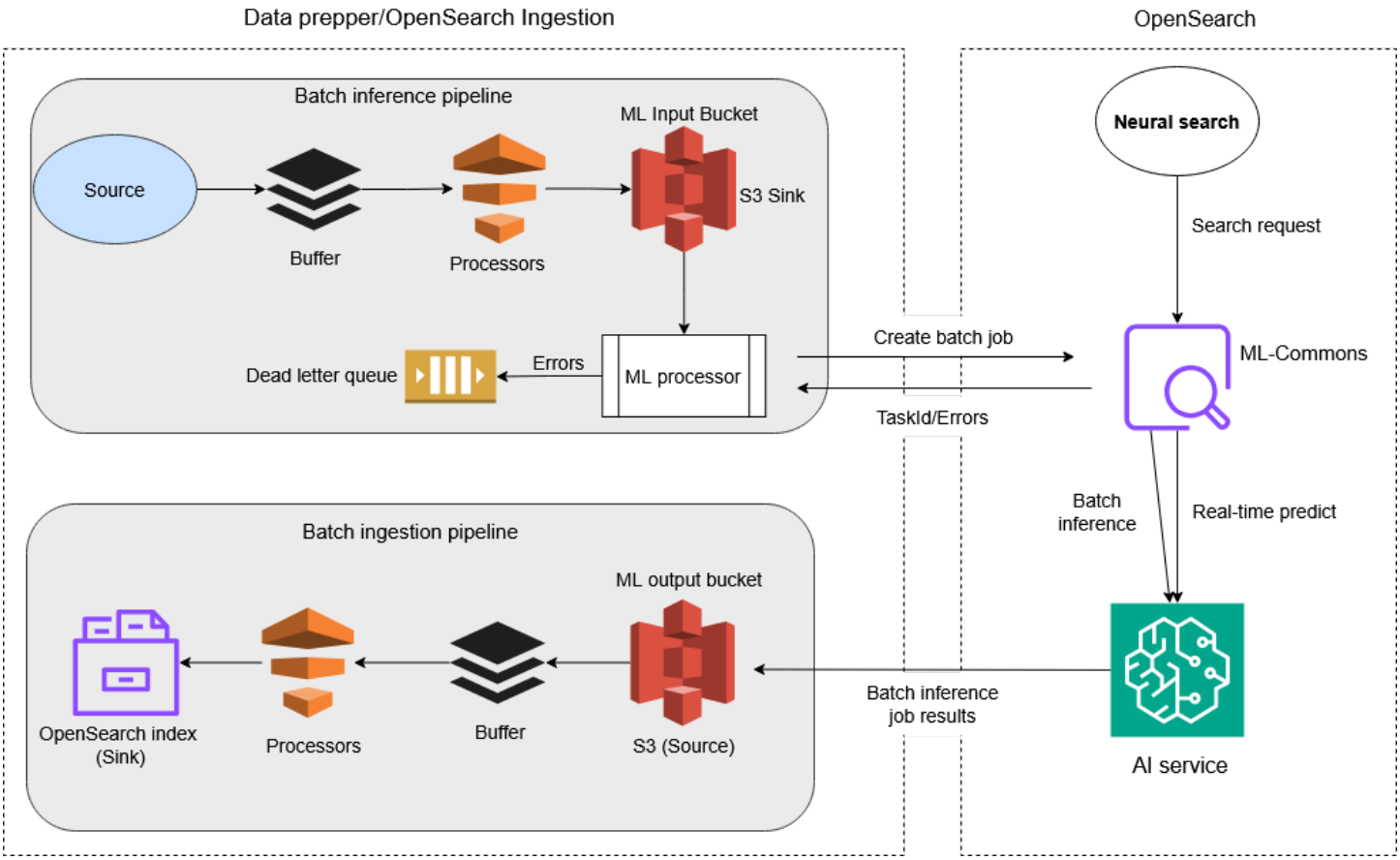
> **Note**
>
> 對於即時推論處理，請使用 適用於第三方平台的 Amazon OpenSearch Service ML 連接器。

離線批次推論處理會利用稱為 ML Commons 的 OpenSearch 功能。ML Commons 透過傳輸和 REST API 呼叫提供 ML 演算法。這些呼叫會為每個 ML 請求選擇正確的節點和資源，並監控 ML 任務以確保正常執行時間。如此一來，ML Commons 可讓您利用現有的開放原始碼 ML 演算法，並減少開發新的 ML 功能所需的工作量。如需 ML Commons 的詳細資訊，請參閱 OpenSearch.org 文件中的機器學習。

運作方式

您可以透過將機器學習推論處理器新增至管道，在 OpenSearch Ingestion 上建立離線批次推論管道。https://docs.opensearch.org/latest/ingest-pipelines/processors/ml-inference/此處理器可讓您的管道連線至 SageMaker 等 AI 服務，以執行批次推論任務。您可以將處理器設定為透過目標網域上執行的 AI 連接器（使用 batch_predict 支援）連接到所需的 AI 服務。

OpenSearch Ingestion 使用具有 ML Commons 的`ml_inference`處理器來建立離線批次推論任務。ML Commons 接著會使用 batch_predict API，該 API 會使用部署在 Amazon Bedrock、Amazon SageMaker、Cohere 和 OpenAI 中外部模型伺服器上的模型，在離線非同步模式下對大型資料集執行推論。下圖顯示 OpenSearch Ingestion 管道，可協調多個元件以端對端執行此程序：



管道元件的運作方式如下：

管道 1 （資料準備和轉換）* ：

- 來源：從 OpenSearch Ingestion 支援的外部來源掃描資料。
- 資料處理者：原始資料會經過處理，並轉換為正確的格式，以便在整合式 AI 服務上進行批次推論。
- S3 （接收器）：處理的資料會暫存在 Amazon S3 儲存貯體中，準備做為在整合 AI 服務上執行批次推論任務的輸入。

管道 2 （觸發 ML batch_inference)：

- 來源：管道 1 輸出所建立之新檔案的自動化 S3 事件偵測。
- Ml_inference 處理器：透過非同步批次工作產生 ML 推論的處理器。它透過在目標網域上執行的已設定 AI 連接器連接到 AI 服務。
- 任務 ID：每個批次任務都與 ml-commons 中的任務 ID 相關聯，以進行追蹤和管理。
- OpenSearch ML Commons：ML Commons，託管模型以進行即時神經搜尋、管理遠端 AI 伺服器的連接器，並提供 APIs以進行批次推論和任務管理。

- AI 服務：OpenSearch ML Commons 與 Amazon Bedrock 和 Amazon SageMaker 等 AI 服務互動，對資料執行批次推論，產生預測或洞察。結果會以非同步方式儲存至單獨的 S3 檔案。

管道 3 （大量擷取）：

- S3 （來源）：批次任務的結果會儲存在 S3 中，這是此管道的來源。
- 資料轉換處理器：進一步的處理和轉換會在擷取之前套用至批次推論輸出。這可確保資料在 OpenSearch 索引中正確映射。
- OpenSearch 索引 (Sink)：處理的結果會編製索引到 OpenSearch 中，以進行儲存、搜尋和進一步分析。

> **Note**
>
> *管道 1 所述的程序是選用的。如果您願意，可以略過該程序，只需在 S3 接收器中上傳準備好的資料，即可建立批次任務。

關於 ml_inference 處理器

OpenSearch Ingestion 使用 S3 掃描來源和 ML 推論處理器之間的特殊整合來進行批次處理。S3 Scan 以僅限中繼資料模式運作，可有效率地收集 S3 檔案資訊，而無需讀取實際的檔案內容。`ml_inference` 處理器使用 S3 檔案 URLs與 ML Commons 協調以進行批次處理。此設計可將掃描階段期間不必要的資料傳輸降至最低，以最佳化批次推論工作流程。您可以使用參數定義`ml_inference`處理器。請見此處範例：

```
processor:
    - ml_inference:
        # The endpoint URL of your OpenSearch domain
        host: "https://AWS test-offlinebatch-123456789abcdefg.us-
west-2.es.amazonaws.com"

        # Type of inference operation:
        # - batch_predict: for batch processing
        # - predict: for real-time inference
        action_type: "batch_predict"

        # Remote ML model service provider (Amazon Bedrock or SageMaker)
        service_name: "bedrock"
```

```
        # Unique identifier for the ML model
        model_id: "AWS TestModelID123456789abcde"

        # S3 path where batch inference results will be stored
        output_path: "s3://amzn-s3-demo-bucket/"

        # Supports ISO_8601 notation strings like PT20.345S or PT15M
        # These settings control how long to keep your inputs in the processor for
retry on throttling errors
        retry_time_window: "PT9M"

        # AWS configuration settings
        aws:
            # AWS ## where the Lambda function is deployed
            region: "us-west-2"
            # IAM role ARN for Lambda function execution
            sts_role_arn: "arn:aws::iam::account_id:role/Admin"

        # Dead-letter queue settings for storing errors
        dlq:
          s3:
            region: us-west-2
            bucket: batch-inference-dlq
            key_path_prefix: bedrock-dlq
            sts_role_arn: arn:aws:iam::account_id:role/OSI-invoke-ml

        # Conditional expression that determines when to trigger the processor
        # In this case, only process when bucket matches "amzn-s3-demo-bucket"
        ml_when: /bucket == "amzn-s3-demo-bucket"
```

使用 ml_inference 處理器改善擷取效能

OpenSearch Ingestion `ml_inference` 處理器可大幅增強啟用 ML 搜尋的資料擷取效能。處理器非常適合需要機器學習模型產生資料的使用案例，包括語意搜尋、多模式搜尋、文件擴充和查詢理解。在語意搜尋中，處理器可以按數量級加速大量、高維度向量的建立和擷取。

相較於即時模型調用，處理器的離線批次推論功能具有獨特的優勢。雖然即時處理需要具有容量限制的即時模型伺服器，但批次推論會根據需求動態擴展運算資源，並平行處理資料。例如，當 OpenSearch Ingestion 管道收到十億個來源資料請求時，它會為 ML 批次推論輸入建立 100 個 S3 檔案。然後，`ml_inference`處理器使用 100 個 `ml.m4.xlarge` Amazon Elastic Compute Cloud (Amazon EC2) 執行個體啟動 SageMaker 批次任務，在 14 小時內完成十億個請求的向量化，這項任務幾乎不可能在即時模式下完成。

設定 ml_inference 處理器以擷取語意搜尋的資料請求

下列程序會逐步引導您設定和設定 OpenSearch Ingestion `ml_inference` 處理器，以使用文字內嵌模型擷取 10 億個資料請求以進行語意搜尋。

主題

- 步驟 1：在 OpenSearch 中建立連接器並註冊模型
- 步驟 2：為 ML 離線批次推論建立 OpenSearch 擷取管道
- 步驟 3：準備資料以供擷取
- 步驟 4：監控批次推論任務
- 步驟 5：執行搜尋

步驟 1：在 OpenSearch 中建立連接器並註冊模型

針對下列程序，請使用 ML Commons batch_inference_sagemaker_connector_blueprint 在 Amazon SageMaker 中建立連接器和模型。如果您偏好使用 OpenSearch CloudFormation 整合範本，請參閱本節（替代程序）步驟 1：使用 CloudFormation 整合範本建立連接器和模型稍後的 。

在 OpenSearch 中建立連接器並註冊模型

1. 在 SageMaker 中建立用於批次轉換的 Deep Java Library (DJL) ML 模型。若要檢視其他 DJL 模型，請參閱 GitHub 上的 semantic_search_with_CFN_template_for_Sagemaker：

```
POST https://api.sagemaker.us-east-1.amazonaws.com/CreateModel
{
   "ExecutionRoleArn": "arn:aws:iam::123456789012:role/aos_ml_invoke_sagemaker",
   "ModelName": "DJL-Text-Embedding-Model-imageforjsonlines",
   "PrimaryContainer": {
      "Environment": {
         "SERVING_LOAD_MODELS" : "djl://ai.djl.huggingface.pytorch/sentence-
transformers/all-MiniLM-L6-v2"
      },
      "Image": "763104351884.dkr.ecr.us-east-1.amazonaws.com/djl-inference:0.29.0-
cpu-full"
   }
}
```

2. 在 `actions` 欄位中使用 `batch_predict` 作為新`action`類型建立連接器：

```
POST /_plugins/_ml/connectors/_create
```

```
{
  "name": "DJL Sagemaker Connector: all-MiniLM-L6-v2",
  "version": "1",
  "description": "The connector to sagemaker embedding model all-MiniLM-L6-v2",
  "protocol": "aws_sigv4",
  "credential": {
  "roleArn": "arn:aws:iam::111122223333:role/SageMakerRole"
},
  "parameters": {
    "region": "us-east-1",
    "service_name": "sagemaker",
    "DataProcessing": {
      "InputFilter": "$.text",
      "JoinSource": "Input",
      "OutputFilter": "$"
    },
    "MaxConcurrentTransforms": 100,
    "ModelName": "DJL-Text-Embedding-Model-imageforjsonlines",
    "TransformInput": {
      "ContentType": "application/json",
      "DataSource": {
        "S3DataSource": {
          "S3DataType": "S3Prefix",
          "S3Uri": "s3://offlinebatch/msmarcotests/"
        }
      },
      "SplitType": "Line"
    },
    "TransformJobName": "djl-batch-transform-1-billion",
    "TransformOutput": {
      "AssembleWith": "Line",
      "Accept": "application/json",
      "S3OutputPath": "s3://offlinebatch/msmarcotestsoutputs/"
    },
    "TransformResources": {
      "InstanceCount": 100,
      "InstanceType": "ml.m4.xlarge"
    },
    "BatchStrategy": "SingleRecord"
  },
  "actions": [
    {
      "action_type": "predict",
      "method": "POST",
```

```
      "headers": {
        "content-type": "application/json"
      },
      "url": "https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/
OpenSearch-sagemaker-060124023703/invocations",
      "request_body": "${parameters.input}",
      "pre_process_function": "connector.pre_process.default.embedding",
      "post_process_function": "connector.post_process.default.embedding"
    },
    {
      "action_type": "batch_predict",
      "method": "POST",
      "headers": {
        "content-type": "application/json"
      },
      "url": "https://api.sagemaker.us-east-1.amazonaws.com/CreateTransformJob",
      "request_body": """{ "BatchStrategy": "${parameters.BatchStrategy}",
 "ModelName": "${parameters.ModelName}", "DataProcessing" :
 ${parameters.DataProcessing}, "MaxConcurrentTransforms":
 ${parameters.MaxConcurrentTransforms}, "TransformInput":
 ${parameters.TransformInput}, "TransformJobName" :
 "${parameters.TransformJobName}", "TransformOutput" :
 ${parameters.TransformOutput}, "TransformResources" :
 ${parameters.TransformResources}}"""
    },
    {
      "action_type": "batch_predict_status",
      "method": "GET",
      "headers": {
        "content-type": "application/json"
      },
      "url": "https://api.sagemaker.us-east-1.amazonaws.com/DescribeTransformJob",
      "request_body": """{ "TransformJobName" :
 "${parameters.TransformJobName}"}"""
    },
    {
      "action_type": "cancel_batch_predict",
      "method": "POST",
      "headers": {
        "content-type": "application/json"
      },
      "url": "https://api.sagemaker.us-east-1.amazonaws.com/StopTransformJob",
      "request_body": """{ "TransformJobName" :
 "${parameters.TransformJobName}"}"""
```

```
    }
  ]
}
```

3. 使用傳回的連接器 ID 註冊 SageMaker 模型：

```
POST /_plugins/_ml/models/_register
{
    "name": "SageMaker model for batch",
    "function_name": "remote",
    "description": "test model",
    "connector_id": "example123456789-abcde"
}
```

4. 使用 batch_predict動作類型叫用模型：

```
POST /_plugins/_ml/models/teHr3JABBiEvs-eod7sn/_batch_predict
{
  "parameters": {
    "TransformJobName": "SM-offline-batch-transform"
  }
}
```

回應包含批次任務的任務 ID：

```
{
 "task_id": "exampleIDabdcefd_1234567",
 "status": "CREATED"
}
```

5. 使用任務 ID 呼叫 Get Task API 來檢查批次任務狀態：

```
GET /_plugins/_ml/tasks/exampleIDabdcefd_1234567
```

回應包含任務狀態：

```
{
  "model_id": "nyWbv5EB_tT1A82ZCu-e",
  "task_type": "BATCH_PREDICTION",
  "function_name": "REMOTE",
  "state": "RUNNING",
  "input_type": "REMOTE",
```

```
  "worker_node": [
    "WDZnIMcbTrGtnR4Lq9jPDw"
  ],
  "create_time": 1725496527958,
  "last_update_time": 1725496527958,
  "is_async": false,
  "remote_job": {
    "TransformResources": {
      "InstanceCount": 1,
      "InstanceType": "ml.c5.xlarge"
    },
    "ModelName": "DJL-Text-Embedding-Model-imageforjsonlines",
    "TransformOutput": {
      "Accept": "application/json",
      "AssembleWith": "Line",
      "KmsKeyId": "",
      "S3OutputPath": "s3://offlinebatch/output"
    },
    "CreationTime": 1725496531.935,
    "TransformInput": {
      "CompressionType": "None",
      "ContentType": "application/json",
      "DataSource": {
        "S3DataSource": {
          "S3DataType": "S3Prefix",
          "S3Uri": "s3://offlinebatch/sagemaker_djl_batch_input.json"
        }
      },
      "SplitType": "Line"
    },
    "TransformJobArn": "arn:aws:sagemaker:us-east-1:111122223333:transform-job/SM-
offline-batch-transform15",
    "TransformJobStatus": "InProgress",
    "BatchStrategy": "SingleRecord",
    "TransformJobName": "SM-offline-batch-transform15",
    "DataProcessing": {
      "InputFilter": "$.content",
      "JoinSource": "Input",
      "OutputFilter": "$"
    }
  }
}
```

（替代程序）步驟 1：使用 CloudFormation 整合範本建立連接器和模型

如果您願意，您可以使用 AWS CloudFormation 自動建立 ML 推論所需的所有必要 Amazon SageMaker 連接器和模型。此方法使用 Amazon OpenSearch Service 主控台中提供的預先設定範本來簡化設定。如需詳細資訊，請參閱使用 CloudFormation 設定語意搜尋的遠端推論。

部署可建立所有必要 SageMaker 連接器和模型的 CloudFormation 堆疊

1. 開啟 Amazon OpenSearch Service 主控台。
2. 在導覽窗格中選擇整合。
3. 在搜尋欄位中，輸入 **SageMaker**，然後選擇透過 Amazon SageMaker 與文字內嵌模型整合。
4. 選擇設定網域，然後選擇設定 VPC 網域或設定公有網域。
5. 在範本欄位中輸入資訊。針對啟用離線批次推論，選擇 true 來佈建資源以進行離線批次處理。
6. 選擇建立以建立 CloudFormation 堆疊。
7. 建立堆疊後，在主控台中 CloudFormation 開啟輸出索引標籤 尋找 connector_id 和 model_id。稍後當您設定管道時，將需要這些值。

步驟 2：為 ML 離線批次推論建立 OpenSearch 擷取管道

使用下列範例建立適用於 ML 離線批次推論的 OpenSearch 擷取管道。如需建立 OpenSearch Ingestion 管道的詳細資訊，請參閱 建立 Amazon OpenSearch Ingestion 管道。

開始之前

在下列範例中，您可以為 `sts_role_arn` 參數指定 IAM 角色 ARN。使用下列程序來驗證此角色是否對應至可存取 OpenSearch 中 ml-commons 的後端角色。

1. 導覽至 OpenSearch Service 網域的 OpenSearch Dashboards 外掛程式。您可以在 OpenSearch Service 主控台的網域儀表板上找到儀表板端點。
2. 從主選單選擇安全性、角色，然後選取 ml_full_access 角色。
3. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
4. 在後端角色下，輸入需要呼叫網域許可的 Lambda 角色 ARN。以下是範例：arn：aws：iam：：*111122223333*：role/*lambda-role*
5. 選擇 Map (映射)，並確認使用者或角色顯示在 Mapped users (已映射的使用者) 中。

為 ML 離線批次推論建立 OpenSearch 擷取管道的範例

```
version: '2'
extension:
  osis_configuration_metadata:
    builder_type: visual
sagemaker-batch-job-pipeline:
  source:
    s3:
      acknowledgments: true
      delete_s3_objects_on_read: false
      scan:
        buckets:
          - bucket:
              name: name
              data_selection: metadata_only
              filter:
                include_prefix:
                  - sagemaker/sagemaker_djl_batch_input
                exclude_suffix:
                  - .manifest
          - bucket:
              name: name
              data_selection: data_only
              filter:
                include_prefix:
                  - sagemaker/output/
        scheduling:
          interval: PT6M
      aws:
        region: name
      default_bucket_owner: account_ID
      codec:
        ndjson:
          include_empty_objects: false
      compression: none
      workers: '1'
  processor:
    - ml_inference:
        host: "https://search-AWStest-offlinebatch-123456789abcdef.us-
west-2.es.amazonaws.com"
        aws_sigv4: true
        action_type: "batch_predict"
        service_name: "sagemaker"
        model_id: "model_ID"
```

```
        output_path: "s3://AWStest-offlinebatch/sagemaker/output"
        aws:
          region: "us-west-2"
          sts_role_arn: "arn:aws:iam::account_ID:role/Admin"
        ml_when: /bucket == "AWStest-offlinebatch"
        dlq:
          s3:
            region: us-west-2
            bucket: batch-inference-dlq
            key_path_prefix: bedrock-dlq
            sts_role_arn: arn:aws:iam::account_ID:role/OSI-invoke-ml
    - copy_values:
        entries:
          - from_key: /text
            to_key: chapter
          - from_key: /SageMakerOutput
            to_key: chapter_embedding
          - delete_entries:
            with_keys:
          - text
          - SageMakerOutput
  sink:
    - opensearch:
        hosts: ["https://search-AWStest-offlinebatch-123456789abcdef.us-
west-2.es.amazonaws.com"]
        aws:
          serverless: false
          region: us-west-2
        routes:
          - ml-ingest-route
        index_type: custom
        index: test-nlp-index
  routes:
    - ml-ingest-route: /chapter != null and /title != null
```

步驟 3：準備資料以供擷取

若要準備資料以進行 ML 離線批次推論處理，請使用您自己的工具或程序自行準備資料，或使用 OpenSearch Data Prepper。使用管道來取用資料來源中的資料，或建立機器學習資料集，以確認資料組織成正確的格式。

下列範例使用 MS MARCO 資料集，其中包含自然語言處理任務的真實使用者查詢集合。資料集以 JSONL 格式建構，其中每一行代表傳送至 ML 內嵌模型的請求：

```
{"_id": "1185869", "text": ")what was the immediate impact of the Paris Peace Treaties
 of 1947?", "metadata": {"world war 2"}}
{"_id": "1185868", "text": "_________ justice is designed to repair the harm to victim,
 the community and the offender caused by the offender criminal act. question 19
 options:", "metadata": {"law"}}
{"_id": "597651", "text": "what is amber", "metadata": {"nothing"}}
{"_id": "403613", "text": "is autoimmune hepatitis a bile acid synthesis disorder",
 "metadata": {"self immune"}}
...
```

若要使用 MS MARCO 資料集進行測試，假設您建構分佈於 100 個檔案的 10 億個輸入請求，每個請求都包含 1，000 萬個請求。檔案會存放在 Amazon S3 中，字首為 s3：//offlinebatch/sagemaker/sagemaker_djl_batch_input/。OpenSearch Ingestion 管道會同時掃描這 100 個檔案，並啟動具有 100 個工作者的 SageMaker 批次任務以進行平行處理，讓 10 億份文件能夠有效率地向 OpenSearch 進行向量化和擷取。

在生產環境中，您可以使用 OpenSearch 擷取管道來產生用於批次推論輸入的 S3 檔案。管道支援各種資料來源，並依排程運作，以持續將來源資料轉換為 S3 檔案。然後，AI 伺服器會透過排定的離線批次任務自動處理這些檔案，以確保持續資料處理和擷取。

步驟 4：監控批次推論任務

您可以使用 SageMaker 主控台或 監控批次推論任務 AWS CLI。您也可以使用 Get Task API 來監控批次任務：

```
GET /_plugins/_ml/tasks/_search
{
  "query": {
    "bool": {
      "filter": [
        {
          "term": {
            "state": "RUNNING"
          }
        }
      ]
    }
  },
  "_source": ["model_id", "state", "task_type", "create_time", "last_update_time"]
}
```

API 會傳回作用中批次任務的清單：

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 3,
      "relation": "eq"
    },
    "max_score": 0.0,
    "hits": [
      {
        "_index": ".plugins-ml-task",
        "_id": "nyWbv5EB_tT1A82ZCu-e",
        "_score": 0.0,
        "_source": {
          "model_id": "nyWbv5EB_tT1A82ZCu-e",
          "state": "RUNNING",
          "task_type": "BATCH_PREDICTION",
          "create_time": 1725496527958,
          "last_update_time": 1725496527958
        }
      },
      {
        "_index": ".plugins-ml-task",
        "_id": "miKbv5EB_tT1A82ZCu-f",
        "_score": 0.0,
        "_source": {
          "model_id": "miKbv5EB_tT1A82ZCu-f",
          "state": "RUNNING",
          "task_type": "BATCH_PREDICTION",
          "create_time": 1725496528123,
          "last_update_time": 1725496528123
        }
      },
      {
        "_index": ".plugins-ml-task",
        "_id": "kiLbv5EB_tT1A82ZCu-g",
```

```
        "_score": 0.0,
        "_source": {
          "model_id": "kiLbv5EB_tT1A82ZCu-g",
          "state": "RUNNING",
          "task_type": "BATCH_PREDICTION",
          "create_time": 1725496529456,
          "last_update_time": 1725496529456
        }
      }
    ]
  }
}
```

步驟 5：執行搜尋

在監控批次推論任務並確認已完成之後，您可以執行各種類型的 AI 搜尋，包括語意、混合、對話式（使用 RAG)、神經稀疏和多模態。如需 OpenSearch Service 支援的 AI 搜尋詳細資訊，請參閱 AI 搜尋。

若要搜尋原始向量，請使用knn查詢類型，提供vector陣列做為輸入，並指定傳回的結果k數目：

```
GET /my-raw-vector-index/_search
{
  "query": {
    "knn": {
      "my_vector": {
        "vector": [0.1, 0.2, 0.3],
        "k": 2
      }
    }
  }
}
```

若要執行 AI 支援的搜尋，請使用 neural 查詢類型。指定query_text輸入、您在 OpenSearch Ingestion 管道中設定的內嵌模型model_id的 ，以及傳回的結果k數目。若要從搜尋結果中排除內嵌，請在 _source.excludes 參數中指定內嵌欄位的名稱：

```
GET /my-ai-search-index/_search
{
  "_source": {
    "excludes": [
      "output_embedding"
```

```
    ]
  },
  "query": {
    "neural": {
      "output_embedding": {
        "query_text": "What is AI search?",
        "model_id": "mBGzipQB2gmRjlv_dOoB",
        "k": 2
      }
    }
  }
}
```

搭配 OpenTelemetry Collector 使用 OpenSearch 擷取管道 OpenTelemetry

您可以使用 OpenTelemetry Collector 將日誌、追蹤和指標擷取至 OpenSearch Ingestion 管道。單一管道可用來將所有日誌、追蹤和指標擷取到網域或集合上的不同索引。您也可以使用管道單獨擷取日誌、追蹤或指標。

主題

- 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道
- 跨帳戶連線
- 限制
- OpenTelemetry 來源的建議 CloudWatch 警示

先決條件

設定 OpenTelemetry 組態檔案時，您必須設定下列項目才能進行擷取：

- 擷取角色需要 `osis:Ingest`許可才能與管道互動。如需詳細資訊，請參閱擷取角色。
- 端點值必須包含您的管道端點。例如 `https://pipeline-endpoint.us-east-1.osis.amazonaws.com`.
- 服務值必須為 `osis`。
- OTLP/HTTP Exporter 的壓縮選項必須符合管道所選來源上的壓縮選項。

```
extensions:
    sigv4auth:
        region: "region"
        service: "osis"

exporters:
    otlphttp:
        logs_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/v1/logs"
        metrics_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/v1/
metrics"
        traces_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/v1/
traces"
        auth:
            authenticator: sigv4auth
        compression: none

service:
    extensions: [sigv4auth]
    pipelines:
        traces:
        receivers: [jaeger]
        exporters: [otlphttp]
```

步驟 1：設定管道角色

設定 OpenTelemetry 收集器組態之後，請設定您要在管道組態中使用的管道角色。管道角色沒有 OTLP 來源所需的特定許可，只有授予管道存取 OpenSearch 網域或集合的許可。

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 OTLP 做為來源。您也可以將 OpenTelemetry 日誌、指標和追蹤設定為個別來源。

OTLP 來源管道組態：

```
version: 2
otlp-pipeline:
    source:
        otlp:
            logs_path: /otlp-pipeline/v1/logs
            traces_path: /otlp-pipeline/v1/traces
            metrics_path: /otlp-pipeline/v1/metrics
```

```
    sink:
        - opensearch:
            hosts: ["https://search-mydomain.region.es.amazonaws.com"]
            index: "ss4o_metrics-otel-%{yyyy.MM.dd}"
            index_type: custom
            aws:
                region: "region"
```

OpenTelemetry Logs 管道組態：

```
version: 2
otel-logs-pipeline:
  source:
    otel_logs_source:
        path: /otel-logs-pipeline/v1/logs
  sink:
    - opensearch:
        hosts: ["https://search-mydomain.region.es.amazonaws.com"]
        index: "ss4o_metrics-otel-%{yyyy.MM.dd}"
        index_type: custom
        aws:
            region: "region"
```

OpenTelemetry 指標管道組態：

```
version: 2
otel-metrics-pipeline:
  source:
    otel_metrics_source:
        path: /otel-metrics-pipeline/v1/metrics
  sink:
    - opensearch:
        hosts: ["https://search-mydomain.region.es.amazonaws.com"]
        index: "ss4o_metrics-otel-%{yyyy.MM.dd}"
        index_type: custom
        aws:
            region: "region"
```

OpenTelemetry Traces 管道組態：

```
version: 2
otel-trace-pipeline:
  source:
    otel_trace_source:
        path: /otel-traces-pipeline/v1/traces
  sink:
    - opensearch:
        hosts: ["https://search-mydomain.region.es.amazonaws.com"]
        index: "ss4o_metrics-otel-%{yyyy.MM.dd}"
        index_type: custom
        aws:
            region: "region"
```

您可以使用預先設定的藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

跨帳戶連線

具有 OpenTelemetry 來源的 OpenSearch 擷取管道具有跨帳戶擷取功能。Amazon OpenSearch Ingestion 可讓您將管道 AWS 帳戶 從虛擬私有雲端 (VPC) 跨 共用到個別 VPC 中的管道端點。如需詳細資訊，請參閱the section called “設定跨帳戶擷取的管道”。

限制

OpenSearch 擷取管道無法接收任何大於 20mb 的請求。此值由使用者在 `max_request_length`選項中設定。此選項預設為 10mb。

OpenTelemetry 來源的建議 CloudWatch 警示

建議使用下列 CloudWatch 指標來監控擷取管道的效能。這些指標可協助您識別從匯出處理的資料量、從串流處理的事件量、處理匯出和串流事件的錯誤，以及寫入目的地的文件數量。您可以設定 CloudWatch 警示，在其中一個指標在指定的時間內超過指定的值時執行動作。

OTLP 來源的 CloudWatch 指標格式為 `{pipeline-name}.otlp.{logs | traces | metrics}.{metric-name}`。例如 `otel-pipeline.otlp.metrics.requestTimeouts.count`。

如果使用個別 OpenTelemetry 來源，則指標會格式化為 `{pipeline-name}.{source-name}.{metric-name}`。例如 `trace-pipeline.otel_trace_source.requestTimeouts.count`。

所有三種 OpenTelemetry 資料類型都會有相同的指標，但為了簡潔起見，這些指標只會在下表中列出 OTLP 來源日誌類型資料。

指標	Description
otel-pipeline.BlockingBuffer.bufferUsage.value	指出使用了多少緩衝區。
otel-pipeline.otlp.logs.requestTimeouts.count	已逾時的請求數目。
otel-pipeline.otlp.logs.requestsReceived.count	OpenTelemetry Collector 收到的請求數量。
otel-pipeline.otlp.logs.badRequests.count	OpenTelemetry Collector 收到的格式不正確請求數量。
otel-pipeline.otlp.logs.requestsTooLarge.count	OpenTelemetry Collector 收到的請求數量超過 20mb 的上限。
otel-pipeline.otlp.logs.internalServerError.count	The number of HTTP 500 errors received from the OpenTelemetry Collector.
otel-pipeline.opensearch.bulkBadRequestErrors.count	Count of errors during bulk requests due to malformed request.
otel-pipeline.opensearch.bulkRequestLatency.avg	Average latency for bulk write requests made to OpenSearch.
otel-pipeline.opensearch.bulkRequestNotFoundErrors.count	Number of bulk requests that failed because the target data could not be found.
otel-pipeline.opensearch.bulkRequestNumberOfRetries.count	Number of retries by OpenSearch Ingestion pipelines to write OpenSearch cluster.
otel-pipeline.opensearch.bulkRequestSizeBytes.sum	Total size in bytes of all bulk requests made to OpenSearch.
otel-pipeline.opensearch.documentErrors.count	Number of errors when sending documents to OpenSearch. The documents causing the errors witll be sent to DLQ.

指標	Description
otel-pipeline.opensearch.documentsSuccess.count	Number of documents successfully written to an OpenSearch cluster or collection.
otel-pipeline.opensearch.documentsSuccessFirstAttempt.count	Number of documents successfully indexed in OpenSearch on the first attempt.
otel-pipeline.opensearch.documentsVersionConflictErrors.count	Count of errors due to version conflicts in documents during processing.
otel-pipeline.opensearch.PipelineLatency.avg	Average latency of OpenSearch Ingestion pipeline to process the data by reading from the source to writing to the destination.
otel-pipeline.opensearch.PipelineLatency.max	Maximum latency of OpenSearch Ingestion pipeline to process the data by reading from the source to writing the destination.
otel-pipeline.opensearch.recordsIn.count	Count of records successfully ingested into OpenSearch. This metric is essential for tracking the volume of data being processed and stored.
otel-pipeline.opensearch.s3.dlqS3RecordsFailed.count	Number of records that failed to write to DLQ.
otel-pipeline.opensearch.s3.dlqS3RecordsSuccess.count	Number of records that are written to DLQ.
otel-pipeline.opensearch.s3.dlqS3RequestLatency.count	Count of latency measurements for requests to the Amazon S3 dead-letter queue.
otel-pipeline.opensearch.s3.dlqS3RequestLatency.sum	Total latency for all requests to the Amazon S3 dead-letter queue

指標	Description
`otel-pipeline.opensearch.s3.dlqS3RequestSizeBytes.sum`	Total size in bytes of all requests made to the Amazon S3 dead-letter queue.
`otel-pipeline.recordsProcessed.count`	Total number of records processed in the pipeline, a key metric for overal throughput.
`otel-pipeline.opensearch.bulkRequestInvalidInputErrors.count`	Count of bulk request errors in OpenSearch due to invalid input, crucial for monitoring data quality and operational issues.

搭配 Amazon Managed Service for Prometheus 使用 OpenSearch 擷取管道

您可以使用 Amazon Managed Service for Prometheus 做為 OpenSearch Ingestion 管道的目的地，以時間序列格式存放指標。Prometheus 接收器可讓您將 OpenTelemetry 指標或其他時間序列資料從管道傳送至 Amazon Managed Service for Prometheus 工作區，以進行監控、提醒和分析。

`prometheus` 接收器外掛程式可讓 OpenSearch Ingestion 管道使用 Prometheus 遠端寫入通訊協定，將指標資料寫入 Amazon Managed Service for Prometheus 工作區。此整合可讓您：

- 在 Amazon Managed Service for Prometheus 中存放時間序列指標資料
- 使用 Amazon Managed Service for Prometheus 和 Amazon Managed Grafana 監控指標並發出提醒
- 同時將指標路由到多個目的地（例如，OpenSearch 和 Amazon Managed Service for Prometheus)
- 從外部客服人員處理 OpenTelemetry 指標或在管道中產生指標

主題

- 先決條件
- 步驟 1：設定管道角色
- 步驟 2：建立管道
- 監控和疑難排解
- 限制
- 最佳實務

先決條件

設定 Prometheus 接收器之前，請確定您有下列項目：

- Amazon Managed Service for Prometheus 工作區： AWS 區域 在與 OpenSearch Ingestion 管道相同的 AWS 帳戶 和 中建立工作區。如需說明，請參閱《Amazon Managed Service for Prometheus 使用者指南》中的建立工作區。
- IAM 許可：設定具有寫入 Amazon Managed Service for Prometheus 許可的 IAM 角色。如需詳細資訊，請參閱the section called “步驟 1：設定管道角色”。

> **Note**
>
> Amazon Managed Service for Prometheus 工作區必須使用 AWS 服務受管 AWS KMS 金鑰。OpenSearch Ingestion 管道中的 Amazon Managed Service for Prometheus 接收器目前不支援客戶受管 AWS KMS 金鑰。

步驟 1：設定管道角色

Prometheus 接收器會自動繼承管道角色的 IAM 身分驗證許可，因此在接收器設定中不需要額外的角色組態（例如 `sts_role_arn`）。

下列範例政策顯示使用 Amazon Managed Service for Prometheus 做為接收器的必要許可：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AMPRemoteWrite",
      "Effect": "Allow",
      "Action": [
        "aps:RemoteWrite"
      ],
      "Resource": "arn:aws:aps:region:account-id:workspace/workspace-id"
    }
  ]
}
```

取代下列預留位置：

- *region*：您的 AWS 區域（例如，us-east-1)
- *account-id*：您的 AWS 帳戶 ID
- *workspace-id*：您的 Amazon Managed Service for Prometheus 工作區 ID

您必須將這些許可連接到管道角色。

確保您的管道角色具有信任關係，允許 OpenSearch Ingestion 擔任它：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

步驟 2：建立管道

設定許可後，您可以設定 OpenSearch Ingestion 管道，以使用 Amazon Managed Service for Prometheus 做為接收器。

基本組態

下列範例顯示最小 Prometheus 接收器組態：

```
version: "2"
sink:
  - prometheus:
      url: "https://aps-workspaces.region.amazonaws.com/workspaces/workspace-id/api/v1/
remote_write"
      aws:
        region: "region"
```

您必須在prometheus接收器組態中指定 url選項，也就是 Amazon Managed Service for Prometheus 遠端寫入端點。若要格式化 URL，請在 Amazon Managed Service for

Prometheus 主控台中找到您的工作區 ID，並建構 URL，如下所示：`https://aps-workspaces.region.amazonaws.com/workspaces/workspace-id/api/v1/remote_write`。

組態選項

使用下列選項來設定 Prometheus 接收器的批次處理和排清行為：

Prometheus 接收器組態選項

選項	必要	Type	說明
max_events	否	Integer	排清至 Prometheus 之前要累積的事件數量上限。預設值為 1000。
max_request_size	否	位元組計數	排清前請求承載的大小上限。預設值為 1mb。
flush_interval	否	持續時間	排清事件之前等待的時間上限。預設值為 10s。允許的值上限為 60s。

範例管道

範例 1：Amazon Managed Service for Prometheus 的 OpenTelemetry 指標

此管道會從外部代理程式接收 OpenTelemetry 指標，並將其寫入 Amazon Managed Service for Prometheus：

```
version: "2"
source:
  otel_metrics_source:
    path: "/v1/metrics"
    output_format: otel

sink:
  - prometheus:
      url: "https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/api/v1/remote_write"
      aws:
        region: "us-east-1"
```

範例 2：雙接收器 - OpenSearch 和 Amazon Managed Service for Prometheus

此管道會將指標路由到 OpenSearch 和 Amazon Managed Service for Prometheus：

```
version: "2"
source:
  otel_metrics_source:
    path: "/v1/metrics"
    output_format: otel

sink:
  - opensearch:
      hosts:
        - "https://search-domain-endpoint.us-east-1.es.amazonaws.com"
      index: "metrics-%{yyyy.MM.dd}"
      aws:
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::123456789012:role/OSI-Pipeline-Role"

  - prometheus:
      url: "https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/api/v1/remote_write"
      aws:
        region: "us-east-1"
```

範例 3：具有篩選的指標

此管道會在傳送至 Amazon Managed Service for Prometheus 之前篩選指標：

```
version: "2"
source:
  otel_metrics_source:
    path: "/v1/metrics"
    output_format: otel

processor:
  - drop_events:
      drop_when: '/name != "http.server.duration" and /name != "http.client.duration"'

sink:
  - prometheus:
      url: "https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/api/v1/remote_write"
```

```
      aws:
        region: "us-east-1"
```

您可以使用預先設定的 Amazon Managed Service for Prometheus 藍圖來建立這些管道。如需詳細資訊，請參閱the section called “使用藍圖”。

使用 Amazon Managed Service for Prometheus 接收器建立管道

使用 AWS 主控台

1. 導覽至 OpenSearch Service 主控台。
2. 選擇擷取下的管道。
3. 選擇 Create pipeline (建立管道)。
4. 選取使用藍圖建置，然後選擇 OpenTelemetry 指標到 Amazon Prometheus 藍圖。
5. 設定管道：
   - 輸入您的 Amazon Managed Service for Prometheus 工作區 ID
   - 指定管道角色 ARN
   - 視需要設定來源和處理器設定
6. 檢閱並建立管道。

使用 AWS CLI

使用所需的組態建立管道組態檔案 （例如 `amp-pipeline.yaml`），然後執行：

```
aws osis create-pipeline \
  --pipeline-name my-amp-pipeline \
  --min-units 2 \
  --max-units 4 \
  --pipeline-configuration-body file://amp-pipeline.yaml
```

使用 AWS CloudFormation

```
Resources:
  MyAMPPipeline:
    Type: AWS::OSIS::Pipeline
    Properties:
      PipelineName: my-amp-pipeline
```

```
      MinUnits: 2
      MaxUnits: 4
      PipelineConfigurationBody: |
        version: "2"
        source:
          otel_metrics_source:
            path: "/v1/metrics"
            output_format: otel
        sink:
          - prometheus:
              url: "https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/api/v1/remote_write"
              aws:
                region: "us-east-1"
```

監控和疑難排解

CloudWatch 指標

使用 CloudWatch 指標監控管道的效能：

- `DocumentsWritten`：成功寫入 Amazon Managed Service for Prometheus 的指標數量
- `DocumentsWriteFailed`：無法寫入的指標數量
- `RequestLatency`：遠端寫入請求的延遲

常見問題

問題：管道無法寫入 Amazon Managed Service for Prometheus

解決方案：

- 驗證 URL 中的工作區 ID 和區域是否正確
- 確保管道角色具有 `aps:RemoteWrite` 許可
- 檢查工作區是否使用服務受管 AWS KMS 金鑰
- 驗證管道和工作區是否位於相同的 AWS 帳戶

問題：身分驗證錯誤

解決方案：

- 驗證信任關係`osis-pipelines.amazonaws.com`允許 擔任管道角色
- 確保管道角色具有必要的`aps:RemoteWrite`許可

問題：高延遲或限流

解決方案：

- 增加管道容量單位
- 在處理器中實作批次處理
- 檢閱 Amazon Managed Service for Prometheus 服務配額

限制

當您為 Amazon Managed Service for Prometheus 設定 OpenSearch Ingestion 管道時，請考慮下列限制：

- Amazon Managed Service for Prometheus 工作區必須使用 AWS 服務受管 AWS KMS 金鑰。目前不支援客戶受管 AWS KMS 金鑰。
- 管道和 Amazon Managed Service for Prometheus 工作區必須位於相同的 中 AWS 帳戶。

最佳實務

- 使用相同的 IAM 角色：Prometheus 接收器會自動使用管道角色。如果使用其他接收器，請確保 `sts_role_arn` 與管道角色相同
- 監控指標：針對失敗的寫入和高延遲設定 CloudWatch 警示
- 實作篩選：在傳送至 Amazon Managed Service for Prometheus 之前，使用處理器篩選不必要的指標
- 適當大小的容量：從最小容量開始，並根據指標磁碟區擴展
- 使用藍圖：利用預先設定的藍圖處理常見使用案例

搭配 Kafka 使用 OpenSearch 擷取管道

您可以使用 Kafka 外掛程式，將資料從自我管理的 Kafka 叢集串流到 Amazon OpenSearch Service 網域和 OpenSearch Serverless 集合。OpenSearch Ingestion 支援使用公有或私有 (VPC) 聯網設定的

Kafka 叢集連線。本主題概述設定擷取管道的先決條件和步驟，包括設定網路設定和身分驗證方法，例如交互 TLS (mTLS)、SASL/SCRAM 或 IAM。

從公有 Kafka 叢集遷移資料

您可以使用 OpenSearch Ingestion 管道從公有自我管理 Kafka 叢集遷移資料，這表示網域 DNS 名稱可以公開解析。若要這麼做，請設定 OpenSearch Ingestion 管道，將自我管理的 Kafka 做為來源，並將 OpenSearch Service 或 OpenSearch Serverless 做為目的地。這會處理從自我管理來源叢集到受 AWS管目的地網域或集合的串流資料。

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 使用公有網路組態建立自我管理的 Kafka 叢集。叢集應包含您要擷取至 OpenSearch Service 的資料。
2. 建立您要將資料遷移至其中的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱the section called “建立 OpenSearch Service 網域”及the section called “建立集合”。
3. 使用 在自我管理叢集上設定身分驗證 AWS Secrets Manager。依照輪換秘密中的步驟啟用AWS Secrets Manager 秘密輪換。
4. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從自我管理的叢集寫入您的網域或集合。

   下列範例網域存取政策允許您在下一個步驟中建立的管道角色將資料寫入網域。請務必`resource`使用自己的 ARN 更新 。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
```

```
      "Resource": [
        "arn:aws:es:us-east-1:111122223333:domain/domain-name"
      ]
    }
  ]
}
```

若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱 the section called “設定角色和使用者”。

步驟 1：設定管道角色

設定 Kafka 管道先決條件之後，請設定管道組態中要使用的管道角色，並新增寫入 OpenSearch Service 網域或 OpenSearch Serverless 集合的許可，以及從 Secrets Manager 讀取秘密的許可。

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 Kafka 作為來源。

您可以指定多個 OpenSearch Service 網域做為資料的目的地。此功能可將傳入資料的條件式路由或複寫到多個 OpenSearch Service 網域。

您也可以將資料從來源 Confluent Kafka 叢集遷移至 OpenSearch Serverless VPC 集合。請確定您在管道組態中提供網路存取政策。您可以使用 Confluent 結構描述登錄檔來定義 Confluent 結構描述。

```
version: "2"
kafka-pipeline:
  source:
    kafka:
      encryption:
        type: "ssl"
      topics:
        - name: "topic-name"
          group_id: "group-id"
      bootstrap_servers:
        - "bootstrap-server.us-east-1.aws.private.confluent.cloud:9092"
      authentication:
        sasl:
          plain:
            username: ${aws_secrets:confluent-kafka-secret:username}
            password: ${aws_secrets:confluent-kafka-secret:password}
      schema:
```

```
        type: confluent
        registry_url: https://my-registry.us-east-1.aws.confluent.cloud
        api_key: "${{aws_secrets:schema-secret:schema_registry_api_key}}"
        api_secret: "${{aws_secrets:schema-secret:schema_registry_api_secret}}"
        basic_auth_credentials_source: "USER_INFO"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
      index: "confluent-index"
extension:
  aws:
    secrets:
      confluent-kafka-secret:
        secret_id: "my-kafka-secret"
        region: "us-east-1"
      schema-secret:
        secret_id: "my-self-managed-kafka-schema"
        region: "us-east-1"
```

您可以使用預先設定的藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

從 VPC 中的 Kafka 叢集遷移資料

您也可以使用 OpenSearch Ingestion 管道，從 VPC 中執行的自我管理 Kafka 叢集遷移資料。若要這麼做，請設定 OpenSearch Ingestion 管道，將自我管理的 Kafka 做為來源，並將 OpenSearch Service 或 OpenSearch Serverless 做為目的地。這會處理從自我管理來源叢集到受 AWS管目的地網域或集合的串流資料。

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 使用包含您要擷取至 OpenSearch Service 之資料的 VPC 網路組態建立自我管理 Kafka 叢集。
2. 建立您要將資料遷移至其中的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱建立 OpenSearch Service 網域和建立集合。
3. 使用 在自我管理叢集上設定身分驗證 AWS Secrets Manager。依照輪換秘密中的步驟啟用AWS Secrets Manager 秘密輪換。
4. 取得可存取自我管理 Kafka 的 VPC ID。選擇要由 OpenSearch Ingestion 使用的 VPC CIDR。

> **Note**
>
> 如果您使用 AWS 管理主控台 建立管道，也必須將 OpenSearch Ingestion 管道連接至 VPC，才能使用自我管理的 Kafka。若要這樣做，請尋找網路組態區段，選取連接至 VPC 核取方塊，然後從其中一個提供的預設選項中選擇 CIDR，或選取您自己的選項。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。
> 若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 與自我管理 OpenSearch 之間的 IP 地址發生衝突，請確定自我管理的 OpenSearch VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

5. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從自我管理的叢集寫入您的網域或集合。

   下列範例網域存取政策允許您在下一個步驟中建立的管道角色將資料寫入網域。請務必`resource`使用自己的 ARN 更新 。

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
         },
         "Action": [
           "es:DescribeDomain",
           "es:ESHttp*"
         ],
         "Resource": [
           "arn:aws:es:us-east-1:111122223333:domain/domain-name"
         ]
       }
     ]
   }
   ```

   若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱 the section called “設定角色和使用者”。

步驟 1：設定管道角色

在您設定管道先決條件之後，請設定您要在管道組態中使用的管道角色，並在角色中新增下列許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": ["arn:aws:secretsmanager:us-
east-1:111122223333:secret:secret-name"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
```

```
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals":
                    {
                        "aws:RequestTag/OSISManaged": "true"
                    }
            }
        }
    ]
}
```

您必須對您用來建立 OpenSearch Ingestion 管道的 IAM 角色提供上述 Amazon EC2 許可，因為管道使用這些許可來建立和刪除 VPC 中的網路介面。管道只能透過此網路界面存取 Kafka 叢集。

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 Kafka 作為來源。

您可以指定多個 OpenSearch Service 網域做為資料的目的地。此功能可將傳入資料的條件式路由或複寫到多個 OpenSearch Service 網域。

您也可以將資料從來源 Confluent Kafka 叢集遷移至 OpenSearch Serverless VPC 集合。請確定您在管道組態中提供網路存取政策。您可以使用 Confluent 結構描述登錄檔來定義 Confluent 結構描述。

```
 version: "2"
kafka-pipeline:
  source:
    kafka:
      encryption:
        type: "ssl"
      topics:
        - name: "topic-name"
          group_id: "group-id"
```

```
      bootstrap_servers:
        - "bootstrap-server.us-east-1.aws.private.confluent.cloud:9092"
      authentication:
        sasl:
          plain:
            username: ${aws_secrets:confluent-kafka-secret:username}
            password: ${aws_secrets:confluent-kafka-secret:password}
      schema:
        type: confluent
        registry_url: https://my-registry.us-east-1.aws.confluent.cloud
        api_key: "${{aws_secrets:schema-secret:schema_registry_api_key}}"
        api_secret: "${{aws_secrets:schema-secret:schema_registry_api_secret}}"
        basic_auth_credentials_source: "USER_INFO"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
      index: "confluent-index"
extension:
  aws:
    secrets:
      confluent-kafka-secret:
        secret_id: "my-kafka-secret"
        region: "us-east-1"
      schema-secret:
        secret_id: "my-self-managed-kafka-schema"
        region: "us-east-1"
```

您可以使用預先設定的藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

使用 Amazon OpenSearch Ingestion 從自我管理的 OpenSearch 叢集遷移資料

您可以使用 Amazon OpenSearch Ingestion 管道搭配自我管理的 OpenSearch 或 Elasticsearch，將資料遷移至 Amazon OpenSearch Service 網域和 OpenSearch Serverless 集合。OpenSearch Ingestion 支援公有和私有網路組態，以便從自我管理的 OpenSearch 和 Elasticsearch 遷移資料。

從公有 OpenSearch 叢集遷移

您可以使用 OpenSearch Ingestion 管道從具有公有組態的自我管理 OpenSearch 或 Elasticsearch 叢集遷移資料，這表示網域 DNS 名稱可以公開解析。若要這樣做，請使用自我管理的 OpenSearch 或 Elasticsearch 作為來源，並將 OpenSearch Service 或 OpenSearch Serverless 作為目的地來設定 OpenSearch Ingestion 管道。 OpenSearch 這可將您的資料從自我管理的來源叢集有效地遷移到 AWS 受管目的地網域或集合。

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 建立自我管理的 OpenSearch 或 Elastisearch 叢集，其中包含您要遷移和設定公有 DNS 名稱的資料。如需說明，請參閱 OpenSearch 文件中的建立叢集。
2. 建立您要將資料遷移至其中的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱the section called “建立 OpenSearch Service 網域”及the section called “建立集合”。
3. 使用 在自我管理叢集上設定身分驗證 AWS Secrets Manager。依照輪換秘密中的步驟啟用AWS Secrets Manager 秘密輪換。
4. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從自我管理的叢集寫入您的網域或集合。

   下列範例網域存取政策允許您在下一個步驟中建立的管道角色將資料寫入網域。請務必`resource`使用自己的 ARN 更新 。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-east-1:111122223333:domain/domain-name"
```

```
      ]
    }
  ]
}
```

若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱 the section called “設定角色和使用者”。

步驟 1：設定管道角色

設定 OpenSearch 管道先決條件之後，請設定管道組態中要使用的管道角色，並新增寫入 OpenSearch Service 網域或 OpenSearch Serverless 集合的許可，以及從 Secrets Manager 讀取秘密的許可。

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 OpenSearch 作為來源。

您可以指定多個 OpenSearch Service 網域做為資料的目的地。此功能可將傳入資料的條件式路由或複寫到多個 OpenSearch Service 網域。

您也可以將資料從來源 OpenSearch 或 Elasticsearch 叢集遷移至 OpenSearch Serverless VPC 集合。請確定您在管道組態中提供網路存取政策。

```
version: "2"
opensearch-migration-pipeline:
  source:
    opensearch:
      acknowledgments: true
      host: [ "https://my-self-managed-cluster-name:9200" ]
      indices:
        include:
          - index_name_regex: "include-.*"
        exclude:
          - index_name_regex: '\..*'
      authentication:
        username: ${aws_secrets:secret:username}
        password: ${aws_secrets:secret:password}
        scheduling:
           interval: "PT2H"
           index_read_count: 3
           start_time: "2023-06-02T22:01:30.00Z"
```

```
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
          #Uncomment the following lines if your destination is an OpenSearch
 Serverless collection
          #serverless: true
          # serverless_options:
          #     network_policy_name: "network-policy-name"
      index: "${getMetadata(\"opensearch-index\")}"
      document_id: "${getMetadata(\"opensearch-document_id\")}"
      enable_request_compression: true
      dlq:
        s3:
          bucket: "bucket-name"
          key_path_prefix: "apache-log-pipeline/logs/dlq"
          region: "us-east-1"
extension:
  aws:
    secrets:
      secret:
        secret_id: "my-opensearch-secret"
        region: "us-east-1"
        refresh_interval: PT1H
```

您可以使用預先設定的藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

從 VPC 中的 OpenSearch 叢集遷移資料

您也可以使用 OpenSearch Ingestion 管道，從 VPC 中執行的自我管理 OpenSearch 或 Elasticsearch 叢集遷移資料。若要這樣做，請使用自我管理的 OpenSearch 或 Elasticsearch 作為來源，並將 OpenSearch Service 或 OpenSearch Serverless 作為目的地來設定 OpenSearch Ingestion 管道。OpenSearch 這可將您的資料從自我管理的來源叢集有效地遷移到 AWS受管目的地網域或集合。

先決條件

建立 OpenSearch Ingestion 管道之前，請執行下列步驟：

1. 使用包含您要遷移之資料的 VPC 網路組態建立自我管理 OpenSearch 或 Elastisearch 叢集。
2. 建立您要將資料遷移至其中的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱建立 OpenSearch Service 網域和建立集合。

3. 使用 在自我管理叢集上設定身分驗證 AWS Secrets Manager。依照輪換秘密中的步驟啟用AWS Secrets Manager 秘密輪換。
4. 取得可存取自我管理 OpenSearch 或 Elasticsearch 的 VPC ID。選擇要由 OpenSearch Ingestion 使用的 VPC CIDR。

> **Note**
>
> 如果您使用 AWS 管理主控台 建立管道，也必須將 OpenSearch Ingestion 管道連接至 VPC，才能使用自我管理的 OpenSearch 或 Elasticsearch。若要這樣做，請尋找來源網路選項區段，選取連接至 VPC 核取方塊，然後從其中一個提供的預設選項中選擇 CIDR。您可以從私有地址空間使用任何 CIDR，如 RFC 1918 最佳實務所定義。
> 若要提供自訂 CIDR，請從下拉式功能表中選取其他。若要避免 OpenSearch Ingestion 與自我管理 OpenSearch 之間的 IP 地址發生衝突，請確定自我管理的 OpenSearch VPC CIDR 與 OpenSearch Ingestion 的 CIDR 不同。

5. 將資源型政策連接至您的網域，或將資料存取政策連接至您的集合。這些存取政策允許 OpenSearch Ingestion 將資料從自我管理的叢集寫入您的網域或集合。

   下列範例網域存取政策允許您在下一個步驟中建立的管道角色將資料寫入網域。請務必`resource`使用自己的 ARN 更新 。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-east-1:111122223333:domain/example.com"
      ]
    }
  ]
```

```
}
```

若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱 the section called “設定角色和使用者”。

步驟 1：設定管道角色

在您設定管道先決條件之後，請設定您要在管道組態中使用的管道角色，並在角色中新增下列許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": ["arn:aws:secretsmanager:us-
east-1:111122223333:secret:secret-name"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals":
                    {
                        "aws:RequestTag/OSISManaged": "true"
                    }
            }
        }
    ]
}
```

您必須對您用來建立 OpenSearch Ingestion 管道的 IAM 角色提供上述 Amazon EC2 許可，因為管道使用這些許可來建立和刪除 VPC 中的網路介面。管道只能透過此網路界面存取 OpenSearch 叢集。

步驟 2：建立管道

然後，您可以如下所示設定 OpenSearch Ingestion 管道，指定 OpenSearch 作為來源。

您可以指定多個 OpenSearch Service 網域做為資料的目的地。此功能可將傳入資料的條件式路由或複寫到多個 OpenSearch Service 網域。

您也可以將資料從來源 OpenSearch 或 Elasticsearch 叢集遷移至 OpenSearch Serverless VPC 集合。請確定您在管道組態中提供網路存取政策。

```
version: "2"
opensearch-migration-pipeline:
  source:
```

```
    opensearch:
      acknowledgments: true
      host: [ "https://my-self-managed-cluster-name:9200" ]
      indices:
        include:
          - index_name_regex: "include-.*"
        exclude:
          - index_name_regex: '\..*'
      authentication:
        username: ${aws_secrets:secret:username}
        password: ${aws_secrets:secret:password}
        scheduling:
           interval: "PT2H"
           index_read_count: 3
           start_time: "2023-06-02T22:01:30.00Z"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
          #Uncomment the following lines if your destination is an OpenSearch
 Serverless collection
          #serverless: true
          # serverless_options:
          #     network_policy_name: "network-policy-name"
      index: "${getMetadata(\"opensearch-index\")}"
      document_id: "${getMetadata(\"opensearch-document_id\")}"
      enable_request_compression: true
      dlq:
        s3:
          bucket: "bucket-name"
          key_path_prefix: "apache-log-pipeline/logs/dlq"
          region: "us-east-1"
extension:
  aws:
    secrets:
      secret:
        secret_id: "my-opensearch-secret"
        region: "us-east-1"
        refresh_interval: PT1H
```

您可以使用預先設定的藍圖來建立此管道。如需詳細資訊，請參閱the section called “使用藍圖”。

搭配 Amazon Kinesis Data Streams 使用 OpenSearch 擷取管道

搭配 Amazon Kinesis Data Streams 使用 OpenSearch Ingestion 管道，將來自多個串流的資料擷取至 Amazon OpenSearch Service 網域和集合。OpenSearch 擷取管道整合串流擷取基礎設施，以提供從 Kinesis 持續擷取串流記錄的高規模、低延遲方式。

主題

- Amazon Kinesis Data Streams 作為來源
- Amazon Kinesis Data Streams 跨帳戶做為來源

Amazon Kinesis Data Streams 作為來源

透過下列程序，您將了解如何設定使用 Amazon Kinesis Data Streams 做為資料來源的 OpenSearch Ingestion 管道。本節涵蓋必要的先決條件，例如建立 OpenSearch Service 網域或 OpenSearch Serverless Collection，以及逐步完成設定管道角色和建立管道的步驟。

先決條件

若要設定管道，您需要一或多個作用中的 Kinesis Data Streams。這些串流必須正在接收記錄或準備好接收來自其他來源的記錄。如需詳細資訊，請參閱 OpenSearch 擷取概觀。

設定管道

1. 建立 OpenSearch Service 網域或 OpenSearch Serverless 集合

   若要建立網域或集合，請參閱 OpenSearch Ingestion 入門。

   若要建立具有正確許可的 IAM 角色，以存取寫入資料至集合或網域，請參閱以資源為基礎的政策。
2. 設定具有 許可的管道角色

   設定您想要在管道組態中使用的管道角色，並將下列許可新增至管道組態。使用您的資訊取代*###*。

   JSON

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

```
    "Statement": [
        {
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:DescribeStreamConsumer",
                "kinesis:DescribeStreamSummary",
                "kinesis:GetRecords",
                "kinesis:GetShardIterator",
                "kinesis:ListShards",
                "kinesis:ListStreams",
                "kinesis:ListStreamConsumers",
                "kinesis:RegisterStreamConsumer",
                "kinesis:SubscribeToShard"
            ],
            "Resource": [
                "arn:aws:kinesis:us-east-1:111122223333:stream/stream-name"
            ]
        }
    ]
}
```

如果在串流上啟用伺服器端加密，下列 AWS KMS 政策允許 解密記錄。使用您的資訊取代#####。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "allowDecryptionOfCustomManagedKey",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id"
```

```
        }
    ]
}
```

為了讓管道將資料寫入網域，網域必須具有網域層級存取政策，允許 sts_role_arn 管道角色存取它。

下列範例是網域存取政策，允許在上一個步驟 (pipeline-role) 中建立的管道角色將資料寫入ingestion-domain網域。使用您的資訊取代######。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::your-account-id:role/pipeline-role"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:AWS ##:account-id:domain/domain-name/*"
    }
  ]
}
```

3. 建立管道

設定指定 Kinesis-data-streams 做為來源的 OpenSearch Ingestion 管道。您可以在 OpenSearch 擷取主控台中找到可用於建立此類管道的準備就緒藍圖。（選用）若要使用 建立管道 AWS CLI，您可以使用名為 "**AWS-KinesisDataStreamsPipeline**" 的藍圖。使用您的資訊取代######。

```
version: "2"
kinesis-pipeline:
  source:
    kinesis_data_streams:
      acknowledgments: true
      codec:
        # Based on whether kinesis records are aggregated or not, you could choose
 json, newline or ndjson codec for processing the records.
```

```
        # JSON codec supports parsing nested CloudWatch Events into individual log
entries that will be written as documents into OpenSearch.
        # json:
          # key_name: "logEvents"
          # These keys contain the metadata sent by CloudWatch Subscription Filters
          # in addition to the individual log events:
          # include_keys: [ 'owner', 'logGroup', 'logStream' ]
        newline:
      streams:
        - stream_name: "stream name"
          # Enable this if ingestion should start from the start of the stream.
          # initial_position: "EARLIEST"
          # checkpoint_interval: "PT5M"
          # Compression will always be gzip for CloudWatch, but will vary for other
sources:
          # compression: "gzip"
        - stream_name: "stream name"
          # Enable this if ingestion should start from the start of the stream.
          # initial_position: "EARLIEST"
          # checkpoint_interval: "PT5M"
          # Compression will always be gzip for CloudWatch, but will vary for other
sources:
          # compression: "gzip"

        # buffer_timeout: "1s"
        # records_to_accumulate: 100
        # Change the consumer strategy to "polling". Default consumer strategy will
use enhanced "fan-out" supported by KDS.
        # consumer_strategy: "polling"
        # if consumer strategy is set to "polling", enable the polling config
below.
        # polling:
          # max_polling_records: 100
          # idle_time_between_reads: "250ms"
      aws:
        # Provide the Role ARN with access to Amazon Kinesis Data Streams. This
role should have a trust relationship with osis-pipelines.amazonaws.com
        sts_role_arn: "arn:aws:iam::111122223333:role/Example-Role"
        # Provide the AWS ## of the Data Stream.
        region: "us-east-1"

  sink:
    - opensearch:
        # Provide an Amazon OpenSearch Serverless domain endpoint
```

```
        hosts: [ "https://search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-
east-1.es.amazonaws.com" ]
        index: "index_${getMetadata(\"stream_name\")}"
        # Ensure adding unique document id as a combination of the metadata
 attributes available.
        document_id: "${getMetadata(\"partition_key\")}_
${getMetadata(\"sequence_number\")}_${getMetadata(\"sub_sequence_number\")}"
        aws:
          # Provide a Role ARN with access to the domain. This role should have a
 trust relationship with osis-pipelines.amazonaws.com
          sts_role_arn: "arn:aws:iam::111122223333:role/Example-Role"
          # Provide the AWS ## of the domain.
          region: "us-east-1"
          # Enable the 'serverless' flag if the sink is an Amazon OpenSearch
 Serverless collection
          serverless: false
          # serverless_options:
            # Specify a name here to create or update network policy for the
 serverless collection
            # network_policy_name: "network-policy-name"
        # Enable the 'distribution_version' setting if the OpenSearch Serverless
 domain is of version Elasticsearch 6.x
        # distribution_version: "es6"
        # Enable and switch the 'enable_request_compression' flag if the default
 compression setting is changed in the domain. See https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/gzip.html
        # enable_request_compression: true/false
        # Optional: Enable the S3 DLQ to capture any failed requests in an S3
 bucket. Delete this entire block if you don't want a DLQ.
        dlq:
          s3:
            # Provide an S3 bucket
            bucket: "your-dlq-bucket-name"
            # Provide a key path prefix for the failed requests
            # key_path_prefix: "kinesis-pipeline/logs/dlq"
            # Provide the region of the bucket.
            region: "us-east-1"
            # Provide a Role ARN with access to the bucket. This role should have a
 trust relationship with osis-pipelines.amazonaws.com
            sts_role_arn: "arn:aws:iam::111122223333:role/Example-Role"
```

組態選項

如需 Kinesis 組態選項，請參閱 OpenSearch 文件中的組態選項。

可用的中繼資料屬性

- stream_name – 從中擷取記錄的 Kinesis Data Streams 名稱
- partition_key – 正在擷取的 Kinesis Data Streams 記錄的分割區索引鍵
- sequence_number – 正在擷取的 Kinesis Data Streams 記錄序號
- sub_sequence_number – 正在擷取的 Kinesis Data Streams 記錄的子序號

4. （選用）設定 Kinesis Data Streams 管道的建議運算單位 (OCUs)

   OpenSearch Kinesis Data Streams 來源管道也可以設定為從多個串流擷取串流記錄。為了加快擷取速度，我們建議您為每個新增的串流新增額外的運算單位。

資料一致性

OpenSearch Ingestion end-to-end確認，以確保資料耐久性。當管道從 Kinesis 讀取串流記錄時，它會根據與串流相關聯的碎片動態分配讀取串流記錄的工作。在擷取 OpenSearch 網域或集合中的所有記錄之後，管道會在收到確認時自動檢查點串流。這可避免重複處理串流記錄。

若要根據串流名稱建立索引，請在 opensearch sink 區段中將索引定義為 "index_${getMetadata(\"stream_name\")}"。

Amazon Kinesis Data Streams 跨帳戶做為來源

您可以使用 Amazon Kinesis Data Streams 跨帳戶授予存取權，以便 OpenSearch Ingestion 管道可以存取另一個帳戶中的 Kinesis Data Streams 作為來源。完成下列步驟以啟用跨帳戶存取：

設定跨帳戶存取

1. 在具有 Kinesis 串流的帳戶中設定資源政策

   使用您的資訊取代######。

   JSON

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

```
        {
            "Sid": "StreamReadStatementID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/Pipeline-Role"
            },
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:GetRecords",
                "kinesis:GetShardIterator",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:us-east-1:444455556666:stream/
stream-name"
        },
        {
            "Sid": "StreamEFOReadStatementID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/Pipeline-Role"
            },
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:us-
east-1:444455556666:stream/stream-name/consumer/consumer-name"
        }
    ]
}
```

2. （選用）設定消費者和消費者資源政策

這是選用步驟，只有在您計劃使用增強型廣發消費者策略來讀取串流記錄時才需要。如需詳細資訊，請參閱使用專用輸送量開發增強型廣發消費者。

a. 設定取用者

若要重複使用現有的消費者，您可以略過此步驟。如需詳細資訊，請參閱《Amazon Kinesis Data Streams API 參考》中的 RegisterStreamConsumer。

在下列範例 CLI 命令中，將#####取代為您自己的資訊。

Example：CLI 命令範例

```
aws kinesis register-stream-consumer \
--stream-arn "arn:aws:kinesis:AWS ##:account-id:stream/stream-name" \
--consumer-name consumer-name
```

b. 設定消費者資源政策

在下列陳述式中，將#####取代為您自己的資訊。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ConsumerEFOReadStatementID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/Pipeline-Role"
            },
            "Action": [
                "kinesis:DescribeStreamConsumer",
                "kinesis:SubscribeToShard"
            ],
            "Resource": "arn:aws:kinesis:us-
east-1:444455556666:stream/stream-1/consumer/consumer-name"
        }
    ]
}
```

3. 管道組態

對於跨帳戶擷取，kinesis_data_streams請為每個串流在 下新增下列屬性：

- stream_arn - 屬於串流所在帳戶之串流的 arn

- consumer_arn - 這是選用屬性，如果選擇預設增強型廣發消費者策略，則必須指定此屬性。指定此欄位的實際消費者來源。使用您的資訊取代######。

```
version: "2"
     kinesis-pipeline:
       source:
         kinesis_data_streams:
           acknowledgments: true
           codec:
             newline:
           streams:
             - stream_arn: "arn:aws:kinesis:region:stream-account-id:stream/stream-
name"
               consumer_arn: "consumer arn"
               # Enable this if ingestion should start from the start of the
 stream.
               # initial_position: "EARLIEST"
               # checkpoint_interval: "PT5M"
             - stream_arn: "arn:aws:kinesis:region:stream-account-id:stream/stream-
name"
               consumer_arn: "consumer arn"
                # initial_position: "EARLIEST"

           # buffer_timeout: "1s"
           # records_to_accumulate: 100
           # Enable the consumer strategy to "polling". Default consumer strategy
 will use enhanced "fan-out" supported by KDS.
           # consumer_strategy: "polling"
           # if consumer strategy is set to "polling", enable the polling config
 below.
           # polling:
             # max_polling_records: 100
             # idle_time_between_reads: "250ms"
         aws:
           # Provide the Role ARN with access to Kinesis. This role should have a
 trust relationship with osis-pipelines.amazonaws.com
           sts_role_arn: "arn:aws:iam::111122223333:role/Example-Role"
           # Provide the AWS ## of the domain.
           region: "us-east-1"

       sink:
```

```
        - opensearch:
            # Provide an OpenSearch Serverless domain endpoint
            hosts: [ "https://search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-
east-1.es.amazonaws.com" ]
            index: "index_${getMetadata(\"stream_name\")}"
            # Mapping for documentid based on partition key, shard sequence number
 and subsequence number metadata attributes
            document_id: "${getMetadata(\"partition_key\")}_
${getMetadata(\"sequence_number\")}_${getMetadata(\"sub_sequence_number\")}"
            aws:
              # Provide a Role ARN with access to the domain. This role should
 have a trust relationship with osis-pipelines.amazonaws.com
              sts_role_arn: "arn:aws:iam::111122223333:role/Example-Role"
              # Provide the AWS ## of the domain.
              region: "us-east-1"
              # Enable the 'serverless' flag if the sink is an OpenSearch
 Serverless collection
              serverless: false
                # serverless_options:
                # Specify a name here to create or update network policy for the
 serverless collection
              # network_policy_name: network-policy-name
            # Enable the 'distribution_version' setting if the OpenSearch
 Serverless domain is of version Elasticsearch 6.x
            # distribution_version: "es6"
            # Enable and switch the 'enable_request_compression' flag if
 the default compression setting is changed in the domain. See https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/gzip.html
            # enable_request_compression: true/false
            # Optional: Enable the S3 DLQ to capture any failed requests in an S3
 bucket. Delete this entire block if you don't want a DLQ.
            dlq:
              s3:
                # Provide an Amazon S3 bucket
                bucket: "your-dlq-bucket-name"
                # Provide a key path prefix for the failed requests
                # key_path_prefix: "alb-access-log-pipeline/logs/dlq"
                # Provide the AWS ## of the bucket.
                region: "us-east-1"
                # Provide a Role ARN with access to the bucket. This role should
 have a trust relationship with osis-pipelines.amazonaws.com
                sts_role_arn: "arn:aws:iam::111122223333:role/Example-Role"
```

4. OSI 管道角色 Kinesis 資料串流

   a. IAM 政策

   將下列政策新增至管道角色。使用您的資訊取代######。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStreamConsumer",
                "kinesis:SubscribeToShard"
            ],
            "Resource": [
            "arn:aws:kinesis:us-east-1:111122223333:stream/my-stream"
            ]
        },
        {
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:DescribeStreamSummary",
                "kinesis:GetRecords",
                "kinesis:GetShardIterator",
                "kinesis:ListShards",
                "kinesis:ListStreams",
                "kinesis:ListStreamConsumers",
                "kinesis:RegisterStreamConsumer"
            ],
            "Resource": [
                "arn:aws:kinesis:us-east-1:111122223333:stream/my-stream"
            ]
        }
    ]
}
```

b. 信任政策

若要從串流帳戶擷取資料，您需要在管道擷取角色和串流帳戶之間建立信任關係。將下列項目新增至管道角色。使用您的資訊取代######。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::111122223333:root"
      },
     "Action": "sts:AssumeRole"
  }]
}
```

後續步驟

將資料匯出至管道後，您可以從設定為管道接收器的 OpenSearch Service 網域進行查詢。下列資源可協助您開始使用：

- 可觀測性
- the section called “追蹤”
- 可觀測性

搭配 使用 OpenSearch 擷取管道 AWS Lambda

使用AWS Lambda 處理器透過自訂程式碼，從 OpenSearch Ingestion 支援的任何來源或目的地豐富資料。使用 Lambda 處理器，您可以套用自己的資料轉換或擴充，然後將處理的事件傳回管道以進行進一步處理。此處理器可啟用自訂資料處理，並讓您完全控制資料在流經管道前的處理方式。

> **Note**
>
> Lambda 處理器處理之單一事件的承載大小限制為 5 MB。此外，Lambda 處理器僅支援 JSON 陣列格式的回應。

先決條件

使用 Lambda 處理器建立管道之前，請建立下列資源：

- 可豐富和轉換來源資料的 AWS Lambda 函數。如需說明，請參閱建立您的第一個 Lambda 函數。
- 將成為管道接收器的 OpenSearch Service 網域或 OpenSearch Serverless 集合。如需詳細資訊，請參閱the section called “建立 OpenSearch Service 網域”及the section called “建立集合”。
- 管道角色，其中包含寫入網域或集合目的地的許可。如需詳細資訊，請參閱the section called “管道角色”。

  管道角色還需要附加的許可政策，允許它叫用管道組態中指定的 Lambda 函數。例如：

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "allowinvokeFunction",
            "Effect": "Allow",
            "Action": [
                "lambda:invokeFunction",
                "lambda:InvokeAsync",
                "lambda:ListFunctions"
            ],
            "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:function-name"

        }
    ]
}
```

建立管道

若要使用 AWS Lambda 做為處理器，請設定 OpenSearch Ingestion 管道並指定 aws_lambda做為處理器。您也可以使用AWS Lambda 自訂擴充藍圖來建立管道。如需詳細資訊，請參閱the section called “使用藍圖”。

下列範例管道會從 HTTP 來源接收資料、使用日期處理器和 AWS Lambda 處理器來充實資料，並將處理的資料擷取至 OpenSearch 網域。

```
version: "2"
lambda-processor-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
  processor:
      - date:
        destination: "@timestamp"
        from_time_received: true
    - aws_lambda:
        function_name: "my-lambda-function"

        tags_on_failure: ["lambda_failure"]
        batch:
            key_name: "events"
        aws:
          region: region
  sink:
    - opensearch:
        hosts: [ "https://search-mydomain.us-east-1es.amazonaws.com" ]
        index: "table-index"
        aws:
          region: "region"
          serverless: false
```

下列範例 AWS Lambda 函數會將新的鍵/值對 ("transformed": "true") 新增至所提供事件陣列中的每個元素，然後傳回修改後的版本，以轉換傳入的資料。

```
import json

def lambda_handler(event, context):
    input_array = event.get('events', [])
    output = []
```

```
    for input in input_array:
        input["transformed"] = "true";
        output.append(input)

    return output
```

批次處理

管道會將批次事件傳送至 Lambda 處理器，並動態調整批次大小，以確保其保持在 5 MB 的限制以下。

以下是管道批次的範例：

```
batch:
    key_name: "events"

input_arrary = event.get('events', [])
```

> **Note**
>
> 當您建立管道時，請確定 Lambda 處理器組態中的 key_name選項符合 Lambda 處理常式中的事件金鑰。

條件式篩選

條件式篩選可讓您根據事件資料中的特定條件，控制 AWS Lambda 處理器調用 Lambda 函數的時間。當您想要選擇性地處理某些類型的事件，同時忽略其他事件時，此功能特別有用。

下列範例組態使用條件式篩選：

```
processors:
  - aws_lambda:
      function_name: "my-lambda-function"
      aws:
        region: "region"
      lambda_when: "/sourceIp == 10.10.10.10"
```

使用 Amazon OpenSearch 擷取移轉網域和集合之間的資料

您可以使用 OpenSearch Ingestion 管道，在 Amazon OpenSearch Service 網域或 OpenSearch Serverless VPC 集合之間遷移資料。若要這樣做，您可以設定管道，其中將一個網域或集合設定為來源，並將另一個網域或集合設定為目的地。這樣可以有效地將您的資料從一個網域或集合遷移到另一個網域或集合。

若要遷移資料，您必須擁有下列資源：

- 來源 OpenSearch Service 網域或 OpenSearch Serverless VPC 集合。此網域或集合包含您要遷移的資料。如果您使用的是網域，則必須執行 OpenSearch 1.0 版或更新版本，或 Elasticsearch 7.4 版或更新版本。網域還必須具有授予管道角色適當許可的存取政策。
- 您要將資料遷移到的個別網域或 VPC 集合。此網域或集合將充當管道接收器。
- OpenSearch Ingestion 將用於讀取和寫入集合或網域的管道角色。您可以在管道組態中包含此角色的 Amazon Resource Name (ARN)。如需詳細資訊，請參閱下列資源：
  - the section called “授予管道對網域的存取權”
  - the section called “授予管道對集合的存取權”

主題

- 限制
- OpenSearch Service 做為來源
- 指定多個 OpenSearch Service 網域目的地
- 將資料遷移至 OpenSearch Serverless VPC 集合

限制

當您將 OpenSearch Service 網域或 OpenSearch Serverless 集合指定為目的地時，適用下列限制：

- 管道無法寫入多個 VPC 網域。
- 您只能在使用 VPC 存取的 OpenSearch Serverless 集合之間遷移資料。不支援公有集合。
- 您無法在單一管道組態中指定 VPC 和公有網域的組合。
- 在單一管道組態中，您最多可以有 20 個非管道接收器。
- 您可以在單一管道組態 AWS 區域 中指定最多三個不同的目的地。

- 如果任何目的地停機太久，或佈建的容量不足以接收傳入資料，具有多個目的地的管道可能會隨著時間降低處理速度。

OpenSearch Service 做為來源

您指定為來源的網域或集合是資料遷移的來源。

在 IAM 中建立管道角色

若要建立 OpenSearch 擷取管道，您必須先建立管道角色，以授予網域或集合之間的讀取和寫入存取權。請依下列步驟操作：

1. 在 IAM 中建立新的許可政策以連接到管道角色。請確定您允許從來源讀取和寫入接收器的許可。如需設定 OpenSearch Service 網域 IAM 管道許可的詳細資訊，請參閱 the section called “授予管道對網域的存取權”和 the section called “授予管道對集合的存取權”。
2. 在管道角色中指定要從來源讀取的下列許可：

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"es:ESHttpGet",
         "Resource":[
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/",
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/_cat/
indices",
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/_search",
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/_search/
scroll",
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/*/_search"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpPost",
         "Resource":[
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/*/_search/
point_in_time",
```

```
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/*/_search/
scroll"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpDelete",
         "Resource":[
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/_search/
point_in_time",
            "arn:aws:es:us-east-1:111122223333:domain/domain-name/_search/
scroll"
         ]
      }
   ]
}
```

建立管道

將政策連接至管道角色後，請使用 AWSOpenSearchDataMigrationPipeline 遷移藍圖來建立管道。此藍圖包含用於在 OpenSearch Service 網域或集合之間遷移資料的預設組態。如需詳細資訊，請參閱the section called “使用藍圖”。

> **Note**
>
> OpenSearch Ingestion 使用您的來源網域版本和分佈來確定要用於遷移的機制。有些版本支援 `point_in_time`選項。OpenSearch Serverless 使用 `search_after`選項，因為它不支援 `point_in_time`或 `scroll`。

新的索引可能正在遷移過程中建立，或者文件可能在遷移過程中更新。因此，您可能需要對網域索引資料執行單次掃描或多次掃描，以挑選新的或更新的資料。

透過在管道組態`interval`中設定 `index_read_count`和 ，指定要執行的掃描數量。下列範例示範如何執行多次掃描：

```
scheduling:
    interval: "PT2H"
    index_read_count: 3
    start_time: "2023-06-02T22:01:30.00Z"
```

OpenSearch Ingestion 使用以下組態，以確保您的資料寫入相同的索引，並維護相同的文件 ID：

```
index: "${getMetadata(\"opensearch-index\")}"
document_id: "${getMetadata(\"opensearch-document_id\")}"
```

指定多個 OpenSearch Service 網域目的地

您可以指定多個公有 OpenSearch Service 網域做為資料的目的地。您可以使用此功能來執行條件式路由，或將傳入資料複寫至多個 OpenSearch Service 網域。您最多可以將 10 個不同的公有 OpenSearch Service 網域指定為接收器。

在下列範例中，傳入的資料會以條件路由至不同的 OpenSearch Service 網域：

```
...
  route:
    - 2xx_status: "/response >= 200 and /response < 300"
    - 5xx_status: "/response >= 500 and /response < 600"
  sink:
    - opensearch:
        hosts: [ "https://search-response-2xx.region.es.amazonaws.com" ]
        aws:
          region: "us-east-1"
        index: "response-2xx"
        routes:
          - 2xx_status
    - opensearch:
        hosts: [ "https://search-response-5xx.region.es.amazonaws.com" ]
        aws:
          region: "us-east-1"
        index: "response-5xx"
        routes:
          - 5xx_status
```

將資料遷移至 OpenSearch Serverless VPC 集合

您可以使用 OpenSearch Ingestion 將資料從來源 OpenSearch Service 網域或 OpenSearch Serverless 集合遷移至 VPC 集合目的地。您必須在管道組態中提供網路存取政策。如需將資料擷取至 OpenSearch Serverless VPC 集合的詳細資訊，請參閱 the section called “教學課程：將資料擷取至集合”。

將資料遷移至 VPC 集合

1. 建立 OpenSearch Serverless 集合。如需說明，請參閱the section called “教學課程：將資料擷取至集合”。
2. 為集合建立網路政策，指定對集合端點和儀表板端點的 VPC 存取。如需說明，請參閱the section called “網路存取”。
3. 如果您還沒有管道角色，請建立管道角色。如需說明，請參閱the section called “管道角色”。
4. 建立管道。如需說明，請參閱the section called “使用藍圖”。

使用 AWS SDKs 與 Amazon OpenSearch Ingestion 互動

本節包含如何使用 AWS SDKs 與 Amazon OpenSearch Ingestion 互動的範例。程式碼範例示範如何建立網域和管道，然後將資料擷取至管道。

主題

- Python

Python

下列範例指令碼使用 適用於 Python (Boto3) 的 AWS SDK建立 IAM 管道角色、寫入資料的網域，以及擷取資料的管道。然後，它會使用 `requests` HTTP 程式庫將範例日誌檔案擷取至管道。

若要安裝所需的相依性，請執行下列命令：

```
pip install boto3
pip install botocore
pip install requests
pip install requests-auth-aws-sigv4
```

在指令碼中，*`account-id`*以您的 AWS 帳戶 ID 取代 的所有執行個體。

```
import boto3
import botocore
from botocore.config import Config
import requests
from requests_auth_aws_sigv4 import AWSSigV4
import time
```

```
# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

opensearch = boto3.client('opensearch', config=my_config)
iam = boto3.client('iam', config=my_config)
osis = boto3.client('osis', config=my_config)

domainName = 'test-domain'  # The name of the domain
pipelineName = 'test-pipeline' # The name of the pipeline

def createPipelineRole(iam, domainName):
    """Creates the pipeline role"""
    response = iam.create_policy(
        PolicyName='pipeline-policy',
        PolicyDocument=f'{{\"Version\":\"2012-10-17\",\"Statement\":[{{\"Effect\":
\"Allow\",\"Action\":\"es:DescribeDomain\",\"Resource\":\"arn:aws:es:us-east-1:account-
id:domain\/{domainName}\"}},{{\"Effect\":\"Allow\",\"Action\":\"es:ESHttp*\",\"Resource
\":\"arn:aws:es:us-east-1:account-id:domain\/{domainName}\/*\"}}]}}'
    )
    policyarn = response['Policy']['Arn']

    response = iam.create_role(
        RoleName='PipelineRole',
        AssumeRolePolicyDocument='{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"Service\":\"osis-pipelines.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}]}'
    )
    rolename=response['Role']['RoleName']

    response = iam.attach_role_policy(
        RoleName=rolename,
        PolicyArn=policyarn
    )

    print('Creating pipeline role...')
    time.sleep(10)
    print('Role created: ' + rolename)

def createDomain(opensearch, domainName):
    """Creates a domain to ingest data into"""
    response = opensearch.create_domain(
        DomainName=domainName,
        EngineVersion='OpenSearch_2.3',
```

```
        ClusterConfig={
            'InstanceType': 't2.small.search',
            'InstanceCount': 5,
            'DedicatedMasterEnabled': True,
            'DedicatedMasterType': 't2.small.search',
            'DedicatedMasterCount': 3
        },
        # Many instance types require EBS storage.
        EBSOptions={
            'EBSEnabled': True,
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies=f'{{\"Version\":\"2012-10-17\",\"Statement\":[{{\"Effect
\":\"Allow\",\"Principal\":{{\"AWS\":\"arn:aws:iam::account-id:role\/PipelineRole
\"}},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-east-1:account-id:domain\/
{domainName}\/*\"}}]}}',
        NodeToNodeEncryptionOptions={
            'Enabled': True
        }
    )
    return(response)

def waitForDomainProcessing(opensearch, domainName):
    """Waits for the domain to be active"""
    try:
        response = opensearch.describe_domain(
            DomainName=domainName
        )
        # Every 30 seconds, check whether the domain is processing.
        while 'Endpoint' not in response['DomainStatus']:
            print('Creating domain...')
            time.sleep(60)
            response = opensearch.describe_domain(
                DomainName=domainName)

        # Once we exit the loop, the domain is ready for ingestion.
        endpoint = response['DomainStatus']['Endpoint']
        print('Domain endpoint ready to receive data: ' + endpoint)
        createPipeline(osis, endpoint)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found.')
```

```
        else:
            raise error

def createPipeline(osis, endpoint):
    """Creates a pipeline using the domain and pipeline role"""
    try:
        definition = f'version: \"2\"\nlog-pipeline:\n  source:\n    http:\n      path:
 \"/${{pipelineName}}/logs\"\n  processor:\n    - date:\n        from_time_received:
 true\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://{endpoint}\" ]\n        index: \"application_logs\"\n        aws:\n
   region: \"us-east-1\"'
        response = osis.create_pipeline(
            PipelineName=pipelineName,
            MinUnits=4,
            MaxUnits=9,
            PipelineConfigurationBody=definition,
            PipelineRoleArn="arn:aws:iam::account-id:role/PipelineRole"
        )

        response = osis.get_pipeline(
                PipelineName=pipelineName
        )

        # Every 30 seconds, check whether the pipeline is active.
        while response['Pipeline']['Status'] == 'CREATING':
            print('Creating pipeline...')
            time.sleep(30)
            response = osis.get_pipeline(
                PipelineName=pipelineName)

        # Once we exit the loop, the pipeline is ready for ingestion.
        ingestionEndpoint = response['Pipeline']['IngestEndpointUrls'][0]
        print('Pipeline ready to ingest data at endpoint: ' + ingestionEndpoint)
        ingestData(ingestionEndpoint)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceAlreadyExistsException':
            print('Pipeline already exists.')
            response = osis.get_pipeline(
                PipelineName=pipelineName
            )
            ingestionEndpoint = response['Pipeline']['IngestEndpointUrls'][0]
            ingestData(ingestionEndpoint)
        else:
```

```
            raise error


def ingestData(ingestionEndpoint):
    """Ingests a sample log file into the pipeline"""
    endpoint = 'https://' + ingestionEndpoint
    r = requests.request('POST', f'{endpoint}/log-pipeline/logs',

 data='[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","requ
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]',
    auth=AWSSigV4('osis'))
    print('Ingesting sample log file into pipeline')
    print('Response: ' + r.text)

def main():
    createPipelineRole(iam, domainName)
    createDomain(opensearch, domainName)
    waitForDomainProcessing(opensearch, domainName)

if __name__ == "__main__":
    main()
```

Amazon OpenSearch 擷取中的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 Cloud AWS 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。在 AWS 合規計畫中，第三方稽核員會定期測試並驗證我們的安全功效。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 OpenSearch Ingestion 時套用共同責任模型。下列主題說明如何設定 OpenSearch Ingestion 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 OpenSearch Ingestion 資源。

主題

- 設定 Amazon OpenSearch Ingestion 管道的 VPC 存取
- 設定跨帳戶擷取的 OpenSearch 擷取管道
- Amazon OpenSearch Ingestion 的 Identity and Access Management
- 使用 記錄 Amazon OpenSearch Ingestion API 呼叫 AWS CloudTrail
- Amazon OpenSearch 擷取和介面端點 API (AWS PrivateLink)

設定 Amazon OpenSearch Ingestion 管道的 VPC 存取

您可以使用介面 VPC 端點存取 Amazon OpenSearch Ingestion 管道。VPC 是您的專用虛擬網路 AWS 帳戶。它在邏輯上與 AWS 雲端中的其他虛擬網路隔離。透過 VPC 端點存取管道可啟用 OpenSearch Ingestion 與 VPC 內其他服務之間的安全通訊，而不需要網際網路閘道、NAT 裝置或 VPN 連接。所有流量都會安全地保留在 AWS 雲端中。

OpenSearch Ingestion 透過建立採用 技術的介面端點來建立此私有連線 AWS PrivateLink。我們在您在管道建立期間指定的每個子網路中建立端點網路介面。這些是請求者管理的網路介面，可做為目的地為 OpenSearch Ingestion 管道之流量的進入點。您也可以自行選擇建立和管理介面端點。

使用 VPC 可讓您在 VPC 邊界內透過 OpenSearch Ingestion 管道強制執行資料流程，而不是透過公有網際網路。不在 VPC 內的管道會透過公開端點和網際網路傳送和接收資料。

具有 VPC 存取的管道可以寫入公有或 VPC OpenSearch Service 網域，以及寫入公有或 VPC OpenSearch Serverless 集合。

主題

- 考量事項
- 限制
- 先決條件
- 設定管道的 VPC 存取
- 自我管理 VPC 端點
- VPC 存取適用的服務連結角色

考量事項

當您設定管道的 VPC 存取時，請考慮下列事項。

- 管道不需要與其接收器位於相同的 VPC 中。您也不需要在兩個 VPCs之間建立連線。OpenSearch Ingestion 會為您處理連線。
- 您只能為管道指定一個 VPC。
- 與公有管道不同，VPC 管道必須與寫入的網域或集合目的地位於相同 AWS 區域 位置。您可以為管道設定 S3 來源，以便寫入跨區域。
- 您可以選擇將管道部署到 VPC 的一個、兩個或三個子網路。子網路會分散至您的 Ingestion OpenSearch Compute Units (OCUs) 部署所在的相同可用區域。
- 如果您只將管道部署在一個子網路中，且可用區域故障，您將無法擷取資料。為了確保高可用性，建議您使用兩個或三個子網路設定管道。
- 指定安全群組是選用的。如果您未提供安全群組，OpenSearch Ingestion 會使用 VPC 中指定的預設安全群組。

限制

具有 VPC 存取的管道有下列限制。

- 您無法在建立管道後變更管道的網路組態。如果您在 VPC 中啟動管道，您稍後無法將其變更為公有端點，反之亦然。
- 您可以使用介面 VPC 端點或公有端點啟動管道，但無法同時執行兩者。建立管道時，您必須選擇其中一個。
- 使用 VPC 存取佈建管道之後，您無法將其移至不同的 VPC，也無法變更其子網路或安全群組設定。
- 如果您的管道寫入使用 VPC 存取的網域或集合目的地，您無法在管道建立後返回並變更目的地 (VPC 或公有）。您必須使用新的接收器刪除並重新建立管道。您仍然可以從公有接收器切換到具有 VPC 存取的接收器。
- 您無法提供 VPC 管道的跨帳戶擷取存取權。

先決條件

您必須先執行下列動作，才能使用 VPC 存取佈建管道：

- 建立 VPC

  若要建立 VPC，您可以使用 Amazon VPC 主控台、CLI AWS 或其中一個 AWS SDKs。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的使用 VPC。如果您已有 VPC，則可以略過此步驟。
- 預留 IP 地址

OpenSearch Ingestion 會在您在管道建立期間指定的每個子網路中放置彈性網路界面。每個網路界面都與 IP 地址關聯。您必須為每個子網路為網路介面預留一個 IP 地址。

設定管道的 VPC 存取

您可以在 OpenSearch Service 主控台或使用 啟用管道的 VPC 存取 AWS CLI。

主控台

您可以在管道建立期間設定 VPC 存取。在來源網路選項下，選擇 VPC 存取並設定下列設定：

設定	Description
端點管理	選擇您要自行建立 VPC 端點，還是讓 OpenSearch Ingestion 為您建立端點。
VPC	選擇您想使用的虛擬私有雲端 (VPC) ID。VPC 和管道必須位於相同的 中 AWS 區域。
子網路	選擇一或多個子網路。OpenSearch Service 會在子網路中放置 VPC 端點和彈性網路介面。
安全群組	選擇一或多個 VPC 安全群組，允許所需的應用程式在連接埠 (80 或 443) 和管道公開的通訊協定 (HTTP 或 HTTPs) 上連接 OpenSearch 擷取管道。
VPC 連接選項	如果您的來源需要跨 VPC 通訊，例如 Amazon DocumentDB、自我管理的 OpenSearch 或 Confluent Kafka，OpenSearch Ingestion 會在您指定的子網路中建立彈性網路界面 (ENIs)，以連線至這些來源。OpenSearch Ingestion 使用每個可用區域中ENIs 來到達指定的來源。連接至 VPC 選項會將 OpenSearch 擷取資料平面 VPC 連接至您指定的 VPC。 選取受管 VPC 的 CIDR 保留以部署網路介面。

CLI

若要使用 設定 VPC 存取 AWS CLI，請指定 `--vpc-options` 參數：

```
aws osis create-pipeline \
  --pipeline-name vpc-pipeline \
```

```
  --min-units 4 \
  --max-units 10 \
  --vpc-options
 SecurityGroupIds={sg-12345678,sg-9012345},SubnetIds=subnet-1212234567834asdf \
  --pipeline-configuration-body "file://pipeline-config.yaml"
```

自我管理 VPC 端點

建立管道時，您可以使用端點管理來建立具有自我管理端點或服務受管端點的管道。端點管理是選用的，預設為 OpenSearch Ingestion 管理的端點。

若要在 中使用自我管理 VPC 端點建立管道 AWS 管理主控台，請參閱使用 OpenSearch Service 主控台建立管道。若要在 中建立具有自我管理 VPC 端點的管道 AWS CLI，您可以在 create-pipeline 命令中使用 `--vpc-options` 參數：

```
--vpc-options SubnetIds=subnet-abcdef01234567890,VpcEndpointManagement=CUSTOMER
```

您可以在指定端點服務時，自行建立管道的端點。若要尋找您的端點服務，請使用 get-pipeline 命令，這會傳回類似以下的回應：

```
"vpcEndpointService" : "com.amazonaws.osis.us-east-1.pipeline-
id-1234567890abcdef1234567890",
"vpcEndpoints" : [
  {
    "vpcId" : "vpc-1234567890abcdef0",
    "vpcOptions" : {
      "subnetIds" : [ "subnet-abcdef01234567890", "subnet-021345abcdef6789" ],
      "vpcEndpointManagement" : "CUSTOMER"
    }
  }
```

使用`vpcEndpointService`回應中的 ，透過 AWS 管理主控台 或 建立 VPC 端點 AWS CLI。

如果您使用自我管理的 VPC 端點，則必須在 VPC `enableDnsHostnames`中啟用 DNS 屬性`enableDnsSupport`和 。請注意，如果您有管道具有停止和重新啟動的自我管理端點，則必須在帳戶中重新建立 VPC 端點。

VPC 存取適用的服務連結角色

服務連結角色是一個唯一的 IAM 角色類型，它將許可委派給服務，以便服務可代表您建立和管理資源。如果您選擇服務受管 VPC 端點，OpenSearch Ingestion 需要稱為

AWSServiceRoleForAmazonOpenSearchIngestionService 的服務連結角色，才能存取您的 VPC、建立管道端點，並將網路介面放置在 VPC 的子網路中。

如果您選擇自我管理 VPC 端點，OpenSearch Ingestion 需要稱為 AWSServiceRoleForOpensearchIngestionSelfManagedVpce 的服務連結角色。如需這些角色、其許可以及如何刪除這些角色的詳細資訊，請參閱 the section called “管道建立角色”。

OpenSearch Ingestion 會自動建立角色。若要讓此自動建立成功，在帳戶中建立第一個管道的使用者必須具有 `iam:CreateServiceLinkedRole`動作的許可。如需進一步了解，請參閱 IAM 使用者指南中的服務連結角色許可。您可以在建立角色之後，在 AWS Identity and Access Management (IAM) 主控台中檢視角色。

設定跨帳戶擷取的 OpenSearch 擷取管道

對於 HTTP 和 OTel 等推送型來源，Amazon OpenSearch Ingestion 可讓您將管道 AWS 帳戶 從虛擬私有雲端 (VPC) 跨 共用到個別 VPC 中的管道端點。與其組織中的其他團隊共用分析的團隊會使用此功能，以更簡化的方式共用日誌分析。

本節使用以下術語：

- 管道擁有者 - 擁有和管理 OpenSearch Ingestion 管道的帳戶。只有一個帳戶可以擁有管道。
- 連線帳戶 — 連線至 並使用共用管道的帳戶。多個帳戶可以連接到相同的管道。

若要設定 VPCs 以跨 共用 OpenSearch Ingestion 管道 AWS 帳戶，請完成下列任務，如此處所述：

- （管道擁有者）授予連線帳戶對管道的存取權
- （連線帳戶）為每個連線的 VPC 建立管道端點連線

開始之前

設定 VPCs以跨管道共用 OpenSearch Ingestion 管道之前 AWS 帳戶，請完成下列任務：

任務	詳細資訊
建立一或多個 OpenSearch 擷取管道	將最小 OpenSearch 運算單位 (OSUs) 設定為 2 或更高。如需詳細資訊，請參閱建立 Amazon OpenSearch Ingestion 管道。如需更新管道的

任務	詳細資訊
	資訊，請參閱 更新 Amazon OpenSearch 擷取管道。
為 OpenSearch 擷取建立一或多個 VPCs	若要啟用跨帳戶管道共用，管道和管道端點涉及的任何 VPC 都必須設定下列 DNS 值： • `enableDnsSupport=true` • `enableDnsHostnames=true` 如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 的 DNS 屬性。

授予連線帳戶對管道的存取權

本節中的程序說明如何使用 OpenSearch Service 主控台和 AWS CLI ，透過建立資源政策來設定跨帳戶管道存取。資源政策可讓管道擁有者指定可存取管道的其他帳戶。建立之後，只要管道存在或直到刪除政策，管道政策就會存在。

> **Note**
>
> 資源政策不會取代使用 IAM 許可的標準 OpenSearch Ingestion 授權。資源政策是用於啟用跨帳戶管道存取的新增授權機制。

主題

- 授予連線帳戶對管道的存取權 （ 主控台 ）
- 授予連線帳戶對管道的存取權 (CLI)

授予連線帳戶對管道的存取權 （ 主控台 ）

使用下列程序，透過 Amazon OpenSearch Service 主控台授予連線帳戶對管道的存取權。

建立管道端點連線

1. 在 Amazon OpenSearch Service 主控台的導覽窗格中，展開擷取，然後選取管道。

2. 在管道區段中，選擇您要授予連線帳戶存取權的管道名稱。
3. 選擇 VPC 端點索引標籤。
4. 在授權委託人區段中，選擇授權帳戶。
5. 在 AWS 帳戶 ID 欄位中，輸入 12 位數的帳戶 ID，然後選取授權。

授予連線帳戶對管道的存取權 (CLI)

使用下列程序，透過 授予連線帳戶對管道的存取權 AWS CLI。

授予連線帳戶對管道的存取權

1. 更新至最新版本的 AWS CLI (2.0 版）。如需詳細資訊，請參閱安裝或更新至最新版本的 AWS CLI。
2. 在帳戶中開啟 CLI， AWS 區域 並使用您要共用的管道。
3. 執行下列命令來建立管道的資源政策。此政策提供管道的 `osis:CreatePipelineEndpoint` 許可。政策包含參數，您可以在其中列出要允許的 AWS 帳戶 IDs。

> **Note**
>
> 在下列命令中，您必須僅提供 12 位數帳戶 ID，以使用簡短形式的帳戶 ID。使用 ARN 將無法運作。您還必須在 的 CLI 參數和 下的政策 JSON `resource-arn`中提供管道的 Amazon Resource Name (ARN)`Resource`，如下所示。

```
aws --region region osis put-resource-policy \
  --resource-arn arn:aws:osis:region:pipeline-owner-account-ID:pipeline/pipeline-name
  --policy 'IAM-policy'
```

針對 *IAM-policy* 使用類似下列的政策

JSON

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

```
 "Sid": "AllowAccess",
 "Effect": "Allow",
 "Principal": {
 "AWS": [
 "111122223333",
 "444455556666"
 ]
 },
 "Action":
 "osis:CreatePipelineEndpoint",
 "Resource": "arn:aws:osis:us-east-1:123456789012:pipeline/pipeline-name"
 }
 ]
}
```

為每個連線的 VPC 建立管道端點連線

管道擁有者使用先前的程序授予其 VPC 中管道的存取權後，連線帳戶中的使用者會在其 VPC 中建立管道端點。本節包含使用 OpenSearch Service 主控台和 建立端點的程序 AWS CLI。當您建立端點時，OpenSearch Ingestion 會執行下列動作：

- 如果 AWSServiceRoleForAmazonOpenSearchIngestionService 服務連結角色不存在，請在您的帳戶中建立該角色。此角色授予連線帳戶中的使用者呼叫 CreatePipelineEndpoint API 動作的許可。
- 建立管道端點。
- 設定管道端點，從管道擁有者 VPC 中的共用管道擷取資料。

主題

- 建立管道端點連線（主控台）
- 建立管道端點連線 (CLI)

建立管道端點連線（主控台）

使用下列程序，透過 OpenSearch Service 主控台建立管道端點連線。

建立管道端點連線

1. 在 Amazon OpenSearch Service 主控台的導覽窗格中，展開擷取，然後選取 VPC 端點。
2. 在 VPC 端點頁面中，選擇建立。

3. 針對管道位置，選擇選項。如果您選擇目前帳戶，請從清單中選擇管道。如果您選擇跨帳戶，請在欄位中指定管道 ARN。管道擁有者必須已授予管道的存取權，如中所述授予連線帳戶對管道的存取權。
4. 在 VPC 設定區段中，針對 VPC，從清單中選擇 VPC。
5. 針對 Subnet (子網路)，請選擇子網路。
6. 針對安全群組，選擇群組。
7. 選擇建立端點。

等待您建立的端點狀態轉換為 ACTIVE。一旦管道為 ACTIVE，您會看到名為 的新欄位ingestEndpointUrl。使用此端點來存取管道，並使用 FluentBit 等用戶端擷取資料。如需使用 FluentBit 擷取資料的詳細資訊，請參閱 搭配 Fluent Bit 使用 OpenSearch 擷取管道。

> **Note**
>
> ingestEndpointUrl 是所有連線帳戶的相同 URL。

建立管道端點連線 (CLI)

使用下列程序，使用 建立管道端點連線 AWS CLI。

建立管道端點連線

1. 如果您尚未更新至最新版本的 AWS CLI (2.0 版) 。如需詳細資訊，請參閱安裝或更新至最新版本的 AWS CLI。
2. 使用共用管道在 中的連線帳戶中開啟 AWS 區域 CLI。
3. 執行下列命令來建立管道端點。

> **Note**
>
> 您必須為連線帳戶 VPC 提供至少一個子網路和一個安全群組。安全群組必須包含連接埠 443，並在連線帳戶 VPC 中支援用戶端。

```
aws osis --region region create-pipeline-endpoint \
  --pipeline-arn arn:aws:osis:region:connecting-account-ID:pipeline/shared-
pipeline-name
```

```
  --vpc-options SecurityGroupIds={sg-security-group-ID-1,sg-security-group-
ID-2},SubnetIds=subnet-subnet-ID
```

4. 執行下列命令，列出在上一個命令中指定的區域中的端點：

```
aws osis --region region list-pipeline-endpoints
```

等待您建立的端點狀態轉換為 ACTIVE。一旦管道為 ACTIVE，您會看到名為 的新欄位`ingestEndpointUrl`。使用此端點來存取管道，並使用 FluentBit 等用戶端擷取資料。如需使用 FluentBit 擷取資料的詳細資訊，請參閱 搭配 Fluent Bit 使用 OpenSearch 擷取管道。

> **Note**
>
> `ingestEndpointUrl` 是所有連線帳戶的相同 URL。

移除管道端點

如果您不想再提供共用管道的存取權，您可以使用下列其中一種方法移除管道端點：

- 刪除管道端點（連線帳戶）。
- 撤銷管道端點（管道擁有者）。

使用下列程序刪除連線帳戶中的管道端點。

刪除管道端點（連線帳戶）

1. 使用共用管道在 中的連線帳戶中開啟 AWS 區域 CLI。
2. 執行下列命令，列出 區域中的管道端點：

```
aws osis --region region list-pipeline-endpoints
```

   記下您要刪除的管道 ID。

3. 執行下列命令來刪除管道端點：

```
aws osis --region region delete-pipeline-endpoint \
  --endpoint-id 'ID'
```

身為共用管道的管道擁有者，請使用下列程序來撤銷管道端點。

撤銷管道端點（管道擁有者）

1. 使用共用管道在 中的連線帳戶中開啟 AWS 區域 CLI。
2. 執行下列命令，列出 區域中的管道端點連線：

```
aws osis --region region list-pipeline-endpoint-connections
```

記下您要刪除的管道 ID。

3. 執行下列命令來刪除管道端點：

```
aws osis --region region revoke-pipeline-endpoint-connections \
  --pipeline-arn pipeline-arn --endpoint-ids ID
```

命令僅支援指定一個端點 ID。

Amazon OpenSearch Ingestion 的 Identity and Access Management

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證（登入）和授權（具有許可），以使用 OpenSearch Ingestion 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- OpenSearch Ingestion 的身分型政策
- OpenSearch 擷取的政策動作
- OpenSearch Ingestion 的政策資源
- Amazon OpenSearch Ingestion 的政策條件索引鍵
- ABAC 搭配 OpenSearch 擷取
- 搭配 OpenSearch Ingestion 使用臨時憑證
- OpenSearch Ingestion 的服務連結角色
- OpenSearch Ingestion 的身分型政策範例

OpenSearch Ingestion 的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

OpenSearch Ingestion 的身分型政策範例

若要檢視 OpenSearch Ingestion 身分型政策的範例，請參閱 the section called “身分型政策範例”。

OpenSearch 擷取的政策動作

支援政策動作：是

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

OpenSearch Ingestion 中的政策動作在動作之前使用以下字首：

```
osis
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
      "osis:action1",
      "osis:action2"
         ]
```

您可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 `List` 文字的所有動作，請包含以下動作：

```
"Action": "osis:List*"
```

若要檢視 OpenSearch Ingestion 身分型政策的範例，請參閱 OpenSearch Serverless 的身分型政策範例。

OpenSearch Ingestion 的政策資源

支援政策資源：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (*) 表示該陳述式適用於所有資源。

```
"Resource": "*"
```

Amazon OpenSearch Ingestion 的政策條件索引鍵

支援服務特定政策條件金鑰：否

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素會根據定義的條件，指定陳述式的執行時機。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 OpenSearch Ingestion 條件索引鍵的清單，請參閱《服務授權參考》中的 Amazon OpenSearch Ingestion 的條件索引鍵。若要了解您可以使用條件金鑰的動作和資源，請參閱 Amazon OpenSearch Ingestion 定義的動作。

ABAC 搭配 OpenSearch 擷取

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，依據稱為標籤的屬性來定義許可。您可以將標籤連接至 IAM 實體 AWS 和資源，然後設計 ABAC 政策，以便在委託人的標籤符合資源上的標籤時允許操作。

如需根據標籤控制存取，請使用 `aws:ResourceTag/`*`key-name`*、`aws:RequestTag/`*`key-name`* 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的使用屬性型存取控制 (ABAC)。

如需標記 OpenSearch Ingestion 資源的詳細資訊，請參閱 the section called “標記管道”。

搭配 OpenSearch Ingestion 使用臨時憑證

支援臨時憑證：是

臨時登入資料提供 AWS 資源的短期存取權，當您使用聯合或切換角色時會自動建立。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的臨時安全憑證與可與 IAM 搭配運作的AWS 服務。

OpenSearch Ingestion 的服務連結角色

支援服務連結角色：是

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

OpenSearch Ingestion 使用稱為 的服務連結角色`AWSServiceRoleForAmazonOpenSearchIngestionService`。名為 的服務連結角色`AWSServiceRoleForOpensearchIngestionSelfManagedVpce`也適用於具有自我管理 VPC 端點的管道。如需建立和管理 OpenSearch Ingestion 服務連結角色的詳細資訊，請參閱 the section called “管道建立角色”。

OpenSearch Ingestion 的身分型政策範例

根據預設，使用者和角色沒有建立或修改 OpenSearch Ingestion 資源的許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需 Amazon OpenSearch Ingestion 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的 Amazon OpenSearch Ingestion 的動作、資源和條件索引鍵。

主題

- 政策最佳實務
- 在主控台中使用 OpenSearch 擷取

- 管理 OpenSearch 擷取管道
- 將資料擷取至 OpenSearch 擷取管道

政策最佳實務

身分型政策相當強大。他們會判斷您帳戶中的某人是否可以建立、存取或刪除 OpenSearch Ingestion 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 OpenSearch Ingestion 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 IAM 安全最佳實務。

在主控台中使用 OpenSearch 擷取

若要在 OpenSearch Service 主控台中存取 OpenSearch Ingestion，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 AWS 帳戶中 OpenSearch Ingestion 資源的詳細資訊。如果您建立比最基本

必要許可更嚴格的身分型政策，則對於具有該政策的實體 (例如 IAM 角色等) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合您嘗試執行之 API 作業的動作就可以了。

下列政策允許使用者在 OpenSearch Service 主控台中存取 OpenSearch Ingestion：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ]
        }
    ]
}
```

或者，您可以使用 the section called “AmazonOpenSearchIngestionReadOnlyAccess” AWS 受管政策，授予 所有 OpenSearch Ingestion 資源的唯讀存取權 AWS 帳戶。

管理 OpenSearch 擷取管道

此政策是「管道管理員」政策的範例，可讓使用者管理 Amazon OpenSearch Ingestion 管道。使用者可以建立、檢視和刪除管道。

JSON

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

```
        {
            "Resource": "arn:aws:osis:us-east-1:111122223333:pipeline/*",
            "Action": [
                "osis:CreatePipeline",
                "osis:DeletePipeline",
                "osis:UpdatePipeline",
                "osis:ValidatePipeline",
                "osis:StartPipeline",
                "osis:StopPipeline"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ],
            "Effect": "Allow"
        }
    ]
}
```

將資料擷取至 OpenSearch 擷取管道

此範例政策允許使用者或其他實體將資料擷取至其帳戶中的 Amazon OpenSearch Ingestion 管道。使用者無法修改管道。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:us-east-1:123456789012:pipeline/*",
            "Action": [
                "osis:Ingest"
            ],
            "Effect": "Allow"
```

```
        }
    ]
}
```

使用 記錄 Amazon OpenSearch Ingestion API 呼叫 AWS CloudTrail

Amazon OpenSearch Ingestion 已與 服務整合 AWS CloudTrail，此服務可提供使用者、角色或 OpenSearch Ingestion 中 AWS 服務所採取之動作的記錄。

CloudTrail 會將 OpenSearch Ingestion 的所有 API 呼叫擷取為事件。擷取的呼叫包括來自 OpenSearch Service 主控台 OpenSearch Ingestion 區段的呼叫，以及對 OpenSearch Ingestion API 操作的程式碼呼叫。

如果您建立線索，則可以將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 OpenSearch Ingestion 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。

您可以使用 CloudTrail 所收集的資訊，判斷對 OpenSearch Ingestion 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

CloudTrail 中的 OpenSearch 擷取資訊

當您建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。當活動在 OpenSearch Ingestion 中發生時，該活動會與事件歷史記錄中的其他 AWS 服務事件一起記錄在 CloudTrail 事件中。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱「使用 CloudTrail 事件歷史記錄檢視事件」。

若要持續記錄 中的事件 AWS 帳戶，包括 OpenSearch Ingestion 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。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。

線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案

CloudTrail 會記錄所有 OpenSearch Ingestion 動作，並記錄在 OpenSearch Ingestion API 參考中。例如，對 `CreateCollection`、`ListCollections` 以及 `DeleteCollection` 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

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：

- 是否使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出請求。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 OpenSearch Ingestion 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。

事件代表來自任何來源的單一請求。其中包含了請求的動作、動作的日期和時間、請求參數等相關資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的是展示 `DeletePipeline` 動作的 CloudTrail 日誌項目。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/test-user",
        "accountId": "123456789012",
        "accessKeyId": "access-key",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-04-21T16:48:33Z",
                "mfaAuthenticated": "false"
```

```
            }
        }
    },
    "eventTime": "2023-04-21T16:49:22Z",
    "eventSource": "osis.amazonaws.com",
    "eventName": "UpdatePipeline",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36",
    "requestParameters": {
        "pipelineName": "my-pipeline",
        "pipelineConfigurationBody": "version: \"2\"\nlog-pipeline:\n  source:\n
 http:\n        path: \"/test/logs\"\n  processor:\n    - grok:\n        match:\n
        log: [ '%{COMMONAPACHELOG}' ]\n    - date:\n        from_time_received: true
\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://search-b5zd22mwxhggheqpj5ftslgyle.us-west-2.es.amazonaws.com\" ]\n
  index: \"apache_logs2\"\n        aws_sts_role_arn: \"arn:aws:iam::709387180454:role/
canary-bootstrap-OsisRole-J1BARLD26QKN\"\n        aws_region: \"us-west-2\"\n
 aws_sigv4: true\n"
    },
    "responseElements": {
        "pipeline": {
            "pipelineName": "my-pipeline",sourceIPAddress
            "pipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/my-pipeline",
            "minUnits": 1,
            "maxUnits": 1,
            "status": "UPDATING",
            "statusReason": {
                "description": "An update was triggered for the pipeline. It is still
 available to ingest data."
            },
            "pipelineConfigurationBody": "version: \"2\"\nlog-pipeline:\n  source:\n
    http:\n        path: \"/test/logs\"\n  processor:\n    - grok:\n        match:
\n          log: [ '%{COMMONAPACHELOG}' ]\n    - date:\n        from_time_received:
 true\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://search-b5zd22mwxhggheqpj5ftslgyle.us-west-2.es.amazonaws.com\" ]\n
  index: \"apache_logs2\"\n        aws_sts_role_arn: \"arn:aws:iam::709387180454:role/
canary-bootstrap-OsisRole-J1BARLD26QKN\"\n        aws_region: \"us-west-2\"\n
 aws_sigv4: true\n",
            "createdAt": "Mar 29, 2023 1:03:44 PM",
            "lastUpdatedAt": "Apr 21, 2023 9:49:21 AM",
            "ingestEndpointUrls": [
                "my-pipeline-tu33ldsgdltgv7x7tjqiudvf7m.us-west-2.osis.amazonaws.com"
```

```
            ]
        }
    },
    "requestID": "12345678-1234-1234-1234-987654321098",
    "eventID": "12345678-1234-1234-1234-987654321098",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "709387180454",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "osis.us-west-2.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

Amazon OpenSearch 擷取和介面端點 API (AWS PrivateLink)

您可以建立介面 VPC 端點，在 VPC 和 OpenSearch Ingestion API 端點之間建立私有連線。界面端點是採用 AWS PrivateLink 技術。

AWS PrivateLink 可讓您在沒有網際網路閘道、NAT 裝置、VPN 連接或 Direct Connect 連線的情況下，私下存取 OpenSearch Ingestion API 操作。VPC 中的資源不需要公有 IP 地址即可與 OpenSearch Ingestion API 端點通訊，即可建立、修改或刪除管道。VPC 與 OpenSearch Ingestion 之間的流量不會離開 Amazon 網路。

> **Note**
>
> 本主題涵蓋用於存取 OpenSearch Ingestion API 的 VPC 端點，可讓您從 VPC 內管理管道（建立、更新、刪除）。這與為管道本身設定 VPC 存取不同，這會控制資料如何從 VPC 內的來源擷取到管道。如需設定管道 VPC 存取的詳細資訊，請參閱 the section called “設定管道的 VPC 存取”。

每個界面端點都是由您子網路中的一或多個彈性網路界面表示。如需彈性網路界面的詳細資訊，請參閱 Amazon EC2 使用者指南中的彈性網路界面。

如需 VPC 端點的詳細資訊，請參閱《Amazon VPC 使用者指南》中的界面 VPC 端點 (AWS PrivateLink)。如需 OpenSearch Ingestion API 操作的詳細資訊，請參閱 OpenSearch Ingestion API 參考。

VPC 端點的考量事項

為 OpenSearch Ingestion API 端點設定介面 VPC 端點之前，請務必檢閱《Amazon VPC 使用者指南》中的介面端點屬性和限制。

所有與管理 OpenSearch Ingestion 資源相關的 OpenSearch Ingestion API 操作都可以從您的 VPC 使用 AWS PrivateLink。

OpenSearch Ingestion API 端點支援 VPC 端點政策。根據預設，允許透過端點完整存取 OpenSearch Ingestion API 操作。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用 VPC 端點控制對服務的存取。

可用性

OpenSearch Ingestion API 目前在所有 OpenSearch Ingestion 區域中支援 VPC 端點。

目前不支援 FIPS 端點。

為 OpenSearch Ingestion API 建立介面 VPC 端點

您可以使用 Amazon VPC 主控台或 AWS Command Line Interface () 為 OpenSearch Ingestion API 建立 VPC 端點AWS CLI。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的建立介面端點。

使用服務名稱 為 OpenSearch Ingestion API 建立 VPC 端點`com.amazonaws.region.osis`。

如果您為端點啟用私有 DNS，則可以使用 AWS 區域的預設 DNS 名稱向 OpenSearch Ingestion 提出 API 請求，例如 `osis.us-east-1.amazonaws.com`。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的透過介面端點存取服務。

為 OpenSearch Ingestion API 建立 VPC 端點政策

您可以將端點政策連接至 VPC 端點，以控制對 OpenSearch Ingestion API 的存取。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的使用 VPC 端點控制對服務的存取。

範例：OpenSearch Ingestion API 動作的 VPC 端點政策

以下是 OpenSearch Ingestion API 的端點政策範例。連接到端點時，此政策會授予所有資源上所有主體對所列 OpenSearch Ingestion API 動作的存取權。

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "osis:CreatePipeline",
            "osis:UpdatePipeline",
            "osis:DeletePipeline"
         ],
         "Resource":"*"
      }
   ]
}
```

範例：拒絕來自指定 AWS 帳戶的所有存取的 VPC 端點政策

下列 VPC 端點政策拒絕 AWS 使用端點對 資源123456789012的所有存取。此政策允許來自其他帳戶的所有動作。

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": { "AWS": [ "123456789012" ] }
     }
   ]
```

```
}
```

標記 Amazon OpenSearch 擷取管道

標籤可讓您將任意資訊指派給 Amazon OpenSearch Ingestion 管道，以便您可以分類和篩選該資訊。標籤是您指派或 AWS 指派給 AWS 資源的中繼資料標籤。每個標籤皆包含鍵與值。對於您指派的標籤，您可以定義索引鍵和值。例如，您可以將鍵定義為 `stage`，將資源的值定義為 `test`。

標籤可協助您執行以下操作：

- 識別和組織您的 AWS 資源。許多 AWS 服務支援標記，因此您可以將相同的標籤指派給來自不同服務的資源，以指出資源相關。例如，您可以將相同的標籤指派給指派給 Amazon OpenSearch Service 網域的 OpenSearch Ingestion 管道。
- 追蹤您的 AWS 成本。您可以在 AWS 帳單與成本管理 儀表板上啟用這些標籤。 AWS 會使用標籤來分類您的成本，並傳送每月成本分配報告給您。如需詳細資訊，請參閱《AWS Billing 使用者指南》中的使用成本分配標籤。
- 使用屬性型存取控制限制對管道的存取。如需詳細資訊，請參閱《IAM 使用者指南》中的根據標籤金鑰控制存取權。

在 OpenSearch Ingestion 中，主要資源是管道。您可以使用 OpenSearch Service 主控台、 AWS CLI、OpenSearch Ingestion APIs 或 AWS SDKs，從管道新增、管理和移除標籤。

主題

- 必要許可
- 處理標籤 (主控台)
- 處理標籤 (AWS CLI)

必要許可

OpenSearch Ingestion 使用下列 AWS Identity and Access Management Access Analyzer (IAM) 許可來標記管道：

- `osis:TagResource`
- `osis:ListTagsForResource`
- `osis:UntagResource`

如需每個許可的詳細資訊，請參閱《服務授權參考》中的 OpenSearch Ingestion 的動作、資源和條件索引鍵。

處理標籤 (主控台)

主控台是標記管道的最簡單方法。

建立標籤

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。
2. 選取您要新增標籤的管道，然後前往標籤索引標籤。
3. 選擇 Manage (管理) 和 Add new tag (新增標籤)。
4. 輸入標籤索引鍵和選用的值。
5. 選擇儲存。

若要刪除標籤，請按照同樣的步驟進行，然後在 Manage tags (管理標籤) 頁面上選擇 Remove (移除)。

如需使用主控台處理標籤的詳細資訊，請參閱 《AWS 管理主控台入門指南》中的標籤編輯器。

處理標籤 (AWS CLI)

若要使用 標記管道 AWS CLI，請傳送`TagResource`請求：

```
aws osis tag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tags Key=service,Value=osis Key=source,Value=otel
```

使用 `UntagResource`命令從管道移除標籤：

```
aws osis untag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tag-keys service
```

使用 `ListTagsForResource`命令檢視管道的現有標籤：

```
aws osis list-tags-for-resource
```

```
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
```

使用 Amazon CloudWatch 記錄和監控 Amazon OpenSearch 擷取

Amazon OpenSearch Ingestion 會將指標和日誌發佈至 Amazon CloudWatch。

主題

- 監控管道日誌
- 監控管道指標

監控管道日誌

您可以啟用 Amazon OpenSearch Ingestion 管道的記錄，公開管道操作和擷取活動期間引發的錯誤和警告訊息。OpenSearch Ingestion 會將所有日誌發佈至 Amazon CloudWatch Logs。CloudWatch Logs 可監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。

OpenSearch Ingestion 的日誌可能表示請求處理失敗、從來源到目的地的身分驗證錯誤，以及其他有助於故障診斷的警告。對於其日誌，OpenSearch Ingestion 會使用 `INFO`、`WARN`、 `ERROR`和 的日誌層級`FATAL`。我們建議為所有管道啟用日誌發佈。

必要許可

若要讓 OpenSearch Ingestion 將日誌傳送至 CloudWatch Logs，您必須以具有特定 IAM 許可的使用者身分登入。

您需要下列 CloudWatch Logs 許可，才能建立和更新日誌交付資源：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
```

```
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:GetLogDelivery",
                "logs:ListLogDeliveries"
            ]
        }
    ]
}
```

啟用日誌發佈

您可以在現有管道上或在建立管道時啟用日誌發佈。如需在管道建立期間啟用日誌發佈的步驟，請參閱the section called “建立管道”。

主控台

在現有管道上啟用日誌發佈

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/osis/home。您將進入管道頁面。
2. 開啟您要啟用日誌的管道，然後選擇動作、編輯日誌發佈選項。
3. 啟用發佈至 CloudWatch Logs。
4. 建立新的日誌群組或選取現有的日誌群組。我們建議您將名稱格式化為路徑，例如 `/aws/vendedlogs/OpenSearchIngestion/pipeline-name/audit-logs`。此格式可讓您更輕鬆地套用 CloudWatch 存取政策，將許可授予特定路徑下的所有日誌群組，例如 `/aws/vendedlogs/OpenSearchIngestion`。

   > **Important**
   >
   > 您必須在日誌群組名稱`vendedlogs`中包含 字首，否則建立會失敗。

5. 選擇儲存。

CLI

若要使用 啟用日誌發佈 AWS CLI，請傳送下列請求：

```
aws osis update-pipeline \
```

```
  --pipeline-name my-pipeline \
  --log-publishing-options  IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="/
aws/vendedlogs/OpenSearchIngestion/pipeline-name"}
```

監控管道指標

您可以使用 Amazon CloudWatch 監控 Amazon OpenSearch Ingestion 管道，該管道會收集原始資料並將其處理為可讀且幾近即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

OpenSearch Ingestion 主控台會根據每個管道的效能索引標籤上來自 CloudWatch 的原始資料顯示一系列圖表。

OpenSearch Ingestion 會從大多數支援的外掛程式報告指標。如果以下某些外掛程式沒有自己的資料表，表示它們不會報告任何外掛程式特定的指標。管道指標會發佈在 AWS/OSIS 命名空間中。

主題

- 常見指標
- 緩衝區指標
- Signature V4 指標
- 繫結封鎖緩衝區指標
- Otel 追蹤來源指標
- Otel 指標來源指標
- Http 指標
- S3 指標
- 彙總指標
- 日期指標
- Lambda 指標
- Grok 指標
- Otel 追蹤原始指標
- Otel 追蹤群組指標
- 服務映射有狀態指標
- OpenSearch 指標
- 系統和計量指標

常見指標

下列指標適用於所有處理器和接收器。

每個指標的字首都是子管道名稱和外掛程式名稱，格式為 *<sub_pipeline_name><plugin><metric_name>*。例如，名為 recordsIn.count 的子管道my-pipeline和日期處理器的指標全名為 my-pipeline.date.recordsIn.count。

指標尾碼	Description
recordsIn.count	將記錄輸入至管道元件。此指標適用於處理器和接收器。 相關統計資料：總和 維度： PipelineName
recordsOut.count	從管道元件傳出記錄。此指標適用於處理器和來源。 相關統計資料：總和 維度： PipelineName
timeElapsed.count	在管道元件執行期間記錄的資料點計數。此指標適用於處理器和接收器。 相關統計資料：總和 維度： PipelineName
timeElapsed.sum	管道元件執行期間經過的總時間。此指標適用於處理器和接收器，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
timeElapsed.max	管道元件執行期間經過的時間上限。此指標適用於處理器和接收器，以毫秒為單位。 相關統計資料：上限 維度： PipelineName

緩衝區指標

下列指標適用於 OpenSearch Ingestion 自動為所有管道設定的預設邊界封鎖緩衝區。

每個指標的字首都是子管道名稱和緩衝區名稱，格式為 <*sub_pipeline_name*><*buffer_name*><*metric_name*>。例如，名為 `recordsWritten.count` 之子管道的 指標全名`my-pipeline`為 `my-pipeline.BlockingBuffer.recordsWritten.count`。

指標尾碼	Description
`recordsWritten.count`	寫入緩衝區的記錄數目。 相關統計資料：總和 維度： `PipelineName`
`recordsRead.count`	從緩衝區讀取的記錄數目。 相關統計資料：總和 維度： `PipelineName`
`recordsInFlight.value`	從緩衝區讀取的未核取記錄數。 相關統計資料：平均 維度： `PipelineName`
`recordsInBuffer.value`	緩衝區中目前記錄的數量。 相關統計資料：平均 維度： `PipelineName`
`recordsProcessed.count`	從緩衝區讀取並由管道處理的記錄數目。 相關統計資料：總和 維度： `PipelineName`
`recordsWriteFailed.count`	管道無法寫入接收器的記錄數目。

指標尾碼	Description
	相關統計資料：總和 維度： PipelineName
writeTimeElapsed.count	寫入緩衝區時記錄的資料點計數。 相關統計資料：總和 維度： PipelineName
writeTimeElapsed.sum	寫入緩衝區所經過的總時間，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
writeTimeElapsed.max	寫入緩衝區時所經過的時間上限，以毫秒為單位。 相關統計資料：上限 維度： PipelineName
writeTimeouts.count	緩衝區的寫入逾時計數。 相關統計資料：總和 維度： PipelineName
readTimeElapsed.count	從緩衝區讀取時記錄的資料點計數。 相關統計資料：總和 維度： PipelineName
readTimeElapsed.sum	從緩衝區讀取所經過的總時間，以毫秒為單位。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
readTimeElapsed.max	從緩衝區讀取所經過的時間上限，以毫秒為單位。 相關統計資料：上限 維度： PipelineName
checkpointTimeElapsed.count	檢查點時記錄的資料點計數。 相關統計資料：總和 維度： PipelineName
checkpointTimeElapsed.sum	檢查點所經過的總時間，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
checkpointTimeElapsed.max	檢查點所經過的時間上限，以毫秒為單位。 相關統計資料：上限 維度： PipelineName

Signature V4 指標

下列指標適用於管道的擷取端點，並與來源外掛程式 (http、 otel_trace和) 建立關聯otel_metrics。所有對擷取端點的請求都必須使用 Signature 第 4 版簽署。這些指標可協助您在連線至管道時識別授權問題，或確認您已成功驗證。

每個指標的字首都是子管道名稱 和 osis_sigv4_auth。例如 *sub_pipeline_name*.osis_sigv4_auth.httpAuthSuccess.count。

指標尾碼	Description
httpAuthSuccess.count	管道的成功 Signature V4 請求數量。 相關統計資料：總和

指標尾碼	Description
	維度： PipelineName
httpAuthFailure.count	管道的失敗 Signature V4 請求數目。 相關統計資料：總和 維度： PipelineName
httpAuthServerError.count	傳回伺服器錯誤的管道的 Signature V4 請求數目。 相關統計資料：總和 維度： PipelineName

繫結封鎖緩衝區指標

下列指標適用於週框封鎖緩衝區。每個指標的字首都是子管道名稱 和 BlockingBuffer。例如 *sub_pipeline_name*.BlockingBuffer.bufferUsage.value。

指標尾碼	Description
bufferUsage.value	buffer_size 根據緩衝區中記錄數量的 使用百分比。buffer_size 代表寫入緩衝區的記錄數量上限，以及尚未檢查的傳輸中記錄。 相關統計資料：平均 維度： PipelineName

Otel 追蹤來源指標

下列指標適用於 OTel 追蹤來源。每個指標的字首都是子管道名稱 和 otel_trace_source。例如 *sub_pipeline_name*.otel_trace_source.requestTimeouts.count。

指標尾碼	Description
requestTimeouts.count	逾時的請求數量。

指標尾碼	Description
	相關統計資料：總和 維度： PipelineName
requestsReceived.count	外掛程式收到的請求數量。 相關統計資料：總和 維度： PipelineName
successRequests.count	外掛程式已成功處理的請求數量。 相關統計資料：總和 維度： PipelineName
badRequests.count	外掛程式處理格式無效的請求數量。 相關統計資料：總和 維度： PipelineName
requestsTooLarge.count	內容中跨度數目大於緩衝區容量的請求數目。 相關統計資料：總和 維度： PipelineName
internalServerError.count	外掛程式使用自訂例外狀況類型處理的請求數。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.count	處理外掛程式請求時記錄的資料點計數。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
requestProcessDuration.sum	外掛程式處理的請求總延遲，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.max	外掛程式所處理請求的最大延遲，以毫秒為單位。 相關統計資料：上限 維度： PipelineName
payloadSize.count	傳入請求承載大小的分佈計數，以位元組為單位。 相關統計資料：總和 維度： PipelineName
payloadSize.sum	傳入請求承載大小的總分佈，以位元組為單位。 相關統計資料：總和 維度： PipelineName
payloadSize.max	傳入請求承載大小的最大分佈，以位元組為單位。 相關統計資料：上限 維度： PipelineName

Otel 指標來源指標

下列指標適用於 OTel 指標來源。每個指標的字首都是子管道名稱 和 otel_metrics_source。例如 *sub_pipeline_name*.otel_metrics_source.requestTimeouts.count。

指標尾碼	Description
requestTimeouts.count	逾時的外掛程式請求總數。

指標尾碼	Description
	相關統計資料：總和 維度： PipelineName
requestsReceived.count	外掛程式收到的請求總數。 相關統計資料：總和 維度： PipelineName
successRequests.count	外掛程式成功處理的請求數 (200 個回應狀態碼）。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.count	外掛程式處理的請求延遲計數，以秒為單位。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.sum	外掛程式處理的請求總延遲，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.max	外掛程式所處理請求的最大延遲，以毫秒為單位。 相關統計資料：上限 維度： PipelineName
payloadSize.count	傳入請求承載大小的分佈計數，以位元組為單位。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
payloadSize.sum	傳入請求承載大小的總分佈，以位元組為單位。 相關統計資料：總和 維度： PipelineName
payloadSize.max	傳入請求承載大小的最大分佈，以位元組為單位。 相關統計資料：上限 維度： PipelineName

Http 指標

下列指標適用於 HTTP 來源。每個指標的字首都是子管道名稱 和 http。例如 *sub_pipeline_name*.http.requestsReceived.count。

指標尾碼	Description
requestsReceived.count	/log/ingest 端點收到的請求數量。 相關統計資料：總和 維度： PipelineName
requestsRejected.count	外掛程式拒絕的請求數 (429 個回應狀態碼）。 相關統計資料：總和 維度： PipelineName
successRequests.count	外掛程式成功處理的請求數 (200 個回應狀態碼）。 相關統計資料：總和 維度： PipelineName
badRequests.count	外掛程式處理的內容類型或格式無效 (400 回應狀態碼） 的請求數量。

指標尾碼	Description
	相關統計資料：總和 維度： PipelineName
requestTimeouts.count	HTTP 來源伺服器中逾時的請求數目 (415 回應狀態碼）。 相關統計資料：總和 維度： PipelineName
requestsTooLarge.count	內容中事件大小大於緩衝區容量 (413 回應狀態碼） 的請求數量。 相關統計資料：總和 維度： PipelineName
internalServerError.count	外掛程式使用自訂例外狀況類型 (500 個回應狀態碼） 處理的請求數。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.count	外掛程式處理的請求延遲計數，以秒為單位。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.sum	外掛程式處理的請求總延遲，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
requestProcessDuration.max	外掛程式所處理請求的最大延遲，以毫秒為單位。 相關統計資料：上限 維度： PipelineName

指標尾碼	Description
payloadSize.count	傳入請求承載大小的分佈計數，以位元組為單位。 相關統計資料：總和 維度： PipelineName
payloadSize.sum	傳入請求承載大小的總分佈，以位元組為單位。 相關統計資料：總和 維度： PipelineName
payloadSize.max	傳入請求承載大小的最大分佈，以位元組為單位。 相關統計資料：上限 維度： PipelineName

S3 指標

下列指標適用於 S3 來源。每個指標的字首都是子管道名稱 和 s3。例如 *sub_pipeline_name*.s3.s3ObjectsFailed.count。

指標尾碼	Description
s3ObjectsFailed.count	外掛程式無法讀取的 S3 物件總數。 相關統計資料：總和 維度： PipelineName
s3ObjectsNotFound.count	由於 S3 的Not Found錯誤，外掛程式無法讀取的 S3 物件數量。這些指標也會計入s3ObjectsFailed 指標。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
s3ObjectsAccessDenied.count	由於 S3 的 Access Denied或 Forbidden 錯誤，外掛程式無法讀取的 S3 物件數量。這些指標也會計入s3ObjectsFailed 指標。 相關統計資料：總和 維度： PipelineName
s3ObjectReadTimeElapsed.count	外掛程式為 S3 物件執行 GET 請求、剖析它，以及將事件寫入緩衝區所需的時間。 相關統計資料：總和 維度： PipelineName
s3ObjectReadTimeElapsed.sum	外掛程式執行 S3 物件的 GET 請求、剖析它以及將事件寫入緩衝區所需的總時間，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
s3ObjectReadTimeElapsed.max	外掛程式執行 S3 物件的 GET 請求、剖析它，以及將事件寫入緩衝區所需的時間上限，以毫秒為單位。 相關統計資料：上限 維度： PipelineName
s3ObjectSizeBytes.count	S3 物件大小的分佈計數，以位元組為單位。 相關統計資料：總和 維度： PipelineName
s3ObjectSizeBytes.sum	S3 物件大小的總分佈，以位元組為單位。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
s3ObjectSizeBytes.max	S3 物件大小的最大分佈，以位元組為單位。 相關統計資料：上限 維度： PipelineName
s3ObjectProcessedBytes.count	外掛程式處理的 S3 物件分佈計數，以位元組為單位。 相關統計資料：總和 維度： PipelineName
s3ObjectProcessedBytes.sum	外掛程式處理的 S3 物件總分佈，以位元組為單位。 相關統計資料：總和 維度： PipelineName
s3ObjectProcessedBytes.max	外掛程式處理的 S3 物件最大分佈，以位元組為單位。 相關統計資料：上限 維度： PipelineName
s3ObjectsEvents.count	外掛程式收到的 S3 事件分佈計數。 相關統計資料：總和 維度： PipelineName
s3ObjectsEvents.sum	外掛程式收到的 S3 事件總分佈。 相關統計資料：總和 維度： PipelineName
s3ObjectsEvents.max	外掛程式收到的 S3 事件最大分佈。 相關統計資料：上限 維度： PipelineName

指標尾碼	Description
sqsMessageDelay.count	當 S3 在物件完全剖析時記錄物件建立的事件時間時，所記錄的資料點計數。 相關統計資料：總和 維度： PipelineName
sqsMessageDelay.sum	S3 記錄物件建立的事件時間到物件完全剖析之間的總時間量，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
sqsMessageDelay.max	S3 記錄物件建立的事件時間到物件完全剖析之間的時間上限，以毫秒為單位。 相關統計資料：上限 維度： PipelineName
s3ObjectsSucceeded.count	外掛程式成功讀取的 S3 物件數量。 相關統計資料：總和 維度： PipelineName
sqsMessagesReceived.count	外掛程式從佇列接收的 Amazon SQS 訊息數量。 相關統計資料：總和 維度： PipelineName
sqsMessagesDeleted.count	外掛程式從佇列刪除的 Amazon SQS 訊息數量。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
sqsMessagesFailed.count	外掛程式無法剖析的 Amazon SQS 訊息數量。 相關統計資料：總和 維度： PipelineName

彙總指標

下列指標適用於彙總處理器。每個指標的字首都是子管道名稱 和 aggregate。例如 *sub_pipeline_name*.aggregate.actionHandleEventsOut.count。

指標尾碼	Description
actionHandleEvents Out.count	從handleEvent 呼叫傳回至已設定動作的事件數量。 相關統計資料：總和 維度： PipelineName
actionHandleEvents Dropped.count	從handleEvent 呼叫傳回至已設定動作的事件數量。 相關統計資料：總和 維度： PipelineName
actionHandleEvents ProcessingErrors.count	針對導致錯誤的已設定動作handleEvent 對 進行的呼叫數量。 相關統計資料：總和 維度： PipelineName
actionConcludeGrou pEventsOut.count	從concludeGroup 呼叫傳回至已設定動作的事件數量。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
actionConcludeGroupEventsDropped.count	尚未從condludeGroup 呼叫傳回至已設定動作的事件數量。 相關統計資料：總和 維度： PipelineName
actionConcludeGroupEventsProcessingErrors.count	針對導致錯誤的已設定動作concludeGroup 對 進行的呼叫數量。 相關統計資料：總和 維度： PipelineName
currentAggregateGroups.value	目前群組的數量。此計量會在群組結束時減少，並在事件啟動建立新群組時增加。 相關統計資料：平均 維度： PipelineName

日期指標

下列指標適用於日期處理器。每個指標的字首都是子管道名稱 和 date。例如 *sub_pipeline_name*.date.dateProcessingMatchSuccess.count。

指標尾碼	Description
dateProcessingMatchSuccess.count	至少符合match組態選項中指定其中一個模式的記錄數目。 相關統計資料：總和 維度： PipelineName
dateProcessingMatchFailure.count	不符合match組態選項中指定之任何模式的記錄數目。 相關統計資料：總和

指標尾碼	Description
	維度： PipelineName

Lambda 指標

下列指標適用於AWS Lambda處理器。每個指標的字首都是子管道名稱 和 lambda。例如 *sub_pipeline_name*.lambda.recordsSuccessfullySentToLambda.count。

指標尾碼	Description
recordsSuccessfullySentToLambda.count	Lambda 函數成功處理的記錄數量。 相關統計資料：總和 維度： PipelineName
recordsFailedToSendToLambda.count	無法傳送至 Lambda 函數的記錄數目。 相關統計資料：總和 維度： PipelineName
lambdaFunctionLatency.avg lambdaFunctionLatency.max	Lambda 函數調用的延遲。 相關統計資料：平均和最大值 維度： PipelineName
numberOfRequestsSucceeded.count	成功的 Lambda 調用請求總數。 相關統計資料：總和 維度： PipelineName
numberOfRequestsFailed.count	失敗的 Lambda 調用請求總數。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
requestPayloadSize.avg	傳送至 Lambda 的請求承載大小。 相關統計資料：平均 維度： PipelineName
responsePayloadSize.avg	從 Lambda 收到的回應承載大小。 相關統計資料：平均 維度： PipelineName

Grok 指標

下列指標適用於 Grok 處理器。每個指標的字首都是子管道名稱 和 grok。例如 *sub_pipeline_name*.grok.grokProcessingMatch.count。

指標尾碼	Description
grokProcessingMatch.count	從match組態選項中找到至少一個模式相符的記錄數目。 相關統計資料：總和 維度： PipelineName
grokProcessingMismatch.count	不符合match組態選項中指定之任何模式的記錄數目。 相關統計資料：總和 維度： PipelineName
grokProcessingErrors.count	記錄處理錯誤的數量。 相關統計資料：總和 維度： PipelineName
grokProcessingTimeouts.count	比對時逾時的記錄數目。

指標尾碼	Description
	相關統計資料：總和 維度： PipelineName
grokProcessingTime.count	當個別記錄與match組態選項中的模式相符時，記錄的資料點計數。 相關統計資料：總和 維度： PipelineName
grokProcessingTime.sum	每個個別記錄與match組態選項中的模式相符所需的總時間，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
grokProcessingTime.max	每個個別記錄與match組態選項中的模式相符所需的時間上限，以毫秒為單位。 相關統計資料：上限 維度： PipelineName

Otel 追蹤原始指標

下列指標適用於 OTel 追蹤原始處理器。每個指標的字首都是子管道名稱 和 otel_trace_raw。例如 *sub_pipeline_name*.otel_trace_raw.traceGroupCacheCount.value。

指標尾碼	Description
traceGroupCacheCount.value	追蹤群組快取中的追蹤群組數量。 相關統計資料：總和 維度： PipelineName
spanSetCount.value	跨度集集合中的跨度集數目。

指標尾碼	Description
	相關統計資料：總和 維度： PipelineName

Otel 追蹤群組指標

下列指標適用於 OTel 追蹤群組處理器。每個指標的字首都是子管道名稱 和 otel_trace_group。例如 *sub_pipeline_name*.otel_trace_group.recordsInMissingTraceGroup.count。

指標尾碼	Description
recordsInMissingTraceGroup.count	缺少追蹤群組欄位的輸入記錄數目。 相關統計資料：總和 維度： PipelineName
recordsOutFixedTraceGroup.count	追蹤群組欄位已成功填入的輸出記錄數目。 相關統計資料：總和 維度： PipelineName
recordsOutMissingTraceGroup.count	缺少追蹤群組欄位的輸出記錄數目。 相關統計資料：總和 維度： PipelineName

服務映射有狀態指標

下列指標適用於 Service-map 狀態處理器。每個指標的字首都是子管道名稱 和 service-map-stateful。例如 *sub_pipeline_name*.service-map-stateful.spansDbSize.count。

指標尾碼	Description
spansDbSize.value	MapDB 中跨目前和先前時段持續時間的記憶體內位元組大小。 相關統計資料：平均 維度： PipelineName
traceGroupDbSize.value	在目前和先前時段持續時間內，MapDB 中追蹤群組的記憶體內位元組大小。 相關統計資料：平均 維度： PipelineName
spansDbCount.value	MapDB 中跨越目前和先前時段持續時間的計數。 相關統計資料：總和 維度： PipelineName
traceGroupDbCount.value	MapDB 中目前和先前時段持續時間的追蹤群組計數。 相關統計資料：總和 維度： PipelineName
relationshipCount.value	在目前和先前時段持續時間中存放的關係計數。 相關統計資料：總和 維度： PipelineName

OpenSearch 指標

下列指標適用於 OpenSearch 接收器。每個指標的字首都是子管道名稱 和 opensearch。例如 *sub_pipeline_name*.opensearch.bulkRequestErrors.count。

指標尾碼	Description
bulkRequestErrors.count	傳送大量請求時遇到的錯誤總數。 相關統計資料：總和 維度： PipelineName
documentsSuccess.count	依大量請求成功傳送至 OpenSearch Service 的文件數量，包括重試次數。 相關統計資料：總和 維度： PipelineName
documentsSuccessFirstAttempt.count	第一次嘗試時，透過大量請求成功傳送至 OpenSearch Service 的文件數量。 相關統計資料：總和 維度： PipelineName
documentErrors.count	大量請求無法傳送的文件數量。 相關統計資料：總和 維度： PipelineName
bulkRequestFailed.count	失敗的大量請求數量。 相關統計資料：總和 維度： PipelineName
bulkRequestNumberOfRetries.count	失敗大量請求的重試次數。 相關統計資料：總和 維度： PipelineName
bulkBadRequestErrors.count	傳送大量請求時遇到的Bad Request錯誤數目。

指標尾碼	Description
	相關統計資料：總和 維度： PipelineName
bulkRequestNotAllowedErrors.count	傳送大量請求時遇到的Request Not Allowed錯誤數目。 相關統計資料：總和 維度： PipelineName
bulkRequestInvalidInputErrors.count	傳送大量請求時遇到的Invalid Input錯誤數目。 相關統計資料：總和 維度： PipelineName
bulkRequestNotFoundErrors.count	傳送大量請求時遇到的Request Not Found錯誤數目。 相關統計資料：總和 維度： PipelineName
bulkRequestTimeoutErrors.count	傳送大量請求時遇到的Request Timeout錯誤數目。 相關統計資料：總和 維度： PipelineName
bulkRequestServerErrors.count	傳送大量請求時遇到的Server Error錯誤數目。 相關統計資料：總和 維度： PipelineName
bulkRequestSizeBytes.count	大量請求承載大小的分佈計數，以位元組為單位。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
bulkRequestSizeBytes.sum	大量請求承載大小的總分佈，以位元組為單位。 相關統計資料：總和 維度： PipelineName
bulkRequestSizeBytes.max	大量請求承載大小的最大分佈，以位元組為單位。 相關統計資料：上限 維度： PipelineName
bulkRequestLatency.count	將請求傳送至外掛程式時記錄的資料點計數，包括重試。 相關統計資料：總和 維度： PipelineName
bulkRequestLatency.sum	傳送至外掛程式的請求總延遲，包括重試，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
bulkRequestLatency.max	傳送至外掛程式的請求最大延遲，包括重試，以毫秒為單位。 相關統計資料：上限 維度： PipelineName
s3.dlqS3RecordsSuccess.count	成功傳送至 S3 無效字母佇列的記錄數目。 相關統計資料：總和 維度： PipelineName

指標尾碼	Description
s3.dlqS3RecordsFailed.count	無法傳送至 S3 無效字母佇列的復原次數。 相關統計資料：總和 維度： PipelineName
s3.dlqS3RequestSuccess.count	S3 無效字母佇列的成功請求數量。 相關統計資料：總和 維度： PipelineName
s3.dlqS3RequestFailed.count	S3 無效字母佇列的失敗請求數。 相關統計資料：總和 維度： PipelineName
s3.dlqS3RequestLatency.count	將請求傳送至 S3 無效字母佇列時記錄的資料點計數，包括重試。 相關統計資料：總和 維度： PipelineName
s3.dlqS3RequestLatency.sum	傳送至 S3 無效字母佇列的請求總延遲，包括重試，以毫秒為單位。 相關統計資料：總和 維度： PipelineName
s3.dlqS3RequestLatency.max	傳送至 S3 無效字母佇列的請求最大延遲，包括重試，以毫秒為單位。 相關統計資料：上限 維度： PipelineName

指標尾碼	Description
s3.dlqS3RequestSizeBytes.count	S3 無效字母佇列之請求承載大小的分佈計數，以位元組為單位。 相關統計資料：總和 維度： PipelineName
s3.dlqS3RequestSizeBytes.sum	S3 無效字母佇列的請求承載大小總分佈，以位元組為單位。 相關統計資料：總和 維度： PipelineName
s3.dlqS3RequestSizeBytes.max	S3 無效字母佇列的請求承載大小上限，以位元組為單位。 相關統計資料：上限 維度： PipelineName

系統和計量指標

下列指標適用於整個 OpenSearch Ingestion 系統。這些指標不會加上任何字首。

指標	Description
system.cpu.usage.value	所有資料節點的可用 CPU 用量百分比。 相關統計資料：平均 維度：PipelineName 、area、 id
system.cpu.count.value	所有資料節點的 CPU 用量總量。 相關統計資料：平均 維度：PipelineName 、area、 id
jvm.memory.max.value	可用於記憶體管理的記憶體數量上限，以位元組為單位。

指標	Description
	相關統計資料：平均 維度：PipelineName 、area、 id
jvm.memory.used.value	使用的記憶體總量，以位元組為單位。 相關統計資料：平均 維度：PipelineName 、area、id簽署
jvm.memory.committed.value	Java 虛擬機器 (JVM) 承諾使用的記憶體量，以位元組為單位。 相關統計資料：平均 維度：PipelineName 、area、 id
computeUnits	管道正在使用的擷取 OpenSearch 運算單位 （ 擷取 OCUs) 數量。 相關統計資料：Max、Sum、Average 維度： PipelineName

Amazon OpenSearch Ingestion 的最佳實務

本主題提供建立和管理 Amazon OpenSearch Ingestion 管道的最佳實務，並包含適用於許多使用案例的一般準則。每個工作負載都是獨一無二的，具有獨特的特性，因此沒有任何一個通用建議適合每個使用案例。

主題

- 一般最佳實務
- 建議 CloudWatch 警示

一般最佳實務

下列一般最佳實務適用於建立和管理管道。

- 為了確保高可用性，請使用兩個或三個子網路設定 VPC 管道。如果您只將管道部署在一個子網路中，且可用區域故障，您將無法擷取資料。
- 在每個管道中，我們建議將子管道的數量限制為 5 個或更少。
- 如果您使用的是 S3 來源外掛程式，請使用平均大小的 S3 檔案來獲得最佳效能。
- 如果您使用的是 S3 來源外掛程式，請在 S3 儲存貯體中每 0.25 GB 的檔案大小新增 S30 秒的額外可見性逾時，以獲得最佳效能。
- 在管道組態中包含無效字母佇列 (DLQ)，以便您可以卸載失敗的事件，並使其可供分析。如果您的接收器因為不正確的映射或其他問題而拒絕資料，您可以將資料路由到 DLQ，以便對問題進行故障診斷和修正。
- 如果您在管道中使用彙總處理器，我們建議您使用 "`local_mode: true`" 旗標，以取得管道的最佳效能。

建議 CloudWatch 警示

當 CloudWatch 指標在經過一些時間超過指定的值時，CloudWatch 警示會執行動作。例如，如果您 AWS 的叢集運作狀態超過一分鐘`red`，建議您傳送電子郵件給您。本節包含 Amazon OpenSearch Ingestion 的一些建議警示，以及如何回應這些警示。

如需有關設定警示的詳細資訊，請參閱 《Amazon CloudWatch 使用者指南》中的建立 Amazon CloudWatch 警示。

警示	問題
`computeUnits` 最大值為 = `maxUnits`已設定 15 分鐘、連續 3 次	管道已達到最大容量，可能需要`maxUnits`更新。增加管道的最大容量
`opensearch.documentErrors.count` sum = 1 分鐘*`{sub_pipeline_name}`* `.opensearch.record`	管道無法寫入 OpenSearch 接收器。檢查管道許可，並確認網域或集合運作狀態良好。如果已設定失敗事件，您也可以檢查無效字母佇列 (DLQ)。

警示	問題
sIn.count 的總和，連續 1 次	
bulkRequestLatency.max 最大值為 >= x 持續 1 分鐘，連續 1 次	管道正在經歷將資料傳送至 OpenSearch 接收器的高延遲。這可能是由於接收器過小或碎片策略不佳，導致接收器落後。持續高延遲可能會影響管道效能，並可能導致用戶端背壓。
httpAuthFailure.count 總和 >= 1，持續 1 分鐘，連續 1 次	未驗證擷取請求。確認所有用戶端都已正確啟用 Signature 第 4 版身分驗證。
system.cpu.usage.value 平均 >= 80%，持續 15 分鐘，連續 3 次	持續高 CPU 用量可能有問題。考慮增加管道的最大容量。
bufferUsage.value 平均 >= 80%，持續 15 分鐘，連續 3 次	持續的高緩衝區用量可能有問題。考慮增加管道的最大容量。

您可能會考慮的其他警示

請考慮根據您經常使用的 Amazon OpenSearch Ingestion 功能設定下列警示。

警示	問題
dynamodb.exportJobFailure.count 總和 1	嘗試觸發匯出至 Amazon S3 失敗。
opensearch.EndtoEn	EndtoEndLatency 高於從 DynamoDB 串流讀取所需的 。這可能是由於 OpenSearch 叢集規模過小，或管道 OCU 容量上限對 DynamoDB 資

警示	問題
dLatency.avg 平均 > X 持續 15 分鐘，連續 4 次	料表上的 WCU 輸送量而言太低所致。 匯出後 EndtoEndLatency 會較高，但 應該會隨著時間減少，因為它會趕上最新的 DynamoDB 串流。
dynamodb.changeEventsProcessed.count sum == 0 表示 X 分鐘	不會從 DynamoDB 串流收集任何記錄。這可能是由於 資料表上沒有活動，或存取 DynamoDB 串流時發生問題所致。
opensearch.s3.dlqS3RecordsSuccess.count sum >= sum opensearch.documentSuccess.count 持續 1 分鐘，連續 1 次	與 OpenSearch 接收器相比，傳送到 DLQ 的記錄數量較多。檢閱 OpenSearch 接收器外掛程式指標，以調查和判斷根本原因。
grok.grokProcessingTimeouts.count sum = recordsIn.count 總和 1 分鐘，連續 5 次	當 Grok 處理器嘗試模式比對時，所有資料都會逾時。這可能會影響效能並減慢您的管道速度。考慮調整您的模式以減少逾時。
grok.grokProcessingErrors.count sum >= 1 持續 1 分鐘，連續 1 次	Grok 處理器無法比對管道中資料的模式，導致錯誤。檢閱您的資料和 Grok 外掛程式組態，以確保模式符合預期。

警示	問題
grok.grokProcessingMismatch.count sum = recordsIn.count 總和 1 分鐘，連續 5 次	Grok 處理器無法將模式與管道中的資料相符。檢閱您的資料和 Grok 外掛程式組態，以確保模式符合預期。
date.dateProcessingMatchFailure.count sum = recordsIn.count 總和 1 minut，連續 5 次	日期處理器無法將任何模式與管道中的資料相符。檢閱您的資料和日期外掛程式組態，以確保預期模式。
s3.s3ObjectsFailed.count 總和 >= 1 持續 1 分鐘，連續 1 次	發生此問題是因為 S3 物件不存在，或管道的權限不足。擷取 s3ObjectsNotFound.count 和 s3ObjectsAccessDenied.count 指標以判斷根本原因。確認 S3 物件存在和/或更新許可。
s3.sqsMessagesFailed.count 總和 >= 1，持續 1 分鐘，連續 1 次	S3 外掛程式無法處理 Amazon SQS 訊息。如果您的 SQS 佇列已啟用 DLQ，請檢閱失敗的訊息。佇列可能會收到管道嘗試處理的無效資料。
http.badRequests.count 總和 >= 1 持續 1 分鐘，連續 1 次	用戶端傳送錯誤的請求。確認所有用戶端傳送的是適當的承載。
http.requestsTooLarge.count 總和 >= 1，持續 1 分鐘，連續 1 次	來自 HTTP 來源外掛程式的請求包含過多的資料，超過緩衝容量。調整用戶端的批次大小。

警示	問題
http.internalServerError.count 總和 >= 0，持續 1 分鐘，連續 1 次	HTTP 來源外掛程式無法接收事件。
http.requestTimeouts.count 總和 >= 0，持續 1 分鐘，連續 1 次	來源逾時可能是管道佈建不足的結果。考慮增加管道maxUnits來處理額外的工作負載。
otel_trace.badRequests.count 總和 >= 1，持續 1 分鐘，連續 1 次	用戶端傳送錯誤的請求。確認所有用戶端傳送的是適當的承載。
otel_trace.requestsTooLarge.count 總和 >= 1，持續 1 分鐘，連續 1 次	來自 Otel Trace 來源外掛程式的請求包含過多的資料，這超過緩衝區容量。調整用戶端的批次大小。
otel_trace.internalServerError.count 總和 >= 0，持續 1 分鐘，連續 1 次	Otel Trace 來源外掛程式無法接收事件。

警示	問題
otel_trace.requestTimeouts.count 總和 >= 0，持續 1 分鐘，連續 1 次	來源逾時可能是管道佈建不足的結果。考慮增加管道maxUnits來處理額外的工作負載。
otel_metrics.requestTimeouts.count 總和 >= 0，持續 1 分鐘，連續 1 次	來源逾時可能是管道佈建不足的結果。考慮增加管道maxUnits來處理額外的工作負載。

Amazon OpenSearch Serverless

Amazon OpenSearch Serverless 是 Amazon OpenSearch Service 的隨需自動擴展組態。與需要手動容量管理的佈建 OpenSearch 網域不同，OpenSearch Serverless 集合會根據應用程式的需求自動擴展運算資源。

OpenSearch Serverless 為不常見、間歇性或無法預測的工作負載提供經濟實惠的解決方案。它會根據應用程式的用量，自動擴展運算容量來最佳化成本。無伺服器集合使用與佈建 OpenSearch Service 網域相同的高容量、分散式和高可用性儲存磁碟區。

OpenSearch Serverless 集合一律會加密。您可以選擇加密金鑰，但無法停用加密。如需詳細資訊，請參閱the section called “加密”

優勢

OpenSearch Serverless 具有下列優勢：

- 比佈建簡單：OpenSearch Serverless 可消除管理 OpenSearch 叢集和容量的大部分複雜性。它會自動調整叢集的大小並進行調校，負責碎片和索引生命週期管理。它也會管理服務軟體更新和 OpenSearch 版本升級。所有更新和升級都屬於非中斷式。
- 具有成本效益：使用 OpenSearch Serverless 時，您僅需為實際使用的資源付費。如此就不需要為尖峰工作負載預先佈建和過度佈建。
- 高可用性：OpenSearch Serverless 透過備援功能支援生產工作負載，以防止可用區域中斷和基礎架構故障。
- 可擴展性：OpenSearch Serverless 會自動調整資源規模，以維持一致的快速資料擷取速率和查詢回應時間。

什麼是 Amazon OpenSearch Serverless？

Amazon OpenSearch Serverless 是 Amazon OpenSearch Service 的隨需無伺服器選項，可消除佈建、設定和調校 OpenSearch 叢集的操作複雜性。對於偏好不自我管理叢集或缺乏專用資源和專業知識來操作大規模部署的組織來說，這是理想的選擇。使用 OpenSearch Serverless，您可以搜尋和分析大量資料，而無需管理基礎基礎設施。

OpenSearch Serverless 集合是一組 OpenSearch 索引，可共同運作以支援特定工作負載或使用案例。相較於需要手動佈建的自我管理 OpenSearch 叢集，集合可簡化操作。

集合使用與佈建 OpenSearch Service 網域相同的高容量、分散式和高可用性儲存體，但透過消除手動組態和調校來進一步降低複雜性。集合內的資料會在傳輸中加密。OpenSearch Serverless 也支援 OpenSearch Dashboards，提供用於資料分析的介面。

目前，無伺服器集合執行 OpenSearch 2.17.x 版。隨著新版本發佈，OpenSearch Serverless 會自動升級集合，以整合新功能、錯誤修正和效能改善。

OpenSearch Serverless 支援與 OpenSearch 開放原始碼套件相同的擷取和查詢 API 操作，因此您可以繼續使用現有的用戶端和應用程式。您的用戶端必須相容於 OpenSearch 2.x，以便使用 OpenSearch Serverless。如需詳細資訊，請參閱the section called “將資料擷取至集合”。

主題

- OpenSearch Serverless 使用案例
- 運作方式
- 選擇集合類型
- 定價
- 支援的 AWS 區域
- 限制
- 比較 OpenSearch Service 和 OpenSearch Serverless

OpenSearch Serverless 使用案例

OpenSearch Serverless 支援兩種主要使用案例：

- 日誌分析：日誌分析區段專注於分析大量半結構化、機器產生的時間序列資料，以提供操作和使用者行為洞察。
- 全文檢索搜尋：全文檢索搜尋區段強化內部網路中的應用程式 (內容管理系統、法律文件) 以及面向網際網路的應用程式，例如電子商務網站內容搜尋。

建立集合時，您可以選擇上述其中一種使用案例。如需詳細資訊，請參閱the section called “選擇集合類型”。

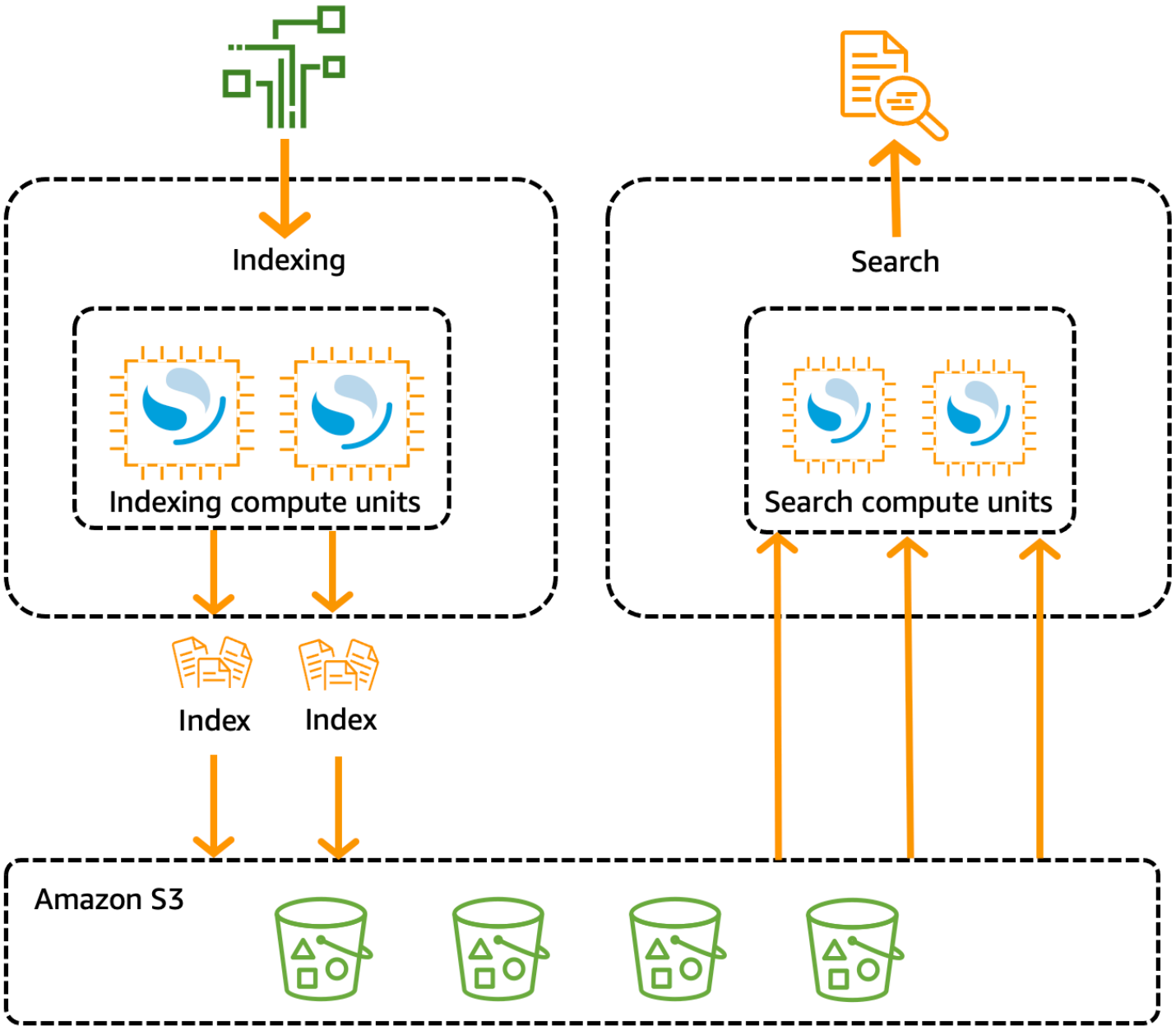
運作方式

傳統 OpenSearch 叢集有一組執行個體，可執行索引編製和搜尋操作，而索引儲存與運算容量緊密結合。相反地，OpenSearch Serverless 使用雲端原生架構，將索引（擷取）元件與搜尋（查詢）元件分開，並將 Amazon S3 作為索引的主要資料儲存體。

此分開的架構可讓您獨立擴展搜尋和索引編製功能，並且獨立於 S3 中的索引資料。該架構也會隔離擷取和查詢操作，以便同時執行這兩項操作，而不會爭用資源。

將資料寫入集合時，OpenSearch Serverless 會將資料分發到索引編製運算單元。索引編製運算單元會擷取傳入資料，並將索引移至 S3。對集合資料執行搜尋時，OpenSearch Serverless 會將請求路由至容納所查詢資料的搜尋運算單元。搜尋運算單元會直接從 S3 下載索引資料 (如果尚未在本機快取資料)、執行搜尋操作並執行彙總。

下圖說明了此分開的架構：



用於資料擷取、搜尋和查詢的 OpenSearch Serverless 運算容量是以 OpenSearch 運算單元 (OCU) 來測量。每個 OCU 都是 6 GiB 記憶體和對應虛擬 CPU (vCPU) 的組合，並會建立 Amazon S3 的資料管道。每個 OCU 都包含足夠的暫時性熱儲存，可容納 120 GiB 的索引資料。

建立第一個集合時，OpenSearch Serverless 會執行個體化兩個 OCU，一個用於索引編製，另一個用於搜尋。為確保高可用性，它也會啟動另一個可用區域中的待命節點集。為了開發和測試目的，您可以停用集合的啟用備援設定，這會消除兩個待命複本，並且只會執行個體化兩個 OCUs。根據預設，會啟用備援作用中複本，這表示帳戶的第一個集合總共會執行個體化四個 OCUs。

即使在任何集合端點上沒有任何活動，這些 OCU 也會存在。所有後續的集合都會共用這些 OCU。當您在同一個帳戶中建立其他集合時，OpenSearch Serverless 只會根據您指定的容量限制，視需要新增額外的搜尋和擷取 OCU 來支援集合。容量會隨著運算用量的減少而縮減。

如需有關這些 OCU 如何計費的資訊，請參閱 the section called “定價”。

選擇集合類型

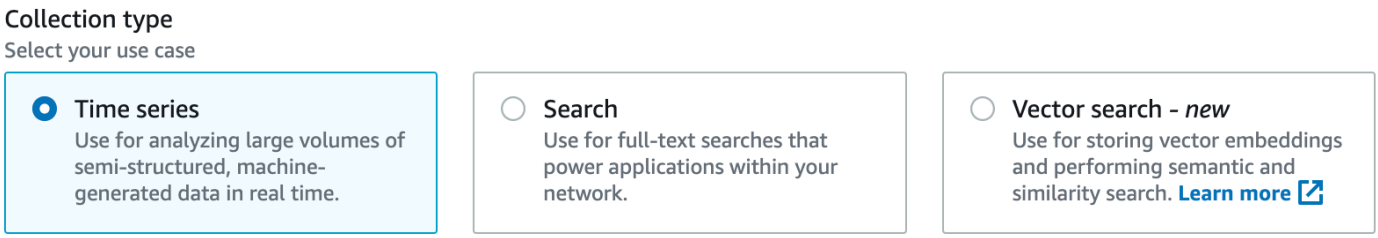
OpenSearch Serverless 支援三種主要集合類型：

時間序列 – 日誌分析區段，可即時分析大量半結構化、機器產生的資料，提供操作、安全性、使用者行為和業務效能的洞見。

搜尋 – 啟用內部網路內應用程式的全文搜尋，例如內容管理系統和法律文件儲存庫，以及電子商務網站搜尋和內容探索等面向網際網路的應用程式。

向量搜尋 – 向量內嵌的語意搜尋可簡化向量資料管理，並啟用機器學習 (ML) 擴增搜尋體驗。它支援生成式 AI 應用程式，例如聊天機器人、個人助理和詐騙偵測。

您可以在第一次建立集合時選擇集合類型：



您選擇的集合類型取決於您計劃擷取至集合的資料類型，以及計劃的查詢方式。您無法在建立後變更集合類型。

集合類型有以下顯著差異：

- 對於搜尋和向量搜尋集合，所有資料都會存放在熱儲存體中，以確保快速的查詢回應時間。時間序列集合使用熱儲存和暖儲存組合，其中最新的資料會保留在熱儲存中，以優化更頻繁存取資料的查詢回應時間。
- 對於時間序列和向量搜尋集合，您無法依自訂文件 ID 編製索引，也無法依 upsert 請求更新。此操作保留給搜尋使用案例。您可以改為依文件 ID 更新。如需詳細資訊，請參閱the section called “支援的 OpenSearch API 操作與許可”。
- 對於搜尋和時間序列集合，您無法使用 k-NN 類型索引。

定價

AWS 會向您收取下列 OpenSearch Serverless 元件的費用：

- 資料擷取運算
- 搜尋和查詢運算
- 保留在 Amazon S3 中的儲存

一個 OCU 包含 6 GB 的 RAM、對應的 vCPU、GP3 儲存和資料傳輸到 Amazon S3。您可以計費的最小單位是每小時 0.5 個 OCU. AWS bills OCU，具有第二級精細程度。在您的帳戶陳述式中，您會看到一個以 OCU 小時為單位的運算項目，其中包含用於資料擷取的標籤和用於搜尋的標籤。 AWS 也會每月針對存放在 Amazon S3 中的資料向您收費。它不會向您收取使用 OpenSearch Dashboards 的費用。

當您建立具有備援作用中複本的集合時，您至少需要為擷取支付 1 個 OCU (0.5 個 OCU × 2)，包括主要和待命，以及為搜尋支付 1 個 OCU (0.5 個 OCU × 2)：

- 1 個用於擷取的 OCU (0.5 個 OCU × 2)，包括主要和待命
- 1 個用於搜尋的 OCU (0.5 個 OCU × 2)

如果您停用備援作用中複本，則會針對您帳戶中的第一個集合向您收取至少 1 個 OCU (0.5 個 OCU x 2) 的費用。所有後續的集合可以共用這些 OCU。

OpenSearch Serverless 會根據支援集合所需的運算能力和儲存體， OCUs 以 1 個 OCU 的增量新增額外的 OCU。您可以為您的帳戶設定 OCU 數量上限，以控制成本。

> **Note**
>
> 具有唯一 的集合 AWS KMS keys 無法與其他集合共用 OCUs。

OpenSearch Serverless 會嘗試使用所需的最低資源來考慮不斷變化的工作負載。隨時佈建OCUs 數量可能不同，也不確切。隨著時間的推移，OpenSearch Serverless 使用的演算法會持續改進，以將系統用量降至最低。

如需完整定價資訊，請參閱 Amazon OpenSearch Service 定價。

支援的 AWS 區域

OpenSearch Serverless 可在 AWS 區域 OpenSearch Service 提供的子集中使用。如需支援區域的清單，請參閱《》中的 Amazon OpenSearch Service 端點和配額AWS 一般參考。

限制

OpenSearch Serverless 具有下列限制：

- 不支援某些 OpenSearch API 操作。請參閱 the section called “支援的 OpenSearch API 操作與許可”。
- 不支援某些 OpenSearch 外掛程式。請參閱 the section called “支援的 OpenSearch 外掛程式”。
- 目前無法將您的資料從受管的 OpenSearch Service 網域自動遷移至無伺服器集合。您必須將資料從網域重新索引至集合。
- 不支援集合的跨帳戶存取權。您無法在加密或資料存取政策中包含其他帳戶的集合。
- 不支援自訂 OpenSearch 外掛程式。
- OpenSearch Serverless 集合支援自動化快照。不支援手動快照。如需詳細資訊，請參閱the section called “使用快照備份集合”。
- 不支援跨區域搜尋和複寫。
- 您可以在單一帳戶和區域中擁有的無伺服器資源數量有限制。請參閱 OpenSearch Serverless 配額。
- 向量搜尋集合中索引的重新整理間隔約為 60 秒。搜尋和時間序列集合中索引的重新整理間隔約為 10 秒。
- 碎片數量、間隔數量和重新整理間隔不可修改，並由 OpenSearch Serverless 處理。碎片策略是以集合類型和流量為基礎。例如，時間序列集合會根據寫入流量瓶頸擴展主要碎片。
- 支援 OpenSearch 版本最高達 2.1 的地理空間功能。

比較 OpenSearch Service 和 OpenSearch Serverless

在 OpenSearch Serverless 中，某些概念和功能不同於佈建的 OpenSearch Service 網域的對應功能。例如，一個重要的差異是 OpenSearch Serverless 沒有叢集或節點的概念。

下表說明 OpenSearch Serverless 中的重要功能和概念與佈建的 OpenSearch Service 網域中的同等功能有何不同。

功能	OpenSearch Service	OpenSearch Serverless
網域與集合比較	索引會保留在網域中，這些是預先佈建的 OpenSearch 叢集。 如需詳細資訊，請參閱建立和管理網域。	索引會保留在集合中，這些是代表特定工作負載或使用案例的索引邏輯分組。 如需詳細資訊，請參閱the section called “管理集合”。
節點類型和容量管理	您可以使用符合成本和效能規格的節點類型來建置叢集。您必須計算自己的儲存要求，並選擇網域的執行個體類型。 如需詳細資訊，請參閱the section called “調整網域大小”。	OpenSearch Serverless 會根據容量用量，為您的帳戶自動擴展和佈建額外的運算單元。 如需詳細資訊，請參閱the section called “管理容量限制”。
帳單	您需對於每小時 EC2 執行個體的使用，以及連接到執行個體的任何 EBS 儲存磁碟區的累積大小支付費用。 如需詳細資訊，請參閱the section called “定價”。	我們會向您收取資料擷取運算、搜尋和查詢運算以及保留在 S3 中的儲存的費用 (以 OCU 小時計算)。 如需詳細資訊，請參閱the section called “定價”。
加密	網域的靜態加密是選擇性。 如需詳細資訊，請參閱the section called “靜態加密”。	集合需要靜態加密。 如需詳細資訊，請參閱the section called “加密”。

功能	OpenSearch Service	OpenSearch Serverless
資料存取控制	網域內資料的存取權由 IAM 政策和精細存取控制決定。	集合內的資料存取權由資料存取政策決定。
支援的 OpenSearch 操作	OpenSearch Service 支援所有 OpenSearch API 操作的子集。 如需詳細資訊，請參閱the section called “受支援的操作”。	OpenSearch Serverless 支援 OpenSearch API 操作的不同子集。 如需詳細資訊，請參閱the section called “支援的操作和外掛程式”。
Dashboards 登入	使用使用者名稱和密碼登入。 如需詳細資訊，請參閱the section called “以主要使用者身分存取 OpenSearch Dashboards”。	如果您登入 AWS 主控台並導覽至儀表板 URL，您將自動登入。 如需詳細資訊，請參閱the section called “存取 OpenSearch Dashboards”。
API	使用 OpenSearch Service API 操作，以程式設計方式與 OpenSearch Service 互動。	使用 OpenSearch Serverless API 操作，以程式設計方式與 OpenSearch Serverless 互動。
網路存取	網域的網路設定會套用至網域端點和 OpenSearch Dashboards 端點。兩者的網路存取緊密結合。	網域端點和 OpenSearch Dashboards 端點的網路設定會分開。您可以選擇不設定 OpenSearch Dashboards 的網路存取。 如需詳細資訊，請參閱the section called “網路存取”。
簽署請求	使用 OpenSearch 高階和低階 REST 用戶端來簽署請求。將服務名稱指定為 `es`。	OpenSearch Serverless 此時支援 OpenSearch Service 支援的用戶端子集。 簽署請求時，請將服務名稱指定為 `aoss`。需要有 `x-amz-content-sha256` 標頭。如需詳細資訊，請參閱the section called “其他用戶端”。

功能	OpenSearch Service	OpenSearch Serverless
OpenSearch 版本升級	新版本的 OpenSearch 推出時，您可以手動升級網域。您有責任確保您的網域符合升級要求，並且已解決任何突破性變更。	OpenSearch Serverless 會自動將您的集合升級至新的 OpenSearch 版本。一旦有新版本可用，不一定需要升級。
服務軟體更新	您可以在服務軟體更新可用時，手動將更新套用至您的網域。	OpenSearch Serverless 將自動更新您的集合，以使用最新錯誤修正、功能和效能改善。
VPC 存取	您可以在 VPC 中佈建您的網域。 您也可以建立其他 OpenSearch Service 受管 VPC 端點來存取網域。	您可以為帳戶建立一個或多個 OpenSearch Serverless 受管 VPC 端點。然後，請將這些端點包含在網路政策中。
SAML 身分驗證	您可以針對每個網域啟用 SAML 身分驗證。 如需詳細資訊，請參閱the section called “OpenSearch Dashboards 的 SAML 身分驗證”。	您可以在帳戶層級設定一個或多個 SAML 提供者，然後在資料存取政策中包含相關聯的使用者和群組 ID。 如需詳細資訊，請參閱the section called “SAML 身分驗證”。
Transport Layer Security (TLS)	OpenSearch Service 支援 TLS 1.2，但建議您使用 TLS 1.3。	OpenSearch Serverless 支援 TLS 1.2，但建議您使用 TLS 1.3。

教學課程：Amazon OpenSearch Serverless 入門

本教學課程會逐步引導您快速啟動和執行 Amazon OpenSearch Serverless 搜尋集合的基本步驟。搜尋集合可讓您為內部網路中的應用程式和面向網際網路的應用程式提供支援，例如電子商務網站搜尋和內容搜尋。

若要了解如何使用向量搜尋集合，請參閱 the section called “使用向量搜尋集合”。如需使用集合的詳細資訊，請參閱本指南中的 the section called “管理集合”和其他主題。

在本教學課程中，您會完成下列步驟：

1. 設定許可
2. 建立集合
3. 上傳並搜尋資料
4. 刪除集合

> **Note**
>
> 建議您僅針對 使用 ASCII 字元`IndexName`。如果您不將 ASCII 字元用於 `IndexName`，CloudWatch 指標`IndexName`中的 會轉換為非 ASCII 字元的 URL 編碼格式。

步驟 1：設定許可

若要完成本教學課程並在一般情形下使用 OpenSearch Serverless，您必須擁有正確的 IAM 許可。在本教學課程中，您將建立集合、上傳並搜尋資料，然後刪除該集合。

使用者或角色必須連接身分型政策，該政策包含以下最低許可：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:CreateCollection",
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:DeleteCollection",
        "aoss:CreateAccessPolicy",
        "aoss:ListAccessPolicies",
        "aoss:UpdateAccessPolicy",
        "aoss:CreateSecurityPolicy",
```

```
        "aoss:GetSecurityPolicy",
        "aoss:UpdateSecurityPolicy",
        "iam:ListUsers",
        "iam:ListRoles"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

如需有關 OpenSearch Serverless IAM 許可的詳細資訊，請參閱 the section called “身分和存取權管理”。

步驟 2：建立集合

集合是一組 OpenSearch 索引，可共同運作以支援特定工作負載或使用案例。

建立 OpenSearch Serverless 集合

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇 Collections (集合)，然後選擇 Create collection (建立集合)。
3. 將集合命名為 movies。
4. 對於集合類型，選擇 Search (搜尋)。如需詳細資訊，請參閱選擇集合類型。
5. 針對安全性，選擇標準建立。
6. 在加密下，選取使用 AWS 擁有的金鑰。這是 AWS KMS key OpenSearch Serverless 用來加密資料的 。
7. 在 Network (網路) 下，設定集合的網路設定。
   - 對於存取類型，選取 Public (公用)。
   - 針對 資源類型，選擇啟用對 OpenSearch 端點的存取和啟用對 OpenSearch Dashboards 的存取。您將使用 OpenSearch Dashboards 上傳並搜尋資料，因此您需同時啟用兩者。
8. 選擇下一步。
9. 對於 Configure data access (設定資料存取)，請設定集合的存取設定。資料存取政策讓使用者和角色可以存取集合中的資料。在本教學課程中，我們將為單一使用者提供為 movies 集合中資料編製索引和進行搜尋所需的許可。

建立可存取 movies 集合的單一規則。將規則命名為 Movies collection access (電影集合存取)。

10. 選擇 Add principals (新增主體)、IAM users and roles (IAM 使用者和角色)，然後選取您將用來登入 OpenSearch Dashboards，並為資料編制索引的使用者或角色。選擇儲存。
11. 在 Index permissions (索引許可) 下，選取所有許可。
12. 選擇下一步。
13. 對於存取政策設定，請選擇 Create a new data access policy (建立新的資料存取政策) 並命名政策 movies。
14. 選擇下一步。
15. 檢閱集合設定，然後選擇 Submit (提交)。等待幾分鐘，讓收集狀態變成 `Active`。

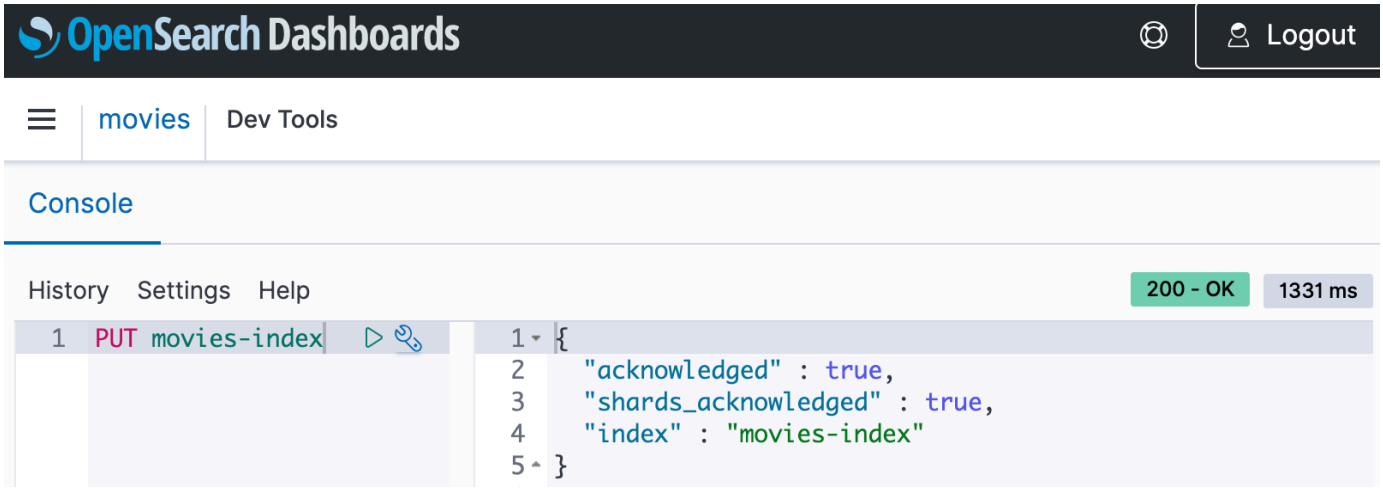
步驟 3：上傳並搜尋資料

您可以使用 Postman 或 cURL 將資料上傳至 OpenSearch Serverless 集合。為了簡潔起見，這些範例使用 OpenSearch Dashboards 主控台中的 Dev Tools (開發工具)。

為 movies 集合中的資料編製索引和進行搜尋

1. 在左側導覽窗格中選擇 Collections (集合)，然後選擇 movies 集合以開啟其詳細資訊頁面。
2. 選擇集合的 OpenSearch Dashboards URL。URL 採用的格式為 `https://dashboards.{region}.aoss.amazonaws.com/_login/?collectionId={collection-id}`。
3. 在 OpenSearch Dashboards 中，開啟左側導覽窗格，然後選擇 Dev Tools (開發工具)。
4. 若要建立名為 movies-index 的單一索引，請傳送以下請求：

```
PUT movies-index
```



5. 若要將單一文件的索引編製為 movies-index，請傳送以下請求：

```
PUT movies-index/_doc/1
{
  "title": "Shawshank Redemption",
  "genre": "Drama",
  "year": 1994
}
```

6. 若要在 OpenSearch Dashboards 中搜尋資料，您需要設定至少一個索引模式。OpenSearch 使用這些模式來識別您要分析的索引。開啟左側導覽窗格，依序選擇 Stack Management (堆疊管理)、Index Patterns (索引模式)，然後選擇 Create index pattern (建立索引模式)。對於本教學課程，輸入 movies。
7. 選擇 Next step (下一步)，然後選擇 Create index pattern (建立索引模式)。建立模式之後，您可以檢視各種文件欄位，例如 `title` 和 `genre`。
8. 若要開始搜尋資料，請再次開啟左側導覽窗格，然後選擇 Discover (探索)，或使用開發工具中的搜尋 API。

處理錯誤

執行索引和搜尋操作時，您可能會收到下列錯誤回應：

- HTTP `507` – 表示發生內部伺服器錯誤。此錯誤通常表示 OpenSearch 運算單位 OCUs) 因請求的磁碟區或複雜性而超載。雖然 OpenSearch Serverless 會自動擴展以管理負載，但部署其他資源可能會延遲。

若要緩解此錯誤，請實作指數退避重試政策。此方法會暫時降低請求率，以有效管理負載。如需詳細資訊，請參閱 SDK AWS SDKs 和工具參考指南中的重試行為。

- HTTP 402 – 表示您已達到 OpenSearch 運算單位 (OCU) 容量上限。最佳化工作負載以減少 OCU 用量或請求增加配額。

步驟 4：刪除集合

movies 集合用於測試目的，因此請確保在完成實驗後將其刪除。

刪除 OpenSearch Serverless 集合

1. 返回 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇 Collections (集合)，然後選擇 movies 集合。
3. 選擇 Delete (刪除)，並確認刪除。

後續步驟

您現在已知道如何建立集合並為資料編製索引，您可能想要嘗試以下一些練習：

- 查看用於建立集合的更多進階選項。如需詳細資訊，請參閱the section called “管理集合”。
- 了解如何設定安全政策以大規模管理集合安全性。如需詳細資訊，請參閱the section called “OpenSearch Serverless 的安全功能”。
- 探索將資料的索引編製為集合的其他方法。如需詳細資訊，請參閱the section called “將資料擷取至集合”。

Amazon OpenSearch Serverless 集合

Amazon OpenSearch Serverless 中的集合是代表分析工作負載的一個或多個索引的邏輯分組。OpenSearch Serverless 會自動管理和調整集合，需要最少的手動輸入。

主題

- 管理 Amazon OpenSearch Serverless 集合
- 使用向量搜尋集合
- 搭配 Amazon OpenSearch Serverless 使用資料生命週期政策
- 使用 AWS SDKs與 Amazon OpenSearch Serverless 互動

- 使用 CloudFormation 建立 Amazon OpenSearch Serverless 集合
- 使用快照備份集合
- Amazon OpenSearch Serverless 中的 Zstandard Codec 支援
- 使用衍生來源儲存儲存體

管理 Amazon OpenSearch Serverless 集合

Amazon OpenSearch Serverless 中的集合是代表分析工作負載的一個或多個索引的邏輯分組。OpenSearch Serverless 會自動管理和調整集合，需要最少的手動輸入。

主題

- 設定集合的許可
- Serverless 的自動語意擴充
- 建立集合
- 存取 OpenSearch Dashboards
- 檢視集合
- 刪除集合

設定集合的許可

OpenSearch Serverless 使用下列 AWS Identity and Access Management (IAM) 許可來建立和管理集合。您可以指定 IAM 條件，將使用者限制在特定集合內。

- `aoss:CreateCollection` ：建立集合。
- `aoss:ListCollections` ：列出目前帳戶中的集合。
- `aoss:BatchGetCollection` ：取得有關一個或多個集合的詳細資訊。
- `aoss:UpdateCollection` ：修改集合。
- `aoss:DeleteCollection` ：刪除集合。

下列身分型存取政策範例提供使用者管理名為 Logs 的單一集合所需的最低許可：

```
[
   {
      "Sid":"Allows managing logs collections",
      "Effect":"Allow",
```

```
      "Action":[
         "aoss:CreateCollection",
         "aoss:ListCollections",
         "aoss:BatchGetCollection",
         "aoss:UpdateCollection",
         "aoss:DeleteCollection",
         "aoss:CreateAccessPolicy",
         "aoss:CreateSecurityPolicy"
      ],
      "Resource":"*",
      "Condition":{
         "StringEquals":{
            "aoss:collection":"Logs"
         }
      }
   }
]
```

需要加密、網路和資料存取政策，才能讓集合正常運作，因此要將 `aoss:CreateAccessPolicy` 和 `aoss:CreateSecurityPolicy` 包含在內。如需詳細資訊，請參閱the section called “身分和存取權管理”。

> **Note**
>
> 如果您要在帳戶中建立第一個集合，您也需要 `iam:CreateServiceLinkedRole` 許可。如需詳細資訊，請參閱the section called “集合建立角色”。

Serverless 的自動語意擴充

簡介

相較於語義搜尋，自動語意擴充功能可協助將搜尋相關性提高高達 20%。自動語意擴充可免除管理您自己的 ML （機器學習） 模型基礎設施以及與搜尋引擎整合的繁重工作。此功能適用於所有三種無伺服器集合類型：搜尋、時間序列和向量。

什麼是語意搜尋

傳統搜尋引擎依賴word-to-word字比對 （稱為語彙搜尋） 來尋找查詢的結果。雖然這適用於電視型號等特定查詢，但它難以進行更抽象的搜尋。例如，在搜尋 "shoes for the Beach" 時，語彙搜尋只會比對目錄項目中的個別單字 "shoes"、"beach"、"for" 和 "the"，可能會缺少不包含確切搜尋詞彙的相關產品，例如 "water-resistantss" 或 "surf shoes"。

語意搜尋會傳回查詢結果，不僅包含關鍵字比對，還包含使用者搜尋的意圖和內容意義。例如，如果使用者搜尋「如何治療問題」，語意搜尋系統可能會傳回下列結果：

- Migraine 補救措施
- 疼痛管理技巧
- Over-the-counter止痛藥

模型詳細資訊和效能基準

雖然此功能可處理幕後的技術複雜性，而不會暴露基礎模型，但我們透過簡短的模型描述和基準結果來提供透明度，協助您在關鍵工作負載中做出採用特徵的明智決策。

自動語意擴充使用服務受管、預先訓練的稀疏模型，可有效運作，而無需自訂微調。此模型會分析您指定的欄位，並根據從各種訓練資料中學到的關聯將它們擴展到稀疏向量。展開的詞彙及其重要性權重會以原生 Lucene 索引格式存放，以便有效擷取。我們已使用僅限文件模式最佳化此程序，其中編碼只會在資料擷取期間發生。搜尋查詢只是字符化，而不是透過稀疏模型處理，使解決方案同時符合成本效益和效能。

我們在功能開發期間的效能驗證使用 MS MARCO 段落擷取資料集，段落平均 334 個字元。對於相關性評分，我們在英文內容的 BEIR 基準上測量了前 10 個搜尋結果的平均標準化折扣累積收益 (NDCG) (ndcg@10)，在多語言內容的 MIRACL 上測量了平均 ndcg@10。我們透過用戶端的第 90 個百分位數 (p90) 測量和搜尋回應 p90 取得值來評估延遲。這些基準提供搜尋相關性和回應時間的基準效能指標。以下是關鍵基準號碼 -

- 英文 - 相較於語彙搜尋，相關性改善 20%。它也比語義搜尋降低了 7.7% 的 P90 搜尋延遲 (BM25 為 26 毫秒，而自動語意擴充為 24 毫秒）。
- 多語言 - 相較於語意搜尋，相關性改善 105%，而相較於語意搜尋，P90 搜尋延遲增加 38.4% (BM25 為 26 毫秒，自動語意擴充為 36 毫秒）。

鑑於每個工作負載的獨特性質，我們建議您在做出實作決策之前，先使用自己的基準標準在開發環境中評估此功能。

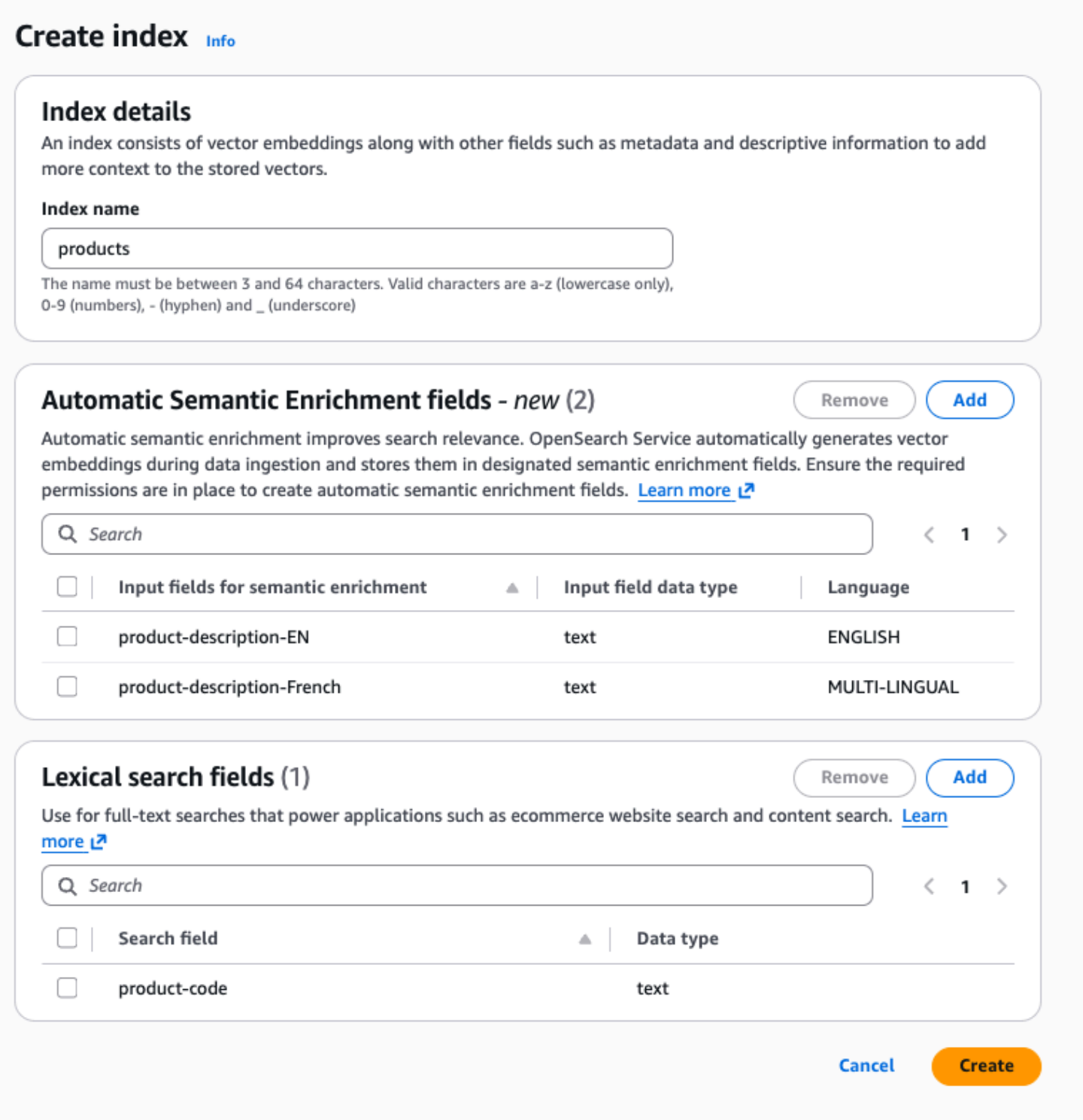
支援的語言

此功能支援英文。此外，模型也支援阿拉伯文、孟加拉文、中文、芬蘭文、法文、印地文、印尼文、日文、韓文、波斯文、俄文、西班牙文、斯瓦希里文和特拉古文。

為無伺服器集合設定自動語意擴充索引

為您的文字欄位設定啟用自動語意擴充的索引非常簡單，您可以在新索引建立期間透過主控台、APIs 和 CloudFormation 範本進行管理。若要為現有索引啟用它，您需要為文字欄位重新建立已啟用自動語意擴充的索引。

主控台體驗 - AWS 主控台可讓您使用自動語意擴充欄位輕鬆建立索引。選取集合後，您會在主控台頂端找到建立索引按鈕。按一下建立索引按鈕後，您會找到定義自動語意擴充欄位的選項。在一個索引中，您可以結合英文和多語言的自動語意擴充，以及語彙欄位。



API 體驗 - 若要使用 AWS 命令列界面 (AWS CLI) 建立自動語意擴充索引，請使用 create-index 命令：

```
aws opensearchserverless create-index \
--id [collection_id] \
--index-name [index_name] \
```

```
--index-schema [index_body] \
```

在下列索引結構描述範例中， title_semantic 欄位的欄位類型設定為文字，且參數 semantic_enrichment 設定為狀態 ENABLED。設定語意_enrichment 參數可在 title_semantic 欄位上啟用自動語意擴充。您可以使用 language_options 欄位來指定英文或多語言。

```
    aws opensearchserverless create-index \
    --id XXXXXXXXX \
    --index-name 'product-catalog' \
    --index-schema '{
    "mappings": {
        "properties": {
            "product_id": {
                "type": "keyword"
            },
            "title_semantic": {
                "type": "text",
                "semantic_enrichment": {
                    "status": "ENABLED",
                    "language_options": "english"
                }
            },
            "title_non_semantic": {
                "type": "text"
            }
        }
    }
}'
```

若要描述建立的索引，請使用下列命令：

```
aws opensearchserverless get-index \
--id [collection_id] \
--index-name [index_name] \
```

您也可以使用 CloudFormation 範本 (Type：AWS::OpenSearchServerless::CollectionIndex)，在集合佈建期間和建立集合之後建立語意搜尋。

資料擷取和搜尋

建立啟用自動語意擴充的索引後，此功能會在資料擷取程序期間自動運作，而不需要額外的組態。

資料擷取：當您將文件新增至索引時，系統會自動：

- 分析您為語意擴充指定的文字欄位
- 使用 OpenSearch Service 受管稀疏模型產生語意編碼
- 將這些豐富的表示與原始資料一起存放

此程序使用 OpenSearch 的內建 ML 連接器和擷取管道，這些管道會在幕後自動建立和管理。

搜尋：語意擴充資料已編製索引，因此查詢會有效率地執行，而不會再次叫用 ML 模型。這表示您可以改善搜尋相關性，而不需要額外的搜尋延遲額外負荷。

設定自動語意擴充的許可

在建立自動語意擴充索引之前，您需要設定必要的許可。本節說明所需的許可，以及如何設定這些許可。

IAM 政策許可

使用下列 AWS Identity and Access Management (IAM) 政策授予使用自動語意擴充的必要許可：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AutomaticSemanticEnrichmentPermissions",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateIndex",
                "aoss:GetIndex",
                "aoss:UpdateIndex",
                "aoss:DeleteIndex",
                "aoss:APIAccessAll"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

金鑰許可

- aoss:*Index 許可啟用索引管理
- aoss:APIAccessAll 許可允許 OpenSearch API 操作
- 若要限制特定集合的許可，請將 "Resource": "*"取代為集合的 ARN

設定資料存取許可

若要設定自動語意擴充的索引，您必須擁有適當的資料存取政策，以授予存取索引、管道和模型收集資源的許可。如需資料存取政策的詳細資訊，請參閱 Amazon OpenSearch Serverless 的資料存取控制。如需設定資料存取政策的程序，請參閱 建立資料存取政策 (主控台)。

資料存取許可

```
[
    {
        "Description": "Create index permission",
        "Rules": [
            {
                "ResourceType": "index",
                "Resource": ["index/collection_name/*"],
                "Permission": [
                  "aoss:CreateIndex",
                  "aoss:DescribeIndex",
                  "aoss:UpdateIndex",
                  "aoss:DeleteIndex"
                ]
            }
        ],
        "Principal": [
            "arn:aws:iam::account_id:role/role_name"
        ]
    },
    {
        "Description": "Create pipeline permission",
        "Rules": [
            {
                "ResourceType": "collection",
```

```
                "Resource": ["collection/collection_name"],
                "Permission": [
                  "aoss:CreateCollectionItems",
                  "aoss:DescribeCollectionItems"
                ]
            }
        ],
        "Principal": [
            "arn:aws:iam::account_id:role/role_name"
        ]
    },
    {
        "Description": "Create model permission",
        "Rules": [
            {
                "ResourceType": "model",
                "Resource": ["model/collection_name/*"],
                "Permission": ["aoss:CreateMLResource"]
            }
        ],
        "Principal": [
            "arn:aws:iam::account_id:role/role_name"
        ]
    },
]
```

網路存取許可

若要允許服務 APIs 存取私有集合，您必須設定允許服務 API 與集合之間必要存取的網路政策。如需網路政策的詳細資訊，請參閱 Amazon OpenSearch Serverless 的網路存取。

```
[
   {
      "Description":"Enable automatic semantic enrichment in a private collection",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/collection_name"
            ]
         }
      ],
      "AllowFromPublic":false,
```

```
      "SourceServices":[
         "aoss.amazonaws.com"
      ],
   }
]
```

設定私有集合的網路存取許可

1. 登入 OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽中，選擇網路政策。然後執行下列其中一項：
   - 選擇現有的政策名稱，然後選擇編輯
   - 選擇建立網路政策並設定政策詳細資訊
3. 在存取類型區域中，選擇私有（建議），然後選取AWS 服務私有存取。
4. 在搜尋欄位中，選擇服務，然後選擇 aoss.amazonaws.com。
5. 在資源類型區域中，選取啟用對 OpenSearch 端點的存取方塊。
6. 在搜尋集合（搜尋）或輸入特定字首字詞（搜尋）欄位中，選取集合名稱。然後輸入或選取要與網路政策建立關聯的集合名稱。
7. 選擇建立新網路政策或更新現有網路政策。

查詢重寫

自動語意擴充會自動將您現有的「比對」查詢轉換為語意搜尋查詢，而不需要修改查詢。如果比對查詢是複合查詢的一部分，系統會周遊您的查詢結構、尋找比對查詢，並將它們取代為神經稀疏查詢。目前，此功能僅支援取代「比對」查詢，無論是獨立查詢還是複合查詢的一部分。不支援「多重_比對」。此外，此功能支援所有複合查詢來取代其巢狀比對查詢。複合查詢包括：bool、Boosting、constant_score、dis_max、 function_score 和 hybrid。

自動語意擴充的限制

當套用至包含自然語言內容small-to-medium欄位時，自動語意搜尋最有效，例如電影標題、產品描述、評論和摘要。雖然語意搜尋可增強大多數使用案例的相關性，但可能不適用於某些案例。在決定是否為特定使用案例實作自動語意擴充時，請考慮下列限制。

- 非常長的文件 – 目前的稀疏模型只會處理每個英文文件的前 8，192 個字符。對於多語言文件，它是 512 個字符。對於冗長的文章，請考慮實作文件區塊，以確保完整的內容處理。
- 日誌分析工作負載 – 語意擴充會大幅增加索引大小，這對於日誌分析而言可能不必要，其中完全相符通常就足夠。其他語意內容很少改善日誌搜尋的有效性，足以證明增加的儲存需求。

- 自動語意擴充與衍生來源功能不相容。
- 調節 – OpenSearch Serverless 的索引推論請求目前上限為 100 TPS。這是軟性限制；請聯絡 AWS Support 以取得更高的限制。

定價

OpenSearch Serverless 會根據索引時間產生稀疏向量時消耗的 OpenSearch Compute Units (OCUs) 來計費自動語意擴充。您只需支付索引期間實際用量的費用。您可以使用 Amazon CloudWatch 指標 SemanticSearchOCU 監控此消耗。如需模型字符限制、每個 OCU 的磁碟區輸送量，以及範例計算的特定詳細資訊，請造訪 OpenSearch Service 定價。

建立集合

您可以使用 主控台或 AWS CLI 來建立無伺服器集合。這些步驟涵蓋如何建立搜尋或時間序列集合。若要建立向量搜尋集合，請參閱 the section called “使用向量搜尋集合”。

主題

- 建立集合 (主控台)
- 建立集合 (CLI)

建立集合 (主控台)

使用本節中的程序，透過使用 建立集合 AWS 管理主控台。這些步驟涵蓋如何建立搜尋或時間序列集合。若要建立向量搜尋集合，請參閱 the section called “使用向量搜尋集合”。

主題

- 設定集合設定
- 設定其他搜尋欄位

設定集合設定

使用下列程序來設定集合的相關資訊。

使用主控台設定集合設定

1. 導覽至 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home/。
2. 展開左側導覽窗格中的 Serverless (無伺服器)，然後選擇 Collections (集合)。

3. 選擇 Create collection (建立集合)。
4. 提供集合的名稱和描述。名稱必須符合下列條件：

   - 對您的帳戶和 是唯一的 AWS 區域
   - 只能包含小寫字母 a-z、數字 0-9 和連字號 (-)
   - 包含 3 到 32 個之間字元數
5. 選擇集合類型：

   - Time series (時間序列)：著重於分析大量半結構化、機器產生之資料的日誌分析區段。至少 24 小時的資料會存放在熱索引上，其餘資料會保留在暖儲存中。
   - 搜尋：支援內部網路中的應用程式以及面向網際網路之應用程式的全文檢索搜尋。所有搜尋資料均存放在熱儲存中，以確保快速的查詢回應時間。

     > **Note**
     >
     > 如果您要啟用自動語意搜尋，請選擇此選項，如 中所述設定集合設定。

   - 向量搜尋 – 向量內嵌的語意搜尋，可簡化向量資料管理。支援機器學習 (ML) 增強型搜尋體驗和生成式 AI 應用程式，例如聊天機器人、個人助理和詐騙偵測。

   如需詳細資訊，請參閱the section called “選擇集合類型”。
6. 針對部署類型，選擇集合的備援設定。根據預設，每個集合都有備援，這表示索引和搜尋 OpenSearch Compute Units (OCUs) 各自在不同的可用區域中都有自己的待命複本。基於開發和測試目的，您可以選擇停用備援，將集合中的 OCUs 數量減少到兩個。如需詳細資訊，請參閱the section called “運作方式”。
7. 針對安全性，選擇標準建立。
8. 針對加密，選擇用來加密資料的 AWS KMS 金鑰。如果您輸入的集合名稱符合加密政策中定義的模式，OpenSearch Serverless 會通知您。您可以選擇保留此相符項目，也可以使用唯一的加密設定進行覆寫。如需詳細資訊，請參閱the section called “加密”。
9. 針對網路存取設定，設定集合的網路存取。

   - 針對存取類型，選取公有或私有。

     如果您選擇私有，請指定哪些 VPC 端點和 AWS 服務 可以存取集合。

     - 用於存取的 VPC 端點 – 指定一或多個 VPC 端點以允許存取。若要建立 VPC 端點，請參閱 the section called “資料平面 VPC 端點”。

- AWS 服務 私有存取 – 選取要允許存取的一或多個支援服務。
- 針對資源類型，選取使用者是否可以透過其 OpenSearch 端點（透過 cURL、Postman 等進行 API 呼叫）、透過 OpenSearch Dashboards 端點（使用視覺化效果並透過主控台進行 API 呼叫）或兩者存取集合。

> **Note**
>
> AWS 服務 私有存取僅適用於 OpenSearch 端點，不適用於 OpenSearch Dashboards 端點。

如果您輸入的集合名稱符合網路政策中定義的模式，OpenSearch Serverless 會通知您。您可以選擇保留此相符項目，也可以使用自訂網路設定進行覆寫。如需詳細資訊，請參閱the section called “網路存取”。

10. (選用) 將一個或多個標籤新增至集合。如需詳細資訊，請參閱the section called “標記集合”。
11. 選擇下一步。

設定其他搜尋欄位

您在建立集合工作流程第 2 頁看到的選項取決於您正在建立的集合類型。本節說明如何為每個集合類型設定其他搜尋欄位。本節也說明如何設定自動語意擴充。略過任何不適用於集合類型的區段。

主題

- 設定自動語意擴充
- 設定時間序列搜尋欄位
- 設定語彙搜尋欄位
- 設定向量搜尋欄位

設定自動語意擴充

當您建立或編輯集合時，您可以設定自動語意擴充，以簡化 Amazon OpenSearch Service 中的語意搜尋實作和功能。語意搜尋會傳回查詢結果，不僅包含關鍵字比對，還包含使用者搜尋的意圖和內容意義。如需詳細資訊，請參閱Serverless 的自動語意擴充。

設定自動語意擴充

1. 在索引詳細資訊區段中，針對索引名稱指定名稱。
2. 在自動語意擴充欄位區段中，選擇新增語意搜尋欄位。
3. 在語意擴充欄位的輸入欄位名稱中，輸入您要擴充的欄位名稱。
4. 資料類型為文字。這無法變更。
5. 針對語言，選擇英文或多語言。
6. 選擇新增欄位。
7. 完成集合的選用欄位設定後，請選擇下一步。檢閱您的變更，然後選擇提交以建立集合。

設定時間序列搜尋欄位

時間序列搜尋欄位區段中的選項與時間序列資料和資料串流相關。如需這些主題的詳細資訊，請參閱使用資料串流管理 Amazon OpenSearch Service 中的時間序列資料。

設定時間序列搜尋欄位

1. 在時間序列搜尋欄位區段中，選擇新增時間序列欄位。
2. 針對欄位名稱，輸入名稱。
3. 針對資料類型，從清單中選擇類型。
4. 選擇新增欄位
5. 完成集合的選用欄位設定後，請選擇下一步。檢閱您的變更，然後選擇提交以建立集合。

設定語彙搜尋欄位

文字搜尋會尋找搜尋查詢與索引詞彙或關鍵字之間的完全相符項目。

設定語彙搜尋欄位

1. 在語彙搜尋欄位區段中，選擇新增搜尋欄位。
2. 針對欄位名稱，輸入名稱。
3. 針對資料類型，從清單中選擇類型。
4. 選擇新增欄位
5. 完成集合的選用欄位設定後，請選擇下一步。檢閱您的變更，然後選擇提交以建立集合。

設定向量搜尋欄位

設定向量搜尋欄位

1. 在向量欄位區段中，選擇新增向量欄位。
2. 針對欄位名稱，輸入名稱。
3. 針對引擎，從清單中選擇類型。
4. 輸入維度的數量。
5. 針對距離指標，從清單中選擇類型。
6. 完成集合的選用欄位設定後，請選擇下一步。
7. 檢閱您的變更，然後選擇提交以建立集合。

建立集合 (CLI)

使用本節中的程序，使用 建立 OpenSearch Serverless 集合 AWS CLI。

主題

- 開始之前
- 建立集合
- 使用自動語意擴充索引建立集合

開始之前

使用 建立集合之前 AWS CLI，請使用下列程序來建立集合所需的政策。

> **Note**
>
> 在下列每個程序中，當您指定集合的名稱時，名稱必須符合下列條件：
>
> - 對您的帳戶和 是唯一的 AWS 區域
> - 只能包含小寫字母 a-z、數字 0-9 和連字號 (-)
> - 包含 3 到 32 個之間字元數

建立集合的必要政策

1. 開啟 AWS CLI 並執行下列命令，以建立資源模式符合集合預期名稱的加密政策。

```
&aws opensearchserverless create-security-policy \
  --name policy name \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/collection name\"]}],\"AWSOwnedKey\":true}"
```

例如，如果您計劃將集合命名為 logs-application，則可以建立如下所示的加密政策：

```
&aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/logs-application\"]}],\"AWSOwnedKey\":true}"
```

如果您計劃將政策用於其他集合，則可以使規則的範圍更廣泛，例如 `collection/logs*` 或 `collection/*`。

2. 執行下列命令，使用網路政策來設定集合的網路設定。您可以在建立集合之後建立網路政策，但建議您事先執行此操作。

```
&aws opensearchserverless create-security-policy \
  --name policy name \
  --type network --policy "[{\"Description\":\"description\",\"Rules\":
[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/collection name\"]},
{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/collection name\"]}],
\"AllowFromPublic\":true}]"
```

使用先前的 logs-application 範例，您可建立下列網路政策：

```
&aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type network --policy "[{\"Description\":\"Public access for logs collection
\",\"Rules\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/logs-
application\"]},{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/logs-
application\"]}],\"AllowFromPublic\":true}]"
```

建立集合

下列程序使用 CreateCollection API 動作來建立類型 SEARCH或 的集合TIMESERIES。如果您未在請求中指定集合類型，即會預設為 TIMESERIES。如需這些類型的詳細資訊，請參閱 選擇集合類型。若要建立向量搜尋集合，請參閱 the section called “使用向量搜尋集合”。

如果您的集合使用 加密 AWS 擁有的金鑰，則 kmsKeyArnauto不是 ARN。

> **Important**
>
> 建立集合後，除非該集合符合資料存取政策，否則您將無法存取它。如需詳細資訊，請參閱Amazon OpenSearch Serverless 的資料存取控制。

建立集合

1. 確認您已建立 中所述的必要政策開始之前。
2. 執行下列命令。對於 ，type請指定 SEARCH或 TIMESERIES。

```
&aws opensearchserverless create-collection --name "collection name" --
type collection type --description "description"
```

使用自動語意擴充索引建立集合

使用下列程序，使用為自動語意擴充設定的索引建立新的 OpenSearch Serverless 集合。程序使用 OpenSearch Serverless CreateIndex API 動作。

使用為自動語意擴充設定的索引建立新的集合

執行下列命令來建立集合和索引。

```
&aws opensearchserverless create-index \
--region Region ID \
--id collection name --index-name index name \
--index-schema \
'mapping in json'
```

範例如下。

```
&aws opensearchserverless create-index \
--region us-east-1 \
--id conversation_history --index-name conversation_history_index \
--index-schema \
'{
    "mappings": {
        "properties": {
```

```
            "age": {
                "type": "integer"
            },
            "name": {
                "type": "keyword"
            },
            "user_description": {
                "type": "text"
            },
            "conversation_history": {
                "type": "text",
                "semantic_enrichment": {
                    "status": "ENABLED",
                    // Specifies the sparse tokenizer for processing multi-lingual text
                    "language_option": "MULTI-LINGUAL",
                    // If embedding_field is provided, the semantic embedding field
 will be set to the given name rather than original field name + "_embedding"
                    "embedding_field": "conversation_history_user_defined"
                }
            },
            "book_title": {
                "type": "text",
                "semantic_enrichment": {
                    // No embedding_field is provided, so the semantic embedding field
 is set to "book_title_embedding"
                    "status": "ENABLED",
                    "language_option": "ENGLISH"
                }
            },
            "abstract": {
                "type": "text",
                "semantic_enrichment": {
                    // If no language_option is provided, it will be set to English.
                    // No embedding_field is provided, so the semantic embedding field
 is set to "abstract_embedding"
                    "status": "ENABLED"
                }
            }
        }
    }
}'
```

存取 OpenSearch Dashboards

使用 建立集合之後 AWS 管理主控台，您可以導覽至集合的 OpenSearch Dashboards URL。您可以在左側導覽窗格中選擇集合，然後選取集合以開啟其詳細資訊頁面，以尋找儀表板 URL。URL 採用的格式為 `https://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc`。導覽至 URL 後，您將自動登入 Dashboards。

如果您已有可用的 OpenSearch Dashboards URL，但未在 上 AWS 管理主控台，從瀏覽器呼叫 Dashboards URL 將重新導向至 主控台。輸入 AWS 登入資料後，您會自動登入 Dashboards。如需有關存取 SAML 集合的資訊，請參閱使用 SAML 存取 OpenSearch Dashboards。

OpenSearch Dashboards 主控台逾時為一小時，無法設定。

> **Note**
>
> 2023 年 5 月 10 日，OpenSearch 推出了 OpenSearch Dashboards 的通用全域端點。您現在可以使用採用 格式的 URL，在瀏覽器中導覽至 OpenSearch Dashboardshttps://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc。為了確保回溯相容性，我們將繼續支援格式為的現有集合特定 OpenSearch Dashboards 端點https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/_dashboards。

檢視集合

您可以在 Amazon OpenSearch Service 主控台的集合索引標籤 AWS 帳戶 上檢視 中的現有集合。

若要列出集合及其 ID，請傳送 ListCollections 請求。

```
&aws opensearchserverless list-collections
```

回應範例

```
{
   "collectionSummaries":[
      {
         "arn":"arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "id":"07tjusf2h91cunochc",
         "name":"my-collection",
```

```
         "status":"CREATING"
      }
   ]
}
```

若要限制搜尋結果，請使用集合篩選條件。此請求會篩選對處於 ACTIVE 狀態之集合的回應：

```
&aws opensearchserverless list-collections --collection-filters '{ "status":
 "ACTIVE" }'
```

若要取得有關一個或多個集合的詳細資訊，包括 OpenSearch 端點和 OpenSearch Dashboards 端點，請傳送 BatchGetCollection 請求：

```
&aws opensearchserverless batch-get-collection --ids "07tjusf2h91cunochc"
 "1iu5usc4rame"
```

> **Note**
>
> 您可以在請求中包含 `--names` 或 `--ids`，但不能同時包含兩者。

回應範例

```
{
   "collectionDetails":[
      {
         "id": "07tjusf2h91cunochc",
         "name": "my-collection",
         "status": "ACTIVE",
         "type": "SEARCH",
         "description": "",
         "arn": "arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "kmsKeyArn": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "createdDate": 1667446262828,
         "lastModifiedDate": 1667446300769,
         "collectionEndpoint": "https://07tjusf2h91cunochc.us-
east-1.aoss.amazonaws.com",
         "dashboardEndpoint": "https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/
_dashboards"
      },
```

```
      {
         "id": "178ukvtg3i82dvopdid",
         "name": "another-collection",
         "status": "ACTIVE",
         "type": "TIMESERIES",
         "description": "",
         "arn": "arn:aws:aoss:us-east-1:123456789012:collection/178ukvtg3i82dvopdid",
         "kmsKeyArn": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "createdDate": 1667446262828,
         "lastModifiedDate": 1667446300769,
         "collectionEndpoint": "https://178ukvtg3i82dvopdid.us-
east-1.aoss.amazonaws.com",
         "dashboardEndpoint": "https://178ukvtg3i82dvopdid.us-
east-1.aoss.amazonaws.com/_dashboards"
      }
   ],
   "collectionErrorDetails":[]
}
```

刪除集合

刪除集合會刪除集合中的所有資料和索引。刪除集合後，您將無法復原集合。

使用主控台刪除集合

1. 從 Amazon OpenSearch Service 主控台的 Collections (集合) 面板中，選取您要刪除的集合。
2. 選擇 Delete (刪除)，並確認刪除。

若要使用 刪除集合 AWS CLI，請傳送 DeleteCollection 請求：

```
&aws opensearchserverless delete-collection --id 07tjusf2h91cunochc
```

回應範例

```
{
   "deleteCollectionDetail":{
      "id":"07tjusf2h91cunochc",
      "name":"my-collection",
      "status":"DELETING"
   }
```

```
}
```

使用向量搜尋集合

OpenSearch Serverless 中的向量搜尋集合類型提供可擴展且高效能的相似性搜尋功能。這可讓您輕鬆地建置現代機器學習 (ML) 增強型搜尋體驗和生成式人工智慧 (AI) 應用程式，而無需管理基礎向量資料庫基礎設施。

向量搜尋集合的使用案例包括影像搜尋、文件搜尋、音樂擷取、產品建議、影片搜尋、以位置為基礎的搜尋、詐騙偵測和異常偵測。

由於 OpenSearch Serverless 的向量引擎由 OpenSearch 中的 k 近鄰 (k-NN) 搜尋功能提供支援，因此您可以簡單地取得相同的功能和無伺服器環境。引擎支援 k-NN 外掛程式 API。透過這些操作，您可以利用全文搜尋、進階篩選、彙總、地理空間查詢、巢狀查詢，以更快速地擷取資料，以及增強的搜尋結果。

向量引擎提供距離指標，例如歐幾里德距離、餘弦相似性和點產品相似性，可容納 16，000 個維度。您可以存放具有各種中繼資料資料類型的欄位，例如數字、布林值、日期、關鍵字和地理位置。您也可以使用文字存放欄位以取得描述性資訊，以將更多內容新增至存放的向量。協調資料類型可降低複雜性、提高可維護性，並避免資料重複、版本相容性挑戰和授權問題。

> **Note**
>
> 記下以下資訊：
>
> - Amazon OpenSearch Serverless 支援 Faiss 16 位元純量量化，可用於在 32 位元浮點數和 16 位元向量之間執行轉換。若要進一步了解，請參閱 Faiss 16 位元純量量化。您也可以使用二進位向量來降低記憶體成本。如需詳細資訊，請參閱二進位向量。
> - Amazon OpenSearch Serverless 支援磁碟型向量搜尋。磁碟型向量搜尋可大幅降低低記憶體環境中向量工作負載的操作成本。如需詳細資訊，請參閱磁碟型向量搜尋。

向量搜尋集合入門

在本教學課程中，您將完成下列步驟，以即時儲存、搜尋和擷取向量內嵌：

1. 設定許可
2. 建立集合
3. 上傳並搜尋資料

4. 刪除集合

步驟 1：設定許可

若要完成本教學課程（以及一般使用 OpenSearch Serverless），您必須擁有正確的 AWS Identity and Access Management (IAM) 許可。在本教學課程中，您會建立集合、上傳和搜尋資料，然後刪除集合。

使用者或角色必須連接身分型政策，該政策包含以下最低許可：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:CreateCollection",
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:DeleteCollection",
        "aoss:CreateAccessPolicy",
        "aoss:ListAccessPolicies",
        "aoss:UpdateAccessPolicy",
        "aoss:CreateSecurityPolicy",
        "iam:ListUsers",
        "iam:ListRoles"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

如需有關 OpenSearch Serverless IAM 許可的詳細資訊，請參閱 the section called “身分和存取權管理”。

步驟 2：建立集合

集合是一組 OpenSearch 索引，可一起運作以支援特定工作負載或使用案例。

建立 OpenSearch Serverless 集合

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇 Collections (集合)，然後選擇 Create collection (建立集合)。
3. 命名集合外殼。
4. 針對集合類型，選擇向量搜尋。如需詳細資訊，請參閱the section called “選擇集合類型”。
5. 在部署類型下，清除啟用備援 （ 作用中複本 ）。這會在開發或測試模式中建立集合，並將集合中的 OpenSearch 運算單位 (OCUs數目減少為兩個。如果您想要在本教學課程中建立生產環境，請保持選取核取方塊。
6. 在安全下，選取輕鬆建立以簡化您的安全組態。根據預設，向量引擎中的所有資料都會在傳輸中和靜態時加密。向量引擎支援精細的 IAM 許可，讓您可以定義誰可以建立、更新和刪除加密、網路、集合和索引。
7. 選擇下一步。
8. 檢閱集合設定，然後選擇 Submit (提交)。等待幾分鐘，讓收集狀態變成 `Active`。

步驟 3：上傳並搜尋資料

索引是具有常見資料結構描述的文件集合，可讓您儲存、搜尋和擷取向量內嵌和其他欄位。您可以使用 OpenSearch Dashboards 中的開發工具主控台或 Postman 或 awscurl 等 HTTP 工具，在 OpenSearch Serverless 集合中建立和上傳資料至索引。本教學課程使用開發工具。

索引和搜尋房屋集合中的資料

1. 若要為新集合建立單一索引，請在開發工具主控台中傳送下列請求。根據預設，這會建立具有`nmslib`引擎和歐幾里得距離的索引。

```
PUT housing-index
{
   "settings": {
      "index.knn": true
   },
   "mappings": {
      "properties": {
         "housing-vector": {
            "type": "knn_vector",
            "dimension": 3
         },
```

```
         "title": {
            "type": "text"
         },
         "price": {
            "type": "long"
         },
         "location": {
            "type": "geo_point"
         }
      }
   }
}
```

2. 若要將單一文件編製索引為 housing-index，請傳送下列請求：

```
POST housing-index/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. 若要搜尋與索引中屬性類似的屬性，請傳送下列查詢：

```
GET housing-index/_search
{
    "size": 5,
    "query": {
        "knn": {
            "housing-vector": {
                "vector": [
                    10,
                    20,
                    30
                ],
                "k": 5
            }
        }
```

```
    }
}
```

步驟 4：刪除集合

由於房屋集合是用於測試目的，因此請務必在完成實驗時將其刪除。

刪除 OpenSearch Serverless 集合

1. 返回 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇集合，然後選取屬性集合。
3. 選擇刪除並確認刪除。

篩選的搜尋

您可以使用篩選條件來精簡語意搜尋結果。若要建立索引並對文件執行篩選搜尋，請將上一個教學課程中的上傳和搜尋資料替換為下列指示。其他步驟保持不變。如需篩選條件的詳細資訊，請參閱使用篩選條件進行 k-NN 搜尋。

索引和搜尋房屋集合中的資料

1. 若要為您的集合建立單一索引，請在開發工具主控台中傳送下列請求：

```
PUT housing-index-filtered
{
  "settings": {
    "index.knn": true
  },
  "mappings": {
    "properties": {
      "housing-vector": {
        "type": "knn_vector",
        "dimension": 3,
        "method": {
          "engine": "faiss",
          "name": "hnsw"
        }
      },
      "title": {
        "type": "text"
```

```
      },
      "price": {
        "type": "long"
      },
      "location": {
        "type": "geo_point"
      }
    }
  }
}
```

2. 若要將單一文件索引為已housing-index-filtered，請傳送下列請求：

```
POST housing-index-filtered/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. 若要以指定價格在地理點的指定距離內搜尋西雅圖的公寓，請傳送下列請求：

```
GET housing-index-filtered/_search
{
  "size": 5,
  "query": {
    "knn": {
      "housing-vector": {
        "vector": [
          0.1,
          0.2,
          0.3
        ],
        "k": 5,
        "filter": {
          "bool": {
            "must": [
              {
```

```
                "query_string": {
                  "query": "Find me 2 bedroom apartment in Seattle under $3000 ",
                  "fields": [
                    "title"
                  ]
                }
              },
              {
                "range": {
                  "price": {
                    "lte": 3000
                  }
                }
              },
              {
                "geo_distance": {
                  "distance": "100miles",
                  "location": {
                    "lat": 48,
                    "lon": 121
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}
```

數十億個規模的工作負載

向量搜尋集合支援具有數十億個向量的工作負載。您不需要為擴展目的重新編製索引，因為自動擴展會為您執行此操作。如果您有數百萬個具有大量維度的向量（或更多），且需要超過 200 OCUs，請聯絡 AWS Support 來提高您帳戶的 OpenSearch 運算單位 (OCUs上限。

限制

向量搜尋集合有下列限制：

- 向量搜尋集合不支援 Apache Lucene ANN 引擎。

- 向量搜尋集合僅支援具有 Faiss 的 HNSW 演算法，不支援 IVF 和 IVFQ。
- 向量搜尋集合不支援暖機、統計資料和模型訓練 API 操作。
- 向量搜尋集合不支援內嵌或預存指令碼。
- AWS 管理主控台 向量搜尋集合的 中不提供索引計數資訊。
- 向量搜尋集合上索引的重新整理間隔為 60 秒。

後續步驟

現在您知道如何建立向量搜尋集合和索引資料，建議您嘗試下列一些練習：

- 使用 OpenSearch Python 用戶端來使用向量搜尋集合。請參閱 GitHub 上的本教學課程。
- 使用 OpenSearch Java 用戶端來使用向量搜尋集合。請參閱 GitHub 上的本教學課程。
- 設定 LangChain 以使用 OpenSearch 作為向量存放區。LangChain 是一種開放原始碼架構，用於開發採用語言模型的應用程式。如需詳細資訊，請參閱 LangChain 文件。

搭配 Amazon OpenSearch Serverless 使用資料生命週期政策

Amazon OpenSearch Serverless 中的資料生命週期政策會定義 OpenSearch Serverless 在時間序列集合中保留資料的時間長度。例如，您可以設定政策，在 OpenSearch Serverless 刪除日誌資料之前將其保留 30 天。

您可以在 中的每個時間序列集合中為每個索引設定個別政策 AWS 帳戶。OpenSearch Serverless 會保留文件至少您在政策中指定的持續時間。然後，它會盡力自動刪除文件，通常是在保留期間的 48 小時或 10% 內，以較長者為準。

只有時間序列集合支援資料生命週期政策。搜尋和向量搜尋集合不會。

主題

- 資料生命週期政策
- 所需的許可
- 政策優先順序
- 政策語法
- 建立資料生命週期政策
- 更新資料生命週期政策

- 刪除資料生命週期政策

資料生命週期政策

在資料生命週期政策中，您可以指定一系列規則。資料生命週期政策可讓您管理與符合這些規則的索引或集合相關聯的資料保留期。這些規則定義索引或索引群組中資料的保留期間。每個規則都包含資源類型 (index)、保留期間，以及保留期間適用的資源 （ 索引 ） 清單。

您可以使用下列其中一種格式定義保留期：

- `"MinIndexRetention": "24h"` – OpenSearch Serverless 會保留指定期間內的索引資料，以小時或天為單位。您可以將此期間設定為從 24h到 3650d。
- `"NoMinIndexRetention": true` – OpenSearch Serverless 無限期保留索引資料。

在下列範例政策中，第一個規則會為集合 中的所有索引指定 15 天的保留期間`marketing`。第二個規則指定`finance`集合`log`中以 開頭的所有索引名稱都未設定保留期間，且將無限期保留。

```
{
   "lifeCyclePolicyDetail": {
      "type": "retention",
      "name": "my-policy",
      "policyVersion": "MTY4ODI0NTM2OTk1N18x",
      "policy": {
         "Rules": [
            {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/*"
            ],
            "MinIndexRetention": "15d"
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/finance/log*"
            ],
            "NoMinIndexRetention": true
         }
         ]
      },
      "createdDate": 1688245369957,
```

```
      "lastModifiedDate": 1688245369957
   }
}
```

在下列範例政策規則中，OpenSearch Serverless 會無限期保留帳戶內所有集合之所有索引中的資料。

```
{
   "Rules": [
      {
         "ResourceType": "index",
         "Resource": [
            "index/*/*"
         ]
      }
   ],
   "NoMinIndexRetention": true
}
```

所需的許可

OpenSearch Serverless 的生命週期政策使用以下 AWS Identity and Access Management (IAM) 許可。您可以指定 IAM 條件，將使用者限制為與特定集合和索引相關聯的資料生命週期政策。

- `aoss:CreateLifecyclePolicy` – 建立資料生命週期政策。
- `aoss:ListLifecyclePolicies` – 列出目前帳戶中的所有資料生命週期政策。
- `aoss:BatchGetLifecyclePolicy` – 檢視與帳戶或政策名稱相關聯的資料生命週期政策。
- `aoss:BatchGetEffectiveLifecyclePolicy` – 檢視指定資源的資料生命週期政策 (`index` 是唯一支援的資源）。
- `aoss:UpdateLifecyclePolicy` – 修改指定的資料生命週期政策，並變更其保留設定或資源。
- `aoss:DeleteLifecyclePolicy` – 刪除資料生命週期政策。

下列身分型存取政策可讓使用者檢視所有資料生命週期政策，並使用資源模式 更新政策`index/application-logs`：

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aoss:UpdateLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": "application-logs"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:ListLifecyclePolicies",
                "aoss:BatchGetLifecyclePolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

政策優先順序

在某些情況下，資料生命週期政策規則可能會在政策內或跨政策之間重疊。發生這種情況時，具有更特定資源名稱或索引模式的規則會使用更一般的資源名稱或模式覆寫兩個規則通用的任何索引的規則。

例如，在下列政策中，兩個規則適用於索引 `index/sales/logstash`。在這種情況下，第二個規則優先，因為 `index/sales/log*` 是 最長的相符項目`index/sales/logstash`。因此，OpenSearch Serverless 不會設定索引的保留期間。

```
{
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/sales/*",
            ],
            "MinIndexRetention": "15d"
```

```
        },
        {
           "ResourceType":"index",
           "Resource":[
              "index/sales/log*",
           ],
           "NoMinIndexRetention": true
        }
     ]
  }
```

政策語法

提供一個或多個規則。這些規則會為您的 OpenSearch Serverless 索引定義資料生命週期設定。

每個規則都包含下列元素。您可以在每個規則NoMinIndexRetention中提供MinIndexRetention或 ，但不能同時提供兩者。

Element	Description
Resource Type (資源類型)	套用規則的資源類型。資料生命週期政策唯一支援的選項是 index。
Resource	資源名稱和/或模式的清單。模式由字首和萬用字元 (*) 組成，允許相關聯的許可套用至多個資源。例如 index/*<collection-name|pattern>* /*<index-name|pattern>* 。
MinIndexRetention	將文件保留在索引中的最短期間，以天 (d) 或小時 (h) 為單位。下限為 24h，上限為 3650d。
NoMinIndexRetention	如果為 true，OpenSearch Serverless 會無限期保留文件。

在下列範例中，第一個規則會套用至autoparts-inventory模式 (index/autoparts-inventory/*) 下的所有索引，並要求資料保留至少 20 天，才能進行任何動作，例如刪除或封存。

第二個規則以符合auto*/gear模式 (index/auto*/gear) 的索引為目標，將最短保留期間設定為 24 小時。

第三個規則特別適用於tires索引，沒有最短保留期間，這表示可以立即刪除或封存此索引中的資料，或根據其他條件。這些規則有助於管理具有不同保留時間或沒有保留限制的索引資料的保留。

```
{
  "Rules": [
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/*"
      ],
      "MinIndexRetention": "20d"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/auto*/gear"
      ],
      "MinIndexRetention": "24h"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/tires"
      ],
      "NoMinIndexRetention": true
    }
  ]
}
```

建立資料生命週期政策

若要建立資料生命週期政策，您可以定義根據指定條件管理資料保留和刪除的規則。

主控台

建立資料生命週期政策

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇資料生命週期政策。
3. 選擇建立資料生命週期政策。
4. 輸入政策的描述性名稱。

5. 針對資料生命週期，選擇新增並選取政策的集合和索引。

   首先選擇索引所屬的集合。然後，從清單中選擇索引或輸入索引模式。若要選取所有集合做為來源，請輸入星號 (*)。

6. 對於資料保留，您可以選擇無限期保留資料，或取消選取無限制 （絕不刪除），並指定一段時間後 OpenSearch Serverless 會自動從 Amazon S3 刪除資料。

7. 選擇儲存，然後選擇建立。

AWS CLI

若要使用 建立資料生命週期政策 AWS CLI，請使用 create-lifecycle-policy 命令搭配下列選項：

- `--name` – 政策的名稱。
- `--type` – 政策的類型。目前，唯一可用的值是 `retention`。
- `--policy` – 資料生命週期政策。此參數同時接受內嵌政策和 .json 檔案。您必須將內嵌政策編碼為 JSON 逸出字串。若要在 檔案中提供政策，請使用 格式 `--policy file://my-policy.json`。

Example

```
aws opensearchserverless create-lifecycle-policy \
  --name my-policy \
  --type retention \
  --policy "{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":[\"index/autoparts-
inventory/*\"],\"MinIndexRetention\": \"81d\"},{\"ResourceType\":\"index\",\"Resource
\":[\"index/sales/orders*\"],\"NoMinIndexRetention\":true}]}"
```

更新資料生命週期政策

若要更新資料生命週期政策，您可以修改現有的規則，以反映資料保留或刪除需求的變更。這可讓您隨著資料管理需求的演進調整政策。

從更新政策到 OpenSearch Serverless 開始強制執行新的保留期間，可能會有幾分鐘的延遲時間。

主控台

更新資料生命週期政策

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。

2. 在左側導覽窗格中，選擇資料生命週期政策。
3. 選取您要更新的資料生命週期政策，然後選擇編輯。
4. 使用視覺化編輯器或 JSON 編輯器修改政策。
5. 選擇儲存。

AWS CLI

若要使用 更新資料生命週期政策 AWS CLI，請使用 update-lifecycle-policy 命令。

您必須在請求中包含 `--policy-version` 參數。您可以使用 list-lifecycle-policies 或 batch-get-lifecycle-policy 命令來擷取政策版本。我們建議您包含最新的政策版本，以防止意外覆寫其他人所做的變更。

下列請求會使用新的政策 JSON 文件更新資料生命週期政策。

Example

```
aws opensearchserverless update-lifecycle-policy \
  --name my-policy \
  --type retention \
  --policy-version MTY2MzY5MTY1MDA3Ml8x \
  --policy file://my-new-policy.json
```

刪除資料生命週期政策

當您刪除資料生命週期政策時，OpenSearch Serverless 不會再在任何相符的索引上強制執行它。

主控台

刪除資料生命週期政策

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇資料生命週期政策。
3. 選取您要刪除的政策，然後選擇刪除並確認刪除。

AWS CLI

若要使用 刪除資料生命週期政策 AWS CLI，請使用 delete-lifecycle-policy 命令。

Example

```
aws opensearchserverless delete-lifecycle-policy \
  --name my-policy \
  --type retention
```

使用 AWS SDKs與 Amazon OpenSearch Serverless 互動

本節包含如何使用 AWS SDKs 與 Amazon OpenSearch Serverless 互動的範例。這些程式碼範例顯示如何建立安全政策和集合，以及如何查詢集合。

> **Note**
>
> 我們目前正在建置這些程式碼範例。如果您想貢獻程式碼範例 (Java、Go 等)，請直接在 GitHub 儲存庫中開立提取請求。

主題

- Python
- JavaScript

Python

下列範例指令碼會使用 Python 的 適用於 Python (Boto3) 的 AWS SDK 以及 opensearch-py 用戶端，來建立加密、網路和資料存取政策、建立相符的集合，以及為某些範例資料編製索引。

若要安裝所需的相依性，請執行下列命令：

```
pip install opensearch-py
pip install boto3
pip install botocore
pip install requests-aws4auth
```

在指令碼中，將 `Principal` 元素取代為簽署請求的使用者或角色的 Amazon Resource Name (ARN)。您也可以選擇性地修改 `region`。

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3
```

```
import botocore
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

client = boto3.client('opensearchserverless')
service = 'aoss'
region = 'us-east-1'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def createEncryptionPolicy(client):
    """Creates an encryption policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Encryption policy for TV collections',
            name='tv-policy',
            policy="""
                {
                    \"Rules\":[
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[
                                \"collection\/tv-*\"
                            ]
                        }
                    ],
                    \"AWSOwnedKey\":true
                }
                """,
            type='encryption'
        )
        print('\nEncryption policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] The policy name or rules conflict with an existing
 policy.')
        else:
```

```
            raise error


def createNetworkPolicy(client):
    """Creates a network policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Network policy for TV collections',
            name='tv-policy',
            policy="""
                [{
                    \"Description\":\"Public access for TV collection\",
                    \"Rules\":[
                        {
                            \"ResourceType\":\"dashboard\",
                            \"Resource\":[\"collection\/tv-*\"]
                        },
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[\"collection\/tv-*\"]
                        }
                    ],
                    \"AllowFromPublic\":true
                }]
                """,
            type='network'
        )
        print('\nNetwork policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] A network policy with this name already exists.')
        else:
            raise error


def createAccessPolicy(client):
    """Creates a data access policy that matches all collections beginning with tv-"""
    try:
        response = client.create_access_policy(
            description='Data access policy for TV collections',
            name='tv-policy',
            policy="""
```

```
                [{
                    \"Rules\":[
                        {
                            \"Resource\":[
                                \"index\/tv-*\/*\"
                            ],
                            \"Permission\":[
                                \"aoss:CreateIndex\",
                                \"aoss:DeleteIndex\",
                                \"aoss:UpdateIndex\",
                                \"aoss:DescribeIndex\",
                                \"aoss:ReadDocument\",
                                \"aoss:WriteDocument\"
                            ],
                            \"ResourceType\": \"index\"
                        },
                        {
                            \"Resource\":[
                                \"collection\/tv-*\"
                            ],
                            \"Permission\":[
                                \"aoss:CreateCollectionItems\"
                            ],
                            \"ResourceType\": \"collection\"
                        }
                    ],
                    \"Principal\":[
                        \"arn:aws:iam::123456789012:role\/Admin\"
                    ]
                }]
                """,
            type='data'
        )
        print('\nAccess policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] An access policy with this name already exists.')
        else:
            raise error


def createCollection(client):
```

```
    """Creates a collection"""
    try:
        response = client.create_collection(
            name='tv-sitcoms',
            type='SEARCH'
        )
        return(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] A collection with this name already exists. Try
 another name.')
        else:
            raise error


def waitForCollectionCreation(client):
    """Waits for the collection to become active"""
    response = client.batch_get_collection(
        names=['tv-sitcoms'])
    # Periodically check collection status
    while (response['collectionDetails'][0]['status']) == 'CREATING':
        print('Creating collection...')
        time.sleep(30)
        response = client.batch_get_collection(
            names=['tv-sitcoms'])
    print('\nCollection successfully created:')
    print(response["collectionDetails"])
    # Extract the collection endpoint from the response
    host = (response['collectionDetails'][0]['collectionEndpoint'])
    final_host = host.replace("https://", "")
    indexData(final_host)


def indexData(host):
    """Create an index and add some sample data"""
    # Build the OpenSearch client
    client = OpenSearch(
        hosts=[{'host': host, 'port': 443}],
        http_auth=awsauth,
        use_ssl=True,
        verify_certs=True,
        connection_class=RequestsHttpConnection,
        timeout=300
```

```
    )
    # It can take up to a minute for data access rules to be enforced
    time.sleep(45)

    # Create index
    response = client.indices.create('sitcoms-eighties')
    print('\nCreating index:')
    print(response)

    # Add a document to the index.
    response = client.index(
        index='sitcoms-eighties',
        body={
            'title': 'Seinfeld',
            'creator': 'Larry David',
            'year': 1989
        },
        id='1',
    )
    print('\nDocument added:')
    print(response)


def main():
    createEncryptionPolicy(client)
    createNetworkPolicy(client)
    createAccessPolicy(client)
    createCollection(client)
    waitForCollectionCreation(client)


if __name__ == "__main__":
    main()
```

JavaScript

下列範例指令碼會使用 JavaScript 的適用於 Node.js 中 JavaScript 的 SDK 以及 opensearch-js 用戶端，來建立加密、網路和資料存取政策、建立相符的集合，以及為某些範例資料編製索引。

若要安裝所需的相依性，請執行下列命令：

```
npm i aws-sdk
npm i aws4
```

```
npm i @opensearch-project/opensearch
```

在指令碼中，將 `Principal` 元素取代為簽署請求的使用者或角色的 Amazon Resource Name (ARN)。您也可以選擇性地修改 `region`。

```
var AWS = require('aws-sdk');
var aws4 = require('aws4');
var {
    Client,
    Connection
} = require("@opensearch-project/opensearch");
var {
    OpenSearchServerlessClient,
    CreateSecurityPolicyCommand,
    CreateAccessPolicyCommand,
    CreateCollectionCommand,
    BatchGetCollectionCommand
} = require("@aws-sdk/client-opensearchserverless");
var client = new OpenSearchServerlessClient();

async function execute() {
    await createEncryptionPolicy(client)
    await createNetworkPolicy(client)
    await createAccessPolicy(client)
    await createCollection(client)
    await waitForCollectionCreation(client)
}

async function createEncryptionPolicy(client) {
    // Creates an encryption policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
            description: 'Encryption policy for TV collections',
            name: 'tv-policy',
            type: 'encryption',
            policy: " \
        { \
            \"Rules\":[ \
                { \
                    \"ResourceType\":\"collection\", \
                    \"Resource\":[ \
                        \"collection\/tv-*\" \
                    ] \
```

```
                } \
            ], \
            \"AWSOwnedKey\":true \
        }"
        });
        const response = await client.send(command);
        console.log("Encryption policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] The policy name or rules conflict with an
 existing policy.');
        } else
            console.error(error);
    };
}

async function createNetworkPolicy(client) {
    // Creates a network policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
            description: 'Network policy for TV collections',
            name: 'tv-policy',
            type: 'network',
            policy: " \
            [{ \
                \"Description\":\"Public access for television collection\", \
                \"Rules\":[ \
                    { \
                        \"ResourceType\":\"dashboard\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    }, \
                    { \
                        \"ResourceType\":\"collection\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    } \
                ], \
                \"AllowFromPublic\":true \
            }]"
        });
        const response = await client.send(command);
        console.log("Network policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
```

```
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] A network policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createAccessPolicy(client) {
    // Creates a data access policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateAccessPolicyCommand({
            description: 'Data access policy for TV collections',
            name: 'tv-policy',
            type: 'data',
            policy: " \
            [{ \
                \"Rules\":[ \
                    { \
                        \"Resource\":[ \
                            \"index\/tv-*\/*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateIndex\", \
                            \"aoss:DeleteIndex\", \
                            \"aoss:UpdateIndex\", \
                            \"aoss:DescribeIndex\", \
                            \"aoss:ReadDocument\", \
                            \"aoss:WriteDocument\" \
                        ], \
                        \"ResourceType\": \"index\" \
                    }, \
                    { \
                        \"Resource\":[ \
                            \"collection\/tv-*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateCollectionItems\" \
                        ], \
                        \"ResourceType\": \"collection\" \
                    } \
                ], \
                \"Principal\":[ \
                    \"arn:aws:iam::123456789012:role\/Admin\" \
```

```
                ] \
            }]"
        });
        const response = await client.send(command);
        console.log("Access policy created:");
        console.log(response['accessPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] An access policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createCollection(client) {
    // Creates a collection to hold TV sitcoms indexes
    try {
        var command = new CreateCollectionCommand({
            name: 'tv-sitcoms',
            type: 'SEARCH'
        });
        const response = await client.send(command);
        return (response)
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] A collection with this name already
 exists. Try another name.');
        } else
            console.error(error);
    };
}

async function waitForCollectionCreation(client) {
    // Waits for the collection to become active
    try {
        var command = new BatchGetCollectionCommand({
            names: ['tv-sitcoms']
        });
        var response = await client.send(command);
        while (response.collectionDetails[0]['status'] == 'CREATING') {
            console.log('Creating collection...')
            await sleep(30000) // Wait for 30 seconds, then check the status again
            function sleep(ms) {
```

```
                return new Promise((resolve) => {
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        console.log('Collection successfully created:');
        console.log(response['collectionDetails']);
        // Extract the collection endpoint from the response
        var host = (response.collectionDetails[0]['collectionEndpoint'])
        // Pass collection endpoint to index document request
        indexDocument(host)
    } catch (error) {
        console.error(error);
    };
}

async function indexDocument(host) {

    var client = new Client({
        node: host,
        Connection: class extends Connection {
            buildRequestObject(params) {
                var request = super.buildRequestObject(params)
                request.service = 'aoss';
                request.region = 'us-east-1'; // e.g. us-east-1
                var body = request.body;
                request.body = undefined;
                delete request.headers['content-length'];
                request.headers['x-amz-content-sha256'] = 'UNSIGNED-PAYLOAD';
                request = aws4.sign(request, AWS.config.credentials);
                request.body = body;

                return request
            }
        }
    });

    // Create an index
    try {
        var index_name = "sitcoms-eighties";

        var response = await client.indices.create({
            index: index_name
```

```
        });

        console.log("Creating index:");
        console.log(response.body);

        // Add a document to the index
        var document = "{ \"title\": \"Seinfeld\", \"creator\": \"Larry David\", \"year
\": \"1989\" }\n";

        var response = await client.index({
            index: index_name,
            body: document
        });

        console.log("Adding document:");
        console.log(response.body);
    } catch (error) {
        console.error(error);
    };
}

execute()
```

使用 CloudFormation 建立 Amazon OpenSearch Serverless 集合

您可以使用 CloudFormation 建立 Amazon OpenSearch Serverless 資源，例如集合、安全政策和 VPC 端點。如需全面的 OpenSearch Serverless CloudFormation 參考資料，請參閱《CloudFormation 使用者指南》中的 Amazon OpenSearch Serverless。

下列範例 CloudFormation 範本會建立簡單的資料存取政策、網路政策和安全政策，以及相符的集合。這是使用 Amazon OpenSearch Serverless 快速啟動和執行，並佈建必要元素以建立和使用集合的好方法。

> **Important**
>
> 此範例使用公有網路存取，不建議用於生產工作負載。我們建議您使用 VPC 存取權來保護您的集合。如需詳細資訊，請參閱 AWS::OpenSearchServerless::VpcEndpoint 和 the section called “資料平面 VPC 端點”。

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Description: 'Amazon OpenSearch Serverless template to create an IAM user, encryption
 policy, data access policy and collection'
Resources:
  IAMUSer:
    Type: 'AWS::IAM::User'
    Properties:
      UserName:  aossadmin
  DataAccessPolicy:
    Type: 'AWS::OpenSearchServerless::AccessPolicy'
    Properties:
      Name: quickstart-access-policy
      Type: data
      Description: Access policy for quickstart collection
      Policy: !Sub >-
        [{"Description":"Access for cfn user","Rules":
[{"ResourceType":"index","Resource":["index/*/*"],"Permission":["aoss:*"]},
        {"ResourceType":"collection","Resource":["collection/quickstart"],"Permission":
["aoss:*"]}],
        "Principal":["arn:aws:iam::${AWS::AccountId}:user/aossadmin"]}]
  NetworkPolicy:
    Type: 'AWS::OpenSearchServerless::SecurityPolicy'
    Properties:
      Name: quickstart-network-policy
      Type: network
      Description: Network policy for quickstart collection
      Policy: >-
        [{"Rules":[{"ResourceType":"collection","Resource":["collection/
quickstart"]}, {"ResourceType":"dashboard","Resource":["collection/
quickstart"]}],"AllowFromPublic":true}]
  EncryptionPolicy:
    Type: 'AWS::OpenSearchServerless::SecurityPolicy'
    Properties:
      Name: quickstart-security-policy
      Type: encryption
      Description: Encryption policy for quickstart collection
      Policy: >-
        {"Rules":[{"ResourceType":"collection","Resource":["collection/
quickstart"]}],"AWSOwnedKey":true}
  Collection:
    Type: 'AWS::OpenSearchServerless::Collection'
    Properties:
      Name: quickstart
      Type: TIMESERIES
      Description: Collection to holds timeseries data
```

```
    DependsOn: EncryptionPolicy
Outputs:
  IAMUser:
    Value: !Ref IAMUSer
  DashboardURL:
    Value: !GetAtt Collection.DashboardEndpoint
  CollectionARN:
    Value: !GetAtt Collection.Arn
```

使用快照備份集合

快照是 Amazon OpenSearch Serverless 集合的point-in-time備份，可提供災難復原功能。OpenSearch Serverless 會自動建立和管理集合的快照，以確保業務連續性和資料保護。每個快照都包含索引中繼資料（索引的設定和映射）、叢集中繼資料（索引範本和別名）和索引資料（儲存在索引中的所有文件和資料）。

OpenSearch Serverless 提供無需手動組態的自動每小時備份、零維護開銷、無額外儲存成本、快速復原意外資料遺失，以及能夠從快照還原特定索引。

在使用快照之前，請先了解這些重要的考量事項。建立快照需要一些時間才能完成，而且不會立即完成。快照建立期間的新文件或更新將不會包含在快照中。您只能將快照還原至其原始集合，不能還原至新的集合。還原時，索引會收到與其原始版本不同的新 UUIDs。如果未提供新的索引名稱或字首模式，還原至 OpenSearch Serverless 中現有的開啟索引，將會覆寫該索引的資料。這與 OpenSearch 核心行為不同。您一次只能執行一個還原操作，而且無法同時在相同集合上啟動多個還原操作。在作用中還原操作期間嘗試還原索引會導致操作失敗。在還原操作期間，您對索引的請求會失敗。

所需的許可

若要使用快照，請在資料存取政策中設定下列許可。如需資料存取政策的詳細資訊，請參閱 資料存取政策與 IAM 政策比較。

資料存取政策	API
aoss：DescribeSnapshot	GET /_cat/snapshots/aoss-automated GET _snapshot/aoss-automated/snapshot/
aoss：RestoreSnapshot	POST /_snapshot/aoss-automated/snapshot/_restore

資料存取政策	API
aoss：DescribeCollectionItems	GET /_cat/recovery

您可以使用下列 AWS CLI 命令來設定政策：

1. create-access-policy
2. delete-access-policy
3. get-access-policy
4. update-access-policy

以下是用於建立存取政策的範例 CLI 命令。在 命令中，將##內容取代為您的特定資訊。

```
aws opensearchserverless create-access-policy \
--type data \
--name Example-data-access-policy \
--region aws-region \
--policy '[
  {
    "Rules": [
      {
        "Resource": [
          "collection/Example-collection"
        ],
        "Permission": [
          "aoss:DescribeSnapshot",
          "aoss:RestoreSnapshot",
          "aoss:DescribeCollectionItems"
        ],
        "ResourceType": "collection"
      }
    ],
    "Principal": [
      "arn:aws:iam::111122223333:user/UserName"
    ],
    "Description": "Data policy to support snapshot operations."
  }
]'
```

使用快照

根據預設，當您建立新的集合時，OpenSearch Serverless 會每小時自動建立快照。您無需執行任何動作。每個快照都包含集合中的所有索引。OpenSearch Serverless 建立快照後，您可以使用下列程序列出快照並檢閱快照的詳細資訊。

列出快照

使用下列程序列出集合中的所有快照，並檢閱其詳細資訊。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合。
3. 選擇集合的名稱以開啟其詳細資訊頁面。
4. 選擇快照索引標籤以顯示所有產生的快照。
5. 檢閱快照資訊，包括：
   - 快照 ID - 快照的唯一識別符
   - 狀態 - 目前狀態（可用、進行中）
   - 建立時間 - 拍攝快照的時間

AWS CLI

- 使用下列命令列出集合中的所有快照：

```
GET /_cat/snapshots/aoss-automated
```

OpenSearch Serverless 會傳回類似以下的回應：

```
id                                status  start_epoch start_time end_epoch
 end_time    duration    indexes successful_shards failed_shards total_shards
snapshot-ExampleSnapshotID1     SUCCESS 1737964331  07:52:11   1737964382
 07:53:02    50.4s       1
snapshot-ExampleSnapshotID2     SUCCESS 1737967931  08:52:11   1737967979
 08:52:59    47.7s       2
```

```
snapshot-ExampleSnapshotID3     SUCCESS 1737971531  09:52:11   1737971581
 09:53:01    49.1s       3
snapshot-ExampleSnapshotID4 IN_PROGRESS 1737975131  10:52:11   -          -
       4.8d       3
```

取得快照詳細資訊

使用下列程序擷取特定快照的詳細資訊。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合。
3. 選擇集合的名稱以開啟其詳細資訊頁面。
4. 選擇 Snapshots (快照) 索引標籤。
5. 選擇快照任務 ID 以顯示快照的詳細資訊，包括中繼資料、包含的索引和時間資訊。

AWS CLI

- 使用下列命令擷取快照的相關資訊。在 命令中，將*##*內容取代為您的特定資訊。

```
GET _snapshot/aoss-automated/snapshot/
```

範例請求：

```
GET _snapshot/aoss-automated/snapshot-ExampleSnapshotID1/
```

回應範例：

```
{
    "snapshots": [
        {
            "snapshot": "snapshot-ExampleSnapshotID1-5e01-4423-9833Example",
            "uuid": "Example-5e01-4423-9833-9e9eb757Example",
            "version_id": 136327827,
            "version": "2.11.0",
            "remote_store_index_shallow_copy": true,
```

```
            "indexes": [
                "Example-index-0117"
            ],
            "data_streams": [],
            "include_global_state": true,
            "metadata": {},
            "state": "SUCCESS",
            "start_time": "2025-01-27T09:52:11.953Z",
            "start_time_in_millis": 1737971531953,
            "end_time": "2025-01-27T09:53:01.062Z",
            "end_time_in_millis": 1737971581062,
            "duration_in_millis": 49109,
            "failures": [],
            "shards": {
                "total": 0,
                "failed": 0,
                "successful": 0
            }
        }
    ]
}
```

快照回應包含數個金鑰欄位：id提供快照操作的唯一識別符、status傳回目前狀態 SUCCESS或 IN_PROGRESS、duration指出完成快照操作所需的時間，以及indexes傳回快照中包含的索引數量。

從快照還原

從快照還原會從先前建立的備份復原資料。此程序對於 OpenSearch Serverless 中的災難復原和資料管理至關重要。在還原之前，請了解還原的索引將具有與其原始版本不同的 UUIDs，如果未提供新的索引名稱或字首模式，還原至 OpenSearch Serverless 中的現有開放索引，將會覆寫該索引的資料，快照只能還原至其原始集合（不支援跨集合還原），而且還原操作會影響叢集效能，因此請進行相應的規劃。

使用下列程序從快照還原備份的索引。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合。

3. 選擇集合的名稱以開啟其詳細資訊頁面。
4. 選擇快照索引標籤以顯示可用的快照。
5. 選擇您要從中還原的快照，然後選擇從快照還原。
6. 在從快照還原對話方塊中：
   - 對於快照名稱，驗證選取的快照 ID。
   - 針對快照範圍，選擇下列其中一項：
     - 集合中的所有索引 - 從快照還原所有索引
     - 特定索引 - 選取要還原的個別索引
   - 針對目的地，選擇要還原的集合。
   - （選用）設定重新命名設定以重新命名還原的索引：
     - 請勿重新命名 - 保留原始索引名稱
     - 新增字首至還原的索引名稱 - 新增字首以避免衝突
     - 使用規則表達式重新命名 - 使用進階重新命名模式
   - （選用）設定通知設定，以便在還原完成或發生錯誤時收到通知。
7. 選擇儲存以開始還原操作。

OpenSearch API

1. 執行下列命令以識別適當的快照。

```
GET /_snapshot/aoss-automated/_all
```

   如需較小的快照清單，請執行下列命令。

```
GET /_cat/snapshots/aoss-automated
```

2. 執行下列命令，在還原之前驗證快照的詳細資訊。在 命令中，將*##*內容取代為您的特定資訊。

```
GET _snapshot/aoss-automated/snapshot-ExampleSnapshotID1/
```

3. 執行下列命令，從特定快照還原 。

```
POST /_snapshot/aoss-automated/snapshot-ID/_restore
```

您可以透過包含請求內文來自訂還原操作。範例如下。

```
POST /_snapshot/aoss-automated/snapshot-
ExampleSnapshotID1-5e01-4423-9833Example/_restore
{
  "indices": "opensearch-dashboards*,my-index*",
  "ignore_unavailable": true,
  "include_global_state": false,
  "include_aliases": false,
  "rename_pattern": "opensearch-dashboards(.+)",
  "rename_replacement": "restored-opensearch-dashboards$1"
}
```

4. 執行下列命令以檢視還原進度。

```
GET /_cat/recovery
```

> **Note**
>
> 使用包含請求內文的命令還原快照時，您可以使用多個參數來控制還原行為。indices 參數指定要還原的索引，並支援萬用字元模式。設定 ignore_unavailable以繼續還原操作，即使快照中的索引遺失。使用 include_global_state 決定是否還原叢集狀態，以及include_aliases控制是否還原相關聯的別名。rename_pattern 和 rename_replacement 參數會在還原操作期間重新命名索引。

Amazon OpenSearch Serverless 中的 Zstandard Codec 支援

索引轉碼器會決定如何壓縮索引的儲存欄位，並將其儲存在磁碟和 S3 中。索引轉碼器是由指定壓縮演算法的靜態index.codec設定所控制。此設定會影響索引碎片大小和索引操作效能。

根據預設，OpenSearch Serverless 中的索引會使用預設轉碼器搭配 LZ4 壓縮演算法。OpenSearch Serverless 也支援 zstd和 zstd_no_dict轉碼器，具有從 1 到 6 的可設定壓縮層級。

> **Important**
>
> 由於 index.codec 是靜態設定，因此無法在建立索引後變更。

如需詳細資訊，請參閱 OpenSearch Index Codecs 文件。

使用 ZSTD 轉碼器建立索引

您可以使用 `index.codec`設定，在建立索引期間指定 ZSTD 轉碼器：

```
PUT /your_index
{
  "settings": {
    "index.codec": "zstd"
  }
}
```

壓縮層級

ZSTD 轉碼器透過 `index.codec.compression_level`設定支援選用的壓縮層級，接受範圍 【1，6】 中的整數。壓縮層級越高，壓縮比越好 （儲存體越小），但壓縮和解壓縮速度越慢。預設壓縮級別為 3。

```
PUT /your_index
{
  "settings": {
    "index.codec": "zstd",
    "index.codec.compression_level": 2
  }
}
```

效能基準測試

根據使用 nyc_taxi 資料集的基準測試，相較於基準，ZSTD 壓縮在 `zstd`、 `zstd_no_dict`和 壓縮層級的不同組合之間達到 26-32% 更佳的壓縮。

指標	ZSTD L1	ZSTD L6	ZSTD_NO_DICT L1	ZSTD_NO_DICT L6
索引大小縮減	28.10%	32%	26.90%	28.70%
編製輸送量變更的索引	-0.50%	-23.80%	-0.50%	-5.30%
全部符合查詢 p90 延遲改善	-16.40%	29.50%	-16.40%	23.40%

指標	ZSTD L1	ZSTD L6	ZSTD_NO_DICT L1	ZSTD_NO_DICT L6
範圍查詢 p90 延遲改善	90.90%	92.40%	-282.90%	92.50%
距離量 p90 Agg 延遲改善	2%	24.70%	2%	13.80%

如需詳細資訊，請參閱 AWS OpenSearch 部落格。

使用衍生來源儲存儲存體

根據預設，OpenSearch Serverless 會將每個擷取的文件存放在 _source欄位中，其中包含原始 JSON 文件內文，並為個別欄位編製索引以供搜尋。雖然 _source 欄位無法搜尋，但會保留，以便在執行擷取請求時傳回完整文件，例如取得和搜尋。啟用衍生來源時，OpenSearch Serverless 會略過儲存_source欄位，而是隨需動態重建欄位，例如在搜尋、get、mget、reindex 或更新操作期間。使用衍生來源設定可將儲存體用量減少高達 50%。

Configuration

若要設定索引的衍生來源，請使用 index.derived_source.enabled設定建立索引：

```
PUT my-index1
{
  "settings": {
    "index": {
      "derived_source": {
        "enabled": true
      }
    }
  }
}
```

重要考量

- 僅支援特定欄位類型。如需支援的欄位和限制清單，請參閱 OpenSearch 文件。如果您使用衍生來源和不支援的欄位建立索引，則索引建立將會失敗。如果您嘗試在已啟用來源的衍生索引中擷取具有不支援欄位的文件，則擷取將會失敗。只有在您知道要新增至索引的欄位類型時，才使用此功能。
- 設定index.derived_source.enabled是靜態設定。這無法在建立索引之後變更。

查詢回應的限制

啟用衍生來源時，它會對查詢回應的產生和傳回方式施加特定限制。

- 指定多種格式的日期欄位一律會針對所有請求的文件使用清單中的第一個格式，無論原始擷取格式為何。
- Geopoint 值會以固定`{"lat": lat_val, "lon": lon_val}`格式傳回，並可能會失去一些精確度。
- 多值陣列可以排序，關鍵字欄位可以重複刪除。

如需詳細資訊，請參閱 OpenSearch 部落格。

效能基準測試

根據使用 nyc_taxi 資料集的基準測試，衍生來源的索引大小相較於基準減少了 58%。

指標	衍生來源
索引大小縮減	58.3%
編製輸送量變更的索引	3.7%
索引 p90 延遲變更	6.9%
全部符合查詢 p90 延遲改善	19%
範圍查詢 p90 延遲改善	-18.8%
距離量 p90 Agg 延遲改善	-7.3%

如需詳細資訊，請參閱 OpenSearch 部落格。

Amazon OpenSearch Serverless 集合群組

Amazon OpenSearch Serverless 中的集合群組會組織多個集合，並啟用使用不同 KMS 金鑰跨集合的運算資源共用。此共用運算模型透過消除每個 KMS 金鑰對個別 OpenSearch 運算單位 (OCUs) 的需求來降低成本。

您建立的每個 OpenSearch Serverless 集合都會受到靜態資料的加密保護，使用 AWS KMS 來存放和管理加密金鑰。相同集合群組中的集合會共用運算資源和 OCU 記憶體空間，即使它們使用不同的 KMS 金鑰進行加密也一樣。

集合群組可隔離安全與效能需求。您可以將具有相同 KMS 金鑰的集合分組為單一集合群組以進行安全隔離，或將集合與相同群組中的不同 KMS 金鑰結合以最佳化成本。此彈性可讓您平衡安全需求與資源效率。

當您將集合新增至集合群組時，OpenSearch Serverless 會將集合指派給群組的共用運算資源。系統會自動管理跨這些資源的工作負載分佈，同時透過使用其指定的 KMS 金鑰加密每個集合的資料來維護安全性。存取控制會持續在集合層級套用，而共用的運算資源會視需要存取多個 KMS 金鑰，以為群組中的集合提供服務。

您可以透過在集合群組層級設定最小和最大 OCU 限制來控制資源配置。這些限制適用於 群組中的所有集合，可協助您管理成本，同時確保一致的效能。

根據預設，集合群組中的集合數量、集合中的索引數量，以及集合群組中的 OCUs 數量都有服務配額（限制）。如需詳細資訊，請參閱 OpenSearch Serverless 配額。

重要概念

集合群組

做為一或多個集合容器 AWS 的資源。每個集合群組都有唯一的識別符，並且可以有自己的容量限制和組態設定。

共用運算

相同集合群組中的集合共用相同的一組 OCUs，無論其使用的 KMS 金鑰為何。此共用可免除每個 KMS 金鑰對個別運算資源的需求，進而降低成本。

容量限制

您可以在集合群組層級為索引和搜尋操作設定最小和最大 OCU 限制。這些限制有助於控制成本並確保一致的效能。

集合群組容量限制

集合群組透過最小和最大 OCU 限制，提供對資源配置的精細控制。這些限制適用於群組中的所有集合，並與帳戶層級容量設定獨立運作。

根據預設，集合群組中的集合數量、集合中的索引數量，以及集合群組中的 OCUs 數量都有服務配額（限制）。如需詳細資訊，請參閱 OpenSearch Serverless 配額。

了解集合群組容量限制

您可以在集合群組層級設定索引和搜尋操作的最小和最大 OCU 限制。這些限制控制 OpenSearch Serverless 如何擴展群組中集合的資源：

- 最小 OCU – OpenSearch Serverless 為集合群組維護的最小 OCUs 數量，以確保一致的基準效能。
  - 如果工作負載需要的 OCU 少於指定的最小值，OpenSearch Serverless 仍會維持 OCU 的指定最小值，而帳單也會反映相同的值。
  - 如果工作負載需要的 OCU 數量高於指定的最小值，OpenSearch Serverless 會維持工作負載所需的 OCU 層級，而且帳單會反映較高的 OCU 使用率。
- 最大 OCU – OpenSearch Serverless 可為集合群組擴展至 的 OCUs 數目上限，協助您控制成本。

集合群組容量限制會與帳戶層級限制分離。帳戶層級的最大 OCU 設定僅適用於未與任何集合群組相關聯的集合，而集合群組的最大 OCU 設定則適用於該特定群組中的集合。

有效的容量限制值

設定集合群組的最小和最大 OCU 限制時，您只能使用下列集合中的值：1、2、4、8、16 和 16 的倍數（例如 32、48、64、80、96），最多 1，696 OCUs。

當您建立集合群組時，最小和最大 OCU 限制都是選用的。如果您未指定最大 OCU 限制，OpenSearch Serverless 會使用預設值 96 個 OCUs。

最小 OCU 限制必須小於或等於最大 OCU 限制。

了解帳戶層級和集合群組 OCU 限制之間的關係

規劃 OpenSearch Serverless 容量時，請務必了解帳戶層級的 OCU 限制和集合群組 OCU 限制如何互動。所有集合群組的最大 OCU 設定加上帳戶層級的最大 OCU 設定總和必須小於或等於每個帳戶的服務配額限制。如需目前限制值，請參閱 OpenSearch Serverless 配額。

> **Note**
>
> 帳戶層級的最大 OCU 設定僅適用於與任何集合群組無關的集合。集合群組內的集合受其各自集合群組限制管理，而非帳戶層級限制。

此限制同時適用於索引和搜尋 OCUs。例如，如果您設定帳戶層級設定和集合群組，您必須確保總計不超過索引 OCUs 的服務配額限制，並且單獨不超過搜尋 OCUs 的服務配額限制。此外，每個帳戶最多可以建立 300 個集合群組。

範例：規劃具有帳戶層級和集合群組限制的容量

如果您將帳戶層級的最大搜尋 OCU 設定為 500，且服務配額限制為 1，700：

- 並建立 2 個集合群組，2 個集合群組的最大 OCU 總和不得超過 1，200 (1，700 - 500)
- 您可以讓每個集合群組的預設最大 OCU 為 96 (96 + 96 + 500 = 692)，為未來的成長騰出空間
- 或者，您可以使用服務配額允許的完整容量，將每個集合群組的上限為 600 (600 + 600 + 500 = 1，700)

此關係對於容量規劃至關重要。在建立新的集合群組或增加最大 OCU 限制之前，請確認您的總配置不超過服務配額限制。如果您達到此限制，則必須減少現有集合群組的最大 OCU 設定，或減少帳戶層級的最大 OCU 設定，以騰出空間進行新的配置。

設定容量限制

您可以在建立集合群組或稍後更新集合群組時設定容量限制。若要使用 設定容量限制 AWS CLI，請使用 CreateCollectionGroup 或 UpdateCollectionGroup 命令：

```
aws opensearchserverless create-collection-group \
    --name my-collection-group \
    --capacity-limits
 maxIndexingCapacityInOCU=32,maxSearchCapacityInOCU=32,minIndexingCapacityInOCU=4,minSearchCapa
```

若要更新現有集合群組的容量限制：

```
aws opensearchserverless update-collection-group \
    --id abcdef123456 \
    --capacity-limits
 maxIndexingCapacityInOCU=48,maxSearchCapacityInOCU=48,minIndexingCapacityInOCU=8,minSearchCapa
```

監控集合群組容量

OpenSearch Serverless 每隔一分鐘發出下列 Amazon CloudWatch Logs 指標，以協助您監控集合群組層級的 OCU 使用率和容量限制：

- `IndexingOCU` – 集合群組目前正在使用的索引 OCUs 數量。
- `SearchOCU` – 集合群組目前正在使用的搜尋 OCUs 數目。

OpenSearch Serverless 也會針對與任何集合群組無關的集合，在帳戶層級發出 OCU 指標。您可以在 CloudWatch 中彙總這些指標，以視覺化方式呈現所有集合群組和帳戶層級集合的 OCUs 總和。

設定警示以在集合群組接近其容量限制時通知您，以便您可以視需要調整設定。如需 OpenSearch Serverless 指標的詳細資訊，請參閱 the section called “監控 OpenSearch Serverless”。

如何強制執行容量限制

OpenSearch Serverless 會在擴展操作期間強制執行集合群組容量限制。當您的集合需要額外資源時，OpenSearch Serverless 會擴展至最大 OCU 限制。當需求減少時，OpenSearch Serverless 會縮減規模，但至少維持最小 OCU 限制，以確保一致的效能。

只有當集合群組包含至少一個集合時，才會強制執行容量限制。空的集合群組不會使用 OCUs 或強制執行容量限制。

如果擴展操作超過最大 OCU 限制或違反最小 OCU 要求，OpenSearch Serverless 會拒絕該操作，以維持符合您設定的限制。

集合群組中的加密和 KMS 金鑰

您建立的每個 OpenSearch Serverless 集合都受到靜態資料的加密保護 AWS KMS ，使用 存放和管理加密金鑰。使用集合群組時，您可以靈活地指定集合的 KMS 金鑰。

您可以透過兩種方式提供與集合相關聯的 KMS 金鑰：

- 在 CreateCollection 請求中 – 使用 `encryption-config` 參數建立集合時，直接指定 KMS 金鑰。
- 在安全政策中 – 在加密安全政策中定義 KMS 金鑰關聯。

當您在這兩個位置指定 KMS 金鑰時，CreateCollection 請求中提供的 KMS 金鑰優先於安全政策組態。

這種靈活性可簡化大規模管理集合，尤其是當您需要使用唯一的 KMS 金鑰建立多個集合時。您可以直接在集合建立期間指定 KMS 金鑰，而不是建立和管理數千個加密政策。

在不同的 KMS 金鑰之間共用 OCUs

集合群組可使用不同的 KMS 金鑰跨集合啟用運算資源共用。相同集合群組中的集合會共用 OCU 記憶體空間，無論其加密金鑰為何。此共用運算模型透過消除每個 KMS 金鑰個別 OCUs的需求來降低成本。

集合群組可隔離安全與效能需求。您可以將具有相同 KMS 金鑰的集合分組為單一集合群組以進行安全隔離，或將集合與相同群組中的不同 KMS 金鑰結合以最佳化成本。這種靈活性可讓您平衡安全需求和資源效率。

系統會使用其指定的 KMS 金鑰加密每個集合的資料，以維護安全性。存取控制會持續在集合層級套用，而共用的運算資源會視需要存取多個 KMS 金鑰，以為群組中的集合提供服務。

必要的 KMS 許可

當您在 CreateCollection 請求中指定 KMS 金鑰時，您需要下列其他許可：

- `kms:DescribeKey` – 允許 OpenSearch Serverless 擷取 KMS 金鑰的相關資訊。
- `kms:CreateGrant` – 允許 OpenSearch Serverless 建立 KMS 金鑰的授予，以啟用加密操作。

使用 AWS 擁有的金鑰時，不需要這些許可。

建立集合群組

本主題說明如何在 Amazon OpenSearch Serverless 中建立、設定和管理集合群組。使用集合群組來組織集合並共用運算資源，以最佳化成本。在集合群組層級設定最小和最大 OCU 限制，以控制效能和支出。

建立集合群組

使用下列程序建立新的集合群組並設定其設定。使用 OpenSearch Serverless 主控台 AWS CLI或 AWS SDKs建立集合群組。當您建立集合群組時，您可以指定容量限制和其他組態選項。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合群組
3. 選擇建立集合群組。

4. 針對集合群組名稱，輸入集合群組的名稱。名稱長度必須為 3-32 個字元，以小寫字母開頭，且僅包含小寫字母、數字和連字號。
5. （選用）針對描述，輸入集合群組的描述。
6. 在容量管理區段中，設定 OCU 限制：
   - 最大索引容量 – 此群組中集合可以擴展的索引 OCUs 數量上限。
   - 最大搜尋容量 – 此群組中集合可以擴展的搜尋 OCUs 數量上限。
   - 最低索引容量 – 要維護的索引 OCUs數量下限，以實現一致的效能。
   - 搜尋容量下限 – 要維護的搜尋 OCUs 數量下限，以保持一致的效能。
7. （選用）在標籤區段中，新增標籤以協助組織和識別您的集合群組。
8. 選擇建立集合群組。

AWS CLI

- 使用 create-collection-group 命令建立新的集合群組。在 命令中，將*##*內容取代為您自己的特定資訊。

```
aws opensearchserverless create-collection-group \
    --name my-collection-group \
    --description "Collection group for production workloads" \
    --capacity-limits
 maxIndexingCapacityInOCU=20,maxSearchCapacityInOCU=20,minIndexingCapacityInOCU=2,minSea
 \
    --tags key=Environment,value=Production key=Team,value=DataEngineering
```

命令會傳回所建立集合群組的詳細資訊，包括其唯一 ID 和 ARN。

將新集合新增至集合群組

建立新的集合時，請指定要與集合建立關聯的現有集合群組名稱。使用下列程序將新集合新增至集合群組。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合
3. 選擇 Create collection (建立集合)。
4. 針對集合名稱，輸入集合的名稱。名稱長度必須為 3-28 個字元，以小寫字母開頭，且僅包含小寫字母、數字和連字號。
5. （選用）針對描述，輸入集合的描述。
6. 在集合群組區段中，選取您要指派集合的集合群組。集合一次只能屬於一個集合群組。

   （選用）您也可以選擇建立新群組。這會將您導覽至建立集合群組工作流程。建立集合群組完成後，請返回此程序的步驟 1，開始建立新的集合。
7. 繼續進行工作流程以建立集合。

> **Important**
>
> 請勿離開建立集合工作流程。這樣做會停止集合設定。集合詳細資訊頁面會在設定完成後顯示。

AWS CLI

- 使用 create-collection 命令建立新的集合，並將其新增至現有的集合群組。在 命令中，將##內容取代為您自己的特定資訊。

```
aws opensearchserverless create-collection \
    --name my-collection \
    --type SEARCH \
    --collection-group-name my-collection-group \
    --description "Collection for search workloads"
```

管理 Amazon OpenSearch Serverless 集合群組

建立 Amazon OpenSearch Serverless 集合群組之後，您可以在需求變更時修改其設定。使用這些管理操作來更新容量限制和檢視集合群組詳細資訊。這些變更可協助您最佳化資源配置，並維持集合的高效率組織。

檢視集合群組

顯示您的 OpenSearch Serverless 集合群組，以檢閱其組態、相關聯的集合和目前狀態。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合
3. 選擇集合群組索引標籤。系統會顯示您帳戶的集合群組。
4. 選擇集合群組的名稱以顯示其詳細資訊。

AWS CLI

- 使用 list-collection-groups 命令列出您帳戶中的所有集合群組。使用 batch-get-collection-group 命令來檢視特定集合群組的詳細資訊。在下列命令中，將##內容取代為您自己的特定資訊。

  若要列出所有集合群組：

  ```
  aws opensearchserverless list-collection-groups
  ```

  若要取得特定集合群組的詳細資訊：

  ```
  aws opensearchserverless batch-get-collection-group \
      --names my-collection-group another-group
  ```

更新集合群組設定

更新您的 OpenSearch Serverless 集合群組設定，以修改容量限制和描述等組態。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合
3. 選擇集合群組索引標籤。系統會顯示您帳戶的集合群組。
4. 選擇集合群組的名稱以顯示其詳細資訊。
5. 在集合群組詳細資訊中，選擇編輯。
6. 進行任何變更，然後選擇儲存。

AWS CLI

- 使用 update-collection-group 命令更新現有集合群組的描述和容量限制。在下列命令中，將*##*內容取代為您擁有的資訊。

```
aws opensearchserverless update-collection-group \
    --id abcdef123456 \
    --description "Updated description for production workloads" \
    --capacity-limits
 maxIndexingCapacityInOCU=30,maxSearchCapacityInOCU=30,minIndexingCapacityInOCU=4,minSea
```

容量限制的變更會立即生效，並可能影響群組中集合的擴展行為。

刪除集合群組

您必須先從群組中移除所有集合，才能刪除集合群組。您無法刪除包含集合的集合群組。

Console

1. 開啟位於 https：//https://console.aws.amazon.com/aos/ 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇無伺服器，然後選擇集合

3. 選擇集合群組索引標籤。系統會顯示您帳戶的集合群組。
4. 選擇您要刪除的集合群組名稱。

> **Important**
>
> 透過更新每個集合以移除集合群組關聯，或將集合移至其他集合群組，從集合群組移除所有集合。

5. 在頁面頂端，選擇 Delete (刪除)。
6. 確認刪除，然後選擇刪除。

AWS CLI

- 使用 delete-collection-group 命令來刪除集合群組。

> **Important**
>
> 透過更新每個集合以移除集合群組關聯，或將集合移至其他集合群組，從集合群組移除所有集合。

在下列命令中，將##內容取代為您擁有的資訊。

刪除空的集合群組：

```
aws opensearchserverless delete-collection-group \
    --id abcdef123456
```

管理 Amazon OpenSearch Serverless 的容量限制

透過 Amazon OpenSearch Serverless，您就不需自行管理容量。OpenSearch Serverless 會根據目前的工作負載自動擴展帳戶的運算容量。Serverless 運算容量是以 OpenSearch 運算單元 (OCU) 來測量。每個 OCU 都是 6 GiB 記憶體和對應虛擬 CPU (vCPU) 的組合，並會建立 Amazon S3 的資料管道。如需有關 OpenSearch Serverless 中分開的架構的詳細資訊，請參閱the section called “運作方式”。

當您建立第一個集合時，OpenSearch Serverless 會根據備援設定執行個體化 OCUs。預設會啟用備援作用中複本，這會執行個體化四個 OCUs（兩個用於索引，兩個用於搜尋）。這可確保另一個可用區域中待命節點的高可用性。

對於開發和測試，您可以停用集合的啟用備援設定。這會移除待命複本，並僅使用兩個 OCUs（一個用於索引，另一個用於搜尋）。

這些 OCU 會永遠存在，即使沒有索引編製或搜尋活動也一樣。所有後續集合都可以共用這些 OCUs，但具有唯一 AWS KMS 金鑰的集合除外，這會執行個體化自己的一組 OCUs。與集合群組相關聯的所有集合可以共用相同的一組 OCUs。單一集合群組中只能包含一種集合類型（搜尋、時間序列或向量搜尋）。如需詳細資訊，請參閱the section called “集合群組”。

OpenSearch Serverless 會自動向外擴展，並在索引和搜尋用量增加時新增 OCUs。當流量減少時，容量會縮減至資料大小所需的最小 OCUs 數量。

對於搜尋和時間序列集合，閒置時所需的 OCUs 數量與資料大小和索引計數成正比。對於向量集合，OCU 需求依賴記憶體 (RAM) 來存放向量圖形，以及存放索引的磁碟空間。非閒置時，OCU 需求會考慮這兩個因素。

向量集合會將索引資料儲存在 OCU 本機儲存中。達到 OCU RAM 限制的速度快於磁碟限制，這會依 RAM 空間限制向量集合。

啟用備援後，OCU 容量會縮減至至少 1 個用於索引的 OCU (0.5 個 OCU x 2)，以及 1 個用於搜尋的 OCU (0.5 個 OCU x 2)。當您停用備援時，您的集合可以縮減至 0.5 OCU 進行索引，而 0.5 OCU 進行搜尋。

擴展也會考慮集合或索引所需的碎片數量。每個 OCU 都支援指定的碎片數量，索引數量應與碎片計數成比例。所需的基本 OCUs 總數是資料、記憶體和碎片需求的上限。如需詳細資訊，請參閱AWS 大數據部落格上任何規模的 Amazon OpenSearch Serverless 經濟實惠搜尋功能。

對於搜尋和向量搜尋集合，所有資料都存放在熱索引上，以確保快速的查詢回應時間。時間序列集合使用熱儲存和暖儲存的組合，將最新的資料保留在熱儲存中，以最佳化更頻繁存取資料的查詢回應時間。如需詳細資訊，請參閱the section called “選擇集合類型”。

> **Note**
>
> 向量搜尋集合無法與搜尋和時間序列集合共用 OCUs，即使向量搜尋集合使用與搜尋或時間序列集合相同的 KMS 金鑰。將針對您的第一個向量集合建立新的一組 OCUs。向量集合的 OCUs 會在相同的 KMS 金鑰集合之間共用。

若要管理集合的容量並控制成本，您可以指定目前帳戶和區域的整體索引編制和搜尋容量上限，而 OpenSearch Serverless 會根據這些規格自動橫向擴展集合資源。

索引編製和搜尋容量會分別調整，因此您可以針對每個容量指定帳戶層級限制：

- 索引編製容量上限：OpenSearch Serverless 可將索引編製容量提高到這個數目的 OCU。
- 搜尋容量上限：OpenSearch Serverless 可將搜尋容量提高到這個數目的 OCU。

> **Note**
>
> 目前，容量設定僅適用於帳戶層級。您無法設定每個集合的容量限制。

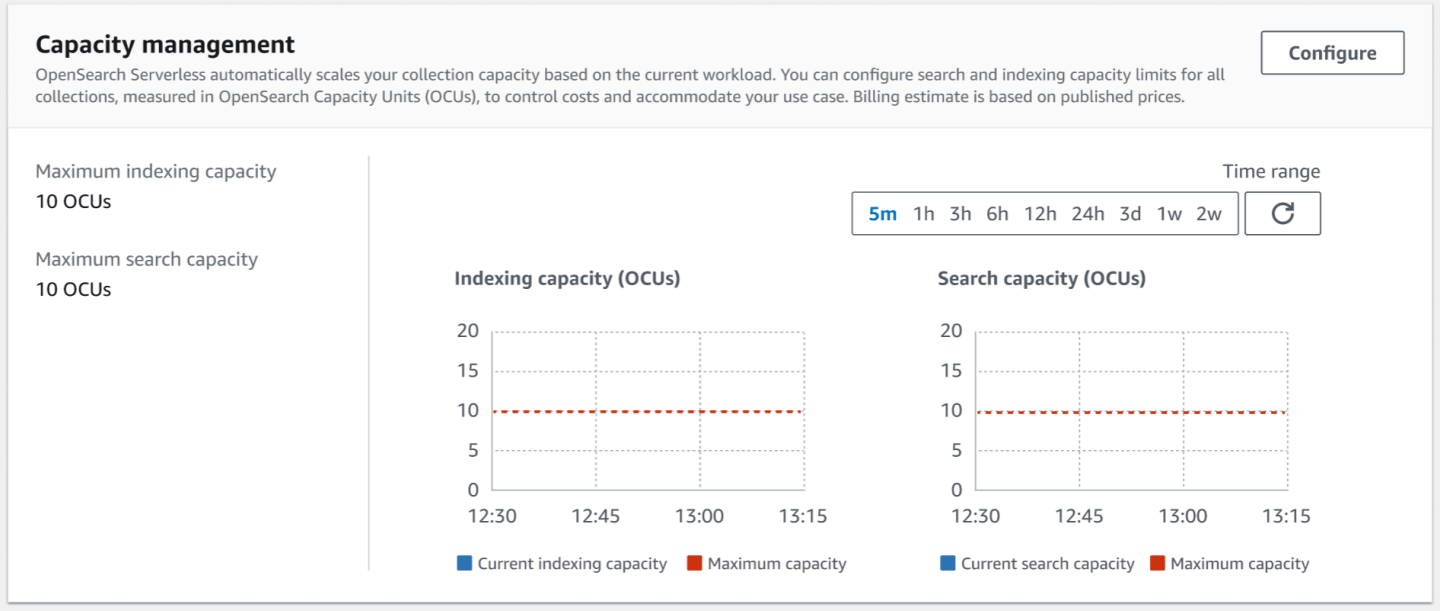
您的目標應該是確保容量上限足以處理尖峰工作負載。OpenSearch Serverless 會根據設定，自動橫向擴展集合的 OCU 數目，以處理索引編制和搜尋工作負載。

主題

- 進行容量設定
- 容量限制上限
- 監控容量用量

進行容量設定

若要在 OpenSearch Serverless 主控台中進行容量設定，請在左側導覽窗格中，展開 Serverless (無伺服器)，然後選取 Dashboard (儀表板)。在 Capacity management (容量管理) 下指定索引編製和搜尋容量上限：



若要使用 設定容量 AWS CLI，請傳送 UpdateAccountSettings 請求：

```
aws opensearchserverless update-account-settings \
    --capacity-limits '{ "maxIndexingCapacityInOCU": 8,"maxSearchCapacityInOCU": 9 }'
```

容量限制上限

集合可包含的索引總數上限為 1000。對於這三種類型的集合，預設的 OCU 容量上限為 10 個用於索引 OCUs，以及 10 個用於搜尋OCUs。帳戶允許的最小 OCU 容量為 1 個 OCU 【0.5 個 OCU x 2】 用於索引，1 個 OCU 【0.5 個 OCU x 2】 用於搜尋。對於所有集合，允許的容量上限為 1，700 個用於索引OCUs，以及 1，700 個用於搜尋OCUs。您可以將 OCU 計數設定為從 2 到最大允許容量的任何數字，以 2 的倍數表示。

每個 OCU 都包含足夠的暫時性熱儲存，可容納 120 GiB 的索引資料。OpenSearch Serverless 支援搜尋和向量搜尋集合中每個索引最多 1 TiB 的資料，以及時間序列集合中每個索引最多 100 TiB 的熱資料。對於時間序列集合，您仍然可以擷取更多資料，這些資料可以儲存為 S3 中的暖資料。

如需所有配額的清單，請參閱 OpenSearch Serverless 配額。

監控容量用量

您可以監控 `Search0CU` 和 `Indexing0CU` 帳戶層級 CloudWatch 指標，以了解集合的擴展方式。建議您設定警示，以在帳戶接近與容量相關的指標閾值時通知您，如此您就可相應調整容量設定。

您也可使用這些指標，判斷容量上限設定是否合適，或者是否需要進行調整。分析這些指標，以將精力集中於集合效率的優化。如需有關 OpenSearch Serverless 傳送至 CloudWatch 的指標的詳細資訊，請參閱the section called “監控 OpenSearch Serverless”。

將資料擷取至 Amazon OpenSearch Serverless 集合

這些章節提供有關支援的擷取管道，以將資料擷取至 Amazon OpenSearch Serverless 集合的詳細資訊。同時還涵蓋了一些您可以用於與 OpenSearch API 操作互動的用戶端。您的用戶端應與 OpenSearch 2.x 相容，以便與 OpenSearch Serverless 整合。

主題

- 所需的最低許可
- OpenSearch 擷取
- Fluent Bit
- Amazon Data Firehose
- Go
- Java
- JavaScript
- Logstash
- Python
- Ruby
- 使用其他用戶端簽署 HTTP 請求

所需的最低許可

若要將資料擷取至 OpenSearch Serverless 集合，寫入資料的主體必須在資料存取政策中指派下列最低許可：

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
```

```
            "index/target-collection/logs"
         ],
         "Permission":[
            "aoss:CreateIndex",
            "aoss:WriteDocument",
            "aoss:UpdateIndex"
         ]
      }
   ],
   "Principal":[
      "arn:aws:iam::123456789012:user/my-user"
   ]
}
]
```

如果您計劃寫入到其他索引，許可的範圍可能會更廣泛。例如，您可以允許對所有索引 (index/*target-collection*/*) 或索引子集合 (index/*target-collection*/*logs**) 的許可，而不是指定單一目標索引。

如需所有可用 OpenSearch API 操作及其相關許可的參考資料，請參閱 the section called “支援的操作和外掛程式”。

OpenSearch 擷取

您可以使用 Amazon OpenSearch Ingestion，而不是使用第三方用戶端直接將資料傳送至 OpenSearch Serverless 集合。 OpenSearch 您可以將資料生產者設定為將資料傳送至 OpenSearch Ingestion，並自動將資料交付至您指定的集合。您也可以設定 OpenSearch Ingestion，在交付資料之前轉換資料。如需詳細資訊，請參閱Amazon OpenSearch Ingestion。

OpenSearch Ingestion 管道需要許可，才能寫入設定為接收器的 OpenSearch Serverless 集合。這些許可包括描述集合並向其傳送 HTTP 請求的能力。如需使用 OpenSearch Ingestion 將資料新增至集合的指示，請參閱 the section called “授予管道對集合的存取權”。

若要開始使用 OpenSearch Ingestion，請參閱 the section called “教學課程：將資料擷取至集合”。

Fluent Bit

您可以使用 AWS for Fluent Bit 映像和 OpenSearch 輸出外掛程式，將資料擷取至 OpenSearch Serverless 集合。

> **Note**
>
> 您必須擁有 AWS 適用於 Fluent Bit 映像的 2.30.0 版或更新版本，才能與 OpenSearch Serverless 整合。

範例組態：

這個組態檔案的範例輸出區段顯示如何使用 OpenSearch Serverless 集合作為目的地。重要的補充是 `AWS_Service_Name` 參數，也就是 `aoss`。`Host` 是集合端點。

```
[OUTPUT]
    Name  opensearch
    Match *
    Host  collection-endpoint.us-west-2.aoss.amazonaws.com
    Port  443
    Index  my_index
    Trace_Error On
    Trace_Output On
    AWS_Auth On
    AWS_Region <region>
    AWS_Service_Name aoss
    tls     On
    Suppress_Type_Name On
```

Amazon Data Firehose

Firehose 支援 OpenSearch Serverless 作為交付目的地。如需將資料傳送至 OpenSearch Serverless 的指示，請參閱《Amazon Data Firehose 開發人員指南》中的建立 Kinesis Data Firehose 交付串流和為您的目的地選擇 OpenSearch Serverless。

您提供給 Firehose 交付的 IAM 角色必須在具有目標集合`aoss:WriteDocument`最低許可的資料存取政策中指定，而且您必須具有預先存在的索引才能傳送資料。如需詳細資訊，請參閱the section called “所需的最低許可”。

將資料傳送至 OpenSearch Serverless 之前，您可能需要執行資料轉換。如需進一步了解如何使用 Lambda 函數來執行此任務，請參閱相同指南中的 Amazon Kinesis Data Firehose 資料轉換。

Go

下列範本程式碼會使用 Go 的 opensearch-go 用戶端，建立與指定 OpenSearch Serverless 集合的安全連線，並建立單一索引。您必須提供 `region` 和 `host` 的值。

```
package main

import (
  "context"
  "log"
  "strings"
  "github.com/aws/aws-sdk-go-v2/aws"
  "github.com/aws/aws-sdk-go-v2/config"
  opensearch "github.com/opensearch-project/opensearch-go/v2"
  opensearchapi "github.com/opensearch-project/opensearch-go/v2/opensearchapi"
  requestsigner "github.com/opensearch-project/opensearch-go/v2/signer/awsv2"
)

const endpoint = "" // serverless collection endpoint

func main() {
 ctx := context.Background()

 awsCfg, err := config.LoadDefaultConfig(ctx,
  config.WithRegion("<AWS_REGION>"),
  config.WithCredentialsProvider(
   getCredentialProvider("<AWS_ACCESS_KEY>", "<AWS_SECRET_ACCESS_KEY>",
 "<AWS_SESSION_TOKEN>"),
  ),
 )
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an AWS request Signer and load AWS configuration using default config folder
 or env vars.
 signer, err := requestsigner.NewSignerWithService(awsCfg, "aoss") // "aoss" for Amazon
 OpenSearch Serverless
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an opensearch client and use the request-signer
```

```
 client, err := opensearch.NewClient(opensearch.Config{
  Addresses: []string{endpoint},
  Signer:    signer,
 })
 if err != nil {
  log.Fatal("client creation err", err)
 }

 indexName := "go-test-index"

  // define index mapping
 mapping := strings.NewReader(`{
  "settings": {
    "index": {
         "number_of_shards": 4
         }
       }
  }`)

 // create an index
 createIndex := opensearchapi.IndicesCreateRequest{
  Index: indexName,
    Body: mapping,
 }
 createIndexResponse, err := createIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("Error ", err.Error())
  log.Println("failed to create index ", err)
  log.Fatal("create response body read err", err)
 }
 log.Println(createIndexResponse)

 // delete the index
 deleteIndex := opensearchapi.IndicesDeleteRequest{
  Index: []string{indexName},
 }

 deleteIndexResponse, err := deleteIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("failed to delete index ", err)
  log.Fatal("delete index response body read err", err)
 }
 log.Println("deleting index", deleteIndexResponse)
}
```

```
func getCredentialProvider(accessKey, secretAccessKey, token string)
 aws.CredentialsProviderFunc {
 return func(ctx context.Context) (aws.Credentials, error) {
  c := &aws.Credentials{
   AccessKeyID:     accessKey,
   SecretAccessKey: secretAccessKey,
   SessionToken:    token,
  }
  return *c, nil
 }
}
```

Java

下列範本程式碼會使用 Java 的 opensearch-java 用戶端，建立與指定 OpenSearch Serverless 集合的安全連線，並建立單一索引。您必須提供 `region` 和 `host` 的值。

與 OpenSearch Service 網域相較，重要的區別在於服務名稱 (是 `aoss` 而不是 `es`)。

```
// import OpenSearchClient to establish connection to OpenSearch Serverless collection
import org.opensearch.client.opensearch.OpenSearchClient;

SdkHttpClient httpClient = ApacheHttpClient.builder().build();
// create an opensearch client and use the request-signer
OpenSearchClient client = new OpenSearchClient(
    new AwsSdk2Transport(
        httpClient,
        "...us-west-2.aoss.amazonaws.com", // serverless collection endpoint
        "aoss" // signing service name
        Region.US_WEST_2, // signing service region
        AwsSdk2TransportOptions.builder().build()
    )
);

String index = "sample-index";

// create an index
CreateIndexRequest createIndexRequest = new
 CreateIndexRequest.Builder().index(index).build();
CreateIndexResponse createIndexResponse = client.indices().create(createIndexRequest);
System.out.println("Create index reponse: " + createIndexResponse);
```

```
// delete the index
DeleteIndexRequest deleteIndexRequest = new
 DeleteIndexRequest.Builder().index(index).build();
DeleteIndexResponse deleteIndexResponse = client.indices().delete(deleteIndexRequest);
System.out.println("Delete index reponse: " + deleteIndexResponse);

httpClient.close();
```

下列範例程式碼會再次建立安全連線，然後搜尋索引。

```
import org.opensearch.client.opensearch.OpenSearchClient;

SdkHttpClient httpClient = ApacheHttpClient.builder().build();

OpenSearchClient client = new OpenSearchClient(
    new AwsSdk2Transport(
        httpClient,
        "...us-west-2.aoss.amazonaws.com", // serverless collection endpoint
        "aoss" // signing service name
        Region.US_WEST_2, // signing service region
        AwsSdk2TransportOptions.builder().build()
    )
);

Response response = client.generic()
    .execute(
        Requests.builder()
            .endpoint("/" + "users" + "/_search?typed_keys=true")
            .method("GET")
            .json("{"
                + "    \"query\": {"
                + "        \"match_all\": {}"
                + "    }"
                + "}")
            .build());

httpClient.close();
```

JavaScript

下列範本程式碼會使用 JavaScript 的 opensearch-js 用戶端，建立與指定 OpenSearch Serverless 集合的安全連線，建立單一索引，新增文件，以及刪除索引。您必須提供 node 和 `region` 的值。

與 OpenSearch Service 網域相較，重要的區別在於服務名稱 (是 aoss 而不是 es)。

Version 3

此範例使用適用於 JavaScript in Node.js 的 SDK 的版本 3。

```
const { defaultProvider } = require('@aws-sdk/credential-provider-node');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () => {
                const credentialsProvider = defaultProvider();
                return credentialsProvider();
            },
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index }),body) {
        console.log((await client.indices.create({ index })).body);
    }

    // add a document to the index
    const document = { foo: 'bar' };
    const response = await client.index({
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}
```

```
main();
```

Version 2

此範例使用適用於 JavaScript in Node.js 的 SDK 的版本 2。

```
const AWS = require('aws-sdk');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () =>
                new Promise((resolve, reject) => {
                    AWS.config.getCredentials((err, credentials) => {
                        if (err) {
                            reject(err);
                        } else {
                            resolve(credentials);
                        }
                    });
                }),
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index })).body) {
        console.log((await client.indices.create({
            index
        })).body);
    }

    // add a document to the index
    const document = {
        foo: 'bar'
    };
    const response = await client.index({
```

```
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}

main();
```

Logstash

您可以使用 Logstash OpenSearch 外掛程式，將日誌發佈至 OpenSearch Serverless 集合。

使用 Logstash 將資料傳送至 OpenSearch Serverless

1. 使用 Docker 或 Linux 安裝 logstash-output-opensearch 外掛程式的 2.0.0 版或更新版本。logstash-output-opensearch

   Docker

   Docker 託管預先安裝了 OpenSearch 輸出外掛程式的 Logstash OSS 軟體：opensearchproject/logstash-oss-with-opensearch-output-plugin。您可以像任何其他映像一樣提取映像：

   ```
   docker pull opensearchproject/logstash-oss-with-opensearch-output-plugin:latest
   ```

   Linux

   首先，如果您尚未安裝最新版本的 Logstash，請先安裝。然後，安裝輸出外掛程式的 2.0.0 版本：

   ```
   cd logstash-8.5.0/
   bin/logstash-plugin install --version 2.0.0 logstash-output-opensearch
   ```

   如果外掛程式已安裝，請將其更新至最新版本：

```
bin/logstash-plugin update logstash-output-opensearch
```

從 2.0.0 版的外掛程式開始， AWS 軟體開發套件使用 第 3 版。如果您使用的是早於 8.4.0 的 Logstash 版本，則必須移除任何預先安裝的 AWS 外掛程式，並安裝`logstash-integration-aws`外掛程式：

```
/usr/share/logstash/bin/logstash-plugin remove logstash-input-s3
/usr/share/logstash/bin/logstash-plugin remove logstash-input-sqs
/usr/share/logstash/bin/logstash-plugin remove logstash-output-s3
/usr/share/logstash/bin/logstash-plugin remove logstash-output-sns
/usr/share/logstash/bin/logstash-plugin remove logstash-output-sqs
/usr/share/logstash/bin/logstash-plugin remove logstash-output-cloudwatch

/usr/share/logstash/bin/logstash-plugin install --version 0.1.0.pre logstash-
integration-aws
```

2. 若要讓 OpenSearch 輸出外掛程式與 OpenSearch Serverless 搭配使用，您必須對 logstash.conf 的 `opensearch` 輸出區段進行下列修改：

   - 將 `aoss` 指定為 `auth_type` 下的 `service_name`。
   - 針對 `hosts` 指定您的集合端點。
   - 新增參數 `default_server_major_version` 和 `legacy_template`。外掛程式需要這些參數才能與 OpenSearch Serverless 搭配使用。

```
output {
  opensearch {
    hosts => "collection-endpoint:443"
    auth_type => {
      ...
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

此組態檔案範例會從 S3 儲存貯體中的檔案取得輸入，並將其傳送至 OpenSearch Serverless 集合：

```
input {
  s3  {
    bucket => "my-s3-bucket"
    region => "us-east-1"
  }
}

output {
  opensearch {
    ecs_compatibility => disabled
    hosts => "https://my-collection-endpoint.us-east-1.aoss.amazonaws.com:443"
    index => my-index
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

3. 然後，使用新的組態執行 Logstash 來測試外掛程式：

```
bin/logstash -f config/test-plugin.conf
```

Python

下列範本程式碼會使用 Python 的 opensearch-py 用戶端，建立與指定 OpenSearch Serverless 集合的安全連線，建立單一索引並搜尋該索引。您必須提供 `region` 和 `host` 的值。

與 OpenSearch Service 網域相較，重要的區別在於服務名稱 (是 `aoss` 而不是 `es`)。

```
from opensearchpy import OpenSearch, RequestsHttpConnection, AWSV4SignerAuth
import boto3

host = ''  # serverless collection endpoint, without https://
region = ''  # e.g. us-east-1
```

```
service = 'aoss'
credentials = boto3.Session().get_credentials()
auth = AWSV4SignerAuth(credentials, region, service)

# create an opensearch client and use the request-signer
client = OpenSearch(
    hosts=[{'host': host, 'port': 443}],
    http_auth=auth,
    use_ssl=True,
    verify_certs=True,
    connection_class=RequestsHttpConnection,
    pool_maxsize=20,
)

# create an index
index_name = 'books-index'
create_response = client.indices.create(
    index_name
)

print('\nCreating index:')
print(create_response)

# index a document
document = {
  'title': 'The Green Mile',
  'director': 'Stephen King',
  'year': '1996'
}

response = client.index(
    index = 'books-index',
    body = document,
    id = '1'
)


# delete the index
delete_response = client.indices.delete(
    index_name
)

print('\nDeleting index:')
```

```
print(delete_response)
```

Ruby

opensearch-aws-sigv4 Gem 套件提供了立即使用的 OpenSearch Serverless 的存取，以及 OpenSearch 服務。它具有 opensearch-ruby 客戶端的所有功能，因為這是此 Gem 套件的相依項目。

執行個體化 Sigv4 簽署者時，請指定 aoss 為服務名稱：

```
require 'opensearch-aws-sigv4'
require 'aws-sigv4'

signer = Aws::Sigv4::Signer.new(service: 'aoss',
                                region: 'us-west-2',
                                access_key_id: 'key_id',
                                secret_access_key: 'secret')

# create an opensearch client and use the request-signer
client = OpenSearch::Aws::Sigv4Client.new(
  { host: 'https://your.amz-opensearch-serverless.endpoint',
    log: true },
  signer)

# create an index
index = 'prime'
client.indices.create(index: index)

# insert data
client.index(index: index, id: '1', body: { name: 'Amazon Echo',
                                            msrp: '5999',
                                            year: 2011 })

# query the index
client.search(body: { query: { match: { name: 'Echo' } } })

# delete index entry
client.delete(index: index, id: '1')

# delete the index
client.indices.delete(index: index)
```

使用其他用戶端簽署 HTTP 請求

當您使用其他用戶端建構 HTTP 請求，並簽署對 OpenSearch Serverless 集合的請求時，下列要求適用。

- 您必須將服務名稱指定為 aoss。
- 所有 AWS Signature 版本 4 請求均需要 x-amz-content-sha256 標頭。其提供請求承載的雜湊。如果有請求承載，請將值設定為其安全雜湊演算法 (SHA) 加密雜湊 (SHA256)。如果沒有請求承載，請將值設定為 e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855，這是一個空字串的雜湊。

主題

- 使用 cURL 編製索引
- 使用 Postman 編製索引

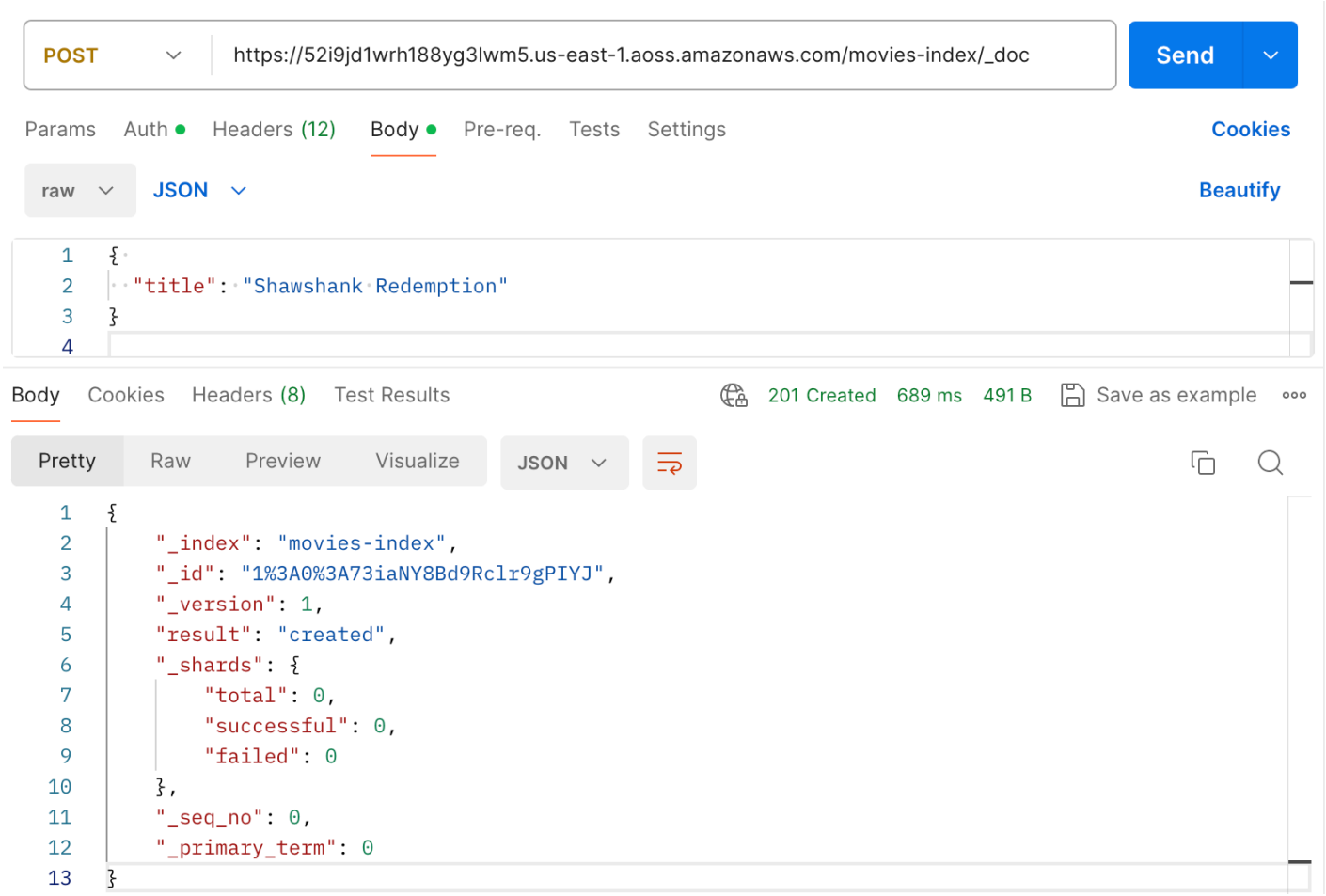
使用 cURL 編製索引

下列範例請求使用用戶端 URL 請求程式庫 (cURL)，將單一文件傳送至集合movies-index中名為 的索引：

```
curl -XPOST \
    --user "$AWS_ACCESS_KEY_ID":"$AWS_SECRET_ACCESS_KEY" \
    --aws-sigv4 "aws:amz:us-east-1:aoss" \
    --header "x-amz-content-sha256: $REQUEST_PAYLOAD_SHA_HASH" \
    --header "x-amz-security-token: $AWS_SESSION_TOKEN" \
    "https://my-collection-endpoint.us-east-1.aoss.amazonaws.com/movies-index/_doc" \
    -H "Content-Type: application/json" -d '{"title": "Shawshank Redemption"}'
```

使用 Postman 編製索引

下圖顯示如何使用 Postman 將請求傳送至集合。如需驗證的指示，請參閱 Postman 中的使用 AWS 簽章驗證工作流程進行驗證。



在 Amazon OpenSearch Serverless 上設定Machine Learning

機器學習

Machine Learning (ML) 以 ML 演算法和遠端模型的形式提供 ML 功能。透過存取這些模型，您可以執行數個 AI 工作流程，例如 RAG 或語意搜尋。ML 支援使用您可以使用連接器設定的最新外部託管模型進行生成式 AI 使用案例的實驗和生產部署。設定連接器之後，您必須將其設定為模型，然後將其部署以執行預測。

連接器

連接器有助於存取第三方 ML 平台上託管的模型。它們可做為 OpenSearch 叢集與遠端模型之間的閘道。如需詳細資訊，請參閱下列 文件：

- 在 OpenSearch 文件網站上建立第三方 ML 平台的連接器
- 外部平台的連接器

- 的連接器 AWS 服務

> **Important**
>
> - 當您建立信任政策時，請將 新增**ml.opensearchservice.amazonaws.com**為 OpenSearch Service 原則。
> - 略過 連接器頁面上顯示如何在政策中設定網域的步驟。
> - 在設定許可步驟中新增 `iam:PassRole`陳述式。
> - 在 OpenSearch Dashboards 中略過映射 ML 角色步驟。不需要後端角色組態。這適用於外部平台的 Connectors AWS 服務 和 Connectors。 https://docs.aws.amazon.com/opensearch-service/latest/developerguide/ml-external-connector.html
> - 在對集合端點的 SigV4 請求中，將服務名稱設定為 ，**aoss**而不是 **es**。

模型

模型是用於各種 AI 工作流程的核心功能。一般而言，您可以將連接器與模型建立關聯，以使用連接器執行預測。模型處於部署狀態後，您可以執行預測。如需詳細資訊，請參閱 OpenSearch 文件網站上的註冊託管在第三方平台上的模型。

> **Note**
>
> OpenSearch Serverless 不支援所有模型功能，例如本機模型。如需詳細資訊，請參閱不支援的Machine Learning API 和功能 APIs。

設定Machine Learning的許可

下一節說明Machine Learning (ML) 所需的集合資料存取政策。將*#####*取代為您的特定資訊。如需詳細資訊，請參閱支援的政策許可。

```
{
    "Rules": [
        {
            "Resource": [
                "model/collection_name/*"
            ],
```

```
            "Permission": [
                "aoss:DescribeMLResource",
                "aoss:CreateMLResource",
                "aoss:UpdateMLResource",
                "aoss:DeleteMLResource",
                "aoss:ExecuteMLResource"
            ],
            "ResourceType": "model"
        }
    ],
    "Principal": [
        "arn:aws:iam::account_id:role/role_name"
    ],
    "Description": "ML full access policy for collection_name"
}
```

- aoss：DescribeMLResource – 准許搜尋和查詢連接器、模型和模型群組。
- aoss：CreateMLResource – 准許建立連接器、模型和模型群組。
- aoss：UpdateMLResource – 准許更新連接器、模型和模型群組。
- aoss：DeleteMLResource – 准許刪除連接器、模型和模型群組。
- aoss：ExecuteMLResource – 准許對模型執行預測。

不支援的Machine Learning API 和功能 APIs

不支援APIs

Amazon OpenSearch Serverless 不支援下列Machine Learning (ML) APIs：

- 本機模型功能
- Model Train API
- 模型預測演算法 API
- 模型批次預測 API
- 代理程式 API 及其對應的工具
- MCP 伺服器 APIs
- 記憶體 APIs
- 控制器 APIs

- 執行演算法 API
- ML 設定檔 AP
- ML 統計資料 API

如需 ML APIs的詳細資訊，請參閱 OpenSearch 文件網站上的 ML APIs。

不支援的功能

Amazon OpenSearch Serverless 不支援下列 ML 功能：

- 代理程式和工具
- 本機模型
- 搜尋和擷取管道中的 ML 推論處理器
  - ML 推論擷取處理器
  - ML 推論搜尋回應處理器
  - ML 推論搜尋請求處理器

如需這些功能的詳細資訊，請參閱 OpenSearch 文件網站上的下列文件：

- 機器學習
- ML 推論處理器
- 搜尋管道

在 OpenSearch Serverless 上設定神經搜尋和混合搜尋

神經搜尋

Amazon OpenSearch Serverless 支援對資料進行語意搜尋操作的神經搜尋功能。神經搜尋使用機器學習模型來了解查詢的語意意義和內容，提供比傳統關鍵字式搜尋更相關的搜尋結果。本節說明如何在 OpenSearch Serverless 中設定神經搜尋，包括所需的許可、支援的處理器，以及與標準 OpenSearch 實作的金鑰差異。

使用神經搜尋，您可以對資料執行語意搜尋，這會考慮語意意義來了解搜尋查詢的意圖。此功能由下列元件提供支援：

- 文字內嵌擷取管道處理器

- 神經查詢
- 神經稀疏查詢

混合搜尋

透過混合式搜尋，您可以結合關鍵字和語意搜尋功能來改善搜尋相關性。若要使用混合式搜尋，請建立搜尋管道來處理搜尋結果並結合文件分數。如需詳細資訊，請參閱 OpenSearch 文件網站上的搜尋管道。使用下列元件實作混合式搜尋：

- 標準化搜尋管道處理器

  支援的標準化技術

  - `min_max`
  - `l2`

  支援的組合技術

  - `arithmetic_mean`
  - `geometric_mean`
  - `harmonic_mean`

  如需標準化和組合技術的詳細資訊，請參閱 OpenSearch 文件網站上的請求內文欄位。
- 混合查詢

神經和混合查詢

根據預設，OpenSearch 會使用關鍵字型 Okapi BM25 演算法計算文件分數，這非常適合包含關鍵字的搜尋查詢。神經搜尋為自然語言查詢提供了新的查詢類型，並能夠結合語意和關鍵字搜尋。

Example: neural

```
"neural": {
  "vector_field": {
    "query_text": "query_text",
    "query_image": "image_binary",
    "model_id": "model_id",
    "k": 100
  }
}
```

如需詳細資訊，請參閱 OpenSearch 文件網站上的神經查詢。

Example: hybrid

```
"hybrid": {
      "queries": [
        array of lexical, neural, or combined queries
      ]
    }
```

如需詳細資訊，請參閱 OpenSearch 文件網站上的混合查詢。

若要在 Amazon OpenSearch Serverless 中設定語意搜尋元件，請遵循 OpenSearch 文件網站上的神經搜尋教學課程中的步驟。請記住這些重要差異：

- OpenSearch Serverless 僅支援遠端模型。您必須將連接器設定為遠端託管模型。您不需要部署或移除遠端模型。如需詳細資訊，請參閱 OpenSearch 文件網站上的語意和混合式搜尋入門。
- 當您搜尋向量索引或搜尋最近建立的搜尋和擷取管道時，最多會有 15 秒的延遲。

設定 許可

OpenSearch Serverless 中的神經搜尋需要下列許可。如需詳細資訊，請參閱支援的政策許可。

Example：神經搜尋政策

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "NeuralSearch",
            "Effect": "Allow",
            "Action": [
              "aoss:CreateIndex",
              "aoss:CreateCollection",
              "aoss:UpdateCollection",
              "aoss:DeleteIndex",
              "aoss:DeleteCollection"
            ],
            "Resource": "arn:aws:aoss:us-east-1:111122223333:collection/your-collection-name"
        }
    ]
}
```

- aoss：*Index – 建立存放文字內嵌的向量索引。
- aoss：*CollectionItems – 建立擷取和搜尋管道。
- aoss：*MLResource – 建立和註冊文字內嵌模型。
- aoss：APIAccessAll – 提供 OpenSearch APIs存取權，以進行搜尋和擷取操作。

以下說明神經搜尋所需的集合資料存取政策。將#####取代為您的特定資訊。

```
            {
                "ResourceType": "index",
                "Resource": ["index/collection_name/*"],
                "Permission": [
                  "aoss:CreateIndex",
                  "aoss:DescribeIndex",
                  "aoss:UpdateIndex",
                  "aoss:DeleteIndex"
                ]
            }
        ],
        "Principal": [
            "arn:aws:iam::account_id:role/role_name"
        ]
    },
    {
        "Description": "Create pipeline permission",
        "Rules": [
            {
                "ResourceType": "collection",
                "Resource": ["collection/collection_name"],
                "Permission": [
                  "aoss:CreateCollectionItems",
                  "aoss:DescribeCollectionItems",
                  "aoss:UpdateCollectionItems",
                  "aoss:DeleteCollectionItems"
                ]
            }
        ],
        "Principal": [
            "arn:aws:iam::account_id:role/role_name"
        ]
    },
    {
        "Description": "Create model permission",
        "Rules": [
            {
                "ResourceType": "model",
                "Resource": ["model/collection_name/*"],
                "Permission": ["aoss:CreateMLResources"]
            }
        ],
        "Principal": [
            "arn:aws:iam::account_id:role/role_name"
        ]
    }
]
```

在 Amazon OpenSearch Serverless 上設定工作流程

工作流程

工作流程支援建置者在 OpenSearch 上創新 AI 應用程式。目前在 OpenSearch 中使用機器學習 (ML) 產品的程序，例如語意搜尋，需要複雜的設定和預先處理任務，以及詳細的使用者查詢，這兩者都可能耗時且容易出錯。工作流程是一種簡化架構，可鏈結 OpenSearch 的多個 API 呼叫。

如需設定和使用方式，請參閱 OpenSearch 網站上的自動化組態。當您在 OpenSearch Serverless 中使用工作流程時，請考慮下列重要差異：

- OpenSearch Serverless 在工作流程步驟中僅使用遠端模型。您不需要部署這些模型。
- OpenSearch Serverless 不支援重新索引工作流程步驟。
- 當您在其他 API 呼叫之後搜尋工作流程和工作流程狀態時， 預期最多延遲 15 秒，才會顯示更新。

OpenSearch Serverless Collections 在 OpenSearch UI 應用程式中用作資料來源時支援工作流程。如需詳細資訊，請參閱管理資料來源關聯。

設定 許可

在您建立和佈建範本之前，請確認您擁有必要的許可。如果您需要協助，請聯絡您的 帳戶管理員。OpenSearch Serverless Workflows 需要下列許可。您可以在 IAM 政策中定義集合資源 ARN，以限制特定集合的許可範圍。

Example：工作流程政策

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "NeuralSearch",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/Cognito_identitypoolname/Auth_Role"
        ]
      },
      "Action": [
        "aoss:CreateIndex",
        "aoss:CreateCollection",
        "aoss:UpdateCollection",
        "aoss:DeleteIndex",
        "aoss:DeleteCollection"
      ],
      "Resource": "arn:aws:aoss:us-east-1:111122223333:collection/your-
collection-name"
    }
  ]
}
```

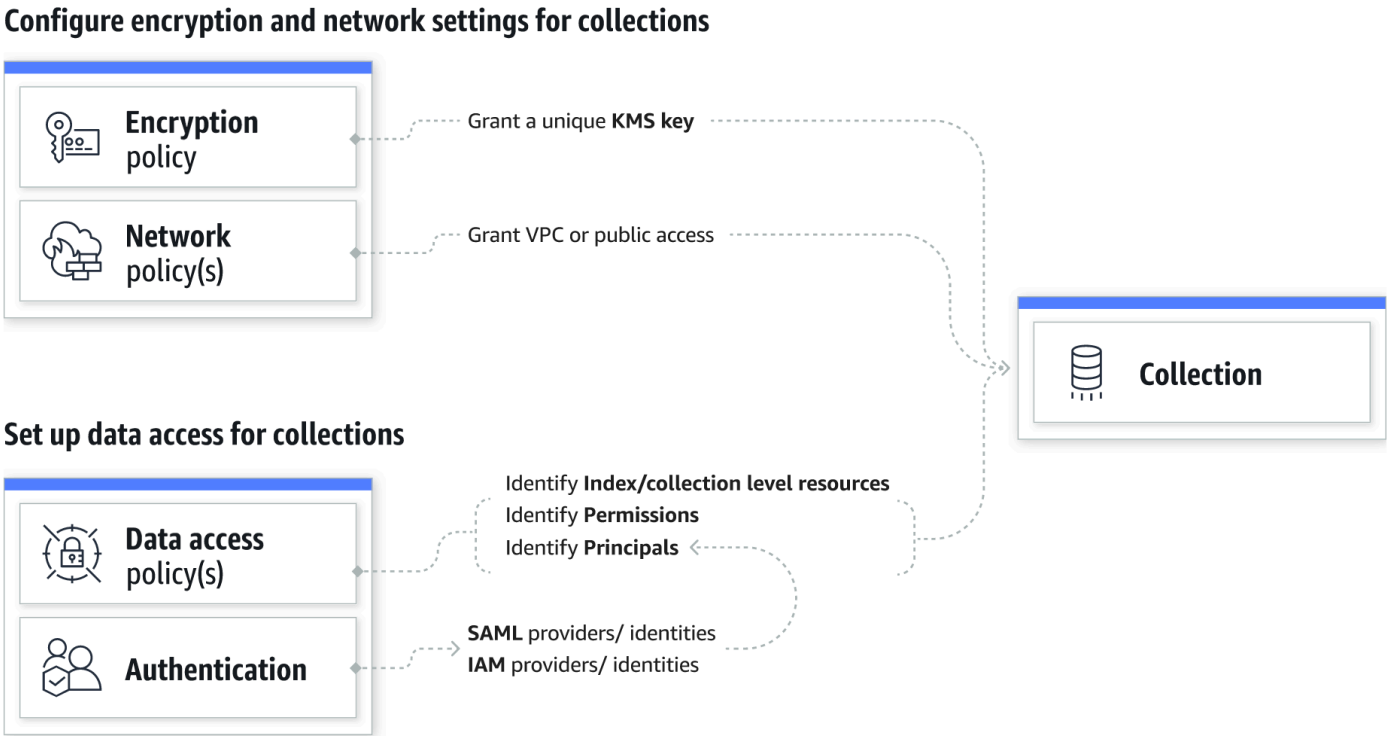
- aoss：*CollectionItems – 准許建立和管理範本，以及佈建搜尋和擷取管道。
- aoss：*Index – 准許使用 OpenSearch API 操作建立和刪除索引。
- aoss：*MLResource – 准許佈建使用設定Machine Learning的工作流程步驟。

Amazon OpenSearch Serverless 的安全概觀

Amazon OpenSearch Serverless 的安全功能在以下方面與 Amazon OpenSearch Service 的安全功能有根本上的不同：

功能	OpenSearch Service	OpenSearch Serverless
資料存取控制	資料存取由 IAM 政策和精細存取控制決定。	資料存取由資料存取政策決定。
靜態加密	網域的靜態加密是選擇性。	集合需要靜態加密。
安全設定和管理	您必須個別設定每個網域的網路、加密和資料存取。	您可以使用安全政策大規模管理多個集合的安全設定。

下圖說明構成功能集合的安全元件。集合必須具有指派的加密金鑰、網路存取設定和相符的資料存取政策，以授予其資源許可。



主題

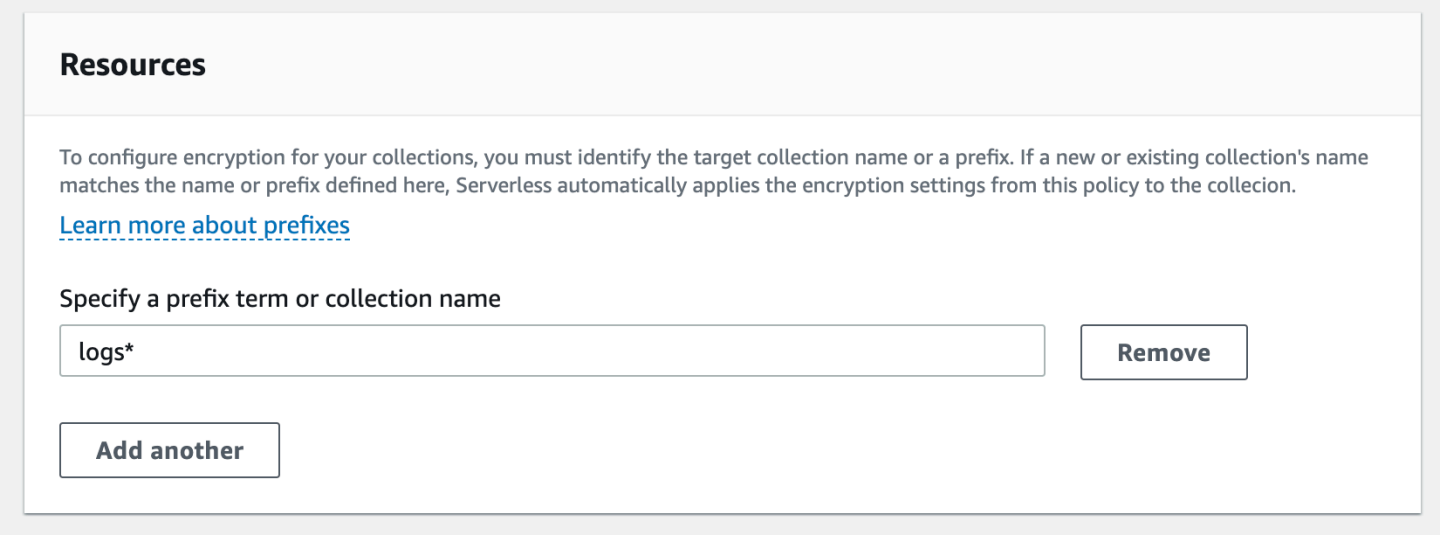
- 加密政策
- 網路政策
- 資料存取政策
- IAM 和 SAML 身分驗證
- 基礎設施安全性

- 開始使用 Amazon OpenSearch Serverless 中的安全功能
- Amazon OpenSearch Serverless 的身分和存取管理
- Amazon OpenSearch Serverless 中的加密
- Amazon OpenSearch Serverless 的網路存取
- Amazon OpenSearch Serverless 中的 FIPS 合規
- Amazon OpenSearch Serverless 的資料存取控制
- 透過 的資料平面存取 AWS PrivateLink
- 透過 控制平面存取 AWS PrivateLink
- Amazon OpenSearch Serverless 的 SAML 身分驗證
- Amazon OpenSearch Serverless 的合規驗證

加密政策

加密政策會定義您的集合是使用 AWS 擁有的金鑰 還是客戶受管金鑰加密。加密政策由兩個元件組成：資源模式和加密金鑰。資源模式定義政策適用於哪個或哪些集合。加密金鑰決定如何保護相關聯的集合。

若要將政策套用至多個集合，請在政策規則中包含萬用字元 (*)。例如，下列政策適用於名稱以「logs」開頭的所有集合。

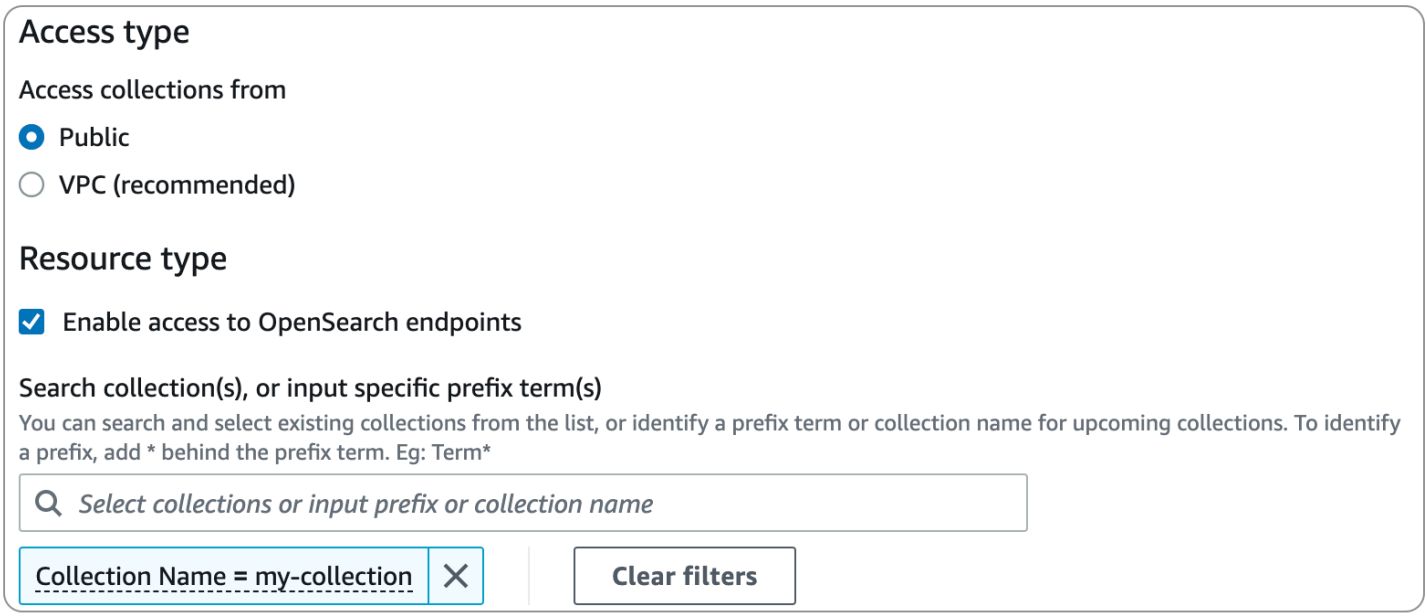


加密政策可簡化建立和管理集合的程序，尤其是以程式設計方式這樣做時。您可以指定名稱來建立集合，加密金鑰會在建立時自動指派給集合。

網路政策

網路政策會定義您的集合是否可私下存取，或從公有網路透過網際網路存取。私有集合可透過 OpenSearch Serverless 受管 VPC 端點，或使用私有存取的特定 Amazon Bedrock AWS 服務 存取。AWS 服務 正如加密政策，網路政策可套用至多個集合，以便您大規模管理許多集合的網路存取。

網路政策由兩個元件組成：存取類型和資源類型。存取類型可以是公有或私有。資源類型決定您選擇的存取權是否套用至集合端點、OpenSearch Dashboards 端點，或兩者。



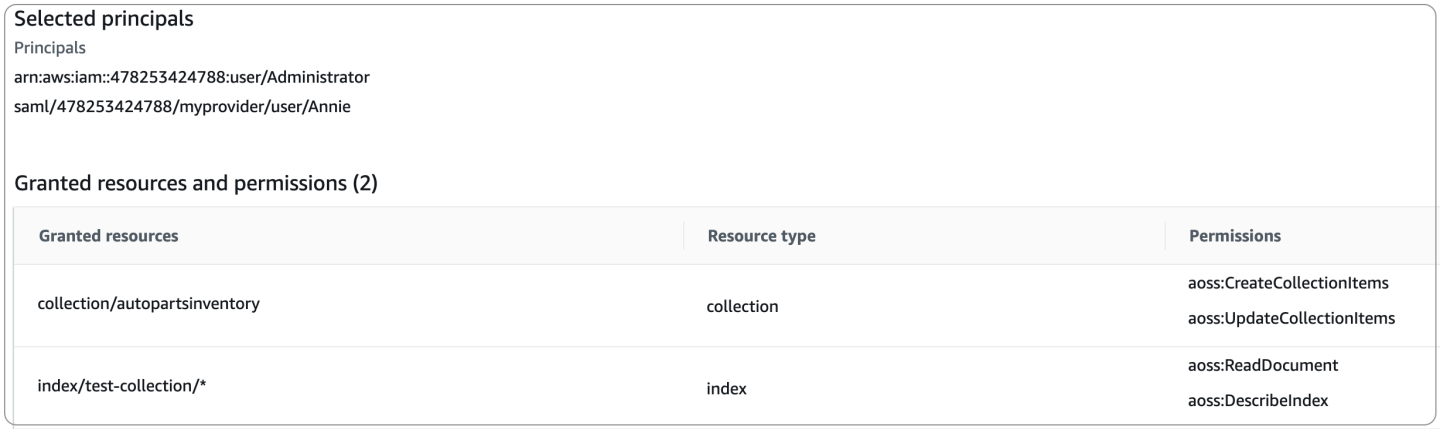
如果您計劃在網路政策內設定 VPC 存取，您必須先建立一個或多個 OpenSearch Serverless 受管 VPC 端點。這些端點可讓您像在 VPC 中一樣存取 OpenSearch Serverless，無需使用網際網路閘道、NAT 裝置、VPN 連線或 Direct Connect 連線。

的私有存取 AWS 服務 只能套用至集合的 OpenSearch 端點，無法套用至 OpenSearch Dashboards 端點。 AWS 服務 無法授予 OpenSearch Dashboards 的存取權。

資料存取政策

資料存取政策定義您的使用者如何存取集合內的資料。資料存取政策會自動將存取許可指派給符合特定模式的集合和索引，以協助您大規模管理集合。可將多個政策套用至單一資源。

資料存取政策由一組規則組成，每個規則都有三個元件：資源類型、授予的資源和一組許可。資源類型可以是集合或索引。授予的資源可以是集合/索引名稱或具有萬用字元 (*) 的模式。許可清單指定政策授予存取權的 OpenSearch API 操作。此外，政策包含主體清單，其中指定要授予存取權的 IAM 角色、使用者和 SAML 身分。



如需有關資料存取政策格式的詳細資訊，請參閱政策語法。

建立資料存取政策之前，您必須擁有一個或多個 IAM 角色或使用者或 SAML 身分，才能在政策中提供存取權。如需詳細資訊，請參閱下節。

> **Note**
>
> 將集合從公有切換到私有存取， 會移除 OpenSearch Serverless 集合主控台中的索引索引標籤。

IAM 和 SAML 身分驗證

IAM 主體和 SAML 身分是資料存取政策的其中一個建構區塊。在存取政策的 `principal` 陳述式中，您可以包含 IAM 角色、使用者和 SAML 身分。接著會將您在相關聯政策規則中指定的許可授予這些主體。

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
```

```
      "Principal":[
         "arn:aws:iam::123456789012:user/Dale",
         "arn:aws:iam::123456789012:role/RegulatoryCompliance",
         "saml/123456789012/myprovider/user/Annie"
      ]
   }
]
```

您可以直接在 OpenSearch Serverless 中設定 SAML 身分驗證。如需詳細資訊，請參閱the section called “SAML 身分驗證”。

基礎設施安全性

Amazon OpenSearch Serverless 受到 AWS 全球網路安全的保護。如需安全服務以及如何 AWS 保護基礎設施的資訊 AWS ，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 Amazon OpenSearch Serverless。用戶端必須支援 Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。如需 TLS 1.3 支援的密碼清單，請參閱 Elastic Load Balancing 文件中的 TLS 通訊協定和密碼。

此外，您必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署請求。或者，您可以使用 AWS Security Token Service (AWS STS) 以產生暫時安全憑證以簽署請求。

開始使用 Amazon OpenSearch Serverless 中的安全功能

下列教學課程可協助您開始使用 Amazon OpenSearch Serverless。兩個教學課程完成的基本步驟相同，但其中一個使用主控台，另一個則使用 AWS CLI。

請注意，這些教學課程中的使用案例皆已簡化。網路和安全政策都相當公開。在生產工作負載中，建議您設定更強大的安全功能，例如 SAML 身分驗證、VPC 存取權和限制性資料存取政策。

主題

- 教學課程：開始使用 Amazon OpenSearch Serverless 中的安全功能 (主控台)
- 教學課程：開始使用 Amazon OpenSearch Serverless (CLI) 中的安全功能

教學課程：開始使用 Amazon OpenSearch Serverless 中的安全功能 (主控台)

此教學課程會帶您逐步了解使用 Amazon OpenSearch Serverless 主控台建立和管理安全性政策的基本步驟。

在本教學課程中，您將完成下列步驟：

1. 設定許可
2. 建立加密政策
3. 建立網路政策
4. 設定資料存取政策
5. 建立集合
6. 上傳並搜尋資料

本教學課程會逐步引導您使用 設定集合 AWS 管理主控台。如需使用 的相同步驟 AWS CLI，請參閱 the section called “教學課程：開始使用安全功能 (CLI)”。

步驟 1：設定許可

> **Note**
>
> 如果您已經使用更廣泛的身分型政策，例如 `Action":"aoss:*"` 或 `Action":"*"`，則可以略過此步驟。不過，在生產環境中，我們建議您遵循最低權限原則，並且僅指派任務完成所需的最低許可。

為完成本教學課程，您必須具備正確的 IAM 許可。使用者或角色必須連接身分型政策，該政策包含以下最低許可：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:CreateCollection",
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:ListSecurityPolicies",
        "aoss:CreateAccessPolicy",
```

```
        "aoss:GetAccessPolicy",
        "aoss:ListAccessPolicies"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

如需 OpenSearch Serverless 許可的完整清單，請參閱 the section called “身分和存取權管理”。

步驟 2：建立加密政策

加密政策會指定 OpenSearch Serverless 用來加密集合的 AWS KMS 金鑰。您可以使用 AWS 受管金鑰 或不同的金鑰來加密集合。為確保本教學課程簡單易懂，我們將使用 AWS 受管金鑰加密集合。

建立加密政策

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，展開 Serverless (無伺服器)，然後選擇 Encryption policies (加密政策)。
3. 選擇 Create encryption policy (建立加密政策)。
4. 將政策命名為 books-policy。如需相關描述，請輸入 Encryption policy for books collection (書籍集合的加密政策)。
5. 在 Resources (資源) 下，輸入 books (書籍)，這就是您將為集合命名的名稱。如果您想要更廣泛的名稱，可以包含星號 (books*)，以便將政策套用至以「書籍」一詞開頭的所有集合。
6. 對於加密，請保持選取使用 AWS 擁有的金鑰。
7. 選擇建立。

步驟 3：建立網路政策

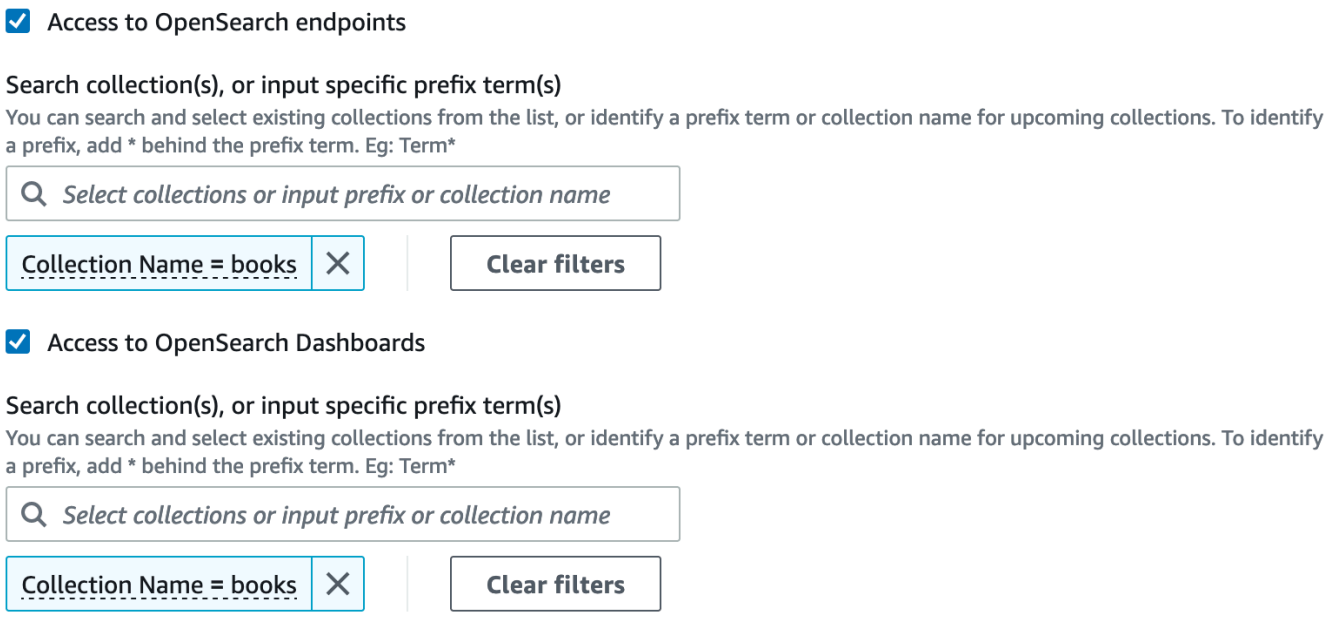
網路政策會決定集合是否可透過網際網路從公用網路存取，或者是否必須透過 OpenSearch Serverless 受管 VPC 端點進行存取。在本教學課程中，我們將設定公用存取權。

建立網路政策

1. 在左側導覽窗格中，選擇 Network policies (網路政策)，然後選擇 Create network policy (建立網路政策)。

2. 將政策命名為 books-policy。如需相關描述，請輸入 Network policy for books collection (書籍集合的網路政策)。
3. 在 Rule 1 (規則 1) 下，將規則命名為 Public access for books collection (書籍集合的公用存取權)。
4. 為確保本教學課程簡單易懂，我們將為書籍集合設定公用存取權。對於存取類型，選取 Public (公用)。
5. 我們要從 OpenSearch Dashboards 存取該集合。若要執行這項操作，您需要設定 Dashboards 以及 OpenSearch 端點的網路存取，否則 Dashboards 將無法運作。

   對於資源類型，請同時啟用 OpenSearch 端點的存取權和 OpenSearch Dashboards 的存取權。
6. 在兩個輸入方塊中，輸入 Collection Name = books (集合名稱 = 書籍)。此設定會縮減政策的範圍，使該政策僅套用至單一集合 (books)。您的規則應如下所示：



7. 選擇建立。

步驟 4：建立資料存取政策

在您設定資料存取之前，將無法存取集合資料。資料存取政策與您在步驟 1 中設定的 IAM 身分型政策不同。資料存取政策允許使用者存取集合中的實際資料。

在本教學課程中，我們將為單一使用者提供將資料索引編製為書籍集合所需的許可。

建立資料存取政策

1. 在左側導覽窗格中，選擇 Data access policies (資料存取政策)，然後選擇 Create access policy (建立存取政策)。
2. 將政策命名為 books-policy。如需相關描述，請輸入 Data access policy for books collection (書籍集合的資料存取政策)。
3. 選取 JSON 作為政策定義方法，並將下列政策貼到 JSON 編輯器中。

   將主體 ARN 取代為您將用來登入 OpenSearch Dashboards 和為資料編製索引的帳戶 ARN。

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/books/*"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:DescribeIndex",
               "aoss:ReadDocument",
               "aoss:WriteDocument",
               "aoss:UpdateIndex",
               "aoss:DeleteIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

   此政策為單一使用者提供在書籍集合中建立索引、為某些資料編製索引以及進行搜尋所需的最低許可。

4. 選擇建立。

步驟 5：建立集合

現在您已設定加密和網路政策，您可以建立相符的集合，而且安全設定會自動套用至該集合。

建立 OpenSearch Serverless 集合

1. 在左側導覽窗格中選擇 Collections (集合)，然後選擇 Create collection (建立集合)。
2. 將該集合命名為 books (書籍)。
3. 對於集合類型，選擇 Search (搜尋)。
4. 在 Encryption (加密) 下，OpenSearch Serverless 會告知您集合名稱與 `books-policy` 加密政策相符。
5. 在 Network access settings (網路存取設定) 下，OpenSearch Serverless 會告知您集合名稱與 `books-policy` 網路政策相符。
6. 選擇下一步。
7. 在 Data access policy options (資料存取政策選項) 下，OpenSearch Serverless 會通知您集合名稱符合 `books-policy` 資料存取政策。
8. 選擇下一步。
9. 檢閱集合組態，然後選擇 Submit (提交)。集合初始化所需的時間通常不到一分鐘。

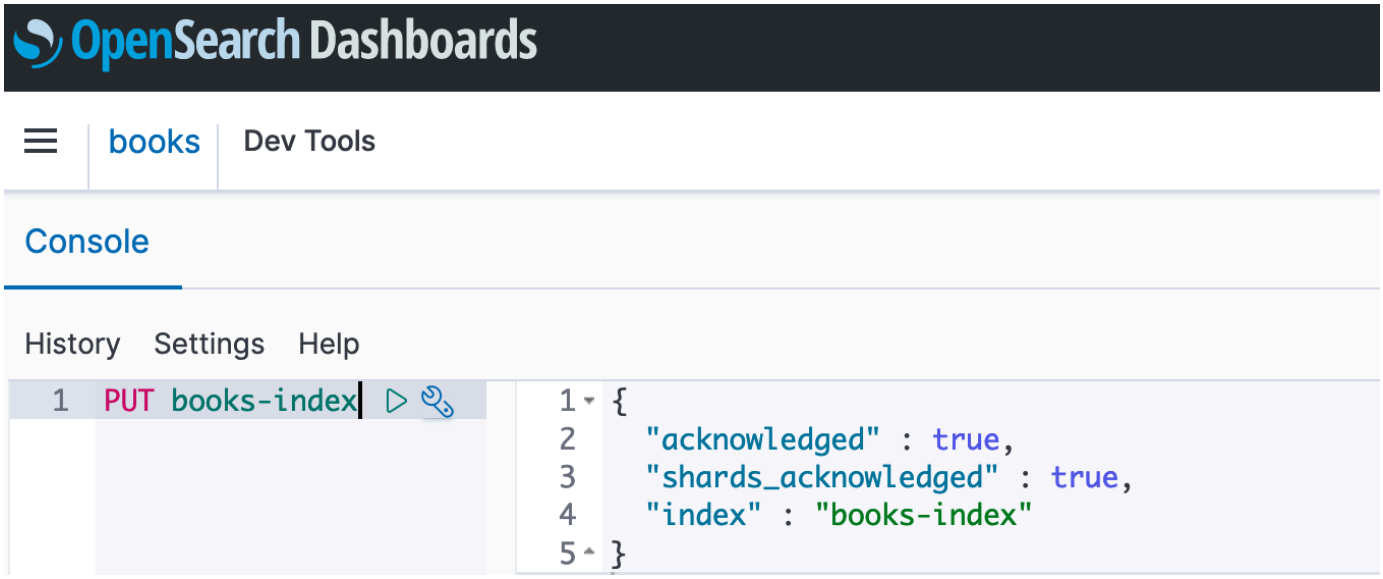
步驟 6：上傳並搜尋資料

您可以使用 Postman 或 curl，將資料上傳至 OpenSearch Serverless 集合。為了簡潔起見，這些範例使用 OpenSearch Dashboards 主控台中的 Dev Tools (開發工具)。

在集合中為資料編製索引和進行搜尋

1. 在左側導覽窗格中選擇 Collections (集合)，然後選擇 books (書籍) 集合以開啟其詳細資訊頁面。
2. 選擇集合的 OpenSearch Dashboards URL。URL 採用的格式為 `https://collection-id.us-east-1.aoss.amazonaws.com/_dashboards`。
3. 使用您在資料存取政策中指定之主體的 AWS 存取金鑰和私密金鑰登入 OpenSearch Dashboards。
4. 在 OpenSearch Dashboards 中，開啟左側導覽選單，然後選擇 Dev Tools (開發工具)。
5. 若要建立名為 books-index 的單一索引，請執行下列命令：

```
PUT books-index
```



6. 若要將單一文件的索引編製為 books-index，請執行下列命令：

```
PUT books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

7. 若要在 OpenSearch Dashboards 中搜尋資料，您需要設定至少一個索引模式。OpenSearch 使用這些模式來識別您要分析的索引。開啟 Dashboards 主選單，選擇 Stack Management (堆疊管理)，選擇 Index Patterns (索引模式)，然後選擇 Create index pattern (建立索引模式)。在本教學課程中，輸入 books-index。
8. 選擇 Next step (下一步)，然後選擇 Create index pattern (建立索引模式)。建立模式之後，您可以檢視各種文件欄位，例如 `author` 和 `title`。
9. 若要開始搜尋資料，請再次開啟主選單，然後選擇 Discover (探索)，或使用搜尋 API。

教學課程：開始使用 Amazon OpenSearch Serverless (CLI) 中的安全功能

本教學課程會逐步引導您完成 主控台入門安全教學課程中所述的步驟，但使用 AWS CLI 而非 OpenSearch Service 主控台。

在本教學課程中，您會完成下列步驟：

1. 建立 IAM 許可政策

2. 將 IAM 政策連接至 IAM 角色
3. 建立加密政策
4. 建立網路政策
5. 建立集合
6. 設定資料存取政策
7. 擷取集合端點
8. 將資料上傳至您的連線
9. 搜尋集合中的資料

本教學課程的目標是使用相當簡單的加密、網路和資料存取設定，來設定單一 OpenSearch Serverless 集合。例如，我們將設定公有網路存取、 AWS 受管金鑰 用於加密的 ，以及將最低許可授予單一使用者的簡化資料存取政策。

在生產案例中，請考慮實作更強大的組態，包括 SAML 身分驗證、自訂加密金鑰和 VPC 存取權。

在 OpenSearch Serverless 中開始使用安全政策

1.

> **Note**
>
> 如果您已經使用更廣泛的身分型政策，例如 `Action":"aoss:*"` 或 `Action":"*"`，則可以略過此步驟。不過，在生產環境中，我們建議您遵循最低權限原則，並且僅指派任務完成所需的最低許可。

若要開始，請建立具有執行本教學課程中步驟所需最低許可 AWS Identity and Access Management 的政策。我們會將該政策命名為 `TutorialPolicy`：

```
aws iam create-policy \
  --policy-name TutorialPolicy \
  --policy-document "{\"Version\": \"2012-10-17\",\"Statement\":
 [{\"Action\": [\"aoss:ListCollections\",\"aoss:BatchGetCollection\",
\"aoss:CreateCollection\",\"aoss:CreateSecurityPolicy\",\"aoss:GetSecurityPolicy\",
\"aoss:ListSecurityPolicies\",\"aoss:CreateAccessPolicy\",\"aoss:GetAccessPolicy\",
\"aoss:ListAccessPolicies\"],\"Effect\": \"Allow\",\"Resource\": \"*\"}]}"
```

回應範例

```
{
    "Policy": {
        "PolicyName": "TutorialPolicy",
        "PolicyId": "ANPAW6WRAECKG6QJWUV7U",
        "Arn": "arn:aws:iam::123456789012:policy/TutorialPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2022-10-16T20:57:18+00:00",
        "UpdateDate": "2022-10-16T20:57:18+00:00"
    }
}
```

2. 將 TutorialPolicy 連接至 IAM 角色，該角色將在集合中為資料編制索引和進行搜尋。我們會將該使用者命名為 TutorialRole：

```
aws iam attach-role-policy \
  --role-name TutorialRole \
  --policy-arn arn:aws:iam::123456789012:policy/TutorialPolicy
```

3. 建立集合之前，您需要建立加密政策，以將 AWS 擁有的金鑰 指派給您在稍後步驟中建立的書籍集合。

   傳送下列請求，以建立書籍集合的加密政策：

```
aws opensearchserverless create-security-policy \
  --name books-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/books\"]}],\"AWSOwnedKey\":true}"
```

   回應範例

```
{
    "securityPolicyDetail": {
        "type": "encryption",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDAwNTk5MF8x",
        "policy": {
            "Rules": [
```

```
                {
                    "Resource": [
                        "collection/books"
                    ],
                    "ResourceType": "collection"
                }
            ],
            "AWSOwnedKey": true
        },
        "createdDate": 1669240005990,
        "lastModifiedDate": 1669240005990
    }
}
```

4. 建立網路政策，該政策會提供書籍集合的公用存取權：

```
aws opensearchserverless create-security-policy --name books-policy --type network
 \
  --policy "[{\"Description\":\"Public access for books collection\",\"Rules
\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/books\"]},
{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/books\"]}],
\"AllowFromPublic\":true}]"
```

回應範例

```
{
    "securityPolicyDetail": {
        "type": "network",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDI1Njk1NV8x",
        "policy": [
            {
                "Rules": [
                    {
                        "Resource": [
                            "collection/books"
                        ],
                        "ResourceType": "dashboard"
                    },
                    {
                        "Resource": [
                            "collection/books"
                        ],
```

```
                        "ResourceType": "collection"
                    }
                ],
                "AllowFromPublic": true,
                "Description": "Public access for books collection"
            }
        ],
        "createdDate": 1669240256955,
        "lastModifiedDate": 1669240256955
    }
}
```

5. 建立書籍集合：

```
aws opensearchserverless create-collection --name books --type SEARCH
```

回應範例

```
{
    "createCollectionDetail": {
        "id": "8kw362bpwg4gx9b2f6e0",
        "name": "books",
        "status": "CREATING",
        "type": "SEARCH",
        "arn": "arn:aws:aoss:us-
east-1:123456789012:collection/8kw362bpwg4gx9b2f6e0",
        "kmsKeyArn": "auto",
        "createdDate": 1669240325037,
        "lastModifiedDate": 1669240325037
    }
}
```

6. 建立資料存取政策，該政策會提供在書籍集合中為資料編製索引和進行搜尋的最低許可。將主體 ARN 取代為步驟 1 中的 TutorialRole ARN：

```
aws opensearchserverless create-access-policy \
  --name books-policy \
  --type data \
  --policy "[{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":
[\"index\/books\/books-index\"],\"Permission\":[\"aoss:CreateIndex
\",\"aoss:DescribeIndex\",\"aoss:ReadDocument\",\"aoss:WriteDocument
```

```
\",\"aoss:UpdateIndex\",\"aoss:DeleteIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:role\/TutorialRole\"]}]"
```

回應範例

```
{
    "accessPolicyDetail": {
        "type": "data",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDM5NDY1M18x",
        "policy": [
            {
                "Rules": [
                    {
                        "Resource": [
                            "index/books/books-index"
                        ],
                        "Permission": [
                            "aoss:CreateIndex",
                            "aoss:DescribeIndex",
                            "aoss:ReadDocument",
                            "aoss:WriteDocument",
                            "aoss:UpdateDocument",
                            "aoss:DeleteDocument"
                        ],
                        "ResourceType": "index"
                    }
                ],
                "Principal": [
                    "arn:aws:iam::123456789012:role/TutorialRole"
                ]
            }
        ],
        "createdDate": 1669240394653,
        "lastModifiedDate": 1669240394653
    }
}
```

TutorialRole 現在應該能夠在書籍集合中為文件編製索引和進行搜尋。

7. 若要呼叫 OpenSearch API，您需要集合端點。傳送下列請求以擷取 collectionEndpoint 參數：

```
aws opensearchserverless batch-get-collection --names books
```

回應範例

```
{
    "collectionDetails": [
        {
            "id": "8kw362bpwg4gx9b2f6e0",
            "name": "books",
            "status": "ACTIVE",
            "type": "SEARCH",
            "description": "",
            "arn": "arn:aws:aoss:us-
east-1:123456789012:collection/8kw362bpwg4gx9b2f6e0",
            "createdDate": 1665765327107,
            "collectionEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com",
            "dashboardEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com/_dashboards"
        }
    ],
    "collectionErrorDetails": []
}
```

> **Note**
>
> 在集合狀態變更為 ACTIVE 之前，您都無法看到集合端點。在集合成功建立前，您可能必須進行多次呼叫才能檢查狀態。

8. 使用 Postman 或 curl 等 HTTP 工具，將資料索引編製為書籍集合。我們將建立名為 books-index 的索引，並新增單一文件。

   使用 TutorialRole 的憑證，將下列請求傳送至您在上一步擷取的集合端點。

```
PUT https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

回應範例

```
{
  "_index" : "books-index",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 0,
    "successful" : 0,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 0
}
```

9. 若要開始在集合中搜尋資料，請使用搜尋 API。下列查詢會執行基本搜尋：

```
GET https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_search
```

回應範例

```
{
    "took": 405,
    "timed_out": false,
    "_shards": {
        "total": 6,
        "successful": 6,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": {
            "value": 2,
            "relation": "eq"
        },
        "max_score": 1.0,
        "hits": [
            {
                "_index": "books-index:0::3xJq14MBUaOS0wL26UU9:0",
                "_id": "F_bt4oMBLle5pYmm5q4T",
                "_score": 1.0,
```

```
                "_source": {
                    "title": "The Shining",
                    "author": "Stephen King",
                    "year": 1977
                }
            }
        ]
    }
}
```

Amazon OpenSearch Serverless 的身分和存取管理

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可以控制經過「身分驗證」 (已登入) 和獲得「授權」 (具有許可) 可使用 OpenSearch Serverless 資源的人員。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 適用於 OpenSearch Serverless 的身分型政策
- OpenSearch Serverless 的政策動作
- OpenSearch Serverless 的政策資源
- Amazon OpenSearch Serverless 的政策條件索引鍵
- 使用 OpenSearch Serverless 的 ABAC
- 將暫時性憑證與 OpenSearch Serverless 搭配使用
- OpenSearch Serverless 的服務連結角色
- 其他政策類型
- OpenSearch Serverless 的身分型政策範例
- Amazon OpenSearch Serverless 的 IAM Identity Center 支援

適用於 OpenSearch Serverless 的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

OpenSearch Serverless 的身分型政策範例

若要檢視 OpenSearch Serverless 身分型政策範例，請參閱 the section called “身分型政策範例”。

OpenSearch Serverless 的政策動作

支援政策動作：是

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

OpenSearch Serverless 中的政策動作會在動作之前使用以下字首：

```
aoss
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
      "aoss:action1",
      "aoss:action2"
         ]
```

您可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 `Describe` 文字的所有動作，請包含以下動作：

```
"Action": "aoss:List*"
```

若要檢視 OpenSearch Serverless 身分型政策範例，請參閱 OpenSearch Serverless 的身分型政策範例。

OpenSearch Serverless 的政策資源

支援政策資源：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (*) 表示該陳述式適用於所有資源。

```
"Resource": "*"
```

Amazon OpenSearch Serverless 的政策條件索引鍵

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素會根據定義的條件，指定陳述式的執行時機。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

除了屬性型存取控制 (ABAC)，OpenSearch Serverless 支援以下條件索引鍵：

- `aoss:collection`
- `aoss:CollectionId`
- `aoss:index`

即使提供存取政策和安全政策的許可，您仍然可以使用這些條件索引鍵。例如：

```
[
   {
      "Effect":"Allow",
      "Action":[
         "aoss:CreateAccessPolicy",
         "aoss:CreateSecurityPolicy"
      ],
      "Resource":"*",
      "Condition":{
         "StringLike":{
            "aoss:collection":"log"
         }
```

```
      }
   }
]
```

在此範例中，此條件適用於包含規則的政策，這些規則與集合名稱或模式相符。這些條件具有以下行為：

- `StringEquals`：適用的政策具有包含確切資源字串 "log" (亦即 `collection/log`) 的規則。
- `StringLike`：適用的政策所具備規則包含的資源字串包含字串 "log" (亦即 `collection/log`，但也是 `collection/logs-application` 或 `collection/applogs123`)。

> **Note**
>
> 集合條件索引鍵不適用於索引層級。例如，在上述政策中，該條件不會套用至包含資源字串 `index/logs-application/*` 的存取或安全政策。

若要查看 OpenSearch Serverless 條件索引鍵的清單，請參閱《服務授權參考》中的 Condition keys for Amazon OpenSearch Serverless (Amazon OpenSearch Serverless 的條件索引鍵)。若要了解您可以將條件索引鍵與哪些動作和資源搭配使用，請參閱 Actions defined by Amazon OpenSearch Serverless (Amazon OpenSearch Serverless 定義的動作)。

使用 OpenSearch Serverless 的 ABAC

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，依據稱為標籤的屬性來定義許可。您可以將標籤連接至 IAM 實體 AWS 和資源，然後設計 ABAC 政策，以便在委託人的標籤符合資源上的標籤時允許操作。

如需根據標籤控制存取，請使用 `aws:ResourceTag/`*`key-name`*、`aws:RequestTag/`*`key-name`* 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的使用屬性型存取控制 (ABAC)。

如需有關標記 OpenSearch Serverless 資源的詳細資訊，請參閱the section called “標記集合”。

將暫時性憑證與 OpenSearch Serverless 搭配使用

支援臨時憑證：是

臨時登入資料提供對 AWS 資源的短期存取，並在您使用聯合或切換角色時自動建立。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱《IAM 使用者指南》中的IAM 中的臨時安全憑證與可與 IAM 搭配運作的AWS 服務。

OpenSearch Serverless 的服務連結角色

支援服務連結角色：是

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需有關建立或管理 OpenSearch Serverless 服務連結角色的詳細資訊，請參閱the section called “集合建立角色”。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種服務，用於分組和集中管理您企業擁有的多個 AWS 帳戶。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱 AWS Organizations 使用者指南中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括支援 RCPs AWS 的服務清單，請參閱AWS Organizations《 使用者指南》中的資源控制政策 (RCPs)。

OpenSearch Serverless 的身分型政策範例

依預設，使用者和角色不具備建立或修改 OpenSearch Serverless 資源的許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需有關 Amazon OpenSearch Serverless 所定義之動作和資源類型的詳細資訊，包括每種資源類型的 ARN 格式，請參閱《服務授權參考》中的 Actions, resources, and condition keys for Amazon OpenSearch Serverless (適用於 Amazon OpenSearch Serverless 的動作、資源和條件索引鍵)。

主題

- 政策最佳實務
- 在主控台中使用 OpenSearch Serverless
- 管理 OpenSearch Serverless 集合
- 檢視 OpenSearch Serverless 集合
- 使用 OpenSearch API 操作
- 適用於 OpenSearch API 操作的 ABAC

政策最佳實務

身分型政策相當強大。他們可以判斷某個人員是否可以在您的帳戶中建立、存取或刪除 OpenSearch Serverless 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 OpenSearch Serverless 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 IAM 安全最佳實務。

在主控台中使用 OpenSearch Serverless

若要在 OpenSearch Service 主控台中存取 OpenSearch Serverless，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 AWS 帳戶中 OpenSearch Serverless 資源的詳細資訊。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (例如 IAM 角色等) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合您嘗試執行之 API 作業的動作就可以了。

下列政策讓使用者可以在 OpenSearch Service 主控台中存取 OpenSearch Serverless：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "aoss:ListCollections",
                "aoss:BatchGetCollection",
                "aoss:ListAccessPolicies",
                "aoss:ListSecurityConfigs",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:ListVpcEndpoints",
                "aoss:GetAccessPolicy",
                "aoss:GetAccountSettings",
```

```
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy"
            ]
        }
    ]
}
```

管理 OpenSearch Serverless 集合

此政策是「集合管理員」政策的範例，讓使用者可以管理 Amazon OpenSearch Serverless 集合。使用者可以建立、檢視和刪除集合。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:aoss:us-east-1:111122223333:collection/*",
            "Action": [
                "aoss:CreateCollection",
                "aoss:DeleteCollection",
                "aoss:UpdateCollection"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:ListCollections",
                "aoss:CreateAccessPolicy",
                "aoss:CreateSecurityPolicy"
            ],
            "Effect": "Allow"
        }
    ]
}
```

檢視 OpenSearch Serverless 集合

此範例政策讓使用者可以檢視其帳戶中所有 Amazon OpenSearch Serverless 集合的詳細資訊。使用者無法修改集合或任何關聯的安全政策。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:ListCollections",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:BatchGetCollection"
            ],
            "Effect": "Allow"
        }
    ]
}
```

使用 OpenSearch API 操作

資料平面 API 操作包含您在 OpenSearch Serverless 中使用的函數，以從服務衍生即時值。控制平面 API 操作包含您用來設定環境的函數。

若要從瀏覽器存取 Amazon OpenSearch Serverless 資料平面 APIs 和 OpenSearch Dashboards，您需要為集合資源新增兩個 IAM 許可。這些許可為 `aoss:APIAccessAll`和 `aoss:DashboardsAccessAll`。

> **Note**
>
> 從 2023 年 5 月 10 日開始，OpenSearch Serverless 需要這兩個新的集合資源 IAM 許可。`aoss:APIAccessAll` 許可允許資料平面存取，而 `aoss:DashboardsAccessAll`許可允許來自瀏覽器的 OpenSearch Dashboards。未新增兩個新的 IAM 許可會導致 403 錯誤。

此範例政策允許使用者存取其帳戶中指定集合的資料平面 APIs，以及存取其帳戶中所有集合的 OpenSearch Dashboards。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": "aoss:APIAccessAll",
            "Resource": "arn:aws:aoss:us-
east-1:111122223333:collection/collection-id"
        },
        {
            "Effect": "Allow",
            "Action": "aoss:DashboardsAccessAll",
            "Resource": "arn:aws:aoss:us-east-1:111122223333:dashboards/default"
        }
    ]
}
```

`aoss:APIAccessAll` 和 都會`aoss:DashboardsAccessAll`提供集合資源的完整 IAM 許可，而 Dashboards 許可也會提供 OpenSearch Dashboards 存取權。每個許可都獨立運作，因此 上的明確拒絕`aoss:APIAccessAll`不會封鎖對 資源的`aoss:DashboardsAccessAll`存取，包括開發工具。在上拒絕也是如此`aoss:DashboardsAccessAll`。OpenSearch Serverless 支援下列全域條件金鑰：

- `aws:CalledVia`
- `aws:CalledViaAWSService`
- `aws:CalledViaFirst`
- `aws:CalledViaLast`
- `aws:CurrentTime`
- `aws:EpochTime`
- `aws:PrincipalAccount`
- `aws:PrincipalArn`
- `aws:PrincipallsAWSService`
- `aws:PrincipalOrgID`

- aws:PrincipalOrgPaths
- aws:PrincipalType
- aws:PrincipalServiceName
- aws:PrincipalServiceNamesList
- aws:ResourceAccount
- aws:ResourceOrgID
- aws:ResourceOrgPaths
- aws:RequestedRegion
- aws:ResourceTag
- aws:SourceIp
- aws:SourceVpce
- aws:SourceVpc
- aws:userid
- aws:username
- aws:VpcSourceIp

以下是在委託人 IAM 政策aws:SourceIp的條件區塊中使用 進行資料平面呼叫的範例：

```
"Condition": {
    "IpAddress": {
         "aws:SourceIp": "203.0.113.0"
    }
}
```

以下是在委託人 IAM 政策aws:SourceVpc的條件區塊中使用 進行資料平面呼叫的範例：

```
"Condition": {
    "StringEquals": {
        "aws:SourceVpc": "vpc-0fdd2445d8EXAMPLE"
    }
}
```

此外，支援下列 OpenSearch Serverless 特定金鑰：

- aoss:CollectionId

- aoss:collection

以下是在委託人 IAM 政策aoss:collection的條件區塊中使用 進行資料平面呼叫的範例：

```
"Condition": {
    "StringLike": {
         "aoss:collection": "log-*"
    }
}
```

適用於 OpenSearch API 操作的 ABAC

身分型政策可讓您使用標籤來控制對 Amazon OpenSearch Serverless 資料平面 APIs存取。下列政策是允許附加主體存取資料平面 APIs 的範例，如果集合具有 team:devops標籤：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "aoss:APIAccessAll",
            "Resource": "arn:aws:aoss:us-
east-1:111122223333:collection/collection-id",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/team": "devops"
                }
            }
        }
    ]
}
```

如果集合具有 environment:production標籤，以下政策是拒絕連接主體存取資料平面 APIs 和儀表板存取的範例：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "aoss:APIAccessAll",
                "aoss:DashboardsAccessAll"
            ],
            "Resource": "arn:aws:aoss:us-
east-1:111122223333:collection/collection-id"
        }
    ]
}
```

Amazon OpenSearch Serverless 不支援資料平面 APIs 的 `RequestTag`和 `TagKeys`全域條件金鑰。

Amazon OpenSearch Serverless 的 IAM Identity Center 支援

Amazon OpenSearch Serverless 的 IAM Identity Center 支援

您可以使用 IAM Identity Center 主體（使用者和群組）透過 Amazon OpenSearch 應用程式存取 Amazon OpenSearch Serverless 資料。若要啟用 Amazon OpenSearch Serverless 的 IAM Identity Center 支援，您需要啟用 IAM Identity Center 的使用。若要進一步了解如何執行此操作，請參閱什麼是 IAM Identity Center？

> **Note**
>
> 若要使用 IAM Identity Center 使用者或群組存取 Amazon OpenSearch Serverless 集合，您必須使用 OpenSearch UI（應用程式）功能。不支援使用 IAM Identity Center 登入資料直接存取 OpenSearch Serverless Dashboards。如需詳細資訊，請參閱 OpenSearch 使用者介面入門。

建立 IAM Identity Center 執行個體之後，客戶帳戶管理員需要為 Amazon OpenSearch Serverless 服務建立 IAM Identity Center 應用程式。這可以透過呼叫 CreateSecurityConfig 來完成。客戶帳戶管理員可以指定要用於授權請求的屬性。使用的預設屬性為 `UserId`和 `GroupId`.

Amazon OpenSearch Serverless 的 IAM Identity Center 整合使用以下 AWS IAM Identity Center (IAM) 許可：

- `aoss:CreateSecurityConfig` – 建立 IAM Identity Center 供應商
- `aoss:ListSecurityConfig` – 列出目前帳戶中的所有 IAM Identity Center 供應商。
- `aoss:GetSecurityConfig` – 檢視 IAM Identity Center 提供者資訊。
- `aoss:UpdateSecurityConfig` – 修改指定的 IAM Identity Center 組態
- `aoss:DeleteSecurityConfig` – 刪除 IAM Identity Centerprovider。

下列身分型存取政策可用於管理所有 IAM Identity Center 組態：

JSON

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Action": [
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> `Resource` 元素必須是萬用字元。

建立 IAM Identity Center 提供者 （主控台）

您可以建立 IAM Identity Center 提供者，以使用 OpenSearch 應用程式啟用身分驗證。若要啟用 OpenSearch Dashboards 的 IAM Identity Center 身分驗證，請執行下列步驟：

1. 登入 Amazon OpenSearch Service 主控台。
2. 在左側導覽面板上，展開無伺服器並選擇身分驗證。
3. 選擇 IAM Identity Center 身分驗證。
4. 選取編輯
5. 勾選使用 IAM Identity Center 驗證旁的方塊。
6. 從下拉式選單中選取使用者和群組屬性索引鍵。使用者屬性將用於根據 `UserName`、 `UserId`和授權使用者`Email`。群組屬性將用於根據 `GroupName`和 對使用者進行身分驗證`GroupId`。
7. 選取 IAM Identity Center 執行個體。
8. 選取儲存

建立 IAM Identity Center 提供者 (AWS CLI)

若要使用 AWS Command Line Interface (AWS CLI) 建立 IAM Identity Center 提供者，請使用下列命令：

```
aws opensearchserverless create-security-config \
--region us-east-2 \
--name "iamidentitycenter-config" \
--description "description" \
--type "iamidentitycenter" \
--iam-identity-center-options '{
    "instanceArn": "arn:aws:sso:::instance/ssoins-99199c99e99ee999",
    "userAttribute": "UserName",
    "groupAttribute": "GroupId"
}'
```

啟用 IAM Identity Center 之後，客戶只能修改使用者和群組屬性。

```
aws opensearchserverless update-security-config \
--region us-east-1 \
--id <id_from_list_security_configs> \
--config-version <config_version_from_get_security_config> \
--iam-identity-center-options-updates '{
    "userAttribute": "UserId",
```

```
    "groupAttribute": "GroupId"
}'
```

若要使用 檢視 IAM Identity Center 供應商 AWS Command Line Interface，請使用下列命令：

```
aws opensearchserverless list-security-configs --type iamidentitycenter
```

刪除 IAM Identity Center 供應商

IAM Identity Center 提供兩個提供者執行個體，一個用於您的組織帳戶，另一個用於您的成員帳戶。如果您需要變更 IAM Identity Center 執行個體，則需要透過 `DeleteSecurityConfig` API 刪除安全組態，並使用新的 IAM Identity Center 執行個體建立新的安全組態。下列命令可用來刪除 IAM Identity Center 提供者：

```
aws opensearchserverless delete-security-config \
--region us-east-1 \
--id <id_from_list_security_configs>
```

授予 IAM Identity Center 對集合資料的存取權

啟用 IAM Identity Center 提供者後，您可以更新集合資料存取政策，以包含 IAM Identity Center 主體。IAM Identity Center 主體需要以下列格式更新：

```
[
   {
"Rules":[
       ...
      ],
      "Principal":[
         "iamidentitycenter/<iamidentitycenter-instance-id>/user/<UserName>",
         "iamidentitycenter/<iamidentitycenter-instance-id>/group/<GroupId>"
      ]
   }
]
```

> **Note**
>
> Amazon OpenSearch Serverless 為所有客戶集合僅支援一個 IAM Identity Center 執行個體，且單一使用者最多可支援 100 個群組。如果您嘗試使用超過允許的執行個體數量，您會遇到與資料存取政策授權處理不一致的情況，並收到403錯誤訊息。

您可以授予對集合、索引或兩者的存取權。如果您希望不同的使用者具有不同的權限，則需要建立多個規則。如需可用許可的清單，請參閱 Amazon OpenSearch Service 中的 Identity and Access Management。如需如何格式化存取政策的資訊，請參閱授予 SAML 身分對集合資料的存取權。

Amazon OpenSearch Serverless 中的加密

靜態加密

您建立的每個 Amazon OpenSearch Serverless 集合的保護都是使用靜態資料加密來實現，靜態資料加密是一種安全功能，可協助防止您的資料遭到未經授權存取。靜態加密使用 AWS Key Management Service (AWS KMS) 來存放和管理加密金鑰。其會使用 256 位元金鑰的進階加密標準演算法 (AES-256) 來執行加密。

主題

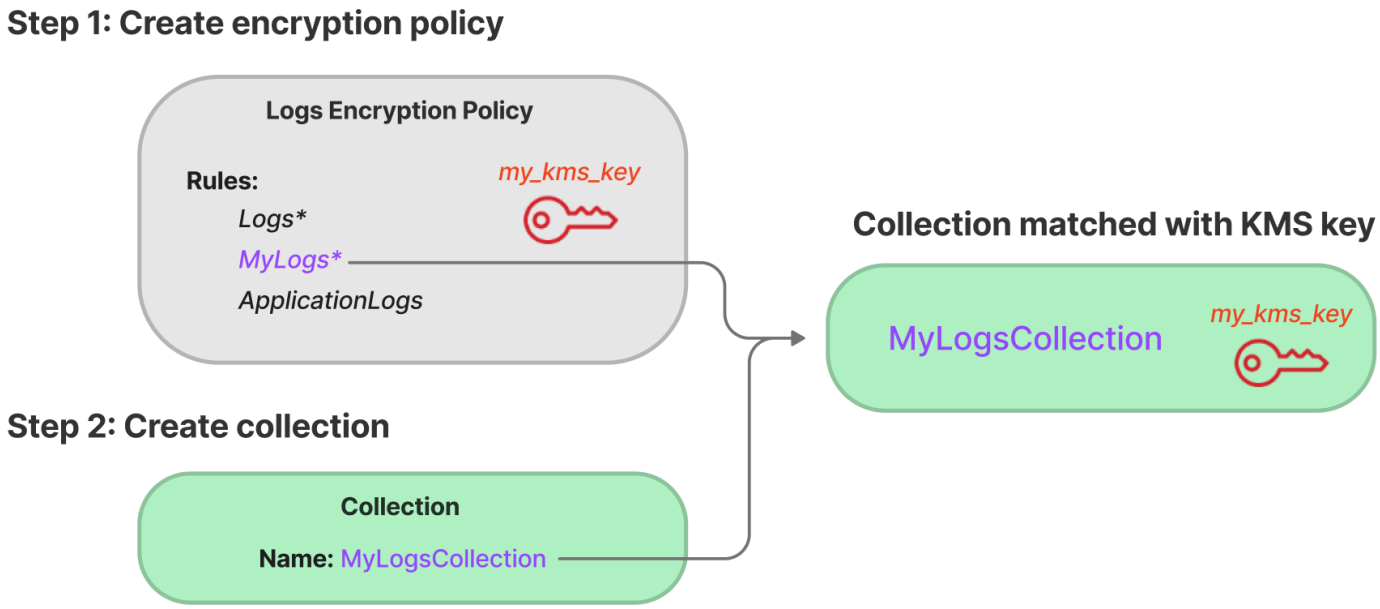
- 加密政策
- 考量事項
- 必要許可
- 客戶自管金鑰的金鑰政策
- OpenSearch Serverless 如何在 中使用授予 AWS KMS
- 建立加密政策 (主控台)
- 建立加密政策 (AWS CLI)
- 檢視加密政策
- 更新加密政策
- 刪除加密政策

加密政策

您可以使用加密政策，透過自動將加密金鑰指派給符合特定名稱或模式的新建集合，大規模管理許多集合。

建立加密政策時，您可以指定字首，這是以萬用字元為基礎的比對規則 (例如 `MyCollection*`)，或輸入單一集合名稱。然後，當您建立符合該名稱或字首模式的集合時，系統會將該政策和對應的 KMS 金鑰自動指派給該集合。

建立集合時，您可以透過兩種方式指定 AWS KMS 金鑰：透過安全政策或直接在`CreateCollection`請求中。如果您在`CreateCollection`請求中提供 AWS KMS 金鑰，則優先於任何相符的安全政策。透過此方法，您可以在需要時靈活地覆寫特定集合的政策型加密設定。



加密政策包含下列元素：

- `Rules`：一個或多個集合比對規則，每個規則都包含下列子元素：
  - `ResourceType`：目前唯一的選項是「集合」。加密政策僅套用至集合資源。
  - `Resource`：政策套用至的一個或多個集合名稱或模式 (格式為 `collection/<collection name|pattern>`)。
- `AWSOwnedKey`：是否要使用 AWS 擁有的金鑰。
- `KmsARN`：如果您將 `AWSOwnedKey` 設定為 false，請指定用於加密關聯集合之 KMS 金鑰的 Amazon Resource Name (ARN)。如果您包含此參數，OpenSearch Serverless 會忽略 `AWSOwnedKey` 參數。

下列範例政策會將客戶受管金鑰指派給任何名為 `autopartsinventory` 的未來集合，以及以「銷售」一詞開頭的集合：

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
```

```
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:kms:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

即使政策符合集合名稱，如果資源模式包含萬用字元 (*)，您也可以選擇在集合建立期間覆寫此自動指派。如果您選擇覆寫自動金鑰指派，OpenSearch Serverless 會為您建立名為 auto-<***collection-name***> 的加密政策，並將其連接到集合。該政策最初只適用於單一集合，但您可以加以修改以包含其他集合。

如果您將政策規則修改為不再符合某個集合，則系統不會將關聯的 KMS 金鑰從該集合取消指派。集合一律會使用其初始加密金鑰保持加密的狀態。如果您想要變更集合的加密金鑰，您必須重新建立集合。

如果來自多個政策的規則符合集合，則會使用更明確的規則。例如，如果某個政策包含的規則適用於 `collection/log*`，而另一個適用於 `collection/logSpecial`，則會使用第二個政策的加密金鑰，因為該金鑰更加明確。

如果某個政策中已有名稱或字首，則無法在另一個政策中使用該名稱或字首。如果您嘗試在不同的加密政策中設定相同的資源模式，則 OpenSearch Serverless 會顯示錯誤。

考量事項

設定集合的加密時應考慮以下事項：

- 所有無伺服器集合都需要靜態加密。
- 您可以選擇使用客戶受管金鑰或 AWS 擁有的金鑰。如果您選擇客戶受管金鑰，建議您啟用自動金鑰輪換。
- 建立集合之後，便無法變更集合的加密金鑰。第一次設定集合時，請仔細選擇要使用的 AWS KMS 集合。
- 集合只能符合單一加密政策。
- 具有唯一 KMS 金鑰的集合無法與其他集合共用 OpenSearch 運算單元 (OCU)。每個具有唯一金鑰的集合都需要專屬的 4 個 OCU。
- 如果您更新加密政策中的 KMS 金鑰，則變更不會影響已指派 KMS 金鑰的現有相符集合。
- OpenSearch Serverless 不會明確檢查客戶受管金鑰的使用者許可。如果使用者有權透過資料存取政策存取集合，他們就能夠擷取和查詢透過關聯金鑰加密的資料。

必要許可

OpenSearch Serverless 的靜態加密會使用下列 AWS Identity and Access Management (IAM) 許可。您可以指定 IAM 條件，將使用者限制在特定集合內。

- `aoss:CreateSecurityPolicy`：建立加密政策。
- `aoss:ListSecurityPolicies`：列出連接的所有加密政策和集合。
- `aoss:GetSecurityPolicy`：查看特定加密政策的詳細資訊。
- `aoss:UpdateSecurityPolicy`：修改加密政策。
- `aoss:DeleteSecurityPolicy`：刪除加密政策。

下列身分型存取政策範例提供使用者透過資源模式 `collection/application-logs` 管理加密政策所需的最低許可。

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "aoss:CreateSecurityPolicy",
            "aoss:UpdateSecurityPolicy",
            "aoss:DeleteSecurityPolicy",
            "aoss:GetSecurityPolicy"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aoss:collection":"application-logs"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "aoss:ListSecurityPolicies"
         ],
         "Resource":"*"
```

```
      }
   ]
}
```

客戶自管金鑰的金鑰政策

當您選取客戶受管金鑰來保護集合時，OpenSearch Serverless 會取得許可，以代表進行選取的主體使用 KMS 金鑰。該主體 (使用者或角色) 必須具備 OpenSearch Serverless 所需的 KMS 金鑰許可。您可以在金鑰政策或 IAM 政策中提供這些許可。

OpenSearch Serverless 會在維護操作期間進行 `GenerateDataKey` 和 `Decrypt` KMS API 呼叫，例如自動擴展和軟體更新。您可能會在一般流量模式之外看到這些呼叫。這些呼叫是正常服務操作的一部分，不表示作用中的使用者流量。

當 OpenSearch Serverless 無法存取加密靜態資料的 KMS 金鑰`KMSKeyInaccessibleException`時，會擲回 。當您停用或刪除 KMS 金鑰，或撤銷允許 OpenSearch Serverless 使用金鑰的授予時，就會發生這種情況。

至少，OpenSearch Serverless 需要具備客戶受管金鑰的下列許可：

- kms:DescribeKey
- kms:CreateGrant

例如：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
        "Action": "kms:DescribeKey",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Dale"
        },
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:ViaService": "aoss.us-east-1.amazonaws.com"
```

```
            }
        }
    },
    {
        "Action": "kms:CreateGrant",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Dale"
        },
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:ViaService": "aoss.us-east-1.amazonaws.com"
            },
            "ForAllValues:StringEquals": {
                "kms:GrantOperations": [
                    "Decrypt",
                    "GenerateDataKey"
                ]
            },
            "Bool": {
                "kms:GrantIsForAWSResource": "true"
            }
        }
    }
  ]
}
```

OpenSearch Serverless 會使用 kms:GenerateDataKey 和 kms:Decrypt 許可建立授予。

如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 在 AWS KMS中使用金鑰政策。

OpenSearch Serverless 如何在 中使用授予 AWS KMS

OpenSearch Serverless 需要授予才能使用客戶受管金鑰。

當您使用新金鑰在帳戶中建立加密政策時，OpenSearch Serverless 會透過傳送 CreateGrant 請求至來代表您建立授予 AWS KMS。中的授予 AWS KMS 用於授予 OpenSearch Serverless 存取客戶帳戶中 KMS 金鑰的權限。

OpenSearch Serverless 需要授予，才能在下列內部操作中使用客戶受管金鑰：

- 將 DescribeKey 請求傳送至 AWS KMS ，以驗證提供的對稱客戶受管金鑰 ID 是否有效。
- 將 GenerateDataKey 請求傳送至 KMS 金鑰，以建立加密物件所用的資料金鑰。
- 將 Decrypt 請求傳送至 AWS KMS 以解密加密的資料金鑰，以便用來加密您的資料。

您可以隨時撤銷授予的存取權，或移除服務對客戶受管金鑰的存取權。如果您這麼做，OpenSearch Serverless 將無法存取由客戶受管金鑰加密的任何資料，這會影響與該資料有關的所有操作，進而導致非同步工作流程中的 `AccessDeniedException` 錯誤和失敗。

當指定的客戶受管金鑰未與任何安全政策或集合相關聯時，OpenSearch Serverless 會淘汰非同步工作流程中的授予。

建立加密政策 (主控台)

在加密政策中，您可以指定政策套用至的 KMS 金鑰和一系列集合模式。當您建立集合時，系統會將對應的 KMS 金鑰指派給符合政策中定義模式之一的任何新集合。建議您先建立加密政策，然後再開始建立集合。

建立 OpenSearch Serverless 加密政策

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽面板中，展開 Serverless (無伺服器)，然後選擇 Encryption policies (加密政策)。
3. 選擇 Create encryption policy (建立加密政策)。
4. 提供政策的名稱和描述。
5. 在 Resources (資源) 下，輸入此加密政策的一個或多個資源模式。目前 AWS 帳戶 和區域中符合其中一種模式的任何新建集合都會自動指派給此政策。例如，如果您輸入 `ApplicationLogs` (不含萬用字元)，然後再使用該名稱建立集合，則會將該政策和對應的 KMS 金鑰指派給該集合。

   您也可以提供字首，例如 `Logs*`，此字首會將政策指派給名稱以 `Logs` 開頭的任何新集合。透過使用萬用字元，您可以大規模管理多個集合的加密設定。
6. 在 Encryption (加密) 下，選擇要使用的 KMS 金鑰。
7. 選擇建立。

下一步：建立集合

設定一個或多個加密政策後，您就可以開始建立與這些政策中定義之規則相符的集合。如需說明，請參閱the section called “建立集合”。

在集合建立的 Encryptions (加密) 步驟中，OpenSearch Serverless 會通知您所輸入的名稱與加密政策中定義的模式相符，並自動將對應的 KMS 金鑰指派給該集合。如果資源模式包含萬用字元 (*)，您可以選擇覆寫相符項目並選取自己的金鑰。

建立加密政策 (AWS CLI)

若要使用 OpenSearch Serverless API 操作建立加密政策，您需指定資源模式和 JSON 格式的加密金鑰。CreateSecurityPolicy 請求同時接受內嵌政策和 .json 檔案。

加密政策採用下列格式。此範例 `my-policy.json` 檔案符合任何名為 `autopartsinventory` 的未來集合，以及名稱以 `sales` 開頭的任何集合。

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:kms:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

若要使用服務擁有的金鑰，請將 `AWSOwnedKey` 設定為 `true`：

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":true
}
```

下列請求會建立加密政策：

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy file://my-policy.json
```

然後，使用 CreateCollection API 操作建立一個或多個符合其中一個資源模式的集合。

檢視加密政策

在建立集合之前，您可能想要預覽帳戶中現有的加密政策，以查看哪個政策的資源模式與集合名稱相符。下列 ListSecurityPolicies 請求會列出您帳戶中的所有加密政策：

```
aws opensearchserverless list-security-policies --type encryption
```

該請求會傳回所有已設定加密政策的相關資訊。使用 policy 元素的內容以檢視政策中定義的模式規則：

```
{
   "securityPolicyDetails": [
      {
         "createdDate": 1663693217826,
         "description": "Sample encryption policy",
         "lastModifiedDate": 1663693217826,
         "name": "my-policy",
         "policy": "{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource\":
[\"collection/autopartsinventory\",\"collection/sales*\"]}],\"AWSOwnedKey\":true}",
         "policyVersion": "MTY2MzY5MzIxNzgyNl8x",
         "type": "encryption"
      }
   ]
}
```

若要檢視有關特定政策 (包括 KMS 金鑰) 的詳細資訊，請使用 GetSecurityPolicy 命令。

更新加密政策

如果您更新加密政策中的 KMS 金鑰，則變更只會套用至與設定的名稱或模式相符的新建集合。這不會影響已指派 KMS 金鑰的現有集合。

這也適用於政策比對規則。如果您新增、修改或刪除規則，則變更僅適用於新建集合。如果您修改政策的規則，使其不再符合集合的名稱，則現有集合不會遺失其指派的 KMS 金鑰。

若要更新 OpenSearch Serverless 主控台中的加密政策，請選擇 Encryption policies (加密政策)，選取要修改的政策，然後選擇 Edit (編輯)。進行變更，然後選擇 Save (儲存)。

若要使用 OpenSearch Serverless API 更新加密政策，請使用 UpdateSecurityPolicy 操作。下列請求會使用新的政策 JSON 文件更新加密政策：

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy-version 2 \
    --policy file://my-new-policy.json
```

刪除加密政策

當您刪除加密政策時，目前使用政策中定義之 KMS 金鑰的任何集合都不會受到影響。若要刪除 OpenSearch Serverless 主控台中的政策，請選取該政策，然後選擇 Delete (刪除)。

您也可以使用 DeleteSecurityPolicy 操作：

```
aws opensearchserverless delete-security-policy --name my-policy --type encryption
```

傳輸中加密

在 OpenSearch Serverless 中，集合中的所有路徑都會使用 Transport Layer Security 1.2 (TLS) 搭配業界標準 AES-256 密碼在傳輸中加密。也可透過 TLS 1.2 存取 Opensearch 的所有 APIs 和儀表板。TLS 是一組業界標準的密碼編譯通訊協定，用於加密透過網路交換的資訊。

Amazon OpenSearch Serverless 的網路存取

Amazon OpenSearch Serverless 集合的網路設定會判斷該集合是否可透過網際網路從公有網路存取，或是否必須私下存取。

私有存取可以套用到下列其中一項或兩項：

- OpenSearch Serverless 受管 VPC 端點
- 支援， AWS 服務 例如 Amazon Bedrock

您可以針對集合的 OpenSearch 端點及其對應的 OpenSearch Dashboards 端點個別設定網路存取。

網路存取是允許從不同來源網路存取的隔離機制。例如，如果集合的 OpenSearch Dashboards 端點可公開存取，但 OpenSearch API 端點無法公開存取，則使用者只能在從公有網路連線時透過 Dashboards 存取集合資料。如果使用者嘗試直接從公用網路呼叫 OpenSearch API，就會遭到封鎖。網路設定可用於來源到資源類型的這類排列。Amazon OpenSearch Serverless 同時支援 IPv4 和 IPv6 連線。

主題

- 網路政策
- 考量事項
- 設定網路政策所需的許可
- 政策優先順序
- 建立網路政策 (主控台)
- 建立網路政策 (AWS CLI)
- 檢視網路政策
- 更新網路政策
- 刪除網路政策

網路政策

網路政策讓您可以自動將網路存取設定指派給符合政策中定義之規則的集合，以便大規模管理許多集合。

在網路政策中，您可以指定一系列規則。這些規則會定義集合端點和 OpenSearch Dashboards 端點的存取許可。每個規則都包含存取類型（公有或私有）和資源類型（集合和/或 OpenSearch Dashboards 端點）。對於每個資源類型 (`collection` 和 `dashboard`)，您可以指定一系列規則來定義政策套用至的集合。

在此範例政策中，第一個規則會指定 VPC 端點對以 一詞開頭之所有集合的集合端點和 Dashboards 端點的存取`marketing*`。它也會指定 Amazon Bedrock 存取。

> **Note**
>
> 對 Amazon Bedrock AWS 服務 等 的私有存取僅適用於集合的 OpenSearch 端點，不適用於 OpenSearch Dashboards 端點。即使 `ResourceType`是 `dashboard`，也 AWS 服務 無法授予 OpenSearch Dashboards 的存取權。

第二個規則會指定對 finance 集合的公用存取權，但僅限於集合端點 (無 Dashboards 存取權)。

```
[
   {
      "Description":"Marketing access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/marketing*"
            ]
         },
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/marketing*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   },
   {
      "Description":"Sales access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

此政策僅針對以「財務」開頭的集合，提供對 OpenSearch Dashboards 的公用存取權。任何直接存取 OpenSearch API 的嘗試都會失敗。

```
[
  {
    "Description": "Dashboards access",
    "Rules": [
      {
        "ResourceType": "dashboard",
        "Resource": [
          "collection/finance*"
        ]
      }
    ],
    "AllowFromPublic": true
  }
]
```

網路政策可套用至現有集合以及未來的集合。例如，您可以建立集合，然後使用符合集合名稱的規則建立網路政策。您不需要建立網路政策，就能建立集合。

考量事項

為集合設定網路存取時，請考慮以下項目：

- 如果您打算設定集合的 VPC 端點存取，您必須先建立至少一個 OpenSearch Serverless 受管 VPC 端點。
- 對 的私有存取 AWS 服務 僅適用於集合的 OpenSearch 端點，不適用於 OpenSearch Dashboards 端點。即使 `ResourceType`是 `dashboard`，也 AWS 服務 無法授予 OpenSearch Dashboards 的存取權。
- 如果集合可從公有網路存取，也可以從所有 OpenSearch Serverless 受管 VPC 端點和所有端點存取 AWS 服務。
- 多個網路政策可套用至單一集合。如需詳細資訊，請參閱the section called “政策優先順序”。

設定網路政策所需的許可

OpenSearch Serverless 的網路存取使用下列 AWS Identity and Access Management (IAM) 許可。您可以指定 IAM 條件，將使用者限制在與特定集合關聯的網路政策內。

- `aoss:CreateSecurityPolicy`：建立網路存取政策。
- `aoss:ListSecurityPolicies`：列出目前帳戶中的所有網路政策。

- aoss:GetSecurityPolicy：檢視網路存取政策規格。
- aoss:UpdateSecurityPolicy：修改指定的網路存取政策，並變更 VPC ID 或公用存取名稱。
- aoss:DeleteSecurityPolicy：刪除網路存取政策 (在將該政策從所有集合中分離之後)。

下列身分型存取政策讓使用者可以檢視所有網路政策，以及更新包含資源模式 collection/application-logs 的政策：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aoss:UpdateSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": "application-logs"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:ListSecurityPolicies",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> 此外，OpenSearch Serverless 需要集合資源的 `aoss:APIAccessAll`和 `aoss:DashboardsAccessAll`許可。如需詳細資訊，請參閱the section called “使用 OpenSearch API 操作”。

政策優先順序

在某些情況下，網路政策規則的重疊情形可能會在政策內部或跨政策發生。發生這種情況時，指定公有存取的規則會覆寫規則，為這兩個規則通用的任何集合指定私有存取。

例如，在下列政策中，這兩個規則都會將網路存取指派給 `finance` 集合，但其中一個規則指定的是 VPC 存取權，而另一個規則指定的是公用存取權。在此情況下，公用存取權僅會覆寫財務集合的 VPC 存取權 (因為其同時存在於兩個規則中)，因此您可從公用網路存取該財務集合。銷售集合將具有來自指定端點的 VPC 存取權。

```
[
   {
      "Description":"Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/sales",
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ]
   },
   {
      "Description":"Rule 2",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
```

```
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

如果來自不同規則的多個 VPC 端點套用至某個集合，則這些為附加規則，且可從所有指定端點存取該集合。如果您將 `AllowFromPublic` 設定為 `true`，但也提供一或多個 `SourceVPCEs` 或 `SourceServices`，則 OpenSearch Serverless 會忽略 VPC 端點和服務識別符，並且相關聯的集合將具有公有存取權。

建立網路政策 (主控台)

網路政策可套用至現有集合以及未來的集合。建議您先建立網路政策，然後再開始建立集合。

建立 OpenSearch Serverless 網路政策

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽面板中，展開 Serverless (無伺服器)，然後選擇 Network policies (網路政策)。
3. 選擇 Create network policy (建立網路政策)。
4. 提供政策的名稱和描述。
5. 提供一個或多個規則。這些規則會定義 OpenSearch Serverless 集合及其 OpenSearch Dashboards 端點的存取許可。

   每個規則都包含下列元素：

Element	Description
規則名稱	描述規則內容的名稱。例如，「行銷團隊的 VPC 存取權」。
存取類型	選擇公有或私有存取。然後，選取下列其中一項或兩項： • 用於存取的 VPC 端點 – 指定一或多個 OpenSearch Serverless 受管 VPC 端點受管 VPC 端點。

Element	Description
	• AWS 服務 私有存取 – 選取一個或多個支援的 AWS 服務。
Resource Type (資源類型)	選擇是提供對 OpenSearch 端點 (允許呼叫 OpenSearch API)、OpenSearch Dashboards (允許存取 OpenSearch 外掛程式的視覺效果和使用者介面) 的存取權，還是提供對兩者的存取權。 **Note** AWS 服務 私有存取僅適用於集合的 OpenSearch 端點，不適用於 OpenSearch Dashboards 端點。即使您選取 OpenSearch Dashboards，也 AWS 服務 只能授予端點存取權。

對於您選取的每個資源類型，您可以選擇要將政策設定套用至其中的現有集合，和/或建立一個或多個資源模式。資源模式包含字首和萬用字元 (*)，並定義政策設定將套用至的集合。

例如，如果您包含名為 `Marketing*` 的模式，則名稱以「行銷」開頭的任何新的或現有集合都會將此政策中的網路設定自動套用至其中。單一萬用字元 (*) 會將政策套用至所有目前和未來的集合。

此外，您可以指定未來集合的名稱，而不使用萬用字元，例如 `Finance`。OpenSearch Serverless 會將政策設定套用至任何具有相同名稱的新建集合。

6. 如果您對政策組態感到滿意，請選擇 Create (建立)。

建立網路政策 (AWS CLI)

若要使用 OpenSearch Serverless API 操作建立網路政策，您需指定 JSON 格式的規則。CreateSecurityPolicy 請求同時接受內嵌政策和 .json 檔案。所有集合和模式採用的格式必須為 `collection/<collection name|pattern>`。

> **Note**
>
> 資源類型 `dashboards` 僅允許對 OpenSearch Dashboards 的許可，但若要讓 OpenSearch Dashboards 能夠正常運作，您還必須允許來自相同來源的集合存取權。如需範例，請參閱以下第二個政策。

若要指定私有存取，請包含下列其中一個或兩個元素：

- `SourceVPCEs` – 指定一或多個 OpenSearch Serverless 受管 VPC 端點。
- `SourceServices` – 指定一個或多個支援的識別符 AWS 服務。目前支援下列服務識別符：
  - `bedrock.amazonaws.com` – Amazon Bedrock

下列範例網路政策提供對 VPC 端點和 Amazon Bedrock 的私有存取，僅針對字首為 的集合收集端點`log*`。經過驗證的使用者無法登入 OpenSearch Dashboards；他們只能以程式設計方式存取集合端點。

```
[
   {
      "Description":"Private access for log collections",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/log*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   }
]
```

下列政策會針對名為 `finance` 的單一集合，提供對 OpenSearch 端點和 OpenSearch Dashboards 的公用存取權。如果該集合不存在，建立集合時，會將網路設定套用至該集合。

```
[
   {
      "Description":"Public access for finance collection",
      "Rules":[
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/finance"
            ]
         },
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

下列請求會建立上述網路政策：

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type network \
    --policy "[{\"Description\":\"Public access for finance collection\",\"Rules
\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/finance\"]},
{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/finance\"]}],
\"AllowFromPublic\":true}]"
```

若要在 JSON 檔案中提供政策，請使用 `--policy file://my-policy.json` 格式

檢視網路政策

在建立集合之前，您可能想要預覽帳戶中的現有網路政策，以查看哪個政策的資源模式與集合名稱相符。下列 ListSecurityPolicies 請求會列出您帳戶中的所有網路政策：

```
aws opensearchserverless list-security-policies --type network
```

此請求會傳回所有已設定網路政策的相關資訊。若要檢視一個特定政策中定義的模式規則，請在回應中的 securityPolicySummaries元素內容中找到政策資訊。請注意此政策type的 name和 ，並在 GetSecurityPolicy 請求中使用這些屬性來接收包含下列政策詳細資訊的回應：

```
{
    "securityPolicyDetail": [
        {
            "type": "network",
            "name": "my-policy",
            "policyVersion": "MTY2MzY5MTY1MDA3Ml8x",
            "policy": "[{\"Description\":\"My network policy rule\",\"Rules\":
[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection/*\"]}],\"AllowFromPublic
\":true}]",
            "createdDate": 1663691650072,
            "lastModifiedDate": 1663691650072
        }
    ]
}
```

若要檢視特定政策的相關詳細資訊，請使用 GetSecurityPolicy 命令。

更新網路政策

當您修改網路的 VPC 端點或公用存取名稱時，所有關聯的集合都會受到影響。若要更新 OpenSearch Serverless 主控台中的網路政策，請展開 Network policies (網路政策)，選取要修改的政策，然後選擇 Edit (編輯)。進行變更，然後選擇 Save (儲存)。

若要使用 OpenSearch Serverless API 更新網路政策，請使用 UpdateSecurityPolicy 命令。您必須在請求中包含政策版本。您可以使用 ListSecurityPolicies 或 GetSecurityPolicy 命令擷取政策版本。將最新的政策版本納入其中，可確保您不會意外覆寫其他人所做的變更。

下列請求會使用新的政策 JSON 文件更新網路政策：

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type network \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

刪除網路政策

您必須將網路政策從所有集合分離，才能將其刪除。若要刪除 OpenSearch Serverless 主控台中的政策，請選取該政策，然後選擇 Delete (刪除)。

您也可以使用 DeleteSecurityPolicy 命令：

```
aws opensearchserverless delete-security-policy --name my-policy --type network
```

Amazon OpenSearch Serverless 中的 FIPS 合規

Amazon OpenSearch Serverless 支援聯邦資訊處理標準 (FIPS) 140-2，這是美國和加拿大政府標準，可指定保護敏感資訊之密碼編譯模組的安全需求。當您使用 OpenSearch Serverless 連線至已啟用 FIPS 的端點時，密碼編譯操作會使用 FIPS 驗證的密碼編譯程式庫進行。

OpenSearch Serverless FIPS 端點可在支援 FIPS AWS 區域 的 中使用。這些端點使用 TLS 1.2 或更新版本，以及 FIPS 驗證的密碼編譯演算法進行所有通訊。如需詳細資訊，請參閱AWS 《 驗證存取使用者指南》中的 FIPS 合規。

主題

- 搭配 OpenSearch Serverless 使用 FIPS 端點
- 使用具有 AWS SDKs FIPS 端點
- 設定 VPC 端點的安全群組
- 使用 FIPS VPC 端點
- 驗證 FIPS 合規
- 解決私有託管區域中的 FIPS 端點連線問題

搭配 OpenSearch Serverless 使用 FIPS 端點

在支援 FIPS AWS 區域 的情況下，OpenSearch Serverless 集合可透過標準和 FIPS 相容端點存取。如需詳細資訊，請參閱AWS 《 驗證存取使用者指南》中的 FIPS 合規。

在下列範例中，將 *collection_id* 和 取代*AWS ##*為您的集合 ID 及其 AWS 區域。

- 標準端點 – **https://*collection_id*.*AWS ##*.aoss.amazonaws.com**。
- FIPS 相容端點 – **https://*collection_id*.*AWS ##*.aoss-fips.amazonaws.com**。

同樣地，OpenSearch Dashboards 可透過標準和 FIPS 相容端點存取：

- 標準儀表板端點 – **https://*collection_id*.*AWS ##*.aoss.amazonaws.com/_dashboards**。
- 符合 FIPS 標準的儀表板端點 – **https://*collection_id*.*AWS ##*.aoss-fips.amazonaws.com/_dashboards**。

> **Note**
>
> 在啟用 FIPS 的區域中，標準和 FIPS 相容的端點都提供 FIPS 相容的密碼編譯。FIPS 特定的端點可協助您滿足合規要求，這些要求特別強制在名稱中使用具有 FIPS 的端點。

使用具有 AWS SDKs FIPS 端點

使用 AWS SDKs時，您可以在建立用戶端時指定 FIPS 端點。在下列範例中，將 *collection_id* 和取代*AWS ##*為您的集合 ID 及其 AWS 區域。

```
# Python SDK example
from opensearchpy import OpenSearch, RequestsHttpConnection, AWSV4SignerAuth
import boto3
host = '"https://collection_id.AWS ##.aoss-fips.amazonaws.com"
region = 'us-west-2'
service = 'aoss'
credentials = boto3.Session().get_credentials()
auth = AWSV4SignerAuth(credentials, region, service)
client = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = auth,
    use_ssl = True,
    verify_certs = True,
    connection_class = RequestsHttpConnection,
    pool_maxsize = 20
)
```

設定 VPC 端點的安全群組

為了確保與 FIPS 相容的 Amazon VPC (VPC) 端點進行適當通訊，請建立或修改安全群組，以允許來自您 VPC 中需要存取 OpenSearch Serverless 資源的傳入 HTTPS 流量 (TCP 連接埠 443)。然後在

建立期間或在建立之後修改端點，將此安全群組與您的 VPC 端點建立關聯。如需詳細資訊，請參閱《Amazon VPC 使用者指南》的建立安全群組。

使用 FIPS VPC 端點

建立 FIPS 相容 VPC 端點後，您可以使用它從 VPC 中的資源存取 OpenSearch Serverless。若要使用 API 操作的端點，請將 SDK 設定為使用區域 FIPS 端點，如 搭配 OpenSearch Serverless 使用 FIPS 端點一節中所述。對於 OpenSearch Dashboards 存取，請使用集合特定的 Dashboards URL，從 VPC 內存取時，會自動透過 FIPS 相容 VPC 端點路由。如需詳細資訊，請參閱搭配使用 OpenSearch Dashboards 與 Amazon OpenSearch Service。

驗證 FIPS 合規

若要驗證您對 OpenSearch Serverless 的連線是否使用 FIPS 相容密碼編譯，請使用 AWS CloudTrail 來監控對 OpenSearch Serverless 發出的 API 呼叫。檢查 CloudTrail 日誌中的 `eventSource` 欄位是否顯示 API `aoss-fips.amazonaws.com` 呼叫。

對於 OpenSearch Dashboards 存取，您可以使用瀏覽器開發人員工具來檢查 TLS 連線詳細資訊，並確認正在使用符合 FIPS 規範的密碼套件。

解決私有託管區域中的 FIPS 端點連線問題

FIPS 端點適用於具有公有存取權的 Amazon OpenSearch Serverless 集合。對於使用新建立的 VPC 端點的新建立 VPC 集合，FIPS 端點會如預期運作。對於其他 VPC 集合，您可能需要執行手動設定，以確保 FIPS 端點正確運作。

在 Amazon Route 53 中設定 FIPS 私有託管區域

1. 請在 https://console.aws.amazon.com/route53/ 開啟 Route 53 主控台。
2. 檢閱您的託管區域：
    a. 找出 AWS 區域 集合所在的託管區域。
    b. 驗證託管區域命名模式：
        - 非 FIPS 格式：*`region`*`.aoss.amazonaws.com`。
        - FIPS 格式：*`region`*`.aoss-fips.amazonaws.com`。
    c. 確認所有託管區域的類型設定為私有託管區域。
3. 如果缺少 FIPS 私有託管區域：
    a. 選取對應的非 FIPS 私有託管區域。

b. 複製相關聯的 VPCs資訊。例如：`vpc-1234567890abcdef0 | us-east-2`。

c. 尋找萬用字元網域記錄。例如：`*.us-east-2.aoss.amazonaws.com`。

d. 將值/路由流量複製到資訊。例如：`uoc1c1qsw7poexampleewjeno1pte3rw.3ym756xh7yj.aoss.searchservices.aws`。

4. 建立 FIPS 私有託管區域：

a. 使用 FIPS 格式建立新的私有託管區域。例如：`us-east-2.aoss-fips.amazonaws.com`。

b. 針對關聯的 VPCs，輸入您從非 FIPS 私有託管區域複製的 VPC 資訊。

5. 使用下列設定新增記錄：

a. 記錄名稱：*

b. 記錄類型：CNAME

c. 值：輸入您先前複製資訊的值/路由流量。

常見問題

如果您遇到與 FIPS 相容之 VPC 端點的連線問題，請使用下列資訊來協助解決問題。

- DNS 解析失敗 - 您無法解析 VPC 中的 FIPS 端點網域名稱
- 連線逾時 - 您對 FIPS 端點的請求逾時
- 存取遭拒錯誤 - 使用 FIPS 端點時身分驗證或授權失敗
- 缺少僅限 VPC 集合的私有託管區域記錄

疑難排解 FIPS 端點連線

1. 驗證您的私有託管區域組態：

a. 確認 FIPS 端點網域 (`*.region.aoss-fips.amazonaws.com`.

b. 確認私有託管區域與正確的 VPC 相關聯。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的私有託管區域，以及《 AWS PrivateLink 指南》中的管理 DNS 名稱。

2. 測試 DNS 解析：

a. 連線至 VPC 中的 EC2 執行個體。

b. 執行以下命令：

```
nslookup collection-id.region.aoss-fips.amazonaws.com
```

c. 確認回應包含 VPC 端點的私有 IP 地址。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的端點政策和 DNS 屬性。

3. 檢查您的安全群組設定：

a. 確認連接至 VPC 端點的安全群組允許來自 資源的 HTTPS 流量 （連接埠 443)。

b. 確認 資源的安全群組允許傳出流量到 VPC 端點。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的端點政策，以及《Amazon VPC 使用者指南》中的安全群組。

4. 檢閱您的網路 ACL 組態：

- 確認網路 ACLs資源與 VPC 端點之間的流量。

如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的網路 ACL。

5. 檢閱您的端點政策：

- 檢查 VPC 端點政策是否允許 OpenSearch Serverless 資源上的必要動作。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的所需 VPC 端點許可和端點政策。

> **Tip**
>
> 如果您在 VPC 中使用自訂 DNS 解析程式，請將它們設定為將`*.amazonaws.com`網域的請求轉送至 AWS 伺服器。

Amazon OpenSearch Serverless 的資料存取控制

透過 Amazon OpenSearch Serverless 中的資料存取控制，無論使用者的存取機制或網路來源為何，您都可讓使用者存取集合和索引。您可以提供 IAM 角色和 SAML 身分的存取權。

您可以透過資料存取政策管理許可，這些政策套用至集合和索引資源。資料存取政策會自動將存取許可指派給符合特定模式的集合和索引，以協助您大規模管理集合。可將多個資料存取政策套用至單一資源。請注意，您必須擁有集合的資料存取政策，才能存取 OpenSearch Dashboards URL。

主題

- 資料存取政策與 IAM 政策比較
- 設定資料存取政策所需的 IAM 許可
- 政策語法
- 支援的政策許可
- OpenSearch Dashboards 上的範例資料集
- 建立資料存取政策 (主控台)
- 建立資料存取政策 (AWS CLI)
- 檢視資料存取政策
- 更新資料存取政策
- 刪除資料存取政策
- 跨帳戶資料存取

資料存取政策與 IAM 政策比較

資料存取政策在邏輯上與 AWS Identity and Access Management (IAM) 政策分開。IAM 許可會控制對無伺服器 API 操作 (例如 `CreateCollection` 和 `ListAccessPolicies`) 的存取。資料存取政策會控制 OpenSearch Serverless 支援之 OpenSearch 操作 (例如 `PUT <index>` 或 `GET _cat/indices`) 的存取。

控制資料存取政策 API 操作 (例如 `aoss:CreateAccessPolicy` 和 `aoss:GetAccessPolicy`，請參閱下一節所述) 存取的 IAM 許可不會影響資料存取政策中指定的許可。

例如，假設 IAM 政策拒絕使用者為 `collection-a` 建立資料存取政策，但允許其為所有集合 (*) 建立資料存取政策：

JSON

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

```
            "Effect": "Deny",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "collection-a"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

如果使用者建立的資料存取政策允許所有集合 (`collection/*` 或 `index/*/*`) 的特定許可，則該政策將套用至所有集合，包括集合 A。

> **Important**
>
> 在資料存取政策中授予許可不足以存取 OpenSearch Serverless 集合中的資料。也必須將 IAM 許可`aoss:APIAccessAll`和 的存取權授予相關聯的委託人`aoss:DashboardsAccessAll`。這兩個許可都授予集合資源的完整存取權，而 Dashboards 許可也提供 OpenSearch Dashboards 的存取權。如果委託人沒有這兩個 IAM 許可，則在嘗試將請求傳送至集合時，會收到 403 個錯誤。如需詳細資訊，請參閱the section called “使用 OpenSearch API 操作”。

設定資料存取政策所需的 IAM 許可

OpenSearch Serverless 的資料存取控制使用下列 IAM 許可。您可以指定 IAM 條件，將使用者限制為特定存取政策名稱。

- `aoss:CreateAccessPolicy`：建立存取政策。

- aoss:ListAccessPolicies：列出所有存取政策。
- aoss:GetAccessPolicy：請參閱有關特定存取政策的詳細資訊。
- aoss:UpdateAccessPolicy：修改存取政策。
- aoss:DeleteAccessPolicy：刪除存取政策。

下列身分型存取政策讓使用者可以檢視所有存取政策，以及更新包含資源模式 collection/logs 的政策。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:GetAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "aoss:UpdateAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": [
                        "logs"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> 此外，OpenSearch Serverless 需要集合資源的 `aoss:APIAccessAll`和 `aoss:DashboardsAccessAll`許可。如需詳細資訊，請參閱the section called “使用 OpenSearch API 操作”。

政策語法

資料存取政策包含一組規則，每個規則均包含下列元素：

Element	Description
`ResourceType`	許可套用的資源類型 (集合或索引)。別名和範本許可位於集合層級，而建立、修改和搜尋資料的許可則位於索引層級。如需詳細資訊，請參閱 Supported policy permissions (支援的政策許可)。
`Resource`	資源名稱和/或模式的清單。模式是字首後接萬用字元 (*)，允許關聯的許可套用至多個資源。 • 集合採用格式 `collection/ <name|pattern>` 。 • 索引採用格式 `index/<collection-name|pattern>  /<index-name|pattern/>` 。
`Permission`	為指定資源授予的許可清單。(如需許可與其允許之 API 操作的完整清單，請參閱 the section called “支援的 OpenSearch API 操作與許可”)。
`Principal`	要授予存取權的一個或多個主體清單。主體可以是 IAM 角色 ARN，也可以是 SAML 身分。這些主體必須在目前的 AWS 帳戶內。資料存取政策不直接支援跨帳戶存取，但您可以在政策中包含角色，而來自不同 的使用者 AWS 帳戶 可以在集合擁有帳戶中擔任該角色。如需詳細資訊，請參閱the section called “跨帳戶資料存取”。

下列範例政策會將別名和範本許可授予名為 `autopartsinventory` 的集合以及任何以字首 `sales*` 開頭的集合。同時也會將讀取和寫入許可授予 `autopartsinventory` 集合中的所有索引，以及以字首 `orders*` 開頭的 `salesorders` 集合中的任何索引。

```
[
```

```
   {
      "Description": "Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/autopartsinventory",
               "collection/sales*"
            ],
            "Permission":[
               "aoss:CreateCollectionItems",
               "aoss:UpdateCollectionItems",
               "aoss:DescribeCollectionItems"
            ]
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/autopartsinventory/*",
               "index/salesorders/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/Dale",
         "arn:aws:iam::123456789012:role/RegulatoryCompliance",
         "saml/123456789012/myprovider/user/Annie",
         "saml/123456789012/anotherprovider/group/Accounting"
      ]
   }
]
```

您無法在政策中明確拒絕存取。因此，所有政策許可均可附加。例如，若一個政策授予使用者 aoss:ReadDocument，而另一個政策授予 aoss:WriteDocument，則使用者將同時擁有兩個許可。如果第三個政策授予相同的使用者 aoss:*，則使用者可以對關聯的索引執行所有動作；限制較多的許可不會覆寫限制較少的許可。

支援的政策許可

資料存取政策支援下列許可。如需了解每個許可允許的 OpenSearch API 操作，請參閱 the section called “支援的 OpenSearch API 操作與許可”。

集合許可

- `aoss:CreateCollectionItems`
- `aoss:DeleteCollectionItems`
- `aoss:UpdateCollectionItems`
- `aoss:DescribeCollectionItems`
- `aoss:*`

索引許可

- `aoss:ReadDocument`
- `aoss:WriteDocument`
- `aoss:CreateIndex`
- `aoss:DeleteIndex`
- `aoss:UpdateIndex`
- `aoss:DescribeIndex`
- `aoss:*`

OpenSearch Dashboards 上的範例資料集

OpenSearch Dashboards 提供範例資料集，其中包含視覺化、儀表板和其他工具，協助您在新增自己的資料之前探索 Dashboards。若要從此範例資料建立索引，您需要一個資料存取政策，以提供您要使用之資料集的許可。下列政策使用萬用字元 (*) 為所有三個範例資料集提供許可。

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/<collection-name>/opensearch_dashboards_sample_data_*"
```

```
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:DescribeIndex",
          "aoss:ReadDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::<account-id>:user/<user>"
    ]
  }
]
```

建立資料存取政策 (主控台)

您可以使用視覺化編輯器，或以 JSON 格式建立資料存取政策。當您建立集合時，系統會將對應的許可指派給符合政策中定義模式之一的任何新集合。

建立 OpenSearch Serverless 資料存取政策

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，展開無伺服器，然後在安全性下選擇資料存取政策。
3. 選擇 Create access policy (建立存取政策)。
4. 提供政策的名稱和描述。
5. 為政策中的第一個規則提供名稱。例如，"Logs collection access" (日誌集合存取)。
6. 選擇 Add principals (新增主體)，然後選取要向其提供資料存取權的一個或多個 IAM 角色或 SAML 使用者和群組。

   > **Note**
   >
   > 若要從下拉式選單中選取主體，您必須具有 `iam:ListUsers` 和 `iam:ListRoles` 許可 (對於 IAM 主體) 和 `aoss:ListSecurityConfigs` 許可 (對於 SAML 身分)。

7. 選擇 Grant (授予)，然後選取別名、範本和索引許可，以授予關聯的主體。如需完整的許可及其允許之存取的完整清單，請參閱 the section called “支援的 OpenSearch API 操作與許可”。
8. (選用) 設定政策的其他規則。

9. 選擇建立。在您建立政策與強制執行許可之間，可能會有大約一分鐘的延遲時間。如果延遲超過 5 分鐘，請聯絡 支援。

> **Important**
>
> 如果您的政策僅包含索引許可 (而且沒有集合許可)，您可能仍會看到相符集合的訊息，表示 `Collection cannot be accessed yet. Configure data access policies so that users can access the data within this collection`。您可以忽略此警告。允許的主體仍然可以對集合執行其指派的索引相關操作。

建立資料存取政策 (AWS CLI)

若要使用 OpenSearch Serverless API 建立資料存取政策，請使用 `CreateAccessPolicy` 命令。該命令接受內嵌政策和 .json 檔案。內嵌政策必須編碼為 JSON 逸出字串。

下列請求會建立資料存取政策：

```
aws opensearchserverless create-access-policy \
    --name marketing \
    --type data \
    --policy "[{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource\":
[\"collection/autopartsinventory\",\"collection/sales*\"],\"Permission\":
[\"aoss:UpdateCollectionItems\"]},{\"ResourceType\":\"index\",\"Resource\":
[\"index/autopartsinventory/*\",\"index/salesorders/orders*\"],\"Permission
\":[\"aoss:ReadDocument\",\"aoss:DescribeIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:user/Shaheen\"]}]"
```

若要在 .json 檔案中提供政策，請使用格式 `--policy file://my-policy.json`。

包含在政策中的主體現在可以使用其獲得存取權的 OpenSearch 操作。

檢視資料存取政策

在建立集合之前，您可能想要預覽帳戶中的現有資料存取政策，以查看哪個政策的資源模式與集合名稱相符。下列 ListAccessPolicies 請求會列出您帳戶中的所有資料存取政策：

```
aws opensearchserverless list-access-policies --type data
```

該請求會傳回所有已設定資料存取政策的相關資訊。若要檢視一個特定政策中定義的模式規則，請在回應中的 accessPolicySummaries元素內容中找到政策資訊。請注意此政策type的 name和 ，並在 GetAccessPolicy 請求中使用這些屬性來接收包含下列政策詳細資訊的回應：

```
{
    "accessPolicyDetails": [
        {
            "type": "data",
            "name": "my-policy",
            "policyVersion": "MTY2NDA1NDE4MDg1OF8x",
            "description": "My policy",
            "policy": "[{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection/autopartsinventory\",\"collection/sales*\"],
\"Permission\":[\"aoss:UpdateCollectionItems\"]},{\"ResourceType\":\"index\",
\"Resource\":[\"index/autopartsinventory/*\",\"index/salesorders/orders*\"],
\"Permission\":[\"aoss:ReadDocument\",\"aoss:DescribeIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:user/Shaheen\"]}]",
            "createdDate": 1664054180858,
            "lastModifiedDate": 1664054180858
        }
    ]
}
```

您可以包含資源篩選條件，將結果限制為包含特定集合或索引的政策：

```
aws opensearchserverless list-access-policies --type data --resource
 "index/autopartsinventory/*"
```

若要檢視有關特定政策的詳細資訊，請使用 GetAccessPolicy 命令。

更新資料存取政策

當您更新資料存取政策時，所有關聯的集合均會受到影響。若要在 OpenSearch Serverless 主控台中更新資料存取政策，請選擇 Data access control (資料存取控制)，選取要修改的政策，然後選擇 Edit (編輯)。進行變更，然後選擇 Save (儲存)。

若要使用 OpenSearch Serverless API 更新資料存取政策，請傳送 UpdateAccessPolicy 請求。您必須包含政策版本，您可以使用 ListAccessPolicies 或 GetAccessPolicy 命令擷取該版本。將最新的政策版本納入其中，可確保您不會意外覆寫其他人所做的變更。

下列 UpdateAccessPolicy 請求會使用新的政策 JSON 文件更新資料存取政策：

```
aws opensearchserverless update-access-policy \
    --name sales-inventory \
    --type data \
    --policy-version MTY2NDA1NDE4MDg1OF8x \
    --policy file://my-new-policy.json
```

在您更新政策與強制執行新許可之間，可能會有幾分鐘的延遲時間。

刪除資料存取政策

當您刪除資料存取政策時，所有關聯的集合均會失去政策中定義的存取權。在刪除政策之前，請確保您的 IAM 和 SAML 使用者具有集合的適當存取權。若要刪除 OpenSearch Serverless 主控台中的政策，請選取該政策，然後選擇 Delete (刪除)。

您也可以使用 DeleteAccessPolicy 命令：

```
aws opensearchserverless delete-access-policy --name my-policy --type data
```

跨帳戶資料存取

雖然您無法建立具有跨帳戶身分或跨帳戶集合的資料存取政策，但您仍然可以使用擔任角色選項來設定跨帳戶存取。例如，如果 *account-a*擁有*account-b*需要存取的集合，則來自 的使用者可以在 中*account-b*擔任角色*account-a*。角色必須具有 IAM 許可`aoss:APIAccessAll`和`aoss:DashboardsAccessAll`，並包含在 上的資料存取政策中*account-a*。

透過 的資料平面存取 AWS PrivateLink

Amazon OpenSearch Serverless 支援控制平面和資料平面操作的兩種 AWS PrivateLink 連線類型。控制平面操作包括建立和刪除集合，以及管理存取政策。資料平面操作用於編製索引和查詢集合中的資料。此頁面涵蓋資料平面 VPC 端點。如需控制平面 AWS PrivateLink 端點的資訊，請參閱 the section called “控制平面 VPC 端點”。

您可以使用 在 VPC 和 Amazon OpenSearch Serverless 之間 AWS PrivateLink 建立私有連線。您可以像在 VPC 中一樣存取 OpenSearch Serverless，無需使用網際網路閘道、NAT 裝置、VPN 連接或 Direct Connect 連線。VPC 中的執行個體無需公有 IP 地址，即可存取 OpenSearch Serverless。如需 VPC 網路存取的詳細資訊，請參閱 Amazon OpenSearch Serverless 的網路連線模式。

您可以建立由 AWS PrivateLink提供支援的介面端點來建立此私有連線。我們會在您為介面端點指定的每個子網路中建立端點網路介面。這些是請求者受管網路介面，可作為目的地為 OpenSearch Serverless 之流量的進入點。

如需詳細資訊，請參閱《AWS PrivateLink 指南》中的「透過 AWS PrivateLink存取 AWS 服務」。

主題

- 集合端點的 DNS 解析
- VPCs和網路存取政策
- VPCs和端點政策
- 考量事項
- 必要許可
- 建立 OpenSearch Serverless 的介面端點
- Amazon OpenSearch Serverless 的共用 VPC 設定

集合端點的 DNS 解析

當您透過 OpenSearch Serverless 主控台建立資料平面 VPC 端點時，服務會建立新的 Amazon Route 53 私有託管區域，並將其連接到 VPC。此私有託管區域包含記錄，可將 OpenSearch Serverless 集合 (`*.us-east-1.aoss.amazonaws.com`) 的萬用字元 DNS 記錄解析為用於端點的介面地址。您只需要 VPC 中的一個 OpenSearch Serverless VPC 端點，即可存取每個端點中的任何和所有集合和儀表板 AWS 區域。每個具有 OpenSearch Serverless 端點的 VPC 都連接了自己的私有託管區域。

OpenSearch Serverless 介面端點也會為區域中所有集合建立公有 Route 53 萬用字元 DNS 記錄。DNS 名稱解析為 OpenSearch Serverless 公有 IP 地址。VPCs中沒有 OpenSearch Serverless VPC 端點的用戶端或公有網路中的用戶端可以使用公有 Route 53 解析程式，並使用這些 IP 地址存取集合和儀表板。VPC 端點的 IP 地址類型 (IPv4、IPv6 或 Dualstack) 取決於您為 OpenSearch Serverless 建立介面端點時提供的子網路。

> **Note**
>
> OpenSearch Serverless 會為 OpenSearch Service 網域解析建立附加的 Amazon Route 53 私有託管區域 (`` `<region>.opensearch.amazonaws.com` ``)。您可以使用 中的 update-vpc-endpoint 命令，將現有的 IPv4 VPC 端點更新為 Dualstack AWS CLI。

指定 VPC 的 DNS 解析程式地址是 VPC CIDR 的第二個 IP 地址。VPC 中的任何用戶端都需要使用該解析程式來取得任何集合的 VPC 端點地址。解析程式使用 OpenSearch Serverless 建立的私有託管區

域。對任何帳戶中的所有集合使用該解析程式就已足夠。也可以將 VPC 解析程式用於某些集合端點，將公有解析程式用於其他端點，但通常不需要。

VPCs和網路存取政策

若要為您的集合授予 OpenSearch APIs和 Dashboards 的網路許可，您可以使用 OpenSearch Serverless 網路存取政策。您可以從 VPC 端點（或公有網際網路）控制此網路存取。由於您的網路政策僅控制流量許可，因此您還必須設定資料存取政策，指定在集合及其索引中的資料上操作的許可。將 OpenSearch Serverless VPC 端點視為服務的存取點、將網路存取政策視為集合和儀表板的網路層級存取點，並將資料存取政策視為集合中任何資料操作的精細存取控制存取點。

由於您可以在網路政策中指定多個 VPC IDs，我們建議您為每個需要存取集合的 VPC 建立 VPC 端點。這些 VPCs可以屬於與擁有 OpenSearch Serverless 集合和網路政策的帳戶不同的 AWS 帳戶。我們不建議您在兩個帳戶之間建立 VPC-to-VPC對 VPC 對等互連或其他代理解決方案，以便一個帳戶的 VPC 可以使用另一個帳戶的 VPC 端點。這比每個 VPC 都有自己的端點更不安全且更具成本效益。另一個 VPC 的管理員在網路政策中設定該 VPC 端點的存取權，將無法輕鬆看見第一個 VPC。

VPCs和端點政策

Amazon OpenSearch Serverless 支援 VPCs端點政策。端點政策是您連接到 VPC 端點的 IAM 資源型政策，以控制哪些 AWS 主體可以使用端點存取您的 AWS 服務。如需詳細資訊，請參閱使用端點政策控制對 VPC 端點的存取。

若要使用端點政策，您必須先建立介面端點。您可以使用 OpenSearch Serverless 主控台或 OpenSearch Serverless API 建立介面端點。建立介面端點之後，您需要將端點政策新增至端點。如需詳細資訊，請參閱the section called “建立介面端點”。

> **Note**
>
> 您無法直接在 OpenSearch Service 主控台中定義端點政策。

端點政策不會覆寫或取代您可能已設定的其他身分型政策、資源型政策、網路政策或資料存取政策。如需更新端點政策的詳細資訊，請參閱使用端點政策控制對 VPC 端點的存取。

根據預設，端點政策會授予 VPC 端點的完整存取權。

```
{
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

雖然預設 VPC 端點政策會授予完整端點存取權，但您可以設定 VPC 端點政策，以允許存取特定角色和使用者。若要執行此作業，請參閱下列範例：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012",
                    "987654321098"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

您可以指定要包含在 VPC 端點政策中的 OpenSearch Serverless 集合做為條件元素。若要執行此作業，請參閱下列範例：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
```

```
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": [
                        "coll-abc"
                    ]
                }
            }
        }
    ]
}
```

支援 aoss:CollectionId 。

```
Condition": {
         "StringEquals": {
               "aoss:CollectionId": "collection-id"
          }
}
```

您可以在 VPC 端點政策中使用 SAML 身分來判斷 VPC 端點存取。您必須在 VPC (*) 端點政策的主體區段中使用萬用字元。若要執行此作業，請參閱下列範例：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "saml:cn": [
                        "saml/111122223333/idp123/group/football",
                        "saml/111122223333/idp123/group/soccer",
                        "saml/111122223333/idp123/group/cricket"
                    ]
```

```
                }
            }
        }
    ]
}
```

此外，您可以設定端點政策以包含特定的 SAML 主體政策。若要執行此作業，請參閱下列內容：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalTag/Department": [
                        "Engineering"]
                    }
                }
            }
        ]
    }
```

如需搭配 Amazon OpenSearch Serverless 使用 SAML 身分驗證的詳細資訊，請參閱 Amazon OpenSearch Serverless 的 SAML 身分驗證。

您也可以在相同的 VPC 端點政策中包含 IAM 和 SAML 使用者。若要執行此作業，請參閱下列範例：

JSON

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

```
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "saml:cn": [
                        "saml/111122223333/idp123/group/football",
                        "saml/111122223333/idp123/group/soccer",
                        "saml/111122223333/idp123/group/cricket"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

您也可以透過界面 VPC 端點從 Amazon EC2 存取 Amazon OpenSearch Serverless 集合。如需詳細資訊，請參閱從 Amazon EC2 存取 OpenSearch Serverless 集合（透過界面 VPC 端點）。

考量事項

設定 OpenSearch Serverless 的介面端點前，請考量下列事項：

- OpenSearch Serverless 支援透過介面端點呼叫所有支援的 OpenSearch API 操作 (非組態 API 操作)。
- 建立 OpenSearch Serverless 的介面端點後，您仍需要將其包含在網路存取政策中，才能存取無伺服器集合。
- 依預設，允許透過端點介面對 OpenSearch Serverless 進行完整存取。您可以將安全群組與端點網路介面建立關聯，以透過介面端點控制傳送至 OpenSearch Serverless 的流量。

- 單一 最多可 AWS 帳戶 有 50 個 OpenSearch Serverless VPC 端點。
- 如果您在網路政策中啟用對集合 API 或儀表板的公有網際網路存取，則任何 VPC 和公有網際網路都可以存取集合。
- 如果您是內部部署和 VPC 外部，則無法直接使用 DNS 解析程式進行 OpenSearch Serverless VPC 端點解析。如果您需要 VPN 存取，VPC 需要 DNS 代理解析程式，外部用戶端才能使用。Route 53 提供傳入端點選項，可讓您從內部部署網路或其他 VPC 將 DNS 查詢解析至 VPC。
- OpenSearch Serverless 建立並連接至 VPC 的私有託管區域由 服務管理，但會顯示在您的 Amazon Route 53 資源中，並向您的 帳戶收費。
- 如需其他考量，請參閱《AWS PrivateLink 指南》中的考量事項。

必要許可

OpenSearch Serverless 的 VPC 存取使用下列 AWS Identity and Access Management (IAM) 許可。您可以指定 IAM 條件，將使用者限制在特定集合內。

- `aoss:CreateVpcEndpoint`：建立 VPC 端點。
- `aoss:ListVpcEndpoints`：列出所有 VPC 端點。
- `aoss:BatchGetVpcEndpoint`：查看有關 VPC 端點子集的詳細資訊。
- `aoss:UpdateVpcEndpoint`：修改 VPC 端點。
- `aoss:DeleteVpcEndpoint`：刪除 VPC 端點。

此外，您需要下列 Amazon EC2 和 Route 53 許可，才能建立 VPC 端點。

- `ec2:CreateTags`
- `ec2:CreateVpcEndpoint`
- `ec2:DeleteVpcEndPoints`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeVpcs`
- `ec2:ModifyVpcEndPoint`
- `route53:AssociateVPCWithHostedZone`

- route53:ChangeResourceRecordSets
- route53:CreateHostedZone
- route53:DeleteHostedZone
- route53:GetChange
- route53:GetHostedZone
- route53:ListHostedZonesByName
- route53:ListHostedZonesByVPC
- route53:ListResourceRecordSets

建立 OpenSearch Serverless 的介面端點

您可以使用主控台或 OpenSearch Serverless API 來建立 OpenSearch Serverless 的介面端點。

建立 OpenSearch Serverless 集合的介面端點

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，展開 Serverless (無伺服器)，然後選擇 VPC endpoints (VPC 端點)。
3. 選擇 Create VPC endpoint (建立 VPC 端點)。
4. 提供端點的名稱。
5. 針對 VPC，選取您將從中存取 OpenSearch Serverless 的 VPC。
6. 針對子網路，選取您將從中存取 OpenSearch Serverless 的一個子網路。
   - 端點的 IP 地址和 DNS 類型是以子網路類型為基礎
     - Dualstack：如果所有子網路同時具有 IPv4 和 IPv6 地址範圍
     - IPv6：如果所有子網路都是僅限 IPv6 的子網路
     - IPv4：如果所有子網路都有 IPv4 地址範圍
7. 對於 Security group (安全群組)，選取要與端點網路介面建立關聯的安全群組。這是一個關鍵步驟，您可以在其中限制您授權至端點之傳入流量的連接埠、通訊協定和來源。請確定安全群組規則允許將使用 VPC 端點與 OpenSearch Serverless 通訊的資源，以便與端點網路介面通訊。
8. 選擇建立端點。

若要使用 OpenSearch Serverless API 建立 VPC 端點，請使用 CreateVpcEndpoint 命令。

> **Note**
>
> 建立端點後，請記下其 ID (例如 `vpce-abc123def4EXAMPLE`)。若要提供端點存取權給您的集合，您必須在一個或多個網路存取政策中包含此 ID。

建立介面端點後，您必須透過網路存取政策，為其提供集合的存取權。如需詳細資訊，請參閱the section called “網路存取”。

Amazon OpenSearch Serverless 的共用 VPC 設定

您可以使用 Amazon Virtual Private Cloud (VPC) AWS 帳戶 與組織中的其他 共用 VPC 子網路，以及在多個資源之間共用聯網基礎設施，例如 VPN AWS 帳戶。

目前，Amazon OpenSearch Serverless 不支援在共用 VPC 中建立 AWS PrivateLink 連線，除非您是該 VPC 的擁有者。 AWS PrivateLink 也不支援在兩者之間共用連線 AWS 帳戶。

不過，根據 OpenSearch Serverless 的彈性和模組化架構，您仍然可以設定共用 VPC。這是因為 OpenSearch Serverless 網路基礎設施與個別集合 (OpenSearch Service) 基礎設施的基礎設施不同。因此，您可以為 AWS PrivateLink VPCe 所在的一個帳戶建立 VPCe 端點，然後在其他帳戶的網路政策中使用 VPCe ID 來限制來自該共用 VPC 的流量。

下列程序是指擁有者帳戶和消費者帳戶。

擁有者帳戶充當常見的聯網帳戶，您可以在其中設定 VPC 並與其他帳戶共用。消費者帳戶是那些在擁有者帳戶與其共用的 VPC 中建立和維護其 OpenSearch Serverless 集合的帳戶。

先決條件

在設定共用 VPC 之前，請確定符合下列要求：

- 預期的擁有者帳戶必須已在 Amazon Virtual Private Cloud 中設定 VPC、子網路、路由表和其他必要資源。如需詳細資訊，請參閱《Amazon VPC 使用者指南》。
- 預期的擁有者帳戶和取用者帳戶必須屬於其中的相同組織 AWS Organizations。如需詳細資訊，請參閱「AWS Organizations 使用者指南」。

在擁有者帳戶/常用聯網帳戶中設定共用 VPC。

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。

2. 請遵循 the section called “建立介面端點” 中的步驟。執行此作業時，請進行下列選擇：

   - 選取與您組織中消費者帳戶共用的 VPC 和子網路。

3. 建立端點之後，請記下產生的 VPCe ID，並將其提供給要在消費者帳戶中執行設定任務的管理員。

   VPCe IDs的格式為 `vpce-abc123def4EXAMPLE`。

在消費者帳戶中設定共用 VPC

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 如果您還沒有集合，請使用 中的資訊the section called “管理集合”來建立集合。
3. 使用 中的資訊來the section called “建立網路政策 (主控台)”建立網路政策。執行此作業時，請進行下列選擇。

   > **Note**
   >
   > 您也可以為此目的更新現有的網路政策。

   a. 針對存取類型，選取 VPC （建議）。

   b. 對於要存取的 VPC 端點，請選擇擁有者帳戶提供給您的 VPCe ID，格式為 `vpce-abc123def4EXAMPLE`。

   c. 在資源類型區域中，執行下列動作：

      - 選取啟用對 OpenSearch 端點的存取方塊，然後選取要用來從該共用 VPC 啟用存取的集合名稱或集合模式。
      - 選取啟用對 OpenSearch Dashboard 的存取方塊，然後選取要用來從該共用 VPC 啟用存取的集合名稱或集合模式。

4. 針對新政策，選擇建立。針對現有的政策，選擇更新。

透過 控制平面存取 AWS PrivateLink

Amazon OpenSearch Serverless 支援控制平面和資料平面操作的兩種 AWS PrivateLink 連線類型。控制平面操作包括建立和刪除集合，以及管理存取政策。資料平面操作用於編製索引和查詢集合中的資

料。此頁面涵蓋控制平面 AWS PrivateLink 端點。如需資料平面 VPC 端點的資訊，請參閱 the section called “資料平面 VPC 端點”。

建立控制平面 AWS PrivateLink 端點

您可以將 OpenSearch Serverless 設定為使用介面 VPC 端點，以改善 VPC 的安全狀態。介面端點採用 AWS PrivateLink 技術。此技術可讓您在沒有網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線的情況下，私下存取 OpenSearch Serverless APIs。

如需 AWS PrivateLink 和 VPC 端點的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 端點。

考量事項

- VPC 端點僅在相同區域內受支援。
- 透過 Amazon Route 53，VPC 端點僅支援 Amazon 提供的 DNS。
- VPC 端點支援端點政策，以控制對 OpenSearch Serverless 集合、政策和 VpcEndpoints 的存取。
- OpenSearch Serverless 僅支援界面端點。不支援閘道端點。

建立 VPC 端點

若要為 Amazon OpenSearch Serverless 建立控制平面 VPC 端點，請使用《Amazon VPC 開發人員指南》中的使用介面 VPC 端點程序存取 AWS 服務。建立下列端點：

- `com.amazonaws.region.aoss`

使用主控台建立控制平面 VPC 端點

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中選擇 Endpoints (端點)。
3. 選擇建立端點。
4. 對於 Service category (服務類別)，選擇 AWS 服務。
5. 針對服務，選擇 `com.amazonaws.region.aoss`。例如 `com.amazonaws.us-east-1.aoss`。
6. 針對 VPC，選擇要在其中建立端點的 VPC。
7. 對於 Subnets (子網路)，請選擇欲建立端點的子網路 (可用區域)。
8. 針對安全群組，選擇要與端點網路介面建立關聯的安全群組。確保允許 HTTPS （連接埠 443)。

9. 針對政策，選擇完整存取以允許所有操作，或選擇自訂以連接自訂政策。
10. 選擇建立端點。

建立端點政策

您可以將端點政策連接至 VPC 端點，以控制對 Amazon OpenSearch Serverless 的存取。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的使用 VPC 端點控制對服務的存取。

Example OpenSearch Serverless 的 VPC 端點政策

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "aoss:ListCollections",
        "aoss:BatchGetCollection"
      ],
      "Resource": "*"
    }
  ]
}
```

Example限制性政策僅允許列出操作

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "aoss:ListCollections",
```

```
      "Resource": "*"
    }
  ]
}
```

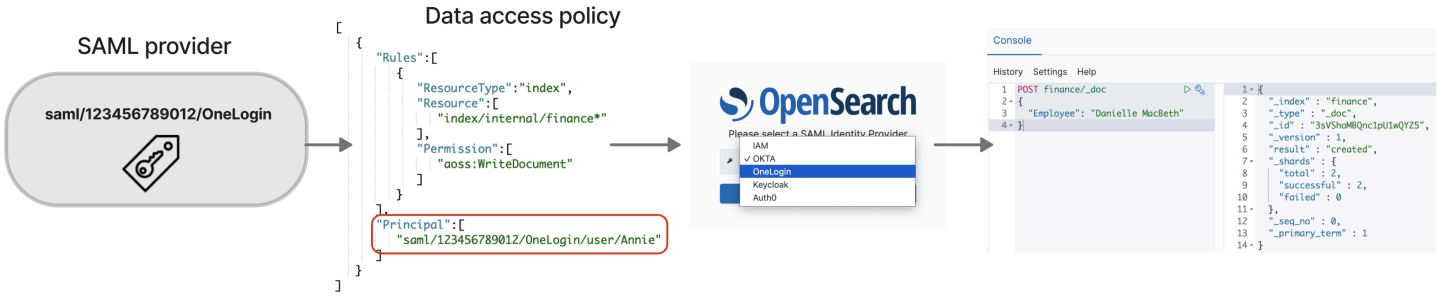
Amazon OpenSearch Serverless 的 SAML 身分驗證

您可以透過 Amazon OpenSearch Serverless 的 SAML 身分驗證，使用現有身分提供者，為無伺服器集合的 OpenSearch Dashboards 端點提供單一登入 (SSO)。

SAML 身分驗證讓您可以使用第三方身分提供者，登入 OpenSearch Dashboards，以便為資料編製索引和進行搜尋。OpenSearch Serverless 支援使用 SAML 2.0 標準的提供者，例如 IAM Identity Center、Okta、Keycloak、Active Directory Federation Services (AD FS) 和 Auth0。您可以設定 IAM Identity Center 來同步來自其他身分來源的使用者和群組，例如 Okta、OneLogin 和 Microsoft Entra ID。如需 IAM Identity Center 支援的身分來源清單以及設定這些來源的步驟，請參閱《IAM Identity Center 使用者指南》中的入門教學課程。

> **Note**
>
> SAML 身分驗證僅適用於透過 Web 瀏覽器存取 OpenSearch Dashboards。經驗證的使用者只能透過 OpenSearch Dashboards 中的開發工具，向 OpenSearch API 操作發出請求。SAML 憑證不能讓您對 OpenSearch API 操作發出直接 HTTP 請求。

若要設定 SAML 身分驗證，您應先設定 SAML 身分提供者 (IdP)。然後，您可以將該 IdP 中的一個或多個使用者納入資料存取政策中。此政策會向它授予集合和/或索引的某些許可。然後，使用者可以登入 OpenSearch Dashboards，並執行資料存取政策中允許的動作。



主題

- 考量事項
- 必要許可

- 建立 SAML 提供者 (主控台)
- 存取 OpenSearch Dashboards
- 授予 SAML 身分對集合資料的存取權
- 建立 SAML 提供者 (AWS CLI)
- 檢視 SAML 提供者
- 更新 SAML 提供者
- 刪除 SAML 提供者

考量事項

設定 SAML 身分驗證時請考量下列事項：

- 不支援已簽署和已加密的請求。
- 不支援已加密的聲明。
- 不支援 IdP 啟動的身分驗證和登出。
- 在非 IAM 身分的情況下，服務控制政策 (SCP) 將不適用或評估 （例如 Amazon OpenSearch Serverless & SAML 中的 SAML 和 Amazon OpenSearch Service 的基本內部使用者授權）。

必要許可

OpenSearch Serverless 的 SAML 身分驗證使用下列 AWS Identity and Access Management (IAM) 許可：

- `aoss:CreateSecurityConfig`：建立 SAML 提供者。
- `aoss:ListSecurityConfig`：列出目前帳戶中的所有 SAML 提供者。
- `aoss:GetSecurityConfig`：檢視 SAML 提供者資訊。
- `aoss:UpdateSecurityConfig`：修改指定的 SAML 提供者組態，包括 XML 中繼資料。
- `aoss:DeleteSecurityConfig`：刪除 SAML 提供者。

下列身分型存取政策讓使用者可以管理所有 IdP 組態：

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

請注意，Resource 元素必須是萬用字元。

建立 SAML 提供者 (主控台)

這些步驟說明如何建立 SAML 提供者。這可透過 OpenSearch Dashboards 的服務提供者 (SP) 啟動的身分驗證來啟用 SAML 身分驗證。不支援 IdP 啟動的身分驗證。

啟用 OpenSearch Dashboards 的 SAML 身分驗證

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽面板上，展開 Serverless (無伺服器)，然後選擇 SAML authentication (SAML 身分驗證)。
3. 選擇 Add SAML provider (新增 SAML 提供者)。
4. 提供提供者的名稱和描述。

> **Note**
>
> 您指定的名稱可公開存取，當使用者登入 OpenSearch Dashboards 時，該名稱會出現在下拉式選單中。確保名稱易於識別，並且不會洩露有關您身分提供者的敏感資訊。

5. 在 Configure your IdP (設定 IdP) 下，複製聲明消費者服務 (ACS) URL。
6. 使用剛才複製的 ACS URL 來設定身分提供者。術語和步驟因提供者而異。請咨詢供應商文件。

例如，在 Okta 中，您可以建立「SAML 2.0 Web 應用程式」，並將 ACS URL 指定為 Single Sign On URL (單一登入 URL)、Recipient URL (收件者 URL) 和 Destination URL (目的地 URL)。對於 Auth0，您可以在 Allowed Callback URLs (允許的回呼 URL) 中加以指定。

7. 如果 IdP 有此值的欄位，請提供對象限制。對象限制是 SAML 聲明中的一個值，用於指定聲明的對象。使用 OpenSearch Serverless，您可以執行下列動作。請務必使用您自己的 AWS 帳戶 ID 取代下列程式碼範例中*###*：

   1. 使用預設對象限制 `:opensearch:`*`111122223333`*。
   2. （選用）使用 設定自訂受眾限制 AWS CLI。如需詳細資訊，請參閱建立 SAML 提供者 (AWS CLI)。

   對象限制欄位的名稱因提供者而異。對於 Okta，該名稱為 Audience URI (SP Entity ID) (對象 URI (SP 實體 ID))。對於 IAM 身分中心，該名稱為 Application SAML audience (應用程式 SAML 對象)。

8. 如果您使用的是 IAM 身分中心，您還需要指定下列屬性映射：`Subject=${user:name}`，格式為 `unspecified`。

9. 設定身分提供者之後，會產生 IdP 中繼資料檔案。此 XML 檔案包含提供者的相關資訊，例如 TLS 憑證、單一登入端點以及身分提供者的實體 ID。

   複製 IdP 中繼資料檔案中的文字，並將其貼到 Provide metadata from your IdP (透過 IdP 提供中繼資料) 欄位下方。或者，選擇 Import from XML file (從 XML 檔案匯入)，然後上傳檔案。中繼資料檔案如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>s
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
```

```
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST" Location="idp-sso-url"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

10. 將自訂使用者 ID 屬性欄位保持空白，以針對使用者名稱使用 SAML 聲明的 NameID元素。如果您的聲明不使用此標準元素，而是將使用者名稱作為自訂屬性，請在此處指定該屬性。屬性區分大小寫。僅支援單一使用者屬性。

    下列範例顯示 SAML 聲明中 NameID 的覆寫屬性：

    ```
    <saml2:Attribute Name="UserId" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
    format:basic">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="xs:string">annie</saml2:AttributeValue>
    </saml2:Attribute>
    ```

11. (選用) 在 Group attribute (群組屬性) 欄位中指定自訂屬性，例如 `role` 或 `group`。僅支援單一群組屬性。沒有預設的群組屬性。如果未指定群組屬性，資料存取政策只能包含使用者主體。

    下列範例顯示 SAML 聲明中的群組屬性：

    ```
    <saml2:Attribute Name="department"
     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">finance</saml2:AttributeValue>
    </saml2:Attribute>
    ```

12. 依預設，OpenSearch Dashboards 會在 24 小時後將使用者登出。您可以指定 OpenSearch Dashboards 逾時，將此值設定為 1 到 12 小時 (15 到 720 分鐘） 之間的任何數字。如果您嘗試將逾時設定為等於或小於 15 分鐘，您的工作階段將重設為一小時。

13. 選擇 Create SAML provider (建立 SAML 提供者)。

存取 OpenSearch Dashboards

設定 SAML 提供者之後，與該提供者相關聯的所有使用者和群組都可以導覽至 OpenSearch Dashboards 端點。Dashboards URL 具有所有集合*collection-endpoint*/_dashboards/的格式。

如果您已啟用 SAML，選取 中的連結會 AWS 管理主控台 引導您前往 IdP 選擇頁面，您可以在其中使用您的 SAML 登入資料登入。首先，使用下拉式清單選取身分提供者：

然後使用 IdP 憑證登入。

如果您沒有啟用 SAML，選取 中的連結會 AWS 管理主控台 引導您以 IAM 使用者或角色身分登入，沒有 SAML 的選項。

授予 SAML 身分對集合資料的存取權

建立 SAML 提供者後，您仍然需要授予基礎使用者和群組對集合內資料的存取權。您可以透過資料存取政策授予存取權。在您提供使用者存取權之前，他們將無法讀取、寫入或刪除集合內的任何資料。

若要授予存取權，請建立資料存取政策，並在 `Principal` 陳述式中指定 SAML 使用者和/或群組 ID：

```
[
   {
      "Rules":[
       ...
      ],
      "Principal":[
         "saml/987654321098/myprovider/user/Shaheen",
         "saml/987654321098/myprovider/group/finance"
      ]
   }
]
```

您可以授予對集合、索引或兩者的存取權。如果您希望不同的使用者擁有不同的許可，請建立多個規則。如需可用許可的清單，請參閱支援的政策許可。如需有關如何格式化存取政策的資訊，請參閱政策語法。

建立 SAML 提供者 (AWS CLI)

若要使用 OpenSearch Serverless API 建立 SAML 提供者，請傳送 CreateSecurityConfig 請求：

```
aws opensearchserverless create-security-config \
    --name myprovider \
    --type saml \
    --saml-options file://saml-auth0.json
```

將包括中繼資料 XML 在內的 `saml-options` 指定為 .json 檔案中的鍵值映射。必須將中繼資料 XML 編碼為 JSON 逸出字串。

```
{
   "sessionTimeout": 70,
   "groupAttribute": "department",
   "userAttribute": "userid",
   "openSearchServerlessEntityId": "aws:opensearch:111122223333:app1",
```

```
   "metadata": "EntityDescriptor xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ... ... ... IDPSSODescriptor\r\n\/EntityDescriptor"
}
```

> **Note**
>
> （選用）使用 設定自訂受眾限制 AWS CLI。如需詳細資訊，請參閱建立 SAML 提供者 (AWS CLI)。

檢視 SAML 提供者

下列 ListSecurityConfigs 請求會列出您帳戶中的所有 SAML 提供者：

```
aws opensearchserverless list-security-configs --type saml
```

該請求會傳回所有現有 SAML 提供者的相關資訊，包括身分提供者產生的完整 IdP 中繼資料：

```
{
   "securityConfigDetails": [
      {
         "configVersion": "MTY2NDA1MjY4NDQ5M18x",
         "createdDate": 1664054180858,
         "description": "Example SAML provider",
         "id": "saml/111122223333/myprovider",
         "lastModifiedDate": 1664054180858,
         "samlOptions": {
            "groupAttribute": "department",
            "metadata": "EntityDescriptorxmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ...... ...IDPSSODescriptor\r\n/EntityDescriptor",
            "sessionTimeout": 120,
            "openSearchServerlessEntityId": "aws:opensearch:111122223333:app1",
            "userAttribute": "userid"
         }
      }
   ]
}
```

若要檢視特定提供者的相關詳細資訊 (包括未來更新的 configVersion)，請傳送 GetSecurityConfig 請求。

更新 SAML 提供者

若要使用 OpenSearch Serverless 主控台更新 SAML 提供者，請選擇 SAML authentication (SAML 身分驗證)、選取身分提供者，然後選擇 Edit (編輯)。您可以修改所有欄位，包括中繼資料和自訂屬性。

若要透過 OpenSearch Serverless API 更新提供者，請傳送 UpdateSecurityConfig 請求，並包含要更新之政策的識別符。您還必須包含組態版本，您可以使用 `ListSecurityConfigs` 或 `GetSecurityConfig` 命令擷取該版本。將最新的版本納入其中，可確保您不會意外覆寫其他人所做的變更。

下列請求會更新提供者的 SAML 選項：

```
aws opensearchserverless update-security-config \
    --id saml/123456789012/myprovider \
    --type saml \
    --saml-options file://saml-auth0.json \
    --config-version MTY2NDA1MjY4NDQ5M18x
```

將 SAML 組態選項指定為 .json 檔案中的鍵值映射。

> **Important**
>
> SAML 選項的更新不是遞增處理的。如果您在進行更新時未指定 `SAMLOptions` 物件中的參數值，則會以空白值覆寫現有值。例如，如果目前的組態包含 `userAttribute` 的值，然後您進行更新但不包含此值，則系統會將該值從組態中移除。透過呼叫 `GetSecurityConfig` 操作進行更新之前，請確保您知道現有值為何。

刪除 SAML 提供者

刪除 SAML 提供者時，對資料存取政策中關聯使用者和群組的任何參考將不再有效。為避免混淆，建議您先移除存取政策中對端點的所有參考，然後再刪除該端點。

若要使用 OpenSearch Serverless 主控台刪除 SAML 提供者，請選擇 Authentication (身分驗證)、選取提供者，然後選擇 Delete (刪除)。

若要透過 OpenSearch Serverless API 刪除提供者，請傳送 DeleteSecurityConfig 請求：

```
aws opensearchserverless delete-security-config --id saml/123456789012/myprovider
```

Amazon OpenSearch Serverless 的合規驗證

在多個合規計畫中，第三方稽核人員會評估 Amazon OpenSearch Serverless 的安全性和 AWS 合規性。這些計劃包括 SOC、PCI 和 HIPAA。

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 AWS 安全文件。

標記 Amazon OpenSearch Serverless 集合

標籤可讓您將任意資訊指派給 Amazon OpenSearch Serverless 集合，以便對該資訊進行分類和篩選。標籤是您指派或 AWS 指派給 AWS 資源的中繼資料標籤。

每個標籤皆包含鍵與值。對於您指派的標籤，您可以定義索引鍵和值。例如，您可以將鍵定義為 `stage`，將資源的值定義為 `test`。

使用標籤，您可以識別和組織您的 AWS 資源。許多 AWS 服務支援標記，因此您可以將相同的標籤指派給來自不同 服務的資源，以指出資源相關。例如，您可以將相同標籤指派給您已指派給 Amazon OpenSearch Service 網域的 OpenSearch Serverless 集合。

在 OpenSearch Service 中，主要資源是集合。您可以使用 OpenSearch Service 主控台、 AWS CLI、OpenSearch Serverless API 操作或 AWS SDKs 來新增、管理和移除集合中的標籤。

必要許可

OpenSearch Serverless 使用下列 AWS Identity and Access Management Access Analyzer (IAM) 許可來標記集合：

- `aoss:TagResource`
- `aoss:ListTagsForResource`
- `aoss:UntagResource`

標記集合（主控台）

主控台是標記集合的最簡單方法。

建立標籤 (主控台)

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 展開左側導覽窗格中的 Serverless (無伺服器)，然後選擇 Collections (集合)。
3. 選取您要新增標籤的集合，然後前往 Tags (標籤) 索引標籤。
4. 選擇 Manage (管理) 和 Add new tag (新增標籤)。
5. 輸入標籤索引鍵和選用的值。
6. 選擇儲存。

若要刪除標籤，請按照同樣的步驟進行，然後在 Manage tags (管理標籤) 頁面上選擇 Remove (移除)。

如需使用主控台處理標籤的詳細資訊，請參閱 《AWS 管理主控台入門指南》中的標籤編輯器。

標記集合 (AWS CLI)

若要使用 標記集合 AWS CLI，請傳送 TagResource 請求：

```
aws opensearchserverless tag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tags Key=service,Value=aoss Key=source,Value=logs
```

使用 ListTagsForResource 命令檢視集合的現有標籤：

```
aws opensearchserverless list-tags-for-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
```

使用 UntagResource 命令移除集合的標籤：

```
aws opensearchserverless untag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tag-keys service
```

Amazon OpenSearch Serverless 中支援的操作和外掛程式

Amazon OpenSearch Serverless 支援各種 OpenSearch 外掛程式，以及 OpenSearch 中提供的索引編製、搜尋和中繼資料 API 操作的子集。您可以將許可包含在資料存取政策內資料表左欄中，以限制對特定操作的存取。

主題

- 支援的 OpenSearch API 操作與許可
- 支援的 OpenSearch 外掛程式

支援的 OpenSearch API 操作與許可

下表列出 OpenSearch Serverless 支援的 API 操作及其對應的資料存取政策許可：

資料存取政策許可	OpenSearch API 操作	描述和警告
`aoss:CreateIndex`	PUT <index>	建立索引。如需詳細資訊，請參閱 Create index (建立索引)。 **Note** 此許可也適用於在 OpenSearch Dashboards 上使用範例資料建立索引。
`aoss:DescribeIndex`	• GET <index> • GET <index>/_mapping • GET <index>/_mappings • GET <index>/_setting • GET <index>/_setting/<setting> • GET <index>/_settings • GET <index>/_settings/<setting>	描述索引。如需詳細資訊，請參閱下列資源： • 取得索引 • 取得映射 • 取得設定 • 索引存在

資料存取政策許可	OpenSearch API 操作	描述和警告
	• GET _cat/indices • GET _mapping • GET _mappings • GET _resolve/index/<index> • HEAD <index>	• CAT 索引（回應不包含 `health`或 `status` 欄位。）
`aoss:WriteDocument`	• DELETE <index>/_doc/<id> • POST <index>/_bulk • POST <index>/_create/<id> (僅適用於搜尋集合類型) • POST <index>/_doc • POST <index>/_update/<id> (僅適用於搜尋集合類型) • POST _bulk • PUT <index>/_create/<id> (僅適用於搜尋集合類型) • PUT <index>/_doc/<id> (僅適用於搜尋集合類型)	編寫和更新文件。如需詳細資訊，請參閱下列資源： • 大批 • 為資料編製索引 **Note** 僅允許對類型為 SEARCH 的集合執行某些操作。如需詳細資訊，請參閱the section called “選擇集合類型”。

資料存取政策許可	OpenSearch API 操作	描述和警告
`aoss:ReadDocument`	• GET <index>/_analyz • GET <index>/_doc/<id> • GET <index>/_explain/<id> • GET <index>/_mget • GET <index>/_source/<id> • GET <index>/_count • GET <index>/_field_caps • GET <index>/_msearch • GET <index>/_rank_eval • GET <index>/_search • GET <index>/_validate/<query> • GET _analyze • GET _field_caps • GET _mget • GET _search • GET /_search/point_in_time/_all • HEAD <index>/_doc/<id> • HEAD <index>/_source/<id> • POST /_plugins/_sql • POST /_plugins/_ppl • POST /_plugins/_sql/_explain • POST /_plugins/_ppl/_explain • POST /_plugins/_ppl/_close • POST <index>/_analyze • POST /_search/point_in_time • POST <index>/_explain/<id> • POST <index>/_count • POST <index>/_field_caps • POST <index>/_rank_eval	閱讀文件。如需詳細資訊，請參閱下列資源： • 執行文字分析 • 取得文件 • Count (計數) • 查詢 DSL • 將評估排名 • 分析 API • 說明 • 時間點 • SQL 和 PPL

資料存取政策許可	OpenSearch API 操作	描述和警告
	• POST <index>/_search • POST _analyze • POST _field_caps • POST _search • DELETE /_search/point_in_time/_all • DELETE /_search/point_in_time	
`aoss:DeleteIndex`	DELETE <target>	刪除索引。如需詳細資訊，請參閱 Delete index (刪除索引)。
`aoss:UpdateIndex`	• POST _mapping • POST <index>/_mapping/ • POST <index>/_mappings/ • POST <index>/_setting • POST <index>/_settings • POST _setting • POST _settings • PUT _mapping • PUT <index>/_mapping • PUT <index>/_mappings/ • PUT <index>/_setting • PUT <index>/_settings • PUT _setting • PUT _settings	更新索引設定。如需詳細資訊，請參閱下列資源： • 映射 • 更新設定

資料存取政策許可	OpenSearch API 操作	描述和警告
`aoss:CreateCollectionItems`	• POST _aliases • POST /_plugins/_flow_framework/workflow • * POST /_plugins/_flow_framework/workflow/<workflow_id>/_provision • PUT _ingest/pipeline/<pipeline-id> • PUT _search/pipeline/<pipeline-id>	• 建立索引別名、管道和範本。如需詳細資訊，請參閱 Create aliases (建立別名)。 • * 佈建或重新佈建範本。ML Commons 用戶端和 OpenSearch Serverless 服務會管理相依政策。

資料存取政策許可	OpenSearch API 操作	描述和警告
`aoss:DescribeCollectionItems`	• GET <index>/_alias/<alias> • GET _alias • GET _alias/<alias> • GET _cat/aliases • GET _cat/templates • GET _cat/templates/<template_name> • GET _component_template • GET _component_template/<component-template> • GET _index_template • GET _index_template/<index-template> • GET _ingest/pipeline/<pipeline-id> • GET _ingest/pipeline/_simulate • GET /_plugins/_flow_framework/workflow/<workflow-id> • GET /_plugins/_flow_framework/workflow/_search • GET /_plugins/_flow_framework/workflow/<workflow-id>/_status • GET /_plugins/_flow_framework/workflow/state/_search • GET /_plugins/_flow_framework/workflow/_steps • GET /_plugins/_flow_framework/workflow/_step ? workflow_step =<step_name> • GET _search/pipeline/<pipeline-id> • HEAD _alias/<alias>	說明如何使用別名、索引和架構範本，以及管道。如需詳細資訊，請參閱下列資源： • 管理別名 • 索引範本

資料存取政策許可	OpenSearch API 操作	描述和警告
	• HEAD _component_template/<component-template> • HEAD _index_template/<name> • HEAD <index>/_alias/<alias> • POST _ingest/pipeline/_simulate • POST /_plugins/_flow_framework/workflow/_search • POST /_plugins/_flow_framework/workflow/state/_search	
`aoss:UpdateCollectionItems`	• POST <index>/_alias/<alias> • POST <index>/_aliases/<alias> • POST _component_template/<component-template> • POST _index_template/<index-template> • * POST /_plugins/_flow_framework/workflow/<workflow_id>/_deprovision • PUT <index>/_alias/<alias> • PUT <index>/_aliases/<alias> • PUT _component_template/<component-template> • PUT _index_template/<index-template> • PUT /_plugins/_flow_framework/workflow/<workflow_id>	更新別名、索引範本和架構範本。如需詳細資訊，請參閱下列資源： • 索引別名 • 索引範本 * 取消佈建範本的 API。ML Commons 用戶端和 OpenSearch Serverless 服務會管理相依政策。

資料存取政策許可	OpenSearch API 操作	描述和警告
`aoss:DeleteCollectionItems`	• DELETE <index>/_alias/<alias> • DELETE _component_template/<component-template> • DELETE _index_template/<index-template> • DELETE <index>/_aliases/<alias> • DELETE _search/pipeline/<pipeline-id> • DELETE _ingest/pipeline/<pipeline-id> • DELETE /_plugins/_flow_framework/workflow/<workflow_id>	刪除別名、索引和架構範本，以及管道。如需詳細資訊，請參閱下列資源： • 刪除別名 • 刪除範本

資料存取政策許可	OpenSearch API 操作	描述和警告
`aoss:DescribeMLResource`	• GET /_plugins/_ml/models/<model_id> • GET /_plugins_ml/models/_search • GET /_plugins/_ml/model_groups/<model_group_id> • GET /_plugins/_ml/model_groups/_search • GET /_plugins/_ml/connectors/<connector_id> • GET /_plugins/_ml/connectors/_search • GET /_plugins/_ml/profile/tasks/<task_id> • POST /_plugins/_ml/models/_search • POST /_plugins/_ml/model_groups/_search • POST /_plugins/_ml/connectors/_search	描述 GET 和搜尋 APIs 以擷取模型和連接器的相關資訊。
`aoss:CreateMLResource`	• POST /_plugins/_ml/models/_register • POST /_plugins/_ml/model_groups/_register • POST /_plugins/_ml/connectors/_create	提供建立 ML 資源的許可。

資料存取政策許可	OpenSearch API 操作	描述和警告
`aoss:UpdateMLResource`	• PUT /_plugins/_ml/models/<model_id> • POST /_plugins/_ml/models/<model_id>/_deploy • POST /_plugins/_ml/models/<model_id>/_undeploy • PUT /_plugins/_ml/model_groups/<model_group_id> • PUT /_plugins/_ml/connectors/<connector_id>	提供更新現有 ML 資源的許可。
`aoss:DeleteMLResource`	• DELETE /_plugins/_ml/models/<model_id> • DELETE _plugins/_ml/model_groups/<model_group_id> • DELETE /_plugins/_ml/connectors/<connector_id> • DELETE /_plugins/_ml/tasks/<task_id>	提供刪除 ML 資源的許可。
`aoss:ExecuteMLResource`	• POST /_plugins/_ml/models/<model_id>/_predict	提供執行模型的許可。

支援的 OpenSearch 外掛程式

OpenSearch Serverless 集合會預先封裝來自 OpenSearch 社群的下列外掛程式。Serverless 會為您自動部署和管理外掛程式。

分析外掛程式

- ICU 分析
- 日文 (kuromoji) 分析
- 韓語 (Nori) 分析

- 語音分析
- 智慧型中文分析
- Stempel 波蘭文分析
- 烏克蘭文分析

映射器外掛程式

- 映射器大小
- 映射器 Murmur3
- 映射器註釋文字

指令碼編寫外掛程式

- Painless
- 運算式
- Mustache

此外，OpenSearch Serverless 包含所有以模組形式提供的外掛程式。

監控 Amazon OpenSearch Serverless

監控是維護 Amazon OpenSearch Serverless 及其他 AWS 解決方案可靠性、可用性和效能的重要部分。 AWS 提供下列監控工具來監看 OpenSearch Serverless、在發生錯誤時回報，並適時採取自動動作：

- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。

  例如，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。
- AWS CloudTrail 會擷取來自或代表 AWS 帳戶發出的 API 呼叫和相關事件。它會將日誌檔案交付到您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱「AWS CloudTrail 使用者指南」。

- Amazon EventBridge 提供近乎即時的系統事件串流，描述 OpenSearch Service 網域中的變更。您可以建立監控特定事件的規則，並在發生這些事件 AWS 服務 時觸發其他 中的自動動作。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

使用 Amazon CloudWatch 監控 OpenSearch Serverless

您可以使用 CloudWatch 監控 Amazon OpenSearch Serverless，CloudWatch 會收集原始資料並將該資料處理成可讀且近乎即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。

您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

OpenSearch Serverless 會在 AWS/AOSS 命名空間中報告下列指標。

指標	Description
ActiveCollection	指出集合是否處於作用中狀態。值 1 表示集合處於 ACTIVE 狀態。系統會在成功建立集合時發出這個值，並在您刪除集合為止保持為 1。指標的值不能為 0。 相關統計資料：上限 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒
DeletedDocuments	已刪除的文件總數。 相關統計資料：平均數、總和 維度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻率：60 秒
IndexingOCU	用於擷取收集資料的 OpenSearch 運算單位 OCUs) 數量。此指標適用於帳戶層級。僅代表不屬於任何集合群組的集合使用量。

指標	Description
	相關統計資料：總和 維度：ClientId 頻率：60 秒
IndexingOCU	用於擷取收集資料的 OpenSearch 運算單位 OCUs) 數量。此指標適用於集合群組層級。 相關統計資料：總和 維度：ClientId、CollectionGroupId 、CollectionGroupName 頻率：60 秒
IngestionDataRate	集合或索引的索引編製速率 (以每秒 GiB 為單位)。此指標僅適用於大量編製索引請求。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻率：60 秒
IngestionDocumentErrors	擷取集合或索引期間的文件錯誤總數。成功發出大量編製索引請求後，撰寫者會處理該請求，並針對請求內所有失敗的文件發出錯誤。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻率：60 秒

指標	Description
IngestionDocumentRate	文件擷取至集合或索引的每秒速率。此指標僅適用於大量編製索引請求。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻率：60 秒
IngestionRequestErrors	對集合發出大量編製索引請求錯誤的總數。當大量編製索引請求因任何原因 (例如驗證或可用性問題) 而失敗時，OpenSearch Serverless 會發出此指標。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒
IngestionRequestLatency	大量寫入集合操作的延遲 (以秒為單位)。 相關統計資料：下限、上限、平均數 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒
IngestionRequestRate	集合接收的大量寫入操作總數。 相關統計資料：下限、上限、平均數 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒

指標	Description
IngestionRequestSuccess	成功在集合中執行索引編製操作的總數。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒
SearchableDocuments	集合或索引中可搜尋文件的總數。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻率：60 秒
SearchRequestErrors	集合每分鐘的查詢錯誤總數。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒
SearchRequestLatency	針對集合完成搜尋操作所需的平均時間 (以毫秒為單位)。 相關統計資料：下限、上限、平均數 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒

指標	Description
SearchOCU	搜尋集合資料所用的 OpenSearch 運算單元 (OCU) 數量。此指標適用於帳戶層級。僅代表不屬於任何集合群組的集合使用量。 相關統計資料：總和 維度：ClientId 頻率：60 秒
SearchOCU	搜尋集合資料所用的 OpenSearch 運算單元 (OCU) 數量。此指標適用於集合群組層級。 相關統計資料：總和 維度：ClientId、CollectionGroupId 、CollectionGroupName 頻率：60 秒
SearchRequestRate	集合每分鐘的搜尋請求總數。 相關統計資料：平均數、上限、總和 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒
StorageUsedInS3	使用的 Amazon S3 儲存量，以位元組為單位。OpenSearch Serverless 會將索引資料儲存在 Amazon S3 中。您必須選擇一分鐘的期間，才能取得準確的值。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻率：60 秒

指標	Description
VectorIndexBuildAccelerationOCU	用於加速向量索引的 OpenSearch 運算單位 (OCU) 數量。OCUs 此指標適用於集合層級。 相關統計資料：總和 維度：ClientId、 CollectionId 頻率：60 秒
2xx, 3xx, 4xx, 5xx	產生指定 HTTP 回應碼 (2xx、3xx、4xx、5xx) 的集合請求數。 相關統計資料：總和 維度：ClientId、CollectionId 、CollectionName 頻率：60 秒

使用 記錄 OpenSearch Serverless API 呼叫 AWS CloudTrail

Amazon OpenSearch Serverless 已與 整合 AWS CloudTrail，此服務可提供使用者、角色或 AWS 服務在 Serverless 中採取之動作的記錄。

CloudTrail 會將 OpenSearch Serverless 的所有 API 呼叫擷取為事件。擷取的呼叫包括從 OpenSearch Service 主控台進行的「無伺服器」區段的呼叫，以及對 OpenSearch Serverless API 操作進行的程式碼呼叫。

如果您建立追蹤，就可以將 CloudTrail 事件持續交付到 Amazon S3 儲存貯體，包括 OpenSearch Serverless 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。

您可以利用 CloudTrail 收集的資訊來判斷向 OpenSearch Serverless 發出的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

CloudTrail 中的 OpenSearch Serverless 資訊

當您建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。當活動在 OpenSearch Serverless 中發生時，該活動會與事件歷史記錄中的其他 AWS 服務事件一起記錄在 CloudTrail 事件中。您可以在 中檢視、搜尋和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱「使用 CloudTrail 事件歷史記錄檢視事件」。

若要持續記錄 中的事件 AWS 帳戶，包括 OpenSearch Serverless 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。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。

線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案

CloudTrail 會記錄所有 OpenSearch Serverles 動作，並會將這些動作記錄在 OpenSearch Serverless API 參考中。例如，對 `CreateCollection`、`ListCollections` 以及 `DeleteCollection` 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

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：

- 是否使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出請求。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

CloudTrail 中的 OpenSearch Serverless 資料事件

資料事件提供有關在資源上執行或在資源中執行的資源操作的資訊 （ 例如，搜尋或索引至 OpenSearch Serverless 集合 ）。這些也稱為資料平面操作。資料事件通常是大量資料的活動。根據預設，CloudTrail 不會記錄資料事件。CloudTrail 事件歷史記錄不會記錄資料事件。

資料事件需支付額外的費用。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。

您可以使用 CloudTrail 主控台 AWS CLI或 CloudTrail API 操作來記錄`AWS::AOSS::Collection`資源類型的資料事件。如需如何記錄資料事件的詳細資訊，請參閱 AWS CloudTrail 使用者指南中的使用 AWS 管理主控台　記錄資料事件和使用 AWS Command Line Interface記錄資料事件。

您可以設定進階事件選取器來篩選 `eventName`、`readOnly` 和 `resources.ARN` 欄位，以僅記錄對您重要的事件。如需這些欄位的詳細資訊，請參閱AWS CloudTrail API 參考中的AdvancedFieldSelector。

了解 OpenSearch Serverless Data Event 項目

於下列範例中：

- `requestParameters` 欄位包含對集合進行 API 呼叫的詳細資訊。它包含基本請求路徑（不含查詢參數）。
- `responseElements` 欄位包含狀態碼，指出您在修改資源時請求的結果。此狀態碼可協助您追蹤您的變更是否已成功處理或需要注意。
- OpenSearch Serverless 只會針對成功完成 IAM 身分驗證的請求記錄 CloudTrail 資料事件。

```
        "accessKeyId": "access-key",
        "userName": "",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA123456789EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/Admin",
            "accountId": "111122223333",
            "userName": "Admin"
          },
          "attributes": {
            "creationDate": "2025-08-15T22:57:38Z",
            "mfaAuthenticated": "false"
          },
          "sourceIdentity": "",
          "ec2RoleDelivery": "",
          "assumedRoot": ""
        },
        "identityProvider": "",
        "credentialId": ""
      },
      "eventTime": "2025-08-15T22:58:00Z",
      "eventSource": "aoss.amazonaws.com",
      "eventName": "Search",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "AWS Internal",
      "userAgent": "python-requests/2.32.3",
      "requestParameters": {
        "pathPrefix": "/_search"
      },
      "responseElements": null,
      "requestID": "2cfee788-EXAM-PLE1-8617-4018cEXAMPLE",
      "eventID": "48d43617-EXAM-PLE1-9d9c-f7EXAMPLE",
      "readOnly": true,
      "resources": [
        {
          "type": "AWS::AOSS::Collection",
          "ARN": "arn:aws:aoss:us-east-1:111122223333:collection/aab9texampletu45xh77"
        }
      ],
      "eventType": "AwsApiCall",
      "managementEvent": false,
      "recipientAccountId": "111122223333",
      "eventCategory": "Data"
    }
  ]
}
```

了解 OpenSearch Serverless Management Events 項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。

事件代表來自任何來源的單一請求。其中包含了請求的動作、動作的日期和時間、請求參數等相關資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

下列範例顯示示範 `CreateCollection`動作的 CloudTrail 日誌項目。

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"AssumedRole",
      "principalId":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/test-user",
      "accountId":"123456789012",
      "accessKeyId":"access-key",
      "sessionContext":{
         "sessionIssuer":{
            "type":"Role",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:role/Admin",
            "accountId":"123456789012",
            "userName":"Admin"
         },
         "webIdFederationData":{

         },
         "attributes":{
            "creationDate":"2022-04-08T14:11:34Z",
            "mfaAuthenticated":"false"
         }
      }
   },
   "eventTime":"2022-04-08T14:11:49Z",
   "eventSource":"aoss.amazonaws.com",
   "eventName":"CreateCollection",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"AWS Internal",
   "userAgent":"aws-cli/2.1.30 Python/3.8.8 Linux/5.4.176-103.347.amzn2int.x86_64 exe/
x86_64.amzn.2 prompt/off command/aoss.create-collection",
   "errorCode":"HttpFailureException",
```

```
   "errorMessage":"An unknown error occurred",
   "requestParameters":{
      "accountId":"123456789012",
      "name":"test-collection",
      "description":"A sample collection",
      "clientToken":"d3a227d2-a2a7-49a6-8fb2-e5c8303c0718"
   },
   "responseElements": null,
   "requestID":"12345678-1234-1234-1234-987654321098",
   "eventID":"12345678-1234-1234-1234-987654321098",
   "readOnly":false,
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "recipientAccountId":"123456789012",
   "eventCategory":"Management",
   "tlsDetails":{
      "clientProvidedHostHeader":"user.aoss-sample.us-east-1.amazonaws.com"
   }
}
```

使用 Amazon EventBridge 監控 OpenSearch Serverless 事件

Amazon OpenSearch Service 與 Amazon EventBridge 整合，可通知您影響您的網域的特定事件。來自 AWS 服務的事件會以近乎即時的方式交付至 EventBridge。相同的事件也會傳送至 Amazon CloudWatch Events，或 Amazon EventBridge 的前身產品。您可以撰寫規則來指出您感興趣的事件，以及在事件符合規則時要採取哪些自動化動作。您可以自動啟用的動作範例包括下列項目：

- 叫用 AWS Lambda 函數
- 叫用 Amazon EC2 執行命令
- 將事件轉傳至 Amazon Kinesis Data Streams
- 啟用 AWS Step Functions 狀態機器
- 通知 Amazon SNS 主題或 Amazon SQS 佇列

如需詳細資訊，請參閱 Amazon EventBridge 使用者指南中的 Amazon EventBridge 入門。

設定通知

您可以使用AWS 使用者通知，在發生 OpenSearch Serverless 事件時接收通知。事件是 OpenSearch Serverless 環境中變更的指標，例如當您達到 OCU 用量上限時。 Amazon EventBridge 會接收事件並

將通知路由至 AWS 管理主控台 通知中心和您選擇的交付管道。當事件符合您指定的規則時，便會收到通知。

OpenSearch 運算單位 (OCU) 事件

OpenSearch Serverless 會將事件傳送至 EventBridge。

接近上限的 OCU 用量

當您的搜尋或索引 OCU 用量達到容量限制的 75% 時，OpenSearch Serverless 會傳送此事件。您的 OCU 用量是根據您設定的容量限制和目前的 OCU 使用量來計算。

範例

以下是此類型 （ 搜尋 OCU) 的範例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Approaching Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your search OCU usage is at 75% and is approaching the configured
 maximum limit."
  }
}
```

以下是此類型 （ 索引 OCU) 的範例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Approaching Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
```

```
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your indexing OCU usage is at 75% and is approaching the configured
 maximum limit."
  }
```

OCU 用量達到上限

當您的搜尋或索引 OCU 用量達到容量限制的 100% 時，OpenSearch Serverless 會傳送此事件。您的 OCU 用量是根據您設定的容量限制和目前的 OCU 使用量來計算。

範例

以下是此類型 （搜尋 OCU) 的範例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Reached Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your search OCU usage has reached the configured maximum limit."
  }
}
```

以下是此類型 （索引 OCU) 的範例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Reached Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
```

```
    "eventTime" : 1678943345789,
    "description": "Your indexing OCU usage has reached the configured maximum limit."
  }
}
```

建立和管理 Amazon OpenSearch Service 網域

本章說明如何建立和管理 Amazon OpenSearch Service 網域。網域是 AWS佈建的同等開放原始碼 OpenSearch 叢集。建立網域時，您可以指定其設定、執行個體類型、執行個體計數和儲存配置。如需開放原始碼叢集的詳細資訊，請參閱 OpenSearch 文件中的建立叢集。

與入門教學課程中的概要說明不同，本章會說明所有選項，並提供相關的參考資訊。您可以使用 OpenSearch Service 主控台、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs的指示來完成每個程序。

建立 OpenSearch Service 網域

本節說明如何使用 OpenSearch Service 主控台或 AWS CLI 搭配 `create-domain`命令使用 來建立 OpenSearch Service 網域。

建立 OpenSearch Service 網域 (主控台)

依照以下程序使用主控台建立 OpenSearch Service 網域。

若要建立 OpenSearch Service 網域 (主控台)

1. 前往 https://aws.amazon.com 並選擇 Sign In to the Console (登入主控台)。
2. 在 Analytics (分析) 下，選擇 Amazon OpenSearch Service。
3. 選擇 Create domain (建立網域)。
4. 在 Domain name (網域名稱) 中，輸入網域名稱。名稱必須符合下列條件：
   - 專屬於您的帳戶和 AWS 區域
   - 開頭為小寫字母
   - 包含 3 到 28 個之間字元數
   - 只能包含小寫字母 a-z、號碼 0-9 和連字號 (-)
5. 針對網域建立方法，選擇標準建立。
6. 針對範本，選擇最符合您網域用途的選項：
   - 適用於需要高可用性和效能之工作負載的生產網域。這些網域使用異地同步備份 （有或沒有待命） 和專用主節點，以提高可用性。
   - 開發/測試開發或測試。這些網域可以使用異地同步備份 （有或沒有待命） 或單一可用區域。

> **Important**
>
> 在接下來的頁面中，不同部署類型有不同的選項。這些步驟包含所有選項。

7. 針對部署選項，選擇具有待命的網域來設定 3-AZ 網域，其中其中一個區域中的節點會保留為待命。此選項會強制執行許多最佳實務，例如指定的資料節點計數、主節點計數、執行個體類型、複本計數和軟體更新設定。
8. 對於 Version (版本)，選擇要使用的 OpenSearch 版本或舊版 Elasticsearch OSS。建議您選擇 OpenSearch 的最新版本。如需詳細資訊，請參閱the section called “支援的 Elasticsearch 和 OpenSearch 版本”。

   (選用) 如果您為網域選擇 OpenSearch 版本，請選取 Enable compatibility mode (啟用相容性模式)，讓 OpenSearch 將其版本報告為 7.10，這會允許某些 Elasticsearch OSS 用戶端和外掛程式在連線之前檢查該版本，以繼續使用該服務。
9. 對於 Instance type (執行個體類型)，選擇資料節點的執行個體類型。如需詳細資訊，請參閱the section called “支援的執行個體類型”。

   > **Note**
   >
   > 並非所有可用區域皆支援所有執行個體類型。如果您選擇具有或不具有待命的異地同步備份，建議您選擇目前世代的執行個體類型，例如 R5 或 I3。
10. 請在 Number of nodes (節點類型) 選擇資料節點數目。

    如需最大值，請參閱 OpenSearch Service 網域和執行個體配額。單一節點叢集適用於開發和測試，但不應將其用於生產工作負載。如需詳細的指導方針，請參閱 the section called “調整網域大小” 和 the section called “設定多可用區域網域”。

    > **Note**
    >
    > （選用）專用協調器節點支援所有 OpenSearch 版本和 ElasticSearch 6.8 到 7.10 版。專用協調器節點可用於已啟用專用叢集管理員的網域。若要啟用專用協調器節點，您將選取執行個體類型和計數。最佳實務是，您應該保持專用協調器節點的執行個體系列與資料節點 (Intel 型執行個體或 Graviton 型執行個體） 相同。
11. 針對儲存類型，選取 Amazon EBS。此清單中可用的磁碟區類型取決於您選擇的執行個體類型。關於建立特大網域的說明資訊，請參閱 the section called “PB 規模”。

12. 對於 EBS 儲存，請設定下列其他設定。根據您選擇的磁碟區類型，部分設定可能不會顯示。

設定	Description
EBS volume type (EBS 磁碟區類型)	選擇 General Purpose (SSD) - gp3 (一般用途 (SSD) - gp3) 和 General Purpose (SSD) - gp2 (一般用途 (SSD) - gp2)，或者上一代 Provisioned IOPS (SSD) (佈建 IOPS (SSD)) 和 Magnetic (磁帶) (標準)。
EBS storage size per node (每個節點的 EBS 儲存空間大小)	輸入您想要連接到每個資料節點的 EBS 磁碟區大小。 EBS 磁碟區大小是依照節點。您可以將資料節點的數量乘以 EBS 磁碟區大小來計算 OpenSearch Service 網域的總叢集大小。EBS 磁碟區的大小上限取決於兩個指定的 EBS 磁碟區類型，以及其所連接的執行個體類型。如需進一步了解，請參閱 EBS 磁碟區大小限制。
佈建 IOPS	如果您已選取佈建 IOPS SSD 磁碟區類型，請輸入磁碟區支援的每秒 I/O 操作數 (IOPS)。

13. （選用）如果您選取磁碟gp3區類型，請展開進階設定，並指定超出儲存價格所包含的額外 IOPS（每個資料節點每佈建 3 TiB 磁碟區大小最多 16，000 個）和輸送量（每個資料節點每佈建 3 TiB 磁碟區大小最多 1，000 MiB/s)。如需詳細資訊，請參閱 Amazon OpenSearch Service 定價。
14. (選用) 若要預覽 UltraWarm 儲存區，請選擇 Enable UltraWarm data nodes (啟用 UltraWarm 資料節點)。每個執行個體類型都有可處理的儲存量上限。將該數量乘以總計可處理暖儲存的暖資料節點數。
15. (選用) 若要啟用 cold storage (冷儲存)，選擇 Enable cold storage (啟用冷儲存)。您必須啟用 UltraWarm 才能啟用冷儲存。
16. 如果您將異地同步備份與待命搭配使用，則會啟用三個專用主節點。選擇您想要的主節點類型。如果您選擇沒有待命網域的異地同步備份，請選取啟用專用主節點，然後選擇您想要的主節點類型和數量。專用主節點可提高叢集穩定性，也是執行個體計數大於 10 的網域所需要的。我們建議生產網域應具備三個專用主節點。

> **Note**
>
> 您可以為專用主節點及資料節點選擇不同的執行個體類型。例如，您可以針對資料節點選擇一般用途或儲存最佳化執行個體，而不是專用主節點的運算最佳化執行個體。

17. (選用) 對於執行 OpenSearch 或 Elasticsearch 5.3 或更高版本的網域，快照組態是不相關的。如需自動拍攝快照的詳細資訊，請參閱the section called “建立索引快照”。
18. 如果您想要使用自訂端點，而不是標準的 `https://search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-east-1.es.amazonaws.com`，請選擇 Enable custom endpoint (啟用自訂端點) 並提供名稱和憑證。如需詳細資訊，請參閱the section called “建立自訂端點”。
19. 對於 Network (網路)，選擇 VPC access (VPC 存取) 或 Public access (公有存取)。如果選擇 Public access (公開存取)，請跳到下一步驟。如果選擇 VPC access (VPC 存取)，請確認符合先決條件，再設定下列設定：

設定	Description
VPC	選擇您想使用的虛擬私有雲端 (VPC) ID。VPC 和網域必須位於相同 AWS 區域，而您必須選取一個租用設定為 Default (預設) 的 VPC。OpenSearch Service 尚不支援使用專用租用戶的 VPC。
子網路	選擇子網路。如果您啟用多個可用區，必須選擇兩個或三個子網路。OpenSearch Service 將會在子網路中放置 VPC 端點和彈性網路介面。 您必須在子網路中保留足夠數量的 IP 地址給網路界面。如需詳細資訊，請參閱在 VPC 子網路中保留 IP 地址。
安全群組	選擇一或多個 VPC 安全群組，允許您所需的應用程式連線到連接埠 (80 或 443) 上的 OpenSearch Service 網域，以及網域公開的通訊協定 (HTTP 或 HTTPS)。如需詳細資訊，請參閱the section called “VPC 支援”。
IAM 角色	保留預設角色。OpenSearch Service 使用此預先定義的角色 (也稱為服務連結角色) 存取您的 VPC，並將 VPC 端點和網路界面放置在 VPC 子網路。如需詳細資訊，請參閱 VPC 存取的服務連結角色。

設定	Description
IP 地址類型	選擇雙堆疊或 IPv4 做為您的 IP 地址類型。雙堆疊可讓您跨 IPv4 和 IPv6 地址類型共用網域資源，並且是建議的選項。如果您將 IP 地址類型設定為雙堆疊，則稍後無法變更地址類型。

20. 啟用或停用精細存取控制：

    - 如果您希望使用 IAM 進行使用者管理，請選擇 Set IAM ARN as master user (將 IAM ARN 設為主要使用者)，並指定 IAM 角色的 ARN。
    - 如果您想要使用內部使用者資料庫，請選擇建立主要使用者，並指定使用者名稱和密碼。

    無論您選擇哪個選項，主要使用者都能夠存取叢集中的所有索引和所有 OpenSearch API。如需應選擇哪個選項的指導，請參閱 the section called “重要概念”。

    如果您停用精細存取控制，您仍然可以透過將網域置放在 VPC 中、套用限制性存取政策，或是同時使用兩者來控制對您網域的存取。您必須啟用節點對節點加密和靜態加密，才能使用精細存取控制。

    > **Note**
    >
    > 我們強烈建議啟用精細存取控制來保護網域中的資料。精細存取控制可提供叢集、索引、文件和欄位層級安全性。

21. （選用）如果您想要針對 OpenSearch Dashboards 使用 SAML 身分驗證，請選擇啟用 SAML 身分驗證並設定網域的 SAML 選項。如需說明，請參閱the section called “OpenSearch Dashboards 的 SAML 身分驗證”。
22. (選用) 如果您想要為 OpenSearch Dashboards 使用 Amazon Cognito 身分驗證，請選擇 Enable Amazon Cognito authentication (啟用 Amazon Cognito 身分驗證)。接著選擇您希望用於 OpenSearch Dashboards 身分驗證的 Amazon Cognito 使用者集區和身分集區。如需建立這些資源的指導方針，請參閱 the section called “OpenSearch Dashboards 的 Amazon Cognito 身分驗證”。
23. （選用）如果您想要使用 IAM Identity Center (IDC) 身分驗證來連接現有的身分來源，並為 AWS 應用程式提供使用者的通用檢視，請選擇啟用透過 IAM Identity Center 進行身分驗證的 API 存取。如需詳細資訊，請參閱《IAM Identity Center 使用者指南》中的受信任身分傳播概觀。
24. （選用）如果您想要使用這些功能，請在進階功能區段中保持選取啟用自然語言查詢產生和 Amazon Q Developer 功能。

a. 選擇啟用 S3 向量做為增強型向量搜尋選項的引擎選項。如需詳細資訊，請參閱使用 Amazon S3 向量引擎的進階搜尋功能。

b. 針對增強型向量搜尋選項選擇啟用 GPU 加速。如需詳細資訊，請參閱 向量索引的 GPU 加速。

25. 針對存取政策，選擇存取政策或設定自己的其中一個。如果您選擇建立自訂政策，您可以自行設定，或從另一個網域中匯入。如需詳細資訊，請參閱the section called “身分和存取權管理”。

> **Note**
>
> 如果已啟用 VPC 存取，就無法使用以 IP 為基礎的政策。您反而可以使用安全群組來控制哪些 IP 地址可以存取網域。如需詳細資訊，請參閱the section called “關於 VPC 網域上的存取政策”。

26. (選用) 若要求對網域發出的所有請求都透過 HTTPS 到達，請選取 Require HTTPS for all traffic to the domain (通往網域的所有流量都需要 HTTPS)。若要啟用node-to-node加密，請選取Node-to-node加密。如需詳細資訊，請參閱the section called “節點對節點加密”。若要啟用靜態資料加密，請選取啟用靜態資料加密。如果您選擇具有待命部署的異地同步備份選項，則會預先選取這些選項。

27. （選用）選取使用 AWS 擁有的金鑰，讓 OpenSearch Service 代表您建立 AWS KMS 加密金鑰（或使用其已建立的金鑰）。否則，請選擇您自己的 KMS 金鑰。如需詳細資訊，請參閱the section called “靜態加密”。

28. 對於離峰時段，選取開始時間以排程需要藍/綠部署的服務軟體更新和自動調整最佳化。離峰更新有助於在高流量期間將叢集專用主節點上的壓力降至最低。

29. 對於 Auto-Tune (自動調整)，選擇是否允許 OpenSearch Service 對您的網域建議與記憶體相關的設定變更，以提高速度和穩定性。如需詳細資訊，請參閱the section called “自動調校”。

    （選用）選取離峰時段以排程週期性時段，在此期間自動調整會更新網域。

30. （選用）選取自動軟體更新以啟用自動軟體更新。

31. (選用) 新增標籤以描述您的網域，以便分類和篩選該資訊。如需詳細資訊，請參閱the section called “標記網域”。

32. (選用) 展開並設定 Advanced cluster settings (進階叢集設定)。如需這些選項的摘要，請參閱the section called “進階叢集設定”。

33. 選擇建立。

建立 OpenSearch Service 網域 (AWS CLI)

您可以使用 AWS CLI建立 OpenSearch Service 網域，而不使用主控台。如需語法，請參閱 AWS CLI 命令參考中的 Amazon OpenSearch Service。

命令範例

第一個範例示範以下 OpenSearch Service 網域組態：

- 使用 OpenSearch 1.2 版建立名為 mylogs 的 OpenSearch Service 網域
- 在網域中填入兩個 `r6g.large.search` 執行個體類型的執行個體
- 每個資料節點使用 100 GiB 一般用途 (SSD) gp3 EBS 磁碟區進行儲存
- 允許匿名存取，但只從一個 IP 地址：192.0.2.0/32

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.2 \
    --cluster-config  InstanceType=r6g.large.search,InstanceCount=2 \
    --ebs-options
 EBSEnabled=true,VolumeType=gp3,VolumeSize=100,Iops=3500,Throughput=125 \
    --access-policies '{"Version": "2012-10-17",        "Statement": [{"Action":
 "es:*", "Principal":"*","Effect": "Allow", "Condition": {"IpAddress":{"aws:SourceIp":
["192.0.2.0/32"]}}}]}'
```

下一個範例示範以下 OpenSearch Service 網域組態：

- 使用 Elasticsearch 7.10 版建立名為 mylogs 的 OpenSearch Service 網域
- 在網域中填入六個 `r6g.large.search` 執行個體類型的執行個體
- 每個資料節點使用 100 GiB 一般用途 (SSD) gp2 EBS 磁碟區進行儲存
- 限制對單一使用者的存取，以使用者的 AWS 帳戶 ID 識別：555555555555
- 跨三個可用區域分佈執行個體

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version Elasticsearch_7.10 \
    --cluster-config
 InstanceType=r6g.large.search,InstanceCount=6,ZoneAwarenessEnabled=true,ZoneAwarenessConfig={A
 \
```

```
    --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
    --access-policies '{"Version": "2012-10-17",        "Statement": [ { "Effect":
"Allow", "Principal": {"AWS": "arn:aws:iam::555555555555:root" }, "Action":"es:*",
"Resource": "arn:aws:es:us-east-1:555555555555:domain/mylogs/*" } ] }'
```

下一個範例示範以下 OpenSearch Service 網域組態：

- 使用 OpenSearch 1.0 版建立名為 mylogs 的 OpenSearch Service 網域
- 在網域中填入十個 `r6g.xlarge.search` 執行個體類型的執行個體
- 在網域填入三個 `r6g.large.search` 執行個體類型的執行個體，以做為專用主節點
- 使用 100 GiB 佈建 IOPS EBS 磁碟區進行儲存，為每個資料節點設定 1000 個 IOPS 基準效能
- 限制存取單一使用者和單一附屬來源；`_search` API

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.0 \
    --cluster-config
 InstanceType=r6g.xlarge.search,InstanceCount=10,DedicatedMasterEnabled=true,DedicatedMasterTyp
 \
    --ebs-options EBSEnabled=true,VolumeType=io1,VolumeSize=100,Iops=1000 \
    --access-policies '{"Version": "2012-10-17",        "Statement": [ { "Effect":
"Allow", "Principal": { "AWS": "arn:aws:iam::555555555555:root" }, "Action": "es:*",
"Resource": "arn:aws:es:us-east-1:555555555555:domain/mylogs/_search" } ] }'
```

> **Note**
>
> 如果您嘗試建立 OpenSearch Service 網域且已存在相同名稱的網域，則 CLI 不會報告錯誤。相反地，它會傳回現有網域的詳細資訊。

建立 OpenSearch Service 網域 (AWS SDKs)

AWS SDKs(Android 和 iOS SDKs除外） 支援 Amazon OpenSearch Service API 參考中定義的所有動作，包括 `CreateDomain`。如需程式碼範例，請參閱 the section called “使用 AWS SDKs”。如需安裝和使用 AWS SDKs 的詳細資訊，請參閱AWS 軟體開發套件。

建立 OpenSearch Service 網域 (AWS CloudFormation)

OpenSearch Service 已與 整合 AWS CloudFormation，這項服務可協助您建立和設定 AWS 資源的模型，以減少建立和管理資源和基礎設施的時間。您建立一個描述您想要建立之 OpenSearch 網域的範本，而 CloudFormation 會為您佈建與設定網域。如需詳細資訊 (包括 OpenSearch 網域的 JSON 和 YAML 範本範例)，請參閱 AWS CloudFormation 使用者指南中的 Amazon OpenSearch Service 資源類型參考。

設定存取政策

Amazon OpenSearch Service 提供數種方式來設定對 OpenSearch Service 網域的存取。如需詳細資訊，請參閱 the section called “身分和存取權管理” 和 the section called “精細定義存取控制”。

主控台提供預先設定存取政策，讓您可以為您網域的特定需求自訂。您也可以從其他 OpenSearch Service 網域匯入存取政策。如需有關這些存取政策如何與 VPC 存取進行互動的詳細資訊，請參閱 the section called “關於 VPC 網域上的存取政策”。

設定存取政策 (主控台)

1. 前往 https://aws.amazon.com，然後選擇 Sign In to the Console (登入主控台)。
2. 在 Analytics (分析) 下，選擇 Amazon OpenSearch Service。
3. 在導覽窗格中，在 My domains (我的網域) 下，選擇您想要更新的網域。
4. 選擇 Actions (動作) 和 Edit security configuration (編輯安全組態)。
5. 編輯存取政策 JSON，或匯入預先設定的選項。
6. 選擇儲存變更。

使用遷移助理遷移至 OpenSearch Service

Amazon OpenSearch Service 的遷移助理是全方位的解決方案，可簡化從自我管理的 Elasticsearch 或 OpenSearch 叢集遷移至 OpenSearch Service 的程序。此工具組可解決遷移的操作複雜性，同時確保資料完整性並驗證遷移後的效能。

概觀

無論您是在 中設定proof-of-concept AWS、轉換生產工作負載，還是升級至最新的 OpenSearch 版本，遷移助理都會提供step-by-step指引、最佳實務和工具，以充分利用 OpenSearch 遷移的完整潛力。

Migration Assistant 提供下列主要優點：

- 中繼資料遷移 - 遷移叢集中繼資料，包括索引設定、類型映射、索引範本和別名
- 資料遷移 - 將現有資料從舊版叢集遷移至 OpenSearch Service 網域
- 即時流量處理 - 以最低延遲攔截即時流量，並將即時流量從自我管理叢集重新導向至 OpenSearch Service 網域
- 流量複寫 - 複寫目標叢集上的生產流量，以驗證準確性和效能
- 效能測試 - 透過擷取和重播請求模式來模擬實際流量，以微調系統效能
- 全球可用性 - 跨最常見的 AWS 區域部署，以實現全球觸角和可擴展性

Migration Assistant 支援從 Elasticsearch 6.x 和 7.x 版以及 OpenSearch 1.x 和 2.x 版遷移。如需詳細資訊，請參閱支援的遷移路徑。

> **Note**
>
> 請注意下列有關遷移助理的其他資訊。
>
> - 此工具支援多躍點遷移（例如，從 Elasticsearch 5.x 遷移至 OpenSearch Service 3.x，在一個躍點中）。
> - 您可以復原遷移。
> - 對於某些使用案例，工具幾乎不需要停機時間。
> - 此工具提供高效能回填，而不會影響來源叢集。

遷移案例

Migration Assistant 旨在處理下列遷移案例：

中繼資料遷移

將索引設定、別名和範本等叢集中繼資料從來源叢集遷移至目標 OpenSearch Service 網域。

回填遷移

將現有或歷史資料從來源叢集遷移至目標 OpenSearch Service 網域，以確保在轉換期間保留您所有的重要資料。

即時流量遷移

將持續的即時流量從來源叢集複寫至目標 OpenSearch Service 網域，讓您在遷移期間維持服務可用性。

> **Important**
>
> 遷移策略並非普遍適用。遷移助理根據工程最佳實務提供指引，但您應該在遷移生產工作負載之前評估您的特定需求並徹底測試。

遷移助理入門

Amazon OpenSearch Service 的遷移助理可作為具有完整文件、部署範本和原始程式碼 AWS 的解決方案。若要開始使用遷移助理：

存取遷移助理資源

1. 在 Migration Assistant for Amazon OpenSearch Service 解決方案概觀中檢閱完整的解決方案文件。
2. 閱讀部署規劃指南，了解成本和需求。
3. 使用部署說明部署解決方案，並遵循 使用指南來執行遷移。

對於開發人員和進階使用者，遷移助理原始程式碼和其他文件可在 OpenSearch 遷移 GitHub 儲存庫中使用。

解決方案架構

在 中部署時 AWS，遷移助理會使用數個 AWS 服務來提供全方位的遷移解決方案：

- AWS CloudFormation - 提供基礎設施即程式碼 (IaC) 範本，以部署和設定遷移助理
- OpenSearch Service - 遷移搜尋和分析工作負載的目標服務
- Amazon Managed Streaming for Apache Kafka - 為持久儲存和重複使用 HTTP 流量提供串流處理功能
- Amazon Elastic Container Service - 在安全、可擴展的容器中執行遷移管理主控台和流量Replayer
- Amazon Elastic File System - 為來源和目標叢集的請求和回應資料提供可擴展的持久性儲存

- Amazon Simple Storage Service - 儲存歷史回填任務和基礎設施作為程式碼內容的快照

如需詳細的架構資訊，請參閱遷移助理架構詳細資訊。

進階叢集設定

使用進階選項來設定下列項目：

請求主體中的索引

指定 HTTP 請求的內文中是否允許明確參照索引。將此屬性設為 `false` 以避免使用者繞開附屬資源的存取控制權。根據預設，此值為 `true`。如需詳細資訊，請參閱the section called “進階選項和 API 考量”。

欄位資料快取分配

指定分配給欄位資料的 Java 堆積空間的百分比。依預設，此設定為 JVM 堆積的 20%。

> **Note**
>
> 許多客戶查詢輪換每日索引。建議您開始基準測試時，針對大部分這些使用案例將 `indices.fielddata.cache.size` 設定為 40% 的 JVM 堆積。對於非常大的索引，您可能需要大型的欄位資料快取。

最大子句計數

指定 Lucene 布林查詢中允許的最大子句數量。預設值為 1,024。超過所允許的子句數量的查詢會導致 `TooManyClauses` 錯誤。如需詳細資訊，請參閱 Lucene 文件。

Amazon OpenSearch Service 的自動語意擴充

簡介

Amazon OpenSearch Service word-to-word比對（詞搜尋）來尋找結果，類似於其他傳統搜尋引擎。這種方法非常適合產品代碼或模型編號等特定查詢，但難以進行抽象搜尋，因為了解使用者意圖變得至關重要。例如，當您搜尋 "shoes for the Beach" 時，語彙搜尋會比對目錄項目中的個別單字 "shoes"、"beach"、"for" 和 "the"，可能缺少不包含確切搜尋詞彙的相關產品，例如 "water-resistant shoess" 或 "surf shoes"。

自動語意強化透過同時考慮關鍵字比對和搜尋背後的內容意義來解決此限制。此功能可了解搜尋意圖，並改善高達 20% 的搜尋相關性。針對索引中的文字欄位啟用此功能，以增強搜尋結果。

> **Note**
>
> 自動語意擴充適用於執行 2.19 版或更新版本的 OpenSearch Service 網域。此外，具有 OpenSearch 2.19 版的網域也需要接受最新的服務軟體版本更新。目前， 功能適用於公有網域，不支援 VPC 網域。

模型詳細資訊和效能基準

雖然此功能可處理幕後的技術複雜性，而不會公開基礎模型，但我們透過簡短的模型描述和基準結果來提供透明度，協助您在關鍵工作負載中做出採用特徵的明智決策。

自動語意擴充使用服務受管、預先訓練的稀疏模型，可有效運作，而無需自訂微調。模型會分析您指定的欄位，並根據從各種訓練資料中學到的關聯將它們擴展到稀疏向量。展開的詞彙及其重要性權重會以原生 Lucene 索引格式存放，以便有效擷取。我們已使用僅限文件模式最佳化此程序，其中編碼只會在資料擷取期間發生。搜尋查詢只是字符化，而不是透過稀疏模型處理，使解決方案同時具有成本效益和效能。

我們在功能開發期間的效能驗證使用 MS MARCO 段落擷取資料集，以平均 334 個字元的段落為特色。對於相關性評分，我們在英文內容的 BEIR 基準上測量了前 10 個搜尋結果的平均標準化折扣累積收益 (NDCG) (ndcg@10)，在多語言內容的 MIRACL 上測量了平均 ndcg@10。我們透過用戶端、第 90 個百分位數 (p90) 測量和搜尋回應 p90 評估了延遲，並取得了 值。這些基準提供搜尋相關性和回應時間的基準效能指標。以下是關鍵基準號碼 -

- 英文 - 相較於語彙搜尋，相關性改善 20%。它也減少了 7.7% 的 P90 搜尋延遲 (BM25 為 26 毫秒，自動語意擴充為 24 毫秒）。
- 多語言 - 相較於語意搜尋，相關性改善 105%，而相較於語意搜尋，P90 搜尋延遲增加 38.4% (BM25 為 26 毫秒，自動語意擴充為 36 毫秒）。

鑑於每個工作負載的獨特性質，我們建議您在做出實作決策之前，先使用自己的基準標準在開發環境中評估此功能。

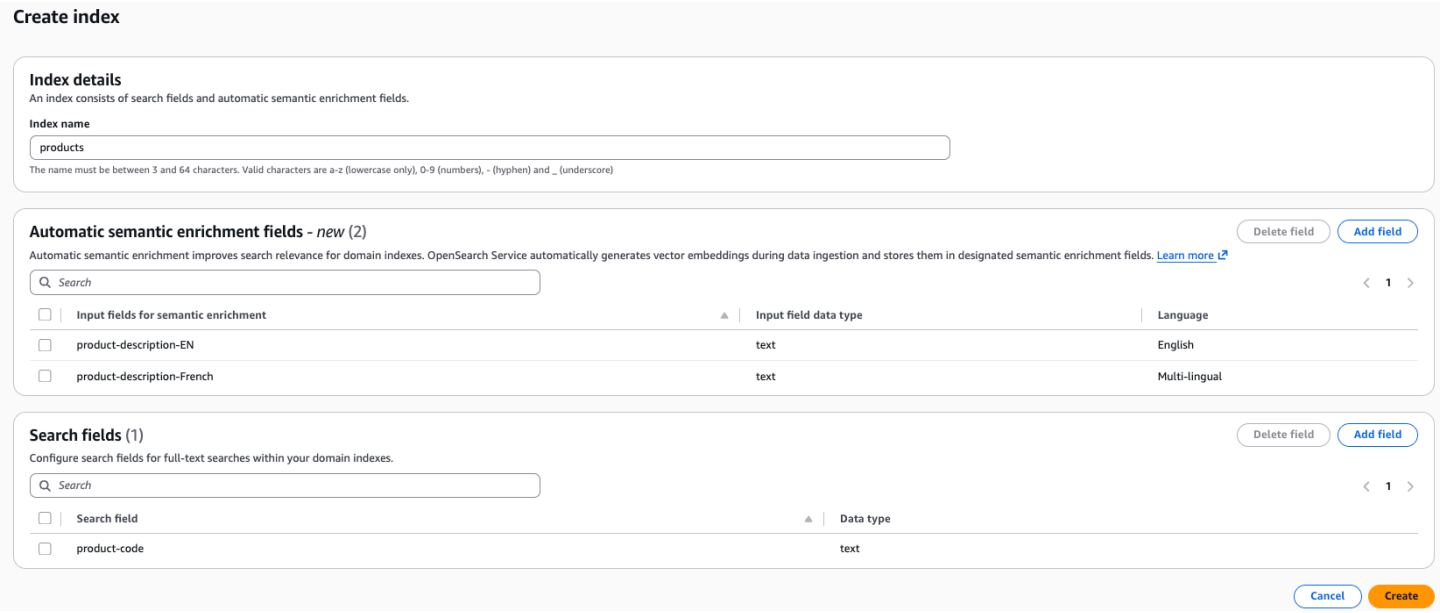
支援的語言

此功能支援英文。此外，模型也支援阿拉伯文、孟加拉文、中文、芬蘭文、法文、印地文、印尼文、日文、韓文、波斯文、俄文、西班牙文、斯瓦希里文和特拉古文。

設定網域的自動語意擴充索引

為您的文字欄位設定啟用自動語意擴充的索引非常簡單，您可以在新索引建立期間透過主控台、APIs 和 CloudFormation 範本進行管理。若要為現有索引啟用它，您需要為文字欄位重新建立已啟用自動語意擴充的索引。

主控台體驗 - AWS 主控台可讓您使用自動語意擴充欄位輕鬆建立索引。選取網域後，您會在主控台頂端找到建立索引按鈕。按一下建立索引按鈕後，您會找到定義自動語意擴充欄位的選項。在一個索引中，您可以結合英文和多語言的自動語意擴充，以及語彙欄位。



API 體驗 - 若要使用 AWS 命令列界面 (AWS CLI) 建立自動語意擴充索引，請使用 create-index 命令：

```
aws opensearch create-index \
--domain-name [domain_name] \
--index-name [index_name] \
--index-schema [index_body] \
```

在下列索引結構描述範例中， title_semantic 欄位的欄位類型設定為文字，且參數 semantic_enrichment 設定為狀態 ENABLED。設定語意_enrichment 參數可在 title_semantic 欄位上啟用自動語意擴充。您可以使用 language_options 欄位來指定英文或多語言。

```
    aws opensearch create-index \
    --id XXXXXXXXX \
```

```
    --index-name 'product-catalog' \
    --index-schema '{
    "mappings": {
        "properties": {
            "product_id": {
                "type": "keyword"
            },
            "title_semantic": {
                "type": "text",
                "semantic_enrichment": {
                    "status": "ENABLED",
                    "language_options": "english"
                }
            },
            "title_non_semantic": {
                "type": "text"
            }
        }
    }
}'
```

若要描述建立的索引，請使用下列命令：

```
aws opensearch get-index \
--domain-name [domain_name] \
--index-name [index_name] \
```

資料擷取和搜尋

建立已啟用自動語意擴充的索引後，此功能會在資料擷取程序期間自動運作，無需額外的組態。

資料擷取：當您將文件新增至索引時，系統會自動：

- 分析您為語意擴充指定的文字欄位
- 使用 OpenSearch Service 受管稀疏模型產生語意編碼
- 將這些豐富的表示與原始資料一起存放

此程序使用 OpenSearch 的內建 ML 連接器和擷取管道，這些管道會在幕後自動建立和管理。

搜尋：語意擴充資料已編製索引，因此查詢會有效率地執行，而不會再次叫用 ML 模型。這表示您可以改善搜尋相關性，而不需要額外的搜尋延遲額外負荷。

設定自動語意擴充的許可

在建立具有自動語意擴充的索引之前，您需要設定必要的許可。本節說明不同索引操作所需的許可，以及如何針對 AWS Identity and Access Management (IAM) 和精細存取控制案例進行設定。

IAM 許可

自動語意擴充操作需要下列 IAM 許可。這些許可會根據您要執行的特定索引操作而有所不同。

CreateIndex API 許可

若要建立具有自動語意擴充的索引，您需要下列 IAM 許可：

- `es:CreateIndex` – 建立具有語意擴充功能的索引。
- `es:ESHttpHead` – 執行 HEAD 請求以檢查索引是否存在。
- `es:ESHttpPut` – 執行建立索引的 PUT 請求。
- `es:ESHttpPost` – 執行索引操作的 POST 請求。

UpdateIndex API 許可

若要使用自動語意擴充更新現有索引，您需要下列 IAM 許可：

- `es:UpdateIndex` – 更新索引設定和映射。
- `es:ESHttpPut` – 執行索引更新的 PUT 請求。
- `es:ESHttpGet` – 執行 GET 請求以擷取索引資訊。
- `es:ESHttpPost` – 執行索引操作的 POST 請求。

GetIndex API 許可

若要擷取具有自動語意擴充之索引的相關資訊，您需要下列 IAM 許可：

- `es:GetIndex` – 擷取索引資訊和設定。
- `es:ESHttpGet` – 執行 GET 請求以擷取索引資料。

DeleteIndex API 許可

若要刪除具有自動語意擴充的索引，您需要下列 IAM 許可：

- `es:DeleteIndex` – 刪除索引及其語意擴充元件。
- `es:ESHttpDelete` – 執行 DELETE 請求以移除索引。

範例 IAM 政策

下列以身分為基礎的存取政策範例提供使用者使用自動語意擴充管理索引所需的許可：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSemanticEnrichmentIndexOperations",
            "Effect": "Allow",
            "Action": [
                "es:CreateIndex",
                "es:UpdateIndex",
                "es:GetIndex",
                "es:DeleteIndex",
                "es:ESHttpHead",
                "es:ESHttpGet",
                "es:ESHttpPut",
                "es:ESHttpPost",
                "es:ESHttpDelete"
            ],
            "Resource": "arn:aws:es:aws-region:111122223333:domain/domain-name"
        }
    ]
}
```

將 *aws-region*、*111122223333* 和 *domain-name* 取代為您的特定值。您可以在資源 ARN 中指定特定索引模式，以進一步限制存取。

精細存取控制許可

如果您的 Amazon OpenSearch Service 網域已啟用精細存取控制，則需要 IAM 許可以外的其他許可。每個索引操作都需要下列許可。

CreateIndex API 許可

啟用精細存取控制時，建立具有自動語意擴充的索引需要下列額外許可：

- `indices:admin/create` – 建立索引操作。
- `indices:admin/mapping/put` – 建立和更新索引映射。
- `cluster:admin/opensearch/ml/create_connector` – 建立機器學習連接器以進行語意處理。
- `cluster:admin/opensearch/ml/register_model` – 註冊機器學習模型以進行語意擴充。
- `cluster:admin/ingest/pipeline/put` – 建立用於資料處理的擷取管道。
- `cluster:admin/search/pipeline/put` – 建立查詢處理的搜尋管道。

UpdateIndex API 許可

啟用精細存取控制時，更新具有自動語意擴充的索引需要下列額外許可：

- `indices:admin/get` – 擷取索引資訊。
- `indices:admin/settings/update` – 更新索引設定。
- `indices:admin/mapping/put` – 更新索引映射。
- `cluster:admin/opensearch/ml/create_connector` – 建立機器學習連接器。
- `cluster:admin/opensearch/ml/register_model` – 註冊機器學習模型。
- `cluster:admin/ingest/pipeline/put` – 建立擷取管道。
- `cluster:admin/search/pipeline/put` – 建立搜尋管道。
- `cluster:admin/ingest/pipeline/get` – 擷取擷取管道資訊。
- `cluster:admin/search/pipeline/get` – 擷取搜尋管道資訊。

GetIndex API 許可

啟用精細存取控制時，需要下列額外許可，才能擷取具有自動語意擴充之索引的相關資訊：

- `indices:admin/get` – 擷取索引資訊。
- `cluster:admin/ingest/pipeline/get` – 擷取擷取管道資訊。
- `cluster:admin/search/pipeline/get` – 擷取搜尋管道資訊。

DeleteIndex API 許可

啟用精細存取控制時，需要下列額外許可才能刪除具有自動語意擴充的索引：

- `indices:admin/delete` – 刪除索引操作。

查詢重寫

自動語意擴充會自動將您現有的「比對」查詢轉換為語意搜尋查詢，而不需要修改查詢。如果比對查詢是複合查詢的一部分，系統會周遊您的查詢結構、尋找比對查詢，並將它們取代為神經稀疏查詢。目前，此功能僅支援取代「比對」查詢，無論是獨立查詢還是複合查詢的一部分。不支援「多重_比對」。此外，此功能支援所有複合查詢來取代其巢狀比對查詢。複合查詢包括：bool、Boosting、constant_score、dis_max、 function_score 和 hybrid。

自動語意擴充的限制

當套用至包含自然語言內容small-to-medium欄位時，自動語意搜尋最有效，例如電影標題、產品描述、評論和摘要。雖然語意搜尋增強了大多數使用案例的相關性，但可能不適用於某些案例。在決定是否針對特定使用案例實作自動語意擴充時，請考慮下列限制。

- 非常長的文件 – 目前的稀疏模型只會處理每個英文文件的前 8，192 個字符。對於多語言文件，它是 512 個字符。對於冗長的文章，請考慮實作文件區塊，以確保完整的內容處理。
- 日誌分析工作負載 – 語意擴充會大幅增加索引大小，這對於日誌分析而言可能不必要，其中完全相符通常就足夠。其他語意內容很少改善日誌搜尋的有效性，足以證明增加的儲存需求。
- 自動語意擴充與衍生來源功能不相容。
- 調節 – OpenSearch Service 網域的索引推論請求目前上限為 200 TPS。這是軟性限制；請聯絡 AWS Support 以取得更高的限制。

定價

Amazon OpenSearch Service 會根據索引時間產生稀疏向量時消耗的 OpenSearch Compute Units (OCUs) 來計費自動語意擴充。您只需支付索引期間實際用量的費用。您可以使用 Amazon CloudWatch 指標 SemanticSearchOCU 監控此消耗。如需模型字符限制、每個 OCU 的磁碟區輸送量，以及範例計算的特定詳細資訊，請造訪 OpenSearch Service 定價。

在 Amazon OpenSearch Service 中進行組態變更

Amazon OpenSearch Service 在更新網域時使用藍/綠部署程序。藍/綠部署會為網域更新建立閒置環境，以複製生產環境，並在這些更新完成後將使用者路由至新環境。在藍/綠部署中，藍色環境是目前的生產環境。綠色環境是閒置環境。

資料會從藍色環境遷移至綠色環境。當新環境準備就緒時，OpenSearch Service 會切換環境，將綠色環境提升為新的生產環境。切換不會遺失資料。此實務可將停機時間降至最低，並在新環境部署失敗時維護原始環境。

主題

- 通常會導致藍/綠部署的變更
- 通常不會導致藍/綠部署的變更
- 藍/綠部署選項
- 判斷變更是否會導致藍/綠部署
- 追蹤組態變更
- 組態變更的階段
- 藍/綠部署的效能影響
- 組態變更的費用
- 對驗證錯誤進行疑難排解

通常會導致藍/綠部署的變更

以下操作會造成藍/綠部署：

- 變更執行個體類型
- 啟用精細存取控制
- 執行服務軟體更新
- 啟用或停用專用主節點
- 啟用或停用無待命的異地同步備份
- 變更儲存類型、磁碟區類型或磁碟區大小
- 選擇不同的 VPC 子網路
- 新增或移除 VPC 安全群組

- 新增或移除專用協調器節點
- 啟用或停用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證
- 選擇不同的 Amazon Cognito 使用者集區或身分集區
- 修改進階設定
- 升級到新的 OpenSearch 版本（在部分或全部升級期間，OpenSearch Dashboards 可能無法使用）
- 啟用靜態資料加密或節點對節點加密
- 啟用或停用 UltraWarm 或冷儲存
- 停用自動調整並還原其變更
- 將選用外掛程式與網域建立關聯，以及取消選用外掛程式與網域的關聯
- 使用兩個專用主節點增加多可用區域網域的專用主節點計數
- 減少 EBS 磁碟區大小
- 如果您所做的上次變更正在進行或在 6 小時內發生，請變更 EBS 磁碟區大小、IOPS 或輸送量
- 啟用將稽核日誌發佈至 CloudWatch。

對於具有待命網域的異地同步備份，您一次只能提出一個變更請求。如果變更已在進行中，則會拒絕新的請求。您可以使用 `DescribeDomainChangeProgress` API 檢查目前變更的狀態。

通常不會導致藍/綠部署的變更

在大多數情況下，以下操作不會造成藍/綠部署：

- 修改存取政策
- 修改自訂端點
- 變更 Transport Layer Security (TLS) 政策
- 變更自動快照時間
- 啟用或停用 Require HTTPS (需要使用 HTTPS)
- 啟用自動調整，或停用且不還原其變更
- 如果您的網域有專用主節點，請變更資料節點或 UltraWarm 節點計數
- 如果您的網域有專用主節點，請變更專用主執行個體類型或計數（具有兩個專用主節點的多可用區域網域除外）
- 啟用或停用將錯誤日誌或慢速日誌發佈至 CloudWatch
- 停用將稽核日誌發佈至 CloudWatch

- 增加 gp3 EBS 類型的磁碟區大小、IOPS 或輸送量。

> **Note**
>
> 在 2026 年 3 月 10 日之前，只有 gp3 最多 3 TiB 的磁碟區才支援就地磁碟區增加。在 2026 年 3 月 10 日，已移除此限制，允許就地磁碟區增加超過 3 TiB。不過，如果您的叢集在此限制移除前的磁碟區大小超過 3 TiB，則第一次磁碟區增加將需要藍/綠部署。該叢集的所有後續磁碟區增加將做為就地更新執行，且不需要藍/綠部署。

- 新增或移除標籤

> **Note**
>
> 視您的服務軟體版本而定，有一些例外情況。如果您想要確定變更不會導致藍/綠部署，請在更新網域之前執行試轉，如果此選項可用。有些變更不提供試轉選項。通常建議您在尖峰流量時段之外變更叢集。

藍/綠部署選項

選取部署策略，以控制當更新時沒有足夠的容量時，叢集如何處理部署。

1. Full Swap 藍/綠 — 預設部署行為。預先需要完整的執行個體容量，確保在容量可用時部署最快。如果無法配置足夠的容量，則不會繼續部署。
2. 容量最佳化 — 建議用於具有 30 個以上資料節點的叢集。會先嘗試完全的藍/綠交換，如果容量不足， 會繼續批次部署。確保即使容量有限，部署也能完成。完成時間可能會增加，因為部署將分批完成。

Console

對於編輯網域流程：

1. 開啟位於 的 Amazon OpenSearch Service 主控台https://console.aws.amazon.com/aos/home。
2. 在導覽窗格的網域下，選擇網域名稱以開啟叢集組態。
3. 按一下叢集組態索引標籤右側的編輯按鈕。
4. 從部署策略選項中選擇網域更新所需的組態：

1. Full Swap 藍/綠 — 預設部署行為。預先需要完整的執行個體容量，確保在容量可用時部署最快。如果無法配置足夠的容量，則不會繼續部署。
2. 容量最佳化 — 建議用於具有 30 個以上資料節點的叢集。會先嘗試完全的藍/綠交換，如果容量不足， 會繼續批次部署。確保即使容量有限，部署也能完成。完成時間可能會增加，因為部署將分批完成。

5. 選擇儲存變更。

對於建立網域流程：

1. 開啟位於 的 Amazon OpenSearch Service 主控台https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中選擇 Domains (網域)。
3. 按一下建立網域按鈕。
4. 選取網域的所有必要組態。
5. 從部署策略選項中選擇網域更新所需的組態：

1. Full Swap 藍/綠 — 預設部署行為。預先需要完整的執行個體容量，確保在容量可用時部署最快。如果無法配置足夠的容量，則不會繼續部署。
2. 容量最佳化 — 建議用於具有 30 個以上資料節點的叢集。會先嘗試完全的藍/綠交換，如果容量不足， 會繼續批次部署。確保即使容量有限，部署也能完成。完成時間可能會增加，因為部署將分批完成。

6. 按一下右側的網域摘要面板上的建立按鈕。

API

您可以使用 UpdateDomainConfig API 設定部署策略。

容量最佳化選項

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
config
{
   "DeploymentStrategyOptions": {
    "DeploymentStrategy": "CapacityOptimized"
   }
}
```

預設選項

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
config
{
   "DeploymentStrategyOptions": {
    "DeploymentStrategy": "Default"
   }
}
```

判斷變更是否會導致藍/綠部署

您可以測試某些類型的計劃組態變更，以判斷它們是否會導致藍/綠部署，而不必遞交這些變更。在您啟動組態變更前，請使用主控台或 API 執行驗證檢查，以確保網域符合更新資格。

Console

驗證組態變更

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇 Domains (網域)。
3. 選取您要進行組態變更的網域。這會開啟網域詳細資訊頁面。選取 Actions (動作) 下拉式功能表，然後選擇 Edit cluster configuration (編輯叢集組態)。
4. 變更網域，例如變更執行個體類型或節點數量。
5. 在乾執行分析下，選擇執行。試轉會驗證您的組態變更是否發生錯誤，並判斷是否需要藍/綠部署。
6. 試轉完成時，結果會出現在頁面底部，以及試轉 ID。分析指出組態變更是否需要藍/綠部署。

   每次試轉都會覆寫先前的試轉。若要保留每次執行的詳細資訊，請儲存其試轉 ID。乾執行可使用 90 天，或直到您進行組態更新為止。
7. 若要繼續進行組態更新，請選擇 Save changes (儲存變更)。否則，請選擇 Cancel (取消)。任一選項都會帶您返回 Cluster configuration (叢集組態) 標籤。在此標籤上，您可以選擇 Dry run details (試轉詳細資訊) 以查看最新試轉的詳細資訊。此頁面也包含試轉前的組態與試轉組態之間的並排比較。

API

您也可以透過組態 API 執行試轉驗證。若要使用 API 測試變更，請將 DryRun 設定為 true，以及將 DryRunMode 設定為 Verbose。詳細資訊模式除了判斷變更是否會啟動藍/綠部署之外，還會執行驗證檢查。例如，此 UpdateDomainConfig 請求會測試啟用 UltraWarm 所產生的部署類型：

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
config
{
   "ClusterConfig": {
    "WarmCount": 3,
    "WarmEnabled": true,
    "WarmType": "ultrawarm1.large.search"
   },
   "DryRun": true,
   "DryRunMode": "Verbose"
}
```

請求會執行驗證檢查並傳回變更將造成之部署類型，但實際上不會執行更新：

```
{
   "ClusterConfig": {
     ...
    },
   "DryRunResults": {
      "DeploymentType": "Blue/Green",
      "Message": "This change will require a blue/green deployment."
    }
}
```

可能的部署類型包括：

- Blue/Green：變更將導致藍/綠部署。
- DynamicUpdate：變更不會導致藍/綠部署。
- Undetermined：網域仍處於處理狀態，因此無法判斷部署類型。
- None：無設定變更。

如果驗證失敗，其會傳回驗證失敗的清單。

```
{
```

```
   "ClusterConfig":{
      "..."
   },
   "DryRunProgressStatus":{
      "CreationDate":"2023-01-12T01:14:33.847Z",
      "DryRunId":"db00ca39-48b2-4774-bbd3-252cf094d205",
      "DryRunStatus":"failed",
      "UpdateDate":"2023-01-12T01:14:33.847Z",
      "ValidationFailures":[
         {
            "Code":"Cluster.Index.WriteBlock",
            "Message":"Cluster has index write blocks."
         }
      ]
   }
}
```

如果狀態仍處於 pending，您可以在 UpdateDomainConfig 回應中，在後續 DescribeDryRunProgress 呼叫中使用試轉 ID 來檢查驗證的狀態。

```
GET https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
dryRun?dryRunId=my-dry-run-id
{
    "DryRunConfig": null,
    "DryRunProgressStatus": {
        "CreationDate": "2023-01-12T01:14:42.998Z",
        "DryRunId": "db00ca39-48b2-4774-bbd3-252cf094d205",
        "DryRunStatus": "succeeded",
        "UpdateDate": "2023-01-12T01:14:49.334Z",
        "ValidationFailures": null
    },
    "DryRunResults": {
        "DeploymentType": "Blue/Green",
        "Message": "This change will require a blue/green deployment."
    }
}
```

若要在不進行驗證檢查的情況下執行試轉分析，請在使用組態 API 時將 DryRunMode 設定為 Basic。

Python

下列 Python 程式碼使用 UpdateDomainConfig API 執行試轉驗證檢查，如果檢查成功， 會呼叫相同的 API 而不進行試轉以開始更新。如果檢查失敗，指令碼會列印錯誤並停止。

```
import time
import boto3

client = boto3.client('opensearch')

response = client.UpdateDomainConfig(
    ClusterConfig={
        'WarmCount': 3,
        'WarmEnabled': True,
        'WarmCount': 123,
    },
    DomainName='test-domain',
    DryRun=True,
    DryRunMode='Verbose'
)

dry_run_id = response.DryRunProgressStatus.DryRunId

retry_count = 0

while True:

    if retry_count == 5:
        print('An error occured')
        break

    dry_run_progress_response = client.DescribeDryRunProgress('test-domain',
 dry_run_id)
    dry_run_status = dry_run_progress_response.DryRunProgressStatus.DryRunStatus

    if dry_run_status == 'succeeded':
        client.UpdateDomainConfig(
            ClusterConfig={
            'WarmCount': 3,
            'WarmEnabled': True,
            'WarmCount': 123,
        })
        break
```

```
    elif dry_run_status == 'failed':
        validation_failures_list =
dry_run_progress_response.DryRunProgressStatus.ValidationFailures
        for item in validation_failures_list:
            print(f"Code: {item['Code']}, Message: {item['Message']}")
        break

    retry_count += 1
    time.sleep(30)
```

追蹤組態變更

您可以一次請求一個組態變更，或在單一請求中將多個變更分組。使用主控台中的網域處理狀態和組態變更狀態欄位來追蹤組態變更。請求其他變更Active之前，請等待網域狀態變成 。

網域可以有下列處理狀態：

- Active – 沒有正在進行的組態變更。您可以提交新的組態變更請求。
- Creating – 正在建立網域。
- Modifying – 組態變更正在進行中，例如新增資料節點、EBS、gp3、IOPS 佈建或設定 KMS 金鑰。
- Upgrading engine version – 引擎版本升級正在進行中。
- Updating service software – 服務軟體更新正在進行中。
- Deleting – 正在刪除網域。
- Isolated – 網域已暫停。

網域可以有下列組態變更狀態：

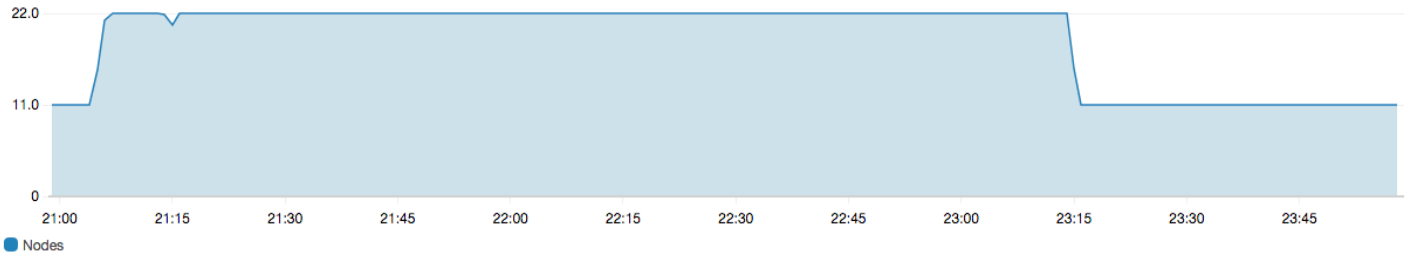
- Pending – 已提交組態變更請求。
- Initializing – 服務正在初始化組態變更。
- Validating – 服務正在驗證請求的變更和資源。
- Awaiting user inputs – 服務預期組態變更會繼續，例如執行個體類型變更。您可以編輯組態變更。
- Applying changes – 服務正在套用請求的組態變更。
- Cancelled – 組態變更已取消。選擇取消復原所有變更。

- `Completed` – 請求的組態變更已成功完成。
- `Validation failed` – 請求的組態變更無法完成。未套用任何組態變更。

> **Note**
>
> 驗證失敗可能是網域中存在的紅色索引、所選執行個體類型無法使用或磁碟空間不足的結果。如需驗證錯誤的清單，請參閱 the section called “對驗證錯誤進行疑難排解”。在驗證失敗事件期間，您可以取消、重試或編輯組態變更。

當組態變更完成時，網域狀態會變更回 `Active`。

您可以檢閱叢集運作狀態和 Amazon CloudWatch 指標，並查看叢集中的節點數量會在網域更新發生時暫時增加，通常是加倍。在下圖中，您可以看到在組態變更期間節點數量從 11 加倍到 22，而在更新完成時恢復回 11。



此暫時增加可能形成叢集專用主節點的負擔，突然可能會有許多節點需要管理。搜尋和索引延遲也可能因此而增加，因為 OpenSearch Service 會將資料從舊叢集複製到新叢集。在叢集務必維持足夠的容量，以處理與這些藍/綠部署相關的負荷。

> **Important**
>
> 組態變更和服務維護期間，您不需要支付任何額外費用。您也只需要針對您為叢集請求的節點數付費。如需詳細規格，請參閱the section called “組態變更的費用”。

為防止專用主節點超載，您可以利用 Amazon CloudWatch 指標監控使用量。如需建議的最大值，請參閱the section called “建議 CloudWatch 警示”。

組態變更的階段

在您啟動組態變更後，OpenSearch Service 會執行一系列步驟來更新您的網域。您可以在 主控台的組態變更狀態下檢視組態變更的進度。更新會經歷的確切步驟依您正在執行的變更類型而定。您也可使用 DescribeDomainChangeProgress API 操作來監控組態變更。

以下是更新在組派變更過程中可能會經歷的階段：

階段名稱	Description
驗證	驗證網域是否有資格進行更新，並在必要時顯示驗證問題。
建立新環境	完成必要的先決條件並建立所需資源，以開始進行藍/綠部署。
佈建新節點	在新環境中建立一組新的執行個體。
新節點上的流量路由	將流量重新引導至新建立的資料節點。
舊節點上的流量路由	停用舊資料節點上的流量。
準備要移除的節點	準備移除節點。僅在您縮減網域時 (例如，從 8 個節點減少至 6 個節點)，系統

階段名稱	Description
	才會執行此步驟。
將碎片複製到新節點	將碎片從舊節點移動至新節點。
終止節點	在移除碎片後，終止和刪除舊節點。
刪除舊資源	刪除與舊環境 (例如負載平衡器) 相關聯的資源。
動態更新	已在更新無需藍/綠部署且可動態套用更新時顯示。
套用專用主要相關變更	在專用主執行個體類型或計數變更時顯示。
套用磁碟區相關的變更	在磁碟區大小、類型、IOPS 和輸送量變更時顯示。

藍/綠部署的效能影響

在藍/綠部署期間，您的 Amazon OpenSearch Service 叢集可用於傳入的搜尋和索引請求。不過，您可能會遇到下列效能問題：

- 由於叢集要管理的節點越多，領導節點的使用量暫時增加。
- 隨著 OpenSearch Service 將資料從舊節點複製到新節點，增加搜尋和索引延遲。
- 隨著叢集負載在藍/綠部署期間增加，對傳入請求的拒絕增加。
- 為了避免延遲問題和請求拒絕，您應該在叢集運作狀態良好且網路流量低時執行藍/綠部署。

組態變更的費用

如果變更網域的組態，OpenSearch Service 會建立新的叢集，如 the section called “組態變更” 中所述。在將舊的遷移到新的期間，您需要支付以下費用：

- 如果您變更執行個體類型，會收取兩個叢集第一個小時的費用。在第一個小時後，則只會收取新叢集的費用。EBS 磁碟區不會收取兩次費用，因為它們是叢集的一部分，因此它們的計費會按執行個體計費。

  範例：您將組態從三個 `m3.xlarge` 執行個體變更為四個 `m4.large` 執行個體。對於第一個小時，您會被收取兩個叢集 (3 * `m3.xlarge` + 4 * `m4.large`) 的費用。在第一個小時後，則只會收取新叢集 (4 * `m4.large`) 的費用。
- 如果您不變更執行個體類型，您只會被收取最大叢集第一個小時的費用。在第一個小時後，則只會收取新叢集的費用。

  範例：您將組態從六個 `m3.xlarge` 執行個體變更為三個 `m3.xlarge` 執行個體。對於第一個小時，您會被收取最大叢集 (6 * `m3.xlarge`) 的費用。在第一個小時後，則只會收取新叢集 (3 * `m3.xlarge`) 的費用。

對驗證錯誤進行疑難排解

當您啟動組態變更或者執行 OpenSearch 或 Elasticsearch 版本升級時，OpenSearch Service 首先會執行一系列驗證檢查，以確保您的網域有資格進行更新。如果其中任何一項檢查失敗，您會在主控台中收到通知，其中包含您必須在更新網域之前解決的特定問題。

如果網域保持無法使用狀態超過 60 天，OpenSearch Service 會隔離您的網域。我們會傳送通知提醒給您，以解決這些問題。如果您未在此時間範圍內修正問題，OpenSearch Service 會刪除您的網域及其資料。

下表列出 OpenSearch Service 可能出現的網域問題，以及解決這些問題的步驟。

問題	錯誤碼	疑難排解步驟
找不到安全群組	SecurityGroupNotFound	與您的 OpenSearch Service 網域相關聯的安全群組不存在。若要解決此問題，請使用指定的名稱建立安全群組。
找不到子網	SubnetNotFound	與您的 OpenSearch Service 網域相關聯的子網不存在。若要解決此問題，在您的 VPC 中建立子網。
未設定服務連結角色	SLRNotConfigured	未設定 OpenSearch Service 的服務連結角色。服務連結角色由 OpenSearch Service 預先定義，並包含服務代表您呼叫其他 AWS 服務所需的所有許可。如果角色不存在，則可能需要手動建立。
IP 地址不足	InsufficientFreeIPsForSubnets	一個或多個 VPC 子網沒有足夠的 IP 地址來更新您的網域。若要計算您需要多少 IP 地址，請參閱 the section called “在 VPC 子網路中保留 IP 地址”。
Cognito 使用者集區不存在	CognitoUserPoolNotFound	OpenSearch Service 找不到 Amazon Cognito 使用者集區。確認您已建立一個且具有正確的 ID。若要尋找 ID，您可以使用 Amazon Cognito 主控台或以下 AWS CLI 命令： `aws cognito-idp list-user-pools --max-results 60 --region us-east-1`
Cognito 身分集區不存在	CognitoIdentityPoolNotFound	OpenSearch Service 找不到 Cognito 身分集區。確認您已建立一個且具有正確的 ID。若要尋找 ID，您可以使用 Amazon Cognito 主控台或以下 AWS CLI 命令： `aws cognito-identity list-identity-pools --max-results 60 --region us-east-1`
找不到使用者集區的 Cognito 網域	CognitoDomainNotFound	使用者集區沒有網域名稱。您可以使用 Amazon Cognito 主控台或下列 AWS CLI 命令來設定： `aws cognito-idp create-user-pool-domain --domain my-domain --user-pool-id id`

問題	錯誤碼	疑難排解步驟
未設定 Cognito 角色	`CognitoRoleNotConfigured`	未設定 IAM 角色，該角色可授予 OpenSearch Service 許可以設定 Amazon Cognito 使用者集區和身分集區，並使用它們進行身分驗證。使用適當的許可集合和信任關係來設定角色。您可以使用 主控台，為您建立預設 CognitoAccessForAmazonOpenSearch 角色，也可以使用 AWS CLI 或 AWS SDK 手動設定角色。
無法描述使用者集區	`UserPoolNotDescribable`	指定的 Amazon Cognito 角色沒有許可，無法描述與您的網域相關聯之使用者集區。請確定角色許可政策允許 `cognito-identity:DescribeUserPool` 動作。請參閱 the section called “關於 CognitoAccessForAmazonOpenSearch 角色” 了解完整的許可政策。
無法描述身分集區	`IdentityPoolNotDescribable`	指定的 Amazon Cognito 角色沒有許可，無法描述與您的網域相關聯之身分集區。請確定角色許可政策允許 `cognito-identity:DescribeIdentityPool` 動作。請參閱 the section called “關於 CognitoAccessForAmazonOpenSearch 角色” 了解完整的許可政策。
無法描述使用者集區和身分集區	`CognitoPoolsNotDescribable`	指定的 Amazon Cognito 角色沒有許可，無法描述與您的網域相關聯之使用者集區和身分集區。請確定角色許可政策允許 `cognito-identity:DescribeIdentityPool` 和 `cognito-identity:DescribeUserPool` 動作。請參閱 the section called “關於 CognitoAccessForAmazonOpenSearch 角色” 了解完整的許可政策。
未啟用 KMS 金鑰	`KMSKeyNotEnabled`	用於加密網域的 AWS Key Management Service (AWS KMS) 金鑰已停用。立即重新啟用金鑰。
自訂憑證未處於 ISSUED (已發行) 狀態	`InvalidCertificate`	如果您的網域使用自訂端點，您可以透過在 AWS Certificate Manager (ACM) 中產生 SSL 憑證或匯入您自己的憑證來保護它。憑證狀態必須為 Issued (已發佈)。如果您收到此錯誤，在 ACM 主控台中檢查憑證的狀態。如果狀態為 Expired (已過期)、Failed (失敗)、Inactive (非作用中) 或者 Pending validation (待定驗證)，請參閱 ACM 疑難排解文件以解決該問題。

問題	錯誤碼	疑難排解步驟
沒有足夠的容量來啟動選擇的執行個體類型	InsufficientInstanceCapacity	請求的執行個體類型容量不可用。例如，您可能已經請求五個 i3.16xlarge.search 節點，但 OpenSearch Service 沒有足夠的 i3.16xlarge.search 主機可用，因此無法滿足請求。檢查 OpenSearch Service 中支援的執行個體類型並選擇不同的執行個體類型。
叢集中的紅色索引	RedCluster	叢集中的一個或多個索引具有紅色狀態，這導致整體紅色叢集狀態。若要進行疑難排解並修正此問題，請參閱 the section called “紅色叢集狀態”。
對記憶體斷路器的請求太多	TooManyRequests	您的網域有太多的搜尋請求和寫入請求，因此 OpenSearch Service 無法更新其組態。您可以減少請求數量，將執行個體垂直擴展到 64 GiB 的 RAM，或者透過新增執行個體進行水平擴展。
新組態無法存放資料 (磁碟空間不足)	InsufficientStorageCapacity	設定的儲存空間大小無法存放您網域上的所有資料。若要解決此問題，請選擇更大的磁碟區、刪除未使用的索引或增加叢集中的節點數目，以立即釋放磁碟空間。

固定到特定節點的碎片	ShardMovementBlocked	網域中的一個或多個索引會連接至特定節點，且無法重新指派。這很可能是因為您已設定碎片分配篩選，這可讓您指定允許哪些節點託管特定索引的碎片。
		
		若要解決此問題，請從所有受影響的索引中移除碎片分配篩選條件：
		
		`PUT my-index/_settings`
		`{`
		`  "settings": {`
		`  "index.routing.allocation.require._name": null`
		`  }`
		`}`

問題	錯誤碼	疑難排解步驟
新組態無法存放所有碎片 (碎片計數)	`TooManyShards`	您網域上的碎片計數太高，這會導致 OpenSearch Service 無法將它們移至新的組態。若要解決此問題，請新增與目前叢集節點相同的組態類型的節點，以便水平擴展您的網域。請注意，EBS 磁碟區大小上限取決於節點的執行個體類型。 若要避免將來發生此問題，請參閱 the section called “選擇碎片數” 並定義適用於您的使用案例的碎片策略。
與您的網域相關聯的子網路不支援 IPv4 地址	`ResultCodeIPv4BlockNotExists`	若要解決此問題，請根據網域的已設定 IP 地址類型，建立子網路或更新 VPC 中的現有子網路。如果您的網域使用僅限 IPv4 的地址類型，請使用IPv4-only的子網路。如果您的網域使用雙堆疊模式，請使用雙堆疊子網路。
與您的網域相關聯的子網路不支援 IPv6 地址	`ResultCodeIPv6BlockNotExists`	若要解決此問題，請根據網域的已設定 IP 地址類型，建立子網路或更新 VPC 中的現有子網路。如果您的網域使用僅限 IPv4 的地址類型，請使用IPv4-only的子網路。如果您的網域使用雙堆疊模式，請使用雙堆疊子網路。

Amazon OpenSearch Service 中的服務軟體更新

> **Note**
>
> 如需每個主要 （非修補程式） 服務軟體更新中所做的變更和新增的說明，請參閱版本備註。

Amazon OpenSearch Service 會定期發行服務軟體更新以新增功能或強化您的網域。主控台中的 Notifications (通知) 面板是查看更新是否可用或檢查更新狀態的最簡單方法。每個通知都包含有關服務軟體更新的詳細資訊。所有服務軟體更新都使用藍/綠部署，將停機時間降至最低。

服務軟體更新與 OpenSearch Service 版本升級不同。如需升級至較新版本 OpenSearch Service 的相關資訊，請參閱 the section called “升級網域”。

OpenSearch Service 會要求您在所需的服務軟體更新可供使用的 30 天內套用。這些更新對於維護安全合規至關重要。

如果您未在 30 天期間內套用必要的更新，您將在 30 天內每 15 天收到提醒通知。在此期間未合規之後，您的網域將會隔離，並產生下列效果：

- 已移除對您網域的所有網路存取
- 網域狀態變更為隔離
- 在您套用必要的更新之前，網域仍然無法使用

在隔離期間，您將繼續每 15 天收到提醒通知，為期 60 天。如果您在此期間未套用必要的更新，您的 OpenSearch Service 網域和所有相關資料都會永久刪除。如需詳細資訊，請參閱疑難排解驗證錯誤。

選用與必要的更新

OpenSearch Service 有兩種廣泛的服務軟體更新類別：

選用更新

選用的服務軟體更新通常包括增強功能和對新功能或功能的支援。您的網域不會強制執行選用更新，也沒有安裝它們的硬性截止日期。更新可用性會透過電子郵件和主控台通知進行通訊。您可以選擇立即套用更新，或將其重新排程為更適當的日期和時間。您也可以在網域的離峰時段進行排程。大多數軟體更新都是選用的。

無論您是否排程更新，如果您在網域上進行造成藍/綠部署的變更，OpenSearch Service 都會為您自動更新服務軟體。

您可以將網域設定為在離峰時間自動套用選用更新。開啟此選項時，OpenSearch Service 會在選用更新可用後等待至少 13 天，然後在七天後排程更新。排定更新時，您會收到主控台通知，您可以選擇在稍後日期重新排定更新。

若要開啟自動軟體更新，請選取建立或更新網域時啟用自動軟體更新。若要使用 設定相同的設定 AWS CLI，請在建立或更新網域`true`時將 `--software-update-options`設定為 。

必要的更新

必要的服務軟體更新通常包括重要的安全性修正或其他強制性更新，以確保網域的持續完整性和功能。所需更新的範例包括 Log4j 常見漏洞與暴露 (CVEs) 和執行個體中繼資料服務第 2 版 (IMDSv2) 的強制執行。一年中的強制性更新數量通常少於三個。

OpenSearch Service 會自動排程這些更新，並在排定的更新七天前透過電子郵件和主控台通知通知您。您可以選擇立即套用更新，或在允許的時間範圍內重新排程更適當的日期和時間。您也可以在網域的下一個離峰時段進行排程。如果您未對必要的更新採取任何動作，且未進行任何導致藍/綠部署的

網域變更，OpenSearch Service 可以在網域的離峰時段內超過指定截止日期（通常是自可用性起 14 天）的任何時間啟動更新。

無論何時排定更新，如果您變更網域而導致藍/綠部署，OpenSearch Service 會自動為您更新網域。

修補程式更新

以 "-P" 和數字結尾的服務軟體版本，例如 R20211203-*P4*，是修補程式版本。修補程式可能包括效能改進、小錯誤修復、安全修復或狀態改善。修補程式版本不包含新功能或重大變更，而且通常不會對使用者造成直接或明顯的影響。服務軟體通知會告訴您修補程式版本是選用還是強制性。

考量事項

在決定是否要更新網域時，請考慮下列各項：

- 手動更新網域可讓您更快速地利用新功能。當您選擇 Update (更新) 時，OpenSearch Service 會將請求置於佇列中，並在有時間時開始更新。
- 初始化服務軟體更新時，OpenSearch Service 會在更新開始和完成時傳送通知。
- 軟體更新使用藍/綠部署盡量減少停機時間。更新可能會暫時損耗叢集的專用主節點，因此請務必維持足夠的容量來處理相關的額外負荷。
- 更新通常在幾分鐘內完成，但如果您的系統負載過重，也可能需要數小時甚至數天的時間。請考慮在設定的離峰時段期間更新您的網域，以避免長時間更新。

啟動服務軟體更新

您可以透過 OpenSearch Service 主控台、 AWS CLI或其中一個 SDKs 請求服務軟體更新。

主控台

請求服務軟體更新

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 選取網域名稱以開啟其組態。
3. 選擇動作、更新並選取下列其中一個選項：
    - 立即套用更新 - 如果有可用的容量，立即排定在目前小時內執行的動作。如果無法使用容量，我們會提供其他可用的時段供您選擇。

- 在離峰時段排程 – 只有在網域啟用離峰時段時才可用。排程在網域設定的離峰時段期間進行更新。無法保證會在下一個立即時段進行更新。視容量而定，它可能會在後續幾天內發生。如需詳細資訊，請參閱the section called “離峰時段”。
- 排定特定日期和時間 – 排定在特定日期和時間進行更新。如果您指定的時間因容量原因而無法使用，您可以選取不同的時段。

如果您將更新排程在稍後的日期（在網域的離峰時段內或之外），您可以隨時重新排程更新。如需說明，請參閱the section called “重新排程動作”。

4. 選擇確認。

AWS CLI

傳送 start-service-software-update AWS CLI 請求以啟動服務軟體更新。此範例會立即將更新新增至佇列：

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "NOW"
```

回應：

```
{
    "ServiceSoftwareOptions": {
        "CurrentVersion": "R20220928-P1",
        "NewVersion": "R20220928-P2",
        "UpdateAvailable": true,
        "Cancellable": true,
        "UpdateStatus": "PENDING_UPDATE",
        "Description": "",
        "AutomatedUpdateDate": "1969-12-31T16:00:00-08:00",
        "OptionalDeployment": true
    }
}
```

> **Tip**
>
> 請求更新後，您有很短的時間可以取消更新。此PENDING_UPDATE狀態的持續時間可能很大，取決於您的 AWS 區域 和 OpenSearch Service 正在執行的並行更新數量。若要取消更新，請使用 主控台或 `cancel-service-software-update` AWS CLI 命令。

如果請求因 而失敗`BaseException`，表示您指定的時間因容量原因而無法使用，您必須指定不同的時間。OpenSearch Service 在回應中提供替代的可用插槽建議。

AWS SDKs

此範例 Python 指令碼使用來自 的 describe_domain 和 start_service_software_update 方法 適用於 Python (Boto3) 的 AWS SDK ，來檢查網域是否符合服務軟體更新的資格，如果符合，則啟動更新。您必須提供 `domain_name` 的值。

```
import boto3
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)

domain_name = ''  # The name of the domain to check and update

client = boto3.client('opensearch', config=my_config)


def getUpdateStatus(client):
    """Determines whether the domain is eligible for an update"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    sso = response['DomainStatus']['ServiceSoftwareOptions']
    if sso['UpdateStatus'] == 'ELIGIBLE':
```

```
        print('Domain [' + domain_name + '] is eligible for a service software update
 from version ' +
              sso['CurrentVersion'] + ' to version ' + sso['NewVersion'])
        updateDomain(client)
    else:
        print('Domain is not eligible for an update at this time.')


def updateDomain(client):
    """Starts a service software update for the eligible domain"""
    response = client.start_service_software_update(
        DomainName=domain_name
    )
    print('Updating domain [' + domain_name + '] to version ' +
          response['ServiceSoftwareOptions']['NewVersion'] + '...')
    waitForUpdate(client)


def waitForUpdate(client):
    """Waits for the domain to finish updating"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    status = response['DomainStatus']['ServiceSoftwareOptions']['UpdateStatus']
    if status == 'PENDING_UPDATE' or status == 'IN_PROGRESS':
        time.sleep(30)
        waitForUpdate(client)
    elif status == 'COMPLETED':
        print('Domain [' + domain_name +
              '] successfully updated to the latest software version')
    else:
        print('Domain is not currently being updated.')

def main():
    getUpdateStatus(client)
```

在離峰時段期間排程軟體更新

在 2023 年 2 月 16 日之後建立的每個 OpenSearch Service 網域，其每日 10 小時時段介於當地時間下午 10：00 到上午 8：00，我們認為是離峰時段。OpenSearch Service 使用此時段來排程網域的服務軟體更新。離峰更新有助於在較高的流量期間，將叢集專用主節點上的壓力降至最低。未經您的同意，OpenSearch Service 無法在此 10 小時時段之外啟動更新。

- 對於選用更新，OpenSearch Service 會通知您更新的可用性，並提示您在即將到來的離峰時段排程更新。
- 對於必要的更新，OpenSearch Service 會在即將到來的離峰時段自動排程更新，並提前三天通知您。您可以重新排程更新（適用於離峰時段內外），但僅限於完成更新所需的時間範圍內。

對於每個網域，您可以選擇以自訂時間覆寫預設的下午 10：00 開始時間。如需說明，請參閱the section called “設定自訂離峰時段”。

主控台

在即將來臨的離峰時段中排程更新

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 選取網域名稱以開啟其組態。
3. 選擇動作、更新。
4. 選取在離峰時段排程。
5. 選擇確認。

您可以在離峰時段索引標籤上檢視排程動作，並隨時重新排程。請參閱 the section called “檢視排程動作”。

CLI

若要在即將來臨的離峰時段使用 排程更新 AWS CLI，請傳送 StartServiceSoftwareUpdate 請求`OFF_PEAK_WINDOW`，並為 `--schedule-at` 參數指定 ：

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "OFF_PEAK_WINDOW"
```

監控服務軟體更新

OpenSearch Service 會在服務軟體更新可用、必要、已啟動、已完成或失敗時傳送通知。您可以在 OpenSearch Service 主控台的通知面板中檢視通知。如果更新為可選，則通知嚴重性為 `Informational`；如果必須更新，則為 `High`。

OpenSearch Service 也會將服務軟體事件傳送至 Amazon EventBridge。您可以使用 EventBridge 設定傳送電子郵件的規則，或在接收事件時執行特定動作。如需逐步解說範例，請參閱 the section called “教學課程：傳送可用更新的 SNS 提醒”。

若要查看傳送至 Amazon EventBridge 的每個服務軟體事件的格式，請參閱 the section called “服務軟體更新事件”。

網域不符合更新條件時，

如果您的網域處於以下任何一種狀態，則不符合服務軟體更新的資格：

State	Description
處理中的網域	組態變更中的網域 請於操作完成後檢查更新資格。
紅色叢集狀態	叢集中一或多個索引是紅色的。如需疑難排解步驟，請參閱the section called “紅色叢集狀態”。
高錯誤率	OpenSearch 叢集在嘗試處理請求時，傳回大量的 5xx 錯誤。此問題通常是因為太多同時讀寫請求造成。請考慮降低叢集流量或擴展網域。
分割大腦	分割大腦表示 OpenSearch 叢集有不只一個主節點，且已分為兩個叢集，這兩個叢集將不會自行重新連結。您也可以使用建議數量的專用主節點來避免大腦分割。如需從大腦分割中恢復的協助，請聯絡 支援。
Amazon Cognito 整合問題	您的網域使用 OpenSearch Dashboards 身分驗證，而 OpenSearch Service 找不到一個或多個 Amazon Cognito 資源。如果缺少 Amazon Cognito 使用者集區，通常就會發生此問題。若要修正問題，請重新建立缺失的資源並設定由 OpenSearch Service 網域使用該項資源。
其他 服務問題	OpenSearch Service 本身的問題可能會使您的網域顯示為不符合更新資格。如果上述情況都不適用於您的網域且問題持續超過一天，請聯絡 支援。

定義 Amazon OpenSearch Service 的離峰時段

當您建立 Amazon OpenSearch Service 網域時，您會定義每日 10 小時的時段，這些時段會被視為離峰時數。OpenSearch Service 會盡可能使用此時段來排程服務軟體更新和自動調整最佳化，以便在流

量時間相對較低時需要藍/綠部署。藍/綠是指為網域更新建立新的環境，並在這些更新完成後將使用者路由到新環境的程序。

雖然藍/綠部署不中斷，但為了將資源耗用以進行藍/綠部署時的任何潛在效能影響降至最低，建議您在網域設定的離峰時段期間排程這些部署。節點替換等更新，或需要立即部署到網域的更新，請勿使用離峰時段。

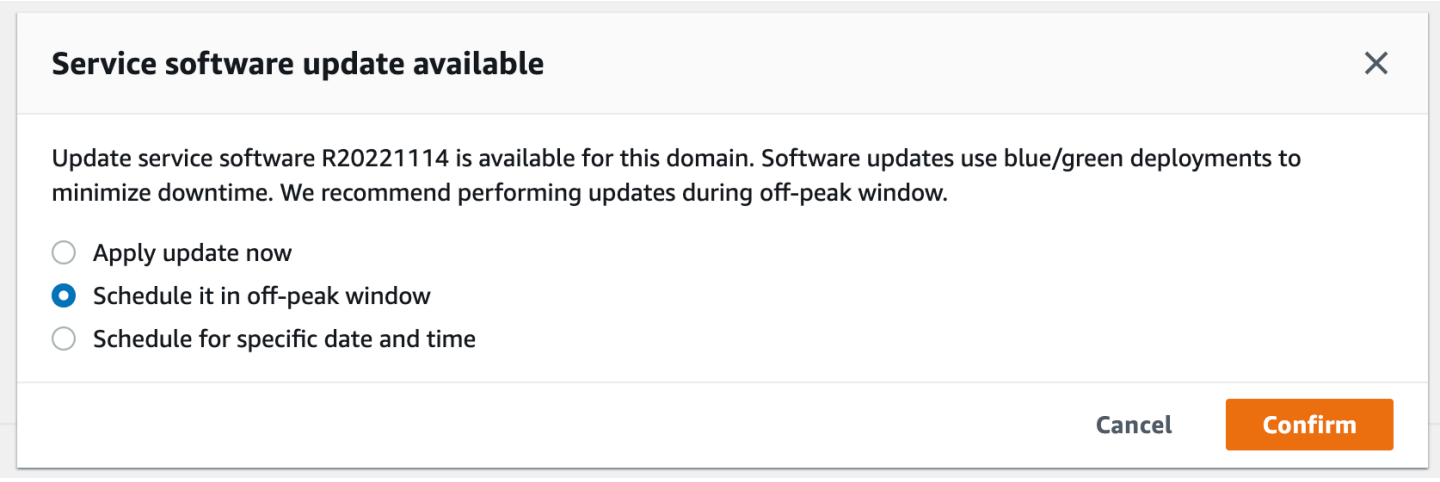
您可以修改離峰時段的開始時間，但無法修改時段的長度。

> **Note**
>
> 離峰時段於 2023 年 2 月 16 日推出。在此日期之前建立的所有網域預設會停用離峰時段。您必須手動啟用和設定這些網域的離峰時段。在此日期之後建立的所有網域預設都會啟用離峰時段。您無法在網域啟用後停用網域的離峰時段。

離峰服務軟體更新

OpenSearch Service 有兩種廣泛的服務軟體更新：選用和必要。這兩種類型都需要藍/綠部署。您的網域不會強制執行選用更新，但如果您在指定的截止日期（通常是自可用性起兩週）之前未採取任何動作，則會自動安裝必要的更新。如需詳細資訊，請參閱the section called “選用與必要的更新”。

當您啟動選用更新時，您可以選擇立即套用更新、排程後續離峰時段，或指定自訂日期和時間來套用更新。



對於必要的更新，OpenSearch Service 會在離峰時段自動排程日期和時間，以執行更新。您會在排程更新前三天收到通知，而且可以選擇在所需的部署期間內重新排程稍後的日期和時間。如需說明，請參閱the section called “重新排程動作”。

離峰自動調整最佳化

先前，自動調整使用維護時段來排程需要藍/綠部署的變更。在推出離峰時段之前已啟用自動調整和維護時段的網域將繼續使用這些更新維護時段，除非您遷移它們以使用離峰時段。

我們建議您遷移網域以使用離峰時段，因為它用於排程網域上的其他活動，例如服務軟體更新。如需說明，請參閱the section called “從自動調整維護時段遷移”。將網域遷移至離峰時段後，您無法還原至使用維護時段。

2023 年 2 月 16 日之後建立的所有網域將使用離峰時段，而不是傳統維護時段來排程藍/綠部署。您無法停用網域的離峰時段。如需需要藍/綠部署的自動調整最佳化清單，請參閱 the section called “變更類型”。

啟用離峰時段

在 2023 年 2 月 16 日之前建立的任何網域（引進離峰時段時）預設會停用此功能。您必須為這些網域手動啟用它。您無法在啟用後停用離峰時段。

主控台

啟用網域的離峰時段

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 選取網域名稱以開啟其組態。
3. 導覽至離峰視窗索引標籤，然後選擇編輯。
4. 以國際標準時間 (UTC) 指定自訂開始時間。例如，若要在美國西部（奧勒岡）區域中設定開始時間為下午 11：30，請指定 07：30。
5. 選擇儲存變更。

CLI

若要使用 修改離峰時段 AWS CLI，請傳送 UpdateDomainConfig 請求：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'Enabled=true,
 OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

如果您未指定自訂時段開始時間，則預設為 00：00 UTC。

設定自訂離峰時段

您可以在國際標準時間 (UTC) 中為您的網域指定自訂離峰時段。例如，如果您希望美國東部（維吉尼亞北部）區域中網域的離峰時段於下午 11：00 開始，則需指定 04：00 UTC。

主控台

修改網域的離峰時段

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 選取網域名稱以開啟其組態。
3. 導覽至離峰視窗索引標籤。您可以檢視設定的離峰時段，以及網域即將進行的排程動作清單。
4. 選擇編輯，並以 UTC 指定新的開始時間。例如，若要設定美國東部（維吉尼亞北部）區域的開始時間為下午 9：00，請指定 02：00 UCT。
5. 選擇儲存變更。

CLI

若要使用 設定自訂離峰時段 AWS CLI，請傳送 UpdateDomainConfig 請求，並以 24 小時時間格式指定小時和分鐘。

例如，下列請求會將時段開始時間變更為 2：00 A.M. UTC：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

如果您未指定時段開始時間，則會針對建立網域 AWS 區域 的 ，預設為下午 10：00 本機時間。

檢視排程動作

您可以檢視每個網域目前排程、進行中或待定的所有動作。動作的嚴重性可以是 HIGH、 MEDIUM和 LOW。

動作可以有下列狀態：

- `Pending update` – 動作位於要處理的佇列中。
- `In progress` – 動作目前正在進行中。
- `Failed` – 動作無法完成。
- `Completed` – 動作已成功完成。
- `Not eligible` – 僅適用於服務軟體更新。無法繼續更新，因為叢集處於運作狀態不佳的狀態。
- `Eligible` – 僅適用於服務軟體更新。網域符合更新的資格。

主控台

OpenSearch Service 主控台會顯示網域組態中的所有排程動作，以及每個動作的嚴重性和目前狀態。

檢視網域的排程動作

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 選取網域名稱以開啟其組態。
3. 導覽至離峰視窗索引標籤。
4. 在排程動作下，檢視網域目前排程、進行中或待定的所有動作。

CLI

若要使用 檢視排程動作 AWS CLI，請傳送 ListScheduledActions 請求：

```
aws opensearch list-scheduled-actions \
  --domain-name my-domain
```

回應：

```
{
    "ScheduledActions": [
        {
            "Cancellable": true,
            "Description": "The Deployment type is : BLUE_GREEN.",
            "ID": "R20220721-P13",
            "Mandatory": false,
            "Severity": "HIGH",
```

```
            "ScheduledBy": "CUSTOMER",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
            "Type": "SERVICE_SOFTWARE_UPDATE",
        },
        {
            "Cancellable": true,
            "Description": "Amazon Opensearch will adjust the young generation JVM
 arguments on your domain to improve performance",
            "ID": "Auto-Tune",
            "Mandatory": true,
            "Severity": "MEDIUM",
            "ScheduledBy": "SYSTEM",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
            "Type": "JVM_HEAP_SIZE_TUNING",
        }
    ]
}
```

重新排程動作

OpenSearch Service 會通知您排定的服務軟體更新和自動調整最佳化。您可以選擇立即套用變更，或將其重新排程為稍後的日期和時間。

> **Note**
>
> OpenSearch Service 可以在您選取的一小時內排程動作。對於 exmple，如果您選擇在下午 5 點套用更新，則可以在下午 5 點到 6 點之間套用更新。

主控台

重新排程動作

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 選取網域名稱以開啟其組態。
3. 導覽至離峰視窗索引標籤。
4. 在排程動作下，選取動作，然後選擇重新排程。

5. 請選擇下列其中一個選項：

   - 立即套用更新 - 如果有可用的容量，立即排定在目前小時內執行的動作。如果無法使用容量，我們會提供其他可用的時段供您選擇。
   - 在離峰時段排程 - 標記要在即將來臨的離峰時段中收取的動作。我們無法保證將在下一個時段實作變更。視容量而定，它可能會在後續幾天內發生。
   - 重新排程此更新 - 可讓您指定套用變更的自訂日期和時間。如果您指定的時間因容量原因而無法使用，您可以選取不同的時段。
   - 取消排定的更新 - 取消更新。此選項僅適用於選用的服務軟體更新。它不適用於自動調整動作或強制性軟體更新。

6. 選擇儲存變更。

CLI

若要使用 重新排程動作 AWS CLI，請傳送 UpdateScheduledAction 請求。若要擷取動作 ID，請傳送`ListScheduledActions`請求。

下列請求會針對特定日期和時間重新排程服務軟體更新：

```
aws opensearch update-scheduled-action \
  --domain-name my-domain \
  --action-id R20220721-P13 \
  --action-type "SERVICE_SOFTWARE_UPDATE" \
  --desired-start-time 1677348395000 \
  --schedule-at TIMESTAMP
```

回應：

```
{
   "ScheduledAction": {
      "Cancellable": true,
      "Description": "Cluster status is updated.",
      "Id": "R20220721-P13",
      "Mandatory": false,
      "ScheduledBy": "CUSTOMER",
      "ScheduledTime": 1677348395000,
      "Severity": "HIGH",
      "Status": "PENDING_UPDATE",
      "Type": "SERVICE_SOFTWARE_UPDATE"
```

```
    }
}
```

如果請求因 而失敗`SlotNotAvailableException`，表示您指定的時間因容量原因而無法使用，您必須指定不同的時間。OpenSearch Service 在回應中提供替代的可用插槽建議。

從自動調整維護時段遷移

如果網域是在 2023 年 2 月 16 日之前建立的，則可以使用維護時段來排程需要藍/綠部署的自動調整最佳化。您可以遷移現有的自動調整網域，以改為使用離峰時段。

> **Note**
>
> 遷移網域以使用離峰時段後，您無法還原至使用維護時段。

主控台

遷移網域以使用離峰時段

1. 在 Amazon OpenSearch Service 主控台中，選取要開啟其組態的網域名稱。
2. 前往自動調整索引標籤，然後選擇編輯。
3. 選取遷移至離峰時段。
4. 對於開始時間 (UTC)，請以國際標準時間 (UTC) 提供離峰時段的每日開始時間。
5. 選擇儲存變更。

CLI

若要使用 從自動調整維護時段遷移至離峰時段 AWS CLI，請傳送 UpdateDomainConfig 請求：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=[]
```

必須開啟離峰時段，才能將網域從自動調整維護時段遷移至離峰時段。您可以在個別的請求或相同的請求中啟用離峰時段。如需說明，請參閱the section called “啟用離峰時段”。

Amazon OpenSearch Service 中的通知

Amazon OpenSearch Service 中的通知包含有關網域效能和運作狀態的重要資訊。OpenSearch Service 會通知您有關服務軟體更新、自動調整增強功能、叢集運作狀態事件和網域錯誤的資訊。通知適用於所有 OpenSearch 和 Elasticsearch OSS 版本。

您可以在 OpenSearch Service 主控台的 Notifications (通知) 面板中檢視通知。OpenSearch Service 的所有通知也會顯示在 Amazon EventBridge 中。如需通知與範例事件的完整清單，請參閱the section called “監控事件”。

開始使用通知

當您建立網域時，系統會自動啟用通知。前往 OpenSearch Service 主控台的通知面板，以監控並確認通知。每個通知都包含一些資訊，例如發佈時間、其相關網域、嚴重性和狀態等級以及簡短說明。您可以在主控台中檢視最多 90 天的歷史通知。

在存取 Notifications (通知) 面板或確認通知後，您可能會收到錯誤訊息，指出沒有執行 `es:ListNotificationsV2` 或 `es:UpdateNotificationStatus` 的許可。若要解決此問題，請在 IAM 中為您的使用者或角色提供下列許可：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "es:DescribeDomain",
      "es:ListDomainNames"
    ],
    "Resource": "arn:aws:es:*:111122223333:domain/*"
  }]
}
```

IAM 主控台會擲回可安全忽略的錯誤 (「IAM 無法辨識一或多個動作」)。您也可以將 `es:UpdateNotificationStatus` 動作限定於特定網域。如需詳細資訊，請參閱 the section called “政策元素參考”。

通知嚴重性

OpenSearch Service 中的通知可能是 informational，這與您已經採取的任何動作或您的網域操作有關，也可能是 actionable，這會要求您採取特定動作，例如套用強制安全性修補程式。每個通知都有相關聯的嚴重性，可能是 `Informational`、`Low`、`Medium`、`High` 或 `Critical`。下表提供各級嚴重性的摘要：

嚴重性	Description	範例
`Informational`	與您的網域操作相關的資訊。	• 可用的服務軟體更新 • 自動調整已開始
`Low`	建議的動作，但如果未採取任何動作，則不會對網域可用性或效能造成不利影響。	• 自動調整已取消 • 高碎片計數警告
`Medium`	如果沒有採取建議的動作，但所採取的動作需要一段較長的時間，則可能會產生影響。	• 服務軟體更新已失敗 • 已超過碎片數量限制
`High`	必須採取緊急動作，以避免不利影響。	• 需要服務軟體更新 • 無法存取 KMS 金鑰
`Critical`	需要立即採取動作，以避免不利影響或從中恢復。	目前無可用

EventBridge 事件範例

下列範例顯示傳送至 Amazon EventBridge 的 OpenSearch Service 通知事件。通知具有 `Informational` 嚴重性，因為更新是可選的：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
```

```
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] available."
  }
}
```

在 Amazon OpenSearch Service 中設定多可用區域網域

為避免因服務中斷造成的資料遺失並盡量減少 Amazon OpenSearch Service 叢集停機時間，您可以在相同區域中將節點發佈至跨兩個或三個可用區域，此組態就是所謂的多可用區域。可用區域是每個 AWS 區域內的隔離位置。

對於執行生產工作負載的網域，我們建議使用具有待命部署的異地同步備份選項，這會建立下列組態：

- 跨三個區域部署的網域。
- 專用主節點和資料節點的新一代執行個體類型。
- 三個專用主節點和三個（或三個的倍數）資料節點。
- 網域中每個索引至少有兩個複本，或三個資料複本的倍數（包括主節點和複本）。

本節其餘部分提供這些組態的說明和內容。

具有待命的異地同步備份

Multi-AZ with Standby 是 Amazon OpenSearch Service 網域的部署選項，可提供 99.99% 的可用性、生產工作負載的一致效能，以及簡化的網域組態和管理。當您將異地同步備份與待命搭配使用時，網域對基礎設施故障具有彈性，不會影響效能或可用性。此部署選項透過強制許多最佳實務來實現此標準，例如指定的資料節點計數、主節點計數、執行個體類型、複本計數、軟體更新設定，以及開啟自動調整。

> **Note**
>
> 具有待命的異地同步備份適用於 OpenSearch 1.3 版及更新版本，並且需要至少具有三個可用區域的區域。

當您將多可用區域與待命搭配使用時，OpenSearch Service 會跨三個可用區域建立網域，每個區域都包含完整的資料副本，並將資料平均分散在每個區域中。您的網域會將其中一個區域中的節點保留為待命，這表示它們不提供搜尋請求。當 OpenSearch Service 在基礎基礎設施中偵測到故障時，會在一分鐘內自動啟用待命節點。網域會繼續提供索引和搜尋請求，而任何影響僅限於執行容錯移轉所需的時間。資料或資源不會重新分佈，這會導致叢集效能不受影響，也不會降低可用性的風險。具有待命的異地同步備份可免費使用。

您有兩個選項可在 上建立具有待命狀態的網域 AWS 管理主控台。首先，您可以使用輕鬆建立方法建立網域，OpenSearch Service 會自動使用預先定義的組態，其中包括下列項目：

- 三個可用區域，一個做為待命
- 三個專用主節點和資料節點
- 在網域上啟用自動調整
- 資料節點的 GP3 儲存體

您也可以選擇標準建立方法，然後選取具有待命的網域做為您的部署選項。這可讓您自訂網域，同時仍然強制待命的關鍵功能，例如三個區域和三個主節點。我們建議選擇的資料節點計數是三的倍數（可用區域的數量）。

建立網域後，您可以導覽至網域詳細資訊頁面，並在叢集組態索引標籤中確認 3-AZ（含待命）顯示在可用區域下方。

如果您在將現有網域遷移至具有待命的異地同步備份時遇到問題，請參閱故障診斷指南中的遷移至具有待命的異地同步備份時發生錯誤。

限制

當您使用具有待命的異地同步備份設定網域時，請考慮下列限制：

- 節點上的碎片總數不能超過 1000，叢集上的碎片總數不能超過 75000，單一碎片的大小不能超過 65 GB。
- 具有待命的異地同步備份僅適用於 `m5`、`c5`、、`r5r6g`、`c6g`、`m6g`、`r6gd`、`r7g`、`c7g`、`m7g`、、`r7gd`和 `i3`執行個體類型。如需支援執行個體的詳細資訊，請參閱支援的執行個體類型。
- 您只能使用佈建 IOPs SSD、一般用途 SSD (GP3) 或執行個體後端儲存搭配待命。
- 如果您在具有待命網域的多可用區域上啟用 UltraWarm，暖節點的數量必須是正在使用的可用區域的倍數。

無待命的異地同步備份

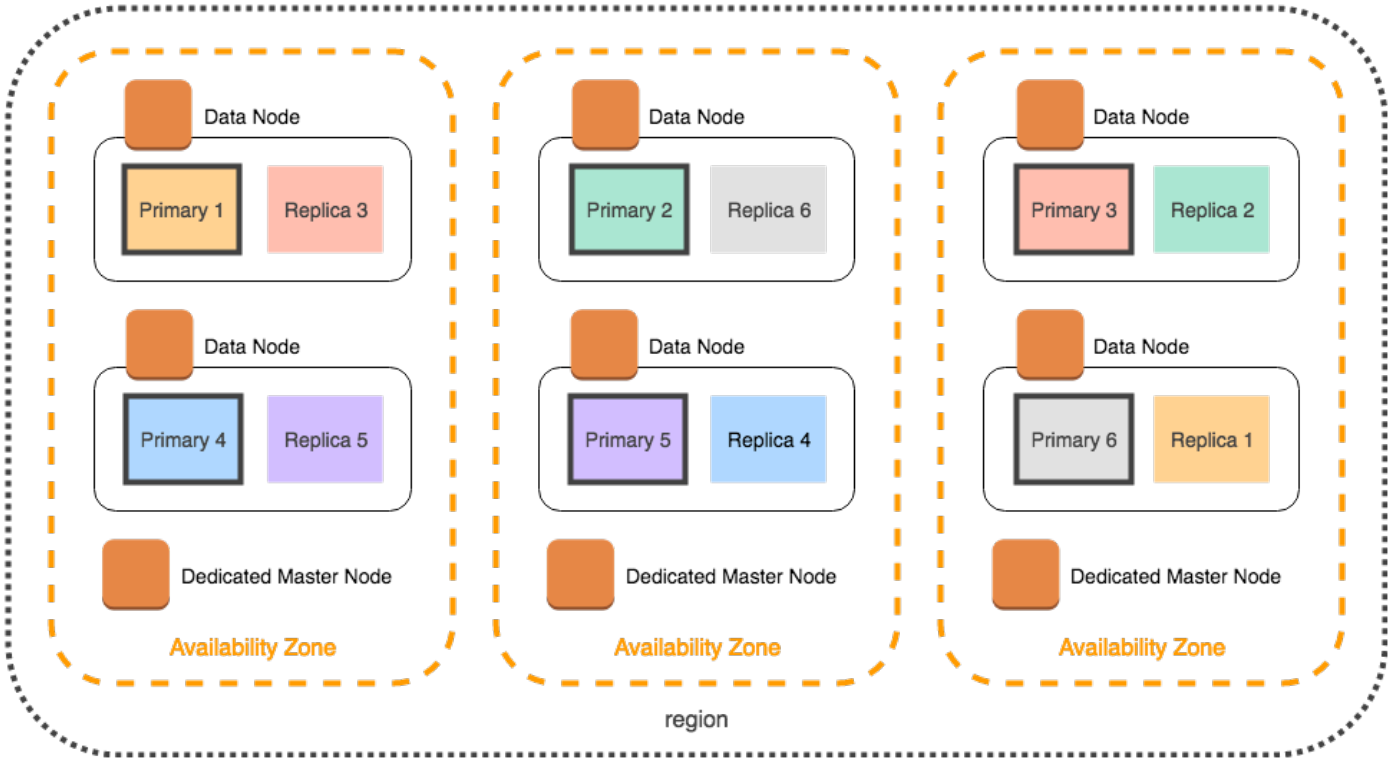
OpenSearch Service 仍然支援沒有待命的異地同步備份，可提供 99.9% 的可用性。節點會分散到可用區域 (N)，而可用性取決於可用區域和資料副本的數量。使用待命時，您必須使用最佳實務來設定網域，而不需要待命，您可以選擇自己的可用區域、節點和複本數量。除非您有現有的工作流程會因為建立具有待命的網域而中斷，否則我們不建議使用此選項。

如果您選擇此選項，仍建議您選取三個可用區域，以保持對節點、磁碟和單一可用區域故障的彈性。發生故障時，叢集會將資料重新分配到剩餘的資源，以維持可用性和備援。此資料移動會增加叢集上的資源使用量，並可能影響效能。如果叢集大小不正確，可能會遇到可用性降低，這主要會破壞多可用區域的目的。

在 上設定無待命網域的唯一方法是 AWS 管理主控台 選擇標準建立方法，然後選取無待命網域做為您的部署選項。

碎片分佈

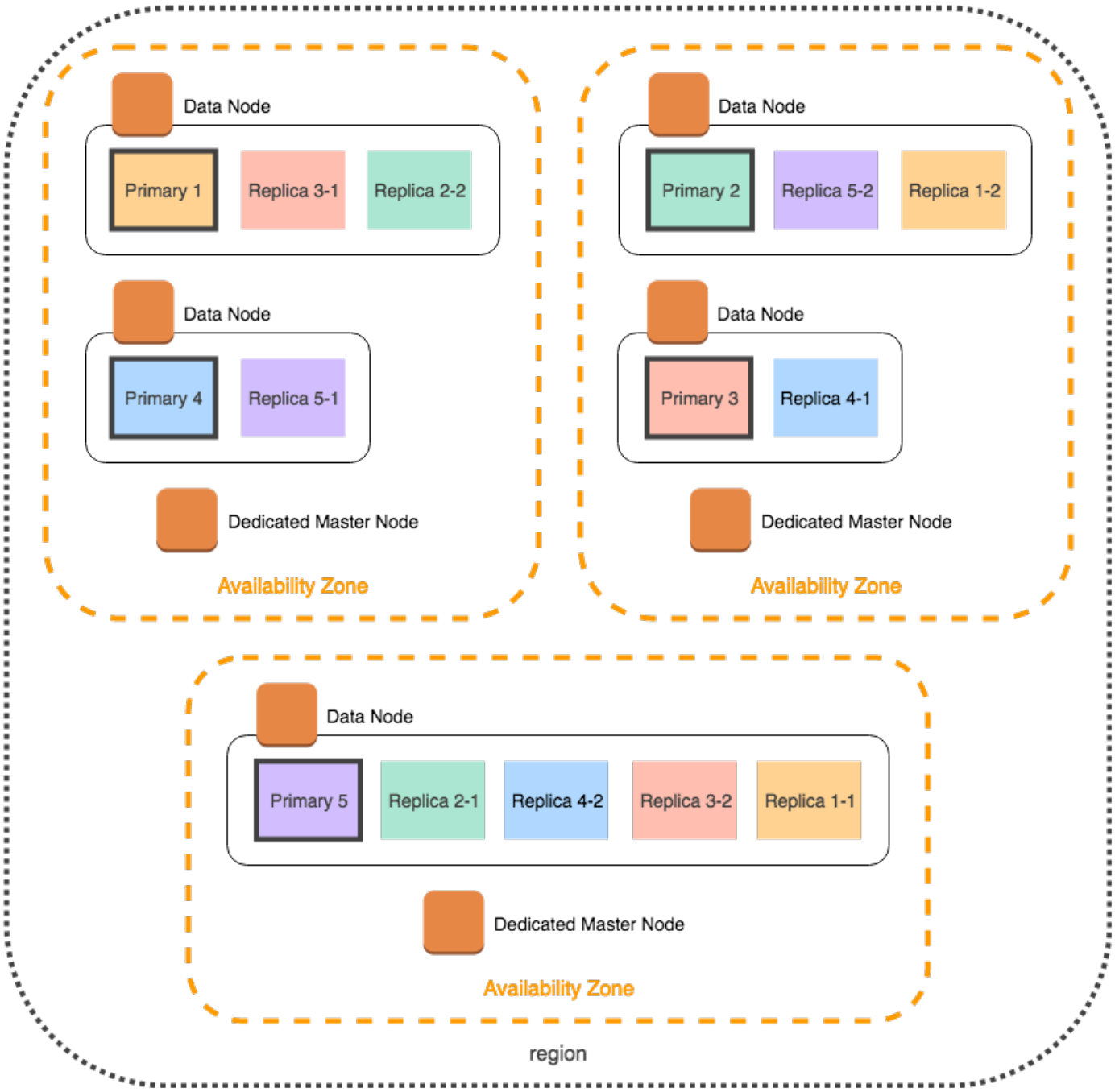
如果您啟用無待命的異地同步備份，您應該為叢集中的每個索引建立至少一個複本。如果沒有複本，OpenSearch Service 就無法將資料複本分發至其他可用區域。還好任何索引的預設組態是複本計數 1。如下圖所示，OpenSearch Service 會盡力將主碎片及其相應的副本碎片分配到不同的區域。



除了依可用區域分配碎片外，OpenSearch Service 也會依節點對其進行分配。然而，特定網域組態會導致不平衡的碎片計數。請考慮以下網域：

- 5 個資料節點
- 5 個主要碎片
- 2 個複本
- 3 個可用區域

在這種情況下，OpenSearch Service 必須超載一個節點，以便跨區域分配主要和複本碎片，如下所示。

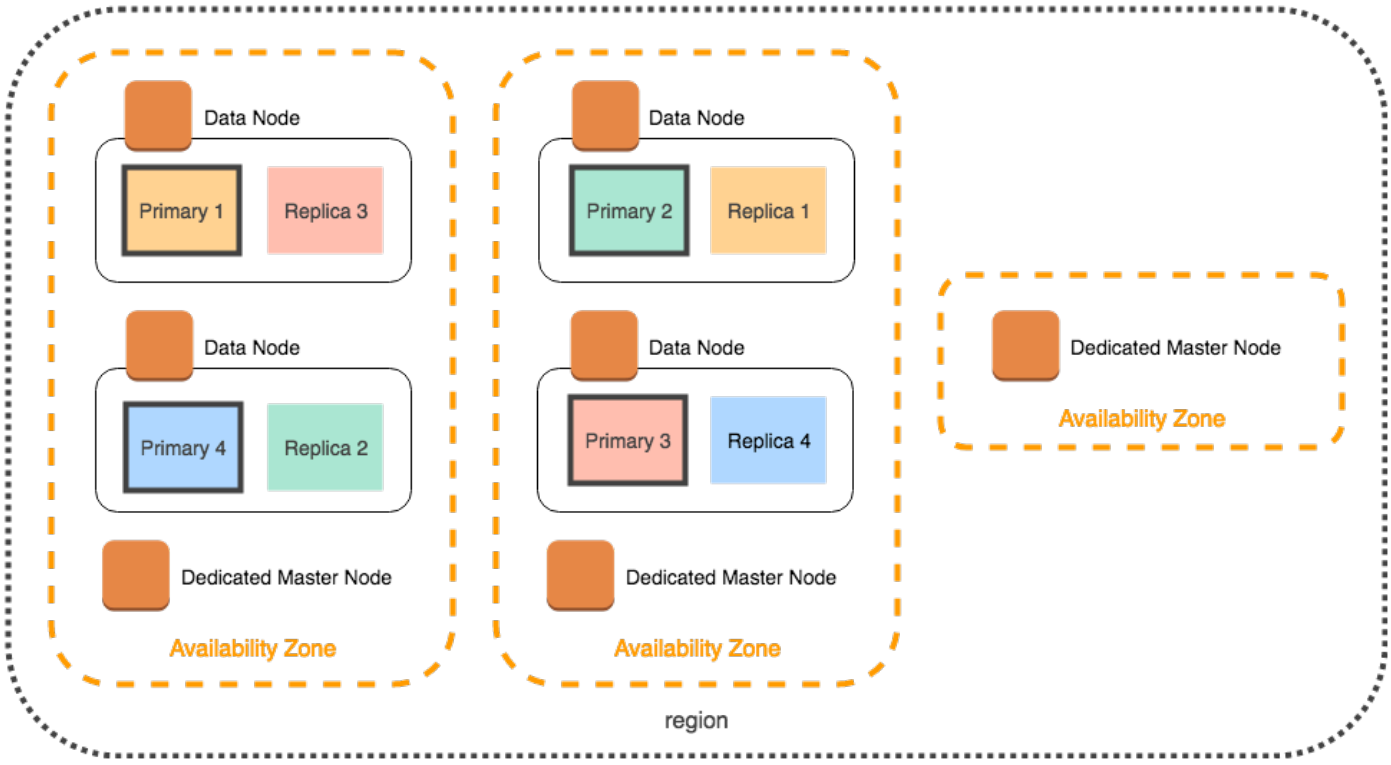


為了避免這些可能導致個別節點壓力和效能受損的情況，我們建議您選擇具有待命的異地同步備份，或在計劃每個索引有兩個或多個複本時，選擇為三個的倍數的執行個體計數。

專用主節點分佈

即使您在設定網域時選擇了兩個可用區域，OpenSearch Service 仍會在三個可用區域自動分配專用主節點。此分發可協助防止區域發生服務中斷時的叢集停機時間。如果建議的三個專用主節點和一個可用

區域在您使用時發生故障，您的叢集仍具有專用主節點的仲裁 (2)，並且可以選擇新的主節點。下圖示範了此組態。



如果您選擇了在三個可用區域中已無法使用的上一代執行個體類型，則適用以下狀況：

- 如果您為該網域選擇了三個可用區域，OpenSearch Service 會發出錯誤。請選擇不同的執行個體類型，然後再試一次。
- 如果您為該網域選擇了兩個可用區域，則 OpenSearch Service 會將專用主節點發佈至兩個區域。

可用區域中斷

可用區域中斷雖然極少發生，但仍有可能出現。下表列出不同多個可用區在中段期間的組態和行為。資料表中的最後一列適用於具有待命的異地同步備份，而所有其他列的組態僅適用於沒有待命的異地同步備份。

區域中的可用區域數量	您選擇的可用區域數量	專用主節點數目	如果一個可用區域中斷時的行為
2 或以上	2	0	停機。您的叢集會失去一半的資料節點，且必須至少替換剩餘可用區域中的其中一個，然後才能選擇主節點。
2	2	3	停機機率為 50/50。OpenSearch Service 會將兩個專用主節點個別分配到兩個可用區域： • 如果擁有一個專用主節點的可用區域遭遇中斷，則剩餘可用區域中的兩個專用主節點可以選擇一個主節點。 • 如果擁有兩個專用主節點的可用區域遭遇中斷，則在剩餘可用區域恢復之前，叢集都無法使用。
3 或以上	2	3	不會停機。OpenSearch Service 會自動在三個可用區域分配專用主節點，因此剩餘的兩個專用主節點可以選擇一個主節點。
3 或以上	3	0	不會停機。概略而言您資料節點的三分之二仍然可以選擇主節點。
3 或以上	3	3	不會停機。剩餘的兩個專用主節點可以選擇主節點。

在所有組態中，無論原因如何，節點故障都可能導致叢集的剩餘資料節點遭遇一段時間的負載增加，而 OpenSearch Service 會自動設定新節點以替換現在失去的節點。

例如，如果三個區域組態中的可用區域中斷，則三分之二的資料節點必須處理與叢集一樣多的請求。當處理這些請求時，剩餘的節點也會在上線時複製碎片到新節點，而這可能近一步影響效能。如果可用性與您的工作負載息息相關，請考慮在叢集中新增資源，以便舒緩此問題。

> **Note**
>
> OpenSearch Service 會以透明的方式管理多可用區域網域，因此您無法手動模擬可用區域中斷。

在 VPC 中啟動您的 Amazon OpenSearch Service 網域

您可以在虛擬私有雲端 (VPC) 中啟動 AWS 資源，例如 Amazon OpenSearch Service 網域。VPC 是您的專用虛擬網路 AWS 帳戶。它在邏輯上與 AWS 雲端中的其他虛擬網路隔離。在 VPC 中放置 OpenSearch Service 網域可在 VPC 中的 OpenSearch Service 和其他服務之間進行安全通訊，而不需要網際網路閘道、NAT 裝置或 VPN 連接。所有流量都會安全地保留在 AWS 雲端中。

> **Note**
>
> 如果您將 OpenSearch Service 網域放置在 VPC 中，您的電腦必須能夠連接到 VPC。此連線通常會採用 VPN、傳輸閘道、受管網路或代理伺服器形式。您無法從 VPC 外部直接存取您的網域。

VPC 與公有網域

以下是 VPC 網域不同於公有網域的一些方面。稍後會更詳細地說明各項差異。

- 因為其邏輯隔離，相較於使用公有端點的網域，位於 VPC 內的網域具有額外的安全層。
- 雖然可以從任何連接網際網路的裝置存取公有網域，但 VPC 網域需要某種形式的 VPN 或代理。
- 相較於公有網域，VPC 網域在 主控台中顯示較少的資訊。特別是，Cluster health (叢集運作狀態) 索引標籤並不包含碎片資訊，並且 Indices (索引) 索引標籤不會出現。
- 網域端點採用不同的形式 (`https://search-domain-name` 與 `https://vpc-domain-name`)。
- 由於安全群組已經強制執行 IP 為基礎的存取政策，您無法將 IP 為基礎的存取政策套用到位於 VPC 內的網域。

限制

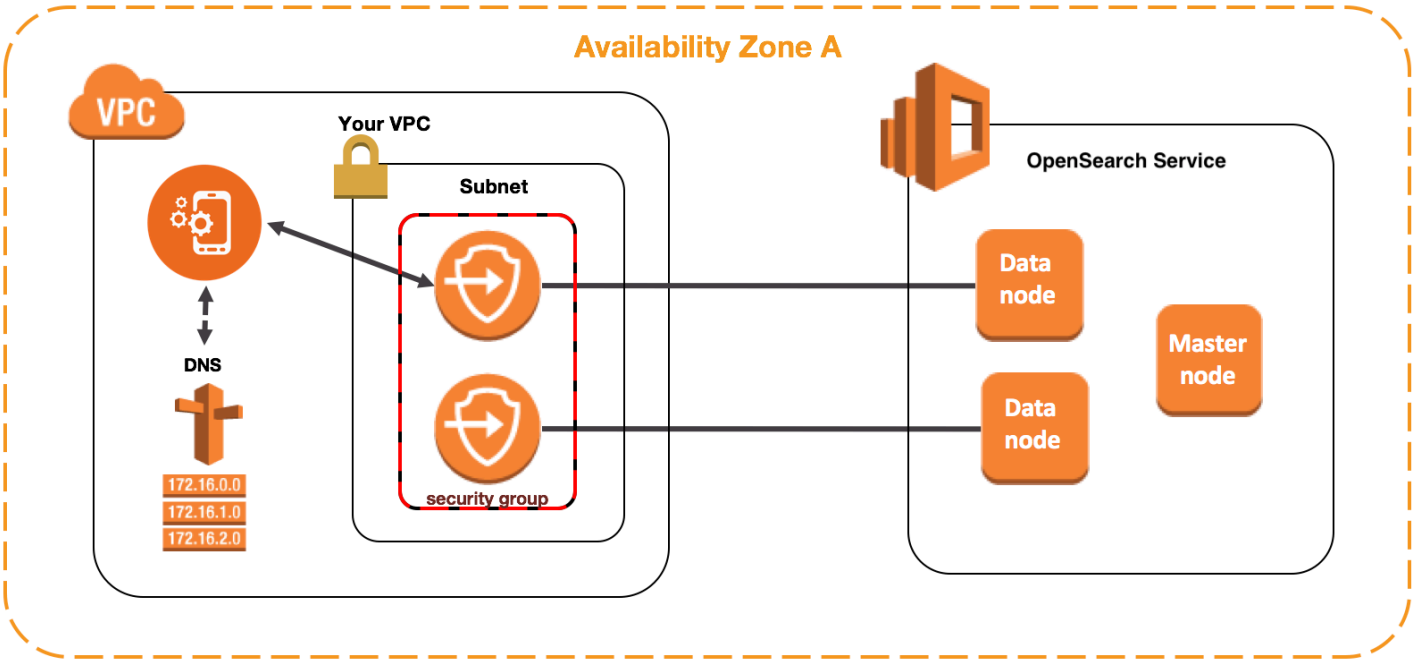
在 VPC 中操作 OpenSearch Service 網域具有下列限制：

- 如果您在 VPC 中啟動新的網域，則無法在稍後進行切換以使用公有端點。反之亦然：如果您建立具有公有端點的網域，您稍後無法將其置於 VPC 之中。反之，您必須建立新網域並遷移您的資料。
- 您可以在 VPC 中啟動您的網域，或者使用公有端點，但是您無法同時進行兩者。您必須在建立網域時選擇其中一個。
- 您無法在使用專用租用的 VPC 中啟動您的網域。您必須使用租用設定為預設的 VPC。
- 您將網域置於 VPC 後，您無法將其移到不同的 VPC，但您可以變更子網路和安全群組設定。
- 若要針對位於 VPC 內的網域存取 OpenSearch Dashboards 的預設安裝，使用者必須能夠存取 VPC。此程序會依網路組態而異，但可能需要連線到 VPN 或受管網路或使用代理伺服器或傳輸閘道。如需進一步了解，請參閱 the section called “關於 VPC 網域上的存取政策”、Amazon VPC 使用者指南和 the section called “控制對 Dashboards 的存取”。

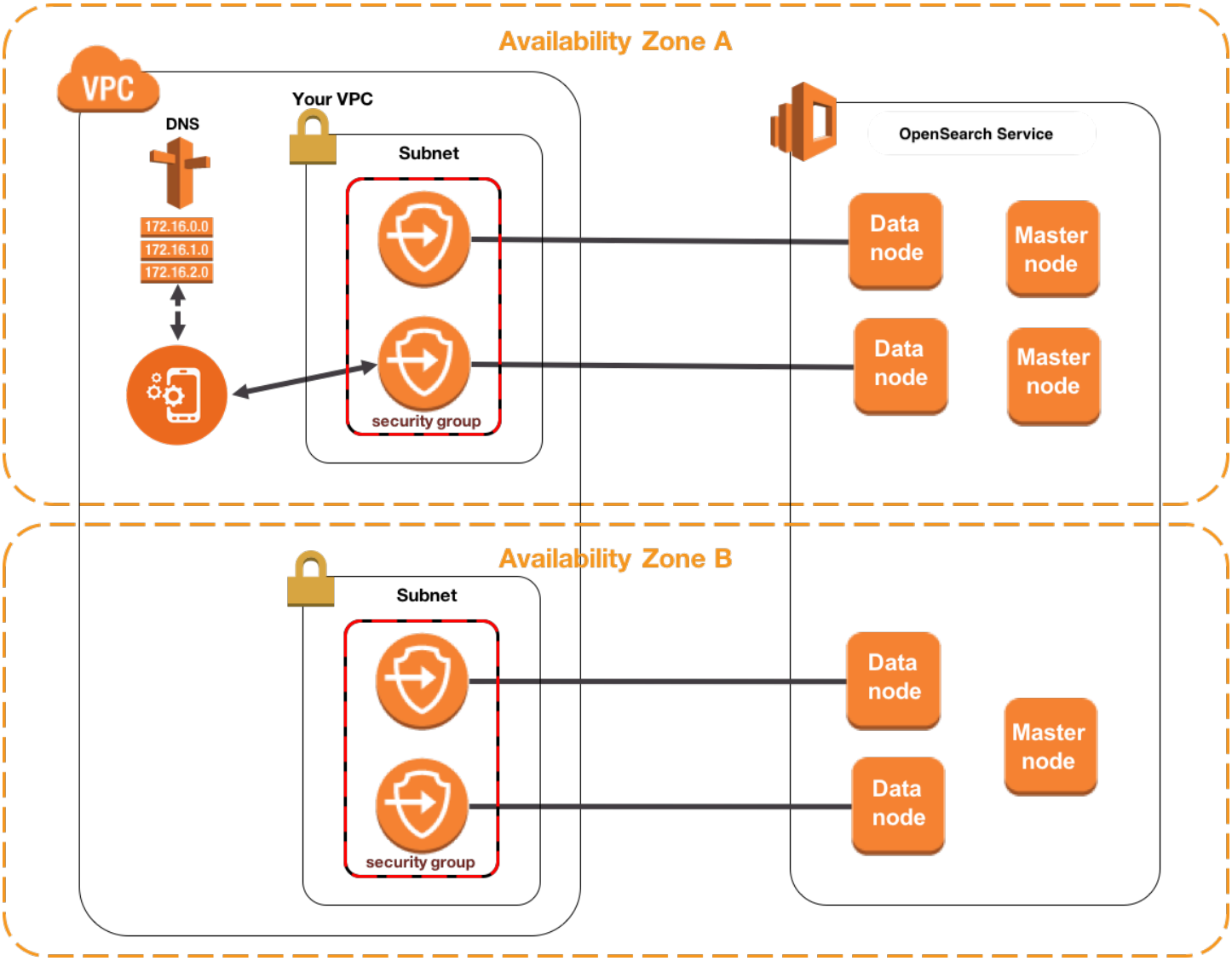
Architecture

為了支援 VPC，OpenSearch Service 會將端點置放到您 VPC 的一個、二個或三個子網路中。如果您為網域啟用了多個可用區，每個子網路都必須位於相同區域中的不同可用區域內。如果您只啟用一個可用區域，OpenSearch Service 會僅將端點置於一個子網路中。

下圖顯示一個可用區域的 VPC 架構：



下圖顯示兩個可用區域的 VPC 架構：



對於每個資料節點，OpenSearch Service 也會將彈性網路介面 (ENI) 放置到 VPC 中。OpenSearch Service 會為每個 ENI 指派一個您的子網路 IPv4 地址範圍中的私有 IP 地址。服務也會為 IP 地址指定公有 DNS 主機名稱 (網域端點)。您必須使用公有 DNS 服務，來解析端點 (DNS 主機名稱) 到資料節點的適當 IP 地址：

- 如果您的 VPC 透過將 `enableDnsSupport` 選項設定為 `true` (預設值)，來使用 Amazon 提供的 DNS 伺服器，則 OpenSearch Service 端點的解析將會取得成功。
- 如果您的 VPC 使用私有 DNS 伺服器且伺服器可以連接公有授權的 DNS 伺服器以解析 DNS 主機名稱，則 OpenSearch Service 端點的解析也會取得成功。

由於 IP 地址可能變更，您應該定期解析網域端點，以便您可以隨時存取正確的資料節點。我們建議您設定 DNS 解析間隔為一分鐘。如果您使用的是用戶端，您也應該確保用戶端中的 DNS 快取已清除。

從公有存取遷移到 VPC 存取

當您建立網域時，可以指定是否應該有公有端點或位於 VPC 中。建立之後，您無法從一個切換到另一個。反之，您必須建立新網域並且手動重新建立索引或遷移您的資料。快照提供方便的方法遷移資料。如需有關拍攝和恢復快照的資訊，請參閱the section called “建立索引快照”。

關於 VPC 網域上的存取政策

將 OpenSearch Service 網域置於 VPC 中可提供固有、強大的安全層。當您建立具有公用存取的網域時，端點的格式如下：

```
https://search-domain-name-identifier.region.es.amazonaws.com
```

如「公有」標籤建議，這個端點可從任何連接網際網路的裝置存取，即使您可以 (且應該) 控制對其的存取。如果您在 Web 瀏覽器中存取端點，您可能會收到 `Not Authorized` 訊息，但請求會到達網域。

當您建立具 VPC 存取的網域時，端點看起來類似公有端點：

```
https://vpc-domain-name-identifier.region.es.amazonaws.com
```

如果您嘗試在 Web 瀏覽器中存取端點，不過您可能會發現請求逾時。若要執行更基本的 `GET` 請求，您的電腦必須能夠連接到 VPC。此連線通常會採用 VPN、傳輸閘道、受管網路或代理伺服器形式。如需有關它可採用的各種形式的詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 範例。關於以開發為中心的範例，請參閱the section called “測試 VPC 網域”。

除了此連線需求外，VPC 還可讓您透過安全群組管理網域的存取。對於許多使用案例，這個安全功能的組合已足夠，而您可能感覺可安心將開放的存取政策套用到網域。

使用開放的存取政策並不表示網際網路上的任何人都可以存取 OpenSearch Service 網域。反之，其表示如果請求達到 OpenSearch Service 網域且經關聯之安全群組允許，則網域會接受請求。唯一的例外情況是，您使用精細存取控制或指定 IAM 角色的存取政策。在這些情況下，如果網域要接受請求，安全群組必須允許它，並且它必須使用有效的憑證進行簽署。

> **Note**
>
> 由於安全群組已經強制執行 IP 為基礎的存取政策，您無法將 IP 為基礎的存取政策套用到位於 VPC 內的 網域。如果您使用公有存取，IP 為基礎的政策仍然可用。

在您開始之前：VPC 存取的先決條件

在您可以啟用 VPC 和新 OpenSearch Service 網域之間的連線之前，必須執行以下操作：

- 建立 VPC

  若要建立 VPC，您可以使用 Amazon VPC 主控台、CLI AWS 或其中一個 AWS SDKs。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的使用 VPC。如果您已有 VPC，則可以略過此步驟。

- 預留 IP 地址

  OpenSearch Service 透過將網路介面置於 VPC 的子網路中來啟用 VPC 到網域的連線。每個網路界面都與 IP 地址關聯。您必須在子網路中保留足夠數量的 IP 地址給網路界面。如需詳細資訊，請參閱在 VPC 子網路中保留 IP 地址。

測試 VPC 網域

VPC 的增強安全性使得連線到您的網域以及執行基本測試成為一個挑戰。如果您已經擁有 OpenSearch Service VPC 網域而寧願不建立 VPN 伺服器，請嘗試下列程序：

1. 對於網域的存取政策，請選擇 Only use fine-grained access control (僅使用精細存取控制)。完成測試後，您隨時可以更新此設定。
2. 在與 OpenSearch Service 網域相同的 VPC、子網路和安全群組中建立 Amazon Linux Amazon EC2 執行個體。

   由於此執行個體是用於進行測試，只需執行極少的工作，因此請選擇較便宜的執行個體類型，如 `t2.micro`。指派公有 IP 地址給執行個體，然後建立新的金鑰對或選擇現有的金鑰對。如果您建立新的金鑰，請將其下載到您的 `~/.ssh` 目錄。

   若要進一步了解如何建立執行個體，請參閱 Amazon EC2 Linux 執行個體入門。
3. 將網際網路閘道新增至您的 VPC。
4. 在 VPC 的路由表中，新增新的路由。對於 Destination (目的地)，指定其中包含您電腦公有 IP 地址的 CIDR 區塊。對於 Target (目標)，指定您剛建立的網際網路閘道。

   例如，您可以指定 `123.123.123.123/32` 以只用於您的電腦，或指定 `123.123.123.0/24` 以用於一個範圍的電腦。
5. 對於安全群組，指定兩個傳入規則：

Type	通訊協定	連接埠範圍	來源
SSH (22)	TCP (6)	22	*## CIDR ##*
HTTPS (443)	TCP (6)	443	*###### ID*

第一個規則可讓您使用 SSH 連接到 EC2 執行個體。第二個規則允許 EC2 執行個體透過 HTTPS 與 OpenSearch Service 網域通訊。

6. 從終端機執行下列命令：

```
ssh -i ~/.ssh/your-key.pem ec2-user@your-ec2-instance-public-ip -N -L
 9200:vpc-domain-name-identifier.region.es.amazonaws.com:443
```

此命令會建立一個 SSH 通道，透過 EC2 執行個體將 https://localhost:9200 的請求轉送到 OpenSearch Service 網域。在命令中指定連接埠 9200 會模擬本機 OpenSearch 安裝，但您可使用您想要的任何連接埠。OpenSearch Service 只接受透過連接埠 80 (HTTP) 或 443 (HTTPS) 的連接。

此命令不會提供任何意見回饋，並且無限期地執行。若要停止命令，請按 `Ctrl + C`。

7. 導覽至 Web 瀏覽器中的 https：//localhost：9200/_dashboards/。您可能需要認可安全例外狀況。

或者，您也可以使用 https://localhost:9200curl、Postman 或您愛用的程式設計語言，傳送請求到。

> **Tip**
>
> 如果因為憑證不相符而遇到 Curl 錯誤，請嘗試 `--insecure` 旗標。

在 VPC 子網路中保留 IP 地址

OpenSearch Service 會透過將網路介面置放在 VPC (或者如果您啟用了多個可用區域，則為 VPC 的多個子網路) 的子網路中，將網域連線到 VPC。每個網路界面都與 IP 地址關聯。在您建立 OpenSearch Service 網域之前，您必須在每個子網路中有足夠數量的 IP 地址，以適應網路介面。

以下是基本公式：OpenSearch Service 在每個子網路預留的 IP 地址數量是資料節點數的三倍，再除以可用區域數量。

範例

- 如果某個網域有 9 個資料節點和 3 個可用區域，則每個子網路的 IP 計數為 9 * 3 / 3 = 9。
- 如果某個網域有 8 個資料節點和 2 個可用區域，則每個子網路的 IP 計數為 8 * 3 / 2 = 12。
- 如果某個網域有 6 個資料節點和 1 個可用區域，則每個子網路的 IP 計數為 6 * 3 / 1 = 18。

當您建立網域時，OpenSearch Service 會保留 IP 地址，將某些 IP 地址用於網域，並為藍/綠部署保留其餘 IP 地址。您可以在 Amazon EC2 主控台的網路介面區段中查看網路介面及其關聯的 IP 地址。Description (描述) 欄顯示網路介面與哪些 OpenSearch Service 網域相關聯。

> **Tip**
>
> 我們建議您為 OpenSearch Service 保留 IP 地址建立專用的子網路。透過使用專用的子網路，可避免與其他應用程式和服務重疊，並確保您可以預留額外的 IP 地址供未來若需要擴展叢集時使用。若要進一步了解，請參閱在您的 VPC 中建立子網路。

您也可以考慮佈建專用協調器節點，以減少 VPC 網域所需的私有 IP 地址保留次數。OpenSearch 會將彈性網路界面 (ENI) 連接到您的專用協調器節點，而不是您的資料節點。專用協調器節點通常代表總資料節點的大約 10%。因此，會為 VPC 網域保留較少數量的私有 IP 地址。

VPC 存取適用的服務連結角色

服務連結角色是一個唯一的 IAM 角色類型，它將許可委派給服務，以便服務可代表您建立和管理資源。OpenSearch Service 需要服務連結角色來存取您的 VPC、建立網域端點以及將網路介面置於您的 VPC 子網路中。

當您使用 OpenSearch Service 主控台在 VPC 中建立網域時，OpenSearch Service 會自動建立角色。若要讓此自動建立成功，您必須有 `iam:CreateServiceLinkedRole` 動作的許可。如需進一步了解，請參閱 IAM 使用者指南中的服務連結角色許可。

在 OpenSearch Service 建立角色後，您可以使用 IAM 主控台檢視它 (`AWSServiceRoleForAmazonOpenSearchService`)。

如需此角色許可以及如何刪除它的完整資訊，請參閱the section called “使用服務連結角色”。

在 Amazon OpenSearch Service 中建立索引快照

Amazon OpenSearch Service 中的快照是叢集的索引和狀態的備份。狀態包含叢集設定、節點資訊、索引設定和碎片分配。

OpenSearch Service 快照的形式如下：

- 自動快照僅適用叢集復原。如果發生紅色叢集狀態或資料遺失，您可以使用它們來還原您的網域。如需詳細資訊，請參閱下面的還原快照。OpenSearch Service 會將自動快照存放在預先設定的 Simple Storage Service (Amazon S3) 儲存貯體，不另外收取費用。
- 手動快照適用於叢集復原，或者將資料從一個叢集移至另一個叢集。您必須初始化手動快照。這些快照會存放在您自己的 Simple Storage Service (Amazon S3) 儲存貯體中，而且需支付標準 S3 費用。如果您擁有自我管理 OpenSearch 叢集的快照，您可以使用該快照遷移至 OpenSearch Service 網域。如需詳細資訊，請參閱遷移至 Amazon OpenSearch Service。

所有 OpenSearch Service 網域都會建立自動快照，但頻率在下列情況下不同：

- 對於執行 OpenSearch 或 Elasticsearch 5.3 版和更新版本的網域，OpenSearch Service 會每隔 1 小時自動拍攝快照，且最多可保留其中 336 個快照 (連續保留 14 天)。由於每小時快照的增量改進性質，所以中斷較少。在發生網域問題時，它們還提供較新的復原點。
- 對於執行 Elasticsearch 5.1 版和舊版的網域，OpenSearch Service 會每隔 1 天自動拍攝快照 (在您指定期間)，最多可保留其中 14 個快照，且不會保留任何快照資料超過 30 天。

如果您的叢集進入紅色狀態，所有自動快照都會失敗，而叢集狀態會持續存在。如果您未在兩週內修正該問題，則可能會永久遺失叢集中的資料。如需疑難排解步驟，請參閱the section called “紅色叢集狀態”。

先決條件

若要手動建立快照，您必須使用 IAM 和 Simple Storage Service (Amazon S3)。嘗試拍攝快照之前，請先確認符合以下先決條件：

先決條件	Description
S3 bucket (S3 儲存貯體)	建立 S3 儲存貯體以儲存 OpenSearch Service 網域的手動快照。如需說明，請參閱《Amazon Simple Storage Service 使用者指南》中的建立一般用途儲存貯體。

先決條件	Description
	請記住儲存貯體的名稱以在下列位置使用它： • 連接至您 IAM 角色的 IAM 政策的 `Resource` 陳述式 • 用於註冊快照儲存庫的 Python 用戶端 (如果您使用此方法) **Important** 請勿將 Amazon Glacier 生命週期規則套用至此儲存貯體。手動快照不支援 Amazon Glacier 儲存類別。

先決條件	Description
IAM 角色	建立 IAM 角色以將許可委派給 OpenSearch Service。如需說明，請參閱 IAM 使用者指南中的建立 IAM 角色 (主控台)。本章其餘各節稱此角色為 TheSnapshotRole 。 連接 IAM 政策 將下列政策連接至 TheSnapshotRole 以允許存取 S3 儲存貯體： `{ "Version":"2012-10-17", "Statement": [{ "Action": [ "s3:ListBucket" ], "Effect": "Allow", "Resource": [ "arn:aws:s3::: amzn-s3-demo-bucket " ] }, { "Action": [ "s3:GetObject", "s3:PutObject", "s3:DeleteObject" ], "Effect": "Allow", "Resource": [ "arn:aws:s3::: amzn-s3-demo-bucket /*" ] } ] }` 如需將政策連接至角色的指示，請參閱《IAM 使用者指南》中的新增 IAM 身分許可 (主控台)。 編輯信任關係

先決條件	Description
	編輯 TheSnapshotRole 的信任關係，在 Principal 陳述式中指定 OpenSearch Service，如下例所示： `{ "Version":"2012-10-17", "Statement": [{ "Sid": "", "Effect": "Allow", "Principal": { "Service": "es.amazonaws.com" }, "Action": "sts:AssumeRole" }] }` 如需編輯信任關係的指示，請參閱《IAM 使用者指南》中的更新角色信任政策。

先決條件	Description

許可	為了註冊快照儲存庫，您需要能夠將 TheSnapshotRole 傳遞至 OpenSearch Service。您也需要存取 es:ESHttpPut 動作。若要授予這兩個許可，請將下列政策連接至其憑證用於簽署請求的 IAM 角色：
	
	`{ "Version":"2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "iam:PassRole", "Resource": "arn:aws:iam:: 123456789012 :role/TheSnapshotRole " }, { "Effect": "Allow", "Action": "es:ESHttpPut", "Resource": "arn:aws:es: us-east-1 :123456789012 :domain/domain-name /*" } ] }`
	
	如果您的使用者或角色沒有傳遞 的iam:PassRole 許可TheSnapshotRole ，當您在下一個步驟嘗試註冊儲存庫時，可能會遇到下列常見錯誤：
	
	`$ python register-repo.py {"Message":"User: arn:aws:iam:: 123456789012 :user/MyUserAccount is not authorized to perform: iam:PassRole on resource: arn:aws:iam:: 123456789012 :role/TheSnapshotRole "}`

註冊手動快照儲存庫

您需要先使用 OpenSearch Service 來註冊快照儲存庫，才能手動拍攝索引快照。此一次性操作要求您使用允許存取 的登入資料來簽署 AWS 請求TheSnapshotRole，如中所述the section called “先決條件”。

步驟 1：映射 OpenSearch Dashboards 中的快照角色 (如果使用精細存取控制)

精細存取控制會在註冊儲存庫時進行額外的步驟。即使您將 HTTP 基本身分驗證用於所有其他目的，您也需要將 `manage_snapshots` 角色映射至擁有 `iam:PassRole` 許可能夠傳遞 `TheSnapshotRole` 的 IAM 角色。

1. 導覽至 OpenSearch Service 網域的 OpenSearch Dashboards 外掛程式。您可以在 OpenSearch Service 主控台的網域儀表板上找到 Dashboards 端點。
2. 從主選單中選擇 Security (安全性)、Roles (角色)，然後選取 manage_snapshots 角色。
3. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
4. 新增擁有許可傳遞 `TheSnapshotRole` 之角色的 ARN。將角色 ARN 放在 Backend roles (後端角色) 下。

   ```
   arn:aws:iam::123456789123:role/role-name
   ```

5. 選擇 Map (映射)，並確認使用者或角色顯示在 Mapped users (已映射的使用者) 中。

步驟 2：註冊儲存庫

下列快照索引標籤示範如何註冊快照目錄。如需在遷移至新網域後加密手動快照和註冊快照的特定選項，請參閱相關索引標籤。

Snapshots

若要註冊快照儲存庫，請將 PUT 請求傳送到 OpenSearch Service 網域端點。您可以使用 curl、範例 Python 用戶端、Postman 或其他方法傳送已簽署的請求來註冊快照儲存庫。請注意，您無法在 OpenSearch Dashboards 主控台中使用 PUT 請求來註冊儲存庫。

請求採用下列格式：

```
PUT domain-endpoint/_snapshot/my-snapshot-repo-name
{
  "type": "s3",
  "settings": {
    "bucket": "amzn-s3-demo-bucket",
    "base_path": "my/snapshot/directory",
    "region": "region",
    "role_arn": "arn:aws:iam::123456789012:role/TheSnapshotRole"
  }
}
```

> **Note**
>
> 儲存庫名稱不能以 "cs-" 開頭。此外，您不應該從多個網域寫入同一個儲存庫。只有一個網域應具有儲存庫的寫入存取權。

如果您的網域存放在 Virtual Private Cloud (VPC) 中，您的電腦必須連接到 VPC，才能讓請求成功註冊快照儲存庫。存取 VPC 因網路組態而異，但可能牽涉到連線 VPN 或公司網路。若要確認您可以連上 OpenSearch Service 網域，請在 Web 瀏覽器中導覽至 `https://your-vpc-domain.region.es.amazonaws.com`，並確認您接收預設的 JSON 回應。

當您的 Amazon S3 儲存貯體位於 OpenSearch 網域 AWS 區域 以外的 時，請將 參數新增至`"endpoint": "s3.amazonaws.com"`請求。

Encrypted snapshots

您目前無法使用 AWS Key Management Service (KMS) 金鑰來加密手動快照，但您可以使用伺服器端加密 (SSE) 來保護快照。

若要使用做為快照儲存庫之儲存貯體的 S3-managed金鑰開啟 SSE，請將`"server_side_encryption": true`新增至 PUT 請求的 `"settings"`區塊。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的搭配 Amazon S3 受管金鑰 (SSE-S3) 使用伺服器端加密。

或者，您可以在用作快照儲存庫的 S3 儲存貯體上使用 AWS KMS 金鑰進行伺服器端加密。如果您使用此方法，請務必提供用於加密 S3 儲存貯體之 AWS KMS 金鑰的TheSnapshotRole許可。如需詳細資訊，請參閱 AWS KMS中的金鑰政策。

Domain migration

註冊快照儲存庫是一次性操作。但要從一個網域遷移到另一個網域，必須在舊網域和新網域上註冊相同的快照儲存庫。儲存庫名稱是任意的。

遷移至新網域或註冊具有多個網域的相同儲存庫時，請考慮下列準則：

- 在新網域上註冊儲存庫時，請將 `"readonly": true` 新增至 PUT 請求的 `"settings"` 區塊。此設定可防止您意外地從舊網域中覆寫資料。只有一個網域應具有儲存庫的寫入存取權。
- 如果您要將資料遷移至不同 中的網域 （ 例如，從位於 us-east-2 的舊網域和儲存貯體遷移至 us-west-2 中的新網域） AWS 區域，請在 PUT 陳述式`"endpoint": "s3.amazonaws.com"`中將 取代`"region": "region"`為 ，然後重試請求。

使用 Python 用戶端範例

Python 用戶端比簡單的 HTTP 請求更容易自動化，並且具有更好的可重複使用性。如果您選擇使用此方法來註冊快照儲存庫，請將下列範例 Python 程式碼儲存為 Python 檔案，例如 `register-repo.py`。用戶端需要 適用於 Python (Boto3) 的 AWS SDK、請求和 requests-aws4auth 套件。用戶端包含其他快照操作的註解範例。

更新範本程式碼中的下列變數：`host`、`region`、`path` 以及 `payload`。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = '' # domain endpoint
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Register repository

path = '/_snapshot/my-snapshot-repo-name' # the OpenSearch API endpoint
url = host + path

payload = {
  "type": "s3",
  "settings": {
    "bucket": "amzn-s3-demo-bucket",
    "base_path": "my/snapshot/directory",
    "region": "us-west-1",
    "role_arn": "arn:aws:iam::123456789012:role/snapshot-role"
  }
}

headers = {"Content-Type": "application/json"}

r = requests.put(url, auth=awsauth, json=payload, headers=headers)

print(r.status_code)
print(r.text)

# # Take snapshot
```

```
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot'
# url = host + path
#
# r = requests.put(url, auth=awsauth)
#
# print(r.text)
#
# # Delete index
#
# path = 'my-index'
# url = host + path
#
# r = requests.delete(url, auth=awsauth)
#
# print(r.text)
#
# # Restore snapshot (all indexes except Dashboards and fine-grained access control)
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
# url = host + path
#
# payload = {
#   "indices": "-.kibana*,-.opendistro_security,-.opendistro-*",
#   "include_global_state": False
# }
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
#
# # Restore snapshot (one index)
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
# url = host + path
#
# payload = {"indices": "my-index"}
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
```

```
# print(r.text)
```

手動拍攝快照

快照不是即時的。它們需要一些時間才能完成，並不確切代表叢集在各個時間點的狀態。當快照仍在進行中，您還是可以對文件編製索引並對叢集發出其他請求，但新文件以及對現有文件的更新一般不包括在快照中。快照包含當 OpenSearch 啟動快照時既已存在的主要碎片。視快照執行緒集區的大小而定，快照中可能附有相距些許時間差的不同碎片。如需快照最佳實務，請參閱 the section called “改善快照效能”。

> **Warning**
>
> 手動快照不包含儲存在 UltraWarm 或冷儲存層中的資料。如果您的網域使用 UltraWarm 或冷儲存，如果您需要保留該資料，請在手動快照之前將這些索引遷移至熱儲存。

快照儲存與效能

OpenSearch 快照是增量式的，這表示它們只會存放自從上次成功快照之後有所變更的資料。這種增量性質表示頻率高和頻率低的快照之間的磁碟用量差異通常很小。換言之，每小時拍攝快照長達一週 (總共 168 個快照)，可能不會比在週末拍攝單一快照使用更多的磁碟空間。此外，您拍攝快照的頻率越高，完成快照所需的時間越少。例如，每日快照可能需要 20-30 分鐘才能完成，而每小時快照可能在幾分鐘內即可完成。有些 OpenSearch 使用者建立快照的頻率高達半小時一次。

建立快照

建立快照時可以指定以下資訊：

- 快照儲存庫的名稱
- 快照的名稱

為了方便和簡潔，本章中的範例使用 Curl 這個常見的 HTTP 用戶端。若要將使用者名稱和密碼傳遞至 curl 請求，請參閱入門教學課程。

如果您的存取政策指定使用者或角色，您必須簽署快照請求。對於 curl，您可以使用 `--aws-sigv4`選項搭配 7.75.0 版或更新版本。您也可以使用範例 Python 用戶端中的註解範例，對 curl 命令使用的相同端點提出已簽署的 HTTP 請求。

若要建立手動快照，請執行下列步驟：

1. 如果快照正在進行中，您就無法取得快照。若要檢查，請執行下列命令：

```
curl -XGET 'domain-endpoint/_snapshot/_status'
```

2. 執行以下命令來手動拍攝快照：

```
curl -XPUT 'domain-endpoint/_snapshot/repository-name/snapshot-name'
```

若要包含或排除某些索引並指定其他設定，請新增請求主體。如需了解請求結構，請參閱 OpenSearch 文件中的 Take snapshots (拍攝快照)。

> **Note**
>
> 拍攝快照所需的時間隨著 OpenSearch Service 網域的大小而增加。長時間執行的快照操作有時會發生以下錯誤：`504 GATEWAY_TIMEOUT`。一般而言，您可以忽略這些錯誤，並等待操作成功完成。執行下列命令來驗證您網域的所有快照狀態：
>
> ```
> curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
> ```

還原快照

還原快照之前，請確定目的地網域未使用多可用區域與待命。啟用待命會導致還原操作失敗。

> **Warning**
>
> 如果您使用索引別名，您應該停止對別名的寫入請求，或在刪除其索引之前將別名切換到另一個索引。停止寫入請求有助於避免以下情況：
>
> 1. 刪除索引也會刪除它的別名。
> 2. 錯誤的寫入請求至現已刪除的別名，會使用和別名相同的名稱建立新索引。
> 3. 因與新索引有命名衝突之故，您不可再使用別名。當您從快照還原，如果您交換別名到另一個索引，此時請指定`"include_aliases": false`。

還原快照

1. 找到您要還原的快照。請確定此索引的所有設定，例如自訂分析器套件或配置需求設定，都與網域相容。若要查看所有快照儲存庫，請執行下列命令：

   ```
   curl -XGET 'domain-endpoint/_snapshot?pretty'
   ```

   在識別儲存庫後，請執行下列命令以查看所有快照：

   ```
   curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
   ```

   > **Note**
   >
   > 大多數自動快照存放在 cs-automated 儲存庫中。如果您的網域加密靜態資料，會將它們存放在 cs-automated-enc 儲存庫中。如果未看到您尋找的手動快照儲存庫，請確定您已將它註冊到網域。

2. (選用) 如果叢集上的索引與快照中的索引存在命名衝突，可以刪除或重新命名 OpenSearch Service 網域中的一個或多個索引。您無法將索引的快照還原到已經包含具相同名稱之索引的 OpenSearch 叢集。

   如果您有索引命名衝突，則具有下列選項：

   - 刪除現有 OpenSearch Service 網域上的索引，然後還原快照。
   - 在您從快照還原索引時將其重新命名，並為其重新編製索引。若要了解如何重新命名索引，請參閱 OpenSearch 文件中的此範例請求。
   - 將快照還原到不同 OpenSearch Service 網域 (只能還原手動快照)。

   下列命令會刪除網域中的所有現有索引：

   ```
   curl -XDELETE 'domain-endpoint/_all'
   ```

   但是，如果您不打算還原所有索引，則只可以刪除一個索引：

   ```
   curl -XDELETE 'domain-endpoint/index-name'
   ```

3. 若要還原快照，請執行以下命令：

   ```
   curl -XPOST 'domain-endpoint/_snapshot/repository-name/snapshot-name/_restore'
   ```

由於 OpenSearch Dashboards 的特殊許可和精細存取控制索引，嘗試還原所有索引可能會失敗，尤其是當您嘗試從自動快照還原時。以下範例只從 my-index 快照儲存庫中的 2020-snapshot 還原一個索引 cs-automated：

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "my-index"}' \
-H 'Content-Type: application/json'
```

或者，您可能想要還原所有索引，Dashboards 和精細存取控制索引除外：

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "-.kibana*,-.opendistro*"}' \
-H 'Content-Type: application/json'
```

> **Note**
>
> 根據您的 OpenSearch 版本，可能還需要排除其他系統索引，例如 -.opensearch-observability*和 -.plugins-ml-config*。如果這些索引已存在於目標網域上，則還原操作可能會失敗。若要排除它們，請將它們新增至indices排除清單。例如："indices": "-.kibana*,-.opendistro*,-.opensearch-observability*,-.plugins-ml-config*"。

您可以使用 rename_pattern和 rename_replacement 參數還原快照，而不刪除其資料。如需這些參數的詳細資訊，請參閱 OpenSearch 文件中的還原快照 API 請求欄位和範例請求。

> **Note**
>
> 如果不是所有涉及的索引都有主要碎片，快照可能一個 state 的 PARTIAL。此值表示至少一個碎片的資料未成功儲存。您仍然可以從部分快照還原，但您可能需要使用舊版的快照來還原任何遺失的索引。

刪除手動快照

若要刪除手動快照，請執行以下命令：

```
DELETE _snapshot/repository-name/snapshot-name
```

使用快照管理自動化快照

您可以在 OpenSearch Dashboards 中設定快照管理 (SM) 政策，以自動化定期建立和刪除快照。SM 可以拍攝一組索引的快照，而索引狀態管理只能為每個索引拍攝一個快照。若要在 OpenSearch Service 中使用 SM，您需要註冊自己的 Amazon S3 儲存庫。如需註冊儲存庫的指示，請參閱註冊手動快照儲存庫。

> **Note**
>
> 從 OpenSearch 2.5 版開始支援快照管理。執行舊版的網域不支援此功能。

在 SM 之前，OpenSearch Service 提供了免費的自動快照功能，預設仍開啟。此功能會將快照傳送至服務維護的`cs-*`儲存庫。若要停用此功能，請聯絡 支援。

如需 SM 功能的詳細資訊，請參閱 OpenSearch 文件中的快照管理。

SM 目前不支援在多種索引類型上建立快照。例如，如果您嘗試使用 在多個索引上建立快照，*而某些索引位於暖層，則快照建立將會失敗。如果您需要快照包含多個索引類型，請使用 ISM 快照動作，直到 SM 支援此選項為止。

設定 許可

如果您是從先前的 OpenSearch Service 網域版本升級至 2.5，則可能不會在網域上定義快照管理安全許可。非管理員使用者必須對應至此角色，才能使用精細存取控制在網域上使用快照管理。若要手動建立快照管理角色，請執行下列步驟：

1. 在 OpenSearch Dashboards 中，移至 Security (安全性)，然後選擇 Permissions (許可)。
2. 選擇 Create action group (建立動作群組) 並設定下列群組：

Group name (群組名稱)	許可
`snapshot_managemen`	• `cluster:admin/opensearch/snapshot_management/*` • `cluster:admin/opensearch/notifications/feature/publish`

Group name (群組名稱)	許可
t_full_access	• cluster:admin/repository/* • cluster:admin/snapshot/*
snapshot_management_read_access	• cluster:admin/opensearch/snapshot_management/policy/get • cluster:admin/opensearch/snapshot_management/policy/search • cluster:admin/opensearch/snapshot_management/policy/explain • cluster:admin/repository/get • cluster:admin/snapshot/get

3. 選擇 Roles (角色)，然後選擇 Create role (建立角色)。
4. 命名角色 snapshot_management_role。
5. 針對叢集許可，選取 snapshot_management_full_access或 snapshot_management_read_access。
6. 選擇建立。
7. 建立角色之後，請將其映射至將管理快照的任何使用者或後端角色。

考量事項

當您設定快照管理時，請考慮下列事項：

- 每個儲存庫允許一個政策。
- 一個政策最多允許 400 個快照。
- 如果您的網域狀態為紅色、處於高 JVM 壓力 (85% 或更高） 或具有停滯快照函數，則不會執行此功能。當叢集的整體索引和搜尋效能受到影響時，SM 也可能受到影響。
- 快照操作只會在先前的操作完成後啟動，因此一個政策不會啟動並行快照操作。
- 具有相同排程的多個政策可能會導致資源尖峰。如果政策的快照索引重疊，碎片層級快照操作只能循序執行，這可能會導致層疊效能問題。如果政策共用儲存庫，該儲存庫的寫入操作將會激增。
- 建議您將快照操作自動化排程為每小時不超過一次，除非您有特殊的使用案例。

使用索引狀態管理自動化快照

您可以使用索引狀態管理 (ISM) 快照操作，根據索引的存留期、大小或文件數量的變更自動觸發索引快照。當您需要每個索引一個快照時，ISM 是最佳的。如果您需要一組索引的快照，請參閱 使用快照管理自動化快照。

若要在 OpenSearch Service 中使用 SM，您需要註冊自己的 Amazon S3 儲存庫。如需使用 `snapshot` 操作的範例 ISM 政策，請參閱範例政策。

使用 Curator 進行快照

如果 ISM 不適用於索引和快照管理，您可以改用 Curator。其提供進階篩選功能，可協助簡化複雜叢集的管理任務。使用 pip 安裝 Curator。

```
pip install elasticsearch-curator
```

您可以使用 Curator 做為命令列界面 (CLI) 或 Python API。如果您使用 Python API，則必須使用版本 7.13.4 或更早版本的舊版 elasticsearch-py 用戶端。不支援 opensearch-py 用戶端。

如果您使用 CLI，在命令列匯出您的登入資料，並且設定 `curator.yml`，如下所示：

```
client:
  hosts: search-my-domain.us-west-1.es.amazonaws.com
  port: 443
  use_ssl: True
  aws_region: us-west-1
  aws_sign_request: True
  ssl_no_validate: False
  timeout: 60

logging:
  loglevel: INFO
```

升級 Amazon OpenSearch Service 網域

> **Note**
>
> OpenSearch 和 Elasticsearch 版本升級與服務軟體更新有所不同。如需為您的 OpenSearch Service 網域更新服務軟體的相關資訊，請參閱the section called “服務軟體更新”。

Amazon OpenSearch Service 為執行 OpenSearch 1.0 或更新版本或者 Elasticsearch 5.1 或更新版本的網域提供就地升級。如果您使用 Amazon Data Firehose 或 Amazon CloudWatch Logs 等服務將資料串流至 OpenSearch Service，請在遷移之前檢查這些服務是否支援較新版本的 OpenSearch。

支援的升級路徑

OpenSearch Service 目前支援以下升級路徑：

原始版本	目標版本
OpenSearch 1.3、2.x 或 3.x	OpenSearch 3.x 如果您正在執行 OpenSearch 1.3 或 2.x，您必須先升級至 OpenSearch 2.19，才能升級至 OpenSearch 3.x。 從 OpenSearch 2.19 升級到 OpenSearch 3.x 時，您可能需要解決下列不相容的索引設定。這些已在 2.x 中提早棄用，若未移除，將導致升級檢查驗證失敗。 • index.knn.algo_param.ef_construction • index.knn.algo_param.m • index.knn.space_type • index.store.hybrid.mmap.extensions 如需已棄用索引設定和修復它們步驟的詳細資訊，請參閱索引設定。如需 OpenSearch 3.x 重大變更的完整清單，請參閱 3.0.0。 如果您的網域包含 OpenSearch 1.3、Elasticsearch 7.10 或舊版中建立的索引，您必須先重新建立索引，才能升級至 OpenSearch 3.x。OpenSearch 3.x 不支援這些較舊的索引，無論是熱儲存、UltraWarm 儲存還是冷儲存。若要重新索引不相容的 UltraWarm 或冷索引： 1. 將索引遷移至熱儲存。 2. 重新索引資料。 3. 將索引遷移回暖儲存或冷儲存。 或者，如果您不再需要索引，請刪除索引。

原始版本	目標版本
OpenSearch 1.3 或 2.x	OpenSearch 2.x 如果網域滿足下列條件，則 OpenSearch 2.17 預設會使用自動模式啟用並行區段搜尋： • 不會明確設定先前的並行搜尋設定。 • 所有資料執行個體（熱和暖）的執行個體類型為 2.xl 或更高。 • 資料執行個體（熱和暖）上超過 1 週的平均 p90 cpu 使用率低於 45%。 如需此處並行區段搜尋設定的詳細資訊，請參閱並行區段搜尋。 2.3 版具有以下突破性變更： • 在 2.0 版中，`type` 參數已從所有 OpenSearch API 端點移除。如需詳細資訊，請參閱 Breaking changes (突破性變更)。 • 如果您的網域包含原先在 Elasticsearch 6.8 中建立的任何索引 (熱索引、UltraWarm 索引或冷索引)，這些索引並不相容於 OpenSearch 2.3。 升級至 2.3 版之前，您必須對不相容的索引重新編製索引。對於不相容的 UltraWarm 索引或冷索引，請將這些索引遷移至熱儲存、對資料重新編製索引，然後將它們遷移回暖儲存或冷儲存。或者，您也可以在不再需要時刪除索引。 如果您意外將網域升級至 2.3 版，而未先執行這些步驟，則無法從目前的儲存層遷移出不相容的索引。您的唯一選項是刪除它們。
OpenSearch 1.x	OpenSearch 1.x
Elasticsearch 7.x	Elasticsearch 7.x 或 OpenSearch 1.x **Important** OpenSearch 1.x 推出了許多突破性變更。如需詳細資訊，請參閱Amazon OpenSearch Service 重新命名。

原始版本	目標版本
Elasticsearch 6.8	Elasticsearch 7.x 或 OpenSearch 1.x **Important** Elasticsearch 7.0 和 OpenSearch 1.0 包括許多突破性的變更。在開始就地升級之前，我們建議應先針對 6.x 網域取得手動快照、在測試 7.x 或 OpenSearch 1.x 網域上還原，並使用該測試網域來識別潛在的升級問題。如需 OpenSearch 1.0 中的突破性變更，請參閱Amazon OpenSearch Service 重新命名。 如同 Elasticsearch 6.x，索引只能包含一個映射類型，但該類型現在必須名為 `_doc`。因此，特定 API 不再要求在請求內文中有映射類型 (例如 `_bulk` API)。 對於新的索引，自我託管的 Elasticsearch 7.x 和 OpenSearch 1.x 的預設碎片計數為 1。Elasticsearch 7.x 和更新版本中的 OpenSearch Service 網域保留之前的預設值 5。
Elasticsearch 6.x	Elasticsearch 6.x
Elasticsearch 5.6	Elasticsearch 6.x **Important** 在 6.x 版建立的索引不再支援多個映射類型。在 5.x 版中建立的索引，在恢復為 6.x 叢集時仍支援多個映射類型。請檢查您的用戶端程式碼是否每個索引只建立單一映射類型。 若要在從 Elasticsearch 5.6 升級到 6.x 期間將停機時間降到最低，OpenSearch Service 會將 `.kibana` 索引重新編製為 `.kibana-6` 、刪除 `.kibana`、建立名為 `.kibana` 的別名以及將新的索引映射到新的別名。
Elasticsearch 5.x	Elasticsearch 5.x

升級程序包含三個步驟：

1. 預先升級檢查 – OpenSearch Service 會檢查是否有阻止升級的問題，且在這些檢查成功前都不會繼續下一個步驟。
2. 快照 - OpenSearch Service 會拍攝 OpenSearch 或 Elasticsearch 叢集的快照，且在快照成功拍攝前不會繼續至下一個步驟。如果升級失敗，OpenSearch Service 會使用此快照來將叢集恢復至其原始狀態。如需詳細資訊，請參閱 the section called “升級後無法降級”。
3. 升級 – OpenSearch Service 開始升級，這會需要 15 分鐘到數小時才能完成。OpenSearch Dashboards 在升級的部分或所有期間可能會無法使用。

升級網域（主控台）

升級程序無法復原，且無法暫停或取消。升級時，您無法對網域做出組態變更。開始升級前，重複確認您是否要繼續。您可以使用這些相同的步驟，來執行預先升級的檢查，而不用實際開始升級。

如果叢集有專用主節點，OpenSearch 無須停機即可完成升級。否則叢集在選擇主節點時，可能會在升級後數秒沒有回應。

若要將網域升級到 OpenSearch 或 Elasticsearch 的較新版本

1. 建立您網域的手動快照。如果您想要返回使用之前的 OpenSearch 版本，此快照可作為備份，您可以在新的網域上還原。
2. 前往 https://aws.amazon.com 並選擇 Sign In to the Console (登入主控台)。
3. 在 Analytics (分析) 下，選擇 Amazon OpenSearch Service。
4. 在導覽窗格中，於 Domains (網域) 下，選擇您想要升級的網域。
5. 選擇 Actions (動作) 和 Upgrade (升級)。
6. 選取要升級到的版本。如果您要升級到 OpenSearch 版本，系統會出現 Enable compatibility mode (啟用相容性模式) 選項。如果您啟用此設定，OpenSearch 會將其版本報告為 7.10，以允許 Elasticsearch OSS 用戶端和外掛程式 (例如 Logstash) 繼續使用 Amazon OpenSearch Service。您可以稍後停用此設定
7. 選擇 Upgrade (升級)。
8. 請檢查網域儀表板上的 Status (狀態) 來監控升級狀態。

升級網域 (CLI)

您可以使用以下操作來確定您的網域的 OpenSearch 或 Elasticsearch 的正確版本、開始就地升級、執行預先升級的檢查並檢視進度：

- `get-compatible-versions` (`GetCompatibleVersions`)
- `upgrade-domain` (`UpgradeDomain`)
- `get-upgrade-status` (`GetUpgradeStatus`)
- `get-upgrade-history` (`GetUpgradeHistory`)

如需詳細資訊，請參閱 AWS CLI 命令參考和 Amazon OpenSearch Service API 參考。

升級網域 (SDK)

此範例使用來自 的 OpenSearchService 低階 Python 用戶端 適用於 Python (Boto) 的 AWS SDK ，來檢查網域是否符合升級至特定版本的資格、將其升級，並持續檢查升級狀態。

```
import boto3
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default Region.

DOMAIN_NAME = ''  # The name of the domain to upgrade
TARGET_VERSION = ''  # The version you want to upgrade the domain to. For example,
 OpenSearch_1.1

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)
client = boto3.client('opensearch', config=my_config)


def check_versions():
    """Determine whether domain is eligible for upgrade"""
    response = client.get_compatible_versions(
        DomainName=DOMAIN_NAME
    )
    compatible_versions = response['CompatibleVersions']
    for i in range(len(compatible_versions)):
        if TARGET_VERSION in compatible_versions[i]["TargetVersions"]:
            print('Domain is eligible for upgrade to ' + TARGET_VERSION)
            upgrade_domain()
```

```
            print(response)
        else:
            print('Domain not eligible for upgrade to ' + TARGET_VERSION)


def upgrade_domain():
    """Upgrades the domain"""
    response = client.upgrade_domain(
        DomainName=DOMAIN_NAME,
        TargetVersion=TARGET_VERSION
    )
    print('Upgrading domain to ' + TARGET_VERSION + '...' + response)
    time.sleep(5)
    wait_for_upgrade()


def wait_for_upgrade():
    """Get the status of the upgrade"""
    response = client.get_upgrade_status(
        DomainName=DOMAIN_NAME
    )
    if (response['UpgradeStep']) == 'UPGRADE' and (response['StepStatus']) ==
 'SUCCEEDED':
        print('Domain successfully upgraded to ' + TARGET_VERSION)
    elif (response['StepStatus']) == 'FAILED':
        print('Upgrade failed. Please try again.')
    elif (response['StepStatus']) == 'SUCCEEDED_WITH_ISSUES':
        print('Upgrade succeeded with issues')
    elif (response['StepStatus']) == 'IN_PROGRESS':
        time.sleep(30)
        wait_for_upgrade()


def main():
    check_versions()


if __name__ == "__main__":
    main()
```

對驗證失敗進行故障排除

當您啟動 OpenSearch 或 Elasticsearch 版本升級時，OpenSearch Service 首先會執行一系列驗證檢查，以確保您的網域有資格進行升級。如果其中任何一項檢查失敗，您會收到通知，其中包含您必須在更新網域之前解決的特定問題。如需潛在問題的清單以及解決問題的步驟，請參閱 the section called "對驗證錯誤進行疑難排解"。

升級疑難排解

就地 升級需要運作狀態良好的網域。您的網域可能不符合升級的資格，或因各種原因而無法升級。下表顯示最常見的問題。

問題	Description
不支援選用外掛程式	當您使用選用外掛程式升級網域時，OpenSearch Service 也會自動升級外掛程式。因此，網域的目標版本也必須支援這些選用的外掛程式。如果網域已安裝不適用於目標版本的選用外掛程式，則升級請求會失敗。
每個節點有太多碎片	OpenSearch 以及 7.x 版本的 Elasticsearch 具有的預設設定為每個節點不超過 1,000 個碎片。如果您目前叢集中的節點超過此設定，則 OpenSearch Service 不允許您升級。如需疑難排解選項，請參閱the section called "超出最大碎片限制"。
處理中的網域	組態變更中的網域 在操作完成後檢查升級資格。
紅色叢集狀態	叢集中一或多個索引是紅色的。如需疑難排解步驟，請參閱the section called "紅色叢集狀態"。
高錯誤率	在嘗試處理請求時，叢集傳回大量的 5xx 錯誤。此問題通常是因為太多同時讀寫請求造成。請考慮降低叢集流量或擴展網域。
分割大腦	分割大腦表示叢集有不只一個主節點，且已分為兩個叢集，這兩個叢集將不會自行重新連結。您也可以使用建議數量的專用主節點來避免大腦分割。如需從大腦分割中恢復的協助，請聯絡 支援。
找不到主節點	OpenSearch Service 無法找到叢集的主節點。如果您的網域使用的是多個可用區，則可用區域故障可能會導致叢集遺失仲裁，且無法選擇新的主節點。如果問題無法自主解決，請聯絡支援。

問題	Description
待處理任務過多	主節點負載太重且待處理任務過多。請考慮降低叢集流量或擴展網域。
儲存磁碟區受損	一或多個節點的磁碟區無法正常運作。此問題通常與其他問題 (例如高錯誤率或待處理任務過多) 一起發生。如果此問題單獨發生且無法自主解決，請聯絡支援。
KMS 金鑰問題	用於加密網域的 KMS 金鑰無法存取或遺失。如需詳細資訊，請參閱the section called “監控靜態加密資料的網域”。
快照處理中	網域正在拍攝快照。在快照完成後檢查升級資格。也確認您是否列出手動快照儲存庫、列出這些儲存庫中的快照與手動拍攝快照。如果 OpenSearch Service 無法確認快照是否在處理中，升級可能會生敗。
快照逾時或故障	預先升級快照耗時太久，無法完成或失敗。請檢查叢集的運作狀態並再試一次。如果問題仍存在，請聯絡 支援。
索引不相容	一或多個索引與目標版本不相容。如果您從舊版 OpenSearch 或 Elasticsearch 中遷移索引，就可能發生此問題。重新編製索引，然後重試。
高磁碟使用量	叢集的磁碟使用量超過 90%。篩除資料或擴展網域，然後重試。
高 JVM 用量	JVM 記憶體壓力超過 75%。 降低叢集的流量或擴展網域，然後重試。
OpenSearch Dashboards 別名問題	`.dashboards` 已設定為別名且映射到不相容的索引 (可能是舊版 OpenSearch Dashboards 的索引)。重新編製索引，然後再試一次。
紅色 Dashboards 狀態	Elasticsearch Dashboards 狀態為紅色。在升級完成時嘗試使用 Dashboards。如果紅色狀態仍持續，手動解決它，然後重試。
跨叢集相容性	只有在可維持升級後來源和目的地網域之間的跨叢集相容性的情況下，才能進行升級。在升級程序期間，系統會識別任何不相容的連線。若要繼續進行，請升級遠端網域或刪除不相容的連線。請注意，如果複寫在網域上處於啟用狀態，一旦刪除連線，就無法繼續複寫。
其他 OpenSearch Service 服務問題	OpenSearch Service 本身的問題可能會使您的網域顯示為不符合升級資格。如果上述問題都不適用於您的網域，且問題持續超過一天，請聯絡支援。

使用快照來遷移資料

就地升級更簡單、更快速且更可靠，可讓您將網域升級至更新的 OpenSearch 或 Elasticsearch 版本。如果您需要從 Elasticsearch 5.1 之前的版本中遷移或想要遷移至全新的叢集，快照會是一個好選擇。

下表說明如何使用快照將資料遷移至使用不同 OpenSearch 或 Elasticsearch 版本的網域。如需有關拍攝和恢復快照的資訊，請參閱the section called “建立索引快照”。

原始版本	目標版本	遷移程序

OpenSearch 1.3 或 2.x	OpenSearch 2.x	1. 檢閱 OpenSearch 2.3 的重大變更，以查看是否需要調整索引或應用程式。
		2. 建立 1.3 或 2.x 網域的手動快照。
		3. 建立比原始 1.3 或 2.x 網域更高版本的 2.x 網域。
		4. 將快照從原始網域還原至 2.x 網域。在操作期間，您可能需要以新名稱還原.opensearch 索引：
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".opensearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		然後，您可以在新網域 .backup-opensearch 上重建索引，並將其別名命名為 .opensearch 。請注意，_restoreREST 呼叫不包含 ，include_global_state 因為 中的預設值_restore為 false。因此，測試網域不會包含任何索引範本，也不會有來自備份的完整狀態。
		5. 如果您不再需要原始網域，請將其刪除。否則，您要繼續負擔網域的費用。

OpenSearch 1.x	OpenSearch 1.x	1. 建立 1.x 網域的手動快照。 2. 建立比原始 1.x 網域更高版本的 1.x 網域。

原始版本	目標版本	遷移程序

		3. 將快照從原始網域還原至新的 1.x 網域。在操作期間，您可能需要以新名稱還原.opensearch 索引：
		
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".opensearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		
		然後，您可以在新網域 .backup-opensearch 上重建索引，並將其別名命名為 .opensearch 。請注意，_restoreREST 呼叫不包含 ，include_global_state 因為 中的預設值_restore為 false。因此，測試網域不會包含任何索引範本，也不會有來自備份的完整狀態。
		4. 如果您不再需要原始網域，請將其刪除。否則，您要繼續負擔網域的費用。

原始版本	目標版本	遷移程序

Elasticsearch 6.x 或 7.x	OpenSearch 1.x	1. 檢閱 OpenSearch 1.0 的重大變更，以查看是否需要調整您的索引或應用程式。
		2. 建立 Elasticsearch 7.x 或 6.x 網域的手動快照。
		3. 建立 OpenSearch 1.x 網域。
		4. 將快照從 Elasticsearch 網域還原到 OpenSearch 網域。在操作期間，您可能需要以新名稱還原.elasticsearch 索引：
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		然後，您可以在新網域 .backup-opensearch 上重建索引，並將其別名命名為 .elasticsearch 。請注意，_restoreREST 呼叫不包含 ，include_global_state 因為 中的預設值_restore為 false。因此，測試網域不會包含任何索引範本，也不會有來自備份的完整狀態。
		5. 如果您不再需要原始網域，請將其刪除。否則，您要繼續負擔網域的費用。

原始版本	目標版本	遷移程序

Elasticsearch 6.x	Elasticsearch 7.x	1. 檢閱 7.0 的重大變更，以查看是否需要調整您的索引或應用程式。
		2. 建立 6.x 網域的手動快照。
		3. 建立 7.x 網域。
		4. 將快照從原始網域還原到 7.x 網域。在操作期間，您可能需要以新名稱還原 `.opensearch` 索引：
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-elasticsearch"`
		`}`
		然後，您可以在新網域 `.backup-elasticsearch` 上重建索引，並將其別名命名為 `.elasticsearch` 。請注意，`_restore`REST 呼叫不包含 `include_global_state` 因為 中的預設值`_restore`為 false。因此，測試網域不會包含任何索引範本，也不會有來自備份的完整狀態。
		5. 如果您不再需要原始網域，請將其刪除。否則，您要繼續負擔網域的費用。

Elasticsearch 6.x	Elasticsearch 6.8	1. 建立 6.x 網域的手動快照。 2. 建立 6.8 網域。 3. 將快照從原始網域還原到 6.8 網域。 4. 如果您不再需要原始網域，請將其刪除。否則，您要繼續負擔網域的費用。

原始版本	目標版本	遷移程序
Elasticsearch 5.x	Elasticsearch 6.x	1. 檢閱 6.0 的重大變更，以得知您是否需要調整您的索引或應用程式。 2. 建立 5.x 網域的手動快照。 3. 建立 6.x 網域。 4. 將快照從原始網域還原到 6.x 網域。 5. 如果您不再需要 5.x 網域，請將其刪除。否則，您要繼續負擔網域的費用。
Elasticsearch 5.x	Elasticsearch 5.6	1. 建立 5.x 網域的手動快照。 2. 建立 5.6 網域。 3. 將快照從原始網域還原到 5.6 網域。 4. 如果您不再需要原始網域，請將其刪除。否則，您要繼續負擔網域的費用。
Elasticsearch 2.3	Elasticsearch 6.x	Elasticsearch 2.3 快照與 6.x 不相容。若要直接將您的資料從 2.3 遷移至 6.x，您必須在新網域手動重新建立索引。 或者，您可以遵照此表中的 2.3 到 5.x 步驟，在全新 5.x 網域中執行 `_reindex` 操作，將您的 2.3 索引轉換為 5.x 索引，接著再遵循 5.x 到 6.x 步驟。
Elasticsearch 2.3	Elasticsearch 5.x	1. 檢閱 5.0 的重大變更，以查看是否需要調整您的索引或應用程式。 2. 建立 2.3 網域的手動快照。 3. 建立 5.x 網域。 4. 將快照從 2.3 網域還原到 5.x 網域。 5. 如果您不再需要 2.3 網域，請將其刪除。否則，您要繼續負擔網域的費用。

原始版本	目標版本	遷移程序
Elasticsearch 1.5	Elasticsearch 5.x	Elasticsearch 1.5 快照與 5.x 不相容。若要將您資料從 1.5 遷移至 5.x，您必須在新網域手動重新建立索引。 **Important** 1.5 快照與 2.3 相容，但 OpenSearch Service 2.3 網域不支援 `_reindex` 操作。因為您無法為其重新編製索引，所以在 1.5 網域中產生的索引仍然無法從 2.3 快照還原到 5.x 網域。
Elasticsearch 1.5	Elasticsearch 2.3	1. 使用遷移外掛程式來確認您是否可以直接升級至 2.3 版。您在遷移前可能需要對資料進行變更。 a. 在 Web 瀏覽器中，開啟 `http://domain-endpoint /_plugin/migration/` 。 b. 選擇 Run checks now (立即執行檢查)。 c. 如有需要，檢閱結果，並依照操作說明來變更您的資料。 2. 建立 1.5 網域的手動快照。 3. 建立 2.3 網域。 4. 將快照從 1.5 網域還原到 2.3 網域。 5. 如果您不再需要 1.5 網域，請將其刪除。否則，您要繼續負擔網域的費用。

為 Amazon OpenSearch Service 建立自訂端點

為您的 Amazon OpenSearch Service 網域建立自訂端點，可讓您更輕鬆地參考 OpenSearch 和 OpenSearch Dashboards URL。您可以加入貴公司的品牌，或只使用比標準端點更短、更容易記憶的端點。

如果您需要切換到新網域，只需更新 DNS 以指向新的 URL 並繼續使用與以前相同的端點。

您可以透過在 AWS Certificate Manager (ACM) 中產生憑證或匯入您自己的憑證來保護自訂端點。

新網域的自訂端點

您可以使用 OpenSearch Service 主控台或組態 API AWS CLI，為新的 OpenSearch Service 網域啟用自訂端點。

若要自訂端點 (主控台)

1. 從 OpenSearch Service 主控台中，選擇 Create domain (建立網域)，然後提供網域名稱。
2. 在 Custom endpoint (自訂端點) 中，選取 Enable custom endpoint (啟用自訂端點)。
3. 對於 Custom hostname (自訂主機名稱)，輸入您偏好的自訂端點主機名稱。主機名稱應該是完整網域名稱 (FQDN)，例如 www.yourdomain.com 或 example.yourdomain.com。

   > **Note**
   >
   > 如果您沒有萬用字元憑證，則必須為自訂端點的子網域獲取新憑證。

4. 對於 AWS 憑證，選擇您要用於此網域的 SSL 憑證。如果沒有可用的憑證，您可以將憑證匯入 ACM 或使用 ACM 來佈建憑證。如需詳細資訊，請參閱 AWS Certificate Manager 使用者指南中的發行和管理憑證。

   > **Note**
   >
   > 憑證必須具有自訂端點名稱，且與您的 OpenSearch Service 網域位於相同的帳戶中。憑證狀態應為 ISSUED (已發佈)。

   - 依照其餘步驟建立網域，然後選擇 Create (建立)。
   - 完成處理後，請選取網域以檢視您的自訂端點。

   若要使用 CLI 或組態 API，請使用 `CreateDomain` 和 `UpdateDomainConfig` 操作。如需詳細資訊，請參閱 AWS CLI 命令參考和 Amazon OpenSearch Service API 參考。

現有網域的自訂端點

若要將自訂端點新增至現有的 OpenSearch Service 網域，請選擇 Edit (編輯)，然後執行上述步驟 2–4。

CNAME 映射

為 OpenSearch Service 網域啟用自訂端點後，您可以在 Amazon Route 53 （或您偏好的 DNS 服務提供者）中建立 CNAME 映射。建立 CNAME 映射可讓您將流量路由到自訂端點及其子網域。如果沒有此映射，您將無法將流量路由到自訂端點。如需在 Route 53 中建立此映射的步驟，請參閱設定新網域的 DNS 路由和為子網域建立新的託管區域。如果是其他供應商，請參閱其文件。

建立 CNAME 記錄，將自訂端點指向自動產生的網域端點。如果您的網域是雙堆疊，您可以將 CNAME 記錄指向兩個服務產生的端點之一。自訂端點的雙堆疊功能取決於您指向 CNAME 記錄的服務產生端點。自訂端點主機名稱是 CNAME 記錄的名稱，而網域端點主機名稱是 CNAME 記錄的值。

如果您使用 OpenSearch Dashboards 的 SAML 身分驗證，則必須使用新的 SSO URL 更新 IdP。

您可以使用 Amazon Route 53 建立別名記錄類型，將網域的自訂端點指向雙堆疊搜尋端點。若要建立別名記錄類型，您必須將網域設定為使用雙堆疊 IP 地址類型。您可以使用 Route 53 API 來執行此操作。

若要使用 Route 53 API 建立別名記錄類型，請指定網域的別名目標。您可以在 OpenSearch Service 主控台的自訂端點區段的託管區域 （雙堆疊） 欄位中找到網域的別名目標，或使用 `DescribeDomain` API 複製 的值`DomainEndpointV2HostedZoneId`。

Amazon OpenSearch Service 的自動調整

Amazon OpenSearch Service 中的自動調整會使用 OpenSearch 叢集中的效能和使用量指標，以建議與記憶體相關的組態變更，包括節點上的佇列和快取大小以及 Java 虛擬機器 (JVM) 設定。這些選擇性變更可提高叢集速度與穩定性。

有些變更會立即部署，而其他變更則會在網域的離峰時段期間排程。您可以隨時還原為預設的 OpenSearch Service 設定。由於自動調整會收集並分析網域的效能指標，因此您可以在 OpenSearch Service 主控台的通知頁面中檢視其建議。

Auto-Tune 可在具有支援執行個體類型的執行任何 OpenSearch 版本或 Elasticsearch 6.7 或更新版本的網域 AWS 區域 上商業上使用。

變更類型

自動調整有兩大類變更：

- 在叢集執行時套用的非破壞性變更。
- 需要藍/綠部署的變更，適用於網域的離峰時段。

根據您網域的效能指標，自動調整可建議調整下列設定：

變更類型	類別	描述
JVM 堆積大小	藍/綠	依預設，OpenSearch Service 針對 JVM 堆積使用執行個體 RAM 的 50%，堆積大小最多可達 32 GiB。 增加此百分比可為 OpenSearch 提供更多記憶體，但對作業系統和其他處理程序而言保留的更少。較大的值可以減少廢棄項目收集暫停的數目，但會增加這些暫停的長度。
JVM 新一代設定	藍/綠	JVM「新一代」設定會影響次要廢棄項目收集的頻率。較頻繁的次要收集可減少主要收集和暫停的數目。
佇列大小	不中斷	依預設，搜尋佇列大小為 `1000`，寫入佇列大小為 `10000`。如果有額外的堆積可用來處理請求，自動調整會自動擴展搜尋和寫入佇列。
快取大小	不中斷	欄位快取會監控堆積內的資料結構，因此監控快取的使用非常重要。自動調整可擴展欄位資料快取大小，以避免記憶體不足和斷路器問題。 碎片請求快取在節點級別進行管理，並且默認的大小上限為 1% 的堆積。自動調整可擴展碎片請求快取大小，以接受比設定的叢集可處理的請求更多的搜尋和索引請求。
請求規模	不中斷	在預設情況下，在進行中請求的匯總規模超過總 JVM 的 10% 時 (`t2` 執行個體類型的 2%，以及 `t3.small` 的 1%)，在現有請求完成之前，OpenSearch 會調節所有新的 `_search` 和 `_bulk` 請求。 自動調整會依據目前系統占用的 JVM 量自動調整此閾值，通常介於 5-15% 之間。例如，如果 JVM 記憶體壓力很高，則自動調整可能會將閾值降低至 5%，此時在叢集穩定且閾值增加之前，您可能會看到更多的拒絕數。

啟用或停用自動調整

OpenSearch Service 預設會在新網域上啟用自動調整。若要在現有網域上啟用或停用自動調整，建議您使用 主控台，以簡化程序。啟用自動調整不會導致藍/綠部署。

您目前無法使用 AWS CloudFormation來啟用或停用自動調整。

主控台

在現有網域上啟用自動調整

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在導覽窗格的網域下，選擇網域名稱以開啟叢集組態。
3. 如果尚未啟用自動調整，請選擇開啟。
4. 或者，選取離峰時段，以排程在網域設定的離峰時段期間需要藍/綠部署的最佳化。如需詳細資訊，請參閱the section called “排程自動調整增強功能”。
5. 選擇 Save changes (儲存變更)。

CLI

若要使用 啟用自動調整 AWS CLI，請傳送 UpdateDomainConfig 請求：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options DesiredState=ENABLED
```

排程自動調整增強功能

在 2023 年 2 月 16 日之前，自動調整使用維護時段來排程需要藍/綠部署的變更。維護時段現在已棄用，因為離峰時段是每日 10 小時的時段，在此期間您的網域通常會遇到低流量。您可以修改離峰時段的預設開始時間，但無法修改長度。

在 2023 年 2 月 16 日推出離峰時段之前，已啟用自動調整維護時段的任何網域都可以繼續使用舊版維護時段，而不會中斷。不過，我們建議您遷移現有的網域，改為使用離峰時段進行網域維護。如需說明，請參閱the section called “從自動調整維護時段遷移”。

主控台

排程離峰時段的自動調整動作

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在導覽窗格的網域下，選擇網域名稱以開啟叢集組態。

3. 前往自動調整索引標籤，然後選擇編輯。
4. 如果尚未啟用自動調整，請選擇開啟。
5. 在離峰時段的排程最佳化下，選取離峰時段。
6. 選擇儲存變更。

CLI

若要設定您的網域在設定的離峰時段期間排程自動調整動作，請在 UpdateDomainConfig 請求UseOffPeakWindow中包含 ：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=null
```

監控自動調整變更

您可以在 中監控 Auto-Tune 統計資料 Amazon CloudWatch。如需指標的完整清單，請參閱 the section called “自動調整指標”。

OpenSearch Service 會將 Auto-Tune 事件傳送至 Amazon EventBridge。您可以使用 EventBridge 設定傳送電子郵件的規則，或在接收事件時執行特定動作。若要查看傳送至 EventBridge 的每個自動調整事件的格式，請參閱 the section called “自動調整事件”。

標記 Amazon OpenSearch Service 網域

標籤可讓您將任意資訊指派給 Amazon OpenSearch Service 網域，以便對該資訊進行分類和篩選。標籤是一個鍵值對，您可定義它並與 OpenSearch Service 網域建立關聯。您可以使用這些標籤，透過將相似標記資源的費用分組來追蹤成本。 AWS 不會將任何語意意義套用至您的標籤。標籤會嚴格解譯為字元字串。所有標籤均包含以下元素：

標籤元素	Description	必要
標籤鍵	標籤金鑰是標籤名稱。鍵對於所連接的 OpenSearch Service 網域必須具有唯一性。如需標籤鍵和標籤值各項基本限制的清單，請參閱使用者定義的標籤限制。	是

標籤元素	Description	必要
標籤值	標籤值即為標籤的字串值。標籤值可以是 null，並且在標籤集內不必具有唯一性。例如，在 project/Trinity 及 cost-center/Trinity 標籤集中，均能擁有一個索引鍵/值組。如需標籤鍵和標籤值各項基本限制的清單，請參閱使用者定義的標籤限制。	否

每個 OpenSearch Service 網域皆有標籤集，其中包含指派給該 OpenSearch Service 網域的所有標籤。 AWS 不會自動將任何標籤指派給 OpenSearch Service 網域。標籤集可以包含 0 到 50 個標籤。如果您對網域新增標籤，而該標籤與現有標籤具有相同的鍵，則新值會覆寫舊值。

標記範例

您可以使用鍵來定義類別，其值可為該類別中的某個項目。例如，您可以定義標籤鍵為 `project` 且標籤值為 `Salix`，表示 OpenSearch Service 網域已指派給 Salix 專案。您也可利用 `environment=test` 或 `environment=production` 之類的鍵，以使用標籤來指定用於測試或生產的 OpenSearch Service 網域。嘗試使用一組一致的標籤鍵，讓您更方便追蹤與 OpenSearch Service 網域關聯的中繼資料。

您也可以使用標籤來整理 AWS 帳單，以反映您自己的成本結構。若要這樣做，請註冊 以取得包含標籤索引鍵值的 AWS 帳戶 帳單。接著，根據具有相同標籤鍵值的資源來整理您的帳單資訊，以便查看合併資源的成本。例如，您可以使用鍵值對為數個 OpenSearch Service 網域加上標籤，然後整理帳單資訊以查看數項服務的每個網域的總成本。如需詳細資訊，請參閱 https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html 帳單與成本管理文件中的AWS 使用成本分配標記。

> **Note**
>
> 標籤是快取用於授權之用。因此，對 OpenSearch Service 網域新增及更新標籤可能需要幾分鐘之後才會生效。

標記網域（主控台）

主控台是標記網域的最簡單方法。

建立標籤 (主控台)

1. 前往 https://aws.amazon.com，然後選擇 Sign In to the Console (登入主控台)。
2. 在 Analytics (分析) 下，選擇 Amazon OpenSearch Service。
3. 選取您要新增標籤的目標網域，然後前往 Tags (標籤) 索引標籤。
4. 選擇 Manage (管理) 和 Add new tag (新增標籤)。
5. 輸入標籤索引鍵和選用的值。
6. 選擇儲存。

若要刪除標籤，請按照同樣的步驟進行，然後在 Manage tags (管理標籤) 頁面上選擇 Remove (移除)。

如需使用主控台處理標籤的詳細資訊，請參閱 《AWS 管理主控台入門指南》中的標籤編輯器。

標記網域 (AWS CLI)

您可以使用 AWS CLI 搭配 --add-tags命令來建立資源標籤。

語法

```
add-tags --arn=<domain_arn> --tag-list Key=<key>,Value=<value>
```

參數	Description
`--arn`	要為其附加標籤的 OpenSearch Service 網域的 Amazon 資源名稱。
`--tag-list`	以空格分隔且格式如下的一系列鍵值組：`Key=<key>,Value=<value>`

範例

以下範例為 logs 網域建立兩個標籤：

```
aws opensearch add-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-list
 Key=service,Value=OpenSearch Key=instances,Value=m3.2xlarge
```

您可以使用 --remove-tags 命令，從 OpenSearch Service 網域移除標籤。

語法

```
remove-tags --arn=<domain_arn> --tag-keys Key=<key>,Value=<value>
```

參數	Description
`--arn`	要為其附加標籤的 OpenSearch Service 網域的 Amazon Resource Name (ARN)。
`--tag-keys`	要從 OpenSearch Service 網域中移除的一組以空格分隔的鍵值對。

範例

以下範例會將上述範例所建立的兩個標籤從 logs 網域移除：

```
aws opensearch remove-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-keys service instances
```

您可以使用 --list-tags 命令來檢視 OpenSearch Service 網域的現有標籤：

語法

```
list-tags --arn=<domain_arn>
```

參數	Description
`--arn`	要為其附加標籤的 OpenSearch Service 網域的 Amazon Resource Name (ARN)。

範例

以下範例會列出 logs 網域的所有資源標籤：

```
aws opensearch list-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs
```

標記網域 (AWS SDKs)

AWS SDKs(Android 和 iOS SDKs除外）支援 Amazon OpenSearch Service API 參考中定義的所有動作，包括 AddTags、 ListTags和 RemoveTags操作。如需安裝和使用 AWS SDKs 的詳細資訊，請參閱AWS 軟體開發套件。

Python

此範例使用適用於 Python 的 AWS SDK (Boto) 中的 OpenSearchService 低階 Python 用戶端，將標籤新增至網域，列出連接至網域的標籤，並從網域中移除標籤。您必須提供 DOMAIN_ARN、TAG_KEY 和 TAG_VALUE 的值。

```
import boto3
from botocore.config import Config  # import configuration

DOMAIN_ARN = ''  # ARN for the domain. i.e "arn:aws:es:us-east-1:123456789012:domain/
my-domain
TAG_KEY = ''  # The name of the tag key. i.e 'Smileyface'
TAG_VALUE = ''  # The value assigned to the tag. i.e 'Practicetag'

# defines the configurations parameters such as region

my_config = Config(region_name='us-east-1')
client = boto3.client('opensearch', config=my_config)


# defines the client variable

def addTags():
    """Adds tags to the domain"""

    response = client.add_tags(ARN=DOMAIN_ARN,
                               TagList=[{'Key': TAG_KEY,
                                         'Value': TAG_VALUE}])

    print(response)


def listTags():
    """List tags that have been added to the domain"""

    response = client.list_tags(ARN=DOMAIN_ARN)
    print(response)


def removeTags():
    """Remove tags that have been added to the domain"""

    response = client.remove_tags(ARN=DOMAIN_ARN, TagKeys=[TAG_KEY])
```

```
    print('Tag removed')
    return response
```

在 Amazon OpenSearch Service 網域上執行管理動作

如果您需要對網域的問題進行疑難排解，Amazon OpenSearch Service 提供多種管理選項，可提供精細的控制。這些選項包括在資料節點上重新啟動 OpenSearch 程序的功能，以及重新啟動資料節點的功能。

OpenSearch Service 會監控節點運作狀態參數，並在發生異常時採取修正動作來保持網域穩定。透過在節點上重新啟動 OpenSearch 程序並重新啟動節點本身的管理選項，您可以控制其中一些緩解動作。

您可以使用 AWS 管理主控台 AWS CLI或 AWS SDK 來執行這些動作。下列各節說明如何使用 主控台執行這些動作。

在 Amazon OpenSearch Service 中的節點上重新啟動 OpenSearch 程序

在節點上重新啟動 OpenSearch 程序

1. 導覽至位於 的 OpenSearch Service 主控台https://console.aws.amazon.com/aos/。
2. 在左側導覽窗格中選擇 Domains (網域)。選擇您要使用的網域名稱。
3. 開啟網域詳細資訊頁面後，導覽至執行個體運作狀態索引標籤。
4. 在資料節點下，選取您要重新啟動程序的節點旁的按鈕。
5. 選取動作下拉式清單，然後選擇重新啟動 OpenSearch/Elasticsearch 程序。
6. 在模態上選擇確認。
7. 若要查看您啟動的動作狀態，請選取節點的名稱。開啟節點詳細資訊頁面後，請選擇節點名稱下方的事件索引標籤，以查看與該節點相關聯的事件清單。

在 Amazon OpenSearch Service 中重新啟動資料節點

重新啟動資料節點

1. 導覽至位於 的 OpenSearch Service 主控台https://console.aws.amazon.com/aos/。
2. 在左側導覽窗格中選擇 Domains (網域)。選擇您要使用的網域名稱。

3. 開啟網域詳細資訊頁面後，導覽至執行個體運作狀態索引標籤。
4. 在資料節點下，選取您要重新啟動程序的節點旁的按鈕。
5. 選取動作下拉式清單，然後選擇重新啟動節點。
6. 在模態上選擇確認。
7. 若要查看您啟動的動作狀態，請選取節點的名稱。開啟節點詳細資訊頁面後，請選擇節點名稱下方的事件索引標籤，以查看與該節點相關聯的事件清單。

在 Amazon OpenSearch Service 中的節點上重新啟動 OpenSearch Dashboards 程序

您可以重新啟動 OpenSearch Dashboards （ 先前為 Kibana) 程序，從凍結界面、載入失敗或無法回應的視覺化等問題中復原。重新啟動 OpenSearch Dashboards 的選項僅適用於正在執行 Dashboards 程序的節點。在大多數 OpenSearch Service 網域中，此程序會在專用協調器節點上執行，而不是資料節點。因此，當您在主控台中開啟動作下拉式清單時， 選項通常只會針對協調器節點顯示。如需詳細資訊，請參閱the section called “專用協調器節點”。

此行為取決於您的網域設定方式。

- 具有專用協調器節點的網域 – Dashboards 程序只會在這些節點上執行，而且只會顯示重新啟動選項。
- 沒有專用協調器節點的網域 – 在較簡單或較舊的設定中，儀表板可能會改為在資料節點上執行，重新啟動選項會顯示在那裡。
- 主節點 – 這些節點僅用於管理叢集狀態和選擇。它們不會執行 Dashboards，而且永遠不會顯示重新啟動選項。

若要判斷哪個節點正在執行 Dashboards 程序，請導覽至您網域的叢集組態區段，並檢閱節點角色。重新啟動的選項僅適用於託管 Dashboards 程序的節點。

在節點上重新啟動 Dashboards 程序

1. 導覽至位於 的 OpenSearch Service 主控台https://console.aws.amazon.com/aos/。
2. 在左側導覽窗格中選擇 Domains (網域)。選擇您要使用的網域名稱。
3. 開啟網域詳細資訊頁面後，導覽至執行個體運作狀態索引標籤。
4. 在執行 Dashboards 程序的節點區段中，選取要重新啟動程序的節點旁的按鈕。
5. 選取動作下拉式清單，然後選擇重新啟動儀表板/Kibana 程序。

6. 在模態上選擇確認。
7. 若要查看您啟動的動作狀態，請選取節點的名稱。開啟節點詳細資訊頁面後，請選擇節點名稱下方的事件索引標籤，以查看與該節點相關聯的事件清單。

限制

管理選項有下列限制：

- Elasticsearch 7.x 版和更新版本支援管理選項。
- 管理選項不支援啟用待命的異地同步備份網域。
- 具有三個或更多資料節點的網域支援 OpenSearch 和 Elasticsearch 程序重新啟動和資料節點重新啟動。
- 具有兩個或多個資料節點的網域支援 Dashboards 和 Kibana 程序。
- 若要重新啟動節點上的 OpenSearch 程序或重新啟動節點，網域不得處於紅色狀態，且所有索引都必須設定複本。

使用 Amazon OpenSearch Service 直接查詢

使用 Amazon OpenSearch Service 直接查詢來分析 Amazon CloudWatch Logs、Amazon S3 和 Amazon Security Lake 中的資料，而無需建置擷取管道。此零 ETL 整合可讓您使用 OpenSearch SQL 或 PPL 查詢已就地的資料，並在探索中探索資料。

若要開始使用，請在 OpenSearch Service 主控台中設定您的資料來源。對於 Amazon S3，請使用網域連線，並在 Query Workbench 中建立具有 SQL 的資料表。CloudWatch Logs 和 Security Lake 使用預先設定的來源和 AWS Glue Data Catalog 資料表。

直接查詢定價

當您使用 OpenSearch Service 直接查詢時，您會為 OpenSearch Service 和用於處理和儲存資料的資源產生個別費用，並存放在 Amazon S3、Amazon CloudWatch Logs 和 Amazon Security Lake。當您執行直接查詢時，您會看到每小時 OpenSearch 運算單位 (OCUs的費用，在您的帳單上列為 DirectQuery OCU 用量類型。

直接查詢有兩種類型：互動式和索引化檢視查詢。

- 互動式查詢用於填入資料選擇器，並對 S3、CloudWatch Logs 或 Security Lake 中的資料進行分析。

  對於 Amazon S3 直接查詢，當您從探索執行新查詢時，OpenSearch Service 會啟動持續至少三分鐘的新工作階段。OpenSearch Service 會保持此工作階段處於作用中狀態，以確保後續查詢快速執行。

  對於 CloudWatch Logs 和 Security Lake 查詢，OpenSearch Service 會使用個別的預熱任務來處理每個查詢，而無需維護延伸工作階段。
- 索引檢視查詢使用運算來維護 OpenSearch Service 中的索引檢視。這些查詢通常需要更長的時間，因為它們會將不同數量的資料擷取至具名索引。透過將資料編製索引，您可以讓未來的互動式查詢更快地執行，或解鎖進階分析功能，例如儀表板或提醒，這些功能需要索引以供參考。

  對於 Amazon S3 資料來源，索引資料會根據購買的執行個體類型儲存在網域中。對於 CloudWatch Logs 和 Security Lake 連線的資料來源，索引資料會存放在 OpenSearch Serverless 集合中，其中會向您收取資料索引 (IndexingOCU)、資料搜尋 (SearchOCU) 和以 GB 形式存放的資料的費用。

如需詳細資訊，請參閱 Amazon OpenSearch Service 定價中的直接查詢和無伺服器區段。

直接查詢限制

一般限制

下列限制適用於 OpenSearch Service 直接查詢。

- 不支援某些資料類型。支援的資料類型僅限於 Parquet、CSV 和 JSON。
- 如果資料結構隨時間變更，您將需要更新索引檢視或out-of-the-box整合，以考慮資料結構變更。
- AWS CloudFormation 尚不支援 範本。
- 相較於使用直接查詢，使用 OpenSearch 索引時，OpenSearch SQL 和 OpenSearch PPL 陳述式有不同的限制。 OpenSearch 直接查詢支援 JOINs、子查詢和查詢等進階命令，而對 OpenSearch 索引上這些命令的支援有限或不存在。如需詳細資訊，請參閱the section called “支援的 SQL 和 PPL 命令”。

Amazon S3 的限制

如果您要直接在 Amazon S3 中查詢資料，則適用下列其他限制：

- S3 的直接查詢僅適用於執行 OpenSearch 2.13 版或更新版本的 OpenSearch Service 網域，且需要存取 AWS Glue Data Catalog。必須使用 OpenSearch Query Workbench 中的 SQL 重新建立現有 AWS Glue Data Catalog 資料表。
- S3 的直接查詢需要您在 Amazon S3 上指定檢查點儲存貯體。此儲存貯體會維護索引檢視的狀態，包括上次重新整理時間和最近擷取的資料。
- 您的 OpenSearch 網域和 AWS Glue Data Catalog 必須位於相同的 中 AWS 帳戶。您的 S3 儲存貯體可以位於不同的帳戶中 （需要將條件新增至您的 IAM 政策），但必須與 AWS 區域 網域位於相同的 中。
- 使用 S3 的 OpenSearch Service 直接查詢僅支援從 Query Workbench 產生的 Spark 資料表。Spark 串流不支援在 AWS Glue Data Catalog 或 Athena 中產生的資料表，這是維護索引檢視所需的項目。
- OpenSearch 執行個體類型的網路承載限制為 10 MiB 或 100 MiB，視您選擇的特定執行個體類型而定。

Amazon CloudWatch Logs 的限制

如果您要直接查詢 CloudWatch Logs 中的資料，則適用下列其他限制：

- 與 CloudWatch Logs 的直接查詢整合僅適用於 OpenSearch Service 集合和 OpenSearch 使用者介面。
- OpenSearch Serverless 集合的網路承載限制為 100 MiB。
- CloudWatch Logs 支援從主控台安裝的 VPC Flow、CloudTrail 和 AWS WAF 儀表板整合。

Amazon Security Lake 的限制

如果您要直接在 Security Lake 中查詢資料，則適用下列其他限制：

- 與 Security Lake 的直接查詢整合僅適用於 OpenSearch Service 集合和 OpenSearch 使用者介面。
- OpenSearch Serverless 集合的網路承載限制為 100 MiB。
- Security Lake 的資料表管理是在 Lake Formation 中執行。
- Security Lake 僅支援具體化視觀表做為索引化視觀表。不支援覆蓋索引。

在 Amazon OpenSearch Service 中使用直接查詢的建議

此頁面提供使用 Amazon OpenSearch Service 直接查詢來分析 CloudWatch Logs、Amazon S3 和 Amazon Security Lake 資料的建議。這些最佳實務可協助您最佳化效能，並確保有效率的查詢，而不需要資料擷取或複製。

主題

- 一般建議
- Amazon S3 建議
- CloudWatch Logs 建議
- Security Lake 建議

一般建議

建議您在使用直接查詢時執行下列動作：

- 使用 `COALESCE SQL`函數來處理遺失的資料欄，並確保傳回結果。

- 對查詢使用限制，以確保您不會提取太多資料。

- 如果您打算分析相同的資料集多次，請建立索引檢視，將資料完全擷取並索引至 OpenSearch，並在完成分析時將其捨棄。
- 不再需要加速任務和索引時將其捨棄。
- 不支援包含相同但僅在 （ 例如 `field1`和 `FIELD1`) 的情況下才不同的欄位名稱的查詢。

  例如，不支援下列查詢：

  ```
  Select AWSAccountId, AwsAccountId from LogGroup
  Select a.@LogStream, b.@logStream from Table A INNER Join Table B ona.id = b.id
  ```

  不過，由於兩個日誌群組中的欄位名稱 (@logStream) 相同，因此支援下列查詢：

  ```
  Select a.@logStream, b.@logStream from Table A INNER Join Table B on a.id = b.id
  ```

- 函數和表達式必須在欄位名稱上操作，並成為具有 `FROM`子句中指定之日誌群組的`SELECT`陳述式的一部分。

  例如，不支援此查詢：

  ```
  SELECT cos(10) FROM LogGroup
  ```

  支援此查詢：

  ```
  SELECT cos(field1) FROM LogGroup
  ```

Amazon S3 建議

如果您使用 Amazon OpenSearch Service 在 Amazon S3 中直接查詢資料，我們也建議您執行下列動作：

- 使用年、月、日、小時的分割區格式將資料擷取至 Amazon S3，以加速查詢。
- 當您建置略過索引時，請針對具有高基數的欄位使用 Bloom 篩選條件，並針對具有較大值範圍的欄位使用最低/最高索引。對於高基數欄位，請考慮使用以值為基礎的方法來改善查詢效率。
- 使用索引狀態管理來維護具體化視觀表和涵蓋索引的儲存體。

CloudWatch Logs 建議

如果您使用 Amazon OpenSearch Service 來引導 CloudWatch Logs 中的查詢資料，我們也建議您執行下列動作：

- 在一個查詢中搜尋多個日誌群組時，請使用適當的語法。如需詳細資訊，請參閱the section called "多日誌群組函數"。
- 使用 SQL 或 PPL 命令時，請將特定欄位括在反引號中，以成功查詢它們。具有特殊字元（非字母和非數字）的欄位需要反引號。例如，將 `@messageOperation.Export,`和 括在反引號`Test::Field`中。您不需要將純字母名稱的資料欄括在反引號中。

  具有簡單欄位的範例查詢：

  ```
  SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
  LIMIT 1000;
  ```

  附加反引號的類似查詢：

  ```
  SELECT `@SessionToken`, `@Operation`, `@StartTime` FROM `LogGroup-A`
  LIMIT 1000;
  ```

Security Lake 建議

如果您使用 Amazon OpenSearch Service 在 Security Lake 中直接查詢資料，我們也建議您執行下列動作：

- 檢查您的 Security Lake 狀態，並確保其順利執行，沒有任何問題。如需詳細的故障診斷步驟，請參閱《Amazon Security Lake 使用者指南》中的對資料湖狀態進行故障診斷。
- 驗證您的查詢存取：
  - 如果您要從與 Security Lake 委派管理員帳戶不同的帳戶查詢 Security Lake，請在 Security Lake 中設定具有查詢存取權的訂閱者。
  - 如果您要從相同帳戶查詢 Security Lake，請檢查 Security Lake 中有關向 LakeFormation 註冊受管 S3 儲存貯體的任何訊息。
- 探索查詢範本和預先建置的儀表板，以快速開始分析。
- 熟悉開放式網路安全結構描述架構 (OCSF) 和 Security Lake：
  - 檢閱 OCSF GitHub 儲存庫中 AWS 來源的結構描述映射範例

  - 了解如何造訪AWS 來源版本 2 (OCSF 1.1.0) 的 Security Lake 查詢，以有效地查詢 Security Lake
  - 使用分割區來改善查詢效能：`region`、 `accountid`和 `time_dt`
- 熟悉 Security Lake 支援用於查詢的 SQL 語法。如需詳細資訊，請參閱the section called “支援的 SQL 命令”。

直接查詢配額

您的帳戶具有下列與 OpenSearch Service 直接查詢相關的配額。

Amazon S3 的配額

每次您啟動對 Amazon S3 資料來源的查詢時，OpenSearch Service 都會開啟工作階段並保持其運作狀態至少三分鐘。這可透過移除後續查詢中的工作階段啟動時間來減少查詢延遲。

Description	上限	可以覆寫
每個網域的連線數	10	是
每個網域的資料來源	20	是
每個網域的索引	5	是
每個資料來源的並行工作階段	10	是
每個查詢的最大 OCU	60	是
查詢執行時間上限（分鐘）	30	是
每個加速的最大 OCUs	20	是
最大暫時性儲存	20	是

CloudWatch Logs 的配額

> **Note**
>
> 如果您想要使用 CloudWatch Logs Insights 執行直接查詢，請務必參考 the section called "CloudWatch Logs 洞察"。

Description	Value	軟性限制？	備註
跨直接查詢 APIs 的帳戶層級 TPS 限制	3 TPS	是	
資料來源數量上限	20	是	限制為 AWS 帳戶。
自動重新整理索引或具體化視觀表上限	30	是	限制是每個資料來源。
並行查詢上限	30	是	限制取決於資料來源，並適用於處於待定或執行狀態的查詢。 包括互動式查詢 （例如 等資料擷取命令SELECT) 和索引查詢 （例如 CREATE/ 等操作ALTER)。
每個查詢的並行 OCU 上限	512	是	OpenSearch 運算單位 (OCU)。根據 15 個執行器和 1 個驅動程式的限制，每個都具有 16 個 vCPU 和 32 GB 記憶體。代表並行處理能力。
查詢執行時間上限，以分鐘為單位	60	否	限制適用於 CloudWatch Logs Insights 中的 OpenSearch PPL/SQL 查詢。
清除過時查詢 IDs 的期間	90 天	是	這是 OpenSearch Service 清除舊項目查詢中繼資料的期間。例如，對超過 90 天的查詢呼叫 GetDirect Query 或 GetDirectQueryResult 失敗。

Security Lake 的配額

Description	Value	軟性限制？	備註
跨直接查詢 APIs 的帳戶層級 TPS 限制	3 TPS	是	
資料來源數量上限	20	是	限制為 AWS 帳戶。
自動重新整理索引或具體化視觀表上限	30	是	限制適用於每個資料來源。 僅包含自動重新整理設定為 true MVs)。
並行查詢上限	30	是	限制適用於處於待定或執行狀態的查詢。 包括互動式查詢 （例如 等資料擷取命令SELECT) 和索引查詢 （例如 CREATE/ALTER/ 等操作DROP)。
每個查詢的並行 OCU 上限	512	是	OpenSearch 運算單位 (OCU)。根據 15 個執行器和 1 個驅動程式的限制，每個都具有 16 個 vCPU 和 32 GB 記憶體。代表並行處理能力。
查詢執行時間上限，以分鐘為單位	30	否	僅適用於互動式查詢 （例如 等資料擷取命令SELECT)。對於REFRESH查詢，限制為 6 小時。
清除過時查詢 IDs 的期間	90 天	是	這是 OpenSearch Service 清除舊項目查詢中繼資料的期間。例如，對超過 90 天的查詢呼叫 GetDirectQuery 或 GetDirectQueryResult 失敗。

支援的 AWS 區域

Amazon S3、CloudWatch Logs 和 Security Lake 中的 OpenSearch Service 直接查詢 AWS 區域 支援下列項目：

AWS 區域 適用於 Amazon S3

- 亞太地區 (香港)

- Asia Pacific (Mumbai)
- 亞太區域 (首爾)
- 亞太區域 (新加坡)
- 亞太地區 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)
- 歐洲 (法蘭克福)
- 歐洲 (愛爾蘭)
- 歐洲 (斯德哥爾摩)
- 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 區域 適用於 CloudWatch Logs

- 亞太地區 (孟買)
- 亞太地區 (香港)
- 亞太地區 (大阪)
- 亞太區域 (首爾)
- 亞太區域 (新加坡)
- 亞太地區 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)
- 歐洲 (法蘭克福)
- 歐洲 (愛爾蘭)
- 歐洲 (斯德哥爾摩)
- 歐洲 (米蘭)
- 歐洲 (西班牙)
- 美國東部 (維吉尼亞北部)
- 美國東部 (俄亥俄)
- 美國西部 (奧勒岡)

- 美國西部 (加利佛尼亞北部)
- Europe (Paris)
- 歐洲 (倫敦)
- 南美洲 (聖保羅)

AWS 區域 適用於 Security Lake

- 亞太地區 (孟買)
- 亞太地區 (新加坡)
- 亞太地區 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)
- 歐洲 (法蘭克福)
- 歐洲 (愛爾蘭)
- 歐洲 (斯德哥爾摩)
- 美國東部 (維吉尼亞北部)
- 美國東部 (俄亥俄)
- 美國西部 (奧勒岡)
- Europe (Paris)
- 歐洲 (倫敦)
- 南美洲 (聖保羅)

直接查詢 OpenSearch Service 中的 Amazon S3 資料

本節將逐步引導您在 Amazon OpenSearch Service 中建立和設定資料來源整合的程序，讓您有效率地查詢和分析存放在 Amazon S3 中的資料。

在以下頁面中，您將了解如何設定 Amazon S3 直接查詢資料來源、導覽必要的先決條件，以及使用 AWS 管理主控台 和 OpenSearch Service API step-by-step程序。它還涵蓋了重要的後續步驟，包括映射 AWS Glue Data Catalog 角色和在 OpenSearch Dashboards 中設定存取控制。

主題

- 在 OpenSearch Service 中建立 Amazon S3 資料來源整合

- 在 OpenSearch Dashboards 中設定和查詢 S3 資料來源

在 OpenSearch Service 中建立 Amazon S3 資料來源整合

您可以透過 AWS 管理主控台 或 API 為 OpenSearch Service 建立新的 Amazon S3 直接查詢資料來源。每個新資料來源都會使用 AWS Glue Data Catalog 來管理代表 Amazon S3 儲存貯體的資料表。

主題

- 先決條件
- 程序
- 後續步驟
- 映射 AWS Glue Data Catalog 角色
- 其他資源

先決條件

開始之前，請確定您已檢閱下列文件：

- the section called “Amazon S3 的限制”
- the section called “Amazon S3 建議”
- the section called “Amazon S3 的配額”

您必須先在 中擁有下列資源，才能建立資料來源 AWS 帳戶：

- 2.13 版或更新版本的 OpenSearch 網域。這是設定直接查詢整合的基礎。如需設定的指示，請參閱 the section called “建立 OpenSearch Service 網域”。
- 一或多個 S3 儲存貯體。您需要指定包含要查詢之資料的儲存貯體，以及存放查詢檢查點的儲存貯體。如需建立 S3 儲存貯體的指示，請參閱《Amazon S3 使用者指南》中的建立儲存貯體。
- （選用）一或多個 AWS Glue 資料表。 在 Amazon S3 上查詢資料需要您在 中設定資料表 AWS Glue Data Catalog 以指向 S3 資料。您必須使用 OpenSearch Query Workbench 建立資料表。現有的 Hive 資料表不相容。

  如果這是您第一次設定 Amazon S3 資料來源，您必須建立管理員資料來源來設定所有 AWS Glue Data Catalog 資料表。您可以安裝 OpenSearch out-of-the-box整合，或使用 OpenSearch Query Workbench 為進階使用案例建立自訂 SQL 資料表。如需為 VPC、CloudTrail 和 AWS WAF 日誌建

立資料表的範例，請參閱 GitHub for VPC、CloudTrail 和 上的文件AWS WAF。建立資料表之後，您可以建立新的 Amazon S3 資料來源，並限制對有限資料表的存取。

- （選用）手動建立的 IAM 角色。 您可以使用此角色來管理對資料來源的存取。或者，您可以讓 OpenSearch Service 自動為您建立具有所需許可的角色。如果您選擇使用手動建立的 IAM 角色，請遵循 中的指引the section called “手動建立 IAM 角色的必要許可”。

程序

您可以使用 AWS 管理主控台 或 OpenSearch Service API 在網域上設定直接查詢資料來源。

使用 設定資料來源 AWS 管理主控台

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇 Domains (網域)。
3. 選取您要為其設定新資料來源的網域。這會開啟網域詳細資訊頁面。
4. 選擇一般網域詳細資訊下方的連線索引標籤，然後尋找直接查詢區段。
5. 選擇設定資料來源。
6. 輸入新資料來源的名稱和選用描述。
7. 選擇具有 的 Amazon S3 AWS Glue Data Catalog。
8. 在 IAM 許可存取設定下，選擇如何管理存取。
    a. 如果您想要自動為此資料來源建立角色，請遵循下列步驟：
        i. 選取建立新角色。
        ii. 輸入 IAM 角色的名稱。
        iii. 選取一或多個包含您要查詢之資料的 S3 儲存貯體。
        iv. 選取要存放查詢檢查點的檢查點 S3 儲存貯體。
        v. 選取一或多個 AWS Glue 資料庫或資料表，以定義可查詢哪些資料。如果尚未建立資料表，請提供預設資料庫的存取權。
    b. 如果您想要使用自己管理的現有角色，請遵循下列步驟：
        i. 選取使用現有角色。
        ii. 從下拉式功能表中選取現有角色。

> **Note**
>
> 使用您自己的角色時，您必須從 IAM 主控台連接必要的政策，以確保它具有所有必要的許可。如需詳細資訊，請參閱 中的範例政策the section called “手動建立 IAM 角色的必要許可”。

9. 選擇設定。這會使用 OpenSearch Dashboards URL 開啟資料來源詳細資訊畫面。您可以導覽至此 URL 以完成後續步驟。

OpenSearch Service API

使用 AddDataSource API 操作在您的網域中建立新的資料來源。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource

{
   "DataSourceType": {
        "S3GlueDataCatalog": {
            "RoleArn": "arn:aws:iam::account-id:role/role-name"
        }
    }
   "Description": "data-source-description",
   "Name": "my-data-source"
}
```

後續步驟

造訪 OpenSearch Dashboards

建立資料來源之後，OpenSearch Service 會為您提供 OpenSearch Dashboards 連結。您可以使用此功能來設定存取控制、定義資料表、安裝out-of-the-box整合，以及查詢您的資料。

如需詳細資訊，請參閱the section called “設定 S3 資料來源”。

映射 AWS Glue Data Catalog 角色

如果您在建立資料來源後已啟用精細存取控制，則必須將非管理員使用者對應至具有 AWS Glue Data Catalog 存取權的 IAM 角色，才能執行直接查詢。若要手動建立可映射至 IAM `glue_access`角色的後端角色，請執行下列步驟：

> **Note**
>
> 索引用於對資料來源的任何查詢。具有特定資料來源請求索引的讀取存取權的使用者可以針對該資料來源讀取所有查詢。具有結果索引讀取存取權的使用者可以針對該資料來源讀取所有查詢的結果。

1. 從 OpenSearch Dashboards 的主選單中，選擇安全性、角色和建立角色。
2. 為角色命名 glue_access。
3. 針對叢集許可，選取 `indices:data/write/bulk*`、`indices:data/read/scroll`、`indices:data/read/scroll/clear`。
4. 針對索引，輸入您要授予使用者角色存取權的下列索引：
   - `.query_execution_request_`*`<name of data source>`*
   - `query_execution_result_`*`<name of data source>`*
   - `.async-query-scheduler`
   - `flint_*`
5. 針對索引許可，選取 `indices_all`。
6. 選擇建立。
7. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
8. 在後端角色下，新增需要呼叫網域許可之角色的 AWS Glue ARN。

   ```
   arn:aws:iam::account-id:role/role-name
   ```
9. 選取映射，並確認已映射使用者下顯示該角色。

如需映射角色的詳細資訊，請參閱 the section called “將角色映射至使用者”。

其他資源

手動建立 IAM 角色的必要許可

為網域建立資料來源時，您可以選擇 IAM 角色來管理對資料的存取。您有兩種選擇：

1. 自動建立新的 IAM 角色
2. 使用您手動建立的現有 IAM 角色

如果您使用手動建立的角色，則需要將正確的許可連接到角色。許可必須允許存取特定資料來源，並允許 OpenSearch Service 擔任該角色。這是必要的，以便 OpenSearch Service 可以安全地存取您的資料並與之互動。

下列範例政策示範建立和管理資料來源所需的最低權限許可。如果您有更廣泛的許可，例如 `s3:*`或 `AdminstratorAccess`政策，這些許可會包含範例政策中的最低權限許可。

在下列範例政策中，將`######`取代為您自己的資訊。

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"HttpActionsForOpenSearchDomain",
         "Effect":"Allow",
         "Action":"es:ESHttp*",
"Resource":"arn:aws:es:us-east-1:111122223333:domain/example.com/*"
      },
      {
         "Sid":"AmazonOpenSearchS3GlueDirectQueryReadAllS3Buckets",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:ListBucket"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceAccount":"111122223333"
            }
```

```
            },
            "Resource":"*"
        },
        {
            "Sid":"AmazonOpenSearchDirectQueryGlueCreateAccess",
            "Effect":"Allow",
            "Action":[
                "glue:CreateDatabase",
                "glue:CreatePartition",
                "glue:CreateTable",
                "glue:BatchCreatePartition"
            ],
            "Resource":"*"
        },
        {
            "Sid":"AmazonOpenSearchS3GlueDirectQueryModifyAllGlueResources",
            "Effect":"Allow",
            "Action":[
                "glue:DeleteDatabase",
                "glue:DeletePartition",
                "glue:DeleteTable",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:GetTable",
                "glue:GetTableVersions",
                "glue:GetTables",
                "glue:UpdateDatabase",
                "glue:UpdatePartition",
                "glue:UpdateTable",
                "glue:BatchGetPartition",
                "glue:BatchDeletePartition",
                "glue:BatchDeleteTable"
            ],
            "Resource":[
                "arn:aws:glue:us-east-1:111122223333:table/*",
                "arn:aws:glue:us-east-1:111122223333:database/*",
                "arn:aws:glue:us-east-1:111122223333:catalog",
                "arn:aws:es:us-east-1:111122223333:domain/domain_name"
            ],
            "Condition":{
                "StringEquals":{
                    "aws:ResourceAccount":"111122223333"
```

```
            }
         }
      },
      {
         "Sid":"ReadAndWriteActionsForS3CheckpointBucket",
         "Effect":"Allow",
         "Action":[
            "s3:ListMultipartUploadParts",
            "s3:DeleteObject",
            "s3:GetObject",
            "s3:PutObject",
            "s3:GetBucketLocation",
            "s3:ListBucket"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceAccount":"111122223333"
            }
         },
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
      }
   ]
}
```

若要支援不同帳戶中的 Amazon S3 儲存貯體，您需要將條件包含在 Amazon S3 政策中，並新增適當的帳戶。

在下列範例條件中，將######取代為您自己的資訊。

```
"Condition": {
               "StringEquals": {
                   "aws:ResourceAccount": "{{accountId}}"
               }
```

角色也必須具有下列信任政策，指定目標 ID。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
       {
          "Effect":"Allow",
          "Principal":{
             "Service": "directquery.opensearchservice.amazonaws.com"
          },
          "Action":"sts:AssumeRole"
       }
     ]
}
```

如需建立角色的指示，請參閱使用自訂信任政策建立角色。

如果您在 OpenSearch Service 中啟用了精細存取控制，系統會自動為您的資料來源建立新的 OpenSearch 精細存取控制角色。新的精細存取控制角色的名稱將是 `AWSOpenSearchDirectQuery <name of data source>`。

根據預設，角色只能存取直接查詢資料來源索引。雖然您可以設定角色來限制或授予對資料來源的存取權，但建議您不要調整此角色的存取權。如果您刪除資料來源，則會刪除此角色。如果任何其他使用者映射到角色，這將移除其存取權。

在 OpenSearch Dashboards 中設定和查詢 S3 資料來源

現在您已建立資料來源，您可以設定安全設定、定義 Amazon S3 資料表或設定加速資料索引。本節會在您查詢資料之前，逐步解說您在 OpenSearch Dashboards 中使用資料來源的各種使用案例。

若要設定下列區段，您必須先導覽至 OpenSearch Dashboards 中的資料來源。在左側導覽的管理下，選擇資料來源。在管理資料來源下，選取您在主控台中建立的資料來源名稱。

使用 Query Workbench 建立 Spark 資料表

從 OpenSearch Service 到 Amazon S3 的直接查詢會使用 中的 Spark 資料表 AWS Glue Data Catalog。您可以從 Query Workbench 內建立資料表，而不必離開 OpenSearch Dashboards。

若要管理資料來源中現有的資料庫和資料表，或建立您要使用直接查詢的新資料表，請從左側導覽中選擇查詢工作台，然後從資料來源下拉式清單中選擇 Amazon S3 資料來源。

若要為以 Parquet 格式存放在 S3 中的 VPC 流程日誌設定資料表，請執行下列查詢：

```
CREATE TABLE
datasourcename.gluedatabasename.vpclogstable (version INT, account_id STRING,
 interface_id STRING,
srcaddr STRING, dstaddr STRING, srcport INT, dstport INT, protocol INT, packets
 BIGINT,
bytes BIGINT, start BIGINT, end BIGINT, action STRING, log_status STRING,
`aws-account-id` STRING, `aws-service` STRING, `aws-region` STRING, year STRING,
month STRING, day STRING, hour STRING)

USING parquet PARTITIONED BY (aws-account-id, aws-service, aws-region, year, month,
day, hour)

LOCATION "s3://accountnum-vpcflow/AWSLogs"
```

建立資料表之後，請執行下列查詢，以確保其與直接查詢相容：

```
MSCK REPAIR TABLE  datasourcename.databasename.vpclogstable
```

熱門 AWS 日誌類型的設定整合

您可以將存放在 Amazon S3 中的 AWS 日誌類型與 OpenSearch Service 整合。使用 OpenSearch Dashboards 安裝建立 AWS Glue Data Catalog 資料表、已儲存查詢和儀表板的整合。這些整合使用索引檢視來保持儀表板更新。

如需安裝整合的指示，請參閱 OpenSearch 文件中的安裝整合資產。

當您選取整合時，請確定其具有 S3 Glue標籤。

當您設定整合時，請為連線類型指定 S3 連線。然後，選取整合的資料來源、資料的 Amazon S3 位置、管理加速索引的檢查點，以及使用案例所需的資產。

> **Note**
>
> 確定檢查點的 S3 儲存貯體具有檢查點位置的寫入許可。如果沒有這些許可，整合的加速將會失敗。

設定存取控制

在資料來源的詳細資訊頁面上，尋找存取控制區段，然後選擇編輯。如果網域已啟用精細存取控制，請選擇受限制，然後選取您要提供新資料來源存取權的角色。只有當您只希望管理員能夠存取資料來源時，也可以選擇管理員。

> **Important**
>
> 索引用於對資料來源的任何查詢。具有特定資料來源請求索引的讀取存取權的使用者可以針對該資料來源讀取所有查詢。具有結果索引讀取存取權的使用者可以針對該資料來源讀取所有查詢的結果。

在 OpenSearch Discover 中查詢 S3 資料

設定資料表並設定所需的選用查詢加速之後，您就可以開始分析資料。若要查詢您的資料，請從下拉式選單中選取您的資料來源。如果您使用的是 Amazon S3 和 OpenSearch Dashboards，請前往探索並選取資料來源名稱。

如果您使用略過的索引或尚未建立索引，您可以使用 SQL 或 PPL 來查詢資料。如果您已設定具體化視觀表或涵蓋索引，表示您已經有 索引，並且可以在儀表板中使用儀表板查詢語言 (DQL)。您也可以使用 PPL 搭配可觀測性外掛程式，以及 SQL 搭配 Query Workbench 外掛程式。目前，只有可觀測性和 Query Workbench 外掛程式支援 PPL 和 SQL。如需使用 OpenSearch Service API 查詢資料，請參閱非同步 API 文件。

> **Note**
>
> 不支援所有 SQL 和 PPL 陳述式、命令和函數。如需支援的命令清單，請參閱 the section called “支援的 SQL 和 PPL 命令”。
> 如果您已建立具體化檢視或涵蓋索引，您可以使用 DQL 來查詢資料，前提是您已在其中編製索引。

疑難排解

在某些情況下，結果可能無法如預期傳回。如果您遇到任何問題，請確定您正在遵循 the section called “建議”。

直接查詢 OpenSearch Service 中的 Amazon CloudWatch Logs 資料

本節將逐步引導您在 Amazon OpenSearch Service 中建立和設定資料來源整合，讓您有效率地查詢和分析儲存在 CloudWatch Logs 中的資料。

在以下頁面中，您將了解如何設定 CloudWatch Logs 直接查詢資料來源、導覽必要的先決條件，以及使用 step-by-step程序 AWS 管理主控台。

主題

- 在 OpenSearch Service 中建立 Amazon CloudWatch Logs 資料來源整合
- 在 OpenSearch Dashboards 中設定和查詢 CloudWatch Logs 資料來源

在 OpenSearch Service 中建立 Amazon CloudWatch Logs 資料來源整合

如果您使用 Amazon OpenSearch Serverless 滿足您的可觀測性需求，您現在可以分析 Amazon CloudWatch Logs，而無需將資料複製或擷取至 OpenSearch Service。此功能利用直接查詢來查詢資料，類似於從 OpenSearch Service 分析 Amazon S3 中的資料。您可以從 AWS 管理主控台建立新的連線資料來源來開始。

您可以建立新的資料來源來分析 CloudWatch Logs 資料，而不必建置 Amazon OpenSearch Serverless，即可直接在 CloudWatch Logs 中查詢操作日誌。這可讓您分析位於 OpenSearch Service 外部的存取操作資料。透過跨 OpenSearch Service 和 CloudWatch Logs 查詢，您可以開始分析 CloudWatch Logs 中的日誌，然後回到 OpenSearch 中的監控資料來源，而無需切換工具。

若要使用此功能，您可以透過 AWS 管理主控台為 OpenSearch Service 建立 CloudWatch Logs 直接查詢資料來源。

主題

- 先決條件
- 程序
- 後續步驟
- 其他資源

先決條件

開始之前，請確定您已檢閱下列文件：

- the section called “Amazon CloudWatch Logs 的限制”
- the section called “CloudWatch Logs 建議”
- the section called “CloudWatch Logs 的配額”

您必須先在 中擁有下列資源，才能建立資料來源 AWS 帳戶：

- 啟用 CloudWatch Logs。設定 CloudWatch Logs 以收集與 OpenSearch 資源 AWS 帳戶 相同的日誌。如需說明，請參閱《Amazon CloudWatch Logs 使用者指南》中的 CloudWatch Logs 入門。Amazon CloudWatch
- 一或多個 CloudWatch 日誌群組。 您可以指定包含要查詢之資料的日誌群組。如需建立日誌群組的指示，請參閱《Amazon CloudWatch Logs 使用者指南》中的在 CloudWatch Logs 中建立日誌群組。 Amazon CloudWatch
- （選用）手動建立的 IAM 角色。 您可以使用此角色來管理對資料來源的存取。或者，您可以讓 OpenSearch Service 自動為您建立具有所需許可的角色。如果您選擇使用手動建立的 IAM 角色，請遵循 中的指引the section called “手動建立 IAM 角色的必要許可”。

程序

您可以使用 設定集合層級查詢資料來源 AWS 管理主控台。

使用 設定集合層級資料來源 AWS 管理主控台

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，前往中央管理並選擇已連線的資料來源。
3. 選擇連線。
4. 選擇 CloudWatch 作為資料來源類型。
5. 選擇下一步。
6. 在資料連線詳細資訊下，輸入名稱和選用的描述。
7. 在 IAM 角色下，選擇如何管理對日誌群組的存取。
    a. 如果您想要自動為此資料來源建立角色，請遵循下列步驟：
        i. 選取建立新角色。
        ii. 輸入 IAM 角色的名稱。
        iii. 選取一或多個日誌群組，以定義可查詢哪些資料。

b. 如果您想要使用自己管理的現有角色，請遵循下列步驟：

i. 選取使用現有角色。

ii. 從下拉式功能表中選取現有角色。

> **Note**
>
> 使用您自己的角色時，您必須從 IAM 主控台連接必要的政策，以確保它具有所有必要的許可。如需詳細資訊，請參閱the section called “手動建立 IAM 角色的必要許可”。

8. （選用）在存取政策下，設定資料來源的存取政策。存取政策控制是否接受或拒絕對 OpenSearch Service 直接查詢資料來源的請求。如果您未設定存取政策，則只有資料來源擁有者可以存取。您可以設定存取政策以啟用跨帳戶存取，允許其他 中的主體 AWS 帳戶 存取資料來源。

   您可以使用視覺化編輯器或提供 JSON 政策文件來建立存取政策。使用視覺化編輯器，您可以透過指定委託人 AWS 帳戶 ID、帳戶 ARN、IAM 使用者 ARN、IAM 角色 ARN、來源 IP 地址或 CIDR 區塊來允許或拒絕存取。視覺化編輯器最多支援 10 個元素。若要定義超過 10 個元素的政策，請使用 JSON 編輯器。

   您也可以選擇匯入政策，從另一個資料來源匯入現有的存取政策。

9. （選用）在標籤下，將標籤新增至資料來源。

10. 選擇下一步。

11. 在設定 OpenSearch 下，選擇如何設定 OpenSearch。

    a. 使用預設設定：

    - 檢閱預設資源名稱和資料保留設定。我們建議您使用自訂名稱。

      當您使用預設設定時，會為您建立新的 OpenSearch 應用程式和 Essentials 工作區，無需額外費用。OpenSearch 可讓您分析多個資料來源。它包含工作區，可為熱門使用案例提供量身打造的體驗。Workspaces 支援存取控制，可讓您為使用案例建立私有空間，並僅與協作者共用。

    b. 使用自訂設定：

    i. 請選擇 Customize (自訂)。

    ii. 視需要編輯集合名稱和資料保留設定。

iii. 選取您要使用的 OpenSearch 應用程式和工作區。

12. 選擇下一步。
13. 如果您需要進行任何變更，請檢閱您的選擇，然後選擇編輯。
14. 選擇連線以設定資料來源。建立資料來源時，請保留在此頁面上。準備就緒後，系統會將您導向至資料來源詳細資訊頁面。

後續步驟

造訪 OpenSearch Dashboards

建立資料來源之後，OpenSearch Service 會為您提供 OpenSearch Dashboards URL。您可以使用此功能來設定存取控制、定義資料表、設定熱門日誌類型的日誌類型型儀表板，以及使用 SQL 或 PPL 查詢資料。

如需詳細資訊，請參閱the section called “設定 CloudWatch Logs 資料來源”。

其他資源

手動建立 IAM 角色的必要許可

建立資料來源時，您可以選擇 IAM 角色來管理對資料的存取。您有兩種選擇：

1. 自動建立新的 IAM 角色
2. 使用您手動建立的現有 IAM 角色

如果您使用手動建立的角色，則需要將正確的許可連接到角色。許可必須允許存取特定資料來源，並允許 OpenSearch Service 擔任該角色。這是必要的，以便 OpenSearch Service 可以安全地存取您的資料並與之互動。

下列範例政策示範建立和管理資料來源所需的最低權限許可。如果您有更廣泛的許可，例如 `logs:*`或 `AdminstratorAccess`政策，這些許可會包含範例政策中最低權限的許可。

在下列範例政策中，將`######`取代為您自己的資訊。

JSON

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

```
    "Statement": [
        {
            "Sid": "AmazonOpenSearchDirectQueryAllLogsAccess",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:StartQuery",
                "logs:GetLogGroupFields"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            },
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:*"
            ]
        }
    ]
}
```

角色也必須具有下列信任政策，指定目標 ID。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "TrustPolicyForAmazonOpenSearchDirectQueryService",
            "Effect": "Allow",
            "Principal": {
                "Service": "directquery.opensearchservice.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:opensearch:us-
east-1:111122223333:datasource/rolename"
                }
            }
```

```
        }
    ]
}
```

如需建立角色的指示，請參閱使用自訂信任政策建立角色。

根據預設，角色只能存取直接查詢資料來源索引。雖然您可以設定角色來限制或授予對資料來源的存取，但建議您不要調整此角色的存取。如果您刪除資料來源，則會刪除此角色。如果任何其他使用者映射到角色，這將移除其存取權。

直接查詢資料來源的範例存取政策

直接查詢資料來源的存取政策遵循 IAM 政策語法。政策文件必須為有效的 JSON 格式。下列範例政策會授予直接查詢資料來源的特定 AWS 帳戶 存取權。

在下列範例政策中，將######取代為您自己的資訊。

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Principal": {
     "AWS": "arn:aws:iam::account-id:root"
     },
     "Action": [
       "opensearch:StartDirectQuery",
       "opensearch:GetDirectQuery",
       "opensearch:CancelDirectQuery",
       "opensearch:GetDirectQueryResult"
     ],
     "Resource": "arn:aws:opensearch:region:account-id:datasource/data-source-name"
   }
 ]
}
```

如果您未設定存取政策，則只有資料來源擁有者可以存取資料來源。

在 OpenSearch Dashboards 中設定和查詢 CloudWatch Logs 資料來源

現在您已建立資料來源，即可開始使用 OpenSearch Dashboards。本節將引導您在 OpenSearch Dashboards 中使用資料來源來了解各種使用案例。

從探索頁面查詢日誌群組

在 OpenSearch 探索頁面中，您可以使用您設定的新直接查詢資料來源來查詢 CloudWatch Logs 日誌群組。若要這樣做，請選擇探索日誌，然後使用搜尋列來使用 SQL 或 PPL 建置查詢。您可以篩選、排序和視覺化日誌群組傳回的資料。若要了解 CloudWatch Logs 整合支援哪些陳述式、命令和限制，請參閱 the section called “支援的 SQL 和 PPL 命令”。

為您的資料來源建立儀表板檢視

當您使用 OpenSearch Service 時，您可以使用預先建置的儀表板範本快速分析熱門 AWS 日誌類型。對於 CloudWatch Logs，有 VPC、CloudTrail、 AWS WAF和 Network Firewall 日誌的範本。這些範本可讓您快速建立專為特定資料量身打造的儀表板。其中包括針對該特定日誌類型量身打造的儀表板。這可讓您快速開始並執行分析這些熱門 AWS 日誌來源，而無需從頭開始建置所有項目。

> **Note**
>
> 儀表板使用索引檢視，該檢視使用直接查詢 OpenSearch Compute Units (OCUs) 以及無伺服器集合indexingOCUs、searchingOCUs，從 CloudWatch Logs 擷取資料。

請依照下列步驟，使用其中一個預先建置的範本來建立儀表板，讓您可以立即開始探索和分析資料。

建立儀表板檢視

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 從左側導覽窗格中，選擇中央管理，然後選擇已連線的資料來源。
3. 選取資料來源以開啟詳細資訊頁面。
4. 選擇 Create dashboard (建立儀表板)。
5. 選擇您要建立的儀表板類型。
6. 輸入儀表板的名稱。
7. 輸入儀表板的選用描述。
8. 選取要在儀表板上檢視的一或多個日誌群組。
9. 選擇您要重新整理儀表板中資料的頻率。
10. 選擇您要使用的 OpenSearch 工作區。
    a. 若要建立新的工作區，請選取建立新工作區並輸入名稱。
    b. 若要使用現有的工作區，請選取選取現有的工作區。

11. 選擇 Create dashboard (建立儀表板)。

在 OpenSearch Discover 中查詢 CloudWatch Logs 資料

若要查詢資料，請從下拉式功能表中選取資料來源。如果您使用的是 CloudWatch Logs，請從 Essentials 工作區導覽至探索，並開始使用 OpenSearch SQL 或 Piped Processing Language (PPL) 查詢資料。如需支援的命令清單，請參閱 the section called “支援的 SQL 和 PPL 命令”。

> **Note**
>
> 如果您已建立具體化檢視，您可以使用 DQL 來查詢資料，前提是您已在其中編製索引。

疑難排解

在某些情況下，結果可能無法如預期傳回。如果您遇到任何問題，請確定您正在遵循 the section called “建議”。

直接在 OpenSearch Service 中查詢 Amazon Security Lake 資料

本節將逐步引導您在 Amazon OpenSearch Service 中建立和設定資料來源整合的程序，讓您有效率地查詢和分析存放在 Security Lake 中的資料。

在以下頁面中，您將了解如何設定 Security Lake 直接查詢資料來源、導覽必要的先決條件，以及使用 step-by-step程序 AWS 管理主控台。

主題

- 在 OpenSearch Service 中建立 Amazon Security Lake 資料來源整合
- 在 OpenSearch Dashboards 中設定和查詢 Security Lake 資料來源

在 OpenSearch Service 中建立 Amazon Security Lake 資料來源整合

您可以使用 Amazon OpenSearch Serverless 直接查詢 Amazon Security Lake 中的安全資料。若要這樣做，您可以建立資料來源，讓您能夠在 Security Lake 資料上使用 OpenSearch 零 ETL 功能。當您建立資料來源時，您可以直接搜尋、取得洞見和分析存放在 Security Lake 中的資料。您可以加速查詢效能，並在使用隨需索引的特定 Security Lake 資料集上使用進階 OpenSearch 分析。

主題

- 先決條件
- 程序
- 後續步驟
- 其他資源

先決條件

開始之前，請確定您已檢閱下列文件：

- the section called “Amazon Security Lake 的限制”
- the section called “Security Lake 建議”
- the section called “Security Lake 的配額”

在建立資料來源之前，請在 Security Lake 中採取下列動作：

- 啟用 Security Lake。設定 Security Lake 在與 OpenSearch 資源 AWS 區域 相同的 上收集日誌。如需說明，請參閱《Amazon Security Lake 使用者指南》中的 Amazon Security Lake 入門。
- 設定 Security Lake 許可。請確定您已接受資源管理的服務連結角色許可，而且主控台不會在問題頁面下顯示任何問題。如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的 Security Lake 的服務連結角色。
- 共用 Security Lake 資料來源。在與 Security Lake 相同的帳戶中存取 OpenSearch 時，請確保 Security Lake 主控台中沒有向 Lake Formation 註冊 Security Lake 儲存貯體的訊息。對於跨帳戶 OpenSearch 存取，請在 Security Lake 主控台中設定 Lake Formation 查詢訂閱者。使用與您的 OpenSearch 資源相關聯的帳戶作為訂閱者。如需詳細資訊，請參閱《Amazon Security Lake 使用者指南》中的 Security Lake 中的訂閱者管理。

此外，您還必須在 中擁有下列資源 AWS 帳戶：

- （選用）手動建立的 IAM 角色。您可以使用此角色來管理對資料來源的存取。或者，您可以讓 OpenSearch Service 自動為您建立具有所需許可的角色。如果您選擇使用手動建立的 IAM 角色，請遵循 中的指引the section called “手動建立 IAM 角色的必要許可”。

程序

您可以從 內設定資料來源以與 Security Lake 資料庫連線 AWS 管理主控台。

使用 設定資料來源 AWS 管理主控台

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，前往中央管理並選擇已連線的資料來源。
3. 選擇連線。
4. 選擇 Security Lake 作為資料來源類型。
5. 選擇下一步。
6. 在資料連線詳細資訊下，輸入名稱和選用描述。
7. 在 IAM 許可存取設定下，選擇如何管理對資料來源的存取。

   a. 如果您想要自動為此資料來源建立角色，請遵循下列步驟：

      i. 選取建立新角色。

      ii. 輸入 IAM 角色的名稱。

      iii. 選取一或多個 AWS Glue 資料表，以定義可查詢哪些資料。

   b. 如果您想要使用自己管理的現有角色，請遵循下列步驟：

      i. 選取使用現有角色。

      ii. 從下拉式功能表中選取現有角色。

   > **Note**
   >
   > 使用您自己的角色時，您必須從 IAM 主控台連接必要的政策，以確保它具有所有必要的許可。如需詳細資訊，請參閱the section called “手動建立 IAM 角色的必要許可”。

8. （選用）在標籤下，將標籤新增至資料來源。
9. 選擇下一步。
10. 在設定 OpenSearch 下，選擇如何設定 OpenSearch。

    - 檢閱預設資源名稱和資料保留設定。

      當您使用預設設定時，會為您建立新的 OpenSearch 應用程式和 Essentials 工作區，無需額外費用。OpenSearch 可讓您分析多個資料來源。它包含工作區，可為熱門使用案例提供量身打造的體驗。Workspaces 支援存取控制，可讓您為使用案例建立私有空間，並僅與協作者共用。

11. 使用自訂設定：

a. 請選擇 Customize (自訂)。
b. 視需要編輯集合名稱和資料保留設定。
c. 選取您要使用的 OpenSearch 應用程式和工作區。

12. 選擇下一步。
13. 如果您需要進行任何變更，請檢閱您的選擇，然後選擇編輯。
14. 選擇連線以設定資料來源。建立資料來源時，請保留在此頁面上。準備就緒後，系統會將您導向至資料來源詳細資訊頁面。

後續步驟

造訪 OpenSearch Dashboards 並建立儀表板

建立資料來源之後，OpenSearch Service 會為您提供 OpenSearch Dashboards URL。您可以使用它來查詢使用 SQL 或 PPL 的資料。Security Lake 整合隨附 SQL 和 PPL 的預先封裝查詢範本，讓您開始分析日誌。

如需詳細資訊，請參閱the section called “設定 Security Lake 資料來源”。

其他資源

手動建立 IAM 角色的必要許可

建立資料來源時，您可以選擇 IAM 角色來管理對資料的存取。您有兩種選擇：

1. 自動建立新的 IAM 角色
2. 使用您手動建立的現有 IAM 角色

如果您使用手動建立的角色，則需要將正確的許可連接到角色。許可必須允許存取特定資料來源，並允許 OpenSearch Service 擔任該角色，以便 OpenSearch Service 可以安全地存取您的資料並與之互動。此外，針對您要查詢的任何資料庫和資料表，將 LakeFormation 許可授予角色。將`DESCRIBE`許可授予您要從直接查詢連線查詢的 SecurityLake 資料庫上的角色。授予資料庫內資料表之資料來源角色的至少`SELECT and DESCRIBE`許可。

下列範例政策示範建立和管理資料來源所需的最低權限許可。如果您有更廣泛的許可，例如`AdminstratorAccess`政策，這些許可會包含範例政策中最低權限的許可。

在下列範例政策中，將`######`取代為您自己的資訊。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AmazonOpenSearchDirectQueryServerlessAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll",
                "aoss:DashboardsAccessAll"
            ],
            "Resource": "arn:aws:aoss:us-
east-1:111122223333:collection/collectionname/*"
        },
        {
            "Sid": "AmazonOpenSearchDirectQueryGlueAccess",
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:GetTable",
                "glue:GetTableVersions",
                "glue:GetTables",
                "glue:SearchTables",
                "glue:BatchGetPartition"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:111122223333:table/databasename/*",
                "arn:aws:glue:us-east-1:111122223333:database/databasename",
                "arn:aws:glue:us-east-1:111122223333:catalog",
                "arn:aws:glue:us-east-1:111122223333:database/default"
            ]
        },
        {
            "Sid": "AmazonOpenSearchDirectQueryLakeFormationAccess",
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess"
            ],
            "Resource": [
```

```
                "*"
            ]
        }
    ]
}
```

角色也必須具有下列信任政策，指定目標 ID。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "directquery.opensearchservice.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

如需建立角色的指示，請參閱使用自訂信任政策建立角色。

根據預設，角色只能存取直接查詢資料來源索引。雖然您可以設定角色來限制或授予對資料來源的存取權，但建議您不要調整此角色的存取權。如果您刪除資料來源，則會刪除此角色。如果任何其他使用者映射到角色，這將移除其存取權。

查詢使用客戶受管金鑰加密的 Security Lake 資料

如果與資料連線相關聯的 Security Lake 儲存貯體使用伺服器端加密與客戶受管的 進行加密 AWS KMS key，您必須將 LakeFormation 服務角色新增至金鑰政策。這可讓服務存取和讀取查詢的資料。

在下列範例政策中，將######取代為您自己的資訊。

```
{
    "Sid": "Allow LakeFormation to access the key",
    "Effect": "Allow",
    "Principal": {
```

```
        "AWS": "arn:aws:iam::account:role/aws-service-role/lakeformation.amazonaws.com/
AWSServiceRoleForLakeFormationDataAccess"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

在 OpenSearch Dashboards 中設定和查詢 Security Lake 資料來源

現在您已建立資料來源，您可以在 OpenSearch Dashboards 中進行設定。

本節會在您查詢資料之前，逐步解說您在 OpenSearch Dashboards 中使用資料來源的各種使用案例。若要開始使用，您需要導覽至 OpenSearch Dashboards 中的資料來源。在左側選單的管理下，選擇資料來源。然後，選取您先前在 OpenSearch Service 主控台中建立的資料來源名稱。

從 Discover 查詢 Security Lake 資料表

如果您已根據 Security Lake 日誌建立資料表，您現在可以直接從 OpenSearch Discover 查詢這些資料表。這可讓您直接從熟悉的探索界面無縫存取和分析存放在 Security Lake 中的資料。透過直接從 Discover 查詢 Security Lake，您可以避免手動擷取、轉換資料，並將資料載入個別的搜尋索引。為了快速開始分析日誌，Discover 包含一組 PPL 和 SQL 儲存的查詢。

首先，選取您設定的資料來源。選取您要查詢的關聯資料庫和資料表，然後使用搜尋列針對您的資料表撰寫查詢。若要了解 Security Lake 整合支援哪些陳述式、命令和限制，請參閱 the section called “支援的 SQL 和 PPL 命令”。

若要利用 Security Lake 可用的預先建置查詢，請前往 Discover 右上角的 ...，選擇開啟查詢，然後選擇範本。Security Lake 支援的日誌來源有許多預先建置的查詢。搜尋符合您使用案例的範本、複製要在搜尋列中使用的查詢，並以您自己的資訊取代範本欄位（例如區域和動作）。

從探索加速資料

若要在 OpenSearch 中增強效能並啟用更快的後續查詢和分析，您可以將查詢的結果從探索擷取到 OpenSearch 索引檢視中。

建立索引檢視

1. 在探索中，選擇建立索引化檢視。
2. 在查詢編輯器中，輸入所需的查詢。您可以在這裡建立新的查詢，或使用先前搜尋的現有查詢。
3. 為您的新索引檢視指定名稱。選擇可協助您稍後識別檢視的描述性名稱。
4. 設定索引檢視的資料保留設定。您可以指定資料應該保留在索引中多久，讓您平衡效能與儲存成本。
5. 建立索引檢視。建立之後，您的索引檢視將可用於更快速的查詢和分析。

如果您先前已建立索引檢視，您可以從探索存取它們。

使用現有的索引檢視

1. 從探索中，選擇選取索引檢視以查看 Security Lake 現有索引檢視的清單。
2. 選擇您要使用的索引檢視。這會將檢視套用至您目前的查詢，進而大幅加快資料擷取和分析的速度。

為您的資料來源建立儀表板檢視

使用 OpenSearch Service 時，您可以使用預先建置的儀表板範本來分析熱門 AWS 日誌類型。對於 Security Lake，有 VPC、CloudTrail 和 WAF 日誌的範本。這些範本可讓您建立專為特定資料量身打造的儀表板。其中包括針對該特定日誌類型量身打造的預先建置查詢和儀表板。這可讓您快速開始並執行分析這些熱門 AWS 日誌來源，而無需從頭開始建置所有項目。

> **Note**
>
> 儀表板使用索引檢視，從 Security Lake 擷取資料並有助於直接查詢和收集運算。

請依照下列步驟，使用其中一個預先建置的範本來建立儀表板，讓您可以立即開始探索和分析資料。

建立儀表板檢視

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 從左側導覽窗格中，選擇中央管理，然後選擇已連線的資料來源。
3. 選取資料來源以開啟詳細資訊頁面。
4. 選擇 Create dashboard (建立儀表板)。

5. 選擇您要建立的儀表板類型。
6. 輸入儀表板的名稱。
7. 輸入儀表板的選用描述。
8. 選取要在儀表板上檢視的一或多個 AWS Glue 資料表。
9. 選擇您要重新整理儀表板中資料的頻率。
10. 選擇您要使用的 OpenSearch 工作區。
    a. 若要建立新工作區，請選取建立新工作區。
    b. 若要使用現有的工作區，請選取選取現有的工作區。
11. 輸入工作區的名稱。
12. 選擇 Create dashboard (建立儀表板)。

疑難排解

在某些情況下，結果可能無法如預期傳回。如果您遇到任何問題，請確定您正在遵循 the section called "建議"。

在 Amazon OpenSearch Service 中管理資料來源

管理您的資料來源是維護直接查詢資料來源和其他 AWS 解決方案的可靠性、可用性和效能的重要部分。 AWS 提供下列工具來監控、報告錯誤，並適時採取自動動作。

主題

- 使用 CloudWatch 指標資料來源進行監控
- 啟用和停用資料來源
- 使用 AWS Budget 監控
- 刪除資料來源

使用 CloudWatch 指標資料來源進行監控

您可以使用 CloudWatch 監控直接查詢。CloudWatch 可收集原始資料，將這些資料轉換為可讀取且幾近即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。

您也可以設定警示來監控特定閾值，並在達到這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱什麼是 Amazon CloudWatch。

Amazon S3 報告下列指標：

指標	Description
AsyncQueryCreateAPI	對 API 提出以建立非同步查詢的請求總數。 相關統計資料：平均數、上限、總和 維度：ClientId、DomainName 頻率：60 秒
AsyncQueryGetApiRequestCount	對 API 發出的請求總數，用於擷取非同步查詢結果。 相關統計資料：平均數、上限、總和 維度：ClientId、 DomainName 頻率：60 秒
AsyncQueryCancelApiRequestCount	對 API 發出的取消非同步查詢請求總數。 相關統計資料：平均數、上限、總和 維度：ClientId、 DomainName 頻率：60 秒
AsyncQueryGetApiFailedRequestCusErrCount	由於客戶相關錯誤 （例如無效的查詢 ID) 擷取非同步查詢結果時失敗的請求數量。 相關統計資料：平均數、上限、總和 維度：ClientId、 DomainName 頻率：60 秒
AsyncQueryCancelApiFailedRequestCusErrCount	由於客戶相關錯誤 （例如無效的查詢 ID) 擷取非同步查詢結果時失敗的請求數量。

指標	Description
	相關統計資料：平均數、上限、總和 維度：ClientId、DomainName 頻率：60 秒
AsyncQueryCancelApiFailedRequestSysErrCount	由於客戶相關錯誤建立非同步查詢時失敗的請求數量。 相關統計資料：平均數、上限、總和 維度：ClientId、 DomainName 頻率：60 秒
AsyncQueryGetApiFailedRequestSysErrCount	由於系統相關錯誤，擷取非同步查詢結果時失敗的請求數目。 相關統計資料：平均數、上限、總和 維度：ClientId、 DomainName 頻率：60 秒

CloudWatch Logs 和 Security Lake 會報告下列指標：

指標	Description
DirectQueryRate	針對資料來源提出的請求率。 相關統計資料：總和、最大值、最小值、平均值 維度：DataSourceName 頻率：60 秒
DirectQueryLatency	在資料來源上執行查詢時觀察到的延遲。 相關統計資料：Average、P90, P99、Sum、Minimum、Maximum

指標	Description
	維度：DataSourceName 頻率：60 秒
FailedDirectQueries	在資料來源查詢上觀察到的查詢失敗總數。 相關統計資料：總和、最大值、最小值、平均值 維度：DataSourceName 頻率：60 秒
DirectQueryConsumedOCU	用於在資料來源上執行查詢的 OCUs 數量。 相關統計資料：Average、P90, P99、Sum、Minimum、Maximum 維度：DataSourceName 頻率：60 秒

啟用和停用資料來源

> **Note**
>
> 以下資訊僅適用於 Amazon S3 資料來源。

對於您想要停止資料來源的直接查詢用量的情況，您可以選擇停用資料來源。停用資料來源將完成執行現有的查詢，並停止執行所有新的查詢。

加速設定以提高查詢效能，例如略過索引、具體化視觀表、涵蓋索引等，一旦資料來源停用，就會設定為手動。一旦資料來源在停用後設定為作用中，使用者查詢將如預期般執行。先前設定並設為手動的加速，將需要手動設定為再次依排程執行。

使用 AWS Budget 監控

Amazon OpenSearch Service 正在將帳戶層級的 OCU 用量資料填入 Billing and Cost Management 的 Cost Explorer。您可以在帳戶層級考慮 OCU 用量，並在超過閾值時設定閾值和提醒。

在 Cost Explorer 中篩選的用量類型格式看起來像 RegionCode-DirectQueryOCU (OCU 小時）。如果您想要在 DirectQueryOCU (OCU-Hours) 用量達到閾值時收到通知，您可以建立 AWS Budgets 帳戶，並根據您設定的閾值設定提醒。或者，對於 Amazon S3，您可以設定 Amazon SNS 主題，以便在符合閾值條件時關閉資料來源。

> **Note**
>
> AWS Budgets 中的用量資料並非即時資料，最多可延遲 8 小時。

刪除資料來源

當您刪除資料來源時，Amazon OpenSearch Service 會從網域或集合中移除該資料來源。OpenSearch Service 也會移除與資料來源相關聯的索引。您的交易資料不會從其他資料中刪除 AWS 服務，但其他 AWS 服務 資料不會將新資料傳送至 OpenSearch Service。

您可以使用 AWS 管理主控台 或 OpenSearch Service API 刪除資料來源整合。

AWS 管理主控台

刪除 Amazon S3 資料來源

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 從左側導覽窗格中，選擇網域。
3. 選取您要刪除資料來源的網域。這會開啟網域詳細資訊頁面。選擇一般資訊下方的連線索引標籤，然後尋找直接查詢區段。
4. 選取您要刪除的資料來源，選擇刪除並確認刪除。

刪除 CloudWatch Logs 或 Security Lake 資料來源

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 從左側導覽窗格中，選擇中央管理，然後選擇已連線的資料來源。
3. 選取您要刪除的資料來源，選擇刪除並確認刪除。

OpenSearch Service API

若要刪除 Amazon S3 資料來源，請使用 DeleteDataSource API 操作。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource/data-source-name
```

若要刪除 CloudWatch Logs 或 Security Lake 資料來源，請使用 DeleteDirectQueryDataSource API 操作。

最佳化 Amazon OpenSearch Service 資料來源的查詢效能

當您存取外部資料來源時，Amazon OpenSearch Service 中的查詢效能可能會變慢。這可能是因為網路延遲、資料轉換或大型資料磁碟區等因素所造成。若要改善效能，請考慮根據使用案例建立選取資料量的索引：

- 加速 Amazon S3 上的直接查詢（略過索引）
- 在 和 Security Lake 上建置儀表板視覺化（材料化檢視）
- 使用索引檢視擷取查詢結果，以進行離線檢閱或改善 Security Lake 的效能（材料化檢視）

如需加速查詢的完整文件，包括範例查詢，請參閱開放原始碼文件中使用 OpenSearch 索引最佳化查詢效能。

主題

- 略過索引
- 具體化視觀表
- 涵蓋索引

略過索引

略過索引只會擷取存放在 Amazon S3 中的資料的中繼資料。當您查詢具有略過索引的資料表時，查詢規劃器會使用索引來重寫查詢，有效率地識別資料的位置，而無需掃描所有分割區和檔案。此方法有助於縮小儲存資料的確切位置。

建立略過索引的方式有兩種。第一個方法是從資料來源詳細資訊內自動產生略過的索引。第二個是使用 Query Workbench，使用 SQL 陳述式手動建立略過索引。

若要從資料來源自動產生略過索引，請前往儀表板管理和加速資料，然後選取資料庫和資料表（您可能需要重新整理以取得最新的資料庫和資料表）。然後，您可以選擇產生以自動產生略過索引，或手動選取您要編製索引的每個欄位，並指定加速（略過索引類型）。最後，選擇建立加速，以建立填入新略過索引的重複任務。

只有 Amazon S3 資料來源支援略過索引。

如需使用 Query Workbench 設定略過索引的詳細資訊，請參閱 OpenSearch 文件中的略過索引。

具體化視觀表

具體化視觀表使用複雜的查詢，例如彙總，以支援 OpenSearch Dashboards 視覺化。它們會根據查詢擷取您的資料子集，並將其存放在 OpenSearch 索引中。然後，您可以使用此索引來建立視覺化效果。

Amazon S3 和 Security Lake 資料來源支援具體化檢視。

如需使用 Query Workbench 設定具體化視觀表的詳細資訊，請參閱 OpenSearch 文件中的具體化視觀表。

涵蓋索引

涵蓋索引從資料表中指定的資料欄擷取資料，而 OpenSearch 會根據此資料建立新的索引。您可以使用此新索引進行視覺化和其他 OpenSearch 功能，例如異常偵測或地理空間分析。

僅 Amazon S3 資料來源支援覆蓋索引。

如需設定涵蓋索引的詳細資訊，請參閱 OpenSearch 文件中的涵蓋索引。

支援的 SQL 和 PPL 命令

OpenSearch SQL 和 OpenSearch 管道處理語言 (PPL) 是用於在 OpenSearch、CloudWatch Logs Insights 和 Security Lake 中查詢、分析和處理資料的語言。您可以在 OpenSearch Discover 中使用 OpenSearch SQL 和 OpenSearch PPL 來查詢 CloudWatch Logs、Amazon S3 或 Security Lake 中的資料。除了 Logs Insights QL 之外，CloudWatch Logs Insights Insights 也同時支援 OpenSearch PPL 和 OpenSearch SQL 查詢語言，這是用於分析 CloudWatch Logs 的專用查詢語言。

- OpenSearch SQL：如果您習慣使用關聯式資料庫，OpenSearch SQL 會提供熟悉的選項。OpenSearch SQL 提供 SQL 功能的子集，使其成為執行臨機操作查詢和資料分析任務的理想選擇。使用 OpenSearch SQL，您可以使用 SELECT、 FROM、WHERE、GROUP BY、HAVING 等命令，以及 SQL 中可用的各種其他 SQL 命令和函數。您可以跨資料表（或日誌群組）執行

JOINs、使用子查詢關聯資料表（或日誌群組）的資料，並使用一組豐富的 JSON、數學、字串、條件式和其他 SQL 函數，對日誌和安全資料執行強大的分析。

- OpenSearch PPL (Piped Processing Language)：使用 OpenSearch PPL，您可以使用管道互連命令擷取、查詢和分析資料，讓您更輕鬆地了解和編寫複雜的查詢。其語法是以 Unix 管道為基礎，並啟用命令鏈結來轉換和處理資料。使用 PPL，您可以篩選和彙總資料，並使用 JOINs、子查詢、lookUP 等命令，以及一組豐富的數學、字串、日期、條件式和其他函數進行分析。

雖然 OpenSearch PPL 和 OpenSearch SQL 查詢語言中的大多數命令在 CloudWatch Logs 和 OpenSearch 中都很常見，但其中每個服務都支援一組命令和函數。如需詳細資訊，請參閱以下頁面上的資料表。

- the section called “支援的 SQL 命令”
  - the section called “CloudWatch Logs 洞察”
  - the section called “一般 SQL 限制”
- the section called “支援的 PPL 命令”
  - the section called “使用 OpenSearch PPL 的 CloudWatch Logs Insights 使用者其他資訊”

支援的 OpenSearch SQL 命令和函數

下列參考表顯示 OpenSearch Discover 中支援哪些 SQL 命令來查詢 Amazon S3、Security Lake 或 CloudWatch Logs 中的資料，以及 CloudWatch Logs Insights 中支援哪些 SQL 命令。CloudWatch Logs Insights 和 OpenSearch Discover 中支援用於查詢 CloudWatch Logs 的 SQL 語法相同，並在下表中參考為 CloudWatch Logs。

> **Note**
>
> OpenSearch 也支援 SQL 查詢在 OpenSearch 中擷取並存放在索引中的資料。此 SQL 方言與直接查詢中使用的 SQL 不同，在索引上稱為 OpenSearch SQL。

主題

- 命令
- 函數
- 一般 SQL 限制
- 使用 OpenSearch SQL 的 CloudWatch Logs Insights 使用者其他資訊

命令

> **Note**
>
> 在範例命令欄中，根據您查詢的資料來源*<tableName/logGroup>*，視需要取代。
>
> - 範例命令：`SELECT Body , Operation FROM` **`<tableName/logGroup>`**
> - 如果您要查詢 Amazon S3 或 Security Lake，請使用：`SELECT Body , Operation FROM` **`table_name`**
> - 如果您要查詢 CloudWatch Logs，請使用：`SELECT Body , Operation FROM` **`` `LogGroupA` ``**

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “SELECT 子句”	顯示投影值。	支援的	支援的	支援的	`SELECT method, status FROM <tableName/logGroup>`
the section called “WHERE 子句”	根據提供的欄位條件篩選日誌事件。	支援的	支援的	支援的	`SELECT * FROM <tableName/logGroup> WHERE status = 100`
the section called “GROUP BY 子句”	根據類別對日誌事件進行分組，並根據統計資料	支援的	支援的	支援的	`SELECT method, status, COUNT(*) AS request_count, SUM(bytes) AS total_bytes FROM <tableName/logGroup> GROUP BY method,`

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
	尋找平均值。				`status`
the section called “HAVING 子句”	根據分組條件篩選結果。	支援的	支援的	支援的	`SELECT method, status, COUNT(*) AS request_count, SUM(bytes) AS total_bytes FROM <tableName/logGroup> GROUP BY method, status HAVING COUNT(*) > 5`
the section called “ORDER BY 子句”	根據訂單子句中的欄位來排序結果。您可以依遞減或遞增順序排序。	支援的	支援的	支援的	`SELECT * FROM <tableName/logGroup> ORDER BY status DESC`

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令

the section called “JOIN 子句”
 	
(INNER | CROSS | LEFT OUTER)
	根據常見欄位聯結兩個資料表的結果。	支援（必須使用 Inner和 Left Outer關鍵字才能聯結；SELECT陳述式中僅支援一個 JOIN 操作）	支援（必須使用內部、左側外部和跨關鍵字才能聯結）	支援（必須使用內部、左側外部和跨關鍵字才能聯結）	`SELECT`
					`  A.Body,`
					`  B.Timestamp`
					`FROM`
					`  <tableNameA/logGroupA>  AS A`
					`INNER JOIN`
					`  <tableNameB/logGroupB>  AS B`
					`  ON A.`requestId` = B.`requestId``

the section called “LIMIT 子句”	將結果限制為前 N 列。	支援的	支援的	支援的	`SELECT`
					`  *`
					`FROM`
					`  <tableName/logGroup>`
					`LIMIT`
					`  10`

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令

the section called “CASE 子句”	評估條件，並在符合第一個條件時傳回值。	支援的	支援的	支援的	SELECT
					    method,
					    status,
					    CASE
					        WHEN status BETWEEN 100 AND 199 THEN 'Informational'
					        WHEN status BETWEEN 200 AND 299 THEN 'Success'
					        WHEN status BETWEEN 300 AND 399 THEN 'Redirection'
					        WHEN status BETWEEN 400 AND 499 THEN 'Client Error'
					        WHEN status BETWEEN 500 AND 599 THEN 'Server Error'
					        ELSE 'Unknown Status'
					    END AS status_category,
					    CASE method
					        WHEN 'GET' THEN 'Read Operation'
					        WHEN 'POST' THEN 'Create Operation'
					        WHEN 'PUT' THEN 'Update Operation'
					        WHEN 'PATCH' THEN 'Partial Update Operation'
					        WHEN 'DELETE' THEN 'Delete Operation'
					        ELSE 'Other Operation'
					    END AS operation_type,
					    bytes,
					    datetime
					FROM *<tableName/logGroup>*

命令	Descripti on	CloudWa h Logs	Amazon S3	Security Lake	範例 命令

the section called “常見資料表表達式”	在 SELECT SERT、I TE、DEI 或 MERGE 陳述式中建立具名暫時結果集。	不支援	支援的	支援的	`WITH RequestStats AS (`
					`  SELECT`
					`  method,`
					`  status,`
					`  bytes,`
					`  COUNT(*) AS request_count`
					`  FROM`
					`  tableName`
					`  GROUP BY`
					`  method,`
					`  status,`
					`  bytes`
					`)`
					`SELECT`
					`  method,`
					`  status,`
					`  bytes,`
					`  request_count`
					`FROM`
					`  RequestStats`
					`WHERE`
					`  bytes > 1000`

the section called “EXPLAIN	顯示 SQL 陳述式的執行計畫，而不實際執行。	不支援	支援的	支援的	`EXPLAIN`
					`SELECT`
					`  k,`
					`  SUM(v)`
					`FROM`
					`  VALUES`
					`  (1, 2),`
					`  (1, 3) AS t(k, v)`
					`GROUP BY`
					`  k`

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “LATERAL SUBQUERY 子句”	允許 FROM 子句中的子查詢參考相同 FROM 子句中先前項目的資料欄。	不支援	支援的	支援的	`SELECT * FROM tableName LATERAL ( SELECT * FROM t2 WHERE t1.c1 = t2.c1 )`
the section called “LATERAL VIEW 子句”	將資料表產生函數套用至基底資料表的每一列，以產生虛擬資料表。	不支援	支援的	支援的	`SELECT * FROM tableName LATERAL VIEW EXPLODE(ARRAY(30, 60)) tableName AS c_age LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age`
the section called “LIKE 述詞”	使用萬用字元比對字串與模式。	支援的	支援的	支援的	`SELECT method, status, request, host FROM <tableName/logGroup> WHERE method LIKE 'D%'`

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “OFFSET”	指定開始從查詢傳回資料列之前要略過的資料列數。	與查詢中的 LIMIT 子句搭配使用時支援。例如： • 支援：SELECT * FROM Table LIMIT 100 OFFSET 10 • 不支援：SELECT * FROM Table OFFSET 10	支援的	支援的	SELECT method, status, bytes, datetime FROM <tableName/logGroup> ORDER BY datetime OFFSET 10

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “PIVOT 子句”	將資料列轉換為資料欄，將資料從資料列型格式輪換為資料欄型格式。	不支援	支援的	支援的	`SELECT * FROM ( SELECT method, status, bytes FROM <tableName/logGroup> ) AS SourceTable PIVOT ( SUM(bytes) FOR method IN ('GET', 'POST', 'PATCH', 'PUT', 'DELETE') ) AS PivotTable`
the section called “設定運算子”	結合兩個或多個 SELECT 陳述式的結果（例如 UNION、ERSECT CEPT）。	支援的	支援的	支援的	`SELECT method, status, bytes FROM <tableName/logGroup> WHERE status = '416' UNION SELECT method, status, bytes FROM <tableName/logGroup> WHERE bytes > 20000`

命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “SORT BY 子句”	指定傳回查詢結果的順序。	支援的	支援的	支援的	`SELECT method, status, bytes FROM <tableName/logGroup> SORT BY bytes DESC`
the section called “UNPIVOT	將資料欄轉換為資料列，將資料從資料欄型格式輸換為資料列型格式。	不支援	支援的	支援的	`SELECT status, REPLACE(method, '_bytes', '') AS request_method, bytes, datetime FROM PivotedData UNPIVOT ( bytes FOR method IN ( GET_bytes, POST_bytes, PATCH_bytes, PUT_bytes, DELETE_bytes ) ) AS UnpivotedData`

函數

> **Note**
>
> 在範例命令欄中，根據您查詢的資料來源*<tableName/logGroup>*，視需要取代 。
>
> - 範例命令： `SELECT Body , Operation FROM` **`<tableName/logGroup>`**

- 如果您要查詢 Amazon S3 或 Security Lake，請使用： `SELECT Body , Operation FROM table_name`
- 如果您要查詢 CloudWatch Logs，請使用： `SELECT Body , Operation FROM` **`` `LogGroupA` ``**

可用的 SQL Gramm	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “字串函數”	內建函數，可以操作和轉換 SQL 查詢中的字串和文字資料。例如，轉換案例、合併字串、擷取部分和清理文字。	支援的	支援的	支援的	`SELECT UPPER(method) AS upper_method, LOWER(host) AS lower_host FROM <tableName/logGroup>`
the section called “日期和時間函數”	用於處理和轉換查詢中日期和時間戳記資料的內建函數。例如，date_add、date_sub、datediff 和 current_date。	支援的	支援的	支援的	`SELECT TO_TIMESTAMP(datetime) AS timestamp, TIMESTAMP_SECONDS(UNIX_TIMESTAMP(datetime)) AS from_seconds, UNIX_TIMESTAMP(datetime) AS to_unix, FROM_UTC_TIMESTAMP(datetime, 'PST') AS to_pst, TO_UTC_TIMESTAMP(datetime, 'EST') AS from_est FROM <tableName/logGroup>`

可用的 SQL Gramm	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “彙總函數”	對多個資料列執行計算以產生單一摘要值的內建函數。例如，總和、計數、平均值、最大值和最小值。	支援的	支援的	支援的	`SELECT COUNT(*) AS total_records, COUNT(DISTINCT method) AS unique_methods, SUM(bytes) AS total_bytes, AVG(bytes) AS avg_bytes, MIN(bytes) AS min_bytes, MAX(bytes) AS max_bytes FROM <tableName/logGroup>`
the section called “條件函數”	內建函數，可根據指定的條件執行動作，或以條件方式評估表達式。例如，CASE 和 IF。	支援的	支援的	支援的	`SELECT CASE WHEN method = 'GET' AND bytes < 1000 THEN 'Small Read' WHEN method = 'POST' AND bytes > 10000 THEN 'Large Write' WHEN status >= 400 OR bytes = 0 THEN 'Problem' ELSE 'Normal' END AS request_type FROM <tableName/logGroup>`

可用的 SQL Gramm	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “JSON 函數”	用於在 SQL 查詢中剖析、擷取、修改和查詢 JSON 格式資料的內建函數（例如，from_json、to_json、get_json_object、json_tuple），允許在資料集中操作 JSON 結構。	支援的	支援的	支援的	`SELECT FROM_JSON( @message, 'STRUCT< host: STRING, user-identifier: STRING, datetime: STRING, method: STRING, status: INT, bytes: INT >' ) AS parsed_json FROM <tableName/logGroup>`
the section called “陣列函數”	用於在 SQL 查詢中使用陣列類型資料欄的內建函數，允許存取、修改和分析陣列資料（例如，大小、爆炸、Array_contains) 等操作。	支援的	支援的	支援的	`SELECT scores, size(scores) AS length, array_contains(scores, 90) AS has_90 FROM <tableName/logGroup>`

可用的 SQL Gramm	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “範圍函數”	內建函數，可在與目前資料列（視窗）相關的指定資料列中執行計算，啟用排名、執行總計和移動平均值（例如 ROW_NUMBER、RANK、LAG、LEAD）等操作	支援的	支援的	支援的	`SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) AS field2Rank FROM <tableName/logGroup>`
the section called “轉換函數”	用於在 SQL 查詢中將資料從一種類型轉換為另一種類型的內建函數，啟用資料類型轉換和格式轉換（例如 CAST、TO_DATE、TO_TIMESTAMP、BINARY）	支援的	支援的	支援的	`SELECT CAST('123' AS INT) AS converted_number, CAST(123 AS STRING) AS converted_string FROM <tableName/logGroup>`

可用的 SQL Gramm	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “述詞函數”	根據指定的條件或模式（例如 IN、LIKE、BETWEEN、ISNULL、EXISTS) 評估條件並傳回布林值 (true/false) 的內建函數	支援的	支援的	支援的	`SELECT * FROM <tableName/logGroup> WHERE id BETWEEN 50000 AND 75000`
the section called “映射函數”	將指定的函數套用至集合中的每個元素，將資料轉換為一組新的值。	不支援	支援的	支援的	`SELECT MAP_FILTER( MAP( 'method', method, 'status', CAST(status AS STRING), 'bytes', CAST(bytes AS STRING) ), (k, v) -> k IN ('method', 'status') AND v != 'null' ) AS filtered_map FROM <tableName/logGroup> WHERE status = 100`

可用的 SQL Gramm	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “數學函式”	對數值資料執行數學操作，例如計算平均值、總和或三角值。	支援的	支援的	支援的	`SELECT bytes, bytes + 1000 AS added, bytes - 1000 AS subtracted, bytes * 2 AS doubled, bytes / 1024 AS kilobytes, bytes % 1000 AS remainder FROM <tableName/logGroup>`
the section called “多日誌群組函數”	可讓使用者在 SQL SELECT 陳述式中指定多個日誌群組	支援的	不適用	不適用	`SELECT lg1.Column1, lg1.Column2 FROM `logGroups(logGroupIdentifier: ['LogGroup1', 'LogGroup2'])` AS lg1 WHERE lg1.Column3 = "Success"`
the section called “產生器函數”	建立會產生一系列值的迭代器物件，以便在大型資料集中有效率地使用記憶體。	不支援	支援的	支援的	`SELECT explode(array(10, 20))`

一般 SQL 限制

將 OpenSearch SQL 與 CloudWatch Logs、Amazon S3 和 Security Lake 搭配使用時，適用下列限制。

1. 您只能在 SELECT 陳述式中使用一個 JOIN 操作。
2. 僅支援一個層級的巢狀子查詢。
3. 不支援以分號分隔的多個陳述式查詢。
4. 不支援包含相同但僅在 欄位名稱 （ 例如 field1 和 FIELD1) 時才不同的查詢。

   例如，不支援下列查詢：

   ```
   Select AWSAccountId, awsaccountid from LogGroup
   ```

   不過，下列查詢是由於兩個日誌群組中的欄位名稱 (@logStream) 相同：

   ```
   Select a.`@logStream`, b.`@logStream` from Table A INNER Join Table B on a.id = b.id
   ```
5. 函數和表達式必須在欄位名稱上操作，並成為 SELECT 陳述式的一部分，其中包含 FROM 子句中指定的日誌群組。

   例如，不支援此查詢：

   ```
   SELECT cos(10) FROM LogGroup
   ```

   支援此查詢：

   ```
   SELECT cos(field1) FROM LogGroup
   ```

使用 OpenSearch SQL 的 CloudWatch Logs Insights 使用者其他資訊

CloudWatch Logs 支援 Logs Insights 主控台、API 和 CLI 中的 OpenSearch SQL 查詢。它支援大多數命令，包括 SELECT、 FROM、WHERE、GROUP BY、HAVING、JOINS 和巢狀查詢，以及 JSON、數學、字串和條件式函數。不過，CloudWatch Logs 僅支援讀取操作，因此不允許 DDL 或 DML 陳述式。如需支援命令和函數的完整清單，請參閱上一節中的資料表。

多日誌群組函數

CloudWatch Logs Insights 支援查詢多個日誌群組的功能。若要在 SQL 中解決此使用案例，您可以使用 logGroups命令。此命令專用於在涉及一或多個日誌群組的 CloudWatch Logs Insights 中查詢資料。使用此語法透過在命令中指定多個日誌群組來查詢多個日誌群組，而不是為每個日誌群組撰寫查詢，並將其與UNION命令結合。

語法：

```
`logGroups(
    logGroupIdentifier: ['LogGroup1','LogGroup2', ...'LogGroupn']
)
```

在此語法中，您可以在 logGroupIndentifier 參數中指定最多 50 個日誌群組。若要參考監控帳戶中的日誌群組，請使用 ARNs 而非LogGroup名稱。

查詢範例：

```
SELECT LG1.Column1, LG1.Column2 from `logGroups(
    logGroupIdentifier: ['LogGroup1', 'LogGroup2']
)` as LG1
WHERE LG1.Column1 = 'ABC'
```

查詢 CloudWatch Logs 時，不支援在FROM陳述式之後涉及多個日誌群組的下列語法：

```
SELECT Column1, Column2 FROM 'LogGroup1', 'LogGroup2', ...'LogGroupn'
WHERE Column1 = 'ABC'
```

限制

當您使用 SQL 或 PPL 命令時，請將特定欄位括在反引號中以查詢它們。具有特殊字元（非字母和非數字）的欄位需要反引號。例如，將 @messageOperation.Export,和 括在反引號Test::Field中。您不需要將純字母名稱的資料欄括在反引號中。

具有簡單欄位的範例查詢：

```
SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
LIMIT 1000;
```

附加反引號的相同查詢：

```
SELECT `SessionToken`, `Operation`, `StartTime`  FROM `LogGroup-A`
LIMIT 1000;
```

如需非 CloudWatch Logs 特有的其他一般限制，請參閱 the section called “一般 SQL 限制”。

查詢和配額範例

> **Note**
>
> 以下內容適用於查詢 CloudWatch 資料的 CloudWatch Logs Insights 使用者和 OpenSearch 使用者。

如需您可以在 CloudWatch Logs 中使用的 SQL 查詢範例，請參閱 Amazon CloudWatch Logs Insights 主控台中的已儲存和範例查詢範例。

如需從 OpenSearch Service 查詢 CloudWatch Logs 時所套用限制的相關資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的 CloudWatch Logs 配額。 Amazon CloudWatch 限制包括您可以查詢的 CloudWatch Log 群組數量、您可以執行的並行查詢上限、查詢執行時間上限，以及結果中傳回的資料列數目上限。無論您使用哪種語言查詢 CloudWatch Logs（即 OpenSearch PPL、SQL 和 Logs Insights），限制都相同。

SQL 命令

主題

- 字串函數
- 日期和時間函數
- 彙總函數
- 條件函數
- JSON 函數
- 陣列函數
- 範圍函數
- 轉換函數
- 述詞函數
- 映射函數
- 數學函式

- 產生器函數
- SELECT 子句
- WHERE 子句
- GROUP BY 子句
- HAVING 子句
- ORDER BY 子句
- JOIN 子句
- LIMIT 子句
- CASE 子句
- 常見資料表表達式
- EXPLAIN
- LATERAL SUBQUERY 子句
- LATERAL VIEW 子句
- LIKE 述詞
- OFFSET
- PIVOT 子句
- 設定運算子
- SORT BY 子句
- UNPIVOT

字串函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
ascii(str)	傳回 第一個字元的數值str。
base64(bin)	將引數從二進位轉換為bin基本 64 字串。

函式	Description
bit_length(expr)	傳回字串資料的位元長度或二進位資料的位元數。
btrim(str)	從 移除開頭和結尾空格字元str。
btrim(str， trimStr)	從 移除開頭和結尾trimStr字元str。
char(expr)	傳回二進位等於 的 ASCII 字元expr。如果 n 大於 256，則結果等同於 chr(n % 256)
char_length(expr)	傳回字串資料的字元長度或二進位資料的位元組數。字串資料的長度包含結尾空格。二進位資料的長度包含二進位零。
character_length(expr)	傳回字串資料的字元長度或二進位資料的位元組數。字串資料的長度包含結尾空格。二進位資料的長度包含二進位零。
chr(expr)	傳回二進位等於 的 ASCII 字元expr。如果 n 大於 256，則結果等同於 chr(n % 256)
concat_ws(sep【， str | array(str)】 +)	傳回以 分隔的字串串連sep，略過 null 值。
contains（左、右）	傳回布林值。如果在左側找到右側，則值為 True。如果任一輸入表達式為 NULL，則傳回 NULL。否則， 會傳回 False。左側或右側都必須是 STRING 或 BINARY 類型。
decode(bin， 字元集）	使用第二個引數字元集解碼第一個引數。
decode(expr， search， results 【， search， results 】 ... 【， default】)	依序比較 expr 與每個搜尋值。如果 expr 等於搜尋值，解碼會傳回對應的結果。如果找不到相符項目，則會傳回預設值。如果省略預設值，則會傳回 null。
elt(n， input1， input2， ...)	傳回 n-th 輸入，例如，當 n為 2 input2時傳回 。

函式	Description
encode(str， charset)	使用第二個引數字元集編碼第一個引數。
endwith（左、右）	傳回布林值。如果左結尾為右，則值為 True。如果任一輸入表達式為 NULL，則傳回 NULL。否則， 會傳回 False。左側或右側都必須是 STRING 或 BINARY 類型。
find_in_set(str， str_array)	傳回逗號分隔清單 () 中指定字串 (str) 的索引 (1 型）str_array 。如果找不到字串，或指定的字串 (str) 包含逗號，則傳回 0。
format_number(expr1， expr2)	將數字格式化expr1為 '#，###，###.##'，四捨五入為expr2小數位數。如果 expr2為 0，則結果沒有小數點或小數部分。 expr2也接受使用者指定的格式。這應該像 MySQL 的 FORMAT 一樣運作。
format_string(strfmt、obj、...)	從 printf 格式格式字串傳回格式化字串。
initcap(str)	str 以大寫字母傳回每個單字的第一個字母。所有其他字母皆為小寫。字詞以空格分隔。
instr(str， substr)	傳回 substr中第一次出現 的 (1 型） 索引str。
lcase(str)	傳回 str ，並將所有字元變更為小寫。
left(str， len)	從字串 傳回最左邊 len(len 可以是字串類型）字元str，如果len小於或等於 0，則結果為空字串。
len(expr)	傳回字串資料的字元長度或二進位資料的位元組數。字串資料的長度包含結尾空格。二進位資料的長度包含二進位零。

函式	Description
length(expr)	傳回字串資料的字元長度或二進位資料的位元組數。字串資料的長度包含結尾空格。二進位資料的長度包含二進位零。
levenshtein(str1， str2【， threshold】)	傳回兩個指定字串之間的 Levenshtein 距離。如果設定閾值且距離超過閾值，請傳回 -1。
locate(substr、str【、pos】)	傳回位置 str之後第一次在 substr中出現的位置pos。指定值pos和傳回值是以 1 為基礎。
lower(str)	傳回 str ，並將所有字元變更為小寫。
lpad(str， len【， pad】)	傳回 str，以左填充，長度pad為 len。如果 str 超過 len，則傳回值會縮短為len字元或位元組。如果pad未指定 ，則如果是字元字串，str則會以空格字元填入左側，如果是位元組序列，則會以零填入。
ltrim(str)	從 移除前置空格字元str。
luhn_check(str)	根據 Luhn 演算法檢查數字字串是否有效。此檢查總和函數廣泛套用至信用卡號碼和政府識別號碼，以區分有效號碼與輸入錯誤、不正確的號碼。
mask(input【， upperChar， lowerChar， digitChar， otherChar】)	遮罩指定的字串值。函數會將字元取代為 'X' 或 'x'，並將數字取代為 'n'。這對於建立已移除敏感資訊的資料表複本很有用。
octet_length(expr)	傳回字串資料的位元組長度或二進位資料的位元組數。
overlay (輸入、取代、pos【、len】)	input 以replace開頭為 pos且長度為 的 取代 len。
position(substr、str【、pos】)	傳回位置 str之後第一次在 substr中出現的位置pos。指定值pos和傳回值是以 1 為基礎。

函式	Description
printf(strfmt、obj、...)	從 printf 格式格式字串傳回格式化字串。
regexp_count(str， regexp)	傳回字串 中規則表達式模式regexp相符的次數計數str。
regexp_extract(str、regexp【、idx】)	在 中擷取str符合regexp表達式且對應至 regex 群組索引的第一個字串。
regexp_extract_all(str、regexp【、idx】)	擷取 中str符合regexp表達式且對應至 regex 群組索引的所有字串。
regexp_instr(str， regexp)	搜尋規則表達式的字串，並傳回整數，指出相符子字串的開始位置。位置是以 1 為基礎，而不是以 0 為基礎。如果找不到相符項目， 會傳回 0。
regexp_replace(str、regexp、rep【、position】)	str 將相符的所有子字串取代regexp為 rep。
regexp_substr(str， regexp)	傳回符合字串 regexp內規則表達式的子字串str。如果找不到規則表達式，則結果為 null。
repeat(str， n)	傳回重複指定字串值 n 次的字串。
replace(str， search【， replace】)	search 以 取代 的所有出現次數replace。
right(str， len)	從字串 傳回最右側的 len(len 可以是字串類型） 字元str，如果len小於或等於 0，則結果為空字串。
rpad(str， len【， pad】)	傳回 str，以右填充，長度pad為 len。如果 str 超過 len，則傳回值會縮短為 len 個字元。如果pad未指定 ，則如果字元字串是空格字元，str則會在右側填入 ，如果是二進位字串，則會填入零。

函式	Description
rtrim(str)	從 移除結尾空格字元str。
sentences(str【， lang， country】)	分割str成單字陣列。
soundex(str)	傳回字串的 Soundex 程式碼。
space(n)	傳回由n空格組成的字串。
split(str、regex、 limit)	str 依符合的項目分割，regex並傳回長度上限為 的陣列 limit
split_part(str， delimiter， partNum)	str 依分隔符號分割，並傳回請求的部分分割 (1 型)。如果任何輸入為 Null， 會傳回 Null。如果 partNum超出分割部分的範圍， 會傳回空字串。如果 partNum為 0， 會擲回錯誤。如果 partNum為負數，則部分會從字串結尾回溯計數。如果 delimiter 是空字串，str則 不會分割。
startwith (左、右)	傳回布林值。如果左開頭為右，則值為 True。如果任一輸入表達式為 NULL，則傳回 NULL。否則， 會傳回 False。左側或右側都必須是 STRING 或 BINARY 類型。
substr(str、pos【、len】)	傳回str從 開始pos且長度為 的 子字串len，或從 pos開始且長度為 的位元組陣列配量len。
substr(str FROM pos【 FOR len】】)	傳回str從 開始pos且長度為 的 子字串len，或從 pos開始且長度為 的位元組陣列配量len。
substring(str、pos【、len】)	傳回str從 開始pos且長度為 的 子字串len，或從 pos開始且長度為 的位元組陣列配量len。

函式	Description
substring(str FROM pos【 FOR len】】)	傳回str從 開始pos且長度為 的 子字串len，或從 pos開始且長度為 的位元組陣列配量len。
substring_index(str， delim， count)	在count出現分隔符號 str之前，從 傳回子字串delim。如果 count 為正數，則會傳回最終分隔符號左側的所有項目 （從左側計數）。如果 count 為負數，則會傳回最終分隔符號右側的所有資料 （從右側計數）。函數 substring_index 會在搜尋 時執行區分大小寫的比對delim。
to_binary(str【， fmt】)	根據提供的 str 將輸入轉換為二進位值fmt。fmt可以是不區分大小寫的字串常值 "hex"、"utf-8"、"utf8" 或 "base64"。根據預設，如果fmt省略 ，則轉換的二進位格式為「十六」。如果至少一個輸入參數為 NULL，則函數會傳回 NULL。
to_char(numberExpr， formatExpr)	numberExpr 轉換為以 為基礎的字串formatExpr 。如果轉換失敗，則擲回例外狀況。格式可以包含下列字元，不區分大小寫：'0' 或 '9'：指定介於 0 和 9 之間的預期數字。格式字串中的 0 或 9 序列符合輸入值中的數字序列，產生與格式字串中對應序列相同長度的結果字串。如果 0/9 序列包含的數字多於小數值的相符部分，開頭為 0，且位於小數點之前，則結果字串會加上零。否則，它會填充空格。 '.' 或 'D'：指定小數點的位置 （選用，只允許一次）。' , ' 或 'G'：指定分組 （千） 分隔符號的位置 (,)。每個分組分隔符號的左側和右側必須有一個 0 或 9。 '

函式	Description
to_number(expr， fmt)	將字串 'expr' 轉換為以字串格式 'fmt' 為基礎的數字。如果轉換失敗，則擲回例外狀況。格式可以包含下列字元，不區分大小寫：'0' 或 '9'：指定介於 0 和 9 之間的預期數字。格式字串中的 0 或 9 序列符合輸入字串中的數字序列。如果 0/9 序列以 0 開頭，且位於小數點之前，則只能比對相同大小的數字序列。否則，如果序列以 9 開頭或在小數點之後，它可以比對大小相同或更小的數字序列。 '.' 或 'D'：指定小數點的位置 （選用，只允許一次）。' , ' 或 'G'：指定分組 （千） 分隔符號的位置 (,)。每個分組分隔符號的左側和右側必須有 0 或 9。'expr' 必須符合與數字大小相關的分組分隔符號。 '
to_varchar(numberExpr， formatExpr)	`numberExpr` 轉換為以 為基礎的字串`formatExpr` 。如果轉換失敗，則擲回例外狀況。格式可以包含下列字元，不區分大小寫：'0' 或 '9'：指定介於 0 和 9 之間的預期數字。格式字串中的 0 或 9 序列符合輸入值中的數字序列，產生與格式字串中對應序列相同長度的結果字串。如果 0/9 序列包含的數字多於小數值的相符部分，開頭為 0，且位於小數點之前，則結果字串會加上零。否則，它會填充空格。 '.' 或 'D'：指定小數點的位置 （選用，只允許一次）。' , ' 或 'G'：指定分組 （千） 分隔符號的位置 (,)。每個分組分隔符號的左側和右側必須有一個 0 或 9。 '
translate(input， from， to)	將`from`字串中存在的字元取代為`input`字串中對應的字元，以翻譯`to`字串。
trim(str)	從 移除開頭和結尾空格字元`str`。
trim（兩者都從 str)	從 移除開頭和結尾空格字元`str`。
trim(LEADING FROM str)	從 移除前置空格字元`str`。

函式	Description
trim(TRAILING FROM str)	從 移除結尾空格字元str。
trim(trimStr FROM str)	從 移除開頭和結尾trimStr字元str。
trim（兩個 trimStr FROM str)	從 移除開頭和結尾trimStr字元str。
trim(LEADING trimStr FROM str)	從 移除前置trimStr字元str。
trim(TRAILING trimStr FROM str)	從 移除結尾trimStr字元str。
try_to_binary(str【， fmt】)	這是to_binary 執行相同操作的特殊 版本，但如果無法執行轉換，則傳回 NULL 值，而不是引發錯誤。
try_to_number(expr， fmt)	將字串 'expr' 轉換為以字串格式 為基礎的數字fmt。如果字串 'expr' 不符合預期的格式，則傳回 NULL。格式遵循與 to_number 函數相同的語意。
ucase(str)	傳回 str ，並將所有字元變更為大寫。
unbase64(str)	將引數從基礎 64 字串str轉換為二進位。
upper(str)	傳回 str ，並將所有字元變更為大寫。

範例

```
-- ascii
SELECT ascii('222');
+----------+
|ascii(222)|
+----------+
|        50|
+----------+
SELECT ascii(2);
+--------+
|ascii(2)|
+--------+
```

```
|      50|
+--------+
-- base64
SELECT base64('Feathers');
+-----------------+
|base64(Feathers)|
+-----------------+
|     RmVhdGhlcnM=|
+-----------------+
SELECT base64(x'537061726b2053514c');
+-----------------------------+
|base64(X'537061726B2053514C')|
+-----------------------------+
|                 U3BhcmsgU1FM|
+-----------------------------+
-- bit_length
SELECT bit_length('Feathers');
+---------------------+
|bit_length(Feathers)|
+---------------------+
|                   64|
+---------------------+
SELECT bit_length(x'537061726b2053514c');
+---------------------------------+
|bit_length(X'537061726B2053514C')|
+---------------------------------+
|                               72|
+---------------------------------+
-- btrim
SELECT btrim('    Feathers   ');
+----------------------+
|btrim(    Feathers   )|
+----------------------+
|              Feathers|
+----------------------+
SELECT btrim(encode('    Feathers   ', 'utf-8'));
+-----------------------------------+
|btrim(encode(    Feathers   , utf-8))|
+-----------------------------------+
|                           Feathers|
+-----------------------------------+
SELECT btrim('Feathers', 'Fe');
+---------------------+
|btrim(Alphabet, Al)|
```

```
+--------------------+
|              athers|
+--------------------+
SELECT btrim(encode('Feathers', 'utf-8'), encode('Al', 'utf-8'));
+---------------------------------------------------+
|btrim(encode(Feathers, utf-8), encode(Al, utf-8))|
+---------------------------------------------------+
|                                             athers|
+---------------------------------------------------+
-- char
SELECT char(65);
+--------+
|char(65)|
+--------+
|       A|
+--------+
-- char_length
SELECT char_length('Feathers ');
+-----------------------+
|char_length(Feathers )|
+-----------------------+
|                      9 |
+-----------------------+
SELECT char_length(x'537061726b2053514c');
+----------------------------------+
|char_length(X'537061726B2053514C')|
+----------------------------------+
|                                 9|
+----------------------------------+
SELECT CHAR_LENGTH('Feathers ');
+-----------------------+
|char_length(Feathers )|
+-----------------------+
|                      9|
+-----------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+---------------------------+
|character_length(Feathers )|
+---------------------------+
|                          9|
+---------------------------+
-- character_length
SELECT character_length('Feathers ');
+---------------------------+
```

```
|character_length(Feathers )|
+---------------------------+
|                          9|
+---------------------------+
SELECT character_length(x'537061726b2053514c');
+--------------------------------------+
|character_length(X'537061726B2053514C')|
+--------------------------------------+
|                                     9|
+--------------------------------------+
SELECT CHAR_LENGTH('Feathers ');
+----------------------+
|char_length(Feathers )|
+----------------------+
|                     9|
+----------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+---------------------------+
|character_length(Feathers )|
+---------------------------+
|                          9|
+---------------------------+
-- chr
SELECT chr(65);
+-------+
|chr(65)|
+-------+
|      A|
+-------+
-- concat_ws
SELECT concat_ws(' ', 'Fea', 'thers');
+-----------------------+
|concat_ws( , Fea, thers)|
+-----------------------+
|               Feathers|
+-----------------------+
SELECT concat_ws('s');
+------------+
|concat_ws(s)|
+------------+
|            |
+------------+
SELECT concat_ws('/', 'foo', null, 'bar');
+---------------------------+
```

```
|concat_ws(/, foo, NULL, bar)|
+----------------------------+
|                     foo/bar|
+----------------------------+
SELECT concat_ws(null, 'Fea', 'thers');
+----------------------------+
|concat_ws(NULL, Fea, thers)|
+----------------------------+
|                        NULL|
+----------------------------+
-- contains
SELECT contains('Feathers', 'Fea');
+--------------------------+
|contains(Feathers, Fea)|
+--------------------------+
|                      true|
+--------------------------+
SELECT contains('Feathers', 'SQL');
+--------------------------+
|contains(Feathers, SQL)|
+--------------------------+
|                     false|
+--------------------------+
SELECT contains('Feathers', null);
+--------------------------+
|contains(Feathers, NULL)|
+--------------------------+
|                     NULL|
+--------------------------+
SELECT contains(x'537061726b2053514c', x'537061726b');
+-----------------------------------------------+
|contains(X'537061726B2053514C', X'537061726B')|
+-----------------------------------------------+
|                                           true|
+-----------------------------------------------+
-- decode
SELECT decode(encode('abc', 'utf-8'), 'utf-8');
+---------------------------------+
|decode(encode(abc, utf-8), utf-8)|
+---------------------------------+
|                              abc|
+---------------------------------+
SELECT decode(2, 1, 'Southlake', 2, 'San Francisco', 3, 'New Jersey', 4, 'Seattle',
 'Non domestic');
```

```
+-----------------------------------------------------------------------------------------+
|decode(2, 1, Southlake, 2, San Francisco, 3, New Jersey, 4, Seattle, Non domestic)|
+-----------------------------------------------------------------------------------------+
|                                                                            San Francisco|
+-----------------------------------------------------------------------------------------+
SELECT decode(6, 1, 'Southlake', 2, 'San Francisco', 3, 'New Jersey', 4, 'Seattle',
 'Non domestic');
+-----------------------------------------------------------------------------------------+
|decode(6, 1, Southlake, 2, San Francisco, 3, New Jersey, 4, Seattle, Non domestic)|
+-----------------------------------------------------------------------------------------+
|                                                                             Non domestic|
+-----------------------------------------------------------------------------------------+
SELECT decode(6, 1, 'Southlake', 2, 'San Francisco', 3, 'New Jersey', 4, 'Seattle');
+-------------------------------------------------------------------+
|decode(6, 1, Southlake, 2, San Francisco, 3, New Jersey, 4, Seattle)|
+-------------------------------------------------------------------+
|                                                               NULL|
+-------------------------------------------------------------------+
SELECT decode(null, 6, 'Fea', NULL, 'thers', 4, 'rock');
+-----------------------------------------+
|decode(NULL, 6, Fea, NULL, thers, 4, rock)|
+-----------------------------------------+
|                                    thers|
+-----------------------------------------+
-- elt
SELECT elt(1, 'scala', 'java');
+------------------+
|elt(1, scala, java)|
+------------------+
|              scala|
+------------------+
SELECT elt(2, 'a', 1);
+------------+
|elt(2, a, 1)|
+------------+
|           1|
+------------+
-- encode
SELECT encode('abc', 'utf-8');
+------------------+
|encode(abc, utf-8)|
+------------------+
|        [61 62 63]|
+------------------+
```

```
-- endswith
SELECT endswith('Feathers', 'ers');
+------------------------+
|endswith(Feathers, ers)|
+------------------------+
|                    true|
+------------------------+
SELECT endswith('Feathers', 'SQL');
+-------------------------+
|endswith(Feathers, SQL)|
+-------------------------+
|                     false|
+-------------------------+
SELECT endswith('Feathers', null);
+-------------------------+
|endswith(Feathers, NULL)|
+-------------------------+
|                     NULL|
+-------------------------+
SELECT endswith(x'537061726b2053514c', x'537061726b');
+-----------------------------------------------+
|endswith(X'537061726B2053514C', X'537061726B')|
+-----------------------------------------------+
|                                          false|
+-----------------------------------------------+
SELECT endswith(x'537061726b2053514c', x'53514c');
+---------------------------------------+
|endswith(X'537061726B2053514C', X'53514C')|
+---------------------------------------+
|                                   true|
+---------------------------------------+
-- find_in_set
SELECT find_in_set('ab','abc,b,ab,c,def');
+-------------------------------+
|find_in_set(ab, abc,b,ab,c,def)|
+-------------------------------+
|                              3|
+-------------------------------+
-- format_number
SELECT format_number(12332.123456, 4);
+-----------------------------+
|format_number(12332.123456, 4)|
+-----------------------------+
|                  12,332.1235|
```

```
+------------------------------+
SELECT format_number(12332.123456, '##################.###');
+--------------------------------------------------+
|format_number(12332.123456, ##################.###)|
+--------------------------------------------------+
|                                         12332.123|
+--------------------------------------------------+
-- format_string
SELECT format_string("Hello World %d %s", 100, "days");
+-------------------------------------------+
|format_string(Hello World %d %s, 100, days)|
+-------------------------------------------+
|                      Hello World 100 days|
+-------------------------------------------+
-- initcap
SELECT initcap('Feathers');
+------------------+
|initcap(Feathers)|
+------------------+
|         Feathers|
+------------------+
-- instr
SELECT instr('Feathers', 'ers');
+--------------------+
|instr(Feathers, ers)|
+--------------------+
|                   6|
+--------------------+
-- lcase
SELECT lcase('Feathers');
+---------------+
|lcase(Feathers)|
+---------------+
|       feathers|
+---------------+
-- left
SELECT left('Feathers', 3);
+------------------+
|left(Feathers, 3)|
+------------------+
|               Fea|
+------------------+
SELECT left(encode('Feathers', 'utf-8'), 3);
+---------------------------------+
```

```
|left(encode(Feathers, utf-8), 3)|
+--------------------------------+
|                       [RmVh]|
+--------------------------------+
-- len
SELECT len('Feathers ');
+---------------+
|len(Feathers )|
+---------------+
|             9|
+---------------+
SELECT len(x'537061726b2053514c');
+--------------------------+
|len(X'537061726B2053514C')|
+--------------------------+
|                         9|
+--------------------------+
SELECT CHAR_LENGTH('Feathers ');
+-----------------------+
|char_length(Feathers )|
+-----------------------+
|                      9|
+-----------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+---------------------------+
|character_length(Feathers )|
+---------------------------+
|                          9|
+---------------------------+
-- length
SELECT length('Feathers ');
+------------------+
|length(Feathers )|
+------------------+
|                9|
+------------------+
SELECT length(x'537061726b2053514c');
+----------------------------+
|length(X'537061726B2053514C')|
+----------------------------+
|                           9|
+----------------------------+
SELECT CHAR_LENGTH('Feathers ');
+-----------------------+
```

```
|char_length(Feathers )|
+----------------------+
|                     9|
+----------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+---------------------------+
|character_length(Feathers )|
+---------------------------+
|                          9|
+---------------------------+
-- levenshtein
SELECT levenshtein('kitten', 'sitting');
+---------------------------+
|levenshtein(kitten, sitting)|
+---------------------------+
|                           3|
+---------------------------+
SELECT levenshtein('kitten', 'sitting', 2);
+-------------------------------+
|levenshtein(kitten, sitting, 2)|
+-------------------------------+
|                             -1|
+-------------------------------+
-- locate
SELECT locate('bar', 'foobarbar');
+-------------------------+
|locate(bar, foobarbar, 1)|
+-------------------------+
|                        4|
+-------------------------+
SELECT locate('bar', 'foobarbar', 5);
+-------------------------+
|locate(bar, foobarbar, 5)|
+-------------------------+
|                        7|
+-------------------------+
SELECT POSITION('bar' IN 'foobarbar');
+-------------------------+
|locate(bar, foobarbar, 1)|
+-------------------------+
|                        4|
+-------------------------+
-- lower
SELECT lower('Feathers');
```

```
+---------------+
|lower(Feathers)|
+---------------+
|       feathers|
+---------------+
-- lpad
SELECT lpad('hi', 5, '??');
+---------------+
|lpad(hi, 5, ??)|
+---------------+
|          ???hi|
+---------------+
SELECT lpad('hi', 1, '??');
+---------------+
|lpad(hi, 1, ??)|
+---------------+
|              h|
+---------------+
SELECT lpad('hi', 5);
+--------------+
|lpad(hi, 5,  )|
+--------------+
|            hi|
+--------------+
SELECT hex(lpad(unhex('aabb'), 5));
+--------------------------------+
|hex(lpad(unhex(aabb), 5, X'00'))|
+--------------------------------+
|                      000000AABB|
+--------------------------------+
SELECT hex(lpad(unhex('aabb'), 5, unhex('1122')));
+-------------------------------------+
|hex(lpad(unhex(aabb), 5, unhex(1122)))|
+-------------------------------------+
|                           112211AABB|
+-------------------------------------+
-- ltrim
SELECT ltrim('    Feathers   ');
+----------------------+
|ltrim(    Feathers   )|
+----------------------+
|           Feathers   |
+----------------------+
-- luhn_check
```

```
SELECT luhn_check('8112189876');
+----------------------+
|luhn_check(8112189876)|
+----------------------+
|                  true|
+----------------------+
SELECT luhn_check('79927398713');
+-----------------------+
|luhn_check(79927398713)|
+-----------------------+
|                   true|
+-----------------------+
SELECT luhn_check('79927398714');
+-----------------------+
|luhn_check(79927398714)|
+-----------------------+
|                  false|
+-----------------------+
-- mask
SELECT mask('abcd-EFGH-8765-4321');
+----------------------------------------+
|mask(abcd-EFGH-8765-4321, X, x, n, NULL)|
+----------------------------------------+
|                     xxxx-XXXX-nnnn-nnnn|
+----------------------------------------+
SELECT mask('abcd-EFGH-8765-4321', 'Q');
+----------------------------------------+
|mask(abcd-EFGH-8765-4321, Q, x, n, NULL)|
+----------------------------------------+
|                     xxxx-QQQQ-nnnn-nnnn|
+----------------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q');
+--------------------------------+
|mask(AbCD123-@$#, Q, q, n, NULL)|
+--------------------------------+
|                     QqQQnnn-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#');
+--------------------------------+
|mask(AbCD123-@$#, X, x, n, NULL)|
+--------------------------------+
|                     XxXXnnn-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q');
```

```
+--------------------------------+
|mask(AbCD123-@$#, Q, x, n, NULL)|
+--------------------------------+
|                     QxQQnnn-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q');
+--------------------------------+
|mask(AbCD123-@$#, Q, q, n, NULL)|
+--------------------------------+
|                     QqQQnnn-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q', 'd');
+--------------------------------+
|mask(AbCD123-@$#, Q, q, d, NULL)|
+--------------------------------+
|                     QqQQddd-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q', 'd', 'o');
+-----------------------------+
|mask(AbCD123-@$#, Q, q, d, o)|
+-----------------------------+
|                  QqQQdddoooo|
+-----------------------------+
SELECT mask('AbCD123-@$#', NULL, 'q', 'd', 'o');
+--------------------------------+
|mask(AbCD123-@$#, NULL, q, d, o)|
+--------------------------------+
|                     AqCDdddoooo|
+--------------------------------+
SELECT mask('AbCD123-@$#', NULL, NULL, 'd', 'o');
+---------------------------------+
|mask(AbCD123-@$#, NULL, NULL, d, o)|
+---------------------------------+
|                        AbCDdddoooo|
+---------------------------------+
SELECT mask('AbCD123-@$#', NULL, NULL, NULL, 'o');
+----------------------------------+
|mask(AbCD123-@$#, NULL, NULL, NULL, o)|
+----------------------------------+
|                           AbCD123oooo|
+----------------------------------+
SELECT mask(NULL, NULL, NULL, NULL, 'o');
+-----------------------------+
|mask(NULL, NULL, NULL, NULL, o)|
```

```
+------------------------------+
|                          NULL|
+------------------------------+
SELECT mask(NULL);
+-------------------------+
|mask(NULL, X, x, n, NULL)|
+-------------------------+
|                     NULL|
+-------------------------+
SELECT mask('AbCD123-@$#', NULL, NULL, NULL, NULL);
+----------------------------------------+
|mask(AbCD123-@$#, NULL, NULL, NULL, NULL)|
+----------------------------------------+
|                            AbCD123-@$#|
+----------------------------------------+
-- octet_length
SELECT octet_length('Feathers');
+----------------------+
|octet_length(Feathers)|
+----------------------+
|                     8|
+----------------------+
SELECT octet_length(x'537061726b2053514c');
+-----------------------------------+
|octet_length(X'537061726B2053514C')|
+-----------------------------------+
|                                  9|
+-----------------------------------+
-- overlay
SELECT overlay('Feathers' PLACING '_' FROM 6);
+---------------------------+
|overlay(Feathers, _, 6, -1)|
+---------------------------+
|                  Feathe_ers|
+---------------------------+
SELECT overlay('Feathers' PLACING 'ures' FROM 5);
+-----------------------------+
|overlay(Feathers, ures, 5, -1)|
+-----------------------------+
|                     Features  |
+-----------------------------+
-- position
SELECT position('bar', 'foobarbar');
+-------------------------+
```

```
|position(bar, foobarbar, 1)|
+---------------------------+
|                          4|
+---------------------------+
SELECT position('bar', 'foobarbar', 5);
+---------------------------+
|position(bar, foobarbar, 5)|
+---------------------------+
|                          7|
+---------------------------+
SELECT POSITION('bar' IN 'foobarbar');
+-------------------------+
|locate(bar, foobarbar, 1)|
+-------------------------+
|                        4|
+-------------------------+
-- printf
SELECT printf("Hello World %d %s", 100, "days");
+-----------------------------------+
|printf(Hello World %d %s, 100, days)|
+-----------------------------------+
|                Hello World 100 days|
+-----------------------------------+
-- regexp_count
SELECT regexp_count('Steven Jones and Stephen Smith are the best players', 'Ste(v|
ph)en');
+-------------------------------------------------------------------------------+
|regexp_count(Steven Jones and Stephen Smith are the best players, Ste(v|ph)en)|
+-------------------------------------------------------------------------------+
|                                                                              2|
+-------------------------------------------------------------------------------+
SELECT regexp_count('abcdefghijklmnopqrstuvwxyz', '[a-z]{3}');
+-----------------------------------------------+
|regexp_count(abcdefghijklmnopqrstuvwxyz, [a-z]{3})|
+-----------------------------------------------+
|                                              8|
+-----------------------------------------------+
-- regexp_extract
SELECT regexp_extract('100-200', '(\\d+)-(\\d+)', 1);
+--------------------------------------+
|regexp_extract(100-200, (\d+)-(\d+), 1)|
+--------------------------------------+
|                                   100|
+--------------------------------------+
```

```
-- regexp_extract_all
SELECT regexp_extract_all('100-200, 300-400', '(\\d+)-(\\d+)', 1);
+-----------------------------------------------------+
|regexp_extract_all(100-200, 300-400, (\d+)-(\d+), 1)|
+-----------------------------------------------------+
|                                          [100, 300]|
+-----------------------------------------------------+
-- regexp_instr
SELECT regexp_instr('user@opensearch.org', '@[^.]*');
+-----------------------------------------------+
|regexp_instr(user@opensearch.org, @[^.]*, 0)|
+-----------------------------------------------+
|                                              5|
+-----------------------------------------------+
-- regexp_replace
SELECT regexp_replace('100-200', '(\\d+)', 'num');
+--------------------------------------+
|regexp_replace(100-200, (\d+), num, 1)|
+--------------------------------------+
|                               num-num|
+--------------------------------------+
-- regexp_substr
SELECT regexp_substr('Steven Jones and Stephen Smith are the best players', 'Ste(v|
ph)en');
+-----------------------------------------------------------------------------------+
|regexp_substr(Steven Jones and Stephen Smith are the best players, Ste(v|ph)en)|
+-----------------------------------------------------------------------------------+
|                                                                             Steven|
+-----------------------------------------------------------------------------------+
SELECT regexp_substr('Steven Jones and Stephen Smith are the best players', 'Jeck');
+-------------------------------------------------------------------------+
|regexp_substr(Steven Jones and Stephen Smith are the best players, Jeck)|
+-------------------------------------------------------------------------+
|                                                                     NULL|
+-------------------------------------------------------------------------+
-- repeat
SELECT repeat('123', 2);
+--------------+
|repeat(123, 2)|
+--------------+
|        123123|
+--------------+
-- replace
SELECT replace('ABCabc', 'abc', 'DEF');
```

```
+-------------------------+
|replace(ABCabc, abc, DEF)|
+-------------------------+
|                   ABCDEF|
+-------------------------+
-- right
SELECT right('Feathers', 3);
+-------------------+
|right(Feathers, 3)|
+-------------------+
|                ers|
+-------------------+
-- rpad
SELECT rpad('hi', 5, '??');
+---------------+
|rpad(hi, 5, ??)|
+---------------+
|          hi???|
+---------------+
SELECT rpad('hi', 1, '??');
+---------------+
|rpad(hi, 1, ??)|
+---------------+
|              h|
+---------------+
SELECT rpad('hi', 5);
+--------------+
|rpad(hi, 5,  )|
+--------------+
|         hi   |
+--------------+
SELECT hex(rpad(unhex('aabb'), 5));
+---------------------------------+
|hex(rpad(unhex(aabb), 5, X'00'))|
+---------------------------------+
|                      AABB000000|
+---------------------------------+
SELECT hex(rpad(unhex('aabb'), 5, unhex('1122')));
+------------------------------------+
|hex(rpad(unhex(aabb), 5, unhex(1122)))|
+------------------------------------+
|                            AABB112211|
+------------------------------------+
-- rtrim
```

```
SELECT rtrim('    Feathers   ');
+----------------------+
|rtrim(    Feathers   )|
+----------------------+
|              Feathers|
+----------------------+
-- sentences
SELECT sentences('Hi there! Good morning.');
+--------------------------------------+
|sentences(Hi there! Good morning., , )|
+--------------------------------------+
|                  [[Hi, there], [Go...|
+--------------------------------------+
-- soundex
SELECT soundex('Miller');
+---------------+
|soundex(Miller)|
+---------------+
|           M460|
+---------------+
-- space
SELECT concat(space(2), '1');
+-------------------+
|concat(space(2), 1)|
+-------------------+
|                  1|
+-------------------+
-- split
SELECT split('oneAtwoBthreeC', '[ABC]');
+--------------------------------+
|split(oneAtwoBthreeC, [ABC], -1)|
+--------------------------------+
|             [one, two, three, ]|
+--------------------------------+
SELECT split('oneAtwoBthreeC', '[ABC]', -1);
+--------------------------------+
|split(oneAtwoBthreeC, [ABC], -1)|
+--------------------------------+
|             [one, two, three, ]|
+--------------------------------+
SELECT split('oneAtwoBthreeC', '[ABC]', 2);
+-------------------------------+
|split(oneAtwoBthreeC, [ABC], 2)|
+-------------------------------+
```

```
|              [one, twoBthreeC]|
+-------------------------------+
-- split_part
SELECT split_part('11.12.13', '.', 3);
+--------------------------+
|split_part(11.12.13, ., 3)|
+--------------------------+
|                        13|
+--------------------------+
-- startswith
SELECT startswith('Feathers', 'Fea');
+---------------------------+
|startswith(Feathers, Fea)|
+---------------------------+
|                       true|
+---------------------------+
SELECT startswith('Feathers', 'SQL');
+--------------------------+
|startswith(Feathers, SQL)|
+--------------------------+
|                      false|
+--------------------------+
SELECT startswith('Feathers', null);
+--------------------------+
|startswith(Feathers, NULL)|
+--------------------------+
|                      NULL|
+--------------------------+
SELECT startswith(x'537061726b2053514c', x'537061726b');
+-------------------------------------------------+
|startswith(X'537061726B2053514C', X'537061726B')|
+-------------------------------------------------+
|                                             true|
+-------------------------------------------------+
SELECT startswith(x'537061726b2053514c', x'53514c');
+-----------------------------------------+
|startswith(X'537061726B2053514C', X'53514C')|
+-----------------------------------------+
|                                    false|
+-----------------------------------------+
-- substr
SELECT substr('Feathers', 5);
+-------------------------------+
|substr(Feathers, 5, 2147483647)|
```

```
+-------------------------------+
|                          hers |
+-------------------------------+
SELECT substr('Feathers', -3);
+-------------------------------+
|substr(Feathers, -3, 2147483647)|
+-------------------------------+
|                            ers|
+-------------------------------+
SELECT substr('Feathers', 5, 1);
+----------------------+
|substr(Feathers, 5, 1)|
+----------------------+
|                     h|
+----------------------+
SELECT substr('Feathers' FROM 5);
+----------------------------------+
|substring(Feathers, 5, 2147483647)|
+----------------------------------+
|                            hers |
+----------------------------------+
SELECT substr('Feathers' FROM -3);
+----------------------------------+
|substring(Feathers, -3, 2147483647)|
+----------------------------------+
|                               ers|
+----------------------------------+
SELECT substr('Feathers' FROM 5 FOR 1);
+-------------------------+
|substring(Feathers, 5, 1)|
+-------------------------+
|                        h|
+-------------------------+
-- substring
SELECT substring('Feathers', 5);
+----------------------------------+
|substring(Feathers, 5, 2147483647)|
+----------------------------------+
|                            hers |
+----------------------------------+
SELECT substring('Feathers', -3);
+----------------------------------+
|substring(Feathers, -3, 2147483647)|
+----------------------------------+
```

```
|                                ers|
+-----------------------------------+
SELECT substring('Feathers', 5, 1);
+--------------------------+
|substring(Feathers, 5, 1)|
+--------------------------+
|                         h|
+--------------------------+
SELECT substring('Feathers' FROM 5);
+-----------------------------------+
|substring(Feathers, 5, 2147483647)|
+-----------------------------------+
|                              hers |
+-----------------------------------+
SELECT substring('Feathers' FROM -3);
+-----------------------------------+
|substring(Feathers, -3, 2147483647)|
+-----------------------------------+
|                                ers|
+-----------------------------------+
SELECT substring('Feathers' FROM 5 FOR 1);
+--------------------------+
|substring(Feathers, 5, 1)|
+--------------------------+
|                         h|
+--------------------------+
-- substring_index
SELECT substring_index('www.apache.org', '.', 2);
+-------------------------------------+
|substring_index(www.apache.org, ., 2)|
+-------------------------------------+
|                           www.apache|
+-------------------------------------+
-- to_binary
SELECT to_binary('abc', 'utf-8');
+---------------------+
|to_binary(abc, utf-8)|
+---------------------+
|           [61 62 63]|
+---------------------+
-- to_char
SELECT to_char(454, '999');
+-----------------+
|to_char(454, 999)|
```

```
+-----------------+
|              454|
+-----------------+
SELECT to_char(454.00, '000D00');
+-----------------------+
|to_char(454.00, 000D00)|
+-----------------------+
|                 454.00|
+-----------------------+
SELECT to_char(12454, '99G999');
+----------------------+
|to_char(12454, 99G999)|
+----------------------+
|                12,454|
+----------------------+
SELECT to_char(78.12, '$99.99');
+----------------------+
|to_char(78.12, $99.99)|
+----------------------+
|                $78.12|
+----------------------+
SELECT to_char(-12454.8, '99G999D9S');
+----------------------------+
|to_char(-12454.8, 99G999D9S)|
+----------------------------+
|                  12,454.8-|
+----------------------------+
-- to_number
SELECT to_number('454', '999');
+-------------------+
|to_number(454, 999)|
+-------------------+
|                454|
+-------------------+
SELECT to_number('454.00', '000.00');
+-------------------------+
|to_number(454.00, 000.00)|
+-------------------------+
|                  454.00|
+-------------------------+
SELECT to_number('12,454', '99,999');
+-------------------------+
|to_number(12,454, 99,999)|
+-------------------------+
```

```
|                    12454|
+-------------------------+
SELECT to_number('$78.12', '$99.99');
+-------------------------+
|to_number($78.12, $99.99)|
+-------------------------+
|                    78.12|
+-------------------------+
SELECT to_number('12,454.8-', '99,999.9S');
+-------------------------------+
|to_number(12,454.8-, 99,999.9S)|
+-------------------------------+
|                       -12454.8|
+-------------------------------+
-- to_varchar
SELECT to_varchar(454, '999');
+-----------------+
|to_char(454, 999)|
+-----------------+
|              454|
+-----------------+
SELECT to_varchar(454.00, '000D00');
+-----------------------+
|to_char(454.00, 000D00)|
+-----------------------+
|                 454.00|
+-----------------------+
SELECT to_varchar(12454, '99G999');
+----------------------+
|to_char(12454, 99G999)|
+----------------------+
|                12,454|
+----------------------+
SELECT to_varchar(78.12, '$99.99');
+----------------------+
|to_char(78.12, $99.99)|
+----------------------+
|                $78.12|
+----------------------+
SELECT to_varchar(-12454.8, '99G999D9S');
+---------------------------+
|to_char(-12454.8, 99G999D9S)|
+---------------------------+
|                   12,454.8-|
```

```
+---------------------------+
-- translate
SELECT translate('AaBbCc', 'abc', '123');
+---------------------------+
|translate(AaBbCc, abc, 123)|
+---------------------------+
|                     A1B2C3|
+---------------------------+
-- try_to_binary
SELECT try_to_binary('abc', 'utf-8');
+-------------------------+
|try_to_binary(abc, utf-8)|
+-------------------------+
|               [61 62 63]|
+-------------------------+
select try_to_binary('a!', 'base64');
+-------------------------+
|try_to_binary(a!, base64)|
+-------------------------+
|                     NULL|
+-------------------------+
select try_to_binary('abc', 'invalidFormat');
+---------------------------------+
|try_to_binary(abc, invalidFormat)|
+---------------------------------+
|                             NULL|
+---------------------------------+
-- try_to_number
SELECT try_to_number('454', '999');
+-----------------------+
|try_to_number(454, 999)|
+-----------------------+
|                    454|
+-----------------------+
SELECT try_to_number('454.00', '000.00');
+-----------------------------+
|try_to_number(454.00, 000.00)|
+-----------------------------+
|                       454.00|
+-----------------------------+
SELECT try_to_number('12,454', '99,999');
+-----------------------------+
|try_to_number(12,454, 99,999)|
+-----------------------------+
```

```
|                        12454|
+-----------------------------+
SELECT try_to_number('$78.12', '$99.99');
+-----------------------------+
|try_to_number($78.12, $99.99)|
+-----------------------------+
|                        78.12|
+-----------------------------+
SELECT try_to_number('12,454.8-', '99,999.9S');
+-----------------------------------+
|try_to_number(12,454.8-, 99,999.9S)|
+-----------------------------------+
|                           -12454.8|
+-----------------------------------+
-- ucase
SELECT ucase('Feathers');
+---------------+
|ucase(Feathers)|
+---------------+
|       FEATHERS|
+---------------+
-- unbase64
SELECT unbase64('U3BhcmsgU1FM');
+----------------------+
|unbase64(U3BhcmsgU1FM)|
+----------------------+
|  [53 70 61 72 6B 2...|
+----------------------+
-- upper
SELECT upper('Feathers');
+---------------+
|upper(Feathers)|
+---------------+
|       FEATHERS|
+---------------+
```

日期和時間函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
add_months(start_date、num_months)	傳回 num_months 之後的日期start_date 。
convert_timezone(【sourceTz，】targetTz，sourceTs)	將不含時區的時間戳記sourceTs從sourceTz時區轉換為targetTz。
curdate()	傳回查詢評估開始時的目前日期。相同查詢內的所有 curdate 呼叫都會傳回相同的值。
current_date()	傳回查詢評估開始時的目前日期。相同查詢中 current_date 的所有呼叫都會傳回相同的值。
current_date	傳回查詢評估開始時的目前日期。
current_timestamp()	傳回查詢評估開始時的目前時間戳記。相同查詢中 current_timestamp 的所有呼叫都會傳回相同的值。
current_timestamp	傳回查詢評估開始時的目前時間戳記。
current_timezone()	傳回目前的工作階段本機時區。
date_add(start_date， num_days)	傳回 num_days之後的日期start_date 。
date_diff(endDate， startDate)	傳回從 startDate 到 的天數endDate。
date_format（時間戳記，fmt)	timestamp 轉換為日期格式 所指定格式的字串值fmt。
date_from_unix_date（天）	從 1970-01-01 開始的天數建立日期。
date_part（欄位，來源）	擷取部分日期/時間戳記或間隔來源。
date_sub(start_date， num_days)	傳回num_days早於 的日期start_date 。
date_trunc(fmt， ts)	傳回截斷至格式模型 ts 所指定單位的時間戳記fmt。

函式	Description
dateadd(start_date、num_days)	傳回 num_days之後的日期start_date 。
datediff(endDate， startDate)	傳回從 startDate 到 的天數endDate。
datepart（欄位，來源）	擷取部分日期/時間戳記或間隔來源。
day（日期）	傳回日期/時間戳記的月份日期。
dayofmonth（日期）	傳回日期/時間戳記的月份日期。
dayofweek（日期）	傳回日期/時間戳記的星期幾 (1 = 星期日，2 = 星期一，...，7 = 星期六）。
dayofyear（日期）	傳回日期/時間戳記的年份日期。
extract（欄位 FROM 來源）	擷取部分日期/時間戳記或間隔來源。
from_unixtime(unix_time【， fmt】)	在指定的 unix_time 中傳回 fmt。
from_utc_timestamp（時間戳記，時區）	假設時間戳記如 '2017-07-14 02：40：00.0'，會以 UTC 將其解譯為時間，並在指定時區將該時間轉譯為時間戳記。例如，「GMT+1」會產生「2017-07-14 03：40：00.0」。
hour（時間戳記）	傳回字串/時間戳記的小時元件。
last_day（日期）	傳回日期所屬月份的最後一天。
localtimestamp()	在查詢評估開始時，傳回不含時區的目前時間戳記。相同查詢內的所有當地時間戳記呼叫都會傳回相同的值。
localtimestamp	在查詢評估開始時，傳回工作階段時區的目前本機日期時間。
make_date（年、月、日）	建立年份、月份和日期欄位的日期。
make_dt_interval(【days【， hours【， mins【， secs】】】】)	從天、小時、分鐘和秒建立 DayTimeIn tervalType 持續時間。

函式	Description
make_interval(【years【，months【，weeks【，days【，hours【，mins【，secs】】】】】】】】)	間隔數年、月、週、日、小時、分鐘和秒。
make_timestamp(年、月、日、小時、分鐘、秒【、時區】)	從年、月、日、小時、分鐘、秒和時區欄位建立時間戳記。
make_timestamp_ltz(年、月、日、小時、分鐘、秒【、時區】)	使用從年、月、日、小時、分鐘、秒和時區欄位的當地時區建立目前的時間戳記。
make_timestamp_ntz(年、月、日、小時、分鐘、秒)	從年、月、日、小時、分鐘、秒欄位建立本機日期時間。
make_ym_interval(【年【，月】】)	以年、月為間隔。
minute(時間戳記)	傳回字串/時間戳記的分鐘元件。
month(date)	傳回日期/時間戳記的月份元件。
months_between(timestamp1， timestamp2【， roundOff】)	如果 `timestamp1` 晚於 `timestamp2`，則結果為正值。如果 `timestamp1` 和 `timestamp2` 在當月的同一天，或兩者都是當月的最後一天，則會忽略時間。否則，差異會根據每月 31 天計算，並四捨五入至 8 位數，除非 roundOff=false。
next_day(start_date， day_of_week)	傳回第一個晚於 `start_date` 並依指示命名的日期。如果至少一個輸入參數是 NULL，則函數會傳回 NULL。
now()	傳回查詢評估開始時的目前時間戳記。
quarter(日期)	傳回日期的年份季度，範圍為 1 到 4。
second(時間戳記)	傳回字串/時間戳記的第二個元件。

函式	Description
session_window(time_column， gap_duration)	根據指定資料欄和間隙持續時間的時間戳記產生工作階段視窗。如需詳細說明和範例，請參閱結構化串流指南文件中的「時段類型」。
timestamp_micros（微秒）	從 UTC epoch 後的微秒數建立時間戳記。
timestamp_millis（毫秒）	從 UTC epoch 後的毫秒數建立時間戳記。
timestamp_seconds（秒）	從 UTC epoch 後的秒數（可以是小數）建立時間戳記。
to_date(date_str【， fmt】)	將date_str具有表達式的fmt表達式剖析為日期。傳回具有無效輸入的 null。根據預設，如果省略 ，它會遵循轉換為日期fmt的規則。
to_timestamp(timestamp_str【， fmt】)	將timestamp_str 表達式與fmt表達式剖析為時間戳記。傳回具有無效輸入的 null。根據預設，如果fmt省略 ，它會遵循將規則轉換為時間戳記。
to_timestamp_ltz(timestamp_str【， fmt】)	將具有timestamp_str 表達式的fmt表達式剖析為具有本機時區的時間戳記。傳回具有無效輸入的 null。根據預設，如果fmt省略 ，它會遵循將規則轉換為時間戳記。
to_timestamp_ntz(timestamp_str【， fmt】)	將具有timestamp_str 表達式的fmt表達式剖析為不含時區的時間戳記。傳回具有無效輸入的 null。根據預設，如果fmt省略 ，它會遵循將規則轉換為時間戳記。
to_unix_timestamp(timeExp【， fmt】)	傳回指定時間的 UNIX 時間戳記。
to_utc_timestamp（時間戳記，時區）	假設時間戳記如 '2017-07-14 02：40：00.0'，會將其解譯為指定時區中的時間，並將該時間轉譯為 UTC 中的時間戳記。例如，「GMT+1」會產生「2017-07-14 01：40：00.0」。

函式	Description
trunc(date， fmt)	傳回 ，date並將一天中的時間部分截斷為格式模型 指定的單位fmt。
try_to_timestamp(timestamp_str【， fmt】)	將timestamp_str 表達式與fmt表達式剖析為時間戳記。
unix_date(date)	傳回自 1970-01-01 起的天數。
unix_micros（時間戳記）	傳回自 1970-01-01 00：00：00 UTC 以來的微秒數。
unix_millis（時間戳記）	傳回自 1970-01-01 00：00：00 UTC 以來的毫秒數。截斷更高層級的精確度。
unix_seconds（時間戳記）	傳回自 1970-01-01 00：00：00 UTC 以來的秒數。截斷更高層級的精確度。
unix_timestamp(【timeExp【， fmt】】)	傳回目前或指定時間的 UNIX 時間戳記。
weekday（日期）	傳回日期/時間戳記的星期幾 (0 = 星期一，1 = 星期二，...，6 = 星期日）。
weekofyear（日期）	傳回指定日期當年的一週。一週被視為從星期一開始，而第 1 週是 >3 天的第一週。
window(time_column， window_duration【， slide_duration【， start_time】】)	使用時間戳記指定資料欄，將資料列歸納到一或多個時段。視窗開始包含 ，但視窗結束是排他性的，例如 12：05 將位於視窗 【12：05，12：10) 中，但不會在 【12：00，12：05) 中。Windows 可支援微秒精確度。不支援按月順序排列的 Windows。如需詳細說明和範例，請參閱結構化串流指南文件中的「事件時間視窗操作」。

函式	Description
window_time(window_column)	從時間/工作階段視窗資料欄擷取時間值，可用於視窗的事件時間值。擷取的時間是 (window.end - 1)，反映彙總視窗具有專屬上限 - 【開始、結束）的事實。如需詳細說明和範例，請參閱結構化串流指南文件中的「事件時間視窗操作」。
year (日期)	傳回日期/時間戳記的年份元件。

範例

```
-- add_months
SELECT add_months('2016-08-31', 1);
+-------------------------+
|add_months(2016-08-31, 1)|
+-------------------------+
|               2016-09-30|
+-------------------------+
-- convert_timezone
SELECT convert_timezone('Europe/Brussels', 'America/Los_Angeles',
 timestamp_ntz'2021-12-06 00:00:00');
+-------------------------------------------------------------------------------------------
+
|convert_timezone(Europe/Brussels, America/Los_Angeles, TIMESTAMP_NTZ '2021-12-06
 00:00:00')|
+-------------------------------------------------------------------------------------------
+
|                                                                                2021-12-05
 15:00:00|
+-------------------------------------------------------------------------------------------
+
SELECT convert_timezone('Europe/Brussels', timestamp_ntz'2021-12-05 15:00:00');
+-------------------------------------------------------------------------------------------
+
|convert_timezone(current_timezone(), Europe/Brussels, TIMESTAMP_NTZ '2021-12-05
 15:00:00')|
+-------------------------------------------------------------------------------------------
+
```

```
|                                                                                                 2021-12-05
 07:00:00|
+-------------------------------------------------------------------------------------------------------
+
-- curdate
SELECT curdate();
+--------------+
|current_date()|
+--------------+
|    2024-02-24|
+--------------+
-- current_date
SELECT current_date();
+--------------+
|current_date()|
+--------------+
|    2024-02-24|
+--------------+
SELECT current_date;
+--------------+
|current_date()|
+--------------+
|    2024-02-24|
+--------------+
-- current_timestamp
SELECT current_timestamp();
+--------------------+
| current_timestamp()|
+--------------------+
|2024-02-24 16:36:...|
+--------------------+
SELECT current_timestamp;
+--------------------+
| current_timestamp()|
+--------------------+
|2024-02-24 16:36:...|
+--------------------+
-- current_timezone
SELECT current_timezone();
+------------------+
|current_timezone()|
+------------------+
|        Asia/Seoul|
+------------------+
```

```
-- date_add
SELECT date_add('2016-07-30', 1);
+-----------------------+
|date_add(2016-07-30, 1)|
+-----------------------+
|             2016-07-31|
+-----------------------+
-- date_diff
SELECT date_diff('2009-07-31', '2009-07-30');
+---------------------------------+
|date_diff(2009-07-31, 2009-07-30)|
+---------------------------------+
|                                1|
+---------------------------------+
SELECT date_diff('2009-07-30', '2009-07-31');
+---------------------------------+
|date_diff(2009-07-30, 2009-07-31)|
+---------------------------------+
|                               -1|
+---------------------------------+
-- date_format
SELECT date_format('2016-04-08', 'y');
+--------------------------+
|date_format(2016-04-08, y)|
+--------------------------+
|                      2016|
+--------------------------+
-- date_from_unix_date
SELECT date_from_unix_date(1);
+----------------------+
|date_from_unix_date(1)|
+----------------------+
|            1970-01-02|
+----------------------+
-- date_part
SELECT date_part('YEAR', TIMESTAMP '2019-08-12 01:00:00.123456');
+-----------------------------------------------------------+
|date_part(YEAR, TIMESTAMP '2019-08-12 01:00:00.123456')|
+-----------------------------------------------------------+
|                                                       2019|
+-----------------------------------------------------------+
SELECT date_part('week', timestamp'2019-08-12 01:00:00.123456');
+-----------------------------------------------------------+
|date_part(week, TIMESTAMP '2019-08-12 01:00:00.123456')|
```

```
+-----------------------------------------------------+
|                                                   33|
+-----------------------------------------------------+
SELECT date_part('doy', DATE'2019-08-12');
+---------------------------------+
|date_part(doy, DATE '2019-08-12')|
+---------------------------------+
|                              224|
+---------------------------------+
SELECT date_part('SECONDS', timestamp'2019-10-01 00:00:01.000001');
+------------------------------------------------------------+
|date_part(SECONDS, TIMESTAMP '2019-10-01 00:00:01.000001')|
+------------------------------------------------------------+
|                                                    1.000001|
+------------------------------------------------------------+
SELECT date_part('days', interval 5 days 3 hours 7 minutes);
+-------------------------------------------------+
|date_part(days, INTERVAL '5 03:07' DAY TO MINUTE)|
+-------------------------------------------------+
|                                                5|
+-------------------------------------------------+
SELECT date_part('seconds', interval 5 hours 30 seconds 1 milliseconds 1 microseconds);
+-----------------------------------------------------------------+
|date_part(seconds, INTERVAL '05:00:30.001001' HOUR TO SECOND)|
+-----------------------------------------------------------------+
|                                                       30.001001|
+-----------------------------------------------------------------+
SELECT date_part('MONTH', INTERVAL '2021-11' YEAR TO MONTH);
+-------------------------------------------------+
|date_part(MONTH, INTERVAL '2021-11' YEAR TO MONTH)|
+-------------------------------------------------+
|                                               11|
+-------------------------------------------------+
SELECT date_part('MINUTE', INTERVAL '123 23:55:59.002001' DAY TO SECOND);
+-----------------------------------------------------------------+
|date_part(MINUTE, INTERVAL '123 23:55:59.002001' DAY TO SECOND)|
+-----------------------------------------------------------------+
|                                                               55|
+-----------------------------------------------------------------+
-- date_sub
SELECT date_sub('2016-07-30', 1);
+-----------------------+
|date_sub(2016-07-30, 1)|
+-----------------------+
```

```
|             2016-07-29|
+-----------------------+
-- date_trunc
SELECT date_trunc('YEAR', '2015-03-05T09:32:05.359');
+----------------------------------------+
|date_trunc(YEAR, 2015-03-05T09:32:05.359)|
+----------------------------------------+
|                     2015-01-01 00:00:00|
+----------------------------------------+
SELECT date_trunc('MM', '2015-03-05T09:32:05.359');
+---------------------------------------+
|date_trunc(MM, 2015-03-05T09:32:05.359)|
+---------------------------------------+
|                    2015-03-01 00:00:00|
+---------------------------------------+
SELECT date_trunc('DD', '2015-03-05T09:32:05.359');
+---------------------------------------+
|date_trunc(DD, 2015-03-05T09:32:05.359)|
+---------------------------------------+
|                    2015-03-05 00:00:00|
+---------------------------------------+
SELECT date_trunc('HOUR', '2015-03-05T09:32:05.359');
+----------------------------------------+
|date_trunc(HOUR, 2015-03-05T09:32:05.359)|
+----------------------------------------+
|                     2015-03-05 09:00:00|
+----------------------------------------+
SELECT date_trunc('MILLISECOND', '2015-03-05T09:32:05.123456');
+-----------------------------------------------------+
|date_trunc(MILLISECOND, 2015-03-05T09:32:05.123456)|
+-----------------------------------------------------+
|                              2015-03-05 09:32:...|
+-----------------------------------------------------+
-- dateadd
SELECT dateadd('2016-07-30', 1);
+-----------------------+
|date_add(2016-07-30, 1)|
+-----------------------+
|             2016-07-31|
+-----------------------+
-- datediff
SELECT datediff('2009-07-31', '2009-07-30');
+--------------------------------+
|datediff(2009-07-31, 2009-07-30)|
```

```
+--------------------------------+
|                               1|
+--------------------------------+
SELECT datediff('2009-07-30', '2009-07-31');
+--------------------------------+
|datediff(2009-07-30, 2009-07-31)|
+--------------------------------+
|                              -1|
+--------------------------------+
-- datepart
SELECT datepart('YEAR', TIMESTAMP '2019-08-12 01:00:00.123456');
+-----------------------------------------------------------+
|datepart(YEAR FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-----------------------------------------------------------+
|                                                       2019|
+-----------------------------------------------------------+
SELECT datepart('week', timestamp'2019-08-12 01:00:00.123456');
+-----------------------------------------------------------+
|datepart(week FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-----------------------------------------------------------+
|                                                         33|
+-----------------------------------------------------------+
SELECT datepart('doy', DATE'2019-08-12');
+-----------------------------------+
|datepart(doy FROM DATE '2019-08-12')|
+-----------------------------------+
|                                224|
+-----------------------------------+
SELECT datepart('SECONDS', timestamp'2019-10-01 00:00:01.000001');
+------------------------------------------------------------+
|datepart(SECONDS FROM TIMESTAMP '2019-10-01 00:00:01.000001')|
+------------------------------------------------------------+
|                                                    1.000001|
+------------------------------------------------------------+
SELECT datepart('days', interval 5 days 3 hours 7 minutes);
+---------------------------------------------------+
|datepart(days FROM INTERVAL '5 03:07' DAY TO MINUTE)|
+---------------------------------------------------+
|                                                  5|
+---------------------------------------------------+
SELECT datepart('seconds', interval 5 hours 30 seconds 1 milliseconds 1 microseconds);
+-----------------------------------------------------------------+
|datepart(seconds FROM INTERVAL '05:00:30.001001' HOUR TO SECOND)|
+-----------------------------------------------------------------+
```

```
|                                                                 30.001001|
+--------------------------------------------------------------------------+
SELECT datepart('MONTH', INTERVAL '2021-11' YEAR TO MONTH);
+-----------------------------------------------------+
|datepart(MONTH FROM INTERVAL '2021-11' YEAR TO MONTH)|
+-----------------------------------------------------+
|                                                   11|
+-----------------------------------------------------+
SELECT datepart('MINUTE', INTERVAL '123 23:55:59.002001' DAY TO SECOND);
+-------------------------------------------------------------------+
|datepart(MINUTE FROM INTERVAL '123 23:55:59.002001' DAY TO SECOND)|
+-------------------------------------------------------------------+
|                                                                 55|
+-------------------------------------------------------------------+
-- day
SELECT day('2009-07-30');
+---------------+
|day(2009-07-30)|
+---------------+
|             30|
+---------------+
-- dayofmonth
SELECT dayofmonth('2009-07-30');
+----------------------+
|dayofmonth(2009-07-30)|
+----------------------+
|                    30|
+----------------------+
-- dayofweek
SELECT dayofweek('2009-07-30');
+---------------------+
|dayofweek(2009-07-30)|
+---------------------+
|                    5|
+---------------------+
-- dayofyear
SELECT dayofyear('2016-04-09');
+---------------------+
|dayofyear(2016-04-09)|
+---------------------+
|                  100|
+---------------------+
-- extract
SELECT extract(YEAR FROM TIMESTAMP '2019-08-12 01:00:00.123456');
```

```
+-----------------------------------------------------------+
|extract(YEAR FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-----------------------------------------------------------+
|                                                       2019|
+-----------------------------------------------------------+
SELECT extract(week FROM timestamp'2019-08-12 01:00:00.123456');
+-----------------------------------------------------------+
|extract(week FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-----------------------------------------------------------+
|                                                         33|
+-----------------------------------------------------------+
SELECT extract(doy FROM DATE'2019-08-12');
+----------------------------------+
|extract(doy FROM DATE '2019-08-12')|
+----------------------------------+
|                               224|
+----------------------------------+
SELECT extract(SECONDS FROM timestamp'2019-10-01 00:00:01.000001');
+-------------------------------------------------------------+
|extract(SECONDS FROM TIMESTAMP '2019-10-01 00:00:01.000001')|
+-------------------------------------------------------------+
|                                                     1.000001|
+-------------------------------------------------------------+
SELECT extract(days FROM interval 5 days 3 hours 7 minutes);
+---------------------------------------------------+
|extract(days FROM INTERVAL '5 03:07' DAY TO MINUTE)|
+---------------------------------------------------+
|                                                  5|
+---------------------------------------------------+
SELECT extract(seconds FROM interval 5 hours 30 seconds 1 milliseconds 1 microseconds);
+-----------------------------------------------------------------+
|extract(seconds FROM INTERVAL '05:00:30.001001' HOUR TO SECOND)|
+-----------------------------------------------------------------+
|                                                        30.001001|
+-----------------------------------------------------------------+
SELECT extract(MONTH FROM INTERVAL '2021-11' YEAR TO MONTH);
+---------------------------------------------------+
|extract(MONTH FROM INTERVAL '2021-11' YEAR TO MONTH)|
+---------------------------------------------------+
|                                                 11|
+---------------------------------------------------+
SELECT extract(MINUTE FROM INTERVAL '123 23:55:59.002001' DAY TO SECOND);
+-----------------------------------------------------------------+
|extract(MINUTE FROM INTERVAL '123 23:55:59.002001' DAY TO SECOND)|
```

```
+------------------------------------------------------------------+
|                                                              55|
+------------------------------------------------------------------+
-- from_unixtime
SELECT from_unixtime(0, 'yyyy-MM-dd HH:mm:ss');
+-------------------------------------+
|from_unixtime(0, yyyy-MM-dd HH:mm:ss)|
+-------------------------------------+
|                  1970-01-01 09:00:00|
+-------------------------------------+
SELECT from_unixtime(0);
+-------------------------------------+
|from_unixtime(0, yyyy-MM-dd HH:mm:ss)|
+-------------------------------------+
|                  1970-01-01 09:00:00|
+-------------------------------------+
-- from_utc_timestamp
SELECT from_utc_timestamp('2016-08-31', 'Asia/Seoul');
+-----------------------------------------+
|from_utc_timestamp(2016-08-31, Asia/Seoul)|
+-----------------------------------------+
|                      2016-08-31 09:00:00|
+-----------------------------------------+
-- hour
SELECT hour('2009-07-30 12:58:59');
+-------------------------+
|hour(2009-07-30 12:58:59)|
+-------------------------+
|                       12|
+-------------------------+
-- last_day
SELECT last_day('2009-01-12');
+--------------------+
|last_day(2009-01-12)|
+--------------------+
|          2009-01-31|
+--------------------+
-- localtimestamp
SELECT localtimestamp();
+--------------------+
|    localtimestamp()|
+--------------------+
|2024-02-24 16:36:...|
+--------------------+
```

```
-- make_date
SELECT make_date(2013, 7, 15);
+----------------------+
|make_date(2013, 7, 15)|
+----------------------+
|            2013-07-15|
+----------------------+
SELECT make_date(2019, 7, NULL);
+------------------------+
|make_date(2019, 7, NULL)|
+------------------------+
|                    NULL|
+------------------------+
-- make_dt_interval
SELECT make_dt_interval(1, 12, 30, 01.001001);
+-------------------------------------+
|make_dt_interval(1, 12, 30, 1.001001)|
+-------------------------------------+
|                 INTERVAL '1 12:30...|
+-------------------------------------+
SELECT make_dt_interval(2);
+----------------------------------+
|make_dt_interval(2, 0, 0, 0.000000)|
+----------------------------------+
|               INTERVAL '2 00:00...|
+----------------------------------+
SELECT make_dt_interval(100, null, 3);
+--------------------------------------+
|make_dt_interval(100, NULL, 3, 0.000000)|
+--------------------------------------+
|                                  NULL|
+--------------------------------------+
-- make_interval
SELECT make_interval(100, 11, 1, 1, 12, 30, 01.001001);
+-----------------------------------------------+
|make_interval(100, 11, 1, 1, 12, 30, 1.001001)|
+-----------------------------------------------+
|                          100 years 11 mont...|
+-----------------------------------------------+
SELECT make_interval(100, null, 3);
+-----------------------------------------------+
|make_interval(100, NULL, 3, 0, 0, 0, 0.000000)|
+-----------------------------------------------+
|                                           NULL|
```

```
+-----------------------------------------------+
SELECT make_interval(0, 1, 0, 1, 0, 0, 100.000001);
+-----------------------------------------+
|make_interval(0, 1, 0, 1, 0, 0, 100.000001)|
+-----------------------------------------+
|                      1 months 1 days 1...|
+-----------------------------------------+
-- make_timestamp
SELECT make_timestamp(2014, 12, 28, 6, 30, 45.887);
+----------------------------------------+
|make_timestamp(2014, 12, 28, 6, 30, 45.887)|
+----------------------------------------+
|                     2014-12-28 06:30:...|
+----------------------------------------+
SELECT make_timestamp(2014, 12, 28, 6, 30, 45.887, 'CET');
+-----------------------------------------------+
|make_timestamp(2014, 12, 28, 6, 30, 45.887, CET)|
+-----------------------------------------------+
|                           2014-12-28 14:30:...|
+-----------------------------------------------+
SELECT make_timestamp(2019, 6, 30, 23, 59, 60);
+--------------------------------------+
|make_timestamp(2019, 6, 30, 23, 59, 60)|
+--------------------------------------+
|                    2019-07-01 00:00:00|
+--------------------------------------+
SELECT make_timestamp(2019, 6, 30, 23, 59, 1);
+-------------------------------------+
|make_timestamp(2019, 6, 30, 23, 59, 1)|
+-------------------------------------+
|                   2019-06-30 23:59:01|
+-------------------------------------+
SELECT make_timestamp(null, 7, 22, 15, 30, 0);
+-------------------------------------+
|make_timestamp(NULL, 7, 22, 15, 30, 0)|
+-------------------------------------+
|                                 NULL|
+-------------------------------------+
-- make_timestamp_ltz
SELECT make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887);
+-----------------------------------------------+
|make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887)|
+-----------------------------------------------+
|                           2014-12-28 06:30:...|
```

```
+----------------------------------------------+
SELECT make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887, 'CET');
+-----------------------------------------------------+
|make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887, CET)|
+-----------------------------------------------------+
|                                2014-12-28 14:30:...|
+-----------------------------------------------------+
SELECT make_timestamp_ltz(2019, 6, 30, 23, 59, 60);
+-----------------------------------------+
|make_timestamp_ltz(2019, 6, 30, 23, 59, 60)|
+-----------------------------------------+
|                       2019-07-01 00:00:00|
+-----------------------------------------+
SELECT make_timestamp_ltz(null, 7, 22, 15, 30, 0);
+-----------------------------------------+
|make_timestamp_ltz(NULL, 7, 22, 15, 30, 0)|
+-----------------------------------------+
|                                     NULL|
+-----------------------------------------+
-- make_timestamp_ntz
SELECT make_timestamp_ntz(2014, 12, 28, 6, 30, 45.887);
+-----------------------------------------------+
|make_timestamp_ntz(2014, 12, 28, 6, 30, 45.887)|
+-----------------------------------------------+
|                          2014-12-28 06:30:...|
+-----------------------------------------------+
SELECT make_timestamp_ntz(2019, 6, 30, 23, 59, 60);
+-----------------------------------------+
|make_timestamp_ntz(2019, 6, 30, 23, 59, 60)|
+-----------------------------------------+
|                       2019-07-01 00:00:00|
+-----------------------------------------+
SELECT make_timestamp_ntz(null, 7, 22, 15, 30, 0);
+-----------------------------------------+
|make_timestamp_ntz(NULL, 7, 22, 15, 30, 0)|
+-----------------------------------------+
|                                     NULL|
+-----------------------------------------+
-- make_ym_interval
SELECT make_ym_interval(1, 2);
+----------------------+
|make_ym_interval(1, 2)|
+----------------------+
|  INTERVAL '1-2' YE...|
```

```
+----------------------+
SELECT make_ym_interval(1, 0);
+----------------------+
|make_ym_interval(1, 0)|
+----------------------+
|  INTERVAL '1-0' YE...|
+----------------------+
SELECT make_ym_interval(-1, 1);
+-----------------------+
|make_ym_interval(-1, 1)|
+-----------------------+
|   INTERVAL '-0-11' ...|
+-----------------------+
SELECT make_ym_interval(2);
+----------------------+
|make_ym_interval(2, 0)|
+----------------------+
|  INTERVAL '2-0' YE...|
+----------------------+
-- minute
SELECT minute('2009-07-30 12:58:59');
+---------------------------+
|minute(2009-07-30 12:58:59)|
+---------------------------+
|                         58|
+---------------------------+
-- month
SELECT month('2016-07-30');
+-----------------+
|month(2016-07-30)|
+-----------------+
|                7|
+-----------------+
-- months_between
SELECT months_between('1997-02-28 10:30:00', '1996-10-30');
+---------------------------------------------------------+
|months_between(1997-02-28 10:30:00, 1996-10-30, true)|
+---------------------------------------------------------+
|                                           3.94959677|
+---------------------------------------------------------+
SELECT months_between('1997-02-28 10:30:00', '1996-10-30', false);
+----------------------------------------------------------+
|months_between(1997-02-28 10:30:00, 1996-10-30, false)|
+----------------------------------------------------------+
```

```
|                                  3.9495967741935485|
+--------------------------------------------------------+
-- next_day
SELECT next_day('2015-01-14', 'TU');
+-------------------------+
|next_day(2015-01-14, TU)|
+-------------------------+
|              2015-01-20|
+-------------------------+
-- now
SELECT now();
+--------------------+
|               now()|
+--------------------+
|2024-02-24 16:36:...|
+--------------------+
-- quarter
SELECT quarter('2016-08-31');
+-------------------+
|quarter(2016-08-31)|
+-------------------+
|                  3|
+-------------------+
-- second
SELECT second('2009-07-30 12:58:59');
+---------------------------+
|second(2009-07-30 12:58:59)|
+---------------------------+
|                         59|
+---------------------------+
-- session_window
SELECT a, session_window.start, session_window.end, count(*) as cnt FROM VALUES ('A1',
 '2021-01-01 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:10:00'),
 ('A2', '2021-01-01 00:01:00') AS tab(a, b) GROUP by a, session_window(b, '5 minutes')
 ORDER BY a, start;
+---+-------------------+-------------------+---+
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
| A1|2021-01-01 00:00:00|2021-01-01 00:09:30|  2|
| A1|2021-01-01 00:10:00|2021-01-01 00:15:00|  1|
| A2|2021-01-01 00:01:00|2021-01-01 00:06:00|  1|
+---+-------------------+-------------------+---+
SELECT a, session_window.start, session_window.end, count(*) as cnt FROM VALUES ('A1',
 '2021-01-01 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:10:00'),
```

```
 ('A2', '2021-01-01 00:01:00'), ('A2', '2021-01-01 00:04:30') AS tab(a, b) GROUP by a,
 session_window(b, CASE WHEN a = 'A1' THEN '5 minutes' WHEN a = 'A2' THEN '1 minute'
 ELSE '10 minutes' END) ORDER BY a, start;
+---+-------------------+-------------------+---+
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
| A1|2021-01-01 00:00:00|2021-01-01 00:09:30|  2|
| A1|2021-01-01 00:10:00|2021-01-01 00:15:00|  1|
| A2|2021-01-01 00:01:00|2021-01-01 00:02:00|  1|
| A2|2021-01-01 00:04:30|2021-01-01 00:05:30|  1|
+---+-------------------+-------------------+---+
-- timestamp_micros
SELECT timestamp_micros(1230219000123123);
+----------------------------------+
|timestamp_micros(1230219000123123)|
+----------------------------------+
|              2008-12-26 00:30:...|
+----------------------------------+
-- timestamp_millis
SELECT timestamp_millis(1230219000123);
+-------------------------------+
|timestamp_millis(1230219000123)|
+-------------------------------+
|           2008-12-26 00:30:...|
+-------------------------------+
-- timestamp_seconds
SELECT timestamp_seconds(1230219000);
+-----------------------------+
|timestamp_seconds(1230219000)|
+-----------------------------+
|          2008-12-26 00:30:00|
+-----------------------------+
SELECT timestamp_seconds(1230219000.123);
+---------------------------------+
|timestamp_seconds(1230219000.123)|
+---------------------------------+
|             2008-12-26 00:30:...|
+---------------------------------+
-- to_date
SELECT to_date('2009-07-30 04:17:52');
+---------------------------+
|to_date(2009-07-30 04:17:52)|
+---------------------------+
|                 2009-07-30|
```

```
+----------------------------+
SELECT to_date('2016-12-31', 'yyyy-MM-dd');
+-------------------------------+
|to_date(2016-12-31, yyyy-MM-dd)|
+-------------------------------+
|                     2016-12-31|
+-------------------------------+
-- to_timestamp
SELECT to_timestamp('2016-12-31 00:12:00');
+---------------------------------+
|to_timestamp(2016-12-31 00:12:00)|
+---------------------------------+
|              2016-12-31 00:12:00|
+---------------------------------+
SELECT to_timestamp('2016-12-31', 'yyyy-MM-dd');
+-----------------------------------+
|to_timestamp(2016-12-31, yyyy-MM-dd)|
+-----------------------------------+
|                 2016-12-31 00:00:00|
+-----------------------------------+
-- to_timestamp_ltz
SELECT to_timestamp_ltz('2016-12-31 00:12:00');
+-------------------------------------+
|to_timestamp_ltz(2016-12-31 00:12:00)|
+-------------------------------------+
|                  2016-12-31 00:12:00|
+-------------------------------------+
SELECT to_timestamp_ltz('2016-12-31', 'yyyy-MM-dd');
+----------------------------------------+
|to_timestamp_ltz(2016-12-31, yyyy-MM-dd)|
+----------------------------------------+
|                     2016-12-31 00:00:00|
+----------------------------------------+
-- to_timestamp_ntz
SELECT to_timestamp_ntz('2016-12-31 00:12:00');
+-------------------------------------+
|to_timestamp_ntz(2016-12-31 00:12:00)|
+-------------------------------------+
|                  2016-12-31 00:12:00|
+-------------------------------------+
SELECT to_timestamp_ntz('2016-12-31', 'yyyy-MM-dd');
+----------------------------------------+
|to_timestamp_ntz(2016-12-31, yyyy-MM-dd)|
+----------------------------------------+
```

```
|                     2016-12-31 00:00:00|
+----------------------------------------+
-- to_unix_timestamp
SELECT to_unix_timestamp('2016-04-08', 'yyyy-MM-dd');
+-----------------------------------------+
|to_unix_timestamp(2016-04-08, yyyy-MM-dd)|
+-----------------------------------------+
|                                1460041200|
+-----------------------------------------+
-- to_utc_timestamp
SELECT to_utc_timestamp('2016-08-31', 'Asia/Seoul');
+-----------------------------------------+
|to_utc_timestamp(2016-08-31, Asia/Seoul)|
+-----------------------------------------+
|                     2016-08-30 15:00:00|
+-----------------------------------------+
-- trunc
SELECT trunc('2019-08-04', 'week');
+-----------------------+
|trunc(2019-08-04, week)|
+-----------------------+
|             2019-07-29|
+-----------------------+
SELECT trunc('2019-08-04', 'quarter');
+-------------------------+
|trunc(2019-08-04, quarter)|
+-------------------------+
|                2019-07-01|
+-------------------------+
SELECT trunc('2009-02-12', 'MM');
+---------------------+
|trunc(2009-02-12, MM)|
+---------------------+
|           2009-02-01|
+---------------------+
SELECT trunc('2015-10-27', 'YEAR');
+-----------------------+
|trunc(2015-10-27, YEAR)|
+-----------------------+
|             2015-01-01|
+-----------------------+
-- try_to_timestamp
SELECT try_to_timestamp('2016-12-31 00:12:00');
+--------------------------------------+
```

```
|try_to_timestamp(2016-12-31 00:12:00)|
+-------------------------------------+
|                  2016-12-31 00:12:00|
+-------------------------------------+
SELECT try_to_timestamp('2016-12-31', 'yyyy-MM-dd');
+---------------------------------------+
|try_to_timestamp(2016-12-31, yyyy-MM-dd)|
+---------------------------------------+
|                     2016-12-31 00:00:00|
+---------------------------------------+
SELECT try_to_timestamp('foo', 'yyyy-MM-dd');
+---------------------------------+
|try_to_timestamp(foo, yyyy-MM-dd)|
+---------------------------------+
|                             NULL|
+---------------------------------+
-- unix_date
SELECT unix_date(DATE("1970-01-02"));
+---------------------+
|unix_date(1970-01-02)|
+---------------------+
|                    1|
+---------------------+
-- unix_micros
SELECT unix_micros(TIMESTAMP('1970-01-01 00:00:01Z'));
+---------------------------------+
|unix_micros(1970-01-01 00:00:01Z)|
+---------------------------------+
|                          1000000|
+---------------------------------+
-- unix_millis
SELECT unix_millis(TIMESTAMP('1970-01-01 00:00:01Z'));
+---------------------------------+
|unix_millis(1970-01-01 00:00:01Z)|
+---------------------------------+
|                             1000|
+---------------------------------+
-- unix_seconds
SELECT unix_seconds(TIMESTAMP('1970-01-01 00:00:01Z'));
+---------------------------------+
|unix_seconds(1970-01-01 00:00:01Z)|
+---------------------------------+
|                                1|
+---------------------------------+
```

```
-- unix_timestamp
SELECT unix_timestamp();
+-----------------------------------------------------+
|unix_timestamp(current_timestamp(), yyyy-MM-dd HH:mm:ss)|
+-----------------------------------------------------+
|                                           1708760216|
+-----------------------------------------------------+
SELECT unix_timestamp('2016-04-08', 'yyyy-MM-dd');
+--------------------------------------+
|unix_timestamp(2016-04-08, yyyy-MM-dd)|
+--------------------------------------+
|                            1460041200|
+--------------------------------------+
-- weekday
SELECT weekday('2009-07-30');
+------------------+
|weekday(2009-07-30)|
+------------------+
|                 3|
+------------------+
-- weekofyear
SELECT weekofyear('2008-02-20');
+----------------------+
|weekofyear(2008-02-20)|
+----------------------+
|                     8|
+----------------------+
-- window
SELECT a, window.start, window.end, count(*) as cnt FROM VALUES ('A1', '2021-01-01
 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:06:00'), ('A2',
 '2021-01-01 00:01:00') AS tab(a, b) GROUP by a, window(b, '5 minutes') ORDER BY a,
 start;
+---+-------------------+-------------------+---+
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
| A1|2021-01-01 00:00:00|2021-01-01 00:05:00|  2|
| A1|2021-01-01 00:05:00|2021-01-01 00:10:00|  1|
| A2|2021-01-01 00:00:00|2021-01-01 00:05:00|  1|
+---+-------------------+-------------------+---+
SELECT a, window.start, window.end, count(*) as cnt FROM VALUES ('A1', '2021-01-01
 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:06:00'), ('A2',
 '2021-01-01 00:01:00') AS tab(a, b) GROUP by a, window(b, '10 minutes', '5 minutes')
 ORDER BY a, start;
+---+-------------------+-------------------+---+
```

```
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
| A1|2020-12-31 23:55:00|2021-01-01 00:05:00|  2|
| A1|2021-01-01 00:00:00|2021-01-01 00:10:00|  3|
| A1|2021-01-01 00:05:00|2021-01-01 00:15:00|  1|
| A2|2020-12-31 23:55:00|2021-01-01 00:05:00|  1|
| A2|2021-01-01 00:00:00|2021-01-01 00:10:00|  1|
+---+-------------------+-------------------+---+
-- window_time
SELECT a, window.start as start, window.end as end, window_time(window), cnt FROM
 (SELECT a, window, count(*) as cnt FROM VALUES ('A1', '2021-01-01 00:00:00'), ('A1',
 '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:06:00'), ('A2', '2021-01-01 00:01:00')
 AS tab(a, b) GROUP by a, window(b, '5 minutes') ORDER BY a, window.start);
+---+-------------------+-------------------+--------------------+---+
|  a|              start|                end| window_time(window)|cnt|
+---+-------------------+-------------------+--------------------+---+
| A1|2021-01-01 00:00:00|2021-01-01 00:05:00|2021-01-01 00:04:...|  2|
| A1|2021-01-01 00:05:00|2021-01-01 00:10:00|2021-01-01 00:09:...|  1|
| A2|2021-01-01 00:00:00|2021-01-01 00:05:00|2021-01-01 00:04:...|  1|
+---+-------------------+-------------------+--------------------+---+
-- year
SELECT year('2016-07-30');
+----------------+
|year(2016-07-30)|
+----------------+
|            2016|
+----------------+
```

彙總函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

彙總函數會跨資料列操作值，以執行數學計算，例如總和、平均值、計數、最小值/最大值、標準差和估算，以及一些非數學操作。

語法

```
aggregate_function(input1 [, input2, ...]) FILTER (WHERE boolean_expression)
```

參數

- boolean_expression - 指定任何評估結果類型布林值的表達式。兩個或多個表達式可以使用邏輯運算子 (AND、OR) 結合在一起。

排序集彙總函數

這些彙總函數使用與其他彙總函數不同的語法，以便指定要排序值的表達式 （通常是資料欄名稱）。

語法

```
{ PERCENTILE_CONT | PERCENTILE_DISC }(percentile) WITHIN GROUP (ORDER BY
 { order_by_expression [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [ , ... ] }) FILTER
 (WHERE boolean_expression)
```

參數

- percentile - 您要尋找的值百分位數。百分位數必須是介於 0.0 和 1.0 之間的常數。
- order_by_expression - 運算式 （通常是資料欄名稱），用於在彙總之前排序值。
- boolean_expression - 指定任何評估結果類型布林值的表達式。兩個或多個表達式可以使用邏輯運算子 (AND、OR) 結合在一起。

範例

```
CREATE OR REPLACE TEMPORARY VIEW basic_pays AS SELECT * FROM VALUES
('Jane Doe','Accounting',8435),
('Akua Mansa','Accounting',9998),
('John Doe','Accounting',8992),
('Juan Li','Accounting',8870),
('Carlos Salazar','Accounting',11472),
('Arnav Desai','Accounting',6627),
('Saanvi Sarkar','IT',8113),
('Shirley Rodriguez','IT',5186),
('Nikki Wolf','Sales',9181),
('Alejandro Rosalez','Sales',9441),
('Nikhil Jayashankar','Sales',6660),
('Richard Roe','Sales',10563),
('Pat Candella','SCM',10449),
('Gerard Hernandez','SCM',6949),
('Pamela Castillo','SCM',11303),
('Paulo Santos','SCM',11798),
```

```
('Jorge Souza','SCM',10586)
AS basic_pays(employee_name, department, salary);
SELECT * FROM basic_pays;
+------------------+----------+------+
|    employee_name  |department|salary|
+------------------+----------+------+
| Arnav Desai       |Accounting|  6627|
| Jorge Souza       |       SCM| 10586|
| Jane Doe          |Accounting|  8435|
| Nikhil Jayashankar|     Sales|  6660|
| Diego Vanauf      |     Sales| 10563|
| Carlos Salazar    |Accounting| 11472|
| Gerard Hernandez  |       SCM|  6949|
| John Doe          |Accounting|  8992|
| Nikki Wolf        |     Sales|  9181|
| Paulo Santos      |       SCM| 11798|
| Saanvi Sarkar     |        IT|  8113|
| Shirley Rodriguez |        IT|  5186|
| Pat Candella      |       SCM| 10449|
| Akua Mansa        |Accounting|  9998|
| Pamela Castillo   |       SCM| 11303|
| Alejandro Rosalez |     Sales|  9441|
| Juan Li           |Accounting|  8870|
+------------------+----------+------+
SELECT
department,
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary) AS pc1,
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary) FILTER (WHERE employee_name LIKE
 '%Bo%') AS pc2,
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary DESC) AS pc3,
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary DESC) FILTER (WHERE employee_name
 LIKE '%Bo%') AS pc4,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary) AS pd1,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary) FILTER (WHERE employee_name LIKE
 '%Bo%') AS pd2,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary DESC) AS pd3,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary DESC) FILTER (WHERE employee_name
 LIKE '%Bo%') AS pd4
FROM basic_pays
GROUP BY department
ORDER BY department;
+----------+-------+--------+-------+--------+-----+-----+-----+-----+
|department|    pc1|     pc2|    pc3|     pc4|  pd1|  pd2|  pd3|  pd4|
+----------+-------+--------+-------+--------+-----+-----+-----+-----+
```

```
|Accounting|8543.75| 7838.25| 9746.5|10260.75| 8435| 6627| 9998|11472|
|        IT|5917.75|    NULL|7381.25|    NULL| 5186| NULL| 8113| NULL|
|     Sales|8550.75|    NULL| 9721.5|    NULL| 6660| NULL|10563| NULL|
|       SCM|10449.0|10786.25|11303.0|11460.75|10449|10449|11303|11798|
+----------+-------+--------+-------+--------+-----+-----+-----+-----+
```

條件函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
coalesce(expr1、expr2、...)	如果存在，則傳回第一個非 Null 引數。否則為 null。
if(expr1、expr2、expr3)	如果 expr1評估為 true，則傳回 expr2；否則傳回 expr3。
ifnull(expr1， expr2)	expr2 如果 expr1為 null expr1或其他，則傳回 。
nanvl(expr1， expr2)	expr1 如果不是 NaN，expr2否則傳回 。
nullif(expr1， expr2)	如果 expr1 等於 expr2，expr1否則傳回 null。
nvl(expr1， expr2)	expr2 如果 expr1為 null expr1或其他，則傳回 。
nvl2(expr1、expr2、expr3)	expr2 如果 expr1 不是 null，則傳回 ，expr3否則傳回 。
expr1 和 expr2 【當 expr3 和 expr4】* 【ELSE expr5】 結束時的案例	當 expr1 = true 時， 會傳回 expr2；否則 expr3 = true 時， 會傳回 expr4；否則 會傳回 expr5。

範例

```
-- coalesce
SELECT coalesce(NULL, 1, NULL);
+----------------------+
|coalesce(NULL, 1, NULL)|
+----------------------+
|                     1|
+----------------------+
-- if
SELECT if(1 < 2, 'a', 'b');
+-------------------+
|(IF((1 < 2), a, b))|
+-------------------+
|                  a|
+-------------------+
-- ifnull
SELECT ifnull(NULL, array('2'));
+----------------------+
|ifnull(NULL, array(2))|
+----------------------+
|                   [2]|
+----------------------+
-- nanvl
SELECT nanvl(cast('NaN' as double), 123);
+-------------------------------+
|nanvl(CAST(NaN AS DOUBLE), 123)|
+-------------------------------+
|                          123.0|
+-------------------------------+
-- nullif
SELECT nullif(2, 2);
+------------+
|nullif(2, 2)|
+------------+
|        NULL|
+------------+
-- nvl
SELECT nvl(NULL, array('2'));
+-------------------+
|nvl(NULL, array(2))|
+-------------------+
|                [2]|
+-------------------+
```

```
-- nvl2
SELECT nvl2(NULL, 2, 1);
+----------------+
|nvl2(NULL, 2, 1)|
+----------------+
|               1|
+----------------+
-- when
SELECT CASE WHEN 1 > 0 THEN 1 WHEN 2 > 0 THEN 2.0 ELSE 1.2 END;
+-------------------------------------------------------------+
|CASE WHEN (1 > 0) THEN 1 WHEN (2 > 0) THEN 2.0 ELSE 1.2 END|
+-------------------------------------------------------------+
|                                                          1.0|
+-------------------------------------------------------------+
SELECT CASE WHEN 1 < 0 THEN 1 WHEN 2 > 0 THEN 2.0 ELSE 1.2 END;
+-------------------------------------------------------------+
|CASE WHEN (1 < 0) THEN 1 WHEN (2 > 0) THEN 2.0 ELSE 1.2 END|
+-------------------------------------------------------------+
|                                                          2.0|
+-------------------------------------------------------------+
SELECT CASE WHEN 1 < 0 THEN 1 WHEN 2 < 0 THEN 2.0 END;
+-------------------------------------------------+
|CASE WHEN (1 < 0) THEN 1 WHEN (2 < 0) THEN 2.0 END|
+-------------------------------------------------+
|                                             NULL|
+-------------------------------------------------+
```

JSON 函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
from_json (jsonStr ,	傳回具有指定 `jsonStr` 和 `schema` 的結構值。

函式	Description
schema【，options】)	
get_json_object(json_txt， 路徑)	從 `path` 擷取 json 物件。
json_array_length(jsonArray)	傳回最外部 JSON 陣列中的元素數目。
json_object_keys(json_object)	傳回最外部 JSON 物件的所有索引鍵做為陣列。
json_tuple(jsonStr ， p1 ， p2 ， ... ， pn)	傳回類似函數 get_json_object 的元組，但它需要多個名稱。所有輸入參數和輸出資料欄類型都是字串。
schema_of_json(json【，選項】)	以 JSON 字串的 DDL 格式傳回結構描述。
to_json(expr【，options】)	傳回具有指定結構值的 JSON 字串

範例

```
-- from_json
SELECT from_json('{"a":1, "b":0.8}', 'a INT, b DOUBLE');
+---------------------------+
| from_json({"a":1, "b":0.8}) |
+---------------------------+
```

```
| {1, 0.8}                  |
+---------------------------+

SELECT from_json('{"time":"26/08/2015"}', 'time Timestamp', map('timestampFormat', 'dd/
MM/yyyy'));
+--------------------------------+
| from_json({"time":"26/08/2015"}) |
+--------------------------------+
| {2015-08-26 00:00...           |
+--------------------------------+

SELECT from_json('{"teacher": "Alice", "student": [{"name": "Bob", "rank": 1}, {"name":
 "Charlie", "rank": 2}]}', 'STRUCT<teacher: STRING, student: ARRAY<STRUCT<name: STRING,
 rank: INT>>>');
+-------------------------------------------------------------------------------------------------------
+
| from_json({"teacher": "Alice", "student": [{"name": "Bob", "rank": 1}, {"name":
 "Charlie", "rank": 2}]}) |
+-------------------------------------------------------------------------------------------------------
+
| {Alice, [{Bob, 1}...
                  |
+-------------------------------------------------------------------------------------------------------
+

-- get_json_object
SELECT get_json_object('{"a":"b"}', '$.a');
+-------------------------------+
| get_json_object({"a":"b"}, $.a) |
+-------------------------------+
| b                             |
+-------------------------------+

-- json_array_length
SELECT json_array_length('[1,2,3,4]');
+----------------------------+
| json_array_length([1,2,3,4]) |
+----------------------------+
| 4                          |
+----------------------------+

SELECT json_array_length('[1,2,3,{"f1":1,"f2":[5,6]},4]');
+-----------------------------------------------+
| json_array_length([1,2,3,{"f1":1,"f2":[5,6]},4]) |
```

```
+------------------------------------------------+
| 5                                              |
+------------------------------------------------+

SELECT json_array_length('[1,2');
+----------------------+
| json_array_length([1,2) |
+----------------------+
| NULL                 |
+----------------------+

-- json_object_keys
SELECT json_object_keys('{}');
+--------------------+
| json_object_keys({}) |
+--------------------+
| []                 |
+--------------------+

SELECT json_object_keys('{"key": "value"}');
+---------------------------------+
| json_object_keys({"key": "value"}) |
+---------------------------------+
| [key]                           |
+---------------------------------+

SELECT json_object_keys('{"f1":"abc","f2":{"f3":"a", "f4":"b"}}');
+-----------------------------------------------------------+
| json_object_keys({"f1":"abc","f2":{"f3":"a", "f4":"b"}}) |
+-----------------------------------------------------------+
| [f1, f2]                                                  |
+-----------------------------------------------------------+

-- json_tuple
SELECT json_tuple('{"a":1, "b":2}', 'a', 'b');
+---+---+
| c0| c1|
+---+---+
|  1|  2|
+---+---+

-- schema_of_json
SELECT schema_of_json('[{"col":0}]');
+--------------------------+
```

```
| schema_of_json([{"col":0}]) |
+-----------------------------+
| ARRAY<STRUCT<col:...      |
+-----------------------------+

SELECT schema_of_json('[{"col":01}]', map('allowNumericLeadingZeros', 'true'));
+-----------------------------+
| schema_of_json([{"col":01}]) |
+-----------------------------+
| ARRAY<STRUCT<col:...       |
+-----------------------------+

-- to_json
SELECT to_json(named_struct('a', 1, 'b', 2));
+---------------------------------+
| to_json(named_struct(a, 1, b, 2)) |
+---------------------------------+
| {"a":1,"b":2}                   |
+---------------------------------+

SELECT to_json(named_struct('time', to_timestamp('2015-08-26', 'yyyy-MM-dd')),
 map('timestampFormat', 'dd/MM/yyyy'));
+-------------------------------------------------------------------+
| to_json(named_struct(time, to_timestamp(2015-08-26, yyyy-MM-dd))) |
+-------------------------------------------------------------------+
| {"time":"26/08/20...                                              |
+-------------------------------------------------------------------+

SELECT to_json(array(named_struct('a', 1, 'b', 2)));
+---------------------------------------+
| to_json(array(named_struct(a, 1, b, 2))) |
+---------------------------------------+
| [{"a":1,"b":2}]                       |
+---------------------------------------+

SELECT to_json(map('a', named_struct('b', 1)));
+-----------------------------------+
| to_json(map(a, named_struct(b, 1))) |
+-----------------------------------+
| {"a":{"b":1}}                     |
+-----------------------------------+

SELECT to_json(map(named_struct('a', 1),named_struct('b', 2)));
+---------------------------------------------------+
```

```
| to_json(map(named_struct(a, 1), named_struct(b, 2))) |
+------------------------------------------------------+
| {"[1]":{"b":2}}                                      |
+------------------------------------------------------+

SELECT to_json(map('a', 1));
+------------------+
| to_json(map(a, 1)) |
+------------------+
| {"a":1}          |
+------------------+

SELECT to_json(array(map('a', 1)));
+-------------------------+
| to_json(array(map(a, 1))) |
+-------------------------+
| [{"a":1}]               |
+-------------------------+
```

陣列函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
array(expr， ...)	傳回具有指定元素的陣列。
array_append(array， 元素)	在傳遞為第一個引數的陣列結尾新增 元素。元素類型應類似於陣列的元素類型。Null 元素也會附加到陣列中。但是，如果傳遞陣列，則 NULL 輸出為 NULL
array_compact(array)	從陣列中移除 null 值。
array_contains (陣列，值)	如果陣列包含 值，則傳回 true。
array_distinct(array)	從陣列中移除重複的值。

函式	Description
array_except(array1， array2)	傳回 array1 中的元素陣列，但不傳回 array2 中的元素陣列，無需重複。
array_insert(x， pos， val)	將 val 放入陣列 x 的索引位置。陣列索引從 1 開始。最大負索引為 -1，函數會在目前的最後一個元素之後插入新元素。高於陣列大小的索引會附加陣列，或者如果索引為負值，則為陣列前面加上 'null' 元素。
array_intersect(array1， array2)	傳回 array1 和 array2 交集的元素陣列，無需重複。
array_join(array， delimiter【， nullReplacement】)	使用分隔符號和選用字串來串連指定陣列的元素，以取代 null。如果未設定 nullReplacement 的值，則會篩選任何 null 值。
array_max(array)	傳回陣列中的最大值。NaN 大於雙/浮點數類型的任何非 NaN 元素。略過 NULL 元素。
array_min(array)	傳回陣列中的最小值。NaN 大於雙/浮點數類型的任何非 NaN 元素。略過 NULL 元素。
array_position(array， 元素)	傳回陣列中第一個相符元素的 (1 型) 索引長度，如果找不到相符項目，則傳回 0。
array_prepend(array， 元素)	在傳遞為第一個引數的陣列開頭新增 元素。元素類型應與陣列的元素類型相同。Null 元素也會在陣列前面。但是，如果傳遞的陣列是 NULL 輸出是 NULL
array_remove(array， 元素)	從陣列中移除所有等於 元素的元素。
array_repeat(element， count)	傳回包含元素計數時間的陣列。
array_union(array1， array2)	傳回 array1 和 array2 聯集的元素陣列，無需重複。

函式	Description
arrays_overlap(a1， a2)	如果 a1 至少包含 a2 中也存在的非空元素，則傳回 true。如果陣列沒有常見元素，而且兩者都是非空的，且其中之一包含 null 元素，則會傳回 false。
arrays_zip(a1、a2、...)	傳回合併的結構陣列，其中 N 個結構包含輸入陣列的所有 N 個值。
flatten(arrayOfArrays)	將陣列轉換為單一陣列。
get（陣列，索引）	傳回指定 (0 型） 索引的陣列元素。如果索引指向陣列邊界之外，則此函數會傳回 NULL。
sequence（開始、停止、步驟）	產生從開始到停止（包含） 的元素陣列，依步驟遞增。傳回的元素類型與引數表達式類型相同。支援的類型為：位元組、短、整數、長、日期、時間戳記。開始和停止表達式必須解析為相同類型。如果開始和停止表達式解析為 'date' 或 'timestamp' 類型，則步驟表達式必須解析為 'interval' 或 'year-month interval' 或 'day-time interval' 類型，否則解析為與開始和停止表達式相同的類型。
shuffle(array)	傳回指定陣列的隨機排列。
slice(x， start， length)	子集陣列 x 從索引開頭開始（陣列索引從 1 開始，如果開頭為負值，則從結尾開始），並指定長度。
sort_array(array【， ascendingOrder】)	根據陣列元素的自然順序，以遞增或遞減順序排序輸入陣列。NaN 大於雙/浮點數類型的任何非 NaN 元素。Null 元素將以遞增順序放置在傳回陣列的開頭，或以遞減順序放置在傳回陣列的結尾。

範例

```
-- array
SELECT array(1, 2, 3);
+--------------+
|array(1, 2, 3)|
+--------------+
|     [1, 2, 3]|
+--------------+
-- array_append
SELECT array_append(array('b', 'd', 'c', 'a'), 'd');
+----------------------------------+
|array_append(array(b, d, c, a), d)|
+----------------------------------+
|                    [b, d, c, a, d]|
+----------------------------------+
SELECT array_append(array(1, 2, 3, null), null);
+---------------------------------------+
|array_append(array(1, 2, 3, NULL), NULL)|
+---------------------------------------+
|                     [1, 2, 3, NULL, N...|
+---------------------------------------+
SELECT array_append(CAST(null as Array<Int>), 2);
+---------------------+
|array_append(NULL, 2)|
+---------------------+
|                 NULL|
+---------------------+
-- array_compact
SELECT array_compact(array(1, 2, 3, null));
+----------------------------------+
|array_compact(array(1, 2, 3, NULL))|
+----------------------------------+
|                          [1, 2, 3]|
+----------------------------------+
SELECT array_compact(array("a", "b", "c"));
+----------------------------+
|array_compact(array(a, b, c))|
+----------------------------+
|                     [a, b, c]|
+----------------------------+
-- array_contains
SELECT array_contains(array(1, 2, 3), 2);
+---------------------------------+
|array_contains(array(1, 2, 3), 2)|
```

```
+--------------------------------+
|                            true|
+--------------------------------+
-- array_distinct
SELECT array_distinct(array(1, 2, 3, null, 3));
+--------------------------------------+
|array_distinct(array(1, 2, 3, NULL, 3))|
+--------------------------------------+
|                       [1, 2, 3, NULL]|
+--------------------------------------+
-- array_except
SELECT array_except(array(1, 2, 3), array(1, 3, 5));
+-------------------------------------------+
|array_except(array(1, 2, 3), array(1, 3, 5))|
+-------------------------------------------+
|                                        [2]|
+-------------------------------------------+
-- array_insert
SELECT array_insert(array(1, 2, 3, 4), 5, 5);
+-----------------------------------+
|array_insert(array(1, 2, 3, 4), 5, 5)|
+-----------------------------------+
|                     [1, 2, 3, 4, 5]|
+-----------------------------------+
SELECT array_insert(array(5, 4, 3, 2), -1, 1);
+-----------------------------------+
|array_insert(array(5, 4, 3, 2), -1, 1)|
+-----------------------------------+
|                      [5, 4, 3, 2, 1]|
+-----------------------------------+
SELECT array_insert(array(5, 3, 2, 1), -4, 4);
+-----------------------------------+
|array_insert(array(5, 3, 2, 1), -4, 4)|
+-----------------------------------+
|                      [5, 4, 3, 2, 1]|
+-----------------------------------+
-- array_intersect
SELECT array_intersect(array(1, 2, 3), array(1, 3, 5));
+-----------------------------------------------+
|array_intersect(array(1, 2, 3), array(1, 3, 5))|
+-----------------------------------------------+
|                                         [1, 3]|
+-----------------------------------------------+
-- array_join
```

```
SELECT array_join(array('hello', 'world'), ' ');
+---------------------------------+
|array_join(array(hello, world),  )|
+---------------------------------+
|                      hello world|
+---------------------------------+
SELECT array_join(array('hello', null ,'world'), ' ');
+--------------------------------------+
|array_join(array(hello, NULL, world),  )|
+--------------------------------------+
|                             hello world|
+--------------------------------------+
SELECT array_join(array('hello', null ,'world'), ' ', ',');
+-----------------------------------------+
|array_join(array(hello, NULL, world),  , ,)|
+-----------------------------------------+
|                              hello , world|
+-----------------------------------------+
-- array_max
SELECT array_max(array(1, 20, null, 3));
+--------------------------------+
|array_max(array(1, 20, NULL, 3))|
+--------------------------------+
|                              20|
+--------------------------------+
-- array_min
SELECT array_min(array(1, 20, null, 3));
+--------------------------------+
|array_min(array(1, 20, NULL, 3))|
+--------------------------------+
|                               1|
+--------------------------------+
-- array_position
SELECT array_position(array(312, 773, 708, 708), 708);
+-----------------------------------------------+
|array_position(array(312, 773, 708, 708), 708)|
+-----------------------------------------------+
|                                              3|
+-----------------------------------------------+
SELECT array_position(array(312, 773, 708, 708), 414);
+-----------------------------------------------+
|array_position(array(312, 773, 708, 708), 414)|
+-----------------------------------------------+
|                                              0|
```

```
+-----------------------------------------------+
-- array_prepend
SELECT array_prepend(array('b', 'd', 'c', 'a'), 'd');
+----------------------------------+
|array_prepend(array(b, d, c, a), d)|
+----------------------------------+
|                    [d, b, d, c, a]|
+----------------------------------+
SELECT array_prepend(array(1, 2, 3, null), null);
+---------------------------------------+
|array_prepend(array(1, 2, 3, NULL), NULL)|
+---------------------------------------+
|                     [NULL, 1, 2, 3, N...|
+---------------------------------------+
SELECT array_prepend(CAST(null as Array<Int>), 2);
+----------------------+
|array_prepend(NULL, 2)|
+----------------------+
|                  NULL|
+----------------------+
-- array_remove
SELECT array_remove(array(1, 2, 3, null, 3), 3);
+---------------------------------------+
|array_remove(array(1, 2, 3, NULL, 3), 3)|
+---------------------------------------+
|                            [1, 2, NULL]|
+---------------------------------------+
-- array_repeat
SELECT array_repeat('123', 2);
+--------------------+
|array_repeat(123, 2)|
+--------------------+
|          [123, 123]|
+--------------------+
-- array_union
SELECT array_union(array(1, 2, 3), array(1, 3, 5));
+-----------------------------------------+
|array_union(array(1, 2, 3), array(1, 3, 5))|
+-----------------------------------------+
|                               [1, 2, 3, 5]|
+-----------------------------------------+
-- arrays_overlap
SELECT arrays_overlap(array(1, 2, 3), array(3, 4, 5));
+-------------------------------------------+
```

```
|arrays_overlap(array(1, 2, 3), array(3, 4, 5))|
+----------------------------------------------+
|                                          true|
+----------------------------------------------+
-- arrays_zip
SELECT arrays_zip(array(1, 2, 3), array(2, 3, 4));
+-----------------------------------------+
|arrays_zip(array(1, 2, 3), array(2, 3, 4))|
+-----------------------------------------+
|                      [{1, 2}, {2, 3}, ...|
+-----------------------------------------+
SELECT arrays_zip(array(1, 2), array(2, 3), array(3, 4));
+-------------------------------------------------+
|arrays_zip(array(1, 2), array(2, 3), array(3, 4))|
+-------------------------------------------------+
|                             [{1, 2, 3}, {2, 3...|
+-------------------------------------------------+
-- flatten
SELECT flatten(array(array(1, 2), array(3, 4)));
+---------------------------------------+
|flatten(array(array(1, 2), array(3, 4)))|
+---------------------------------------+
|                           [1, 2, 3, 4]|
+---------------------------------------+
-- get
SELECT get(array(1, 2, 3), 0);
+----------------------+
|get(array(1, 2, 3), 0)|
+----------------------+
|                     1|
+----------------------+
SELECT get(array(1, 2, 3), 3);
+----------------------+
|get(array(1, 2, 3), 3)|
+----------------------+
|                  NULL|
+----------------------+
SELECT get(array(1, 2, 3), -1);
+-----------------------+
|get(array(1, 2, 3), -1)|
+-----------------------+
|                   NULL|
+-----------------------+
-- sequence
```

```
SELECT sequence(1, 5);
+---------------+
| sequence(1, 5)|
+---------------+
|[1, 2, 3, 4, 5]|
+---------------+
SELECT sequence(5, 1);
+---------------+
| sequence(5, 1)|
+---------------+
|[5, 4, 3, 2, 1]|
+---------------+
SELECT sequence(to_date('2018-01-01'), to_date('2018-03-01'), interval 1 month);
+------------------------------------------------------------------------+
|sequence(to_date(2018-01-01), to_date(2018-03-01), INTERVAL '1' MONTH)|
+------------------------------------------------------------------------+
|                                                    [2018-01-01, 2018...|
+------------------------------------------------------------------------+
SELECT sequence(to_date('2018-01-01'), to_date('2018-03-01'), interval '0-1' year to
 month);
+-----------------------------------------------------------------------------------+
|sequence(to_date(2018-01-01), to_date(2018-03-01), INTERVAL '0-1' YEAR TO MONTH)|
+-----------------------------------------------------------------------------------+
|                                                                [2018-01-01, 2018...|
+-----------------------------------------------------------------------------------+
-- shuffle
SELECT shuffle(array(1, 20, 3, 5));
+---------------------------+
|shuffle(array(1, 20, 3, 5))|
+---------------------------+
|              [5, 1, 20, 3]|
+---------------------------+
SELECT shuffle(array(1, 20, null, 3));
+------------------------------+
|shuffle(array(1, 20, NULL, 3))|
+------------------------------+
|              [1, NULL, 20, 3]|
+------------------------------+
-- slice
SELECT slice(array(1, 2, 3, 4), 2, 2);
+------------------------------+
|slice(array(1, 2, 3, 4), 2, 2)|
+------------------------------+
|                        [2, 3]|
```

```
+------------------------------+
SELECT slice(array(1, 2, 3, 4), -2, 2);
+------------------------------+
|slice(array(1, 2, 3, 4), -2, 2)|
+------------------------------+
|                        [3, 4]|
+------------------------------+
-- sort_array
SELECT sort_array(array('b', 'd', null, 'c', 'a'), true);
+-----------------------------------------+
|sort_array(array(b, d, NULL, c, a), true)|
+-----------------------------------------+
|                      [NULL, a, b, c, d]|
+-----------------------------------------+
```

範圍函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

視窗函數在一組資料列上運作，稱為視窗，並根據資料列群組計算每一列的傳回值。視窗函數適用於處理任務，例如計算移動平均值、計算累積統計資料，或存取指定目前資料列相對位置的資料列值。

語法

```
window_function [ nulls_option ] OVER ( [ { PARTITION | DISTRIBUTE } BY
 partition_col_name = partition_col_val ( [ , ... ] ) ] { ORDER | SORT } BY expression
 [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [ , ... ] [ window_frame ] )
```

參數

- 排名函數

  語法: `RANK | DENSE_RANK | PERCENT_RANK | NTILE | ROW_NUMBER`

  分析函數

  語法: `CUME_DIST | LAG | LEAD | NTH_VALUE | FIRST_VALUE | LAST_VALUE`

彙總函數

語法: MAX | MIN | COUNT | SUM | AVG | ...

- nulls_option - 指定在評估視窗函數時是否略過 null 值。RESPECT NULLS 表示不略過 null 值，而 IGNORE NULLS 表示略過。如果未指定，則預設為 RESPECT NULLS。

  語法: { IGNORE | RESPECT } NULLS

  注意： Only LAG | LEAD | NTH_VALUE | FIRST_VALUE | LAST_VALUE 可與 搭配使用IGNORE NULLS。

- window_frame - 指定要啟動視窗的列，以及結束視窗的位置。

  語法: { RANGE | ROWS } { frame_start | BETWEEN frame_start AND frame_end }

  frame_start 和 frame_end 具有下列語法：

  語法: UNBOUNDED PRECEDING | offset PRECEDING | CURRENT ROW | offset FOLLOWING | UNBOUNDED FOLLOWING

  位移：指定與目前資料列位置的位移。

  注意 如果省略 frame_end，則預設為 CURRENT ROW。

範例

```
CREATE TABLE employees (name STRING, dept STRING, salary INT, age INT);
INSERT INTO employees VALUES ("Lisa", "Sales", 10000, 35);
INSERT INTO employees VALUES ("Evan", "Sales", 32000, 38);
INSERT INTO employees VALUES ("Fred", "Engineering", 21000, 28);
INSERT INTO employees VALUES ("Alex", "Sales", 30000, 33);
INSERT INTO employees VALUES ("Tom", "Engineering", 23000, 33);
INSERT INTO employees VALUES ("Jane", "Marketing", 29000, 28);
INSERT INTO employees VALUES ("Jeff", "Marketing", 35000, 38);
INSERT INTO employees VALUES ("Paul", "Engineering", 29000, 23);
INSERT INTO employees VALUES ("Chloe", "Engineering", 23000, 25);
SELECT * FROM employees;
+-----+-----------+------+-----+
| name|       dept|salary|  age|
+-----+-----------+------+-----+
|Chloe|Engineering| 23000|   25|
```

```
| Fred|Engineering| 21000|   28|
| Paul|Engineering| 29000|   23|
|Helen|  Marketing| 29000|   40|
|  Tom|Engineering| 23000|   33|
| Jane|  Marketing| 29000|   28|
| Jeff|  Marketing| 35000|   38|
| Evan|      Sales| 32000|   38|
| Lisa|      Sales| 10000|   35|
| Alex|      Sales| 30000|   33|
+-----+-----------+------+-----+
SELECT name, dept, salary, RANK() OVER (PARTITION BY dept ORDER BY salary) AS rank FROM
 employees;
+-----+-----------+------+----+
| name|       dept|salary|rank|
+-----+-----------+------+----+
| Lisa|      Sales| 10000|   1|
| Alex|      Sales| 30000|   2|
| Evan|      Sales| 32000|   3|
| Fred|Engineering| 21000|   1|
|  Tom|Engineering| 23000|   2|
|Chloe|Engineering| 23000|   2|
| Paul|Engineering| 29000|   4|
|Helen|  Marketing| 29000|   1|
| Jane|  Marketing| 29000|   1|
| Jeff|  Marketing| 35000|   3|
+-----+-----------+------+----+
SELECT name, dept, salary, DENSE_RANK() OVER (PARTITION BY dept ORDER BY salary ROWS
 BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS dense_rank FROM employees;
+-----+-----------+------+----------+
| name|       dept|salary|dense_rank|
+-----+-----------+------+----------+
| Lisa|      Sales| 10000|         1|
| Alex|      Sales| 30000|         2|
| Evan|      Sales| 32000|         3|
| Fred|Engineering| 21000|         1|
|  Tom|Engineering| 23000|         2|
|Chloe|Engineering| 23000|         2|
| Paul|Engineering| 29000|         3|
|Helen|  Marketing| 29000|         1|
| Jane|  Marketing| 29000|         1|
| Jeff|  Marketing| 35000|         2|
+-----+-----------+------+----------+
SELECT name, dept, age, CUME_DIST() OVER (PARTITION BY dept ORDER BY age
```

```
RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cume_dist FROM employees;
+-----+-----------+------+------------------+
| name|       dept|age   |         cume_dist|
+-----+-----------+------+------------------+
| Alex|      Sales|    33|0.3333333333333333|
| Lisa|      Sales|    35|0.6666666666666666|
| Evan|      Sales|    38|               1.0|
| Paul|Engineering|    23|              0.25|
|Chloe|Engineering|    25|              0.75|
| Fred|Engineering|    28|              0.25|
|  Tom|Engineering|    33|               1.0|
| Jane|  Marketing|    28|0.3333333333333333|
| Jeff|  Marketing|    38|0.6666666666666666|
|Helen|  Marketing|    40|               1.0|
+-----+-----------+------+------------------+
SELECT name, dept, salary, MIN(salary) OVER (PARTITION BY dept ORDER BY salary) AS min
FROM employees;
+-----+-----------+------+-----+
| name|       dept|salary|  min|
+-----+-----------+------+-----+
| Lisa|      Sales| 10000|10000|
| Alex|      Sales| 30000|10000|
| Evan|      Sales| 32000|10000|
|Helen|  Marketing| 29000|29000|
| Jane|  Marketing| 29000|29000|
| Jeff|  Marketing| 35000|29000|
| Fred|Engineering| 21000|21000|
|  Tom|Engineering| 23000|21000|
|Chloe|Engineering| 23000|21000|
| Paul|Engineering| 29000|21000|
+-----+-----------+------+-----+
SELECT name, salary,
LAG(salary) OVER (PARTITION BY dept ORDER BY salary) AS lag,
LEAD(salary, 1, 0) OVER (PARTITION BY dept ORDER BY salary) AS lead
FROM employees;
+-----+-----------+------+-----+-----+
| name|       dept|salary|  lag| lead|
+-----+-----------+------+-----+-----+
| Lisa|      Sales| 10000|NULL |30000|
| Alex|      Sales| 30000|10000|32000|
| Evan|      Sales| 32000|30000|    0|
| Fred|Engineering| 21000| NULL|23000|
|Chloe|Engineering| 23000|21000|23000|
|  Tom|Engineering| 23000|23000|29000|
```

```
| Paul|Engineering| 29000|23000|    0|
|Helen|  Marketing| 29000| NULL|29000|
| Jane|  Marketing| 29000|29000|35000|
| Jeff|  Marketing| 35000|29000|    0|
+-----+-----------+------+-----+-----+
SELECT id, v,
LEAD(v, 0) IGNORE NULLS OVER w lead,
LAG(v, 0) IGNORE NULLS OVER w lag,
NTH_VALUE(v, 2) IGNORE NULLS OVER w nth_value,
FIRST_VALUE(v) IGNORE NULLS OVER w first_value,
LAST_VALUE(v) IGNORE NULLS OVER w last_value
FROM test_ignore_null
WINDOW w AS (ORDER BY id)
ORDER BY id;
+--+----+----+----+---------+-----------+----------+
|id|   v|lead| lag|nth_value|first_value|last_value|
+--+----+----+----+---------+-----------+----------+
| 0|NULL|NULL|NULL|     NULL|       NULL|      NULL|
| 1|   x|   x|   x|     NULL|          x|         x|
| 2|NULL|NULL|NULL|     NULL|          x|         x|
| 3|NULL|NULL|NULL|     NULL|          x|         x|
| 4|   y|   y|   y|        y|          x|         y|
| 5|NULL|NULL|NULL|        y|          x|         y|
| 6|   z|   z|   z|        y|          x|         z|
| 7|   v|   v|   v|        y|          x|         v|
| 8|NULL|NULL|NULL|        y|          x|         v|
+--+----+----+----+---------+-----------+----------+
```

轉換函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
bigint(expr)	將值 `expr` 轉換為目標資料類型 `bigint`。
binary(expr)	將值 `expr` 轉換為目標資料類型 `binary`。

函式	Description
布林值 (expr)	將值 `expr` 轉換為目標資料類型 `boolean`。
cast(expr AS 類型)	將值 `expr` 轉換為目標資料類型 `type`。
date(expr)	將值 `expr` 轉換為目標資料類型 `date`。
decimal(expr)	將值 `expr` 轉換為目標資料類型 `decimal`。
double(expr)	將值 `expr` 轉換為目標資料類型 `double`。
float(expr)	將值 `expr` 轉換為目標資料類型 `float`。
int(expr)	將值 `expr` 轉換為目標資料類型 `int`。
smallint(expr)	將值 `expr` 轉換為目標資料類型 `smallint`。
string(expr)	將值 `expr` 轉換為目標資料類型 `string`。
timestamp(expr)	將值 `expr` 轉換為目標資料類型 `timestamp`。
tinyint(expr)	將值 `expr` 轉換為目標資料類型 `tinyint`。

範例

```
-- cast
SELECT cast(field as int);
+---------------+
|CAST(field AS INT)|
+---------------+
|             10|
+---------------+
```

述詞函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
！ expr	邏輯非。
expr1 < expr2	如果 `expr1` 小於 `expr2`，則傳回 true。
expr1 <= expr2	如果 `expr1` 小於或等於 `expr2`，則傳回 true。
expr1 <=> expr2	傳回與非 Null 運算元的 EQUAL(=) 運算子相同的結果，但如果兩者皆為 Null，則傳回 true，如果其中之一為 Null，則傳回 false。
expr1 = expr2	如果 `expr1` 等於 `expr2`，否則傳回 true。
expr1 == expr2	如果 `expr1` 等於 `expr2`，否則傳回 true。
expr1 > expr2	如果 `expr1` 大於 `expr2`，則傳回 true。
expr1 >= expr2	如果 `expr1` 大於或等於 `expr2`，則傳回 true。
expr1 和 expr2	邏輯 AND。
str ilike 模式【 ESCAPE 逸出】	如果 str 符合 `pattern` 與 `escape` 不區分大小寫，則傳回 true；如果任何引數為 null，則傳回 null；否則傳回 false。
expr1 in(expr2、expr3、...)	如果 `expr` 等於任何 valN，則傳回 true。
isnan(expr)	如果 `expr` 是 NaN，則傳回 true，否則傳回 false。
isnotnull(expr)	如果 `expr` 不是 null，則傳回 true，否則傳回 false。
isnull(expr)	如果 `expr` 為 null，則傳回 true，否則傳回 false。
str like 模式【 ESCAPE 逸出】	如果 str 符合 `pattern` 與 `escape`，則傳回 true；如果任何引數為 null，則傳回 null；否則傳回 false。
非 expr	邏輯非。
expr1 或 expr2	邏輯 OR。
regexp(str， regexp)	如果 `str` 符合 `regexp`，否則傳回 true。

函式	Description
regexp_like(str， regexp)	如果 `str` 符合 `regexp`，否則傳回 true。
rlike(str， regexp)	如果 `str` 符合 `regexp`，否則傳回 true。

範例

```
-- !
SELECT ! true;
+----------+
|(NOT true)|
+----------+
|     false|
+----------+
SELECT ! false;
+-----------+
|(NOT false)|
+-----------+
|       true|
+-----------+
SELECT ! NULL;
+----------+
|(NOT NULL)|
+----------+
|      NULL|
+----------+
-- <
SELECT to_date('2009-07-30 04:17:52') < to_date('2009-07-30 04:17:52');
+-------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) < to_date(2009-07-30 04:17:52))|
+-------------------------------------------------------------+
|                                                        false|
+-------------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') < to_date('2009-08-01 04:17:52');
+-------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) < to_date(2009-08-01 04:17:52))|
+-------------------------------------------------------------+
|                                                         true|
+-------------------------------------------------------------+
SELECT 1 < NULL;
+----------+
```

```
|(1 < NULL)|
+----------+
|      NULL|
+----------+
-- <=
SELECT 2 <= 2;
+--------+
|(2 <= 2)|
+--------+
|    true|
+--------+
SELECT 1.0 <= '1';
+----------+
|(1.0 <= 1)|
+----------+
|      true|
+----------+
SELECT to_date('2009-07-30 04:17:52') <= to_date('2009-07-30 04:17:52');
+------------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) <= to_date(2009-07-30 04:17:52))|
+------------------------------------------------------------------+
|                                                              true|
+------------------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') <= to_date('2009-08-01 04:17:52');
+------------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) <= to_date(2009-08-01 04:17:52))|
+------------------------------------------------------------------+
|                                                              true|
+------------------------------------------------------------------+
SELECT 1 <= NULL;
+-----------+
|(1 <= NULL)|
+-----------+
|       NULL|
+-----------+
-- <=>
SELECT 2 <=> 2;
+---------+
|(2 <=> 2)|
+---------+
|     true|
+---------+
SELECT 1 <=> '1';
+---------+
```

```
|(1 <=> 1)|
+---------+
|     true|
+---------+
SELECT true <=> NULL;
+---------------+
|(true <=> NULL)|
+---------------+
|          false|
+---------------+
SELECT NULL <=> NULL;
+---------------+
|(NULL <=> NULL)|
+---------------+
|           true|
+---------------+
-- =
SELECT 2 = 2;
+-------+
|(2 = 2)|
+-------+
|   true|
+-------+
SELECT 1 = '1';
+-------+
|(1 = 1)|
+-------+
|   true|
+-------+
SELECT true = NULL;
+-------------+
|(true = NULL)|
+-------------+
|         NULL|
+-------------+
SELECT NULL = NULL;
+-------------+
|(NULL = NULL)|
+-------------+
|         NULL|
+-------------+
-- ==
SELECT 2 == 2;
+-------+
```

```
|(2 = 2)|
+-------+
|   true|
+-------+
SELECT 1 == '1';
+-------+
|(1 = 1)|
+-------+
|   true|
+-------+
SELECT true == NULL;
+-------------+
|(true = NULL)|
+-------------+
|         NULL|
+-------------+
SELECT NULL == NULL;
+-------------+
|(NULL = NULL)|
+-------------+
|         NULL|
+-------------+
-- >
SELECT 2 > 1;
+-------+
|(2 > 1)|
+-------+
|   true|
+-------+
SELECT 2 > 1.1;
+-------+
|(2 > 1)|
+-------+
|   true|
+-------+
SELECT to_date('2009-07-30 04:17:52') > to_date('2009-07-30 04:17:52');
+-----------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) > to_date(2009-07-30 04:17:52))|
+-----------------------------------------------------------+
|                                                      false|
+-----------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') > to_date('2009-08-01 04:17:52');
+-----------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) > to_date(2009-08-01 04:17:52))|
```

```
+-------------------------------------------------------------+
|                                                        false|
+-------------------------------------------------------------+
SELECT 1 > NULL;
+----------+
|(1 > NULL)|
+----------+
|      NULL|
+----------+
-- >=
SELECT 2 >= 1;
+--------+
|(2 >= 1)|
+--------+
|    true|
+--------+
SELECT 2.0 >= '2.1';
+------------+
|(2.0 >= 2.1)|
+------------+
|       false|
+------------+
SELECT to_date('2009-07-30 04:17:52') >= to_date('2009-07-30 04:17:52');
+-------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) >= to_date(2009-07-30 04:17:52))|
+-------------------------------------------------------------+
|                                                         true|
+-------------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') >= to_date('2009-08-01 04:17:52');
+-------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) >= to_date(2009-08-01 04:17:52))|
+-------------------------------------------------------------+
|                                                        false|
+-------------------------------------------------------------+
SELECT 1 >= NULL;
+-----------+
|(1 >= NULL)|
+-----------+
|       NULL|
+-----------+
-- and
SELECT true and true;
+---------------+
|(true AND true)|
```

```
+---------------+
|           true|
+---------------+
SELECT true and false;
+----------------+
|(true AND false)|
+----------------+
|           false|
+----------------+
SELECT true and NULL;
+---------------+
|(true AND NULL)|
+---------------+
|           NULL|
+---------------+
SELECT false and NULL;
+----------------+
|(false AND NULL)|
+----------------+
|           false|
+----------------+
-- ilike
SELECT ilike('Wagon', '_Agon');
+-------------------+
|ilike(Wagon, _Agon)|
+-------------------+
|               true|
+-------------------+
SELECT '%SystemDrive%\Users\John' ilike '\%SystemDrive\%\\users%';
+-----------------------------------------------------------+
|ilike(%SystemDrive%\Users\John, \%SystemDrive\%\\users%)|
+-----------------------------------------------------------+
|                                                       true|
+-----------------------------------------------------------+
SELECT '%SystemDrive%\\USERS\\John' ilike '\%SystemDrive\%\\\\Users%';
+-----------------------------------------------------------+
|ilike(%SystemDrive%\USERS\John, \%SystemDrive\%\\Users%)|
+-----------------------------------------------------------+
|                                                       true|
+-----------------------------------------------------------+
SELECT '%SystemDrive%/Users/John' ilike '/%SYSTEMDrive/%//Users%' ESCAPE '/';
+-----------------------------------------------------------+
|ilike(%SystemDrive%/Users/John, /%SYSTEMDrive/%//Users%)|
+-----------------------------------------------------------+
```

```
|                                                
```

```
SELECT isnull(1);
+-----------+
|(1 IS NULL)|
+-----------+
|      false|
+-----------+
-- like
SELECT like('Wagon', '_Agon');
+----------------+
|Wagon LIKE _Agon|
+----------------+
|            true|
+----------------+
-- not
SELECT not true;
+----------+
|(NOT true)|
+----------+
|     false|
+----------+
SELECT not false;
+-----------+
|(NOT false)|
+-----------+
|       true|
+-----------+
SELECT not NULL;
+----------+
|(NOT NULL)|
+----------+
|      NULL|
+----------+
-- or
SELECT true or false;
+---------------+
|(true OR false)|
+---------------+
|           true|
+---------------+
SELECT false or false;
+----------------+
|(false OR false)|
+----------------+
|           false|
```

```
+----------------+
SELECT true or NULL;
+--------------+
|(true OR NULL)|
+--------------+
|          true|
+--------------+
SELECT false or NULL;
+---------------+
|(false OR NULL)|
+---------------+
|           NULL|
+---------------+
```

映射函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
element_at(array，索引）	傳回指定 (1 型） 索引的陣列元素。
element_at（映射，索引鍵）	傳回指定金鑰的值。如果地圖中不包含金鑰，則函數會傳回 NULL。
map(key0， value0， key1， value1， ...)	使用指定的鍵/值對建立映射。
map_concat(map， ...)	傳回所有指定映射的聯集
map_contains_key（地圖，金鑰）	如果映射包含 金鑰，則傳回 true。
map_entries(map)	傳回指定映射中所有項目的未排序陣列。
map_from_arrays（索引鍵、值）	使用一對指定的索引鍵/值陣列建立映射。金鑰中的所有元素不應為 null
map_from_entries(arrayOfEntries)	傳回從指定項目陣列建立的映射。

函式	Description
map_keys（地圖）	傳回未排序的陣列，其中包含映射的索引鍵。
map_values(map)	傳回未排序的陣列，其中包含映射的值。
str_to_map(text【， pairDelim【， keyValueDelim】】)	使用分隔符號將文字分割為鍵/值對後建立映射。`pairDelim` 的預設分隔符號為 ' , ' ，`keyValueDelim` 的預設分隔符號為 ' : '。`pairDelim` 和 `keyValueDelim` 都視為規則表達式。
try_element_at(array，索引）	傳回指定 (1 型） 索引的陣列元素。如果索引為 0，系統會擲回錯誤。如果索引 < 0， 會存取從最後一個到第一個的元素。如果索引超過陣列的長度，函數一律會傳回 NULL。
try_element_at(map， key)	傳回指定金鑰的值。如果映射中不包含金鑰，則函數一律會傳回 NULL。

範例

```
-- element_at
SELECT element_at(array(1, 2, 3), 2);
+----------------------------+
|element_at(array(1, 2, 3), 2)|
+----------------------------+
|                           2|
+----------------------------+
SELECT element_at(map(1, 'a', 2, 'b'), 2);
+-----------------------------+
|element_at(map(1, a, 2, b), 2)|
+-----------------------------+
|                            b|
+-----------------------------+
-- map
SELECT map(1.0, '2', 3.0, '4');
+--------------------+
| map(1.0, 2, 3.0, 4)|
+--------------------+
|{1.0 -> 2, 3.0 -> 4}|
```

```
+--------------------+
-- map_concat
SELECT map_concat(map(1, 'a', 2, 'b'), map(3, 'c'));
+-------------------------------------+
|map_concat(map(1, a, 2, b), map(3, c))|
+-------------------------------------+
|                  {1 -> a, 2 -> b, ...|
+-------------------------------------+
-- map_contains_key
SELECT map_contains_key(map(1, 'a', 2, 'b'), 1);
+----------------------------------+
|map_contains_key(map(1, a, 2, b), 1)|
+----------------------------------+
|                              true|
+----------------------------------+
SELECT map_contains_key(map(1, 'a', 2, 'b'), 3);
+----------------------------------+
|map_contains_key(map(1, a, 2, b), 3)|
+----------------------------------+
|                             false|
+----------------------------------+
-- map_entries
SELECT map_entries(map(1, 'a', 2, 'b'));
+---------------------------+
|map_entries(map(1, a, 2, b))|
+---------------------------+
|            [{1, a}, {2, b}]|
+---------------------------+
-- map_from_arrays
SELECT map_from_arrays(array(1.0, 3.0), array('2', '4'));
+-----------------------------------------+
|map_from_arrays(array(1.0, 3.0), array(2, 4))|
+-----------------------------------------+
|                        {1.0 -> 2, 3.0 -> 4}|
+-----------------------------------------+
-- map_from_entries
SELECT map_from_entries(array(struct(1, 'a'), struct(2, 'b')));
+-----------------------------------------------------+
|map_from_entries(array(struct(1, a), struct(2, b)))|
+-----------------------------------------------------+
|                                  {1 -> a, 2 -> b}|
+-----------------------------------------------------+
-- map_keys
SELECT map_keys(map(1, 'a', 2, 'b'));
```

```
+-------------------------+
|map_keys(map(1, a, 2, b))|
+-------------------------+
|                   [1, 2]|
+-------------------------+
-- map_values
SELECT map_values(map(1, 'a', 2, 'b'));
+---------------------------+
|map_values(map(1, a, 2, b))|
+---------------------------+
|                     [a, b]|
+---------------------------+
-- str_to_map
SELECT str_to_map('a:1,b:2,c:3', ',', ':');
+-----------------------------+
|str_to_map(a:1,b:2,c:3, ,, :)|
+-----------------------------+
|         {a -> 1, b -> 2, ...|
+-----------------------------+
SELECT str_to_map('a');
+-------------------+
|str_to_map(a, ,, :)|
+-------------------+
|        {a -> NULL}|
+-------------------+
-- try_element_at
SELECT try_element_at(array(1, 2, 3), 2);
+---------------------------------+
|try_element_at(array(1, 2, 3), 2)|
+---------------------------------+
|                                2|
+---------------------------------+
SELECT try_element_at(map(1, 'a', 2, 'b'), 2);
+---------------------------------+
|try_element_at(map(1, a, 2, b), 2)|
+---------------------------------+
|                                 b|
+---------------------------------+
```

數學函式

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

函式	Description
expr1 % expr2	傳回 `expr1`/`expr2` 之後的其餘部分。
expr1 * expr2	傳回 `expr1`*`expr2`。
expr1 + expr2	傳回 `expr1`+`expr2`。
expr1 - expr2	傳回 `expr1`-`expr2`。
expr1 / expr2	傳回 `expr1`/`expr2`。它一律會執行浮點分割。
abs(expr)	傳回數值或間隔值的絕對值。
acos(expr)	傳回 `expr` 的反餘弦 (a.k.a. arc cosine)，就像由 `java.lang.Math.acos` 運算一樣。
acosh(expr)	傳回 `expr` 的反雙曲餘弦。
asin(expr)	傳回 `expr` 的反正弦 (a.k.a. arc 正弦) arc sin，就像由 `java.lang.Math.asin` 計算一樣。
asinh(expr)	傳回 `expr` 的反雙曲正弦。
atan(expr)	傳回 `expr` 的反正切 (a.k.a. arc tangent)，就像由 `java.lang.Math.atan` 運算一樣
atan2(exprY， exprX)	傳回平面正 x 軸與座標 (`exprX`、`exprY`) 給定點之間的弧度角度，就像由 `java.lang.Math.atan2` 計算一樣。
atanh(expr)	傳回 `expr` 的反雙曲切線。

函式	Description
bin(expr)	傳回以二進位表示的長值 `expr` 的字串表示。
bround(expr， d)	使用 HALF_EVEN 四捨五入模式，傳回四捨五入至小數位數 `expr`。
cbrt(expr)	傳回 `expr` 的立方體根。
ceil(expr【， scale】)	四捨五入後傳回不小於 `expr` 的最小數字。您可以指定選用的 `scale` 參數來控制四捨五入行為。
ceiling(expr【， scale】)	四捨五入後傳回不小於 `expr` 的最小數字。您可以指定選用的 `scale` 參數來控制四捨五入行為。
conv(num， from_base， to_base)	將 `num` 從 `from_base` 轉換為 `to_base`。
cos(expr)	傳回 `expr` 的餘弦，就像由 `java.lang.Math.cos` 運算一樣。
cosh(expr)	傳回 `expr` 的雙曲餘弦，就像由 `java.lang.Math.cosh` 計算一樣。
cot(expr)	傳回 `expr` 的餘切，就像由 `1/java.lang.Math.tan` 計算一樣。
csc(expr)	傳回 `expr` 的餘數，就像由 `1/java.lang.Math.sin` 計算一樣。
degrees(expr)	將弧度轉換為度數。
expr1 div expr2	將 `expr1` 除以 `expr2`。如果運算元為 NULL 或 `expr2` 為 0，則傳回 NULL。結果會轉換為長。
e()	傳回 Euler 的數字 e。
exp(expr)	將 e 傳回 `expr` 的強大功能。

函式	Description
expm1(expr) - 傳回 exp(`expr`)	1
factorial(expr)	傳回 `expr` 的因數。`expr` 為 【0..20】。否則為 null。
floor(expr【， scale】)	四捨五入後傳回不超過 `expr` 的最大數字。您可以指定選用的 `scale` 參數來控制四捨五入行為。
greatest(expr， ...)	傳回所有參數的最大值，略過 null 值。
hex(expr)	將 `expr` 轉換為十六進位。
hypot(expr1， expr2)	傳回 sqrt(`expr1`**2 + `expr2`**2)。
least(expr， ...)	傳回所有參數的最小值，略過 null 值。
ln(expr)	傳回 `expr` 的自然對數（基礎 e)。
log(base， expr)	以 `base` 傳回 `expr` 的對數。
log10(expr)	傳回具有 base 10 之 `expr` 的對數。
log1p(expr)	傳回 log(1 + `expr`)。
log2(expr)	傳回具有 base 2 之 `expr` 的對數。
expr1 mod expr2	傳回 `expr1`/`expr2` 之後的其餘部分。
negative(expr)	傳回 `expr` 的否定值。
pi()	傳回 pi。
pmod(expr1， expr2)	傳回 `expr1` mod `expr2` 的正值。
positive(expr)	傳回 `expr` 的值。
pow(expr1， expr2)	將 `expr1` 提升為 `expr2` 的強大功能。
power(expr1， expr2)	將 `expr1` 提升為 `expr2` 的強大功能。

函式	Description
radians(expr)	將度數轉換為弧度。
rand(【seed】)	在 【0， 1) 中傳回具有獨立且相同分佈 (i.i.d.) 均勻分佈值的隨機值。
randn(【seed】)	傳回隨機值，其中包含從標準常態分佈繪製的獨立且相同分佈 (i.i.d.) 值。
random(【seed】)	在 【0， 1) 中傳回具有獨立且相同分佈 (i.i.d.) 均勻分佈值的隨機值。
rint(expr)	傳回與引數值最接近且等於數學整數的雙值。
round(expr， d)	使用 HALF_UP 四捨五入模式，傳回四捨五入至小數位數 `expr`。
sec(expr)	傳回 `expr` 的正負數，就像由 `1/java.lang.Math.cos` 計算一樣。
shiftleft (基礎、expr)	位元向左移位。
sign(expr)	傳回 -1.0、0.0 或 1.0，因為 `expr` 為負數、0 或正數。
signum(expr)	傳回 -1.0、0.0 或 1.0，因為 `expr` 為負數、0 或正數。
sin(expr)	傳回 `expr` 的正弦，就像由 `java.lang.Math.sin` 計算一樣。
sinh(expr)	傳回 `expr` 的雙曲正弦，就像由 `java.lang.Math.sinh` 運算一樣。
sqrt(expr)	傳回 `expr` 的平方根。
tan(expr)	傳回 `expr` 的正切，就像由 `java.lang.Math.tan` 計算一樣。

函式	Description
tanh(expr)	傳回 `expr` 的雙曲正切，就像由 `java.lang.Math.tanh` 計算一樣。
try_add(expr1， expr2)	傳回 `expr1` 和 `expr2` 的總和，且溢位時的結果為 null。可接受的輸入類型與 `+` 運算子相同。
try_divide(dividend，divisor)	傳回 `dividend`/`divisor`。它一律會執行浮點分割。如果 `expr2` 為 0，其結果一律為 null。`dividend` 必須是數值或間隔。`divisor` 必須是數值。
try_multiply(expr1， expr2)	傳回 `expr1`*`expr2`，且溢位時的結果為 null。可接受的輸入類型與 `*` 運算子相同。
try_subtract(expr1， expr2)	傳回 `expr1`-`expr2`，且溢位時的結果為 null。可接受的輸入類型與 `-` 運算子相同。
unhex(expr)	將十六進位 `expr` 轉換為二進位。
width_bucket(value、min_value、max_value、num_bucket)	傳回 `value` 將在具有 `num_bucket` 儲存貯體的等寬長條圖中指派給的儲存貯體編號，範圍為 `min_value` 到 `max_value`。」

範例

```
-- %
SELECT 2 % 1.8;
+---------+
|(2 % 1.8)|
+---------+
|      0.2|
+---------+
SELECT MOD(2, 1.8);
+-----------+
|mod(2, 1.8)|
+-----------+
|        0.2|
```

```
+-----------+
-- *
SELECT 2 * 3;
+-------+
|(2 * 3)|
+-------+
|      6|
+-------+
-- +
SELECT 1 + 2;
+-------+
|(1 + 2)|
+-------+
|      3|
+-------+
-- -
SELECT 2 - 1;
+-------+
|(2 - 1)|
+-------+
|      1|
+-------+
-- /
SELECT 3 / 2;
+-------+
|(3 / 2)|
+-------+
|    1.5|
+-------+
SELECT 2L / 2L;
+-------+
|(2 / 2)|
+-------+
|    1.0|
+-------+
-- abs
SELECT abs(-1);
+-------+
|abs(-1)|
+-------+
|      1|
+-------+
SELECT abs(INTERVAL -'1-1' YEAR TO MONTH);
+----------------------------------+
```

```
|abs(INTERVAL '-1-1' YEAR TO MONTH)|
+----------------------------------+
|              INTERVAL '1-1' YE...|
+----------------------------------+
-- acos
SELECT acos(1);
+-------+
|ACOS(1)|
+-------+
|    0.0|
+-------+
SELECT acos(2);
+-------+
|ACOS(2)|
+-------+
|    NaN|
+-------+
-- acosh
SELECT acosh(1);
+--------+
|ACOSH(1)|
+--------+
|     0.0|
+--------+
SELECT acosh(0);
+--------+
|ACOSH(0)|
+--------+
|     NaN|
+--------+
-- asin
SELECT asin(0);
+-------+
|ASIN(0)|
+-------+
|    0.0|
+-------+
SELECT asin(2);
+-------+
|ASIN(2)|
+-------+
|    NaN|
+-------+
-- asinh
```

```
SELECT asinh(0);
+--------+
|ASINH(0)|
+--------+
|     0.0|
+--------+
-- atan
SELECT atan(0);
+-------+
|ATAN(0)|
+-------+
|    0.0|
+-------+
-- atan2
SELECT atan2(0, 0);
+-----------+
|ATAN2(0, 0)|
+-----------+
|        0.0|
+-----------+
-- atanh
SELECT atanh(0);
+--------+
|ATANH(0)|
+--------+
|     0.0|
+--------+
SELECT atanh(2);
+--------+
|ATANH(2)|
+--------+
|     NaN|
+--------+
-- bin
SELECT bin(13);
+-------+
|bin(13)|
+-------+
|   1101|
+-------+
SELECT bin(-13);
+--------------------+
|            bin(-13)|
+--------------------+
```

```
|1111111111111111...|
+--------------------+
SELECT bin(13.3);
+---------+
|bin(13.3)|
+---------+
|     1101|
+---------+
-- bround
SELECT bround(2.5, 0);
+--------------+
|bround(2.5, 0)|
+--------------+
|             2|
+--------------+
SELECT bround(25, -1);
+--------------+
|bround(25, -1)|
+--------------+
|            20|
+--------------+
-- cbrt
SELECT cbrt(27.0);
+----------+
|CBRT(27.0)|
+----------+
|       3.0|
+----------+
-- ceil
SELECT ceil(-0.1);
+----------+
|CEIL(-0.1)|
+----------+
|         0|
+----------+
SELECT ceil(5);
+-------+
|CEIL(5)|
+-------+
|      5|
+-------+
SELECT ceil(3.1411, 3);
+---------------+
|ceil(3.1411, 3)|
```

```
+---------------+
|          3.142|
+---------------+
SELECT ceil(3.1411, -3);
+-----------------+
|ceil(3.1411, -3)|
+-----------------+
|             1000|
+-----------------+
-- ceiling
SELECT ceiling(-0.1);
+-------------+
|ceiling(-0.1)|
+-------------+
|            0|
+-------------+
SELECT ceiling(5);
+----------+
|ceiling(5)|
+----------+
|         5|
+----------+
SELECT ceiling(3.1411, 3);
+------------------+
|ceiling(3.1411, 3)|
+------------------+
|             3.142|
+------------------+
SELECT ceiling(3.1411, -3);
+-------------------+
|ceiling(3.1411, -3)|
+-------------------+
|               1000|
+-------------------+
-- conv
SELECT conv('100', 2, 10);
+----------------+
|conv(100, 2, 10)|
+----------------+
|               4|
+----------------+
SELECT conv(-10, 16, -10);
+------------------+
|conv(-10, 16, -10)|
```

```
+------------------+
|               -16|
+------------------+
-- cos
SELECT cos(0);
+------+
|COS(0)|
+------+
|   1.0|
+------+
-- cosh
SELECT cosh(0);
+-------+
|COSH(0)|
+-------+
|    1.0|
+-------+
-- cot
SELECT cot(1);
+------------------+
|            COT(1)|
+------------------+
|0.6420926159343306|
+------------------+
-- csc
SELECT csc(1);
+------------------+
|            CSC(1)|
+------------------+
|1.1883951057781212|
+------------------+
-- degrees
SELECT degrees(3.141592653589793);
+--------------------------+
|DEGREES(3.141592653589793)|
+--------------------------+
|                     180.0|
+--------------------------+
-- div
SELECT 3 div 2;
+---------+
|(3 div 2)|
+---------+
|        1|
```

```
+---------+
SELECT INTERVAL '1-1' YEAR TO MONTH div INTERVAL '-1' MONTH;
+-----------------------------------------------------------+
|(INTERVAL '1-1' YEAR TO MONTH div INTERVAL '-1' MONTH)|
+-----------------------------------------------------------+
|                                                        -13|
+-----------------------------------------------------------+
-- e
SELECT e();
+-----------------+
|              E()|
+-----------------+
|2.718281828459045|
+-----------------+
-- exp
SELECT exp(0);
+------+
|EXP(0)|
+------+
|   1.0|
+------+
-- expm1
SELECT expm1(0);
+--------+
|EXPM1(0)|
+--------+
|     0.0|
+--------+
-- factorial
SELECT factorial(5);
+------------+
|factorial(5)|
+------------+
|         120|
+------------+
-- floor
SELECT floor(-0.1);
+-----------+
|FLOOR(-0.1)|
+-----------+
|         -1|
+-----------+
SELECT floor(5);
+--------+
```

```
|FLOOR(5)|
+--------+
|       5|
+--------+
SELECT floor(3.1411, 3);
+----------------+
|floor(3.1411, 3)|
+----------------+
|           3.141|
+----------------+
SELECT floor(3.1411, -3);
+-----------------+
|floor(3.1411, -3)|
+-----------------+
|                0|
+-----------------+
-- greatest
SELECT greatest(10, 9, 2, 4, 3);
+------------------------+
|greatest(10, 9, 2, 4, 3)|
+------------------------+
|                      10|
+------------------------+
-- hex
SELECT hex(17);
+-------+
|hex(17)|
+-------+
|     11|
+-------+
SELECT hex('SQL');
+------------------+
|    hex(SQL)|
+------------------+
|53514C|
+------------------+
-- hypot
SELECT hypot(3, 4);
+-----------+
|HYPOT(3, 4)|
+-----------+
|        5.0|
+-----------+
-- least
```

```
SELECT least(10, 9, 2, 4, 3);
+---------------------+
|least(10, 9, 2, 4, 3)|
+---------------------+
|                    2|
+---------------------+
-- ln
SELECT ln(1);
+-----+
|ln(1)|
+-----+
|  0.0|
+-----+
-- log
SELECT log(10, 100);
+------------+
|LOG(10, 100)|
+------------+
|         2.0|
+------------+
-- log10
SELECT log10(10);
+---------+
|LOG10(10)|
+---------+
|      1.0|
+---------+
-- log1p
SELECT log1p(0);
+--------+
|LOG1P(0)|
+--------+
|     0.0|
+--------+
-- log2
SELECT log2(2);
+-------+
|LOG2(2)|
+-------+
|    1.0|
+-------+
-- mod
SELECT 2 % 1.8;
+---------+
```

```
|(2 % 1.8)|
+---------+
|      0.2|
+---------+
SELECT MOD(2, 1.8);
+-----------+
|mod(2, 1.8)|
+-----------+
|        0.2|
+-----------+
-- negative
SELECT negative(1);
+-----------+
|negative(1)|
+-----------+
|         -1|
+-----------+
-- pi
SELECT pi();
+-----------------+
|             PI()|
+-----------------+
|3.141592653589793|
+-----------------+
-- pmod
SELECT pmod(10, 3);
+-----------+
|pmod(10, 3)|
+-----------+
|          1|
+-----------+
SELECT pmod(-10, 3);
+------------+
|pmod(-10, 3)|
+------------+
|           2|
+------------+
-- positive
SELECT positive(1);
+-----+
|(+ 1)|
+-----+
|    1|
+-----+
```

```
-- pow
SELECT pow(2, 3);
+---------+
|pow(2, 3)|
+---------+
|      8.0|
+---------+
-- power
SELECT power(2, 3);
+-----------+
|POWER(2, 3)|
+-----------+
|        8.0|
+-----------+
-- radians
SELECT radians(180);
+-----------------+
|     RADIANS(180)|
+-----------------+
|3.141592653589793|
+-----------------+
-- rand
SELECT rand();
+------------------+
|            rand()|
+------------------+
|0.7211420708112387|
+------------------+
SELECT rand(0);
+------------------+
|           rand(0)|
+------------------+
|0.7604953758285915|
+------------------+
SELECT rand(null);
+------------------+
|        rand(NULL)|
+------------------+
|0.7604953758285915|
+------------------+
-- randn
SELECT randn();
+-------------------+
|            randn()|
```

```
+-------------------+
|-0.8175603217732732|
+-------------------+
SELECT randn(0);
+------------------+
|          randn(0)|
+------------------+
|1.6034991609278433|
+------------------+
SELECT randn(null);
+------------------+
|       randn(NULL)|
+------------------+
|1.6034991609278433|
+------------------+
-- random
SELECT random();
+-----------------+
|           rand()|
+-----------------+
|0.394205008255365|
+-----------------+
SELECT random(0);
+------------------+
|           rand(0)|
+------------------+
|0.7604953758285915|
+------------------+
SELECT random(null);
+------------------+
|        rand(NULL)|
+------------------+
|0.7604953758285915|
+------------------+
-- rint
SELECT rint(12.3456);
+-------------+
|rint(12.3456)|
+-------------+
|         12.0|
+-------------+
-- round
SELECT round(2.5, 0);
+-------------+
```

```
|round(2.5, 0)|
+-------------+
|            3|
+-------------+
-- sec
SELECT sec(0);
+------+
|SEC(0)|
+------+
|   1.0|
+------+
-- shiftleft
SELECT shiftleft(2, 1);
+---------------+
|shiftleft(2, 1)|
+---------------+
|              4|
+---------------+
-- sign
SELECT sign(40);
+--------+
|sign(40)|
+--------+
|     1.0|
+--------+
SELECT sign(INTERVAL -'100' YEAR);
+--------------------------+
|sign(INTERVAL '-100' YEAR)|
+--------------------------+
|                      -1.0|
+--------------------------+
-- signum
SELECT signum(40);
+----------+
|SIGNUM(40)|
+----------+
|       1.0|
+----------+
SELECT signum(INTERVAL -'100' YEAR);
+----------------------------+
|SIGNUM(INTERVAL '-100' YEAR)|
+----------------------------+
|                        -1.0|
+----------------------------+
```

```
-- sin
SELECT sin(0);
+------+
|SIN(0)|
+------+
|   0.0|
+------+
-- sinh
SELECT sinh(0);
+-------+
|SINH(0)|
+-------+
|    0.0|
+-------+
-- sqrt
SELECT sqrt(4);
+-------+
|SQRT(4)|
+-------+
|    2.0|
+-------+
-- tan
SELECT tan(0);
+------+
|TAN(0)|
+------+
|   0.0|
+------+
-- tanh
SELECT tanh(0);
+-------+
|TANH(0)|
+-------+
|    0.0|
+-------+
-- try_add
SELECT try_add(1, 2);
+-------------+
|try_add(1, 2)|
+-------------+
|            3|
+-------------+
SELECT try_add(2147483647, 1);
+----------------------+
```

```
|try_add(2147483647, 1)|
+----------------------+
|                  NULL|
+----------------------+
SELECT try_add(date'2021-01-01', 1);
+-----------------------------+
|try_add(DATE '2021-01-01', 1)|
+-----------------------------+
|                   2021-01-02|
+-----------------------------+
SELECT try_add(date'2021-01-01', interval 1 year);
+-------------------------------------------+
|try_add(DATE '2021-01-01', INTERVAL '1' YEAR)|
+-------------------------------------------+
|                                 2022-01-01|
+-------------------------------------------+
SELECT try_add(timestamp'2021-01-01 00:00:00', interval 1 day);
+-----------------------------------------------------------+
|try_add(TIMESTAMP '2021-01-01 00:00:00', INTERVAL '1' DAY)|
+-----------------------------------------------------------+
|                                      2021-01-02 00:00:00|
+-----------------------------------------------------------+
SELECT try_add(interval 1 year, interval 2 year);
+-----------------------------------------+
|try_add(INTERVAL '1' YEAR, INTERVAL '2' YEAR)|
+-----------------------------------------+
|                           INTERVAL '3' YEAR|
+-----------------------------------------+
-- try_divide
SELECT try_divide(3, 2);
+----------------+
|try_divide(3, 2)|
+----------------+
|             1.5|
+----------------+
SELECT try_divide(2L, 2L);
+----------------+
|try_divide(2, 2)|
+----------------+
|             1.0|
+----------------+
SELECT try_divide(1, 0);
+----------------+
|try_divide(1, 0)|
```

```
+----------------+
|            NULL|
+----------------+
SELECT try_divide(interval 2 month, 2);
+--------------------------------+
|try_divide(INTERVAL '2' MONTH, 2)|
+--------------------------------+
|             INTERVAL '0-1' YE...|
+--------------------------------+
SELECT try_divide(interval 2 month, 0);
+--------------------------------+
|try_divide(INTERVAL '2' MONTH, 0)|
+--------------------------------+
|                            NULL|
+--------------------------------+
-- try_multiply
SELECT try_multiply(2, 3);
+------------------+
|try_multiply(2, 3)|
+------------------+
|                 6|
+------------------+
SELECT try_multiply(-2147483648, 10);
+-----------------------------+
|try_multiply(-2147483648, 10)|
+-----------------------------+
|                         NULL|
+-----------------------------+
SELECT try_multiply(interval 2 year, 3);
+---------------------------------+
|try_multiply(INTERVAL '2' YEAR, 3)|
+---------------------------------+
|              INTERVAL '6-0' YE...|
+---------------------------------+
-- try_subtract
SELECT try_subtract(2, 1);
+------------------+
|try_subtract(2, 1)|
+------------------+
|                 1|
+------------------+
SELECT try_subtract(-2147483648, 1);
+----------------------------+
|try_subtract(-2147483648, 1)|
```

```
+--------------------------+
|                      NULL|
+--------------------------+
SELECT try_subtract(date'2021-01-02', 1);
+---------------------------------+
|try_subtract(DATE '2021-01-02', 1)|
+---------------------------------+
|                       2021-01-01|
+---------------------------------+
SELECT try_subtract(date'2021-01-01', interval 1 year);
+-------------------------------------------------+
|try_subtract(DATE '2021-01-01', INTERVAL '1' YEAR)|
+-------------------------------------------------+
|                                       2020-01-01|
+-------------------------------------------------+
SELECT try_subtract(timestamp'2021-01-02 00:00:00', interval 1 day);
+-----------------------------------------------------------------+
|try_subtract(TIMESTAMP '2021-01-02 00:00:00', INTERVAL '1' DAY)|
+-----------------------------------------------------------------+
|                                              2021-01-01 00:00:00|
+-----------------------------------------------------------------+
SELECT try_subtract(interval 2 year, interval 1 year);
+-------------------------------------------------+
|try_subtract(INTERVAL '2' YEAR, INTERVAL '1' YEAR)|
+-------------------------------------------------+
|                                 INTERVAL '1' YEAR|
+-------------------------------------------------+
-- unhex
SELECT decode(unhex('53514C'), 'UTF-8');
+---------------------------------------+
|decode(unhex(53514C), UTF-8)|
+---------------------------------------+
|                                   SQL|
+---------------------------------------+
-- width_bucket
SELECT width_bucket(5.3, 0.2, 10.6, 5);
+-------------------------------+
|width_bucket(5.3, 0.2, 10.6, 5)|
+-------------------------------+
|                              3|
+-------------------------------+
SELECT width_bucket(-2.1, 1.3, 3.4, 3);
+-------------------------------+
|width_bucket(-2.1, 1.3, 3.4, 3)|
```

```
+------------------------------+
|                             0|
+------------------------------+
SELECT width_bucket(8.1, 0.0, 5.7, 4);
+------------------------------+
|width_bucket(8.1, 0.0, 5.7, 4)|
+------------------------------+
|                             5|
+------------------------------+
SELECT width_bucket(-0.9, 5.2, 0.5, 2);
+-------------------------------+
|width_bucket(-0.9, 5.2, 0.5, 2)|
+-------------------------------+
|                              3|
+-------------------------------+
SELECT width_bucket(INTERVAL '0' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10);
+-------------------------------------------------------------------------------+
|width_bucket(INTERVAL '0' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10)|
+-------------------------------------------------------------------------------+
|                                                                              1|
+-------------------------------------------------------------------------------+
SELECT width_bucket(INTERVAL '1' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10);
+-------------------------------------------------------------------------------+
|width_bucket(INTERVAL '1' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10)|
+-------------------------------------------------------------------------------+
|                                                                              2|
+-------------------------------------------------------------------------------+
SELECT width_bucket(INTERVAL '0' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10);
+-----------------------------------------------------------------------+
|width_bucket(INTERVAL '0' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10)|
+-----------------------------------------------------------------------+
|                                                                      1|
+-----------------------------------------------------------------------+
SELECT width_bucket(INTERVAL '1' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10);
+-----------------------------------------------------------------------+
|width_bucket(INTERVAL '1' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10)|
+-----------------------------------------------------------------------+
|                                                                      2|
+-----------------------------------------------------------------------+
```

產生器函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援這些 SQL 函數，請參閱 the section called “支援的 SQL 命令”。

函式	Description
explode(expr)	將陣列 `expr` 的元素分成多個資料列，或將 `expr` 的元素分成多個資料列和資料欄。除非另有指定，否則 會將預設資料欄名稱 `col` 用於陣列的元素，或將 `key` 和 `value` 用於映射的元素。
explode_outer(expr)	將陣列 `expr` 的元素分成多個資料列，或將 `expr` 的元素分成多個資料列和資料欄。除非另有指定，否則 會將預設資料欄名稱 `col` 用於陣列的元素，或將 `key` 和 `value` 用於映射的元素。
inline(expr)	將結構陣列分解為資料表。除非另有指定，否則預設會使用資料欄名稱 col1、col2 等。
inline_outer(expr)	將結構陣列分解為資料表。除非另有指定，否則預設會使用資料欄名稱 col1、col2 等。
posexplode(expr)	將陣列 `expr` 的元素分成多個具有 位置的資料列，或將 `expr` 的元素分成多個具有 位置的資料列和資料欄。除非另有指定，否則 會將資料欄名稱 `pos` 用於位置，將 `col` 用於陣列的元素，將 `key` 和 `value` 用於映射的元素。
posexplode_outer(expr)	將陣列 `expr` 的元素分成多個具有 位置的資料列，或將 `expr` 的元素分成多個具有 位置的資料列和資料欄。除非另有指定，否則 會將資料欄名稱 `pos` 用於位置，將 `col` 用於陣列的元素，將 `key` 和 `value` 用於映射的元素。
stack(n、expr1、...、exprk)	將 `expr1`、...、`exprk` 分隔為 `n` 資料列。除非另有指定，否則預設會使用資料欄名稱 col0、col1 等。

範例

```
-- explode
SELECT explode(array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT explode(collection => array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT * FROM explode(collection => array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

-- explode_outer
SELECT explode_outer(array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT explode_outer(collection => array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT * FROM explode_outer(collection => array(10, 20));
```

```
+---+
|col|
+---+
| 10|
| 20|
+---+

-- inline
SELECT inline(array(struct(1, 'a'), struct(2, 'b')));
+----+----+
|col1|col2|
+----+----+
|   1|   a|
|   2|   b|
+----+----+

-- inline_outer
SELECT inline_outer(array(struct(1, 'a'), struct(2, 'b')));
+----+----+
|col1|col2|
+----+----+
|   1|   a|
|   2|   b|
+----+----+

-- posexplode
SELECT posexplode(array(10,20));
+---+---+
|pos|col|
+---+---+
|  0| 10|
|  1| 20|
+---+---+

SELECT * FROM posexplode(array(10,20));
+---+---+
|pos|col|
+---+---+
|  0| 10|
|  1| 20|
+---+---+

-- posexplode_outer
SELECT posexplode_outer(array(10,20));
```

```
+---+---+
|pos|col|
+---+---+
|  0| 10|
|  1| 20|
+---+---+

SELECT * FROM posexplode_outer(array(10,20));
+---+---+
|pos|col|
+---+---+
|  0| 10|
|  1| 20|
+---+---+

-- stack
SELECT stack(2, 1, 2, 3);
+----+----+
|col0|col1|
+----+----+
|   1|   2|
|   3|NULL|
+----+----+
```

SELECT 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

OpenSearch SQL 支援用於從一或多個資料表擷取結果集的SELECT陳述式。下一節說明整體查詢語法和查詢的不同建構。

語法

```
select_statement
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select_statement, ... ]
[ ORDER BY
    { expression [ ASC | DESC ] [ NULLS { FIRST | LAST } ]
    [ , ... ]
```

```
    }
]
[ SORT BY
    { expression [ ASC | DESC ] [ NULLS { FIRST | LAST } ]
    [ , ... ]
    }
]
[ WINDOW { named_window [ , WINDOW named_window, ... ] } ]
[ LIMIT { ALL | expression } ]
```

而 select_statement 定義為：

```
SELECT [ ALL | DISTINCT ] { [ [ named_expression ] [ , ... ] ] }
FROM { from_item [ , ... ] }
[ PIVOT clause ]
[ UNPIVOT clause ]
[ LATERAL VIEW clause ] [ ... ]
[ WHERE boolean_expression ]
[ GROUP BY expression [ , ... ] ]
[ HAVING boolean_expression ]
```

參數

- ALL

  從關係中選取所有相符的資料列，並預設為啟用。

- DISTINCT

  在移除結果中的重複項目後，從關係中選取所有相符的資料列。

- named_expression

  具有指派名稱的表達式。一般而言，它會表示資料欄表達式。

  語法: `expression [[AS] alias]`

- from_item

  資料表關聯

  聯結關係

  樞紐關係

取消樞紐關係

資料表值函數

內嵌資料表

[LATERAL] (Subquery)

- PIVOT

  PIVOT 子句用於資料觀點。您可以根據特定資料欄值取得彙總值。

- UNPIVOT

  UNPIVOT 子句會將資料欄轉換為資料列。這是 的反向PIVOT，但值的彙總除外。

- 橫向檢視

  LATERAL VIEW 子句會與 等產生器函數搭配使用EXPLODE，這會產生包含一或多個資料列的虛擬資料表。

  LATERAL VIEW 會將資料列套用至每個原始輸出資料列。

- WHERE

  根據提供的述詞篩選 FROM子句的結果。

- GROUP BY

  指定用於分組資料列的表達式。

  這與彙總函數 (MIN、MAX、AVG、、 COUNT SUM等) 搭配使用，以根據每個群組中的分組表達式和彙總值來分組資料列。

  當子FILTER句連接到彙總函數時，只有相符的資料列會傳遞至該函數。

- 哈維

  指定 產生的資料列GROUP BY篩選依據的述詞。

  子HAVING句用於在執行分組後篩選資料列。

  如果指定的 HAVING 沒有 GROUP BY，則表示 GROUP BY沒有分組表達式 （全域彙總）。

- ORDER BY

指定查詢完整結果集的資料列順序。

輸出列會依分割區排序。

此參數與 DISTRIBUTE BY SORT BY和 互斥，無法同時指定。

- SORT BY

  指定列在每個分割區中排序的順序。

  此參數與 互斥ORDER BY，無法同時指定。

- LIMIT

  指定陳述式或子查詢可傳回的資料列數目上限。

  此子句主要與 搭配使用ORDER BY，以產生決定性結果。

- boolean_expression

  指定評估為結果類型布林值的任何表達式。

  兩個或多個表達式可以使用邏輯運算子 (AND、) 結合在一起OR。

- 表達式

  指定評估為值的一或多個值、運算子和 SQL 函數的組合。

- named_window

  指定一或多個來源視窗規格的別名。

  您可以在查詢的精靈定義中參考來源視窗規格。

WHERE 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

WHERE 子句用於根據指定的條件限制查詢子FROM句或子查詢的結果。

語法

```
WHERE boolean_expression
```

參數

- boolean_expression

  指定評估為結果類型布林值的任何表達式。

  兩個或多個表達式可以使用邏輯運算子 (AND、) 結合在一起OR。

範例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
(300, 'Mike', 80),
(400, 'Dan',  50);

-- Comparison operator in `WHERE` clause.
SELECT * FROM person WHERE id > 200 ORDER BY id;
+---+----+---+
| id|name|age|
+---+----+---+
|300|Mike| 80|
|400| Dan| 50|
+---+----+---+

-- Comparison and logical operators in `WHERE` clause.
SELECT * FROM person WHERE id = 200 OR id = 300 ORDER BY id;
+---+----+----+
| id|name| age|
+---+----+----+
|200|Mary|null|
|300|Mike|  80|
+---+----+----+

-- IS NULL expression in `WHERE` clause.
SELECT * FROM person WHERE id > 300 OR age IS NULL ORDER BY id;
+---+----+----+
```

```
| id|name| age|
+---+----+----+
|200|Mary|null|
|400| Dan|  50|
+---+----+----+

-- Function expression in `WHERE` clause.
SELECT * FROM person WHERE length(name) > 3 ORDER BY id;
+---+----+----+
| id|name| age|
+---+----+----+
|100|John|  30|
|200|Mary|null|
|300|Mike|  80|
+---+----+----+

-- `BETWEEN` expression in `WHERE` clause.
SELECT * FROM person WHERE id BETWEEN 200 AND 300 ORDER BY id;
+---+----+----+
| id|name| age|
+---+----+----+
|200|Mary|null|
|300|Mike|  80|
+---+----+----+

-- Scalar Subquery in `WHERE` clause.
SELECT * FROM person WHERE age > (SELECT avg(age) FROM person);
+---+----+---+
| id|name|age|
+---+----+---+
|300|Mike| 80|
+---+----+---+

-- Correlated Subquery in `WHERE` clause.
SELECT id FROM person
WHERE exists (SELECT id FROM person where id = 200);
+---+----+----+
|id |name|age |
+---+----+----+
|200|Mary|null|
+---+----+----+
```

GROUP BY 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

GROUP BY 子句用於根據一組指定的分組表達式對資料列進行分組，並根據一或多個指定的彙總函數對資料列群組進行運算彙總。

系統也會透過 GROUPING SETS、CUBE、 ROLLUP子句，對相同的輸入記錄集進行多個彙總。分組表達式和進階彙總可以在 GROUP BY子句中混合，並在 GROUPING SETS 子句中巢狀化。如需詳細資訊，請參閱 Mixed/Nested Grouping Analytics 一節。

當子FILTER句連接至彙總函數時，只會將相符的資料列傳遞至該函數。

語法

```
GROUP BY group_expression [ , group_expression [ , ... ] ] [ WITH { ROLLUP | CUBE } ]
GROUP BY { group_expression | { ROLLUP | CUBE | GROUPING SETS } (grouping_set
 [ , ...]) } [ , ... ]
```

雖然彙總函數定義為：

```
aggregate_name ( [ DISTINCT ] expression [ , ... ] ) [ FILTER ( WHERE
 boolean_expression ) ]
```

參數

- group_expression

  指定資料列分組的條件。資料列分組是根據分組表達式的結果值執行。

  分組表達式可以是 之類的資料欄名稱GROUP BY a、 之類的資料欄位置GROUP BY 0，或 之類的表達式GROUP BY a + b。

- grouping_set

  分組集由括號中的零或多個逗號分隔表達式指定。當分組集只有一個元素時，可以省略括號。

例如，GROUPING SETS ((a), (b)) 與 GROUPING SETS (a, b) 相同。

語法: { ([expression [, ...]]) | expression }

- 分組集

  將 之後指定的每個分組集的資料列分組GROUPING SETS。

  例如，GROUP BY GROUPING SETS ((warehouse), (product)) 在語義上等同於 GROUP BY warehouse和 結果的聯集GROUP BY product。此子句是 UNION ALL 的速記，其中UNION ALL運算子的每個分支都會對子GROUPING SETS句中指定的每個分組集執行彙總。

  同樣地，GROUP BY GROUPING SETS ((warehouse, product), (product), ()) 在語義上等同於 GROUP BY warehouse, product, GROUP BY product和 全域彙總結果的聯集。

- ROLLUP

  在單一陳述式中指定多個層級的彙總。此子句用於根據多個分組集計算彙總。 ROLLUP 是 的速記GROUPING SETS。

  例如，GROUP BY warehouse, product WITH ROLLUP or GROUP BY ROLLUP(warehouse, product) 等同於 GROUP BY GROUPING SETS((warehouse, product), (warehouse), ())。

  GROUP BY ROLLUP(warehouse, product, (warehouse, location)) 相當於 GROUP BY GROUPING SETS((warehouse, product, location), (warehouse, product), (warehouse), ())。

  ROLLUP 規格的 N 元素會產生 N+1 GROUPING SETS。

- CUBE

  CUBE 子句用於根據 GROUP BY 子句中指定的分組資料欄組合來執行彙總。CUBE 是 GROUPING SETS 的速記。

  例如，GROUP BY warehouse, product WITH CUBE or GROUP BY CUBE(warehouse, product) 等同於 GROUP BY GROUPING SETS((warehouse, product), (warehouse), (product), ())。

  GROUP BY CUBE(warehouse, product, (warehouse, location)) 相當於 GROUP BY GROUPING SETS((warehouse, product, location), (warehouse, product), (warehouse, location), (product, warehouse, location), (warehouse),

(product), (warehouse, product), ())。CUBE 規格的 N 元素會產生 2^NGROUPING SETS。

- 混合/巢狀分組分析

  GROUP BY 子句可以包含多個 group_expressions 和多個 CUBE|ROLLUP|GROUPING SETS。GROUPING SETS也可以具有巢狀CUBE|ROLLUP|GROUPING SETS子句，例如 GROUPING SETS(ROLLUP(warehouse, location)、CUBE(warehouse, location))、GROUPING SETS(warehouse, GROUPING SETS(location, GROUPING SETS(ROLLUP(warehouse, location),CUBE(warehouse, location))))。

  CUBE|ROLLUP 只是 的語法糖GROUPING SETS。請參閱上述章節，了解如何翻譯CUBE|ROLLUP為GROUPING SETS。 group_expression可在此內容GROUPING SETS下視為單一群組。

  對於 GROUP BY子句GROUPING SETS中的多個 ，我們會GROUPING SETS執行原始 的跨產品來產生單一 GROUPING SETS。對於 GROUPING SETS子句GROUPING SETS中的巢狀化，我們只需要取得其分組集並將其分割。

  例如，GROUP BY warehouse, GROUPING SETS((product), ()), GROUPING SETS((location, size), (location), (size), ()) and GROUP BY warehouse, ROLLUP(product), CUBE(location, size) 等同於 GROUP BY GROUPING SETS((warehouse, product, location, size), (warehouse, product, location), (warehouse, product, size), (warehouse, product), (warehouse, location, size), (warehouse, location), (warehouse, size), (warehouse))。

  GROUP BY GROUPING SETS(GROUPING SETS(warehouse), GROUPING SETS((warehouse, product))) 相當於 GROUP BY GROUPING SETS((warehouse), (warehouse, product))。

- aggregate_name

  指定彙總函數名稱 (MIN、MAX、AVG、、 COUNT SUM等) 。

- DISTINCT

  在將重複項目傳遞至彙總函數之前，移除輸入列中的重複項目。

- 篩選條件

  篩選 WHERE子句boolean_expression中 評估為 true 的輸入資料列會傳遞至彙總函數；其他資料列則會遭到捨棄。

範例

```
CREATE TABLE dealer (id INT, city STRING, car_model STRING, quantity INT);
INSERT INTO dealer VALUES
(100, 'Fremont', 'Honda Civic', 10),
(100, 'Fremont', 'Honda Accord', 15),
(100, 'Fremont', 'Honda CRV', 7),
(200, 'Dublin', 'Honda Civic', 20),
(200, 'Dublin', 'Honda Accord', 10),
(200, 'Dublin', 'Honda CRV', 3),
(300, 'San Jose', 'Honda Civic', 5),
(300, 'San Jose', 'Honda Accord', 8);

-- Sum of quantity per dealership. Group by `id`.
SELECT id, sum(quantity) FROM dealer GROUP BY id ORDER BY id;
+---+-------------+
| id|sum(quantity)|
+---+-------------+
|100|           32|
|200|           33|
|300|           13|
+---+-------------+

-- Use column position in GROUP by clause.
SELECT id, sum(quantity) FROM dealer GROUP BY 1 ORDER BY 1;
+---+-------------+
| id|sum(quantity)|
+---+-------------+
|100|           32|
|200|           33|
|300|           13|
+---+-------------+

-- Multiple aggregations.
-- 1. Sum of quantity per dealership.
-- 2. Max quantity per dealership.
SELECT id, sum(quantity) AS sum, max(quantity) AS max FROM dealer GROUP BY id ORDER BY
 id;
+---+---+---+
| id|sum|max|
+---+---+---+
|100| 32| 15|
|200| 33| 20|
|300| 13|  8|
```

```
+---+---+---+

-- Count the number of distinct dealer cities per car_model.
SELECT car_model, count(DISTINCT city) AS count FROM dealer GROUP BY car_model;
+------------+-----+
|   car_model|count|
+------------+-----+
| Honda Civic|    3|
|   Honda CRV|    2|
|Honda Accord|    3|
+------------+-----+

-- Sum of only 'Honda Civic' and 'Honda CRV' quantities per dealership.
SELECT id, sum(quantity) FILTER (
WHERE car_model IN ('Honda Civic', 'Honda CRV')
) AS `sum(quantity)` FROM dealer
GROUP BY id ORDER BY id;
+---+-------------+
| id|sum(quantity)|
+---+-------------+
|100|           17|
|200|           23|
|300|            5|
+---+-------------+

-- Aggregations using multiple sets of grouping columns in a single statement.
-- Following performs aggregations based on four sets of grouping columns.
-- 1. city, car_model
-- 2. city
-- 3. car_model
-- 4. Empty grouping set. Returns quantities for all city and car models.
SELECT city, car_model, sum(quantity) AS sum FROM dealer
GROUP BY GROUPING SETS ((city, car_model), (city), (car_model), ())
ORDER BY city;
+---------+------------+---+
|     city|   car_model|sum|
+---------+------------+---+
|     null|        null| 78|
|     null| HondaAccord| 33|
|     null|    HondaCRV| 10|
|     null|  HondaCivic| 35|
|   Dublin|        null| 33|
|   Dublin| HondaAccord| 10|
|   Dublin|    HondaCRV|  3|
```

```
|   Dublin|  HondaCivic| 20|
|  Fremont|        null| 32|
|  Fremont| HondaAccord| 15|
|  Fremont|    HondaCRV|  7|
|  Fremont|  HondaCivic| 10|
| San Jose|        null| 13|
| San Jose| HondaAccord|  8|
| San Jose|  HondaCivic|  5|
+---------+------------+---+

-- Group by processing with `ROLLUP` clause.
-- Equivalent GROUP BY GROUPING SETS ((city, car_model), (city), ())
SELECT city, car_model, sum(quantity) AS sum FROM dealer
GROUP BY city, car_model WITH ROLLUP
ORDER BY city, car_model;
+---------+------------+---+
|     city|   car_model|sum|
+---------+------------+---+
|     null|        null| 78|
|   Dublin|        null| 33|
|   Dublin| HondaAccord| 10|
|   Dublin|    HondaCRV|  3|
|   Dublin|  HondaCivic| 20|
|  Fremont|        null| 32|
|  Fremont| HondaAccord| 15|
|  Fremont|    HondaCRV|  7|
|  Fremont|  HondaCivic| 10|
| San Jose|        null| 13|
| San Jose| HondaAccord|  8|
| San Jose|  HondaCivic|  5|
+---------+------------+---+

-- Group by processing with `CUBE` clause.
-- Equivalent GROUP BY GROUPING SETS ((city, car_model), (city), (car_model), ())
SELECT city, car_model, sum(quantity) AS sum FROM dealer
GROUP BY city, car_model WITH CUBE
ORDER BY city, car_model;
+---------+------------+---+
|     city|   car_model|sum|
+---------+------------+---+
|     null|        null| 78|
|     null| HondaAccord| 33|
|     null|    HondaCRV| 10|
|     null|  HondaCivic| 35|
```

```
|   Dublin|        null| 33|
|   Dublin| HondaAccord| 10|
|   Dublin|    HondaCRV|  3|
|   Dublin|  HondaCivic| 20|
|  Fremont|        null| 32|
|  Fremont| HondaAccord| 15|
|  Fremont|    HondaCRV|  7|
|  Fremont|  HondaCivic| 10|
| San Jose|        null| 13|
| San Jose| HondaAccord|  8|
| San Jose|  HondaCivic|  5|
+---------+------------+---+

--Prepare data for ignore nulls example
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'Mary', NULL),
(200, 'John', 30),
(300, 'Mike', 80),
(400, 'Dan', 50);

--Select the first row in column age
SELECT FIRST(age) FROM person;
+--------------------+
| first(age, false)  |
+--------------------+
| NULL               |
+--------------------+

--Get the first row in column `age` ignore nulls,last row in column `id` and sum of
 column `id`.
SELECT FIRST(age IGNORE NULLS), LAST(id), SUM(id) FROM person;
+-------------------+------------------+----------+
| first(age, true)  | last(id, false)  | sum(id)  |
+-------------------+------------------+----------+
| 30                | 400              | 1000     |
+-------------------+------------------+----------+
```

HAVING 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

HAVING 子句用於GROUP BY根據指定的條件篩選 產生的結果。它通常與 GROUP BY子句搭配使用。

語法

```
HAVING boolean_expression
```

參數

- boolean_expression

  指定評估為結果類型布林值的任何表達式。兩個或多個表達式可以使用邏輯運算子 (AND、) 結合在一起OR。

  注意 HAVING子句中指定的表達式只能參考：

  1. 常數
  2. 出現在 中的表達式 GROUP BY
  3. 彙總函數

範例

```
CREATE TABLE dealer (id INT, city STRING, car_model STRING, quantity INT);
INSERT INTO dealer VALUES
(100, 'Fremont', 'Honda Civic', 10),
(100, 'Fremont', 'Honda Accord', 15),
(100, 'Fremont', 'Honda CRV', 7),
(200, 'Dublin', 'Honda Civic', 20),
(200, 'Dublin', 'Honda Accord', 10),
(200, 'Dublin', 'Honda CRV', 3),
(300, 'San Jose', 'Honda Civic', 5),
(300, 'San Jose', 'Honda Accord', 8);

-- `HAVING` clause referring to column in `GROUP BY`.
```

```
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING city = 'Fremont';
+-------+---+
|   city|sum|
+-------+---+
|Fremont| 32|
+-------+---+

-- `HAVING` clause referring to aggregate function.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING sum(quantity) > 15;
+-------+---+
|   city|sum|
+-------+---+
| Dublin| 33|
|Fremont| 32|
+-------+---+

-- `HAVING` clause referring to aggregate function by its alias.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING sum > 15;
+-------+---+
|   city|sum|
+-------+---+
| Dublin| 33|
|Fremont| 32|
+-------+---+

-- `HAVING` clause referring to a different aggregate function than what is present in
-- `SELECT` list.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING max(quantity) > 15;
+------+---+
|  city|sum|
+------+---+
|Dublin| 33|
+------+---+

-- `HAVING` clause referring to constant expression.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING 1 > 0 ORDER BY city;
+--------+---+
|    city|sum|
+--------+---+
|  Dublin| 33|
| Fremont| 32|
|San Jose| 13|
+--------+---+
```

```
-- `HAVING` clause without a `GROUP BY` clause.
SELECT sum(quantity) AS sum FROM dealer HAVING sum(quantity) > 10;
+---+
|sum|
+---+
| 78|
+---+
```

ORDER BY 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

ORDER BY 子句用於依使用者指定的順序，以排序方式傳回結果列。與 SORT BY 子句不同，此子句保證輸出中的總順序。

語法

```
ORDER BY { expression [ sort_direction | nulls_sort_order ] [ , ... ] }
```

參數

- ORDER BY

  指定以逗號分隔的表達式清單，以及選用參數sort_directionnulls_sort_order，用於排序資料列。
- sort_direction

  選擇性地指定要以遞增或遞減順序排序資料列。

  排序方向的有效值ASC適用於遞增和DESC遞減。

  如果未明確指定排序方向，則預設資料列會遞增排序。

  語法: [ASC | DESC]
- nulls_sort_order

  選擇性地指定是否在非 NULL NULL值之前/之後傳回值。

如果未指定 null_sort_order，則先NULLs排序，如果排序順序為 ，則先排序；如果排序順序為 ，則先排序 ASC NULLS 排序DESC。

1. 如果指定 NULLS FIRST ，則無論排序順序為何，都會先傳回 NULL 值。

2. 如果指定 NULLS LAST ，則無論排序順序為何，都會最後傳回 NULL 值。

語法: [NULLS { FIRST | LAST }]

範例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
(300, 'Mike', 80),
(400, 'Jerry', NULL),
(500, 'Dan',  50);

-- Sort rows by age. By default rows are sorted in ascending manner with NULL FIRST.
SELECT name, age FROM person ORDER BY age;
+-----+----+
| name| age|
+-----+----+
|Jerry|null|
| Mary|null|
| John|  30|
|  Dan|  50|
| Mike|  80|
+-----+----+

-- Sort rows in ascending manner keeping null values to be last.
SELECT name, age FROM person ORDER BY age NULLS LAST;
+-----+----+
| name| age|
+-----+----+
| John|  30|
|  Dan|  50|
| Mike|  80|
| Mary|null|
|Jerry|null|
+-----+----+
```

```
-- Sort rows by age in descending manner, which defaults to NULL LAST.
SELECT name, age FROM person ORDER BY age DESC;
+-----+----+
| name| age|
+-----+----+
| Mike|  80|
|  Dan|  50|
| John|  30|
|Jerry|null|
| Mary|null|
+-----+----+

-- Sort rows in ascending manner keeping null values to be first.
SELECT name, age FROM person ORDER BY age DESC NULLS FIRST;
+-----+----+
| name| age|
+-----+----+
|Jerry|null|
| Mary|null|
| Mike|  80|
|  Dan|  50|
| John|  30|
+-----+----+

-- Sort rows based on more than one column with each column having different
-- sort direction.
SELECT * FROM person ORDER BY name ASC, age DESC;
+---+-----+----+
| id| name| age|
+---+-----+----+
|500|  Dan|  50|
|400|Jerry|null|
|100| John|  30|
|200| Mary|null|
|300| Mike|  80|
+---+-----+----+
```

JOIN 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

SQL 聯結用於根據聯結條件結合來自兩個關係的資料列。下一節說明整體聯結語法，以及不同類型的聯結以及範例。

語法

```
relation INNER JOIN relation [ join_criteria ]
```

參數

- 關係

  指定要加入的關係。

- join_type

  指定聯結類型。

  語法: `INNER | CROSS | LEFT OUTER`

- join_criteria

  指定來自一個關係的資料列將如何與另一個關係的資料列合併。

  語法: `ON boolean_expression | USING ( column_name [ , ... ] )`

- boolean_expression

  指定傳回類型為布林值的表達式。

聯結類型

- 內部聯結

  內部聯結需要明確指定。它會選取在兩個關係中具有相符值的資料列。

語法: `relation INNER JOIN relation [ join_criteria ]`

- 左聯結

  左聯結會傳回來自左關係的所有值，以及來自右關係的相符值，或者如果沒有相符項目，則附加 NULL。它也稱為左側外部聯結。

  語法: `relation LEFT OUTER JOIN relation [ join_criteria ]`

- 交叉聯結

  交叉聯結會傳回兩個關係的笛卡爾產品。

  語法: `relation CROSS JOIN relation [ join_criteria ]`

範例

```
-- Use employee and department tables to demonstrate different type of joins.
SELECT * FROM employee;
+---+-----+------+
| id| name|deptno|
+---+-----+------+
|105|Chloe|     5|
|103| Paul|     3|
|101| John|     1|
|102| Lisa|     2|
|104| Evan|     4|
|106|  Amy|     6|
+---+-----+------+
SELECT * FROM department;
+------+-----------+
|deptno|   deptname|
+------+-----------+
|     3|Engineering|
|     2|      Sales|
|     1|  Marketing|
+------+-----------+

-- Use employee and department tables to demonstrate inner join.
SELECT id, name, employee.deptno, deptname
FROM employee INNER JOIN department ON employee.deptno = department.deptno;
+---+-----+------+-----------|
| id| name|deptno|   deptname|
```

```
+---+-----+------+-----------|
|103| Paul|     3|Engineering|
|101| John|     1|  Marketing|
|102| Lisa|     2|      Sales|
+---+-----+------+-----------|

-- Use employee and department tables to demonstrate left join.
SELECT id, name, employee.deptno, deptname
FROM employee LEFT JOIN department ON employee.deptno = department.deptno;
+---+-----+------+-----------|
| id| name|deptno|   deptname|
+---+-----+------+-----------|
|105|Chloe|     5|       NULL|
|103| Paul|     3|Engineering|
|101| John|     1|  Marketing|
|102| Lisa|     2|      Sales|
|104| Evan|     4|       NULL|
|106|  Amy|     6|       NULL|
+---+-----+------+-----------|

-- Use employee and department tables to demonstrate cross join.
SELECT id, name, employee.deptno, deptname FROM employee CROSS JOIN department;
+---+-----+------+-----------|
| id| name|deptno|   deptname|
+---+-----+------+-----------|
|105|Chloe|     5|Engineering|
|105|Chloe|     5|  Marketing|
|105|Chloe|     5|      Sales|
|103| Paul|     3|Engineering|
|103| Paul|     3|  Marketing|
|103| Paul|     3|      Sales|
|101| John|     1|Engineering|
|101| John|     1|  Marketing|
|101| John|     1|      Sales|
|102| Lisa|     2|Engineering|
|102| Lisa|     2|  Marketing|
|102| Lisa|     2|      Sales|
|104| Evan|     4|Engineering|
|104| Evan|     4|  Marketing|
|104| Evan|     4|      Sales|
|106|  Amy|     4|Engineering|
|106|  Amy|     4|  Marketing|
|106|  Amy|     4|      Sales|
```

```
+---+-----+------+-----------|
```

LIMIT 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

LIMIT 子句用於限制 SELECT陳述式傳回的資料列數。一般而言，此子句會與 搭配使用ORDER BY，以確保結果具有決定性。

語法

```
LIMIT { ALL | integer_expression }
```

參數

- ALL

  如果指定，查詢會傳回所有資料列。換句話說，如果指定此選項，則不會套用任何限制。

- integer_expression

  指定傳回整數的可摺疊表達式。

範例

```
CREATE TABLE person (name STRING, age INT);
INSERT INTO person VALUES
('Jane Doe', 25),
('Pat C', 18),
('Nikki W', 16),
('John D', 25),
('Juan L', 18),
('Jorge S', 16);

-- Select the first two rows.
SELECT name, age FROM person ORDER BY name LIMIT 2;
+-------+---+
```

```
|   name|age|
+-------+---+
|  Pat C| 18|
|Jorge S| 16|
+------+---+

-- Specifying ALL option on LIMIT returns all the rows.
SELECT name, age FROM person ORDER BY name LIMIT ALL;
+--------+---+
|    name|age|
+--------+---+
|   Pat C| 18|
| Jorge S| 16|
|  Juan L| 18|
|  John D| 25|
| Nikki W| 16|
|Jane Doe| 25|
+--------+---+

-- A function expression as an input to LIMIT.
SELECT name, age FROM person ORDER BY name LIMIT length('OPENSEARCH');
+-------+---+
|   name|age|
+-------+---+
|  Pat C| 18|
|Jorge S| 16|
| Juan L| 18|
| John D| 25|
|Nikki W| 16|
+-------+---+
```

CASE 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

CASE 子句使用規則根據指定的條件傳回特定結果，類似於其他程式設計語言中的 if/else 陳述式。

語法

```
CASE [ expression ] { WHEN boolean_expression THEN then_expression } [ ... ]
[ ELSE else_expression ]
END
```

參數

- boolean_expression

  指定評估為結果類型布林值的任何表達式。

  兩個或多個表達式可以使用邏輯運算子 (AND、) 結合在一起OR。

- then_expression

  根據 boolean_expression 條件指定然後表達式。

  then_expression 和 else_expression應該都是相同類型或可強制使用常見類型。

- else_expression

  指定預設表達式。

  then_expression 和 else_expression 應該都是相同類型或可強制使用常見類型。

範例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
(300, 'Mike', 80),
(400, 'Dan', 50);
SELECT id, CASE WHEN id > 200 THEN 'bigger' ELSE 'small' END FROM person;
+------+-----------------------------------------------------+
|  id  | CASE WHEN (id > 200) THEN bigger ELSE small END  |
+------+-----------------------------------------------------+
| 100  | small                                               |
| 200  | small                                               |
| 300  | bigger                                              |
| 400  | bigger                                              |
+------+-----------------------------------------------------+
SELECT id, CASE id WHEN 100 then 'bigger' WHEN  id > 300 THEN '300' ELSE 'small' END
 FROM person;
```

```
+------
+-------------------------------------------------------------------------------------------------------
+
|  id  | CASE WHEN (id = 100) THEN bigger WHEN (id = CAST((id > 300) AS INT)) THEN 300
 ELSE small END  |
+------
+-------------------------------------------------------------------------------------------------------
+
| 100  | bigger
                |
| 200  | small
                |
| 300  | small
                |
| 400  | small
                |
+------
+-------------------------------------------------------------------------------------------------------
+
```

常見資料表表達式

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

通用資料表表達式 (CTE) 定義了暫時結果集，使用者可以在 SQL 陳述式的範圍內參考此結果集可能多次。CTE 主要用於SELECT陳述式。

語法

```
WITH common_table_expression [ , ... ]
```

而 common_table_expression 定義為：

```
Syntexpression_name [ ( column_name [ , ... ] ) ] [ AS ] ( query )
```

參數

- expression_name

指定常見資料表表達式的名稱。

- query

  SELECT 陳述式。

範例

```
-- CTE with multiple column aliases
WITH t(x, y) AS (SELECT 1, 2)
SELECT * FROM t WHERE x = 1 AND y = 2;
+---+---+
|  x|  y|
+---+---+
|  1|  2|
+---+---+

-- CTE in CTE definition
WITH t AS (
WITH t2 AS (SELECT 1)
SELECT * FROM t2
)
SELECT * FROM t;
+---+
|  1|
+---+
|  1|
+---+

-- CTE in subquery
SELECT max(c) FROM (
WITH t(c) AS (SELECT 1)
SELECT * FROM t
);
+------+
|max(c)|
+------+
|     1|
+------+

-- CTE in subquery expression
SELECT (
WITH t AS (SELECT 1)
```

```
SELECT * FROM t
);
+----------------+
|scalarsubquery()|
+----------------+
|               1|
+----------------+

-- CTE in CREATE VIEW statement
CREATE VIEW v AS
WITH t(a, b, c, d) AS (SELECT 1, 2, 3, 4)
SELECT * FROM t;
SELECT * FROM v;
+---+---+---+---+
|  a|  b|  c|  d|
+---+---+---+---+
|  1|  2|  3|  4|
+---+---+---+---+
```

EXPLAIN

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

EXPLAIN 陳述式用於為輸入陳述式提供邏輯/物理計畫。根據預設，此子句僅提供實體計劃的相關資訊。

語法

```
EXPLAIN [ EXTENDED | CODEGEN | COST | FORMATTED ] statement
```

參數

- 延伸

  產生剖析的邏輯計畫、分析的邏輯計畫、最佳化的邏輯計畫和實體計畫。

  剖析邏輯計畫是從查詢擷取的未解析計畫。

分析的邏輯計畫會轉換，將 unresolvedAttribute和 轉換為unresolvedRelation完全輸入的物件。

最佳化的邏輯計畫會透過一組最佳化規則進行轉換，進而產生實體計畫。

- CODEGEN

  如果有 和實體計畫，則產生陳述式的程式碼。

- COST

  如果計劃節點統計資料可用， 會產生邏輯計劃和統計資料。

- 格式

  產生兩個區段：實體計畫大綱和節點詳細資訊。

- 陳述式

  指定要解釋的 SQL 陳述式。

範例

```
-- Default Output
EXPLAIN select k, sum(v) from values (1, 2), (1, 3) t(k, v) group by k;
+----------------------------------------------------+
|                                                plan|
+----------------------------------------------------+
| == Physical Plan ==
*(2) HashAggregate(keys=[k#33], functions=[sum(cast(v#34 as bigint))])
+- Exchange hashpartitioning(k#33, 200), true, [id=#59]
+- *(1) HashAggregate(keys=[k#33], functions=[partial_sum(cast(v#34 as bigint))])
+- *(1) LocalTableScan [k#33, v#34]
|
+----------------------------------------------------

-- Using Extended
EXPLAIN EXTENDED select k, sum(v) from values (1, 2), (1, 3) t(k, v) group by k;
+----------------------------------------------------+
|                                                plan|
+----------------------------------------------------+
| == Parsed Logical Plan ==
'Aggregate ['k], ['k, unresolvedalias('sum('v), None)]
 +- 'SubqueryAlias `t`
```

```
+- 'UnresolvedInlineTable [k, v], [List(1, 2), List(1, 3)]

 == Analyzed Logical Plan ==
 k: int, sum(v): bigint
 Aggregate [k#47], [k#47, sum(cast(v#48 as bigint)) AS sum(v)#50L]
 +- SubqueryAlias `t`
    +- LocalRelation [k#47, v#48]

 == Optimized Logical Plan ==
 Aggregate [k#47], [k#47, sum(cast(v#48 as bigint)) AS sum(v)#50L]
 +- LocalRelation [k#47, v#48]

 == Physical Plan ==
 *(2) HashAggregate(keys=[k#47], functions=[sum(cast(v#48 as bigint))], output=[k#47,
 sum(v)#50L])
+- Exchange hashpartitioning(k#47, 200), true, [id=#79]
   +- *(1) HashAggregate(keys=[k#47], functions=[partial_sum(cast(v#48 as bigint))],
 output=[k#47, sum#52L])
    +- *(1) LocalTableScan [k#47, v#48]
|
+-----------------------------------------------------+

-- Using Formatted
EXPLAIN FORMATTED select k, sum(v) from values (1, 2), (1, 3) t(k, v) group by k;
+-----------------------------------------------------+
|                                                 plan|
+-----------------------------------------------------+
| == Physical Plan ==
 * HashAggregate (4)
 +- Exchange (3)
    +- * HashAggregate (2)
       +- * LocalTableScan (1)


 (1) LocalTableScan [codegen id : 1]
 Output: [k#19, v#20]

 (2) HashAggregate [codegen id : 1]
 Input: [k#19, v#20]

 (3) Exchange
 Input: [k#19, sum#24L]

 (4) HashAggregate [codegen id : 2]
```

```
 Input: [k#19, sum#24L]
|
+-----------------------------------------------------+
```

LATERAL SUBQUERY 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

LATERAL SUBQUERY 是以關鍵字 開頭的子查詢LATERAL。它提供一種在上述 FROM 子句中參考資料欄的方法。如果沒有LATERAL關鍵字，子查詢只能參考外部查詢中的資料欄，但不能參考 FROM子句中的資料欄。 LATERAL SUBQUERY可讓複雜的查詢更簡單且更有效率。

語法

```
[ LATERAL ] primary_relation [ join_relation ]
```

參數

- primary_relation

  指定主要關係。可為下列其中之一：

  1. 資料表關聯
  2. 別名查詢

     語法: `( query ) [ [ AS ] alias ]`
  3. 別名關係

     `Syntax: ( relation ) [ [ AS ] alias ]`

範例

```
CREATE TABLE t1 (c1 INT, c2 INT);
INSERT INTO t1 VALUES (0, 1), (1, 2);
CREATE TABLE t2 (c1 INT, c2 INT);
```

```
INSERT INTO t2 VALUES (0, 2), (0, 3);
SELECT * FROM t1,
LATERAL (SELECT * FROM t2 WHERE t1.c1 = t2.c1);
+--------+-------+--------+-------+
|  t1.c1 | t1.c2 |  t2.c1 | t2.c2 |
+-------+--------+--------+-------+
|    0   |   1   |    0   |   3   |
|    0   |   1   |    0   |   2   |
+-------+--------+--------+-------+
SELECT a, b, c FROM t1,
LATERAL (SELECT c1 + c2 AS a),
LATERAL (SELECT c1 - c2 AS b),
LATERAL (SELECT a * b AS c);
+--------+-------+--------+
|    a   |   b   |    c   |
+-------+--------+--------+
|    3   |  -1   |   -3   |
|    1   |  -1   |   -1   |
+-------+--------+--------+
```

LATERAL VIEW 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

LATERAL VIEW 子句會與產生器函數搭配使用，例如 EXPLODE，這會產生包含一或多個資料列的虛擬資料表。 LATERAL VIEW會將資料列套用至每個原始輸出資料列。

語法

```
LATERAL VIEW [ OUTER ] generator_function ( expression [ , ... ] ) [ table_alias ] AS
 column_alias [ , ... ]
```

參數

- 外部

  如果OUTER指定，如果輸入陣列/映射為空或 null，則傳回 null。

- generator_function

  指定產生器函數 (EXPLODE、 INLINE等) 。

- table_alias

  的別名generator_function，這是選用的。

- column_alias

  列出 的資料欄別名generator_function，可用於輸出資料列。

  如果 具有多個輸出資料欄，則可以generator_function有多個別名。

範例

```
CREATE TABLE person (id INT, name STRING, age INT, class INT, address STRING);
INSERT INTO person VALUES
(100, 'John', 30, 1, 'Street 1'),
(200, 'Mary', NULL, 1, 'Street 2'),
(300, 'Mike', 80, 3, 'Street 3'),
(400, 'Dan', 50, 4, 'Street 4');
SELECT * FROM person
LATERAL VIEW EXPLODE(ARRAY(30, 60)) tableName AS c_age
LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age;
+------+-------+-------+--------+-----------+--------+--------+
|  id  | name  |  age  | class  |  address  | c_age  | d_age  |
+------+-------+-------+--------+-----------+--------+--------+
| 100  | John  | 30    | 1      | Street 1  | 30     | 40     |
| 100  | John  | 30    | 1      | Street 1  | 30     | 80     |
| 100  | John  | 30    | 1      | Street 1  | 60     | 40     |
| 100  | John  | 30    | 1      | Street 1  | 60     | 80     |
| 200  | Mary  | NULL  | 1      | Street 2  | 30     | 40     |
| 200  | Mary  | NULL  | 1      | Street 2  | 30     | 80     |
| 200  | Mary  | NULL  | 1      | Street 2  | 60     | 40     |
| 200  | Mary  | NULL  | 1      | Street 2  | 60     | 80     |
| 300  | Mike  | 80    | 3      | Street 3  | 30     | 40     |
| 300  | Mike  | 80    | 3      | Street 3  | 30     | 80     |
| 300  | Mike  | 80    | 3      | Street 3  | 60     | 40     |
| 300  | Mike  | 80    | 3      | Street 3  | 60     | 80     |
| 400  | Dan   | 50    | 4      | Street 4  | 30     | 40     |
| 400  | Dan   | 50    | 4      | Street 4  | 30     | 80     |
| 400  | Dan   | 50    | 4      | Street 4  | 60     | 40     |
| 400  | Dan   | 50    | 4      | Street 4  | 60     | 80     |
```

```
+------+-------+-------+--------+-----------+--------+--------+
SELECT c_age, COUNT(1) FROM person
LATERAL VIEW EXPLODE(ARRAY(30, 60)) AS c_age
LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age
GROUP BY c_age;
+--------+-----------+
| c_age  | count(1)  |
+--------+-----------+
| 60     | 8         |
| 30     | 8         |
+--------+-----------+
SELECT * FROM person
LATERAL VIEW EXPLODE(ARRAY()) tableName AS c_age;
+-----+-------+------+--------+----------+--------+
| id  | name  | age  | class  | address  | c_age  |
+-----+-------+------+--------+----------+--------+
+-----+-------+------+--------+----------+--------+
SELECT * FROM person
LATERAL VIEW OUTER EXPLODE(ARRAY()) tableName AS c_age;
+------+-------+-------+--------+-----------+--------+
|  id  | name  |  age  | class  |  address  | c_age  |
+------+-------+-------+--------+-----------+--------+
| 100  | John  | 30    | 1      | Street 1  | NULL   |
| 200  | Mary  | NULL  | 1      | Street 2  | NULL   |
| 300  | Mike  | 80    | 3      | Street 3  | NULL   |
| 400  | Dan   | 50    | 4      | Street 4  | NULL   |
+------+-------+-------+--------+-----------+--------+
```

LIKE 述詞

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

LIKE 述詞用於搜尋特定模式。此述詞也支援具有量化器的多個模式SOME，包括 ANY、 和 ALL。

語法

```
[ NOT ] { LIKE search_pattern [ ESCAPE esc_char ] | [ RLIKE | REGEXP ] regex_pattern }
[ NOT ] { LIKE quantifiers ( search_pattern [ , ... ]) }
```

參數

- search_pattern

  指定 LIKE 子句要搜尋的字串模式。它可以包含特殊模式比對字元：

  - % 符合零個或多個字元。
  - _ 只比對一個字元。
- esc_char

  指定逸出字元。預設逸出字元為 \。
- regex_pattern

  指定要由 RLIKE或 REGEXP子句搜尋的規則運算式搜尋模式。
- 量化器

  指定述詞量化器包括 ANY、 SOME和 ALL。

  ANY 或 SOME表示如果其中一個模式符合輸入，則傳回 true。

  ALL 表示如果所有模式都符合輸入，則傳回 true。

範例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
(300, 'Mike', 80),
(400, 'Dan',  50),
(500, 'Evan_w', 16);
SELECT * FROM person WHERE name LIKE 'M%';
+---+----+----+
| id|name| age|
+---+----+----+
|300|Mike|  80|
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name LIKE 'M_ry';
+---+----+----+
| id|name| age|
+---+----+----+
```

```
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name NOT LIKE 'M_ry';
+---+------+---+
| id|  name|age|
+---+------+---+
|500|Evan_W| 16|
|300|  Mike| 80|
|100|  John| 30|
|400|   Dan| 50|
+---+------+---+
SELECT * FROM person WHERE name RLIKE 'M+';
+---+----+----+
| id|name| age|
+---+----+----+
|300|Mike|  80|
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name REGEXP 'M+';
+---+----+----+
| id|name| age|
+---+----+----+
|300|Mike|  80|
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name LIKE '%\_%';
+---+------+---+
| id|  name|age|
+---+------+---+
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name LIKE '%$_%' ESCAPE '$';
+---+------+---+
| id|  name|age|
+---+------+---+
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name LIKE ALL ('%an%', '%an');
+---+----+----+
| id|name| age|
+---+----+----+
|400| Dan|  50|
+---+----+----+
SELECT * FROM person WHERE name LIKE ANY ('%an%', '%an');
```

```
+---+------+---+
| id|  name|age|
+---+------+---+
|400|   Dan| 50|
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name LIKE SOME ('%an%', '%an');
+---+------+---+
| id|  name|age|
+---+------+---+
|400|   Dan| 50|
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name NOT LIKE ALL ('%an%', '%an');
+---+----+----+
| id|name| age|
+---+----+----+
|100|John|  30|
|200|Mary|null|
|300|Mike|  80|
+---+----+----+
SELECT * FROM person WHERE name NOT LIKE ANY ('%an%', '%an');
+---+------+----+
| id|  name| age|
+---+------+----+
|100|  John|  30|
|200|  Mary|null|
|300|  Mike|  80|
|500|Evan_W|  16|
+---+------+----+
SELECT * FROM person WHERE name NOT LIKE SOME ('%an%', '%an');
+---+------+----+
| id|  name| age|
+---+------+----+
|100|  John|  30|
|200|  Mary|null|
|300|  Mike|  80|
|500|Evan_W|  16|
+---+------+----+
```

OFFSET

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

OFFSET 子句用於指定在開始傳回SELECT陳述式傳回的資料列之前要略過的資料列數。一般而言，此子句會與 搭配使用ORDER BY，以確保結果具有決定性。

語法

```
OFFSET integer_expression
```

參數

- integer_expression

  指定傳回整數的可摺疊表達式。

範例

```
CREATE TABLE person (name STRING, age INT);
INSERT INTO person VALUES
('Jane Doe', 25),
('Pat C', 18),
('Nikki W', 16),
('Juan L', 25),
('John D', 18),
('Jorge S', 16);

-- Skip the first two rows.
SELECT name, age FROM person ORDER BY name OFFSET 2;
+-------+---+
|   name|age|
+-------+---+
| John D| 18|
| Juan L| 25|
|Nikki W| 16|
|Jane Doe| 25|
```

```
+-------+---+

-- Skip the first two rows and returns the next three rows.
SELECT name, age FROM person ORDER BY name LIMIT 3 OFFSET 2;
+-------+---+
|   name|age|
+-------+---+
| John D| 18|
| Juan L| 25|
|Nikki W| 16|
+-------+---+

-- A function expression as an input to OFFSET.
SELECT name, age FROM person ORDER BY name OFFSET length('WAGON');
+-------+---+
|   name|age|
+-------+---+
|Jane Doe| 25|
+-------+---+
```

PIVOT 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

PIVOT 子句用於資料觀點。我們可以根據特定資料欄值取得彙總值，這會轉換為 SELECT子句中使用的多個資料欄。子PIVOT句可以在資料表名稱或子查詢之後指定。

語法

```
PIVOT ( { aggregate_expression [ AS aggregate_expression_alias ] } [ , ... ] FOR
 column_list IN ( expression_list ) )
```

參數

- aggregate_expression

  指定彙總運算式 (SUM(a)、 COUNT(DISTINCT b)等)。
- aggregate_expression_alias

指定彙總運算式的別名。

- column_list

  在 FROM子句中包含資料欄，指定要取代為新資料欄的資料欄。您可以使用括號括住資料欄，例如 (c1, c2)。

- expression_list

  指定新資料欄，用於比對 中的值column_list作為彙總條件。您也可以為其新增別名。

範例

```
CREATE TABLE person (id INT, name STRING, age INT, class INT, address STRING);
INSERT INTO person VALUES
(100, 'John', 30, 1, 'Street 1'),
(200, 'Mary', NULL, 1, 'Street 2'),
(300, 'Mike', 80, 3, 'Street 3'),
(400, 'Dan', 50, 4, 'Street 4');
SELECT * FROM person
PIVOT (
SUM(age) AS a, AVG(class) AS c
FOR name IN ('John' AS john, 'Mike' AS mike)
);
+------+-----------+---------+---------+---------+---------+
|  id  |  address  | john_a  | john_c  | mike_a  | mike_c  |
+------+-----------+---------+---------+---------+---------+
| 200  | Street 2  | NULL    | NULL    | NULL    | NULL    |
| 100  | Street 1  | 30      | 1.0     | NULL    | NULL    |
| 300  | Street 3  | NULL    | NULL    | 80      | 3.0     |
| 400  | Street 4  | NULL    | NULL    | NULL    | NULL    |
+------+-----------+---------+---------+---------+---------+
SELECT * FROM person
PIVOT (
SUM(age) AS a, AVG(class) AS c
FOR (name, age) IN (('John', 30) AS c1, ('Mike', 40) AS c2)
);
+------+-----------+-------+-------+-------+-------+
|  id  |  address  | c1_a  | c1_c  | c2_a  | c2_c  |
+------+-----------+-------+-------+-------+-------+
| 200  | Street 2  | NULL  | NULL  | NULL  | NULL  |
| 100  | Street 1  | 30    | 1.0   | NULL  | NULL  |
| 300  | Street 3  | NULL  | NULL  | NULL  | NULL  |
```

```
| 400  | Street 4  | NULL  | NULL  | NULL  | NULL  |
+------+-----------+-------+-------+-------+-------+
```

設定運算子

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

集合運算子用於將兩個輸入關係合併為單一輸入關係。OpenSearch SQL 支援三種類型的設定運算子：

- EXCEPT 或 MINUS
- INTERSECT
- UNION

輸入關係必須具有相同數量的資料欄，以及個別資料欄的相容資料類型。

EXCEPT

EXCEPT 和 會EXCEPT ALL傳回在一個關係中找到的資料列，但不傳回另一個關係。 EXCEPT (或者，EXCEPT DISTINCT) 只會接受不同的資料列，而 EXCEPT ALL 不會從結果資料列中移除重複的資料列。請注意， MINUS 是 的別名EXCEPT。

語法

```
[ ( ] relation [ ) ] EXCEPT | MINUS [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

範例

```
-- Use table1 and table2 tables to demonstrate set operators in this page.
SELECT * FROM table1;
+---+
|  c|
+---+
|  3|
|  1|
|  2|
```

```
|  2|
|  3|
|  4|
+---+
SELECT * FROM table2;
+---+
|  c|
+---+
|  5|
|  1|
|  2|
|  2|
+---+
SELECT c FROM table1 EXCEPT SELECT c FROM table2;
+---+
|  c|
+---+
|  3|
|  4|
+---+
SELECT c FROM table1 MINUS SELECT c FROM table2;
+---+
|  c|
+---+
|  3|
|  4|
+---+
SELECT c FROM table1 EXCEPT ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  3|
|  3|
|  4|
+---+
SELECT c FROM table1 MINUS ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  3|
|  3|
|  4|
+---+
```

INTERSECT

INTERSECT 和 會INTERSECT ALL傳回在兩個關係中找到的資料列。 INTERSECT (或者，INTERSECT DISTINCT) 只會接受不同的資料列，而 INTERSECT ALL 不會從結果資料列中移除重複的資料列。

語法

```
[ ( ] relation [ ) ] INTERSECT [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

範例

```
(SELECT c FROM table1) INTERSECT (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  2|
+---+
(SELECT c FROM table1) INTERSECT DISTINCT (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  2|
+---+
(SELECT c FROM table1) INTERSECT ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  2|
|  2|
+---+
```

UNION

UNION 和 會UNION ALL傳回在任一關係中找到的資料列。 UNION (或者，UNION DISTINCT) 只會接受不同的資料列，而不會從結果資料列UNION ALL中移除重複的資料列。

語法

```
 [ ( ] relation [ ) ] UNION [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

範例

```
(SELECT c FROM table1) UNION (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  3|
|  5|
|  4|
|  2|
+---+
(SELECT c FROM table1) UNION DISTINCT (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  3|
|  5|
|  4|
|  2|
+---+
SELECT c FROM table1 UNION ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  3|
|  1|
|  2|
|  2|
|  3|
|  4|
|  5|
|  1|
|  2|
|  2|
+---+
```

SORT BY 子句

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

SORT BY 子句用於傳回依使用者指定順序在每個分割區中排序的結果列。當有一個以上的分割區SORT BY可能會傳回部分排序的結果。這與子ORDER BY句不同，子句保證輸出的總順序。

語法

```
SORT BY { expression [ sort_direction | nulls_sort_order ] [ , ... ] }
```

參數

- SORT BY

  指定以逗號分隔的運算式清單，以及選用參數 sort_direction 和 nulls_sort_order，用於排序每個分割區中的資料列。

- sort_direction

  選擇性地指定要以遞增或遞減順序排序資料列。

  排序方向的有效值ASC適用於遞增和DESC遞減。

  如果未明確指定排序方向，則預設資料列會遞增排序。

  語法: [ASC | DESC]

- nulls_sort_order

  選擇性地指定是否在非 NULL 值之前/之後傳回 NULL 值。

  如果null_sort_order未指定 ，則如果排序順序為 ，則 NULLs會先排序，如果排序順序為 ，則 ASC NULLS 排序會最後排序DESC。

  1. 如果指定 NULLS FIRST ，則無論排序順序為何，都會先傳回 NULL 值。

  2. 如果指定 NULLS LAST ，則無論排序順序為何，都會最後傳回 NULL 值。

語法: `[ NULLS { FIRST | LAST } ]`

範例

```
CREATE TABLE person (zip_code INT, name STRING, age INT);
INSERT INTO person VALUES
(94588, 'Shirley Rodriguez', 50),
(94588, 'Juan Li', 18),
(94588, 'Anil K', 27),
(94588, 'John D', NULL),
(94511, 'David K', 42),
(94511, 'Aryan B.', 18),
(94511, 'Lalit B.', NULL);
-- Sort rows by `name` within each partition in ascending manner
SELECT name, age, zip_code FROM person SORT BY name;
+------------------+----+--------+
|              name| age|zip_code|
+------------------+----+--------+
|            Anil K|  27|   94588|
|           Juan Li|  18|   94588|
|            John D|null|   94588|
| Shirley Rodriguez|  50|   94588|
|          Aryan B.|  18|   94511|
|           David K|  42|   94511|
|          Lalit B.|null|   94511|
+------------------+----+--------+
-- Sort rows within each partition using column position.
SELECT name, age, zip_code FROM person SORT BY 1;
+----------------+----+----------+
|              name| age|zip_code|
+------------------+----+--------+
|            Anil K|  27|   94588|
|           Juan Li|  18|   94588|
|            John D|null|   94588|
| Shirley Rodriguez|  50|   94588|
|          Aryan B.|  18|   94511|
|           David K|  42|   94511|
|          Lalit B.|null|   94511|
+------------------+----+--------+

-- Sort rows within partition in ascending manner keeping null values to be last.
SELECT age, name, zip_code FROM person SORT BY age NULLS LAST;
```

```
+----+------------------+--------+
| age|              name|zip_code|
+----+------------------+--------+
|  18|           Juan Li|   94588|
|  27|            Anil K|   94588|
|  50| Shirley Rodriguez|   94588|
|null|            John D|   94588|
|  18|          Aryan B.|   94511|
|  42|           David K|   94511|
|null|          Lalit B.|   94511|
+----+--------------+--------+--------+

-- Sort rows by age within each partition in descending manner, which defaults to NULL
 LAST.
SELECT age, name, zip_code FROM person SORT BY age DESC;
+----+------------------+--------+
| age|              name|zip_code|
+----+------------------+--------+
|  50|          Shirley Rodriguez|   94588|
|  27|            Anil K|   94588|
|  18|           Juan Li|   94588|
|null|            John D|   94588|
|  42|           David K|   94511|
|  18|          Aryan B.|   94511|
|null|          Lalit B.|   94511|
+----+------------------+--------+

-- Sort rows by age within each partition in descending manner keeping null values to
 be first.
SELECT age, name, zip_code FROM person SORT BY age DESC NULLS FIRST;
+----+------------------+--------+
| age|              name|zip_code|
+----+------------------+--------+
|null|            John D|   94588|
|  50| Shirley Rodriguez|   94588|
|  27|            Anil K|   94588|
|  18|           Juan Li|   94588|
|null|          Lalit B.|   94511|
|  42|           David K|   94511|
|  18|          Aryan B.|   94511|
+----+--------------+--------+--------+

-- Sort rows within each partition based on more than one column with each column
 having
```

```
-- different sort direction.
SELECT name, age, zip_code FROM person
SORT BY name ASC, age DESC;
+------------------+----+--------+
|              name| age|zip_code|
+------------------+----+--------+
|            Anil K|  27|   94588|
|           Juan Li|  18|   94588|
|            John D|null|   94588|
| Shirley Rodriguez|  50|   94588|
|          Aryan B.|  18|   94511|
|           David K|  42|   94511|
|          Lalit B.|null|   94511|
+------------------+----+--------+
```

UNPIVOT

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 SQL 命令，請參閱 the section called “支援的 SQL 命令”。

UNPIVOT 子句會將多個資料欄轉換為子SELECT句中使用的多個資料列。子UNPIVOT句可以在資料表名稱或子查詢之後指定。

語法

```
UNPIVOT [ { INCLUDE | EXCLUDE } NULLS ] (
    { single_value_column_unpivot | multi_value_column_unpivot }
) [[AS] alias]

single_value_column_unpivot:
    values_column
    FOR name_column
    IN (unpivot_column [[AS] alias] [, ...])

multi_value_column_unpivot:
    (values_column [, ...])
    FOR name_column
    IN ((unpivot_column [, ...]) [[AS] alias] [, ...])
```

參數

- unpivot_column

  子FROM句中包含資料欄，指定我們要取消樞紐分析的資料欄。

- name_column

  資料欄的名稱，該資料欄會保留未樞紐分析資料欄的名稱。

- values_column

  資料欄的名稱，該資料欄會保留未樞紐分析資料欄的值。

範例

```
CREATE TABLE sales_quarterly (year INT, q1 INT, q2 INT, q3 INT, q4 INT);
INSERT INTO sales_quarterly VALUES
(2020, null, 1000, 2000, 2500),
(2021, 2250, 3200, 4200, 5900),
(2022, 4200, 3100, null, null);
-- column names are used as unpivot columns
SELECT * FROM sales_quarterly
UNPIVOT (
sales FOR quarter IN (q1, q2, q3, q4)
);
+------+---------+-------+
| year | quarter | sales |
+------+---------+-------+
| 2020 | q2      | 1000  |
| 2020 | q3      | 2000  |
| 2020 | q4      | 2500  |
| 2021 | q1      | 2250  |
| 2021 | q2      | 3200  |
| 2021 | q3      | 4200  |
| 2021 | q4      | 5900  |
| 2022 | q1      | 4200  |
| 2022 | q2      | 3100  |
+------+---------+-------+
-- NULL values are excluded by default, they can be included
-- unpivot columns can be alias
-- unpivot result can be referenced via its alias
SELECT up.* FROM sales_quarterly
UNPIVOT INCLUDE NULLS (
```

```
sales FOR quarter IN (q1 AS Q1, q2 AS Q2, q3 AS Q3, q4 AS Q4)
) AS up;
+------+---------+-------+
| year | quarter | sales |
+------+---------+-------+
| 2020 | Q1      | NULL  |
| 2020 | Q2      | 1000  |
| 2020 | Q3      | 2000  |
| 2020 | Q4      | 2500  |
| 2021 | Q1      | 2250  |
| 2021 | Q2      | 3200  |
| 2021 | Q3      | 4200  |
| 2021 | Q4      | 5900  |
| 2022 | Q1      | 4200  |
| 2022 | Q2      | 3100  |
| 2022 | Q3      | NULL  |
| 2022 | Q4      | NULL  |
+------+---------+-------+
-- multiple value columns can be unpivoted per row
SELECT * FROM sales_quarterly
UNPIVOT EXCLUDE NULLS (
(first_quarter, second_quarter)
FOR half_of_the_year IN (
(q1, q2) AS H1,
(q3, q4) AS H2
)
);
+------+------------------+---------------+----------------+
|  id  | half_of_the_year | first_quarter | second_quarter |
+------+------------------+---------------+----------------+
| 2020 | H1               | NULL          | 1000           |
| 2020 | H2               | 2000          | 2500           |
| 2021 | H1               | 2250          | 3200           |
| 2021 | H2               | 4200          | 5900           |
| 2022 | H1               | 4200          | 3100           |
+------+------------------+---------------+----------------+
```

支援的 PPL 命令

下表顯示 OpenSearch Dashboards 支援哪些 PPL 命令來查詢 CloudWatch Logs、Amazon S3 或 Security Lake，以及 CloudWatch Logs Insights 支援哪些命令。在查詢 CloudWatch Logs 時，CloudWatch Logs Insights 使用與 OpenSearch Dashboards 相同的 PPL 語法，而資料表都稱為 CloudWatch Logs。

> **Note**
>
> 當您分析 OpenSearch Service 外部的資料時，命令的執行方式可能與 OpenSearch 索引不同。

主題

- 命令
- 函數
- 使用 OpenSearch PPL 的 CloudWatch Logs Insights 使用者其他資訊

命令

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “fields”	顯示一組需要投影的欄位。	支援的	支援的	支援的	`fields field1, field2`
the section called “where”	根據您指定的條件篩選資料。	支援的	支援的	支援的	`where field1="success" | where field2 != "i-023fe0a90929d8822" | fields field3, col4, col5, col6 | head 1000`

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “統計資料”	執行彙總和計算。	支援的	支援的	支援的	`stats count(), count(`field1`), min(`field1`), max(`field1`), avg(`field1`) by field2 | head 1000`
the section called “parse”	從字串擷取規則表達式 (regex) 模式，並顯示擷取的模式。擷取的模式可以進一步用來建立新的欄位或篩選資料。	支援的	支援的	支援的	`parse `field1` ".*/(?<field2>[^/]+$)" | where field2 = "requestId" | fields field2, `field2` | head 1000`

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “模式”	從文字欄位擷取日誌模式，並將結果附加到搜尋結果。依模式分組日誌可讓您更輕鬆地從大量日誌資料彙總統計資料，以進行分析和故障診斷。	不支援	支援的	支援的	`patterns new_field='no_numbers' pattern='[0-9]' message | fields message, no_numbers`
the section called “sort”	依欄位名稱排序顯示的結果。使用 sort -FieldName 以遞減順序排序。	支援的	支援的	支援的	`stats count(), count(`field1`), min(`field1`) as field1Alias, max(`field1`), avg(`field1`) by field2 | sort -field1Alias | head 1000`

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “評估”	修改或處理欄位的值，並將其存放在不同的欄位中。這有助於以數學方式修改資料欄、將字串函數套用至資料欄，或將日期函數套用至資料欄。	支援的	支援的	支援的	`eval field2 = `field1` * 2 | fields field1, field2 | head 20`
the section called “重新命名”	重新命名搜尋結果中的一個或多個欄位。	支援的	支援的	支援的	`rename field2 as field1 | fields field1`
the section called “head”	將顯示的查詢結果限制為前 N 列。	支援的	支援的	支援的	`fields `@message` | head 20`
the section called “grok”	根據規則表達式剖析具有 grok 模式的文字欄位，並將結果附加到搜尋結果。	支援的	支援的	支援的	`grok email '.+@%{HOSTNAME:host}' | fields email`
the section called “top”	尋找欄位最常見的值。	支援的	支援的	支援的	`top 2 Field1 by Field2`

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “dedup”	根據您指定的欄位移除重複的項目。	支援的	支援的	支援的	`dedup field1 | fields field1, field2, field3`
the section called “join”	將兩個資料集聯結在一起。	支援的	支援的	支援的	`source=customer | join ON c_custkey = o_custkey orders | head 10`
the section called “查詢”	新增或取代查詢索引（維度表）中的資料，以豐富您的搜尋資料。您可以使用維度資料表中的值來擴展索引的欄位，並在符合查詢條件時附加或取代值	不支援	支援的	支援的	`where orderType = 'Cancelled' | lookup account_list, mkt_id AS mkt_code replace amount, account_name as name | stats count(mkt_code), avg(amount) by name`

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “subquery”	在 Piped Processing Language (PPL) 陳述式中執行複雜的巢狀查詢。	支援的	支援的	支援的	`where id in [ subquery source=users | where user in [ subquery source=actions | where action="login" | fields user ] | fields uid ]`
the section called “很少見”	尋找欄位清單中所有欄位頻率最低的值。	支援的	支援的	支援的	`rare Field1 by Field2`
the section called “趨勢線”	計算欄位的移動平均值。	支援的	支援的	支援的	`trendline sma(2, field1) as field1Alias`

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “eventstats”	使用計算的摘要統計資料來豐富您的事件資料。它會分析事件中的指定欄位、運算各種統計指標，然後將這些結果做為新欄位附加到每個原始事件。	支援（除外`count()`）	支援的	支援的	`eventstats sum(field1) by field2`
the section called “flatten”	扁平化欄位，欄位必須為此類型：`struct<?,?> or array<struct<?,?>>`	支援的	支援的	支援的	`source=table | flatten field1`
the section called “欄位摘要”	計算每個欄位的基本統計資料（計數、相異計數、最小值、最大值、平均值、標準差和平均值）。	支援（每個查詢一個欄位）	支援的	支援的	`where field1 != 200 | fieldsummary includefields=field1 nulls=true`

PPL 命令	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “fillnull”	使用您提供的值填入 null 欄位。它可用於一個或多個欄位。	支援的	支援的	支援的	`fields field1 | eval field2=field1 | fillnull value=0 field1`
the section called “擴充”	將包含多個值的欄位細分為不同的資料列，為指定欄位中的每個值建立新的資料列。	支援的	支援的	支援的	`expand employee | stats max(salary) as max by state, company`
the section called “describe”	取得資料表、結構描述和目錄結構和中繼資料的詳細資訊	不支援	支援的	支援的	`describe schema.table`

函數

PPL 函數	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “String” (CONCAT, CONCAT_WS ，LENGTH, LOWER,	PPL 中的內建函數，可以操作和轉換 PPL 查詢中的字	支援的	支援的	支援的	`eval col1Len = LENGTH(col1) | fields col1Len`

PPL 函數	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
LTRIM, POSITION, REVERSE, RIGHT, RTRIM, SUBSTRING, TRIM, UPPER)	串和文字資料。例如，轉換大小寫、合併字串、擷取部分和清除文字。				

PPL 函數	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “日期和時間” (DAY, DAYOFMONTH , DAY_OF_MONTH ,DAYOFWEEK , DAY_OF_WEEK , DAYOFYEAR , DAY_OF_YEAR , DAYNAME, FROM_UNIXTIME , HOUR, HOUR_OF_DAY , LAST_DAY, LOCALTIMESTAMP , LOCALTIME , MAKE_DATE , MINUTE, MINUTE_OF_HOUR , MONTH, MONTHNAME , MONTH_OF_YEAR , NOW, QUARTER, SECOND, SECOND_OF_MINUTE , SUBDATE, SYSDATE, TIMESTAMP , UNIX_TIMESTAMP , WEEK, WEEKDAY, WEEK_OF_YEAR , DATE_ADD,	用於處理和轉換 PPL 查詢中日期和時間戳記資料的內建函數。例如 date_add、at、datediff 和 current_date。	支援的	支援的	支援的	`eval newDate = ADDDATE(DATE('2020-08-26'), 1) | fields newDate`

PPL 函數	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
DATE_SUB, TIMESTAMP ADD , TIMESTAMP DIFF , UTC_TIMESTAMP , CURRENT_TIMEZONE)					
the section called “條件” (EXISTS, IF, IFNULL, ISNOTNULL , ISNULL, NULLIF)	對多個資料列執行計算以產生單一摘要值的內建函數。例如，總和、計數、平均值、最大值和最小值。	支援的	支援的	支援的	`eval field2 = isnull(col1) | fields field2, col1, field3`

PPL 函數	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “Math” (ABS, ACOS, ASIN, ATAN, ATAN2, CEIL, CEILING, CONV, COS, COT, CRC32, DEGREES, E, EXP, FLOOR, LN, LOG, LOG2, LOG10, MOD, PI. POW, POWER, RADIANS, RAND, ROUND, SIGN, SIN, SQRT, CBRT)	用於在 PPL 查詢中執行數學計算和轉換的內建函數。例如：abs（絕對值）、圓（圓數）、sqrt（平方根）、pow（功率計算）和 ceil（四捨五入至最接近的整數）。	支援的	支援的	支援的	`eval field2 = ACOS(col1) | fields col1`
the section called “表達式” （算術運算子 (+、、*)-、述詞運算子 (>. <、IN))	表達式的內建函數，特別是值表達式，會傳回純量值。表達式有不同的類型和形式。	支援的	支援的	支援的	`where age > (25 + 5) | fields age`

PPL 函數	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “IP 位址” (CIDRMATCH)	用於處理 CIDR 等 IP 地址的內建函數。	支援的	支援的	支援的	`where cidrmatch(ip, '********/24') | fields ip`
the section called “JSON” (ARRAY_LENGTH , ARRAY_LENGTH , JSON, JSON_ARRAY , JSON_EXTRACT , JSON_KEYS , JSON_OBJECT , JSON_VALID , TO_JSON_STRING)	處理 JSON 的內建函數，包括陣列、擷取和驗證。	支援的	支援的	支援的	`eval` `json_extract('{"a":"b"}', '$.a')` = `json_extract('{"a":"b"}', '$a')`
the section called “Lambda” (EXISTS, FILTER, REDUCE, TRANSFORM)	處理 JSON 的內建函數，包括陣列、擷取和驗證。	不支援	支援的	支援的	`eval array = json_array(1, -1, 2), result = filter(array, x -> x > 0) | fields result`

PPL 函數	Description	CloudWatch Logs	Amazon S3	Security Lake	範例 命令
the section called “密碼編譯” (MD5, SHA1, SHA2)	內建函數，可讓您產生唯一的資料指紋，可用於驗證、比較或作為更複雜安全通訊協定的一部分。	支援的	支援的	支援的	`eval` `MD5('hello')` = MD5('hello') | fields `MD5('hello')`

使用 OpenSearch PPL 的 CloudWatch Logs Insights 使用者其他資訊

雖然 CloudWatch Logs Insights 支援大多數 OpenSearch PPL 命令和函數，但目前不支援某些命令和函數。例如，它目前不支援 PPL 中的查詢命令。自 2025 年 6 月 2 日起，CloudWatch Logs Insights 現在支援 PPL 中的 JOIN、子查詢、 Flatten、 Fillnull、Expand、Cidrmatch 和 JSON 函數。如需支援查詢命令和函數的完整清單，請參閱上表中的 Amazon CloudWatch Logs 欄。

查詢和配額範例

以下適用於查詢 CloudWatch 資料的 CloudWatch Logs Insights 使用者和 OpenSearch 使用者。

如需有關從 OpenSearch Service 查詢 CloudWatch Logs 時套用限制的資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的 CloudWatch Logs 配額。 Amazon CloudWatch 限制包括您可以查詢的 CloudWatch Log 群組數量、您可以執行的並行查詢數量上限、查詢執行時間上限，以及結果中傳回的資料列數量上限。無論您使用哪種語言查詢 CloudWatch Logs （即 OpenSearch PPL、SQL 和 Logs Insights QL），限制都相同。

PPL 命令

主題

- comment
- 相互關聯命令

- dedup 命令
- describe 命令
- eval 命令
- eventstats 命令
- expand 命令
- explain 命令
- fillnull 命令
- field 命令
- 平面命令
- grok 命令
- head 命令
- join 命令
- 查詢命令
- 剖析命令
- 模式命令
- 罕見命令
- 重新命名命令
- 搜尋命令
- sort 命令
- stats 命令
- 子查詢命令
- top 命令
- 趨勢線命令
- 其中 命令
- 欄位摘要
- expand 命令
- PPL 函數

comment

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

PPL 同時支援行註解和區塊註解。系統不會評估評論文字。

行註解

行註解以兩個斜線 // 開頭，並以新行結尾。

範例：

```
os> source=accounts | top gender // finds most common gender of all the accounts
fetched rows / total rows = 2/2
+----------+
| gender   |
|----------|
| M        |
| F        |
+----------+
```

區塊註解

區塊註解開頭為斜線後接星號 *，結尾為星號後接斜線 */。

範例：

```
os> source=accounts | dedup 2 gender /* dedup the document with gender field keep 2
 duplication */ | fields account_number, gender
fetched rows / total rows = 3/3
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 6                | M        |
| 13               | F        |
+------------------+----------+
```

相互關聯命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

您可以根據常見的維度和時間範圍來關聯不同的資料來源。

當您處理來自共用相同時段但未正式同步之各種垂直的大量資料時，這種相互關聯至關重要。

透過根據時間範圍和類似維度來關聯這些不同的資料來源，您可以充實資料並發現寶貴的洞見。

範例

可觀測性網域有三種不同的資料來源：

- 日誌
- 指標
- 追蹤

這些資料來源可能會共用共同維度。若要從一個資料來源轉換到另一個資料來源，您需要正確建立關聯。使用語意命名慣例，您可以識別跨日誌、追蹤和指標的共用元素。

範例：

```
{
  "@timestamp": "2018-07-02T22:23:00.186Z",
  "aws": {
    "elb": {
      "backend": {
        "http": {
          "response": {
            "status_code": 500
          }
        },
        "ip": "********",
        "port": "80"
      },
      ...
     "target_port": [
```

```
        "10.0.0.1:80"
      ],
      "target_status_code": [
        "500"
      ],
      "traceId": "Root=1-58337262-36d228ad5d99923122bbe354",
      "type": "http"
    }
  },
  "cloud": {
    "provider": "aws"
  },
  "http": {
    "request": {
    ...
  },
  "communication": {
    "source": {
      "address": "**************",
      "ip": "**************",
      "port": 2817
    }
  },
  "traceId": "Root=1-58337262-36d228ad5d99923122bbe354"
}
```

此範例顯示來自位於 AWS 上之服務的 ELB 日誌 AWS。它會顯示狀態碼為 500 的後端 HTTP 回應，指出錯誤。這可能會觸發警示或成為您定期監控程序的一部分。您的下一個步驟是收集有關此事件的相關資料，以進行徹底的調查。

雖然您可能會想查詢與時間範圍相關的所有資料，但這種方法可能會令人不知所措。最終您可能會得到太多的資訊，花更多時間篩選掉不相關的資料，而不是識別根本原因。

反之，您可以透過關聯來自不同來源的資料，來使用更有針對性的方法。您可以使用這些維度進行相互關聯：

- IP - `"ip": "10.0.0.1" | "ip": "***************"`
- 連接埠 - `"port": 2817 | "target_port": "10.0.0.1:80"`

假設您可以存取其他追蹤和指標索引，並且熟悉結構描述結構，則可以建立更精確的相互關聯查詢。

以下是追蹤索引文件的範例，其中包含您可能想要關聯的 HTTP 資訊：

```
{
  "traceId": "c1d985bd02e1dbb85b444011f19a1ecc",
  "spanId": "55a698828fe06a42",
  "traceState": [],
  "parentSpanId": "",
  "name": "mysql",
  "kind": "CLIENT",
  "@timestamp": "2021-11-13T20:20:39+00:00",
  "events": [
    {
      "@timestamp": "2021-03-25T17:21:03+00:00",
       ...
    }
  ],
  "links": [
    {
      "traceId": "c1d985bd02e1dbb85b444011f19a1ecc",
      "spanId": "55a698828fe06a42w2",
      },
      "droppedAttributesCount": 0
    }
  ],
  "resource": {
    "service@name": "database",
    "telemetry@sdk@name": "opentelemetry",
    "host@hostname": "ip-172-31-10-8.us-west-2.compute.internal"
  },
  "status": {
    ...
  },
  "attributes": {
    "http": {
      "user_agent": {
        "original": "Mozilla/5.0"
      },
      "network": {
         ...
        }
      },
      "request": {
         ...
        }
      },
```

```
      "response": {
        "status_code": "200",
        "body": {
          "size": 500
        }
      },
      "client": {
        "server": {
          "socket": {
            "address": "***********",
            "domain": "example.com",
            "port": 80
          },
          "address": "***********",
          "port": 80
        },
        "resend_count": 0,
        "url": {
          "full": "http://example.com"
        }
      },
      "server": {
        "route": "/index",
        "address": "***********",
        "port": 8080,
        "socket": {
         ...
        },
        "client": {
         ...
         }
        },
        "url": {
         ...
        }
      }
    }
  }
}
```

在此方法中，您可以看到ip與 elb 日誌建立關聯的 `traceId`和 http 的用戶端/伺服器，以進一步了解系統的行為和條件。

新的相互關聯查詢命令

以下是允許此類調查的新命令：

```
source alb_logs, traces | where alb_logs.ip="10.0.0.1" AND
 alb_logs.cloud.provider="aws"|
correlate exact fields(traceId, ip) scope(@timestamp, 1D) mapping(alb_logs.ip =
 traces.attributes.http.server.address, alb_logs.traceId = traces.traceId )
```

以下是命令的每個部分的功能：

1. `source alb_logs, traces` - 這會選取您要關聯的資料來源。
2. `where ip="10.0.0.1" AND cloud.provider="aws"` - 這會縮小搜尋範圍。
3. `correlate exact fields(traceId, ip)` - 這會通知系統根據下列欄位的完全相符項目來關聯資料：
   - `ip` 欄位具有明確的篩選條件，因此會用於所有資料來源的相互關聯。
   - `traceId` 欄位沒有明確的篩選條件，因此會比對所有資料來源的相同 traceIds。

欄位名稱表示相互關聯命令內函數的邏輯意義。實際聯結條件取決於您提供的映射陳述式。

術語`exact`表示相互關聯陳述式需要所有欄位相符，才能滿足查詢陳述式。

該術語`approximate`將嘗試在最佳案例案例中進行比對，並且不會拒絕具有部分比對的資料列。

處理不同的欄位映射

如果同一個邏輯欄位 （ 例如 `ip`) 在資料來源中有不同的名稱，您需要提供路徑欄位的明確映射。若要解決此問題，您可以擴展相互關聯條件，以符合具有類似邏輯意義的不同欄位名稱。以下是您可以執行此操作的方式：

```
alb_logs.ip = traces.attributes.http.server.address, alb_logs.traceId = traces.traceId
```

對於參與相互關聯聯結的每個欄位，您應該提供相關的映射陳述式，其中包含此相互關聯命令要聯結的所有資料表。

範例

在此範例中，有 2 個來源： `alb_logs, traces`

有 2 個欄位： `traceId, ip`

有兩個映射陳述式： `alb_logs.ip = traces.attributes.http.server.address, alb_logs.traceId = traces.traceId`

調整相互關聯時間範圍

若要簡化執行引擎 （驅動程式） 完成的工作，您可以新增範圍陳述式。這會明確指示聯結查詢應為此搜尋範圍的時間。

```
scope(@timestamp, 1D)i
```

在此範例中，搜尋範圍著重於每日，因此在同一天出現的相互關聯會分組在一起。這種範圍界定機制可簡化並允許更好地控制結果，從而根據您的需求實現增量搜尋解決方案。

支援驅動程式

新的相互關聯命令實際上是「隱藏」聯結命令。因此，只有下列 PPL 驅動程式支援此命令。在這些驅動程式中，相互關聯命令會直接翻譯為適當的 Catalyst Join 邏輯計畫。

範例

```
source alb_logs, traces, metrics | where ip="10.0.0.1" AND
cloud.provider="aws"| correlate exact on (ip, port) scope(@timestamp,
2018-07-02T22:23:00, 1 D)
```

邏輯計畫：

```
'Project [*]
+- 'Join Inner, ('ip && 'port)
   :- 'Filter (('ip === "10.0.0.1" & 'cloud.provider === "aws") &
 inTimeScope('@timestamp, "2018-07-02T22:23:00", "1 D"))
      +- 'UnresolvedRelation [alb_logs]
   +- 'Join Inner, ('ip & 'port)
      :- 'Filter (('ip === "10.0.0.1" & 'cloud.provider === "aws") &
 inTimeScope('@timestamp, "2018-07-02T22:23:00", "1 D"))
         +- 'UnresolvedRelation [traces]
      +- 'Filter (('ip === "10.0.0.1" & 'cloud.provider === "aws") &
 inTimeScope('@timestamp, "2018-07-02T22:23:00", "1 D"))
         +- 'UnresolvedRelation [metrics]
```

觸媒引擎會根據最有效率的聯結順序來最佳化此查詢。

dedup 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 dedup命令，根據指定的欄位從搜尋結果中移除相同的文件。

語法

使用下列語法：

```
dedup [int] <field-list> [keepempty=<bool>] [consecutive=<bool>]
```

`int`

- 選用。
- 當您指定 <int> 時，dedup命令會保留每個組合的多個事件。<int> 的數字必須大於 0。如果您未指定數字，只會保留第一個發生的事件。所有其他重複項目都會從結果中移除。
- 預設：1

`keepempty`

- 選用。
- 如果為 true， 會保留欄位清單中任何欄位具有 NULL 值或 MISSING 的文件。
- 預設：false

`consecutive`

- 選用。
- 如果為 true， 只會移除具有連續重複值組合的事件。
- 預設：false

`field-list`

- 強制性。

- 欄位的逗號分隔清單。至少需要一個欄位。

範例 1：除以一個欄位

此範例說明如何使用性別欄位對文件進行刪除。

PPL 查詢：

```
os> source=accounts | dedup gender | fields account_number, gender;
fetched rows / total rows = 2/2
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 13               | F        |
+------------------+----------+
```

範例 2：保留 2 個重複的文件

此範例示範如何使用性別欄位刪除文件，並保留兩個重複項目。

PPL 查詢：

```
os> source=accounts | dedup 2 gender | fields account_number, gender;
fetched rows / total rows = 3/3
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 6                | M        |
| 13               | F        |
+------------------+----------+
```

範例 3：預設保留或忽略空白欄位

此範例示範如何透過保留 null 值欄位來刪除文件。

PPL 查詢：

```
os> source=accounts | dedup email keepempty=true | fields account_number, email;
fetched rows / total rows = 4/4
```

```
+------------------+-----------------------+
| account_number   | email                 |
+------------------+-----------------------+
| 1                | john_doe@example.com  |
| 6                | jane_doe@example.com  |
| 13               | null                  |
| 18               | juan_li@example.com   |
+------------------+-----------------------+
```

此範例示範如何忽略空值欄位，對文件進行資料刪除。

PPL 查詢：

```
os> source=accounts | dedup email | fields account_number, email;
fetched rows / total rows = 3/3
+------------------+-----------------------+
| account_number   | email                 |
+------------------+-----------------------+
| 1                | john_doe@example.com  |
| 6                | jane_doe@example.com  |
| 18               | juan_li@example.com   |
+------------------+-----------------------+
```

範例 4：連續文件中的 Dedup

此範例示範如何在連續文件中進行資料刪除。

PPL 查詢：

```
os> source=accounts | dedup gender consecutive=true | fields account_number, gender;
fetched rows / total rows = 3/3
+------------------+----------+
| account_number   | gender   |
+------------------+----------+
| 1                | M        |
| 13               | F        |
| 18               | M        |
+------------------+----------+
```

其他範例

- `source = table | dedup a | fields a,b,c`

- source = table | dedup a,b | fields a,b,c
- source = table | dedup a keepempty=true | fields a,b,c
- source = table | dedup a,b keepempty=true | fields a,b,c
- source = table | dedup 1 a | fields a,b,c
- source = table | dedup 1 a,b | fields a,b,c
- source = table | dedup 1 a keepempty=true | fields a,b,c
- source = table | dedup 1 a,b keepempty=true | fields a,b,c
- source = table | dedup 2 a | fields a,b,c
- source = table | dedup 2 a,b | fields a,b,c
- source = table | dedup 2 a keepempty=true | fields a,b,c
- source = table | dedup 2 a,b keepempty=true | fields a,b,c
- source = table | dedup 1 a consecutive=true| fields a,b,c (不支援連續重複資料刪除)

限制

- 針對 | dedup 2 a, b keepempty=false

```
DataFrameDropColumns('_row_number_)
+- Filter ('_row_number_ <= 2) // allowed duplication = 2
   +- Window [row_number() windowspecdefinition('a, 'b, 'a ASC NULLS FIRST, 'b ASC
 NULLS FIRST, specifiedwindowframe(RowFrame, unboundedpreceding$(), currentrow$()))
 AS _row_number_], ['a, 'b], ['a ASC NULLS FIRST, 'b ASC NULLS FIRST]
      +- Filter (isnotnull('a) AND isnotnull('b)) // keepempty=false
         +- Project
            +- UnresolvedRelation
```

- 針對 | dedup 2 a, b keepempty=true

```
Union
:- DataFrameDropColumns('_row_number_)
:  +- Filter ('_row_number_ <= 2)
:     +- Window [row_number() windowspecdefinition('a, 'b, 'a ASC NULLS FIRST, 'b ASC
 NULLS FIRST, specifiedwindowframe(RowFrame, unboundedpreceding$(), currentrow$()))
 AS _row_number_], ['a, 'b], ['a ASC NULLS FIRST, 'b ASC NULLS FIRST]
:        +- Filter (isnotnull('a) AND isnotnull('b))
:           +- Project
```

```
:              +- UnresolvedRelation
+- Filter (isnull('a) OR isnull('b))
   +- Project
      +- UnresolvedRelation
```

describe 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 `describe`命令取得資料表、結構描述和目錄結構和中繼資料的詳細資訊。以下是 `describe`命令的各種範例和使用案例。

描述

- `describe table` 此命令等於 `DESCRIBE EXTENDED table` SQL 命令
- `describe schema.table`
- `describe schema.`table``
- `describe catalog.schema.table`
- `describe catalog.schema.`table``
- `describe `catalog`.`schema`.`table``

eval 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

`eval` 命令會評估表達式，並將結果附加到搜尋結果。

語法

使用下列語法：

```
eval <field>=<expression> ["," <field>=<expression> ]...
```

- field：強制性。如果欄位名稱不存在，則會新增新的欄位。如果欄位名稱已存在，則會覆寫它。
- expression：強制性。系統支援的任何表達式。

範例 1：建立新欄位

此範例說明如何為每個文件建立新doubleAge欄位。新的 doubleAge是年齡乘以 2 的評估結果。

PPL 查詢：

```
os> source=accounts | eval doubleAge = age * 2 | fields age, doubleAge ;
fetched rows / total rows = 4/4
+-------+-------------+
| age   | doubleAge   |
|-------+-------------|
| 32    | 64          |
| 36    | 72          |
| 28    | 56          |
| 33    | 66          |
+-------+-------------+
```

範例 2：覆寫現有欄位

此範例示範如何使用年齡加 1 覆寫現有的年齡欄位。

PPL 查詢：

```
os> source=accounts | eval age = age + 1 | fields age ;
fetched rows / total rows = 4/4
+-------+
| age   |
|-------|
| 33    |
| 37    |
| 29    |
| 34    |
+-------+
```

範例 3：使用 eval 中定義的欄位建立新的欄位

此範例說明如何使用 eval 命令中定義的ddAge欄位建立新的欄位。新欄位ddAge是 的評估結果doubleAge乘以 2，其中 doubleAge 是在 eval 命令中定義的。

PPL 查詢：

```
os> source=accounts | eval doubleAge = age * 2, ddAge = doubleAge * 2 | fields age,
 doubleAge, ddAge ;
fetched rows / total rows = 4/4
+-------+-------------+---------+
| age   | doubleAge   | ddAge   |
|-------+-------------+---------|
| 32    | 64          | 128     |
| 36    | 72          | 144     |
| 28    | 56          | 112     |
| 33    | 66          | 132     |
+-------+-------------+---------+
```

假設：a、b、 c是 中的現有欄位 table

其他範例

- source = table | eval f = 1 | fields a,b,c,f
- source = table | eval f = 1 (輸出 a、b、c、f 欄位)
- source = table | eval n = now() | eval t = unix_timestamp(a) | fields n,t
- source = table | eval f = a | where f > 1 | sort f | fields a,b,c | head 5
- source = table | eval f = a * 2 | eval h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = b | stats avg(f) by h
- source = table | eval f = ispresent(a)
- source = table | eval r = coalesce(a, b, c) | fields r
- source = table | eval e = isempty(a) | fields e
- source = table | eval e = isblank(a) | fields e
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one', a = 2, 'two', a = 3, 'three', a = 4, 'four', a = 5, 'five', a = 6, 'six', a = 7, 'se7en', a = 8, 'eight', a = 9, 'nine')

- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else 'unknown')
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else concat(a, ' is an incorrect binary digit'))
- source = table | eval f = a in ('foo', 'bar') | fields f
- source = table | eval f = a not in ('foo', 'bar') | fields f

評估與案例範例：

```
source = table | eval e = eval status_category =
case(a >= 200 AND a < 300, 'Success',
a >= 300 AND a < 400, 'Redirection',
a >= 400 AND a < 500, 'Client Error',
a >= 500, 'Server Error'
else 'Unknown')
```

使用另一個案例評估範例：

假設：a、b、 c是 中的現有欄位 table

其他範例

- source = table | eval f = 1 | fields a,b,c,f
- source = table | eval f = 1 (輸出 a、b、c、f 欄位)
- source = table | eval n = now() | eval t = unix_timestamp(a) | fields n,t
- source = table | eval f = a | where f > 1 | sort f | fields a,b,c | head 5
- source = table | eval f = a * 2 | eval h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = b | stats avg(f) by h
- source = table | eval f = ispresent(a)
- source = table | eval r = coalesce(a, b, c) | fields r
- source = table | eval e = isempty(a) | fields e
- source = table | eval e = isblank(a) | fields e

- source = table | eval f = case(a = 0, 'zero', a = 1, 'one', a = 2, 'two', a = 3, 'three', a = 4, 'four', a = 5, 'five', a = 6, 'six', a = 7, 'se7en', a = 8, 'eight', a = 9, 'nine')
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else 'unknown')
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else concat(a, ' is an incorrect binary digit'))
- source = table | eval f = a in ('foo', 'bar') | fields f
- source = table | eval f = a not in ('foo', 'bar') | fields f

評估與案例範例：

```
source = table | eval e = eval status_category =
case(a >= 200 AND a < 300, 'Success',
a >= 300 AND a < 400, 'Redirection',
a >= 400 AND a < 500, 'Client Error',
a >= 500, 'Server Error'
else 'Unknown')
```

使用另一個案例評估範例：

```
source = table |  where ispresent(a) |
eval status_category =
 case(a >= 200 AND a < 300, 'Success',
  a >= 300 AND a < 400, 'Redirection',
  a >= 400 AND a < 500, 'Client Error',
  a >= 500, 'Server Error'
  else 'Incorrect HTTP status code'
 )
 | stats count() by status_category
```

限制

- 不支援覆寫現有欄位。嘗試這樣做的查詢會擲回例外狀況，並顯示「參考「a」不明確」訊息。

```
- `source = table | eval a = 10 | fields a,b,c`
- `source = table | eval a = a * 2 | stats avg(a)`
- `source = table | eval a = abs(a) | where a > 0`
```

```
- `source = table | eval a = signum(a) | where a < 0`
```

eventstats 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 `eventstats`命令，透過計算的摘要統計資料來豐富您的事件資料。它的運作方式是分析事件中的指定欄位、計算各種統計指標，然後將這些結果做為新欄位附加到每個原始事件。

eventstats 的關鍵層面

1. 它會在整個結果集或在定義的群組中執行計算。
2. 原始事件保持不變，新增新欄位以包含統計結果。
3. 命令對於比較分析、識別極端值或為個別事件提供額外的內容特別有用。

統計資料和 eventstats 之間的差異

`stats` 和 `eventstats`命令都用於計算統計資料，但它們的運作方式和產生的內容有一些關鍵差異。

輸出格式

- `stats`：產生只有計算統計資料的摘要資料表。
- `eventstats`：將計算的統計資料新增為現有事件的新欄位，保留原始資料。

事件保留

- `stats`：將結果設定為僅統計摘要，捨棄個別事件。
- `eventstats`：保留所有原始事件，並新增具有計算統計資料的新欄位。

使用案例

- `stats`：最適合建立摘要報告或儀表板。通常用作總結結果的最終命令。
- `eventstats`：當您需要以統計內容豐富事件以進行進一步分析或篩選時很有用。可用於搜尋中，以新增可用於後續命令的統計資料。

語法

使用下列語法：

```
eventstats <aggregation>... [by-clause]
```

聚合

- 強制性。
- 彙總函數。
- 彙總的引數必須是 欄位。

依 -clause

- 選用。
- 語法: by [span-expression,] [field,]...
- 依子句可包含純量函數和彙總函數等欄位和表達式。您也可以使用跨度子句，將特定欄位分割為相等間隔的儲存貯體。eventstats 命令接著會根據這些跨度儲存貯體執行彙總。
- 預設：如果您未依子句指定 ，則 eventstats 命令會彙總整個結果集。

span-expression

- 選用，最多一個。
- 語法: span(field_expr, interval_expr)
- 間隔表達式的單位預設為自然單位。不過，對於日期和時間類型欄位，您需要在使用日期/時間單位時，在間隔表達式中指定單位。

  例如，若要將欄位分割age為儲存貯體 10 年，請使用 span(age, 10)。對於以時間為基礎的欄位，您可以使用 將timestamp欄位分割為每小時間隔span(timestamp, 1h)。

可用的時間單位

跨度間隔單位
毫秒 (ms)
秒 （秒）

跨度間隔單位
分鐘 (m、區分大小寫)
小時 (h)
天 (d)
週 (w)
月 (M、區分大小寫)
季度 (q)
年 (y)

彙總函數

COUNT

COUNT 會傳回 SELECT 陳述式所擷取資料列中的 expr 數量計數。

對於 CloudWatch Logs 使用查詢，COUNT不支援 。

範例：

```
os> source=accounts | eventstats count();
fetched rows / total rows = 4/4
+----------------+---------+-----------+----------+-----+--------+-------------------
+------------+-------------------------+--------+-------+---------+
| account_number | balance  | firstname | lastname | age | gender | address
 | employer   | email                    | city   | state | count() |
+----------------+---------+-----------+----------+-----+--------+-------------------
+------------+-------------------------+--------+-------+---------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
 | AnyCorp    | janedoe@anycorp.com      | Brogan | IL    | 4       |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
 | AnyCompany | marymajor@anycompany.com | Dante  | TN    | 4       |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
 | AnyOrg     |                          | Nogal  | VA    | 4       |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
 |            | juanli@exampleorg.com    | Orick  | MD    | 4       |
```

```
+----------------+----------+-----------+----------+-----+--------+--------------------
+------------+--------------------------+--------+-------+---------+
```

SUM

SUM(expr) 傳回 expr 的總和。

範例：

```
os> source=accounts | eventstats sum(age) by gender;
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer   | email                    | city   | state | sum(age) by gender |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp    | janedoe@anycorp.com      | Brogan | IL    | 101                |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | AnyCompany | marymajor@anycompany.com | Dante  | TN    | 101                |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg     |                          | Nogal  | VA    | 28                 |
| 18             | 4180     | Juan      | Li       | 33  | M      | 467 Example Court
   |            | juanli@exampleorg.com    | Orick  | MD    | 101                |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
```

AVG

AVG(expr) 傳回 expr 的平均值。

範例：

```
os> source=accounts | eventstats avg(age) by gender;
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                    | city   | state | avg(age) by gender |
```

```
+----------------+----------+-----------+----------+-----+--------
+----------------------+------------+--------------------------+--------+-------
+--------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 33.67              |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 33.67              |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 28.00              |
| 18             | 4180     | Juan      | Li       | 33  | M      | 467 Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 33.67              |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+--------------------+
```

MAX

MAX(expr) 傳回 expr 的最大值。

範例

```
os> source=accounts | eventstats max(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                    | city   | state | max(age)  |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 36        |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 36        |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 36        |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 36        |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+
```

MIN

MIN(expr) 傳回 expr 的最小值。

範例

```
os> source=accounts | eventstats min(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                     | city   | state | min(age)  |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 28        |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 28        |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | *** Any Street
   | AnyOrg      |                          | Nogal  | VA    | 28        |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 28        |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+
```

STDDEV_SAMP

STDDEV_SAMP(expr) 傳回 expr 的範例標準差。

範例

```
os> source=accounts | eventstats stddev_samp(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+------------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                     | city   | state | stddev_samp(age)       |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+------------------------+
```

```
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 3.304037933599835      |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 3.304037933599835      |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 3.304037933599835      |
| 18             | 4180     | Juan      | Li       | 33  | M      | 467 Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 3.304037933599835      |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+------------------------+
```

STDDEV_POP

STDDEV_POP(expr) 傳回 expr 的母群體標準差。

範例

```
os> source=accounts | eventstats stddev_pop(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+------------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                    | city   | state | stddev_pop(age)        |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+------------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 2.****************     |
| 6              | 5686     | Mary      | Major    | 36  | M      | *** Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 2.****************     |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | *** Any Street
   | AnyOrg      |                          | Nogal  | VA    | 2.****************     |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 2.****************     |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+------------------------+
```

PERCENTILE 或 PERCENTILE_APPROX

PERCENTILE(expr, percent) 或 以指定的百分比PERCENTILE_APPROX(expr, percent)傳回 expr 的近似百分位數值。

百分比

- 數字必須是介於 0 到 100 之間的常數。

範例

```
os> source=accounts | eventstats percentile(age, 90) by gender;
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+--------------------------------+
| account_number | balance  | firstname | lastname | age | gender | address
    | employer    | email                    | city   | state | percentile(age, 90) by
 gender  |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+--------------------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 36
       |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 36
       |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 28
       |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 36
       |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+--------------------------------+
```

範例 1：依群組計算欄位的平均值、總和和計數

此範例顯示依性別計算所有帳戶群組的平均存留期、總存留期和事件計數。

```
os> source=accounts | eventstats avg(age) as avg_age, sum(age) as sum_age, count() as
 count by gender;
```

```
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+-----------+-------+
| account_number | balance  | firstname | lastname | age | gender | address
    | employer    | email                    | city   | state | avg_age   | sum_age   |
 count |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+-----------+-------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
    | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 33.666667 | 101       |
 3     |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
    | Any Company | marymajor@anycompany.com | Dante  | TN    | 33.666667 | 101       |
 3     |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
    | AnyOrg      |                          | Nogal  | VA    | 28.000000 | 28        |
 1     |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
    |             | juanli@exampleorg.com    | Orick  | MD    | 33.666667 | 101       |
 3     |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----------+-----------+-------+
```

範例 2：依跨度計算計數

此範例會取得間隔 10 年的存留期計數。

```
os> source=accounts | eventstats count(age) by span(age, 10) as age_span
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+----------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                    | city   | state | age_span |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+----------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 3        |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 3        |
```

```
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 1        |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 3        |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+----------+
```

範例 3：依性別和跨度計算計數

此範例會依 5 年的間隔取得年齡計數，並依性別分組。

```
os> source=accounts | eventstats count() as cnt by span(age, 5) as age_span, gender
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                    | city   | state | cnt |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----+
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 2   |
| 6              | 5686     | Mary      | Majo     | 36  | M      | 671 Example Street
   | Any Company | hattiebond@anycompany.com | Dante  | TN    | 1   |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | *** Any Street
   | AnyOrg      |                          | Nogal  | VA    | 1   |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 2   |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+-----+
```

Usage

- source = table | eventstats avg(a)
- source = table | where a < 50 | eventstats avg(c)
- source = table | eventstats max(c) by b
- source = table | eventstats count(c) by b | head 5
- source = table | eventstats distinct_count(c)

- source = table | eventstats stddev_samp(c)
- source = table | eventstats stddev_pop(c)
- source = table | eventstats percentile(c, 90)
- source = table | eventstats percentile_approx(c, 99)

包含跨度的彙總

- source = table | eventstats count(a) by span(a, 10) as a_span
- source = table | eventstats sum(age) by span(age, 5) as age_span | head 2
- source = table | eventstats avg(age) by span(age, 20) as age_span, country | sort - age_span | head 2

時間範圍的彙總（翻轉視窗調整函數）

- source = table | eventstats sum(productsAmount) by span(transactionDate, 1d) as age_date | sort age_date
- source = table | eventstats sum(productsAmount) by span(transactionDate, 1w) as age_date, productId

依多個層級彙總群組

- source = table | eventstats avg(age) as avg_state_age by country, state | eventstats avg(avg_state_age) as avg_country_age by country
- source = table | eventstats avg(age) as avg_city_age by country, state, city | eval new_avg_city_age = avg_city_age - 1 | eventstats avg(new_avg_city_age) as avg_state_age by country, state | where avg_state_age > 18 | eventstats avg(avg_state_age) as avg_adult_country_age by country

expand 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 expand命令來扁平化 類型的欄位：

- Array<Any>
- Map<Any>

語法

使用下列語法：

```
expand <field> [As alias]
```

field

- 要擴展 （ 分解 ） 的欄位。必須為支援的類型。

別名

- 選用。要使用的名稱，而非原始欄位名稱。

Usage

expand 命令會為指定陣列或映射欄位中的每個元素產生資料列，其中：

- 陣列元素會成為個別的資料列。
- 映射索引鍵/值對會分成不同的資料列，每個索引鍵/值都以資料列表示。

- 提供別名時，展開的值會以別名表示，而非原始欄位名稱。
- 這可以與其他命令搭配使用，例如 stats、 和 eval，parse以在擴展後操作或擷取資料。

範例

- `source = table | expand employee | stats max(salary) as max by state, company`
- `source = table | expand employee as worker | stats max(salary) as max by state, company`
- `source = table | expand employee as worker | eval bonus = salary * 3 | fields worker, bonus`
- `source = table | expand employee | parse description '(?<email>.+@.+)' | fields employee, email`
- `source = table | eval array=json_array(1, 2, 3) | expand array as uid | fields name, occupation, uid`
- `source = table | expand multi_valueA as multiA | expand multi_valueB as multiB`

explain 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

`explain` 命令可協助您了解查詢執行計劃，讓您能夠分析和最佳化查詢，以獲得更好的效能。此簡介提供 explain 命令的用途及其在查詢最佳化中重要性的簡要概觀。

Comment

- `source=accounts | top gender // finds most common gender of all the accounts`（行註解）
- `source=accounts | dedup 2 gender /* dedup the document with gender field keep 2 duplication */ | fields account_number, gender`（區塊註解）

描述

- `describe table` 此命令等於 `DESCRIBE EXTENDED table` SQL 命令
- `describe schema.table`

- describe schema.`table`
- describe catalog.schema.table
- describe catalog.schema.`table`
- describe `catalog`.`schema`.`table`

說明

- explain simple | source = table | where a = 1 | fields a,b,c
- explain extended | source = table
- explain codegen | source = table | dedup a | fields a,b,c
- explain cost | source = table | sort a | fields a,b,c
- explain formatted | source = table | fields - a
- explain simple | describe table

欄位

- source = table
- source = table | fields a,b,c
- source = table | fields + a,b,c
- source = table | fields - b,c
- source = table | eval b1 = b | fields - b1,c

欄位摘要

- source = t | fieldsummary includefields=status_code nulls=false
- source = t | fieldsummary includefields= id, status_code, request_path nulls=true
- source = t | where status_code != 200 | fieldsummary includefields= status_code nulls=true

巢狀欄位

- source = catalog.schema.table1, catalog.schema.table2 | fields A.nested1, B.nested1
- source = catalog.table | where struct_col2.field1.subfield > 'valueA' | sort int_col | fields int_col, struct_col.field1.subfield, struct_col2.field1.subfield
- source = catalog.schema.table | where struct_col2.field1.subfield > 'valueA' | sort int_col | fields int_col, struct_col.field1.subfield, struct_col2.field1.subfield

篩選條件

- source = table | where a = 1 | fields a,b,c
- source = table | where a >= 1 | fields a,b,c
- source = table | where a < 1 | fields a,b,c
- source = table | where b != 'test' | fields a,b,c
- source = table | where c = 'test' | fields a,b,c | head 3
- source = table | where ispresent(b)
- source = table | where isnull(coalesce(a, b)) | fields a,b,c | head 3
- source = table | where isempty(a)
- source = table | where isblank(a)
- source = table | where case(length(a) > 6, 'True' else 'False') = 'True'
- source = table | where a not in (1, 2, 3) | fields a,b,c
- source = table | where a between 1 and 4 - 注意：這會傳回 >= 1 和 <= 4，即 【1，4】
- source = table | where b not between '2024-09-10' and '2025-09-10' - 注意：這會傳回 b >= '**********' 和 b <= '2025-09-10'
- source = table | where cidrmatch(ip, '***********/24')
- source = table | where cidrmatch(ipv6, '2003:db8::/32')
- source = table | trendline sma(2, temperature) as temp_trend

IP 相關查詢

- source = table | where cidrmatch(ip, '***************')
- source = table | where isV6 = false and isValid = true and cidrmatch(ipAddress, '***************')
- source = table | where isV6 = true | eval inRange = case(cidrmatch(ipAddress, '2003:***::/32'), 'in' else 'out') | fields ip, inRange

複雜篩選條件

```
source = table | eval status_category =
case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
else 'Incorrect HTTP status code')
| where case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
else 'Incorrect HTTP status code'
) = 'Incorrect HTTP status code'
```

```
source = table
| eval factor = case(a > 15, a - 14, isnull(b), a - 7, a < 3, a + 1 else 1)
| where case(factor = 2, 'even', factor = 4, 'even', factor = 6, 'even', factor = 8,
 'even' else 'odd') = 'even'
| stats count() by factor
```

具有邏輯條件的篩選條件

- source = table | where c = 'test' AND a = 1 | fields a,b,c
- source = table | where c != 'test' OR a > 1 | fields a,b,c | head 1
- source = table | where c = 'test' NOT a > 1 | fields a,b,c

評估

假設：a、b、 c是 中的現有欄位 table

- source = table | eval f = 1 | fields a,b,c,f
- source = table | eval f = 1 (輸出 a、b、c、f 欄位)
- source = table | eval n = now() | eval t = unix_timestamp(a) | fields n,t
- source = table | eval f = a | where f > 1 | sort f | fields a,b,c | head 5
- source = table | eval f = a * 2 | eval h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = b | stats avg(f) by h
- source = table | eval f = ispresent(a)
- source = table | eval r = coalesce(a, b, c) | fields r
- source = table | eval e = isempty(a) | fields e
- source = table | eval e = isblank(a) | fields e
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one', a = 2, 'two', a = 3, 'three', a = 4, 'four', a = 5, 'five', a = 6, 'six', a = 7, 'se7en', a = 8, 'eight', a = 9, 'nine')
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else 'unknown')
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else concat(a, ' is an incorrect binary digit'))
- source = table | eval digest = md5(fieldName) | fields digest
- source = table | eval digest = sha1(fieldName) | fields digest
- source = table | eval digest = sha2(fieldName,256) | fields digest
- source = table | eval digest = sha2(fieldName,512) | fields digest

fillnull 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

Description

使用 `fillnull`命令，將 Null 值取代為搜尋結果一或多個欄位中的指定值。

語法

使用下列語法：

```
fillnull [with <null-replacement> in <nullable-field>["," <nullable-field>]] | [using
 <source-field> = <null-replacement> [","<source-field> = <null-replacement>]]
```

- null-replacement：必要。用來取代 null 值的值。
- nullable-field：必要。欄位參考。此欄位中的 null 值將取代為 null 取代中指定的值。

範例 1：填入一個欄位

此範例示範如何在單一欄位上使用 fillnull：

```
os> source=logs | fields status_code | eval input=status_code | fillnull with 0 in
 status_code;
| input | status_code |
|-------|-------------|
| 403   | 403         |
| 403   | 403         |
| NULL  | 0           |
| NULL  | 0           |
| 200   | 200         |
| 404   | 404         |
| 500   | 500         |
| NULL  | 0           |
| 500   | 500         |
| 404   | 404         |
| 200   | 200         |
| 500   | 500         |
| NULL  | 0           |
| NULL  | 0           |
| 404   | 404         |
```

範例 2：套用至多個欄位的 fillnull

此範例顯示套用至多個欄位的 fillnull。

```
os> source=logs | fields request_path, timestamp | eval
 input_request_path=request_path, input_timestamp = timestamp | fillnull with '???' in
 request_path, timestamp;
| input_request_path | input_timestamp       | request_path | timestamp             |
|-------------------------------------------------------------------------------------|
| /contact           | NULL                  | /contact     | ???                   |
| /home              | NULL                  | /home        | ???                   |
| /about             | 2023-10-01 10:30:00   | /about       | 2023-10-01 10:30:00   |
| /home              | 2023-10-01 10:15:00   | /home        | 2023-10-01 10:15:00   |
| NULL               | 2023-10-01 10:20:00   | ???          | 2023-10-01 10:20:00   |
| NULL               | 2023-10-01 11:05:00   | ???          | 2023-10-01 11:05:00   |
| /about             | NULL                  | /about       | ???                   |
| /home              | 2023-10-01 10:00:00   | /home        | 2023-10-01 10:00:00   |
| /contact           | NULL                  | /contact     | ???                   |
| NULL               | 2023-10-01 10:05:00   | ???          | 2023-10-01 10:05:00   |
| NULL               | 2023-10-01 10:50:00   | ???          | 2023-10-01 10:50:00   |
| /services          | NULL                  | /services    | ???                   |
| /home              | 2023-10-01 10:45:00   | /home        | 2023-10-01 10:45:00   |
| /services          | 2023-10-01 11:00:00   | /services    | 2023-10-01 11:00:00   |
| NULL               | 2023-10-01 10:35:00   | ???          | 2023-10-01 10:35:00   |
```

範例 3：將 fillnull 套用至具有各種 null 取代值的多個欄位。

此範例顯示 fillnull，其中包含用於取代 null 的各種值。

- `/error` 在 `request_path` 欄位中
- `1970-01-01 00:00:00` 在 `timestamp` 欄位中

```
os> source=logs | fields request_path, timestamp | eval
 input_request_path=request_path, input_timestamp = timestamp | fillnull using
 request_path = '/error', timestamp='1970-01-01 00:00:00';

| input_request_path | input_timestamp       | request_path | timestamp             |
|-------------------------------------------------------------------------------------|
| /contact           | NULL                  | /contact     | 1970-01-01 00:00:00   |
| /home              | NULL                  | /home        | 1970-01-01 00:00:00   |
| /about             | 2023-10-01 10:30:00   | /about       | 2023-10-01 10:30:00   |
| /home              | 2023-10-01 10:15:00   | /home        | 2023-10-01 10:15:00   |
| NULL               | 2023-10-01 10:20:00   | /error       | 2023-10-01 10:20:00   |
| NULL               | 2023-10-01 11:05:00   | /error       | 2023-10-01 11:05:00   |
| /about             | NULL                  | /about       | 1970-01-01 00:00:00   |
```

```
| /home              | 2023-10-01 10:00:00   | /home        | 2023-10-01 10:00:00    |
| /contact           | NULL                  | /contact     | 1970-01-01 00:00:00    |
| NULL               | 2023-10-01 10:05:00   | /error       | 2023-10-01 10:05:00    |
| NULL               | 2023-10-01 10:50:00   | /error       | 2023-10-01 10:50:00    |
| /services          | NULL                  | /services    | 1970-01-01 00:00:00    |
| /home              | 2023-10-01 10:45:00   | /home        | 2023-10-01 10:45:00    |
| /services          | 2023-10-01 11:00:00   | /services    | 2023-10-01 11:00:00    |
| NULL               | 2023-10-01 10:35:00   | /error       | 2023-10-01 10:35:00    |
```

field 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 `fields`命令來保留或移除搜尋結果中的欄位。

語法

使用下列語法：

```
field [+|-] <field-list>
```

- `index`：選用。

  如果使用加號 (+)，則只會保留欄位清單中指定的欄位。

  如果使用減號 (-)，則會移除欄位清單中指定的所有欄位。

  預設：+
- `field list`：強制性。要保留或移除的欄位逗號分隔清單。

範例 1：從結果中選取指定的欄位

此範例說明如何從搜尋結果中擷取 `firstname`、 `account_number`和 `lastname` 欄位。

PPL 查詢：

```
os> source=accounts | fields account_number, firstname, lastname;
```

```
fetched rows / total rows = 4/4
+------------------+-------------+------------+
| account_number   | firstname   | lastname   |
|------------------+-------------+------------|
| 1                | Jane        | Doe        |
| 6                | John        | Doe        |
| 13               | Jorge       | Souza      |
| 18               | Juan        | Li         |
+------------------+-------------+------------+
```

範例 2：從結果中移除指定的欄位

此範例示範如何從搜尋結果中移除 account_number 欄位。

PPL 查詢：

```
os> source=accounts | fields account_number, firstname, lastname | fields -
 account_number ;
fetched rows / total rows = 4/4
+-------------+------------+
| firstname   | lastname   |
|-------------+------------|
| Jane        | Doe        |
| John        | Doe        |
| Jorge       | Souza      |
| Juan        | Li         |
+-------------+------------+
```

其他範例

- source = table
- source = table | fields a,b,c
- source = table | fields + a,b,c
- source = table | fields - b,c
- source = table | eval b1 = b | fields - b1,c

巢狀欄位範例：

```
`source = catalog.schema.table1, catalog.schema.table2 | fields A.nested1, B.nested1`
```

```
`source = catalog.table | where struct_col2.field1.subfield > 'valueA' | sort int_col |
 fields  int_col, struct_col.field1.subfield, struct_col2.field1.subfield`
`source = catalog.schema.table | where struct_col2.field1.subfield > 'valueA' | sort
 int_col | fields  int_col, struct_col.field1.subfield, struct_col2.field1.subfield`
```

平面命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用平面命令來展開下列類型的欄位：

- `struct<?,?>`
- `array<struct<?,?>>`

語法

使用下列語法：

```
flatten <field>
```

- 欄位：要扁平化的欄位。欄位必須為支援的類型。

結構描述

col_name	data_type
_time	string
橋接器	array<struct<length：bigint，name：string>>
城市	string
coor	struct<alt：bigint，lat：double，long：double>

col_name	data_type
國家/地區	string

資料

_time	橋接器	城市	coor	國家/地區
2024-09-13T12：00：00	【{801，Tower Bridge}，{928，London Bridge}】	倫敦	{35、51.5074、-0.1278}	英國
2024-09-13T12：00：00	【{232，Pont Neuf}，{160，Pont Alexandre III}】	巴黎	{35、48.8566、2.3522}	法國
2024-09-13T12：00：00	【{48，Rialto Bridge}，{11，Bridge of Sighs}】	維尼斯	{2，45.4408，12.3155}	義大利
2024-09-13T12：00：00	【{***、Charles Bridge}、{343、Legion Bridge}】	布拉格	{200、50.0755、14.4378}	捷克
2024-09-13T12：00：00	【{375，Chain Bridge}，	布達佩斯	{96、47.4979、19.0402}	匈牙利

_time	橋接器	城市	coor	國家/地區
	{333，Liberty Bridge}】			
1990-09-13T12：00：00	NULL	華沙	NULL	波蘭

範例 1：平面化結構

此範例示範如何扁平化結構欄位。

PPL 查詢：

```
source=table | flatten coor
```

_time	橋接器	城市	國家/地區	alt	lat	long
2024-09-13T12：00：00	【{801，Tower Bridge}，{928，London Bridge}】	倫敦	英國	35	51.5074	-0.1278
2024-09-13T12：00：00	【{232，Pont Neuf}，{160，Pont Alexandre III}】	巴黎	法國	35	48.8566	2.3522
2024-09-13T12：00：00	【{48，Rialto Bridge}，{11，	維尼斯	義大利	2	45.4408	12.3155

_time	橋接器	城市	國家/地區	alt	lat	long
	Bridge of Sighs}】					
2024-09-13T12：00：00	【{516，Charles Bridge}，{343，Legion Bridge}】	布拉格	捷克	200	50.0755	14.4378
2024-09-13T12：00：00	【{375，Chain Bridge}，{333，Liberty Bridge}】	布達佩斯	匈牙利	96	47.4979	19.0402
1990-09-13T12：00：00	NULL	華沙	波蘭	NULL	NULL	NULL

範例 2：平面陣列

此範例示範如何扁平化結構欄位的陣列。

PPL 查詢：

```
source=table | flatten bridges
```

_time	城市	coor	國家/地區	length	name
2024-09-13T12：00：00	倫敦	{35、51.5074、-0.1278}	英國	801	塔橋

_time	城市	coor	國家/地區	length	name
2024-09-13T12：00：00	倫敦	{35、51.5074、-0.1278}	英國	928	倫敦大橋
2024-09-13T12：00：00	巴黎	{35、48.8566、2.3522}	法國	232	Pont Neuf
2024-09-13T12：00：00	巴黎	{35、48.8566、2.3522}	法國	160	Pont Alexandre III
2024-09-13T12：00：00	維尼斯	{2，45.4408，12.3155}	義大利	48	Rialto 橋接器
2024-09-13T12：00：00	維尼斯	{2，45.4408，12.3155}	義大利	11	嘆號的橋接
2024-09-13T12：00：00	布拉格	{200、50.0755、14.4378}	捷克	516	查爾斯大橋
2024-09-13T12：00：00	布拉格	{200、50.0755、14.4378}	捷克	343	Legion 橋接器
2024-09-13T12：00：00	布達佩斯	{96、47.4979、19.0402}	匈牙利	375	Chain Bridge
2024-09-13T12：00：00	布達佩斯	{96、47.4979、19.0402}	匈牙利	333	自由大橋

_time	城市	coor	國家/地區	length	name
1990-09-13T12：00：00	華沙	NULL	波蘭	NULL	NULL

範例 3：平面化陣列和結構

此範例示範如何扁平化多個欄位。

PPL 查詢：

```
source=table | flatten bridges | flatten coor
```

_time	城市	國家/地區	length	name	alt	lat	long
2024-09-13T12：00：00	倫敦	英國	801	塔橋	35	51.5074	-0.1278
2024-09-13T12：00：00	倫敦	英國	928	倫敦大橋	35	51.5074	-0.1278
2024-09-13T12：00：00	巴黎	法國	232	Pont Neuf	35	48.8566	2.3522
2024-09-13T12：00：00	巴黎	法國	160	Pont Alexandre III	35	48.8566	2.3522
2024-09-13T12：00：00	維尼斯	義大利	48	Rialto 橋接器	2	45.4408	12.3155

_time	城市	國家/地區	length	name	alt	lat	long
2024-09-13T12：00：00	維尼斯	義大利	11	嘆號的橋接	2	45.4408	12.3155
2024-09-13T12：00：00	布拉格	捷克	516	查爾斯大橋	200	50.0755	14.4378
2024-09-13T12：00：00	布拉格	捷克	343	Legion 橋接器	200	50.0755	14.4378
2024-09-13T12：00：00	布達佩斯	匈牙利	375	Chain Bridge	96	47.4979	19.0402
2024-09-13T12：00：00	布達佩斯	匈牙利	333	自由橋	96	47.4979	19.0402
1990-09-13T12：00：00	華沙	波蘭	NULL	NULL	NULL	NULL	NULL

grok 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

`grok` 命令會剖析具有 grok 模式的文字欄位，並將結果附加至搜尋結果。

語法

使用下列語法：

```
grok <field> <pattern>
```

field

- 強制性。
- 欄位必須是文字欄位。

pattern

- 強制性。
- 用來從指定文字欄位擷取新欄位的 grok 模式。
- 如果新的欄位名稱已存在，則會取代原始欄位。

Grok 模式

grok 模式用於比對每個文件的文字欄位，以擷取新欄位。

範例 1：建立新欄位

此範例示範如何host為每個文件建立新欄位。 host將是 email @欄位中 之後的主機名稱。剖析 null 欄位會傳回空字串。

```
os> source=accounts | grok email '.+@%{HOSTNAME:host}' | fields email, host ;
fetched rows / total rows = 4/4
+-------------------------+-------------+
| email                   | host        |
|-------------------------+-------------|
| jane_doe@example.com    | example.com |
| arnav_desai@example.net | example.net |
| null                    |             |
| juan_li@example.org     | example.org |
+-------------------------+-------------+
```

範例 2：覆寫現有欄位

此範例示範如何在移除街道號碼的情況下覆寫現有address欄位。

```
os> source=accounts | grok address '%{NUMBER} %{GREEDYDATA:address}' | fields address ;
```

```
fetched rows / total rows = 4/4
+------------------+
| address          |
|------------------|
| Example Lane     |
| Any Street       |
| Main Street      |
| Example Court    |
+------------------+
```

範例 3：使用 grok 剖析日誌

此範例示範如何使用 grok 剖析原始日誌。

```
os> source=apache | grok message '%{COMMONAPACHELOG}' | fields COMMONAPACHELOG,
 timestamp, response, bytes ;
fetched rows / total rows = 4/4
+-----------------------------------------------------------------------------------------------------------
+----------------------------+------------+---------+
| COMMONAPACHELOG
                                          | timestamp                  | response   |
 bytes   |
|-----------------------------------------------------------------------------------------------------------
+----------------------------+------------+---------|
| 177.95.8.74 - upton5450 [28/Sep/2022:10:15:57 -0700] "HEAD /e-business/mindshare
 HTTP/1.0" 404 19927                        | 28/Sep/2022:10:15:57 -0700 | 404        |
 19927   |
| 127.45.152.6 - pouros8756 [28/Sep/2022:10:15:57 -0700] "GET /architectures/
convergence/niches/mindshare HTTP/1.0" 100 28722 | 28/Sep/2022:10:15:57 -0700 | 100
    | 28722   |
| *************** - - [28/Sep/2022:10:15:57 -0700] "PATCH /strategize/out-of-the-box
 HTTP/1.0" 401 27439                       | 28/Sep/2022:10:15:57 -0700 | 401        |
 27439   |
| ************** - - [28/Sep/2022:10:15:57 -0700] "POST /users HTTP/1.1" 301 9481
                                       | 28/Sep/2022:10:15:57 -0700 | 301        | 9481
    |
+-----------------------------------------------------------------------------------------------------------
+----------------------------+------------+---------+
```

限制

grok 命令具有與剖析命令相同的限制。

head 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 head命令，在搜尋順序的選用位移之後，傳回指定結果的前 N 個數目。

語法

使用下列語法：

```
head [<size>] [from <offset>]
```

<size>

- 選用整數。
- 要傳回的結果數目。
- 預設：10

<offset>

- 選用 後的整數from。
- 要略過的結果數目。
- 預設：0

範例 1：取得前 10 個結果

此範例示範如何從帳戶索引擷取最多 10 個結果。

PPL 查詢：

```
os> source=accounts | fields firstname, age | head;
fetched rows / total rows = 4/4
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| Jane        | 32    |
```

```
| John         | 36    |
| Jorge        | 28    |
| Juan         | 33    |
+--------------+-------+
```

範例 2：取得第一個 N 結果

此範例顯示帳戶索引的第一個 N 個結果。

PPL 查詢：

```
os> source=accounts | fields firstname, age | head 3;
fetched rows / total rows = 3/3
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| Jane        | 32    |
| John        | 36    |
| Jorge       | 28    |
+-------------+-------+
```

範例 3：偏移 M 後取得第一個 N 結果

此範例示範如何在從帳戶索引略過 M 結果之後擷取第一個 N 結果。

PPL 查詢：

```
os> source=accounts | fields firstname, age | head 3 from 1;
fetched rows / total rows = 3/3
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| John        | 36    |
| Jorge       | 28    |
| Juan        | 33    |
+-------------+-------+
```

join 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

聯結命令可讓您根據常見欄位結合來自多個來源的資料，讓您執行複雜的分析，並從分散式資料集取得更深入的洞見

結構描述

至少有兩個索引，`otel-v1-apm-span-*`（大型） 和 `otel-v1-apm-service-map`（小型）。

索引的相關欄位：

`otel-v1-apm-span-*`

- traceId - 追蹤的唯一識別符。來自相同追蹤的所有範圍都會共用相同的 traceId。
- spanId - 追蹤內跨度的唯一識別符，會在建立跨度時指派。
- parentSpanId - 此跨度父範圍的 spanId。如果這是根跨度，則此欄位必須為空白。
- durationInNanos - startTime 和 endTime 之間的差異。（以 UI `latency`為單位）
- serviceName - 跨度來源的資源。
- traceGroup - 追蹤根範圍的名稱。

`otel-v1-apm-service-map`

- serviceName - 發出跨度的服務名稱。
- destination.domain - 此用戶端所呼叫服務的 serviceName。
- destination.resource - 此用戶端呼叫的範圍名稱 (API、操作等）。
- target.domain - 用戶端呼叫之服務的 serviceName。
- target.resource - 用戶端呼叫的範圍名稱 (API、操作等）。
- traceGroupName - 啟動請求鏈的最上層跨度名稱。

需求

join 支援計算下列項目：

對於每個服務，聯結服務映射索引上的跨度索引，以計算不同類型的篩選條件下的指標。

當依`order`服務的追蹤群組篩選時`client_cancel_order`，此範例查詢會計算延遲。

```
SELECT avg(durationInNanos)
FROM `otel-v1-apm-span-000001` t1
```

```
WHERE t1.serviceName = `order`
  AND ((t1.name in
          (SELECT target.resource
           FROM `otel-v1-apm-service-map`
           WHERE serviceName = `order`
             AND traceGroupName = `client_cancel_order`)
        AND t1.parentSpanId != NULL)
       OR (t1.parentSpanId = NULL
           AND t1.name = `client_cancel_order`))
  AND t1.traceId in
    (SELECT traceId
     FROM `otel-v1-apm-span-000001`
     WHERE serviceName = `order`)
```

遷移至 PPL

join 命令的語法

```
SEARCH source=<left-table>
| <other piped command>
| [joinType] JOIN
    [leftAlias]
    ON joinCriteria
    <right-table>
| <other piped command>
```

重寫

```
SEARCH source=otel-v1-apm-span-000001
| WHERE serviceName = 'order'
| JOIN left=t1 right=t2
    ON t1.traceId = t2.traceId AND t2.serviceName = 'order'
    otel-v1-apm-span-000001 -- self inner join
| EVAL s_name = t1.name -- rename to avoid ambiguous
| EVAL s_parentSpanId = t1.parentSpanId -- RENAME command would be better when it is
 supported
| EVAL s_durationInNanos = t1.durationInNanos
| FIELDS s_name, s_parentSpanId, s_durationInNanos -- reduce colunms in join
| LEFT JOIN left=s1 right=t3
    ON s_name = t3.target.resource AND t3.serviceName = 'order' AND t3.traceGroupName =
 'client_cancel_order'
    otel-v1-apm-service-map
```

```
| WHERE (s_parentSpanId IS NOT NULL OR (s_parentSpanId IS NULL AND s_name =
 'client_cancel_order'))
| STATS avg(s_durationInNanos) -- no need to add alias if there is no ambiguous
```

joinType

- 語法: `INNER | LEFT OUTER | CROSS`
- 選用
- 要執行的聯結類型。`INNER` 如果未指定，則預設為 。

leftAlias

- 語法: `left = <leftAlias>`
- 選用
- 要與左側聯結端搭配使用的子查詢別名，以避免不明確命名。

joinCriteria

- 語法: `<expression>`
- 必要
- 語法以 開頭`ON`。它可以是任何比較表達式。一般而言，聯結條件看起來像 `<leftAlias>.<leftField>=<rightAlias>.<rightField>`。

  例如：`l.id = r.id`。如果聯結條件包含多個條件，您可以在每個比較表達式之間指定 `AND`和 `OR`運算子。例如 `l.id = r.id AND l.email = r.email AND (r.age > 65 OR r.age < 18)`。

更多 範例

從 SQL 查詢遷移 (TPC-H Q13) :

```
SELECT c_count, COUNT(*) AS custdist
FROM
  ( SELECT c_custkey, COUNT(o_orderkey) c_count
    FROM customer LEFT OUTER JOIN orders ON c_custkey = o_custkey
        AND o_comment NOT LIKE '%unusual%packages%'
    GROUP BY c_custkey
  ) AS c_orders
```

```
GROUP BY c_count
ORDER BY custdist DESC, c_count DESC;
```

由 PPL 聯結查詢重寫：

```
SEARCH source=customer
| FIELDS c_custkey
| LEFT OUTER JOIN
    ON c_custkey = o_custkey AND o_comment NOT LIKE '%unusual%packages%'
    orders
| STATS count(o_orderkey) AS c_count BY c_custkey
| STATS count() AS custdist BY c_count
| SORT - custdist, - c_count
```

限制：在右側聯結中不支援子搜尋。

如果支援子搜尋，您可以重寫上述 PPL 查詢，如下所示：

```
SEARCH source=customer
| FIELDS c_custkey
| LEFT OUTER JOIN
   ON c_custkey = o_custkey
   [
      SEARCH source=orders
      | WHERE o_comment NOT LIKE '%unusual%packages%'
      | FIELDS o_orderkey, o_custkey
   ]
| STATS count(o_orderkey) AS c_count BY c_custkey
| STATS count() AS custdist BY c_count
| SORT - custdist, - c_count
```

查詢命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 `lookup`命令新增或取代查詢索引（維度表）中的資料，以豐富您的搜尋資料。此命令可讓您使用維度資料表中的值來延伸索引的欄位。您也可以使用它在符合查詢條件時附加或取代值。`lookup` 命令比 `Join`命令更適合使用靜態資料集來充實來源資料。

語法

使用下列語法：

```
SEARCH source=<sourceIndex>
| <other piped command>
| LOOKUP <lookupIndex> (<lookupMappingField> [AS <sourceMappingField>])...
    [(REPLACE | APPEND) (<inputField> [AS <outputField>])...]
| <other piped command>
```

lookupIndex

- 必要.
- 查詢索引的名稱（維度表）。

lookupMappingField

- 必要.
- 查詢索引中的映射索引鍵，類似於右側資料表中的聯結索引鍵。您可以指定多個欄位，以逗號分隔。

sourceMappingField

- 選用。
- 預設：<lookupMappingField>。
- 來源查詢的映射金鑰，類似於左側的聯結金鑰。

inputField

- 選用。
- 預設：查詢索引中找到相符值的所有欄位。
- 查詢索引中的欄位，其中相符的值會套用至結果輸出。您可以指定多個欄位，以逗號分隔。

outputField

- 選用。
- 預設：<inputField>。

- 輸出中的欄位。您可以指定多個輸出欄位。如果您從來源查詢指定現有的欄位名稱，其值將由 inputField 的相符值取代或附加。如果您指定新的欄位名稱，則會將其新增至結果。

取代 | 附加

- 選用。
- 預設：REPLACE
- 指定如何處理相符的值。如果您指定 REPLACE，則 <lookupIndex> 欄位中相符的值會覆寫結果中的值。如果您指定 APPEND，則 <lookupIndex> 欄位中相符的值只會附加到結果中缺少的值。

Usage

- lookUP <lookupIndex> ID AS cid REPLACE 郵件 AS 電子郵件
- lookUP <lookupIndex> 名稱 REPLACE 郵件 AS 電子郵件
- lookUP <lookupIndex> id AS cid、name APPEND 地址、郵件 AS 電子郵件
- lookUP <lookupIndex> ID

範例

請參閱以下範例。

```
SEARCH source=<sourceIndex>
| WHERE orderType = 'Cancelled'
| LOOKUP account_list, mkt_id AS mkt_code REPLACE amount, account_name AS name
| STATS count(mkt_code), avg(amount) BY name
```

```
SEARCH source=<sourceIndex>
| DEDUP market_id
| EVAL category=replace(category, "-", ".")
| EVAL category=ltrim(category, "dvp.")
| LOOKUP bounce_category category AS category APPEND classification
```

```
SEARCH source=<sourceIndex>
| LOOKUP bounce_category category
```

剖析命令

parse 命令會剖析具有規則表達式的文字欄位，並將結果附加到搜尋結果。

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

語法

使用下列語法：

```
parse <field> <pattern>
```

field

- 強制性。
- 欄位必須是文字欄位。

pattern

- 必要字串。
- 這是規則表達式模式，用於從指定文字欄位擷取新欄位。
- 如果新的欄位名稱已存在，則會取代原始欄位。

Regular expression (常規表達式)

規則表達式模式用於比對每個文件的整個文字欄位與 Java regex 引擎。表達式中的每個具名擷取群組都會成為新的STRING欄位。

範例 1：建立新欄位

此範例示範如何host為每個文件建立新欄位。 host將是 email @欄位中 之後的主機名稱。剖析 null 欄位會傳回空字串。

PPL 查詢：

```
os> source=accounts | parse email '.+@(?<host>.+)' | fields email, host ;
```

```
fetched rows / total rows = 4/4
+----------------------+-------------+
| email                | host        |
|----------------------+-------------|
| jane_doe@example.com  | example.com |
| john_doe@example.net  | example.net |
| null                 |             |
| juan_li@example.org   | example.org |
+----------------------+-------------+
```

範例 2：覆寫現有欄位

此範例示範如何在移除街道號碼的情況下覆寫現有address欄位。

PPL 查詢：

```
os> source=accounts | parse address '\d+ (?<address>.+)' | fields address ;
fetched rows / total rows = 4/4
+------------------+
| address          |
|------------------|
| Example Lane     |
| Example Street   |
| Example Avenue   |
| Example Court    |
+------------------+
```

範例 3：依轉換的剖析欄位篩選和排序

此範例顯示如何排序address欄位中大於 500 的街道號碼。

PPL 查詢：

```
os> source=accounts | parse address '(?<streetNumber>\d+) (?<street>.+)' | where
 cast(streetNumber as int) > 500 | sort num(streetNumber) | fields streetNumber,
 street ;
fetched rows / total rows = 3/3
+----------------+----------------+
| streetNumber   | street         |
|----------------+----------------|
| ***            | Example Street |
| ***            | Example Avenue |
| 880            | Example Lane   |
```

```
+----------------+----------------+
```

限制

剖析命令有一些限制：

- 剖析定義的欄位無法再次剖析。

  下列命令無法運作：

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | parse street '\w+ (?<road>\w+)'
  ```

- 剖析定義的欄位無法以其他命令覆寫。

  where 不符合任何文件，因為 street無法覆寫：

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | eval street='1' | where
   street='1' ;
  ```

- 剖析所使用的文字欄位無法覆寫。

  street 無法成功剖析 ，因為 address 被覆寫：

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | eval address='1' ;
  ```

- 在 stats命令中使用剖析定義的欄位後，就無法進行篩選或排序。

  where 下列命令中的 無法運作：

  ```
  source=accounts | parse email '.+@(?<host>.+)' | stats avg(age) by host | where
   host=pyrami.com ;
  ```

模式命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

patterns 命令會從文字欄位擷取日誌模式，並將結果附加至搜尋結果。依模式分組日誌可讓您更輕鬆地從大量日誌資料彙總統計資料，以進行分析和故障診斷。

語法

使用下列語法：

```
patterns [new_field=<new-field-name>] [pattern=<pattern>] <field>
```

new-field-name

- 選用字串。
- 這是擷取模式的新欄位名稱。
- 預設值為 patterns_field。
- 如果名稱已存在，則會取代原始欄位。

pattern

- 選用字串。
- 這是應該從文字欄位篩選出的字元規則運算式模式。
- 如果不存在，預設模式為英數字元 ([a-zA-Z\d])。

field

- 強制性。
- 欄位必須是文字欄位。

範例 1：建立新欄位

此範例示範如何email針對每個文件使用 中的擷取標點符號。剖析 null 欄位會傳回空字串。

PPL 查詢：

```
os> source=accounts | patterns email | fields email, patterns_field ;
fetched rows / total rows = 4/4
+-----------------------+------------------+
| email                 | patterns_field   |
|-----------------------+------------------|
| jane_doe@example.com  | @.               |
| john_doe@example.net  | @.               |
```

```
| null                  |                  |
| juan_li@example.org   | @.               |
+-----------------------+------------------+
```

範例 2：擷取日誌模式

此範例示範如何使用預設模式從原始日誌欄位擷取標點符號。

PPL 查詢：

```
os> source=apache | patterns message | fields message, patterns_field ;
fetched rows / total rows = 4/4
+-------------------------------------------------------------------------------------------------
+----------------------------------+
| message
                                       | patterns_field                 |
|-------------------------------------------------------------------------------------------------
+----------------------------------|
| 177.95.8.74 - upton5450 [28/Sep/2022:10:15:57 -0700] "HEAD /e-business/mindshare
 HTTP/1.0" 404 19927                          | ... -  [//::: -] " /-/ /."      |
| ************ - pouros8756 [28/Sep/2022:10:15:57 -0700] "GET /architectures/
convergence/niches/mindshare HTTP/1.0" 100 28722 | ... -  [//::: -] " //// /."     |
| *************** - - [28/Sep/2022:10:15:57 -0700] "PATCH /strategize/out-of-the-box
 HTTP/1.0" 401 27439                        | ... - - [//::: -] " //--- /."   |
| ************** - - [28/Sep/2022:10:15:57 -0700] "POST /users HTTP/1.1" 301 9481
                                       | ... - - [//::: -] " / /."       |
+-------------------------------------------------------------------------------------------------
+----------------------------------+
```

範例 3：使用自訂 regex 模式擷取日誌模式

此範例示範如何使用使用者定義的模式，從原始日誌欄位擷取標點符號。

PPL 查詢：

```
os> source=apache | patterns new_field='no_numbers' pattern='[0-9]' message | fields
 message, no_numbers ;
fetched rows / total rows = 4/4
+-------------------------------------------------------------------------------------------------
+-----------------------------------------------------------------------------
+
```

```
| message
                                        | no_numbers
                                        |
|-----------------------------------------------------------------------------------------------
+-------------------------------------------------------------------------------|
| 177.95.8.74 - upton5450 [28/Sep/2022:10:15:57 -0700] "HEAD /e-business/mindshare
 HTTP/1.0" 404 19927                          | ... - upton [/Sep/::: -] "HEAD /e-
business/mindshare HTTP/."                          |
| 127.45.152.6 - pouros8756 [28/Sep/2022:10:15:57 -0700] "GET /architectures/
convergence/niches/mindshare HTTP/1.0" 100 28722 | ... - pouros [/Sep/::: -] "GET /
architectures/convergence/niches/mindshare HTTP/."   |
| *************** - - [28/Sep/2022:10:15:57 -0700] "PATCH /strategize/out-of-the-box
 HTTP/1.0" 401 27439                         | ... - - [/Sep/::: -] "PATCH /strategize/
out-of-the-box HTTP/."                          |
| ************** - - [28/Sep/2022:10:15:57 -0700] "POST /users HTTP/1.1" 301 9481
                                        | ... - - [/Sep/::: -] "POST /users HTTP/."
                                        |
+-----------------------------------------------------------------------------------------------
+-------------------------------------------------------------------------------
+
```

限制

模式命令具有與剖析命令相同的限制。

罕見命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 `rare`命令來尋找欄位清單中所有欄位最不常見的值元組。

> **Note**
>
> 群組區分欄位的每個不同值元組最多傳回 10 個結果。

語法

使用下列語法：

```
rare [N] <field-list> [by-clause] rare_approx [N] <field-list> [by-clause]
```

field-list

- 強制性。
- 欄位名稱的逗號分隔清單。

依 -clause

- 選用。
- 要分組結果的一個或多個欄位。

N

- 要傳回的結果數目。
- 預設：10

罕見_大約

- HyperLogLog++ 演算法使用預估基數，計算罕見 (n) 欄位的近似計數。

範例 1：尋找欄位中最不常見的值

此範例會尋找所有帳戶的最小共同性別。

PPL 查詢：

```
os> source=accounts | rare gender;
os> source=accounts | rare_approx 10 gender;
os> source=accounts | rare_approx gender;
fetched rows / total rows = 2/2
+----------+
| gender   |
|----------|
| F        |
| M        |
+----------+
```

範例 2：尋找依性別組織的最低常見值

此範例會依性別尋找所有帳戶群組的最低共同年齡。

PPL 查詢：

```
os> source=accounts | rare 5 age by gender;
os> source=accounts | rare_approx 5 age by gender;
fetched rows / total rows = 4/4
+----------+-------+
| gender   | age   |
|----------+-------|
| F        | 28    |
| M        | 32    |
| M        | 33    |
| M        | 36    |
+----------+-------+
```

重新命名命令

使用 `rename`命令來變更搜尋結果中一或多個欄位的名稱。

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

語法

使用下列語法：

```
rename <source-field> AS <target-field>["," <source-field> AS <target-field>]...
```

來源欄位

- 強制性。
- 這是您要重新命名的欄位名稱。

target-field

- 強制性。

- 這是您要重新命名的名稱。

範例 1：重新命名一個欄位

此範例說明如何重新命名單一欄位。

PPL 查詢：

```
os> source=accounts | rename account_number as an | fields an;
fetched rows / total rows = 4/4
+------+
| an   |
|------|
| 1    |
| 6    |
| 13   |
| 18   |
+------+
```

範例 2：重新命名多個欄位

此範例示範如何重新命名多個欄位。

PPL 查詢：

```
os> source=accounts | rename account_number as an, employer as emp | fields an, emp;
fetched rows / total rows = 4/4
+------+---------+
| an   | emp     |
|------+---------|
| 1    | Pyrami  |
| 6    | Netagy  |
| 13   | Quility |
| 18   | null    |
+------+---------+
```

限制

- 不支援覆寫現有欄位：

```
source=accounts | grok address '%{NUMBER} %{GREEDYDATA:address}' | fields address
```

搜尋命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 search命令從 索引擷取文件。search 命令只能用作 PPL 查詢中的第一個命令。

語法

使用下列語法：

```
search source=[<remote-cluster>:]<index> [boolean-expression]
```

search

- 選用。
- 可省略的搜尋關鍵字。

index

- 強制性。
- 搜尋命令必須指定要查詢的索引。
- 對於跨叢集搜尋<cluster name>:，索引名稱可由 加上字首。

bool-expression

- 選用。
- 任何評估為布林值的表達式。

範例 1：擷取所有資料

此範例顯示從帳戶索引擷取所有文件。

PPL 查詢：

```
os> source=accounts;
```

```
+------------------+-------------+----------------------+-----------+----------
+--------+---------------+---------+-------+-----------------------+------------+
| account_number   | firstname   | address              | balance   | gender   | city
 | employer       | state   | age   | email                 | lastname   |
|------------------+-------------+----------------------+-----------+----------
+--------+---------------+---------+-------+-----------------------+------------|
| 1                | Jorge       | *** Any Lane         | 39225     | M        | Brogan
 | ExampleCorp    | IL      | 32    | jane_doe@example.com  | Souza      |
| 6                | John        | *** Example Street   | 5686      | M        | Dante
 | AnyCorp        | TN      | 36    | john_doe@example.com  | Doe        |
| 13               | Jane        | *** Any Street       | *****     | F        | Nogal
 | ExampleCompany | VA      | 28    | null                  | Doe        |
| 18               | Juan        | *** Example Court    | 4180      | M        | Orick
 | null           | MD      | 33    | juan_li@example.org   | Li         |
+------------------+-------------+----------------------+-----------+----------
+--------+---------------+---------+-------+-----------------------+------------+
```

範例 2：使用 條件擷取資料

此範例顯示使用 從帳戶索引擷取所有文件。

PPL 查詢：

```
os> SEARCH source=accounts account_number=1 or gender="F";
+------------------+-------------+--------------------+-----------+----------+--------
+----------------+---------+-------+-------------------------+------------+
| account_number   | firstname   | address            | balance   | gender   | city   |
 employer       | state   | age   | email                 -  | lastname   |
|------------------+-------------+--------------------+-----------+----------+--------
+----------------+---------+-------+-------------------------+------------|
| 1                | Jorge       | *** Any Lane       | *****     | M        | Brogan |
 ExampleCorp    | IL      | 32    | jorge_souza@example.com | Souza      |
| 13               | Jane        | *** Any Street     | *****     | F        | Nogal  |
 ExampleCompany | VA      | 28    | null                    | Doe        |
+------------------+-------------+--------------------+-----------+----------+--------
+----------------+---------+-------+-------------------------+------------+
```

sort 命令

使用 sort命令，依指定的欄位排序搜尋結果。

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

語法

使用下列語法：

```
sort <[+|-] sort-field>...
```

[+|-]

- 選用。
- 加號 【+】 表示先有 NULL/MISSING 值的遞增順序。
- 減號 【-】 代表 NULL/MISSING 值最後的遞減順序。
- 預設：先以 NULL/MISSING 值遞增順序。

sort-field

- 強制性。
- 用於排序的欄位。

範例 1：依一個欄位排序

此範例顯示如何以遞增順序排序具有存留期欄位的文件。

PPL 查詢：

```
os> source=accounts | sort age | fields account_number, age;
fetched rows / total rows = 4/4
+------------------+-------+
| account_number   | age   |
|------------------+-------|
| 13               | 28    |
| 1                | 32    |
| 18               | 33    |
| 6                | 36    |
+------------------+-------+
```

範例 2：依一個欄位排序並傳回所有結果

此範例顯示如何以遞增順序排序具有存留期欄位的文件。

PPL 查詢：

```
os> source=accounts | sort age | fields account_number, age;
fetched rows / total rows = 4/4
+------------------+-------+
| account_number   | age   |
|------------------+-------|
| 13               | 28    |
| 1                | 32    |
| 18               | 33    |
| 6                | 36    |
+------------------+-------+
```

範例 3：依欄位遞減排序

此範例顯示如何以遞減順序排序具有存留期欄位的文件。

PPL 查詢：

```
os> source=accounts | sort - age | fields account_number, age;
fetched rows / total rows = 4/4
+------------------+-------+
| account_number   | age   |
|------------------+-------|
| 6                | 36    |
| 18               | 33    |
| 1                | 32    |
| 13               | 28    |
+------------------+-------+
```

範例 4：依多個欄位排序

此範例顯示如何以遞增順序排序文件的性別欄位，並以遞減順序排序年齡欄位。

PPL 查詢：

```
os> source=accounts | sort + gender, - age | fields account_number, gender, age;
fetched rows / total rows = 4/4
+------------------+----------+-------+
```

```
| account_number   | gender   | age   |
|------------------+----------+-------|
| 13               | F        | 28    |
| 6                | M        | 36    |
| 18               | M        | 33    |
| 1                | M        | 32    |
+------------------+----------+-------+
```

範例 5：依欄位排序包含 null 值

此範例顯示如何依預設選項排序雇主欄位 （ 先遞增順序和 null)。結果顯示 null 值位於第一列。

PPL 查詢：

```
os> source=accounts | sort employer | fields employer;
fetched rows / total rows = 4/4
+------------+
| employer   |
|------------|
| null       |
| AnyCompany |
| AnyCorp    |
| AnyOrgty   |
+------------+
```

stats 命令

使用 `stats`命令從搜尋結果計算彙總。

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

NULL/MISSING 值處理

NULL/MISSING 值處理

函式	NULL	缺少
COUNT	未計數	未計數

函式	NULL	缺少
SUM	Ignore	Ignore
AVG	Ignore	Ignore
MAX	Ignore	Ignore
MIN	Ignore	Ignore

語法

使用下列語法：

```
stats <aggregation>... [by-clause]
```

聚合

- 強制性。
- 套用至欄位的彙總函數。

依 -clause

- 選用。
- 語法: `by [span-expression,] [field,]...`
- 指定用於分組彙總結果的欄位和表達式。依項目可讓您使用欄位和表達式將彙總結果分組。您可以使用純量函數、彙總函數，甚至是跨度表達式，將特定欄位分割成相等間隔的儲存貯體。
- 預設：如果未指定 `<by-clause>` ，則 stats 命令會傳回代表整個結果集彙總的單一資料列。

span-expression

- 選用，最多一個。
- 語法: `span(field_expr, interval_expr)`
- 間隔表達式的單位預設為自然單位。如果 欄位是日期和時間類型欄位，且間隔是以日期/時間單位為單位，您可以在間隔表達式中指定單位。

- 例如，將age欄位分割為儲存貯體 10 年，看起來像 `span(age, 10)`。若要將時間戳記欄位分割為每小時間隔，請使用 `span(timestamp, 1h)`。

可用的時間單位

跨度間隔單位
毫秒 (ms)
秒（秒）
分鐘 (m、區分大小寫）
小時 (h)
天 (d)
週 (w)
月 (M、區分大小寫）
季度 (q)
年 (y)

彙總函數

COUNT

傳回 SELECT 陳述式所擷取資料列中的 expr 數量計數。

範例：

```
os> source=accounts | stats count();
fetched rows / total rows = 1/1
+-----------+
| count()   |
|-----------|
| 4         |
+-----------+
```

SUM

使用 SUM(expr) 傳回 expr 的總和。

範例

```
os> source=accounts | stats sum(age) by gender;
fetched rows / total rows = 2/2
+------------+----------+
| sum(age)   | gender   |
|------------+----------|
| 28         | F        |
| 101        | M        |
+------------+----------+
```

AVG

使用 AVG(expr)傳回 expr 的平均值。

範例

```
os> source=accounts | stats avg(age) by gender;
fetched rows / total rows = 2/2
+--------------------+----------+
| avg(age)           | gender   |
|--------------------+----------|
| 28.0               | F        |
| 33.666666666666664 | M        |
+--------------------+----------+
```

MAX

使用 MAX(expr)傳回 expr 的最大值。

範例

```
os> source=accounts | stats max(age);
fetched rows / total rows = 1/1
+------------+
| max(age)   |
|------------|
| 36         |
+------------+
```

MIN

使用 MIN(expr)傳回 expr 的最小值。

範例

```
os> source=accounts | stats min(age);
fetched rows / total rows = 1/1
+------------+
| min(age)   |
|------------|
| 28         |
+------------+
```

STDDEV_SAMP

使用 STDDEV_SAMP(expr)傳回 expr.

範例：

```
os> source=accounts | stats stddev_samp(age);
fetched rows / total rows = 1/1
+--------------------+
| stddev_samp(age)   |
|--------------------|
| 3.304037933599835  |
+--------------------+
```

STDDEV_POP

使用 STDDEV_POP(expr) 傳回 expr 的母群體標準差。

範例：

```
os> source=accounts | stats stddev_pop(age);
fetched rows / total rows = 1/1
+--------------------+
| stddev_pop(age)    |
|--------------------|
| 2.**************** |
+--------------------+
```

接受

使用 TAKE(field [, size])傳回欄位的原始值。它不保證值的順序。

field

- 強制性。
- 欄位必須是文字欄位。

size

- 選用整數。
- 應傳回的值數目。
- 預設值為 10。

範例

```
os> source=accounts | stats take(firstname);
fetched rows / total rows = 1/1
+-----------------------------+
| take(firstname)             |
|-----------------------------|
| [Jane, Mary, Nikki, Juan    |
+-----------------------------+
```

PERCENTILE 或 PERCENTILE_APPROX

使用 PERCENTILE(expr, percent)或 PERCENTILE_APPROX(expr, percent) 以指定百分比傳回 expr 的近似百分位數值。

百分比

- 數字必須是介於 0 到 100 之間的常數。

範例

```
os> source=accounts | stats percentile(age, 90) by gender;
fetched rows / total rows = 2/2
```

```
+-----------------------+----------+
| percentile(age, 90)   | gender   |
|-----------------------+----------|
| 28                    | F        |
| 36                    | M        |
+-----------------------+----------+
```

範例 1：計算事件的計數

此範例示範如何計算帳戶中的事件計數。

```
os> source=accounts | stats count();
fetched rows / total rows = 1/1
+-----------+
| count()   |
|-----------|
| 4         |
+-----------+
```

範例 2：計算欄位的平均值

此範例示範如何計算所有帳戶的平均存留期。

```
os> source=accounts | stats avg(age);
fetched rows / total rows = 1/1
+------------+
| avg(age)   |
|------------|
| 32.25      |
+------------+
```

範例 3：依群組計算欄位的平均值

此範例顯示如何計算所有帳戶的平均存留期，依性別分組。

```
os> source=accounts | stats avg(age) by gender;
fetched rows / total rows = 2/2
+--------------------+----------+
| avg(age)           | gender   |
|--------------------+----------|
| 28.0               | F        |
| 33.666666666666664 | M        |
```

```
+--------------------+----------+
```

範例 4：依群組計算欄位的平均值、總和和計數

此範例示範如何計算平均存留期、總存留期，以及依性別分組之所有帳戶的事件計數。

```
os> source=accounts | stats avg(age), sum(age), count() by gender;
fetched rows / total rows = 2/2
+--------------------+------------+-----------+----------+
| avg(age)           | sum(age)   | count()   | gender   |
|--------------------+------------+-----------+----------|
| 28.0               | 28         | 1         | F        |
| 33.666666666666664 | 101        | 3         | M        |
+--------------------+------------+-----------+----------+
```

範例 5：計算欄位的最大值

此範例會計算所有帳戶的最長存留期。

```
os> source=accounts | stats max(age);
fetched rows / total rows = 1/1
+------------+
| max(age)   |
|------------|
| 36         |
+------------+
```

範例 6：依群組計算欄位的最大值和最小值

此範例會計算所有帳戶的最大和最小存留期值，依性別分組。

```
os> source=accounts | stats max(age), min(age) by gender;
fetched rows / total rows = 2/2
+------------+------------+----------+
| max(age)   | min(age)   | gender   |
|------------+------------+----------|
| 28         | 28         | F        |
| 36         | 32         | M        |
+------------+------------+----------+
```

範例 7：計算欄位的不同計數

若要取得欄位不同值的計數，您可以使用 DISTINCT_COUNT (或 DC) 函數而非 COUNT。此範例會計算所有帳戶的性別欄位計數和相異計數。

```
os> source=accounts | stats count(gender), distinct_count(gender);
fetched rows / total rows = 1/1
+-----------------+--------------------------+
| count(gender)   | distinct_count(gender)   |
|-----------------+--------------------------|
| 4               | 2                        |
+-----------------+--------------------------+
```

範例 8：依跨度計算計數

此範例會取得間隔 10 年的存留期計數。

```
os> source=accounts | stats count(age) by span(age, 10) as age_span
fetched rows / total rows = 2/2
+--------------+------------+
| count(age)   | age_span   |
|--------------+------------|
| 1            | 20         |
| 3            | 30         |
+--------------+------------+
```

範例 9：依性別和跨度計算計數

此範例會計算依性別和年齡分組的記錄，為期 5 年。

```
os> source=accounts | stats count() as cnt by span(age, 5) as age_span, gender
fetched rows / total rows = 3/3
+-------+------------+----------+
| cnt   | age_span   | gender   |
|-------+------------+----------|
| 1     | 25         | F        |
| 2     | 30         | M        |
| 1     | 35         | M        |
+-------+------------+----------+
```

無論命令中指定的順序為何，跨度表達式一律會顯示為第一個分組索引鍵。

```
os> source=accounts | stats count() as cnt by gender, span(age, 5) as age_span
```

```
fetched rows / total rows = 3/3
+-------+------------+----------+
| cnt   | age_span   | gender   |
|-------+------------+----------|
| 1     | 25         | F        |
| 2     | 30         | M        |
| 1     | 35         | M        |
+-------+------------+----------+
```

範例 10：計算計數並依性別和跨度取得電子郵件清單

此範例會依 10 年的間隔取得年齡計數，並依性別分組，此外，每列會取得最多 5 封電子郵件的清單。

```
os> source=accounts | stats count() as cnt, take(email, 5) by span(age, 5) as age_span,
 gender
fetched rows / total rows = 3/3
+-------+-----------------------------------------------------+------------+----------+
| cnt   | take(email, 5)                                      | age_span   | gender   |
|-------+-----------------------------------------------------+------------+----------|
| 1     | []                                                  | 25         | F        |
| 2     | [janedoe@anycompany.com,juanli@examplecompany.org] | 30         | M        |
| 1     | [marymajor@examplecorp.com]                         | 35         | M        |
+-------+-----------------------------------------------------+------------+----------+
```

範例 11：計算欄位的百分位數

此範例示範如何計算所有帳戶的第 90 個百分位數年齡。

```
os> source=accounts | stats percentile(age, 90);
fetched rows / total rows = 1/1
+-----------------------+
| percentile(age, 90)   |
|-----------------------|
| 36                    |
+-----------------------+
```

範例 12：依群組計算欄位的百分位數

此範例顯示如何依性別計算所有帳戶群組的第 90 個百分位數年齡。

```
os> source=accounts | stats percentile(age, 90) by gender;
```

```
fetched rows / total rows = 2/2
+-----------------------+----------+
| percentile(age, 90)   | gender   |
|-----------------------+----------|
| 28                    | F        |
| 36                    | M        |
+-----------------------+----------+
```

範例 13：依性別和跨度計算百分位數

此範例會取得以 10 年為間隔的第 90 個百分位數，並依性別分組。

```
os> source=accounts | stats percentile(age, 90) as p90 by span(age, 10) as age_span,
 gender
fetched rows / total rows = 2/2
+-------+------------+----------+
| p90   | age_span   | gender   |
|-------+------------+----------|
| 28    | 20         | F        |
| 36    | 30         | M        |
+-------+------------+----------+
```

```
- `source = table | stats avg(a) `
- `source = table | where a < 50 | stats avg(c) `
- `source = table | stats max(c) by b`
- `source = table | stats count(c) by b | head 5`
- `source = table | stats distinct_count(c)`
- `source = table | stats stddev_samp(c)`
- `source = table | stats stddev_pop(c)`
- `source = table | stats percentile(c, 90)`
- `source = table | stats percentile_approx(c, 99)`
```

包含跨度的彙總

```
- `source = table  | stats count(a) by span(a, 10) as a_span`
- `source = table  | stats sum(age) by span(age, 5) as age_span | head 2`
- `source = table  | stats avg(age) by span(age, 20) as age_span, country  | sort -
 age_span |  head 2`
```

使用時間範圍的彙總（翻轉視窗調整函數）

```
- `source = table | stats sum(productsAmount) by span(transactionDate, 1d) as age_date
 | sort age_date`
- `source = table | stats sum(productsAmount) by span(transactionDate, 1w) as age_date,
 productId`
```

依多個層級彙總群組

```
- `source = table | stats avg(age) as avg_state_age by country, state | stats
 avg(avg_state_age) as avg_country_age by country`
- `source = table | stats avg(age) as avg_city_age by country, state, city | eval
 new_avg_city_age = avg_city_age - 1 | stats avg(new_avg_city_age) as avg_state_age
 by country, state | where avg_state_age > 18 | stats avg(avg_state_age) as
 avg_adult_country_age by country`
```

子查詢命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 subquery命令在 Piped Processing Language (PPL) 陳述式中執行複雜的巢狀查詢。

```
source=logs | where field in [ subquery source=events | where condition | fields
 field ]
```

在此範例中，主要搜尋 (source=logs) 會依子查詢 () 的結果進行篩選source=events。

子查詢命令支援多層巢狀化，以進行複雜的資料分析。

巢狀子查詢範例

```
source=logs | where id in [ subquery source=users | where user in [ subquery
 source=actions | where action="login" | fields user] | fields uid ]
```

InSubquery 用量

- `source = outer | where a in [ source = inner | fields b ]`
- `source = outer | where (a) in [ source = inner | fields b ]`

- `source = outer | where (a,b,c) in [ source = inner | fields d,e,f ]`
- `source = outer | where a not in [ source = inner | fields b ]`
- `source = outer | where (a) not in [ source = inner | fields b ]`
- `source = outer | where (a,b,c) not in [ source = inner | fields d,e,f ]`
- `source = outer a in [ source = inner | fields b ]`（使用子查詢進行搜尋篩選）
- `source = outer a not in [ source = inner | fields b ]`（使用子查詢進行搜尋篩選）
- `source = outer | where a in [ source = inner1 | where b not in [ source = inner2 | fields c ] | fields b ]`（巢狀）
- `source = table1 | inner join left = l right = r on l.a = r.a AND r.a in [ source = inner | fields d ] | fields l.a, r.a, b, c`（作為聯結篩選條件）

使用 IN-Subquery PPL 的 SQL 遷移範例

TPC-H Q4（具有彙總的子查詢中）

```
select
  o_orderpriority,
  count(*) as order_count
from
  orders
where
  o_orderdate >= date '1993-07-01'
  and o_orderdate < date '1993-07-01' + interval '3' month
  and o_orderkey in (
    select
      l_orderkey
    from
      lineitem
    where l_commitdate < l_receiptdate
  )
group by
  o_orderpriority
order by
  o_orderpriority
```

由 PPL InSubquery 查詢重寫：

```
source = orders
```

```
| where o_orderdate >= "1993-07-01" and o_orderdate < "1993-10-01" and o_orderkey IN
  [ source = lineitem
    | where l_commitdate < l_receiptdate
    | fields l_orderkey
  ]
| stats count(1) as order_count by o_orderpriority
| sort o_orderpriority
| fields o_orderpriority, order_count
```

TPC-H Q20（巢狀子查詢）

```
select
  s_name,
  s_address
from
  supplier,
  nation
where
  s_suppkey in (
    select
      ps_suppkey
    from
      partsupp
    where
      ps_partkey in (
        select
          p_partkey
        from
          part
        where
          p_name like 'forest%'
      )
  )
  and s_nationkey = n_nationkey
  and n_name = 'CANADA'
order by
  s_name
```

由 PPL InSubquery 查詢重寫：

```
source = supplier
| where s_suppkey IN [
    source = partsupp
```

```
    | where ps_partkey IN [
        source = part
        | where like(p_name, "forest%")
        | fields p_partkey
      ]
    | fields ps_suppkey
  ]
| inner join left=l right=r on s_nationkey = n_nationkey and n_name = 'CANADA'
  nation
| sort s_name
```

ExistsSubquery 用量

假設：a、 b是資料表外部的欄位c、、 d 是資料表內部的欄位e、、 f是資料表內部的欄位2。

- source = outer | where exists [source = inner | where a = c]
- source = outer | where not exists [source = inner | where a = c]
- source = outer | where exists [source = inner | where a = c and b = d]
- source = outer | where not exists [source = inner | where a = c and b = d]
- source = outer exists [source = inner | where a = c]（使用子查詢進行搜尋篩選）
- source = outer not exists [source = inner | where a = c]（使用子查詢進行搜尋篩選）
- source = table as t1 exists [source = table as t2 | where t1.a = t2.a]（資料表別名在存在的子查詢中很有用）
- source = outer | where exists [source = inner1 | where a = c and exists [source = inner2 | where c = e]]（巢狀）
- source = outer | where exists [source = inner1 | where a = c | where exists [source = inner2 | where c = e]]（巢狀）
- source = outer | where exists [source = inner | where c > 10]（不存在不相關）
- source = outer | where not exists [source = inner | where c > 10]（不存在不相關）
- source = outer | where exists [source = inner] | eval l = "nonEmpty" | fields l（存在特殊不相關）

ScalarSubquery 用量

假設：a、 b是資料表外部的欄位c、、 d是資料表內部的欄位e、、 f是資料表巢狀的欄位

不相關的純量子查詢

在 Select 中：

- `source = outer | eval m = [ source = inner | stats max(c) ] | fields m, a`
- `source = outer | eval m = [ source = inner | stats max(c) ] + b | fields m, a`

在 中：

- `source = outer | where a > [ source = inner | stats min(c) ] | fields a`

在搜尋篩選條件中：

- `source = outer a > [ source = inner | stats min(c) ] | fields a`

相關純量子查詢

在 Select 中：

- `source = outer | eval m = [ source = inner | where outer.b = inner.d | stats max(c) ] | fields m, a`
- `source = outer | eval m = [ source = inner | where b = d | stats max(c) ] | fields m, a`
- `source = outer | eval m = [ source = inner | where outer.b > inner.d | stats max(c) ] | fields m, a`

在 中：

- `source = outer | where a = [ source = inner | where outer.b = inner.d | stats max(c) ]`
- `source = outer | where a = [ source = inner | where b = d | stats max(c) ]`

- source = outer | where [source = inner | where outer.b = inner.d OR inner.d = 1 | stats count()] > 0 | fields a

在搜尋篩選條件中：

- source = outer a = [source = inner | where b = d | stats max(c)]
- source = outer [source = inner | where outer.b = inner.d OR inner.d = 1 | stats count()] > 0 | fields a

巢狀純量子查詢

- source = outer | where a = [source = inner | stats max(c) | sort c] OR b = [source = inner | where c = 1 | stats min(d) | sort d]
- source = outer | where a = [source = inner | where c = [source = nested | stats max(e) by f | sort f] | stats max(d) by c | sort c | head 1]

（關係）子查詢

InSubquery、ExistsSubquery和 ScalarSubquery 都是子查詢表達式。
但RelationSubquery不是子查詢表達式，而是在聯結或來源子句中常用的子查詢計畫。

- source = table1 | join left = l right = r [source = table2 | where d > 10 | head 5]（聯結右側的子查詢）
- source = [source = table1 | join left = l right = r [source = table2 | where d > 10 | head 5] | stats count(a) by b] as outer | head 1

其他內容

InSubquery、ExistsSubquery和 ScalarSubquery是where子句和搜尋篩選條件中常用的子查詢表達式。

其中命令：

```
| where <boolean expression> | ...
```

搜尋篩選條件：

```
search source=* <boolean expression> | ...
```

子查詢表達式可用於布林表達式：

```
| where orders.order_id in [ source=returns | where return_reason="damaged" | field
 order_id ]
```

`orders.order_id in [ source=... ]` 是 <boolean expression>。

一般而言，我們將這種子查詢子句命名為InSubquery表達式。這是 <boolean expression>。

具有不同聯結類型的子查詢

使用 ScalarSubquery 的範例：

```
source=employees
| join source=sales on employees.employee_id = sales.employee_id
| where sales.sale_amount > [ source=targets | where target_met="true" | fields
 target_value ]
```

與 InSubquery、ExistsSubquery 和 ScalarSubquery 不同，RelationSubquery 不是子查詢表達式。而是子查詢計畫。

```
SEARCH source=customer
| FIELDS c_custkey
| LEFT OUTER JOIN left = c, right = o ON c.c_custkey = o.o_custkey
   [
      SEARCH source=orders
      | WHERE o_comment NOT LIKE '%unusual%packages%'
      | FIELDS o_orderkey, o_custkey
   ]
| STATS ...
```

top 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 top命令來尋找欄位清單中所有欄位的最常見值元組。

語法

使用下列語法：

```
top [N] <field-list> [by-clause] top_approx [N] <field-list> [by-clause]
```

N

- 要傳回的結果數目。
- 預設：10

field-list

- 強制性。
- 欄位名稱的逗號分隔清單。

依 -clause

- 選用。
- 要分組結果的一個或多個欄位。

top_approx

- 使用 HyperLogLog++ 演算法預估基數的 (n) 個熱門欄位的近似計數。

範例 1：尋找欄位中最常見的值

此範例會尋找所有帳戶的最常見性別。

PPL 查詢：

```
os> source=accounts | top gender;
os> source=accounts | top_approx gender;
fetched rows / total rows = 2/2
+----------+
| gender   |
|----------|
| M        |
```

```
| F         |
+-----------+
```

範例 2：尋找欄位中最常見的值（限制為 1)

此範例會尋找所有帳戶的單一最常見性別。

PPL 查詢：

```
os> source=accounts | top_approx 1 gender;
fetched rows / total rows = 1/1
+-----------+
| gender    |
|-----------|
| M         |
+-----------+
```

範例 3：尋找最常見的值，依性別分組

此範例會尋找所有帳戶的最常見存留期，依性別分組。

PPL 查詢：

```
os> source=accounts | top 1 age by gender;
os> source=accounts | top_approx 1 age by gender;
fetched rows / total rows = 2/2
+-----------+-------+
| gender    | age   |
|-----------+-------|
| F         | 28    |
| M         | 32    |
+-----------+-------+
```

趨勢線命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 `trendline`命令來計算欄位的移動平均值。

語法

使用下列語法

```
TRENDLINE [sort <[+|-] sort-field>] SMA(number-of-datapoints, field) [AS alias]
 [SMA(number-of-datapoints, field) [AS alias]]...
```

[+|-]

- 選用。
- 加號 【+】 表示先有 NULL/MISSING 值的遞增順序。
- 減號 【-】 代表 NULL/MISSING 值最後的遞減順序。
- 預設：先以 NULL/MISSING 值遞增順序。

sort-field

- 使用排序時為強制性。
- 用於排序的欄位。

number-of-datapoints

- 強制性。
- 計算移動平均值的資料點數目。
- 必須大於零。

field

- 強制性。
- 應計算移動平均值的欄位名稱。

別名

- 選用。
- 包含移動平均值的產生資料欄名稱。

僅支援簡易移動平均 (SMA) 類型。其計算方式如下：

```
f[i]: The value of field 'f' in the i-th data-point
n: The number of data-points in the moving window (period)
t: The current time index

SMA(t) = (1/n) * Σ(f[i]), where i = t-n+1 to t
```

範例 1：計算一段時間溫度的簡單移動平均值

此範例使用兩個資料點計算溫度的簡單移動平均值。

PPL 查詢：

```
os> source=t | trendline sma(2, temperature) as temp_trend;
fetched rows / total rows = 5/5
+-----------+---------+--------------------+----------+
|temperature|device-id|           timestamp|temp_trend|
+-----------+---------+--------------------+----------+
|         12|     1492|2023-04-06 17:07:...|      NULL|
|         12|     1492|2023-04-06 17:07:...|      12.0|
|         13|      256|2023-04-06 17:07:...|      12.5|
|         14|      257|2023-04-06 17:07:...|      13.5|
|         15|      258|2023-04-06 17:07:...|      14.5|
+-----------+---------+--------------------+----------+
```

範例 2：使用排序計算溫度時間的簡單移動平均值

此範例使用依 device-id 遞減排序的兩個和三個資料點，計算溫度的兩個簡單移動平均值。

PPL 查詢：

```
os> source=t | trendline sort - device-id sma(2, temperature) as temp_trend_2 sma(3,
 temperature) as temp_trend_3;
fetched rows / total rows = 5/5
+-----------+---------+--------------------+------------+------------------+
|temperature|device-id|           timestamp|temp_trend_2|      temp_trend_3|
+-----------+---------+--------------------+------------+------------------+
|         15|      258|2023-04-06 17:07:...|        NULL|              NULL|
|         14|      257|2023-04-06 17:07:...|        14.5|              NULL|
|         13|      256|2023-04-06 17:07:...|        13.5|              14.0|
|         12|     1492|2023-04-06 17:07:...|        12.5|              13.0|
|         12|     1492|2023-04-06 17:07:...|        12.0|12.333333333333334|
+-----------+---------+--------------------+------------+------------------+
```

其中 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

`where` 命令使用 bool-expression 來篩選搜尋結果。它只會在 bool-expression 評估為 true 時傳回結果。

語法

使用下列語法：

```
where <boolean-expression>
```

bool-expression

- 選用。
- 任何可以評估為布林值的表達式。

範例 1：使用 條件篩選結果集

此範例示範如何從符合特定條件的帳戶索引擷取文件。

PPL 查詢：

```
os> source=accounts | where account_number=1 or gender="F" | fields account_number,
 gender;
fetched rows / total rows = 2/2
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 13               | F        |
+------------------+----------+
```

其他範例

具有邏輯條件的篩選條件

- source = table | where c = 'test' AND a = 1 | fields a,b,c
- source = table | where c != 'test' OR a > 1 | fields a,b,c | head 1
- source = table | where c = 'test' NOT a > 1 | fields a,b,c
- source = table | where a = 1 | fields a,b,c
- source = table | where a >= 1 | fields a,b,c
- source = table | where a < 1 | fields a,b,c
- source = table | where b != 'test' | fields a,b,c
- source = table | where c = 'test' | fields a,b,c | head 3
- source = table | where ispresent(b)
- source = table | where isnull(coalesce(a, b)) | fields a,b,c | head 3
- source = table | where isempty(a)
- source = table | where isblank(a)
- source = table | where case(length(a) > 6, 'True' else 'False') = 'True'
- source = table | where a between 1 and 4 - 注意：這會傳回 >= 1 和 <= 4，即 【1，4】
- source = table | where b not between '2024-09-10' and '2025-09-10' - 注意：這會傳回 b >= '**********' 和 b <= '2025-09-10'
- source = table | where cidrmatch(ip, '***********/24')
- source = table | where cidrmatch(ipv6, '2003:db8::/32')

```
source = table | eval status_category =
    case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
    else 'Incorrect HTTP status code')
    | where case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
    else 'Incorrect HTTP status code'
    ) = 'Incorrect HTTP status code'
```

```
source = table
    | eval factor = case(a > 15, a - 14, isnull(b), a - 7, a < 3, a + 1 else 1)
    | where case(factor = 2, 'even', factor = 4, 'even', factor = 6, 'even', factor =
 8, 'even' else 'odd') = 'even'
    |  stats count() by factor
```

欄位摘要

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 命令，請參閱 the section called “命令”。

使用 `fieldsummary`命令來計算每個欄位的基本統計資料（計數、相異計數、最小值、最大值、平均值、平均值），並判斷每個欄位的資料類型。此命令可與上述任何管道搭配使用，並將它們納入考量。

語法

請使用下列語法。對於 CloudWatch Logs 使用案例，查詢中僅支援一個欄位。

```
... | fieldsummary <field-list> (nulls=true/false)
```

包含欄位

- 將統計資料收集到統一結果集的所有資料欄清單。

Null

- 選用。
- 如果設為 true，請在彙總計算中包含 null 值（數值以 null 取代為零）。

範例 1

PPL 查詢：

```
os> source = t | where status_code != 200 | fieldsummary includefields= status_code
 nulls=true
+------------------+-------------+------------+------------+------------+------------
+------------+------------+----------------|
```

```
| Fields           | COUNT       | COUNT_DISTINCT    |  MIN  |  MAX   |  AVG   |  MEAN
  |        STDDEV       | NUlls | TYPEOF |
|------------------+-------------+-------------------+-------+--------+--------+-------
+------------+-----------+----------------|
| "status_code"    |      2      |         2         | 301   |   403  |  352.0 |  352.0
  |  72.12489168102785  |  0    | "int"  |
+------------------+-------------+-------------------+-------+--------+--------+-------
+------------+-----------+----------------|
```

範例 2

PPL 查詢：

```
os> source = t | fieldsummary includefields= id, status_code, request_path nulls=true
+------------------+-------------+-------------------+-------+--------+--------+-------
+------------+-----------+----------------|
| Fields           | COUNT       | COUNT_DISTINCT    |  MIN  |  MAX   |  AVG   |  MEAN
  |        STDDEV       | NUlls | TYPEOF |
|------------------+-------------+-------------------+-------+--------+--------+-------
+------------+-----------+----------------|
|       "id"       |      6      |         6         | 1     |   6    |  3.5   |   3.5
 |  1.8708286933869707  |  0    | "int"  |
+------------------+-------------+-------------------+-------+--------+--------+-------
+------------+-----------+----------------|
| "status_code"    |      4      |         3         | 200   |   403  |  184.0 |  184.0
  |  161.16699413961905 |  2    | "int"  |
+------------------+-------------+-------------------+-------+--------+--------+-------
+------------+-----------+----------------|
| "request_path"   |      2      |         2         | /about| /home  |  0.0   |  0.0
    |      0            |  2    |"string"|
+------------------+-------------+-------------------+-------+--------+--------+-------
+------------+-----------+----------------|
```

expand 命令

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

使用 expand命令來扁平化 Array<Any> 或 Map<Any> 類型的欄位，為每個元素或鍵值對產生個別資料列。

語法

使用下列語法：

```
expand <field> [As alias]
```

field

- 要擴展（分解）的欄位。
- 欄位必須為支援的類型。

別名

- 選用。
- 要使用的名稱，而非原始欄位名稱。

使用指引

expand 命令會為指定陣列或映射欄位中的每個元素產生資料列，其中：

- 陣列元素會成為個別的資料列。
- 映射索引鍵/值對會分成不同的資料列，每個索引鍵/值都以資料列表示。
- 提供別名時，展開的值會以別名表示，而非原始欄位名稱。

您可以使用此命令搭配其他命令，例如 stats、eval 和 parse，來操作或擷取擴展後的資料。

範例

- `source = table | expand employee | stats max(salary) as max by state, company`
- `source = table | expand employee as worker | stats max(salary) as max by state, company`
- `source = table | expand employee as worker | eval bonus = salary * 3 | fields worker, bonus`
- `source = table | expand employee | parse description '(?<email>.+@.+)' | fields employee, email`

- `source = table | eval array=json_array(1, 2, 3) | expand array as uid | fields name, occupation, uid`
- `source = table | expand multi_valueA as multiA | expand multi_valueB as multiB`

您可以使用 expand 命令搭配其他命令，例如 eval、stats 等。使用多個展開命令會為每個複合陣列或映射內的所有內部元素建立笛卡兒產品。

有效的 SQL 下推查詢

expand 命令會使用 LATERAL VIEW explode 轉換為同等的 SQL 操作，以便在 SQL 查詢層級有效率地分解陣列或映射。

```
SELECT customer exploded_productId
FROM table
LATERAL VIEW explode(productId) AS exploded_productId
```

explode 命令提供下列功能：

- 這是傳回新資料欄的資料欄操作。
- 它會為分解欄中的每個元素建立新的資料列。
- 內部 null 會忽略為分解欄位的一部分 (Null 不會建立/分解任何資料列）。

PPL 函數

主題

- PPL 條件函數
- PPL 密碼編譯雜湊函數
- PPL 日期和時間函數
- PPL 表達式
- PPL IP 地址函數
- PPL JSON 函數
- PPL Lambda 函數
- PPL 數學函數

- PPL 字串函數
- PPL 類型轉換函數

PPL 條件函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

ISNULL

描述：如果 欄位為 null，則`isnull(field)`傳回 true。

引數類型：

- 所有支援的資料類型。

傳回類型：

- BOOLEAN

範例：

```
os> source=accounts | eval result = isnull(employer) | fields result, employer,
 firstname
fetched rows / total rows = 4/4
+----------+-------------+-------------+
| result   | employer    | firstname   |
|----------+-------------+-------------|
| False    | AnyCompany  | Mary        |
| False    | ExampleCorp | Jane        |
| False    | ExampleOrg  | Nikki       |
| True     | null        | Juan        |
+----------+-------------+-------------+
```

ISNOTNULL

描述：如果欄位不是 null，則`isnotnull(field)`傳回 true。

引數類型：

- 所有支援的資料類型。

傳回類型：

- BOOLEAN

範例：

```
os> source=accounts | where not isnotnull(employer) | fields account_number, employer
fetched rows / total rows = 1/1
+------------------+------------+
| account_number   | employer   |
|------------------+------------|
| 18               | null       |
+------------------+------------+
```

EXISTS

範例：

```
os> source=accounts | where exists(email) | fields account_number, email
fetched rows / total rows = 1/1
```

IFNULL

描述：`field2`如果 `field1`為 null，則`ifnull(field1, field2)`傳回 。

引數類型：

- 所有支援的資料類型。
- 如果兩個參數有不同的類型，則函數會失敗語意檢查。

傳回類型：

- 任何

範例：

```
os> source=accounts | eval result = ifnull(employer, 'default') | fields result,
 employer, firstname
fetched rows / total rows = 4/4
+------------+------------+-------------+
| result     | employer   | firstname   |
|------------+------------+-------------|
| AnyCompany | AnyCompany | Mary        |
| ExampleCorp| ExampleCorp| Jane        |
| ExampleOrg | ExampleOrg | Nikki       |
| default    | null       | Juan        |
+------------+------------+-------------+
```

NULLIF

描述：如果兩個參數相同，nullif(field1, field2)則傳回 null，否則傳回 field1。

引數類型：

- 所有支援的資料類型。
- 如果兩個參數有不同的類型，則函數會失敗語意檢查。

傳回類型：

- 任何

範例：

```
os> source=accounts | eval result = nullif(employer, 'AnyCompany') | fields result,
 employer, firstname
fetched rows / total rows = 4/4
+----------------+----------------+-------------+
| result         | employer       | firstname   |
|----------------+----------------+-------------|
| null           | AnyCompany     | Mary        |
| ExampleCorp    | ExampleCorp    | Jane        |
| ExampleOrg     | ExampleOrg     | Nikki       |
| null           | null           | Juan        |
+----------------+----------------+-------------+
```

IF

描述：expr1如果條件為 true，則if(condition, expr1, expr2)傳回 ，否則傳回 expr2。

引數類型：

- 所有支援的資料類型。
- 如果兩個參數有不同的類型，則函數會失敗語意檢查。

傳回類型：

- 任何

範例：

```
os> source=accounts | eval result = if(true, firstname, lastname) | fields result,
 firstname, lastname
fetched rows / total rows = 4/4
+----------+-------------+----------+
| result   | firstname | lastname   |
|----------+-------------+----------|
| Jane     | Jane      | Doe        |
| Mary     | Mary      | Major      |
| Pat      | Pat       | Candella   |
| Dale     | Jorge     | Souza      |
+----------+-----------+------------+

os> source=accounts | eval result = if(false, firstname, lastname) | fields result,
 firstname, lastname
fetched rows / total rows = 4/4
+----------+-------------+------------+
| result   | firstname   | lastname   |
|----------+-------------+------------|
| Doe      | Jane        | Doe        |
| Major    | Mary        | Major      |
| Candella | Pat         | Candella   |
| Souza    | Jorge       | Souza      |
+----------+-------------+------------+

os> source=accounts | eval is_vip = if(age > 30 AND isnotnull(employer), true, false) |
 fields is_vip, firstname, lastname
fetched rows / total rows = 4/4
```

```
+----------+-------------+------------+
| is_vip   | firstname   | lastname   |
|----------+-------------+------------|
| True     | Jane        | Doe        |
| True     | Mary        | Major      |
| False    | Pat         | Candella   |
| False    | Jorge       | Souza      |
+----------+-------------+------------+
```

PPL 密碼編譯雜湊函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

MD5

MD5 會計算 MD5 摘要，並以 32 個字元的十六進位字串傳回值。

用量： `md5('hello')`

引數類型：

- STRING

傳回類型：

- STRING

範例：

```
os> source=people | eval `MD5('hello')` = MD5('hello') | fields `MD5('hello')`
fetched rows / total rows = 1/1
+----------------------------------+
| MD5('hello')                     |
|----------------------------------|
| <32 character hex string>        |
+----------------------------------+
```

SHA1

SHA1 傳回 SHA-1 的十六進位字串結果。

用量： sha1('hello')

引數類型：

- STRING

傳回類型：

- STRING

範例：

```
os> source=people | eval `SHA1('hello')` = SHA1('hello') | fields `SHA1('hello')`
fetched rows / total rows = 1/1
+------------------------------------------+
| SHA1('hello')                            |
|------------------------------------------|
| <40-character SHA-1 hash result>         |
+------------------------------------------+
```

SHA2

SHA2 傳回 SHA-2 系列雜湊函數 (SHA-224、SHA-256, SHA-384和 SHA-512) 的十六進位字串結果。numBits 表示結果的所需位元長度，其值必須為 224、256、384、512

用法：

- sha2('hello',256)
- sha2('hello',512)

引數類型：

- STRING、INTEGER

傳回類型：

- STRING

範例：

```
os> source=people | eval `SHA2('hello',256)` = SHA2('hello',256) | fields
 `SHA2('hello',256)`
fetched rows / total rows = 1/1
+------------------------------------------------------------------+
| SHA2('hello',256)                                                |
|------------------------------------------------------------------|
| <64-character SHA-256 hash result>                               |
+------------------------------------------------------------------+

os> source=people | eval `SHA2('hello',512)` = SHA2('hello',512) | fields
 `SHA2('hello',512)`
fetched rows / total rows = 1/1
+------------------------------------------------------------------+
| SHA2('hello',512)                                                |
                                          |
|------------------------------------------------------------------|
| <128-character SHA-512 hash result>                              |
+------------------------------------------------------------------+
```

PPL 日期和時間函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

DAY

用量： DAY(date)擷取日期的月份日期，範圍介於 1 到 31。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞：DAYOFMONTH、 DAY_OF_MONTH

範例：

```
os> source=people | eval `DAY(DATE('2020-08-26'))` = DAY(DATE('2020-08-26')) | fields
 `DAY(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------+
| DAY(DATE('2020-08-26'))   |
|---------------------------|
| 26                        |
+---------------------------+
```

DAYOFMONTH

用量： DAYOFMONTH(date)擷取日期的月份日期，範圍介於 1 到 31。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞：DAY、 DAY_OF_MONTH

範例：

```
os> source=people | eval `DAYOFMONTH(DATE('2020-08-26'))` =
 DAYOFMONTH(DATE('2020-08-26')) | fields `DAYOFMONTH(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------------+
| DAYOFMONTH(DATE('2020-08-26'))   |
|---------------------------------|
| 26                              |
+---------------------------------+
```

DAY_OF_MONTH

用量： DAY_OF_MONTH(DATE)擷取日期的月份日期，範圍介於 1 到 31。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞：DAY、 DAYOFMONTH

範例：

```
os> source=people | eval `DAY_OF_MONTH(DATE('2020-08-26'))` =
 DAY_OF_MONTH(DATE('2020-08-26')) | fields `DAY_OF_MONTH(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| DAY_OF_MONTH(DATE('2020-08-26'))   |
|-----------------------------------|
| 26                                |
+-----------------------------------+
```

DAYOFWEEK

用量：DAYOFWEEK(DATE)傳回日期的工作日索引 (1 = 星期日，2 = 星期一，...，7 = 星期六）。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞： DAY_OF_WEEK

範例：

```
os> source=people | eval `DAYOFWEEK(DATE('2020-08-26'))` =
 DAYOFWEEK(DATE('2020-08-26')) | fields `DAYOFWEEK(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------------+
| DAYOFWEEK(DATE('2020-08-26'))   |
|---------------------------------|
| 4                               |
+---------------------------------+
```

DAY_OF_WEEK

用量：DAY_OF_WEEK(DATE)傳回日期的工作日索引 (1 = 星期日，2 = 星期一，...，7 = 星期六）。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞： DAYOFWEEK

範例：

```
os> source=people | eval `DAY_OF_WEEK(DATE('2020-08-26'))` =
 DAY_OF_WEEK(DATE('2020-08-26')) | fields `DAY_OF_WEEK(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| DAY_OF_WEEK(DATE('2020-08-26'))   |
|-----------------------------------|
| 4                                 |
+-----------------------------------+
```

DAYOFYEAR

用量：DAYOFYEAR(DATE)傳回日期的年份日期，範圍介於 1 到 366。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞： DAY_OF_YEAR

範例：

```
os> source=people | eval `DAYOFYEAR(DATE('2020-08-26'))` =
 DAYOFYEAR(DATE('2020-08-26')) | fields `DAYOFYEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------------+
| DAYOFYEAR(DATE('2020-08-26'))   |
|---------------------------------|
| 239                             |
+---------------------------------+
```

DAY_OF_YEAR

用量：DAY_OF_YEAR(DATE)傳回日期的年份日期，範圍介於 1 到 366。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞： DAYOFYEAR

範例：

```
os> source=people | eval `DAY_OF_YEAR(DATE('2020-08-26'))` =
 DAY_OF_YEAR(DATE('2020-08-26')) | fields `DAY_OF_YEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| DAY_OF_YEAR(DATE('2020-08-26'))   |
|-----------------------------------|
| 239                               |
+-----------------------------------+
```

DAYNAME

用量： DAYNAME(DATE) 傳回日期的工作日名稱，包括星期一、星期二、星期三、星期四、星期五、星期六和星期日。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：STRING

範例：

```
os> source=people | eval `DAYNAME(DATE('2020-08-26'))` = DAYNAME(DATE('2020-08-26')) |
 fields `DAYNAME(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-------------------------------+
| DAYNAME(DATE('2020-08-26'))   |
|-------------------------------|
| Wednesday                     |
+-------------------------------+
```

FROM_UNIXTIME

用量：FROM_UNIXTIME傳回指定為時間戳記或字元字串值之引數的表示法。此函數會執行UNIX_TIMESTAMP函數的反向轉換。

如果您提供第二個引數， FROM_UNIXTIME 會使用它來格式化類似於 DATE_FORMAT函數的結果。

如果時間戳記不在 1970-01-01 00：00：00 到 3001-01-18 23：59：59.999999 (0 到 32536771199.999999 epoch 時間） 的範圍內，函數會傳回 NULL。

引數類型：DOUBLE、STRING

傳回類型映射：

DOUBLE -> TIMESTAMP

DOUBLE、STRING -> STRING

範例：

```
os> source=people | eval `FROM_UNIXTIME(1220249547)` = FROM_UNIXTIME(1220249547) |
 fields `FROM_UNIXTIME(1220249547)`
fetched rows / total rows = 1/1
+-----------------------------+
| FROM_UNIXTIME(1220249547)   |
|-----------------------------|
| 2008-09-01 06:12:27         |
+-----------------------------+

os> source=people | eval `FROM_UNIXTIME(1220249547, 'HH:mm:ss')` =
 FROM_UNIXTIME(1220249547, 'HH:mm:ss') | fields `FROM_UNIXTIME(1220249547, 'HH:mm:ss')`
fetched rows / total rows = 1/1
+---------------------------------------+
| FROM_UNIXTIME(1220249547, 'HH:mm:ss')   |
|---------------------------------------|
| 06:12:27                              |
+---------------------------------------+
```

HOUR

用量： HOUR(TIME) 擷取時間的小時值。

與一天中的標準時間不同，此函數中的時間值的範圍可以大於 23。因此， 的傳回值HOUR(TIME)可以大於 23。

引數類型：STRING/TIME/TIMESTAMP

傳回類型：INTEGER

同義詞： HOUR_OF_DAY

範例：

```
os> source=people | eval `HOUR(TIME('01:02:03'))` = HOUR(TIME('01:02:03')) | fields
 `HOUR(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+-------------------------+
```

```
| HOUR(TIME('01:02:03'))   |
|--------------------------|
| 1                        |
+--------------------------+
```

HOUR_OF_DAY

用量：從指定時間HOUR_OF_DAY(TIME)擷取小時值。

與一天中的標準時間不同，此函數中的時間值的範圍可以大於 23。因此， 的傳回值HOUR_OF_DAY(TIME)可以大於 23。

引數類型：STRING/TIME/TIMESTAMP

傳回類型：INTEGER

同義詞： HOUR

範例：

```
os> source=people | eval `HOUR_OF_DAY(TIME('01:02:03'))` =
 HOUR_OF_DAY(TIME('01:02:03')) | fields `HOUR_OF_DAY(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+---------------------------------+
| HOUR_OF_DAY(TIME('01:02:03'))   |
|---------------------------------|
| 1                               |
+---------------------------------+
```

LAST_DAY

用量： LAST_DAY 傳回當月的最後一天，做為指定日期引數的 DATE 值。

引數類型：DATE/STRING/TIMESTAMP/TIME

傳回類型：DATE

範例：

```
os> source=people | eval `last_day('2023-02-06')` = last_day('2023-02-06') | fields
 `last_day('2023-02-06')`
fetched rows / total rows = 1/1
+-------------------------+
```

```
| last_day('2023-02-06')   |
|--------------------------|
| 2023-02-28               |
+--------------------------+
```

LOCALTIMESTAMP

用量： LOCALTIMESTAMP() 是 的同義詞NOW()。

範例：

```
> source=people | eval `LOCALTIMESTAMP()` = LOCALTIMESTAMP() | fields
 `LOCALTIMESTAMP()`
fetched rows / total rows = 1/1
+---------------------+
| LOCALTIMESTAMP()    |
|---------------------|
| 2022-08-02 15:54:19 |
+---------------------+
```

LOCALTIME

用量： LOCALTIME() 是 的同義詞NOW()。

範例：

```
> source=people | eval `LOCALTIME()` = LOCALTIME() | fields `LOCALTIME()`
fetched rows / total rows = 1/1
+---------------------+
| LOCALTIME()         |
|---------------------|
| 2022-08-02 15:54:19 |
+---------------------+
```

MAKE_DATE

用量： 會根據指定的年、月和日值MAKE_DATE傳回日期值。所有引數都會四捨五入為整數。

規格：1. MAKE_DATE (整數、整數、整數) -> 日期

引數類型：INTEGER、INTEGER、INTEGER

傳回類型：DATE

範例：

```
os> source=people | eval `MAKE_DATE(1945, 5, 9)` = MAKEDATE(1945, 5, 9) | fields
 `MAKEDATE(1945, 5, 9)`
fetched rows / total rows = 1/1
+------------------------+
| MAKEDATE(1945, 5, 9)   |
|------------------------|
| 1945-05-09             |
+------------------------+
```

MINUTE

用量： MINUTE(TIME) 傳回指定時間的分鐘元件，以 0 到 59 範圍內的整數表示。

引數類型：STRING/TIME/TIMESTAMP

傳回類型：INTEGER

同義詞： MINUTE_OF_HOUR

範例：

```
os> source=people | eval `MINUTE(TIME('01:02:03'))` =  MINUTE(TIME('01:02:03')) |
 fields `MINUTE(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+----------------------------+
| MINUTE(TIME('01:02:03'))   |
|----------------------------|
| 2                          |
+----------------------------+
```

MINUTE_OF_HOUR

用量： MINUTE_OF_HOUR(TIME) 傳回指定時間的分鐘元件，以 0 到 59 範圍內的整數表示。

引數類型：STRING/TIME/TIMESTAMP

傳回類型：INTEGER

同義詞： MINUTE

範例：

```
os> source=people | eval `MINUTE_OF_HOUR(TIME('01:02:03'))` =
 MINUTE_OF_HOUR(TIME('01:02:03')) | fields `MINUTE_OF_HOUR(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| MINUTE_OF_HOUR(TIME('01:02:03'))   |
|-----------------------------------|
| 2                                 |
+-----------------------------------+
```

MONTH

用量：以整數MONTH(DATE)傳回指定日期的月份，範圍介於 1 到 12 （其中 1 代表 1 月，12 代表 12 月）。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞： MONTH_OF_YEAR

範例：

```
os> source=people | eval `MONTH(DATE('2020-08-26'))` =  MONTH(DATE('2020-08-26')) |
 fields `MONTH(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+----------------------------+
| MONTH(DATE('2020-08-26'))   |
|----------------------------|
| 8                          |
+----------------------------+
```

MONTHNAME

用量：以整數MONTHNAME(DATE)傳回指定日期的月份，範圍介於 1 到 12 （其中 1 代表 1 月，12 代表 12 月）。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞： MONTH_OF_YEAR

範例：

```
os> source=people | eval `MONTHNAME(DATE('2020-08-26'))` =
 MONTHNAME(DATE('2020-08-26')) | fields `MONTHNAME(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+--------------------------------+
| MONTHNAME(DATE('2020-08-26'))   |
|--------------------------------|
| August                         |
+--------------------------------+
```

MONTH_OF_YEAR

用量：以整數MONTH_OF_YEAR(DATE)傳回指定日期的月份，範圍介於 1 到 12 （其中 1 代表 1 月，12 代表 12 月）。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

同義詞： MONTH

範例：

```
os> source=people | eval `MONTH_OF_YEAR(DATE('2020-08-26'))` =
 MONTH_OF_YEAR(DATE('2020-08-26')) | fields `MONTH_OF_YEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| MONTH_OF_YEAR(DATE('2020-08-26'))   |
|-----------------------------------|
| 8                                 |
+-----------------------------------+
```

NOW

用量：以 'YYYY-MM-DD hh：mm：ss' 格式TIMESTAMP的值NOW傳回目前的日期和時間。此值以叢集時區表示。

> **Note**
>
> NOW() 會傳回常數時間，指出陳述式何時開始執行。這與 不同SYSDATE()，它會傳回確切的執行時間。

傳回類型：TIMESTAMP

規格：NOW() -> TIMESTAMP

範例：

```
os> source=people | eval `value_1` = NOW(), `value_2` = NOW() | fields `value_1`,
 `value_2`
fetched rows / total rows = 1/1
+---------------------+---------------------+
| value_1             | value_2             |
|---------------------+---------------------|
| 2022-08-02 15:39:05 | 2022-08-02 15:39:05 |
+---------------------+---------------------+
```

QUARTER

用量： QUARTER(DATE)傳回指定日期的年份季度為整數，範圍介於 1 到 4。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

範例：

```
os> source=people | eval `QUARTER(DATE('2020-08-26'))` = QUARTER(DATE('2020-08-26')) |
 fields `QUARTER(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+------------------------------+
| QUARTER(DATE('2020-08-26'))  |
|------------------------------|
| 3                            |
+------------------------------+
```

SECOND

用量： 會以整數的形式SECOND(TIME)傳回指定時間的第二個元件，範圍介於 0 到 59。

引數類型：STRING/TIME/TIMESTAMP

傳回類型：INTEGER

同義詞： SECOND_OF_MINUTE

範例：

```
os> source=people | eval `SECOND(TIME('01:02:03'))` = SECOND(TIME('01:02:03')) | fields
 `SECOND(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+----------------------------+
| SECOND(TIME('01:02:03'))   |
|----------------------------|
| 3                          |
+----------------------------+
```

SECOND_OF_MINUTE

用量： 以整數SECOND_OF_MINUTE(TIME)傳回指定時間的第二個元件，範圍介於 0 到 59。

引數類型：STRING/TIME/TIMESTAMP

傳回類型：INTEGER

同義詞： SECOND

範例：

```
os> source=people | eval `SECOND_OF_MINUTE(TIME('01:02:03'))` =
 SECOND_OF_MINUTE(TIME('01:02:03')) | fields `SECOND_OF_MINUTE(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+-------------------------------------+
| SECOND_OF_MINUTE(TIME('01:02:03'))   |
|-------------------------------------|
| 3                                   |
+-------------------------------------+
```

SUBDATE

用量：從指定日期SUBDATE(DATE, DAYS)減去第二個引數（例如 DATE或 DAYS)。

引數類型：DATE/TIMESTAMP、LONG

傳回類型映射：(DATE、LONG) -> DATE

Antonyms： ADDDATE

範例：

```
os> source=people | eval `'2008-01-02' - 31d` = SUBDATE(DATE('2008-01-02'), 31),
 `'2020-08-26' - 1` = SUBDATE(DATE('2020-08-26'), 1), `ts '2020-08-26 01:01:01' -
 1` = SUBDATE(TIMESTAMP('2020-08-26 01:01:01'), 1) | fields `'2008-01-02' - 31d`,
 `'2020-08-26' - 1`, `ts '2020-08-26 01:01:01' - 1`
fetched rows / total rows = 1/1
+----------------------+--------------------+--------------------------------+
| '2008-01-02' - 31d   | '2020-08-26' - 1   | ts '2020-08-26 01:01:01' - 1   |
|----------------------+--------------------+--------------------------------|
| 2007-12-02 00:00:00  | 2020-08-25         | 2020-08-25 01:01:01            |
+----------------------+--------------------+--------------------------------+
```

SYSDATE

用量：以 'YYYY-MM-DD hh：mm：ss.nnnnnn' 格式TIMESTAMP的值SYSDATE( )傳回目前的日期和時間。

SYSDATE( )會傳回其執行的確切時間。這與 NOW() 不同，它會傳回常數時間，指出陳述式何時開始執行。

選用引數類型：INTEGER (0 到 6) - 指定傳回值中小數秒的位數。

傳回類型：TIMESTAMP

範例：

```
os> source=people | eval `SYSDATE()` = SYSDATE() | fields `SYSDATE()`
fetched rows / total rows = 1/1
+----------------------------+
| SYSDATE()                  |
|----------------------------|
| 2022-08-02 15:39:05.123456 |
+----------------------------+
```

TIMESTAMP

用量： TIMESTAMP(EXPR) 建構時間戳記類型，並以輸入字串expr做為時間戳記。

使用單一引數， TIMESTAMP(expr) 會從輸入建構時間戳記。如果 expr 是字串，則會解譯為時間戳記。對於非字串引數，函數expr會使用 UTC 時區轉換為時間戳記。當 expr 是TIME值時，函數會在轉換之前套用今天的日期。

與兩個引數搭配使用時， 會將時間表達式 (expr2) TIMESTAMP(expr1, expr2)新增至日期或時間戳記表達式 (expr1)，並將結果傳回為時間戳記值。

引數類型：STRING/DATE/TIME/TIMESTAMP

傳回類型映射：

(STRING/DATE/TIME/時間戳記） -> 時間戳記

(STRING/DATE/TIME/TIMESTAMP、STRING/DATE/TIME/TIMESTAMP) -> 時間戳記

範例：

```
os> source=people | eval `TIMESTAMP('2020-08-26 13:49:00')` = TIMESTAMP('2020-08-26
 13:49:00'), `TIMESTAMP('2020-08-26 13:49:00', TIME('12:15:42'))` =
 TIMESTAMP('2020-08-26 13:49:00', TIME('12:15:42')) | fields `TIMESTAMP('2020-08-26
 13:49:00')`, `TIMESTAMP('2020-08-26 13:49:00', TIME('12:15:42'))`
fetched rows / total rows = 1/1
+-----------------------------------
+-----------------------------------------------------+
| TIMESTAMP('2020-08-26 13:49:00')   | TIMESTAMP('2020-08-26 13:49:00',
 TIME('12:15:42'))   |
|-----------------------------------
+-----------------------------------------------------|
| 2020-08-26 13:49:00                | 2020-08-27 02:04:42
     |
+-----------------------------------
+-----------------------------------------------------+
```

UNIX_TIMESTAMP

用量： 將指定日期引數UNIX_TIMESTAMP轉換為 Unix 時間 （ 自 1970 年初開始的 Epoch 起的秒數）。如果未提供引數，則會傳回目前的 Unix 時間。

日期引數可以是 DATE、TIMESTAMP字串或下列其中一種格式的數字：YYMMDD、YYYYMMDD、YYMMDDhhmmss或 YYYYMMDDhhmmss。如果引數包含時間元件，則可能選擇性地包含小數秒。

如果引數的格式無效，或超出 1970-01-01 00：00：00 到 3001-01-18 23：59：59.999999 的範圍 (epoch 時間為 0 到 32536771199.999999)，則函數會傳回 NULL。

函數接受 DATE、 TIMESTAMP或 DOUBLE作為引數類型，或不接受引數。它一律會傳回代表 Unix 時間戳記DOUBLE的值。

對於反向轉換，您可以使用 FROM_UNIXTIME 函數。

引數類型：<NONE>/DOUBLE/DATE/TIMESTAMP

傳回類型：DOUBLE

範例：

```
os> source=people | eval `UNIX_TIMESTAMP(double)` = UNIX_TIMESTAMP(20771122143845),
 `UNIX_TIMESTAMP(timestamp)` = UNIX_TIMESTAMP(TIMESTAMP('1996-11-15 17:05:42')) |
 fields `UNIX_TIMESTAMP(double)`, `UNIX_TIMESTAMP(timestamp)`
fetched rows / total rows = 1/1
+--------------------------+-----------------------------+
| UNIX_TIMESTAMP(double)   | UNIX_TIMESTAMP(timestamp)   |
|--------------------------+-----------------------------|
| 3404817525.0             | 848077542.0                 |
+--------------------------+-----------------------------+
```

WEEK

用量：WEEK(DATE)傳回指定日期的週數。

引數類型：DATE/TIMESTAMP/STRING

傳回類型：INTEGER

同義詞： WEEK_OF_YEAR

範例：

```
os> source=people | eval `WEEK(DATE('2008-02-20'))` = WEEK(DATE('2008-02-20')) | fields
 `WEEK(DATE('2008-02-20'))`
fetched rows / total rows = 1/1
+----------------------------+
| WEEK(DATE('2008-02-20'))   |
|----------------------------|
| 8                          |
+----------------------------+
```

WEEKDAY

用量： WEEKDAY(DATE) 傳回日期的工作日索引 (0 = 星期一，1 = 星期二，...，6 = 星期日）。

它類似於 dayofweek函數，但每天都會傳回不同的索引。

引數類型：STRING/DATE/TIME/TIMESTAMP

傳回類型：INTEGER

範例：

```
os> source=people | eval `weekday(DATE('2020-08-26'))` = weekday(DATE('2020-08-26'))
 | eval `weekday(DATE('2020-08-27'))` = weekday(DATE('2020-08-27')) | fields
 `weekday(DATE('2020-08-26'))`, `weekday(DATE('2020-08-27'))`
fetched rows / total rows = 1/1
+------------------------------+------------------------------+
| weekday(DATE('2020-08-26'))  | weekday(DATE('2020-08-27'))  |
|------------------------------+------------------------------|
| 2                            | 3                            |
+------------------------------+------------------------------+
```

WEEK_OF_YEAR

用量：WEEK_OF_YEAR(DATE)傳回指定日期的週數。

引數類型：DATE/TIMESTAMP/STRING

傳回類型：INTEGER

同義詞： WEEK

範例：

```
os> source=people | eval `WEEK_OF_YEAR(DATE('2008-02-20'))` = WEEK(DATE('2008-02-20'))|
 fields `WEEK_OF_YEAR(DATE('2008-02-20'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| WEEK_OF_YEAR(DATE('2008-02-20'))  |
|-----------------------------------|
| 8                                 |
+-----------------------------------+
```

YEAR

用量：YEAR(DATE)傳回日期的年份，範圍介於 1000 到 9999 之間，或傳回 "zero" 日期的年份。

引數類型：STRING/DATE/TIMESTAMP

傳回類型：INTEGER

範例：

```
os> source=people | eval `YEAR(DATE('2020-08-26'))` = YEAR(DATE('2020-08-26')) | fields
 `YEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+----------------------------+
| YEAR(DATE('2020-08-26'))   |
|----------------------------|
| 2020                       |
+----------------------------+
```

DATE_ADD

用量： 會將指定的間隔DATE_ADD(date, INTERVAL expr unit)新增至指定的日期。

引數類型：DATE、INTERVAL

傳回類型：DATE

Antonyms： DATE_SUB

範例：

```
os> source=people | eval `'2020-08-26' + 1d` = DATE_ADD(DATE('2020-08-26'), INTERVAL 1
 DAY) | fields `'2020-08-26' + 1d`
fetched rows / total rows = 1/1
+---------------------+
| '2020-08-26' + 1d   |
|---------------------|
| 2020-08-27          |
+---------------------+
```

DATE_SUB

用量：從日期DATE_SUB(date, INTERVAL expr unit)減去間隔 expr。

引數類型：DATE、INTERVAL

傳回類型：DATE

Antonyms： DATE_ADD

範例：

```
os> source=people | eval `'2008-01-02' - 31d` = DATE_SUB(DATE('2008-01-02'), INTERVAL
 31 DAY) | fields `'2008-01-02' - 31d`
fetched rows / total rows = 1/1
+---------------------+
| '2008-01-02' - 31d  |
|---------------------|
| 2007-12-02          |
+---------------------+
```

TIMESTAMPADD

用量：在將指定的時間間隔新增至指定日期後傳回TIMESTAMP值。

引數：

- 間隔：Interval（秒、分鐘、小時、日、週、月、季、年）
- 整數：INTEGER
- 日期：DATE、TIMESTAMP 或 STRING

如果您提供 STRING做為日期引數，請將其格式化為有效的 TIMESTAMP。函數會自動將DATE引數轉換為 TIMESTAMP。

範例：

```
os> source=people | eval `TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00')` =
 TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00') | eval `TIMESTAMPADD(QUARTER, -1,
 '2000-01-01 00:00:00')` = TIMESTAMPADD(QUARTER, -1, '2000-01-01 00:00:00') | fields
 `TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00')`, `TIMESTAMPADD(QUARTER, -1, '2000-01-01
 00:00:00')`
fetched rows / total rows = 1/1
+-----------------------------------------------
+-------------------------------------------------+
| TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00') | TIMESTAMPADD(QUARTER, -1, '2000-01-01
 00:00:00') |
|-----------------------------------------------
+-------------------------------------------------|
```

```
| 2000-01-18 00:00:00                          | 1999-10-01 00:00:00
          |
+----------------------------------------------
+-----------------------------------------------+
```

TIMESTAMPDIFF

用量：以指定的間隔單位TIMESTAMPDIFF(interval, start, end)傳回開始和結束日期/時間之間的差異。

引數：

- 間隔：Interval（秒、分鐘、小時、日、週、月、季、年）
- start：DATE、TIMESTAMP 或 STRING
- 結束：DATE、TIMESTAMP 或 STRING

函數會在適當TIMESTAMP時自動將引數轉換為 。將STRING引數格式化為有效的 TIMESTAMP。

範例：

```
os> source=people | eval `TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00', '2001-03-06
 00:00:00')` = TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00', '2001-03-06 00:00:00') |
 eval `TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'), timestamp('1997-01-01
 00:00:00'))` = TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'),
 timestamp('1997-01-01 00:00:00')) | fields `TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00',
 '2001-03-06 00:00:00')`, `TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'),
 timestamp('1997-01-01 00:00:00'))`
fetched rows / total rows = 1/1
+-----------------------------------------------------------------
+-----------------------------------------------------------------------------------------
+
| TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00', '2001-03-06 00:00:00') |
 TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'), timestamp('1997-01-01
 00:00:00')) |
|-----------------------------------------------------------------
+-----------------------------------------------------------------------------------------|
| 4                                                                | -23
                                                                        |
+-----------------------------------------------------------------
+-----------------------------------------------------------------------------------------
+
```

UTC_TIMESTAMP

用量： 以 'YYYY-MM-DD hh：mm：ss' 的值UTC_TIMESTAMP傳回目前的 UTC 時間戳記。

傳回類型：TIMESTAMP

規格：UTC_TIMESTAMP() -> TIMESTAMP

範例：

```
> source=people | eval `UTC_TIMESTAMP()` = UTC_TIMESTAMP() | fields `UTC_TIMESTAMP()`
fetched rows / total rows = 1/1
+---------------------+
| UTC_TIMESTAMP()     |
|---------------------|
| 2022-10-03 17:54:28 |
+---------------------+
```

CURRENT_TIMEZONE

用量：CURRENT_TIMEZONE傳回目前的本機時區。

傳回類型：STRING

範例：

```
> source=people | eval `CURRENT_TIMEZONE()` = CURRENT_TIMEZONE() | fields
 `CURRENT_TIMEZONE()`
fetched rows / total rows = 1/1
+------------------------+
| CURRENT_TIMEZONE()     |
|------------------------|
| America/Chicago        |
+------------------------+
```

PPL 表達式

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

運算式，特別是值運算式，會傳回純量值。表達式有不同的類型和形式。例如，有以原子表達式和算術、述詞和函數表達式為基礎的常值。您可以在不同的子句中使用表達式，例如在 Filter和 Stats命令中使用算術表達式。

運算子

算術表達式是由數值常值和二進位算術運算子形成的表達式，如下所示：

1. +：新增。
2. -：減去。
3. *：乘以。
4. /：除數（對於整數，結果是捨棄分數部分的整數）
5. %：Modulo（僅與整數搭配使用；結果是分割的其餘部分）

優先順序

使用括號來控制算術運算子的優先順序。否則，會先執行優先順序較高的運算子。

類型轉換

查詢運算子簽章時，會執行隱含類型轉換。例如，實數+的整數會比對+(double,double)產生實數的簽章。此規則也適用於函數呼叫。

不同類型的算術表達式範例：

```
os> source=accounts | where age > (25 + 5) | fields age ;
fetched rows / total rows = 3/3
+-------+
| age   |
|-------|
| 32    |
| 36    |
| 33    |
+-------+
```

述詞運算子

述詞運算子是評估為 true 的表達式。MISSING 和 NULL值比較遵循下列規則：

- MISSING 值僅等於MISSING值，且小於其他值。
- NULL 值等於NULL值，大於MISSING值，但小於所有其他值。

運算子

述詞運算子

名稱	描述
>	大於運算子
>=	大於或等於運算子
<	小於運算子
!=	不等於運算子
<=	小於或等於運算子
=	等於運算子
LIKE	簡單模式比對
IN	NULL 值測試
AND	AND 運算子
OR	OR 運算子
XOR	XOR 運算子
NOT	NOT NULL 值測試

您可以比較日期時間。比較不同的日期時間類型 （例如 DATE和 TIME) 時，兩者都會轉換為 DATETIME。下列規則適用於轉換：

- TIME 適用於今天的日期。
- DATE 會在午夜解譯。

基本述詞運算子

比較運算子的範例：

```
os> source=accounts | where age > 33 | fields age ;
```

```
fetched rows / total rows = 1/1
+-------+
| age   |
|-------|
| 36    |
+-------+
```

IN

值清單中運算IN子測試欄位的範例：

```
os> source=accounts | where age in (32, 33) | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 32    |
| 33    |
+-------+
```

OR

OR 運算子的範例：

```
os> source=accounts | where age = 32 OR age = 33 | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 32    |
| 33    |
+-------+
```

NOT

NOT 運算子的範例：

```
os> source=accounts | where age not in (32, 33) | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 36    |
```

```
| 28    |
+-------+
```

PPL IP 地址函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

CIDRMATCH

用量：CIDRMATCH(ip, cidr)檢查指定的 IP 地址是否在指定的 cidr 範圍內。

引數類型：

- STRING、STRING
- 傳回類型：BOOLEAN

範例：

```
os> source=ips | where cidrmatch(ip, '***********/24') | fields ip
fetched rows / total rows = 1/1
+--------------+
| ip           |
|--------------|
| ***********  |
+--------------+

os> source=ipsv6 | where cidrmatch(ip, '2003:db8::/32') | fields ip
fetched rows / total rows = 1/1
+-----------------------------------------+
| ip                                      |
|-----------------------------------------|
| 2003:0db8:****:****:****:****:****:0000 |
+-----------------------------------------+
```

> **Note**
>
> - ip 可以是 IPv4 或 IPv6 地址。

- `cidr` 可以是 IPv4 或 IPv6 區塊。
- `ip` 和 `cidr` 必須是 IPv4 或 IPv6。
- `ip` 和 `cidr` 都必須有效且非空白/非空。

PPL JSON 函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

`JSON`

用量： 會`json(value)`評估字串是否可以剖析為 JSON 格式。如果原始字串是有效的 JSON，則函數會傳回原始字串，如果無效，則傳回 null。

引數類型：STRING

傳回類型：STRING/NULL。有效 JSON 物件格式的 STRING 表達式。

範例：

```
os> source=people | eval `valid_json()` = json('[1,2,3,{"f1":1,"f2":[5,6]},4]') |
 fields valid_json
fetched rows / total rows = 1/1
+----------------------------------+
| valid_json                       |
+----------------------------------+
| [1,2,3,{"f1":1,"f2":[5,6]},4]    |
+----------------------------------+

os> source=people | eval `invalid_json()` = json('{"invalid": "json"') | fields
 invalid_json
fetched rows / total rows = 1/1
+----------------+
| invalid_json   |
+----------------+
| null           |
+----------------+
```

JSON_OBJECT

用量：從鍵值對的成員`json_object(<key>, <value>[, <key>, <value>]...)`傳回 JSON 物件。

引數類型：

- <key> 必須是 STRING。
- <value> 可以是任何資料類型。

傳回類型：JSON_OBJECT。有效 JSON 物件的 StructType 表達式。

範例：

```
os> source=people | eval result = json_object('key', 123.45) | fields result
fetched rows / total rows = 1/1
+------------------+
| result           |
+------------------+
| {"key":123.45}   |
+------------------+

os> source=people | eval result = json_object('outer', json_object('inner', 123.45)) |
 fields result
fetched rows / total rows = 1/1
+------------------------------+
| result                       |
+------------------------------+
| {"outer":{"inner":123.45}}   |
+------------------------------+
```

JSON_ARRAY

用量： 使用值清單`json_array(<value>...)`建立 JSON ARRAY。

引數類型： `<value>`可以是任何類型的值，例如字串、數字或布林值。

傳回類型：ARRAY。有效 JSON 陣列的任何支援資料類型的陣列。

範例：

```
os> source=people | eval `json_array` = json_array(1, 2, 0, -1, 1.1, -0.11)
```

```
fetched rows / total rows = 1/1
+------------------------------+
| json_array                   |
+------------------------------+
| [1.0,2.0,0.0,-1.0,1.1,-0.11] |
+------------------------------+

os> source=people | eval `json_array_object` = json_object("array", json_array(1, 2, 0,
 -1, 1.1, -0.11))
fetched rows / total rows = 1/1
+--------------------------------------+
| json_array_object                    |
+--------------------------------------+
| {"array":[1.0,2.0,0.0,-1.0,1.1,-0.11]} |
+--------------------------------------+
```

TO_JSON_STRING

用量：to_json_string(jsonObject)傳回具有指定 json 物件值的 JSON 字串。

引數類型：JSON_OBJECT

傳回類型：STRING

範例：

```
os> source=people | eval `json_string` = to_json_string(json_array(1, 2, 0, -1, 1.1,
 -0.11)) | fields json_string
fetched rows / total rows = 1/1
+--------------------------------+
| json_string                    |
+--------------------------------+
| [1.0,2.0,0.0,-1.0,1.1,-0.11]   |
+--------------------------------+

os> source=people | eval `json_string` = to_json_string(json_object('key', 123.45)) |
 fields json_string
fetched rows / total rows = 1/1
+-----------------+
| json_string     |
+-----------------+
| {'key', 123.45} |
+-----------------+
```

ARRAY_LENGTH

用量：array_length(jsonArray)傳回最外部陣列中的元素數目。

引數類型：ARRAY。ARRAY 或 JSON_ARRAY 物件。

傳回類型：INTEGER

範例：

```
os> source=people | eval `json_array` = json_array_length(json_array(1,2,3,4)),
 `empty_array` = json_array_length(json_array())
fetched rows / total rows = 1/1
+--------------+---------------+
| json_array   | empty_array   |
+--------------+---------------+
| 4            | 0             |
+--------------+---------------+
```

JSON_EXTRACT

用量： 根據指定的 JSON 路徑，從 JSON 字串json_extract(jsonStr, path)擷取 JSON 物件。如果輸入 JSON 字串無效，函數會傳回 null。

引數類型：STRING、STRING

傳回類型：STRING

- 有效 JSON 物件格式的 STRING 表達式。
- NULL 會在 JSON 無效時傳回。

範例：

```
os> source=people | eval `json_extract('{"a":"b"}', '$.a')` = json_extract('{"a":"b"}',
 '$a')
fetched rows / total rows = 1/1
+---------------------------------+
| json_extract('{"a":"b"}', 'a')   |
+---------------------------------+
| b                               |
```

```
+----------------------------------+

os> source=people | eval `json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[1].b')` =
 json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[1].b')
fetched rows / total rows = 1/1
+-------------------------------------------------------------+
| json_extract('{"a":[{"b":1.0},{"b":2.0}]}', '$.a[1].b')   |
+-------------------------------------------------------------+
| 2.0                                                         |
+-------------------------------------------------------------+

os> source=people | eval `json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[*].b')` =
 json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[*].b')
fetched rows / total rows = 1/1
+-------------------------------------------------------------+
| json_extract('{"a":[{"b":1.0},{"b":2.0}]}', '$.a[*].b')   |
+-------------------------------------------------------------+
| [1.0,2.0]                                                   |
+-------------------------------------------------------------+

os> source=people | eval `invalid_json` = json_extract('{"invalid": "json"')
fetched rows / total rows = 1/1
+----------------+
| invalid_json   |
+----------------+
| null           |
+----------------+
```

JSON_KEYS

用量：json_keys(jsonStr)傳回最外部 JSON 物件的所有索引鍵做為陣列。

引數類型：STRING。有效 JSON 物件格式的 STRING 表達式。

傳回類型：ARRAY【STRING】。函數NULL會傳回任何其他有效的 JSON 字串、空字串或無效的 JSON。

範例：

```
os> source=people | eval `keys` = json_keys('{"f1":"abc","f2":{"f3":"a","f4":"b"}}')
fetched rows / total rows = 1/1
+------------+
| keus       |
```

```
+------------+
| [f1, f2]   |
+------------+

os> source=people | eval `keys` = json_keys('[1,2,3,{"f1":1,"f2":[5,6]},4]')
fetched rows / total rows = 1/1
+--------+
| keys   |
+--------+
| null   |
+--------+
```

JSON_VALID

用量： json_valid(jsonStr) 評估 JSON 字串是否使用有效的 JSON 語法，並傳回 TRUE 或 FALSE。

引數類型：STRING

傳回類型：BOOLEAN

範例：

```
os> source=people | eval `valid_json` = json_valid('[1,2,3,4]'), `invalid_json` =
 json_valid('{"invalid": "json"') | feilds `valid_json`, `invalid_json`
fetched rows / total rows = 1/1
+--------------+----------------+
| valid_json   | invalid_json   |
+--------------+----------------+
| True         | False          |
+--------------+----------------+

os> source=accounts | where json_valid('[1,2,3,4]') and isnull(email) | fields
 account_number, email
fetched rows / total rows = 1/1
+------------------+---------+
| account_number   | email   |
|------------------+---------|
| 13               | null    |
+------------------+---------+
```

PPL Lambda 函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

EXISTS

用量：`exists(array, lambda)`評估 Lambda 述詞是否保留陣列中的一或多個元素。

引數類型：ARRAY、LAMBDA

傳回類型：BOOLEAN。TRUE 如果陣列中至少有一個元素符合 Lambda 述詞，則傳回 ，否則傳回 FALSE。

範例：

```
 os> source=people | eval array = json_array(1, -1, 2), result = exists(array, x -> x >
 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| true      |
+-----------+

os> source=people | eval array = json_array(-1, -3, -2), result = exists(array, x -> x
 > 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| false     |
+-----------+
```

FILTER

用量：使用指定的 Lambda 函數`filter(array, lambda)`篩選輸入陣列。

引數類型：ARRAY、LAMBDA

傳回類型：ARRAY。包含輸入陣列中符合 Lambda 述詞之所有元素的 ARRAY。

範例：

```
 os> source=people | eval array = json_array(1, -1, 2), result = filter(array, x -> x >
 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| [1, 2]    |
+-----------+

os> source=people | eval array = json_array(-1, -3, -2), result = filter(array, x -> x
 > 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| []        |
+-----------+
```

TRANSFORM

用量： 使用 Lambda `transform(array, lambda)`轉換函數轉換陣列中的元素。如果使用二進位 Lambda 函數，第二個引數表示 元素的索引。這類似於功能程式設計`map`中的 。

引數類型：ARRAY、LAMBDA

傳回類型：ARRAY。包含將 Lambda 轉換函數套用至輸入陣列中每個元素之結果的 ARRAY。

範例：

```
os> source=people | eval array = json_array(1, 2, 3), result = transform(array, x -> x
 + 1) | fields result
fetched rows / total rows = 1/1
+--------------+
| result       |
+--------------+
| [2, 3, 4]    |
+--------------+

os> source=people | eval array = json_array(1, 2, 3), result = transform(array, (x, i)
 -> x + i) | fields result
fetched rows / total rows = 1/1
```

```
+--------------+
| result       |
+--------------+
| [1, 3, 5]    |
+--------------+
```

REDUCE

用量：套用 lambda 函數，將陣列`reduce(array, start, merge_lambda, finish_lambda)`減少為單一值。函數會將 merge_lambda 套用至開始值和所有陣列元素，然後將 `finish_lambda` 套用至結果。

引數類型：ARRAY、 ANY、LAMBDA、LAMBDA

傳回類型： ANY。將 Lambda 函數套用至開始值和輸入陣列的最終結果。

範例：

```
 os> source=people | eval array = json_array(1, 2, 3), result = reduce(array, 0, (acc,
 x) -> acc + x) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| 6         |
+-----------+

os> source=people | eval array = json_array(1, 2, 3), result = reduce(array, 10, (acc,
 x) -> acc + x) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| 16        |
+-----------+

os> source=people | eval array = json_array(1, 2, 3), result = reduce(array, 0, (acc,
 x) -> acc + x, acc -> acc * 10) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| 60        |
```

```
+-----------+
```

PPL 數學函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

ABS

用量： ABS(x) 計算 x 的絕對值。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：INTEGER/LONG/FLOAT/DOUBLE

範例：

```
os> source=people | eval `ABS(-1)` = ABS(-1) | fields `ABS(-1)`
fetched rows / total rows = 1/1
+-----------+
| ABS(-1)   |
|-----------|
| 1         |
+-----------+
```

ACOS

用量： ACOS(x) 計算 x 的電弧餘弦。NULL 如果 x 不在 -1 到 1 的範圍內，則傳回 。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `ACOS(0)` = ACOS(0) | fields `ACOS(0)`
fetched rows / total rows = 1/1
+--------------------+
| ACOS(0)            |
|--------------------|
| 1.5707963267948966 |
```

```
+--------------------+
```

ASIN

用量： asin(x) 計算 x 的電弧正弦。NULL 如果 x 不在 -1 到 1 的範圍內，則傳回 。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `ASIN(0)` = ASIN(0) | fields `ASIN(0)`
fetched rows / total rows = 1/1
+-----------+
| ASIN(0)   |
|-----------|
| 0.0       |
+-----------+
```

ATAN

用量： ATAN(x) 計算 x 的電弧切線。 atan(y, x)計算 y/x 的電弧切線，但兩個引數的符號都會決定結果的四分之一。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `ATAN(2)` = ATAN(2), `ATAN(2, 3)` = ATAN(2, 3) | fields
 `ATAN(2)`, `ATAN(2, 3)`
fetched rows / total rows = 1/1
+--------------------+--------------------+
| ATAN(2)            | ATAN(2, 3)         |
|--------------------+--------------------|
| 1.1071487177940904 | 0.5880026035475675 |
+--------------------+--------------------+
```

ATAN2

用量： ATAN2(y, x)會計算 y/x 的電弧切線，但兩個引數的符號都會決定結果的四分之一。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `ATAN2(2, 3)` = ATAN2(2, 3) | fields `ATAN2(2, 3)`
fetched rows / total rows = 1/1
+--------------------+
| ATAN2(2, 3)        |
|--------------------|
| 0.5880026035475675 |
+--------------------+
```

CBRT

用量： CBRT 計算數字的立方體根。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE：

INTEGER/LONG/FLOAT/DOUBLE -> 雙

範例：

```
opensearchsql> source=location | eval `CBRT(8)` = CBRT(8), `CBRT(9.261)` = CBRT(9.261),
 `CBRT(-27)` = CBRT(-27) | fields `CBRT(8)`, `CBRT(9.261)`, `CBRT(-27)`;
fetched rows / total rows = 2/2
+-----------+---------------+-------------+
| CBRT(8)   | CBRT(9.261)   | CBRT(-27)   |
|-----------+---------------+-------------|
| 2.0       | 2.1           | -3.0        |
| 2.0       | 2.1           | -3.0        |
+-----------+---------------+-------------+
```

CEIL

用量：CEILING函數的別名。 CEILING(T)採用值 T 的上限。

限制：CEILING只有在 IEEE 754 雙重類型在儲存時顯示小數時，才能如預期運作。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：LONG

範例：

```
os> source=people | eval `CEILING(0)` = CEILING(0), `CEILING(50.00005)` =
 CEILING(50.00005), `CEILING(-50.00005)` = CEILING(-50.00005) | fields `CEILING(0)`,
 `CEILING(50.00005)`, `CEILING(-50.00005)`
fetched rows / total rows = 1/1
+--------------+---------------------+----------------------+
| CEILING(0)   | CEILING(50.00005)   | CEILING(-50.00005)   |
|--------------+---------------------+----------------------|
| 0            | 51                  | -50                  |
+--------------+---------------------+----------------------+

os> source=people | eval `CEILING(3147483647.12345)` = CEILING(3147483647.12345),
 `CEILING(113147483647.12345)` = CEILING(113147483647.12345),
 `CEILING(3147483647.00001)` = CEILING(3147483647.00001) | fields
 `CEILING(3147483647.12345)`, `CEILING(113147483647.12345)`,
 `CEILING(3147483647.00001)`
fetched rows / total rows = 1/1
+-----------------------------+-------------------------------
+-----------------------------+
| CEILING(3147483647.12345)   | CEILING(113147483647.12345)   |
 CEILING(3147483647.00001)   |
|-----------------------------+-------------------------------
+-----------------------------|
| 3147483648                  | 113147483648                  | 3147483648
     |
+-----------------------------+-------------------------------
+-----------------------------+
```

CONV

用量： 將數字 x 從基礎CONV(x, a, b)轉換為 b 基礎。

引數類型：x：STRING、a：INTEGER、b：INTEGER

傳回類型：STRING

範例：

```
os> source=people | eval `CONV('12', 10, 16)` = CONV('12', 10, 16), `CONV('2C', 16,
 10)` = CONV('2C', 16, 10), `CONV(12, 10, 2)` = CONV(12, 10, 2), `CONV(1111, 2, 10)` =
```

```
 CONV(1111, 2, 10) | fields `CONV('12', 10, 16)`, `CONV('2C', 16, 10)`, `CONV(12, 10,
 2)`, `CONV(1111, 2, 10)`
fetched rows / total rows = 1/1
+---------------------+---------------------+------------------
+---------------------+
| CONV('12', 10, 16)   | CONV('2C', 16, 10)   | CONV(12, 10, 2)   | CONV(1111, 2, 10)
 |
|---------------------+---------------------+------------------
+---------------------|
| c                    | 44                   | 1100              | 15
 |
+---------------------+---------------------+------------------
+---------------------+
```

COS

用量： COS(x)計算 x 的餘弦，其中 x 以弧度表示。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `COS(0)` = COS(0) | fields `COS(0)`
fetched rows / total rows = 1/1
+----------+
| COS(0)   |
|----------|
| 1.0      |
+----------+
```

COT

用量： COT(x) 計算 x 的餘切。如果 x out-of-range的錯誤。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `COT(1)` = COT(1) | fields `COT(1)`
```

```
fetched rows / total rows = 1/1
+--------------------+
| COT(1)             |
|--------------------|
| 0.6420926159343306 |
+--------------------+
```

CRC32

用量： CRC32計算循環冗餘檢查值，並傳回 32 位元的未簽署值。

引數類型：STRING

傳回類型：LONG

範例：

```
os> source=people | eval `CRC32('MySQL')` = CRC32('MySQL') | fields `CRC32('MySQL')`
fetched rows / total rows = 1/1
+------------------+
| CRC32('MySQL')   |
|------------------|
| 3259397556       |
+------------------+
```

DEGREES

用量： 將 x 從弧度DEGREES(x)轉換為度數。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `DEGREES(1.57)` = DEGREES(1.57) | fields `DEGREES(1.57)`
fetched rows / total rows  = 1/1
+-------------------+
| DEGREES(1.57)     |
|-------------------|
| 89.95437383553924 |
+-------------------+
```

E

用量：E()傳回 Euler 的號碼。

傳回類型：DOUBLE

範例：

```
os> source=people | eval `E()` = E() | fields `E()`
fetched rows / total rows = 1/1
+-------------------+
| E()               |
|-------------------|
| 2.718281828459045 |
+-------------------+
```

EXP

用量： EXP(x)傳回提高到 x 能力的 e。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `EXP(2)` = EXP(2) | fields `EXP(2)`
fetched rows / total rows = 1/1
+------------------+
| EXP(2)           |
|------------------|
| 7.38905609893065 |
+------------------+
```

FLOOR

用量：FLOOR(T)採用值 T 的下限。

限制：FLOOR只有在 IEEE 754 雙重類型在儲存時顯示小數時，才能如預期運作。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：LONG

範例：

```
os> source=people | eval `FLOOR(0)` = FLOOR(0), `FLOOR(50.00005)` = FLOOR(50.00005),
 `FLOOR(-50.00005)` = FLOOR(-50.00005) | fields `FLOOR(0)`, `FLOOR(50.00005)`,
 `FLOOR(-50.00005)`
fetched rows / total rows = 1/1
+------------+------------------+--------------------+
| FLOOR(0)   | FLOOR(50.00005)   | FLOOR(-50.00005)   |
|------------+------------------+--------------------|
| 0          | 50                | -51                |
+------------+------------------+--------------------+

os> source=people | eval `FLOOR(3147483647.12345)` = FLOOR(3147483647.12345),
 `FLOOR(113147483647.12345)` = FLOOR(113147483647.12345), `FLOOR(3147483647.00001)`
 = FLOOR(3147483647.00001) | fields `FLOOR(3147483647.12345)`,
 `FLOOR(113147483647.12345)`, `FLOOR(3147483647.00001)`
fetched rows / total rows = 1/1
+--------------------------+----------------------------+--------------------------+
| FLOOR(3147483647.12345)   | FLOOR(113147483647.12345)   | FLOOR(3147483647.00001)   |
|--------------------------+----------------------------+--------------------------|
| 3147483647                | 113147483647                | 3147483647                |
+--------------------------+----------------------------+--------------------------+

os> source=people | eval `FLOOR(282474973688888.022)` = FLOOR(282474973688888.022),
 `FLOOR(9223372036854775807.022)` = FLOOR(9223372036854775807.022),
 `FLOOR(9223372036854775807.0000001)` = FLOOR(9223372036854775807.0000001)
 | fields `FLOOR(282474973688888.022)`, `FLOOR(9223372036854775807.022)`,
 `FLOOR(9223372036854775807.0000001)`
fetched rows / total rows = 1/1
+-----------------------------+-----------------------------------
+-------------------------------------+
| FLOOR(282474973688888.022)   | FLOOR(9223372036854775807.022)   |
 FLOOR(9223372036854775807.0000001)   |
|-----------------------------+-----------------------------------
+-------------------------------------|
| 282474973688888              | 9223372036854775807              | 9223372036854775807
                  |
+-----------------------------+-----------------------------------
+-------------------------------------+
```

LN

用量：LN(x)傳回 x 的自然對數。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `LN(2)` = LN(2) | fields `LN(2)`
fetched rows / total rows = 1/1
+--------------------+
| LN(2)              |
|--------------------|
| 0.6931471805599453 |
+--------------------+
```

LOG

用量：LOG(x)傳回 x 的自然對數，這是 x. log(B， x) 的基本 e 對數，相當於 log(x)/log(B)。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `LOG(2)` = LOG(2), `LOG(2, 8)` = LOG(2, 8) | fields `LOG(2)`,
 `LOG(2, 8)`
fetched rows / total rows = 1/1
+--------------------+-------------+
| LOG(2)             | LOG(2, 8)   |
|--------------------+-------------|
| 0.6931471805599453 | 3.0         |
+--------------------+-------------+
```

LOG2

用量： LOG2(x) 等於 log(x)/log(2)。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `LOG2(8)` = LOG2(8) | fields `LOG2(8)`
```

```
fetched rows / total rows = 1/1
+-----------+
| LOG2(8)   |
|-----------|
| 3.0       |
+-----------+
```

LOG10

用量： LOG10(x) 等於 log(x)/log(10)。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `LOG10(100)` = LOG10(100) | fields `LOG10(100)`
fetched rows / total rows = 1/1
+--------------+
| LOG10(100)   |
|--------------|
| 2.0          |
+--------------+
```

MOD

用量： MOD(n, m) 計算 n 除以 m 的其餘數字。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：如果 m 是非零值，則 n 和 m 類型之間的類型會更寬。如果 m 等於 0，則傳回 NULL。

範例：

```
os> source=people | eval `MOD(3, 2)` = MOD(3, 2), `MOD(3.1, 2)` = MOD(3.1, 2) | fields
 `MOD(3, 2)`, `MOD(3.1, 2)`
fetched rows / total rows = 1/1
+-------------+---------------+
| MOD(3, 2)   | MOD(3.1, 2)   |
|-------------+---------------|
| 1           | 1.1           |
+-------------+---------------+
```

PI

用量：PI() 傳回常數 pi。

傳回類型：DOUBLE

範例：

```
os> source=people | eval `PI()` = PI() | fields `PI()`
fetched rows / total rows = 1/1
+-------------------+
| PI()              |
|-------------------|
| 3.141592653589793 |
+-------------------+
```

POW

用量： POW(x, y) 計算 x 的值，提高到 y 的倍數。輸入錯誤會傳回NULL結果。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

同義詞： POWER(_, _)

範例：

```
os> source=people | eval `POW(3, 2)` = POW(3, 2), `POW(-3, 2)` = POW(-3, 2), `POW(3,
 -2)` = POW(3, -2) | fields `POW(3, 2)`, `POW(-3, 2)`, `POW(3, -2)`
fetched rows / total rows = 1/1
+-------------+--------------+--------------------+
| POW(3, 2)   | POW(-3, 2)   | POW(3, -2)         |
|-------------+--------------+--------------------|
| 9.0         | 9.0          | 0.1111111111111111 |
+-------------+--------------+--------------------+
```

POWER

用量： POWER(x, y) 計算 x 的值，提高到 y 的倍數。輸入錯誤會傳回NULL結果。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

同義詞： POW(_, _)

範例：

```
os> source=people | eval `POWER(3, 2)` = POWER(3, 2), `POWER(-3, 2)` = POWER(-3, 2),
 `POWER(3, -2)` = POWER(3, -2) | fields `POWER(3, 2)`, `POWER(-3, 2)`, `POWER(3, -2)`
fetched rows / total rows = 1/1
+---------------+----------------+--------------------+
| POWER(3, 2)   | POWER(-3, 2)   | POWER(3, -2)       |
|---------------+----------------+--------------------|
| 9.0           | 9.0            | 0.1111111111111111 |
+---------------+----------------+--------------------+
```

RADIANS

用量： 將 x 從度數RADIANS(x)轉換為弧度。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `RADIANS(90)` = RADIANS(90) | fields `RADIANS(90)`
fetched rows / total rows  = 1/1
+--------------------+
| RADIANS(90)        |
|--------------------|
| 1.5707963267948966 |
+--------------------+
```

RAND

用量：RAND( )/RAND(N) 傳回範圍 0 <= 值 < 1.0 的隨機浮點值。如果您指定整數 N，函數會在執行之前初始化種子。這種行為的一個含義是，使用相同的引數 N，每次都會rand(N)傳回相同的值，產生可重複的資料欄值序列。

引數類型：INTEGER

傳回類型：FLOAT

範例：

```
os> source=people | eval `RAND(3)` = RAND(3) | fields `RAND(3)`
fetched rows / total rows = 1/1
+------------+
| RAND(3)    |
|------------|
| 0.73105735 |
+------------+
```

ROUND

用量：將引數 x ROUND(x, d)四捨五入為小數位數。如果您未指定 d，則預設為 0。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型映射：

- （整數/長 【，整數】）-> 長
- （浮點數/雙 【，整數】）-> 長

範例：

```
os> source=people | eval `ROUND(12.34)` = ROUND(12.34), `ROUND(12.34, 1)` =
 ROUND(12.34, 1), `ROUND(12.34, -1)` = ROUND(12.34, -1), `ROUND(12, 1)` = ROUND(12, 1)
 | fields `ROUND(12.34)`, `ROUND(12.34, 1)`, `ROUND(12.34, -1)`, `ROUND(12, 1)`
fetched rows / total rows = 1/1
+----------------+-------------------+--------------------+----------------+
| ROUND(12.34)   | ROUND(12.34, 1)   | ROUND(12.34, -1)   | ROUND(12, 1)   |
|----------------+-------------------+--------------------+----------------|
| 12.0           | 12.3              | 10.0               | 12             |
+----------------+-------------------+--------------------+----------------+
```

SIGN

用量： SIGN傳回引數的符號為 -1、0 或 1，取決於數字是負數、零或正數。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：INTEGER

範例：

```
os> source=people | eval `SIGN(1)` = SIGN(1), `SIGN(0)` = SIGN(0), `SIGN(-1.1)` =
 SIGN(-1.1) | fields `SIGN(1)`, `SIGN(0)`, `SIGN(-1.1)`
fetched rows / total rows = 1/1
+-----------+-----------+--------------+
| SIGN(1)   | SIGN(0)   | SIGN(-1.1)   |
|-----------+-----------+--------------|
| 1         | 0         | -1           |
+-----------+-----------+--------------+
```

SIN

用量： sin(x) 計算 x 的正弦，其中 x 以弧度表示。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型：DOUBLE

範例：

```
os> source=people | eval `SIN(0)` = SIN(0) | fields `SIN(0)`
fetched rows / total rows = 1/1
+----------+
| SIN(0)   |
|----------|
| 0.0      |
+----------+
```

SQRT

用量： SQRT 計算非負數的平方根。

引數類型：INTEGER/LONG/FLOAT/DOUBLE

傳回類型映射：

- （非負值）INTEGER/LONG/FLOAT/DOUBLE -> DOUBLE
- （負）INTEGER/LONG/FLOAT/DOUBLE -> NULL

範例：

```
os> source=people | eval `SQRT(4)` = SQRT(4), `SQRT(4.41)` = SQRT(4.41) | fields
 `SQRT(4)`, `SQRT(4.41)`
```

```
fetched rows / total rows = 1/1
+-----------+--------------+
| SQRT(4)   | SQRT(4.41)   |
|-----------+--------------|
| 2.0       | 2.1          |
+-----------+--------------+
```

PPL 字串函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

CONCAT

用量： 最多可CONCAT(str1, str2, ..., str_9)將 9 個字串加在一起。

引數類型：

- STRING、STRING、....、STRING
- 傳回類型：STRING

範例：

```
os> source=people | eval `CONCAT('hello', 'world')` = CONCAT('hello', 'world'),
 `CONCAT('hello ', 'whole ', 'world', '!')` = CONCAT('hello ', 'whole ', 'world', '!')
 | fields `CONCAT('hello', 'world')`, `CONCAT('hello ', 'whole ', 'world', '!')`
fetched rows / total rows = 1/1
+----------------------------+---------------------------------------------+
| CONCAT('hello', 'world')   | CONCAT('hello ', 'whole ', 'world', '!')    |
|----------------------------+---------------------------------------------|
| helloworld                 | hello whole world!                          |
+----------------------------+---------------------------------------------+
```

CONCAT_WS

用量：使用指定的分隔符號串CONCAT_WS(sep, str1, str2)連兩個或多個字串。

引數類型：

- STRING、STRING、....、STRING

- 傳回類型：STRING

範例：

```
os> source=people | eval `CONCAT_WS(',', 'hello', 'world')` = CONCAT_WS(',', 'hello',
 'world') | fields `CONCAT_WS(',', 'hello', 'world')`
fetched rows / total rows = 1/1
+-----------------------------------+
| CONCAT_WS(',', 'hello', 'world')  |
|-----------------------------------|
| hello,world                       |
+-----------------------------------+
```

LENGTH

用量： length(str) 傳回以位元組為單位的輸入字串長度。

引數類型：

- STRING
- 傳回類型：INTEGER

範例：

```
os> source=people | eval `LENGTH('helloworld')` = LENGTH('helloworld') | fields
 `LENGTH('helloworld')`
fetched rows / total rows = 1/1
+------------------------+
| LENGTH('helloworld')   |
|------------------------|
| 10                     |
+------------------------+
```

LOWER

用量： 將輸入字串lower(string)轉換為小寫。

引數類型：

- STRING
- 傳回類型：STRING

範例：

```
os> source=people | eval `LOWER('helloworld')` = LOWER('helloworld'),
 `LOWER('HELLOWORLD')` = LOWER('HELLOWORLD') | fields `LOWER('helloworld')`,
 `LOWER('HELLOWORLD')`
fetched rows / total rows = 1/1
+----------------------+----------------------+
| LOWER('helloworld')   | LOWER('HELLOWORLD')   |
|----------------------+----------------------|
| helloworld            | helloworld            |
+----------------------+----------------------+
```

LTRIM

用量：從輸入字串ltrim(str)中移除前導空格字元。

引數類型：

- STRING
- 傳回類型：STRING

範例：

```
os> source=people | eval `LTRIM('   hello')` = LTRIM('   hello'), `LTRIM('hello   ')` =
 LTRIM('hello   ') | fields `LTRIM('   hello')`, `LTRIM('hello   ')`
fetched rows / total rows = 1/1
+---------------------+---------------------+
| LTRIM('   hello')   | LTRIM('hello   ')   |
|---------------------+---------------------|
| hello               | hello               |
+---------------------+---------------------+
```

POSITION

用量：POSITION(substr IN str)傳回字串中第一次出現子字串的位置。如果子字串不在字串中，則傳回 0。如果任何引數為 NULL，則傳回 NULL。

引數類型：

- STRING、STRING
- 傳回類型 INTEGER

範例：

```
os> source=people | eval `POSITION('world' IN 'helloworld')` = POSITION('world'
 IN 'helloworld'), `POSITION('invalid' IN 'helloworld')`= POSITION('invalid' IN
 'helloworld')  | fields `POSITION('world' IN 'helloworld')`, `POSITION('invalid' IN
 'helloworld')`
fetched rows / total rows = 1/1
+------------------------------------+---------------------------------------+
| POSITION('world' IN 'helloworld')   | POSITION('invalid' IN 'helloworld')   |
|------------------------------------+---------------------------------------|
| 6                                  | 0                                     |
+------------------------------------+---------------------------------------+
```

REVERSE

用量：REVERSE(str)傳回輸入字串的反向字串。

引數類型：

- STRING
- 傳回類型：STRING

範例：

```
os> source=people | eval `REVERSE('abcde')` = REVERSE('abcde') | fields
 `REVERSE('abcde')`
fetched rows / total rows = 1/1
+--------------------+
| REVERSE('abcde')   |
|--------------------|
| edcba              |
+--------------------+
```

RIGHT

用量： 會從輸入字串right(str, len)傳回最右邊的字元。如果子字串不在字串中，則傳回 0。如果任何引數為 NULL，則傳回 NULL。

引數類型：

- STRING、INTEGER

- 傳回類型：STRING

範例：

```
os> source=people | eval `RIGHT('helloworld', 5)` = RIGHT('helloworld', 5),
 `RIGHT('HELLOWORLD', 0)` = RIGHT('HELLOWORLD', 0) | fields `RIGHT('helloworld', 5)`,
 `RIGHT('HELLOWORLD', 0)`
fetched rows / total rows = 1/1
+--------------------------+--------------------------+
| RIGHT('helloworld', 5)   | RIGHT('HELLOWORLD', 0)   |
|--------------------------+--------------------------|
| world                    |                          |
+--------------------------+--------------------------+
```

RTRIM

用量：從輸入字串rtrim(str)修剪結尾空格字元。

引數類型：

- STRING
- 傳回類型：STRING

範例：

```
os> source=people | eval `RTRIM('   hello')` = RTRIM('   hello'), `RTRIM('hello   ')` =
 RTRIM('hello   ') | fields `RTRIM('   hello')`, `RTRIM('hello   ')`
fetched rows / total rows = 1/1
+----------------------+----------------------+
| RTRIM('   hello')   | RTRIM('hello   ')   |
|----------------------+----------------------|
|    hello            | hello                |
+----------------------+----------------------+
```

SUBSTRING

用量： substring(str, start)或 substring(str, start, length)會傳回輸入字串的子字串。在沒有指定長度的情況下，它會從開始位置傳回整個字串。

引數類型：

- STRING、INTEGER、INTEGER
- 傳回類型：STRING

範例：

```
os> source=people | eval `SUBSTRING('helloworld', 5)` = SUBSTRING('helloworld',
 5), `SUBSTRING('helloworld', 5, 3)` = SUBSTRING('helloworld', 5, 3) | fields
 `SUBSTRING('helloworld', 5)`, `SUBSTRING('helloworld', 5, 3)`
fetched rows / total rows = 1/1
+------------------------------+---------------------------------+
| SUBSTRING('helloworld', 5)   | SUBSTRING('helloworld', 5, 3)   |
|------------------------------+---------------------------------|
| oworld                       | owo                             |
+------------------------------+---------------------------------+
```

TRIM

用量：從輸入字串trim(string)中移除前後空格。

引數類型：

- STRING
- 傳回類型：STRING

範例：

```
os> source=people | eval `TRIM('   hello')` = TRIM('   hello'), `TRIM('hello   ')` =
 TRIM('hello   ') | fields `TRIM('   hello')`, `TRIM('hello   ')`
fetched rows / total rows = 1/1
+--------------------+--------------------+
| TRIM('   hello')   | TRIM('hello   ')   |
|--------------------+--------------------|
| hello              | hello              |
+--------------------+--------------------+
```

UPPER

用量： 會將輸入字串upper(string)轉換為大寫。

引數類型：

- STRING
- 傳回類型：STRING

範例：

```
os> source=people | eval `UPPER('helloworld')` = UPPER('helloworld'),
 `UPPER('HELLOWORLD')` = UPPER('HELLOWORLD') | fields `UPPER('helloworld')`,
 `UPPER('HELLOWORLD')`
fetched rows / total rows = 1/1
+----------------------+----------------------+
| UPPER('helloworld')   | UPPER('HELLOWORLD')   |
|----------------------+----------------------|
| HELLOWORLD            | HELLOWORLD            |
+----------------------+----------------------+
```

PPL 類型轉換函數

> **Note**
>
> 若要查看哪些 AWS 資料來源整合支援此 PPL 函數，請參閱 the section called “函數”。

`TRIM`

用量： 會將 `cast(expr as dateType)`轉換為 `expr` ，`dataType`並傳回 的值`dataType`。

適用下列轉換規則：

類型轉換規則

Src/Target	STRING	NUMBER	BOOLEAN	TIMESTAMP	DATE	TIME
STRING		Note1	Note1	TIMESTAMP()	DATE()	TIME()
NUMBER	Note1		v ! =0	N/A	N/A	N/A
BOOLEAN	Note1	v ? 1 : 0		N/A	N/A	N/A
TIMESTAMP	Note1	N/A	N/A		DATE()	TIME()

Src/Target	STRING	NUMBER	BOOLEAN	TIMESTAMP	DATE	TIME
DATE	Note1	N/A	N/A	N/A		N/A
TIME	Note1	N/A	N/A	N/A	N/A	

轉換為字串範例：

```
os> source=people | eval `cbool` = CAST(true as string), `cint` = CAST(1 as string),
 `cdate` = CAST(CAST('2012-08-07' as date) as string) | fields `cbool`, `cint`, `cdate`
fetched rows / total rows = 1/1
+---------+--------+------------+
| cbool   | cint   | cdate      |
|---------+--------+------------|
| true    | 1      | 2012-08-07 |
+---------+--------+------------+
```

轉換為數字範例：

```
os> source=people | eval `cbool` = CAST(true as int), `cstring` = CAST('1' as int) |
 fields `cbool`, `cstring`
fetched rows / total rows = 1/1
+---------+-----------+
| cbool   | cstring   |
|---------+-----------|
| 1       | 1         |
+---------+-----------+
```

迄今為止的範例：

```
os> source=people | eval `cdate` = CAST('2012-08-07' as date), `ctime` =
 CAST('01:01:01' as time), `ctimestamp` = CAST('2012-08-07 01:01:01' as timestamp) |
 fields `cdate`, `ctime`, `ctimestamp`
fetched rows / total rows = 1/1
+------------+----------+---------------------+
| cdate      | ctime    | ctimestamp          |
|------------+----------+---------------------|
| 2012-08-07 | 01:01:01 | 2012-08-07 01:01:01 |
+------------+----------+---------------------+
```

鏈結轉換範例：

```
os> source=people | eval `cbool` = CAST(CAST(true as string) as boolean) | fields
 `cbool`
fetched rows / total rows = 1/1
+---------+
| cbool   |
|---------|
| True    |
+---------+
```

監控 Amazon OpenSearch Service 網域

監控是維護 Amazon OpenSearch Service 和其他 AWS 解決方案可靠性、可用性和效能的重要部分。AWS 提供下列工具來監控您的 OpenSearch Service 資源、報告問題，並適時採取自動動作：

Amazon CloudWatch

Amazon CloudWatch 即時監控您的 OpenSearch Service 資源。您可以收集和追蹤指標、建立自訂儀表板，以及設定警示，在指標達到某個閾值時通知您或採取動作。如需更多資訊，請參閱 Amazon CloudWatch 使用者指南。

Amazon CloudWatch Logs

您可以使用 Amazon CloudWatch Logs 來監控、存放和存取您的 OpenSearch 日誌檔案。CloudWatch Logs 可監控日誌檔案中的資訊，並在滿足特定閾值時向您發出通知。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。

Amazon EventBridge

Amazon EventBridge 在 OpenSearch Service 網域中傳遞近乎即時的描述變更的系統事件串流。您可以建立監控特定事件的規則，並在發生這些事件 AWS 時觸發其他服務中的自動動作。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

AWS CloudTrail

AWS CloudTrail 會將對 OpenSearch Service 發出的組態 API 呼叫擷取為事件。它可將這些事件傳遞到您指定的 Amazon S3 儲存貯體。使用此資訊，可確認哪些使用者和帳戶提出請求、提出請求的來源 IP 地址以及提出請求的時間。如需詳細資訊，請參閱 AWS CloudTrail 使用者指南。

主題

- 使用 Amazon CloudWatch 監控 OpenSearch 叢集指標
- 使用 Amazon CloudWatch Logs 監控 OpenSearch 日誌
- 在 Amazon OpenSearch Service 中監控稽核日誌
- 使用 Amazon EventBridge 監控 OpenSearch Service 事件
- 使用 監控 Amazon OpenSearch Service API 呼叫 AWS CloudTrail

使用 Amazon CloudWatch 監控 OpenSearch 叢集指標

Amazon OpenSearch Service 將資料從您的網域發佈到 Amazon CloudWatch。CloudWatch 可讓您擷取這些資料點的相關統計數字作為一組排序時間序列資料，也就是指標。OpenSearch Service 會以 60 秒的間隔將大多數指標傳送至 CloudWatch。如果您使用一般用途或磁帶 EBS 磁碟區，EBS 磁碟區指標將僅每隔 5 分鐘更新一次。所有累積指標 （例如 `ThreadpoolWriteRejected`、`ThreadpoolSearchRejected`) 都是記憶體內，且將會失去狀態。指標會在節點捨棄、節點退信、節點取代和藍/綠部署期間重設。如需 Amazon CloudWatch 的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》。

OpenSearch Service 主控台會根據 CloudWatch 的原始資料顯示一系列圖形。根據需求，您可能偏好在 CloudWatch 中檢視叢集資料，而非在主控台中檢視圖形。此服務會封存指標兩週，之後才會捨棄它們。這些指標為免費提供，無須額外付費，但 CloudWatch 仍會針對建立儀表板和警示收費。如需詳細資訊，請參閱 Amazon CloudWatch 定價。

OpenSearch Service 會將以下指標發佈至 CloudWatch：

- the section called “叢集指標”
- the section called “專用主節點指標”
- the section called “EBS 磁碟區指標”
- the section called “執行個體指標”
- the section called “暖指標”
- the section called “專用協調器節點指標”
- the section called “冷儲存指標”
- the section called “提醒指標”
- the section called “異常偵測指標”
- the section called “非同步搜尋指標”
- the section called “SQL 指標”
- the section called “k-NN 指標”
- the section called “跨叢集搜尋指標”
- the section called “跨叢集複寫指標”
- the section called “Learning to Rank 指標”
- the section called “Piped Processing Language 指標”

在 CloudWatch 中檢視指標

首先依服務命名空間對 CloudWatch 指標分組，然後再依各命名空間內不同的維度組合分組。

使用 CloudWatch 主控台檢視指標

1. 在 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在左側導覽窗格中，尋找 Metrics (指標)，然後選擇 All metrics (所有指標)。選取 ES/OpenSearchService 命名空間。
3. 選擇維度以檢視對應指標。個別節點的指標位於 `ClientId, DomainName, NodeId` 維度中。叢集指標位於 `Per-Domain, Per-Client Metrics` 維度中。某些節點指標會在叢集層級彙總，因此包含在這兩個維度中。碎片指標位於 `ClientId, DomainName, NodeId, ShardRole` 維度中。

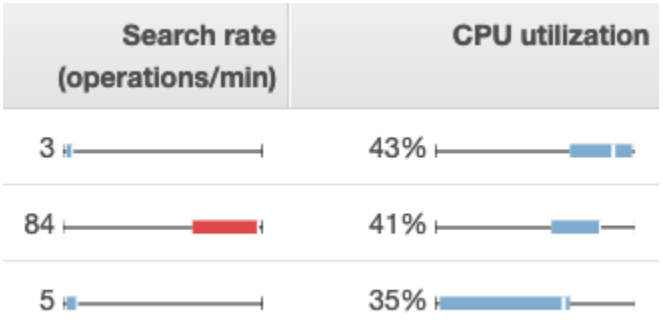
使用 檢視指標清單 AWS CLI

執行以下命令：

```
aws cloudwatch list-metrics --namespace "AWS/ES"
```

解讀 OpenSearch Service 中的運作狀態圖表

若要檢視 OpenSearch Service 中的指標，請使用 Cluster health (叢集運作狀態) 和 Instance health (執行個體運作狀態) 索引標籤。Instance health (執行個體運作狀態) 索引標籤使用方塊圖，可快速查看每個 OpenSearch 節點的運作狀態：



- 每個色彩方塊都能顯示該節點在整段指定時間的值範圍。
- 藍色方塊表示其值與其他節點相符。紅色方塊表示其值出現異常。
- 每個方塊中的白色線條則表示目前節點的值。
- 每個方塊在任何一邊上的「whiskers」則表示所有節點在整段時間的最小值與最大值。

如果您對網域進行組態變更，則 Cluster health (叢集運作狀態) 和 Instance health (執行個體運作狀態) 索引標籤上個別執行個體清單的大小通常會短暫出現加倍的情況，然後才恢復為正確的數字。如需此行為的說明，請參閱the section called “組態變更”。

叢集指標

Amazon OpenSearch Service 提供下列叢集指標。

指標	Description
ClusterStatus.green	1 的值表示將所有索引碎片分配至叢集中的節點。 相關統計資訊：Maximum
ClusterStatus.yellow	1 的值表示將所有索引的主要碎片分配給叢集中的節點，但用於至少一個索引的複寫碎片則不分配。如需詳細資訊，請參閱the section called “黃色叢集狀態”。 相關統計資訊：Maximum
ClusterStatus.red	1 的值表示至少一個索引的主要碎片和複寫碎片未分配至叢集中的節點。如需詳細資訊，請參閱the section called “紅色叢集狀態”。 相關統計資訊：Maximum
Shards.active	作用中主要碎片和複本碎片的總數。 相關統計數字：最大，總和
Shards.unassigned	未分配至叢集中節點的碎片數目。 相關統計數字：最大，總和
Shards.delayedUnassigned	其節點分配已被逾時設定延遲的碎片數量。 相關統計數字：最大，總和
Shards.activePrimary	活動中主要碎片的數量。 相關統計數字：最大，總和

指標	Description
Shards.initializing	正在初始化的碎片數量。 相關統計資料：總和
Shards.relocating	正在重新放置的碎片數量。 相關統計資料：總和
Nodes	OpenSearch Service 叢集中的節點數量，包括專用主節點和暖節點。如需詳細資訊，請參閱the section called “組態變更”。 相關統計資訊：Maximum
SearchableDocuments	叢集中跨所有資料節點的可搜尋文件的總數。 相關統計資料：下限、上限、平均數
DeletedDocuments	叢集中跨所有資料節點的標記進行刪除文件的總數。這些文件不再出現於搜尋結果中，但 OpenSearch 在區段合併時只會從磁碟移除已刪除的文件。此指標會在刪除請求後增加，在區段合併後降低。 相關統計資料：下限、上限、平均數
CPUUtilization	叢集中資料節點的 CPU 用量百分比。上限顯示具有最高 CPU 用量的節點。平均值代表叢集中的所有節點。此指標也適用於個別節點。 相關統計資訊：Maximum、Average

指標	Description
FreeStorageSpace	叢集中資料節點的可用空間。Sum 會顯示叢集的總可用空間，但您必須保留一分鐘的間隔，才能取得正確的值。Minimum 和 Maximum 分別會顯示具有最少和最多可用空間的節點。此指標也適用於個別節點。當此指標達到 0 時，OpenSearch Service 會擲出 ClusterBlockException 。若要恢復，您必須刪除索引、新增更大的執行個體，或為現有執行個體新增 EBS 式儲存。如需詳細資訊，請參閱 the section called “缺少可用儲存空間”。 OpenSearch Service 主控台會以 GiB 為單位顯示此值。Amazon CloudWatch 主控台會以 MiB 為單位顯示它。 **Note** FreeStorageSpace 將永遠低於 OpenSearch _cluster/stats 和 _cat/allocation API 提供的值。OpenSearch Service 會在每個執行個體上保留一定百分比的儲存空間供內部操作使用。如需詳細資訊，請參閱計算儲存需求。 相關統計資訊：Minimum、Maximum、Average、Sum
ClusterUsedSpace	已用於叢集的空間總數。您必須保留一分鐘的間隔，才能取得正確的數值。 OpenSearch Service 主控台會以 GiB 為單位顯示此值。Amazon CloudWatch 主控台會以 MiB 為單位顯示它。 相關統計資訊：Minimum、Maximum

指標	Description
ClusterIndexWritesBlocked	指示您的叢集是否要接受或封鎖外來的寫入請求。0 值表示叢集接受請求。1 值表示叢集封鎖請求。 常見的因素包括：FreeStorageSpace 過低或 JVMMemoryPressure 過高。若要減輕此問題，可考慮增加更多的磁碟空間或擴展您的叢集。 相關統計資訊：Maximum
JVMMemoryPressure	用於叢集中所有資料節點的 Java heap 的最大百分比。OpenSearch Service 針對 Java 堆積使用執行個體 RAM 的一半，堆積大小最多可達 32 GiB。您可以垂直擴展執行個體高達 64 GiB 的 RAM，屆時便能透過新增執行個體進行水平擴展。請參閱 the section called “建議 CloudWatch 警示”。 相關統計資訊：Maximum **Note** 此指標的邏輯在服務軟體 R20220323 中有所變更。如需詳細資訊，請參閱版本備註。
OldGenJVMMemoryPressure	用於叢集中所有資料節點的「舊一代」的 Java 堆積的最大百分比。此指標也適用於節點層級。 相關統計資訊：Maximum
AutomatedSnapshotFailure	叢集中失敗的自動快照數量。1 值表示此網域過去 36 小時未執行任何自動快照。 相關統計資訊：Minimum、Maximum

指標	Description
CPUCreditBalance	可供叢集中資料節點使用的剩餘 CPU 點數。一個 CPU 點數提供一分鐘、一個 CPU 核心的完整效能。如需詳細資訊，請參閱 Amazon EC2 開發人員指南中的 CPU 點數。此指標僅適用於 T2 執行個體類型。 相關統計資訊：Minimum
OpenSearchDashboardsHealthyNodes	OpenSearch Dashboards 的運作狀態檢查。如果最小值、最大值和平均值都等於 1，則 Dashboards 的行為正常。如果您有 10 個節點，其中最大為 1，最小為 0，平均為 0.7，這表示 7 個節點 (70%) 狀況良好，3 個節點 (30%) 狀況不良。 相關統計資料：下限、上限、平均數
OpensearchDashboardsReportingFailedRequestSysErrCount	因伺服器問題或功能限制而失敗的產生 OpenSearch Dashboards 報告的請求數。 相關統計資料：總和
OpensearchDashboardsReportingFailedRequestUserErrCount	因用戶端問題而失敗的產生 OpenSearch Dashboards 報告的請求數。 相關統計資料：總和
OpensearchDashboardsReportingRequestCount	產生 OpenSearch Dashboards 報告的請求總數。 相關統計資料：總和
OpensearchDashboardsReportingSuccessCount	產生 OpenSearch Dashboards 報告的成功請求數。 相關統計資料：總和
KMSKeyError	值 1 表示用來加密靜態資料的 AWS KMS 金鑰已停用。若要使網域恢復正常運作，請重新啟用此金鑰。主控台只會針對加密靜態資料的網域顯示此指標。 相關統計資訊：Minimum、Maximum

指標	Description
KMSKeyInaccessible	值 1 表示用來加密靜態資料的 AWS KMS 金鑰已刪除或撤銷其對 OpenSearch Service 的授予。您無法復原此狀態的網域。但是，如果您有手動快照，您可以用它來將網域的資料遷移至新的網域。主控台只會針對加密靜態資料的網域顯示此指標。 相關統計資訊：Minimum、Maximum
InvalidHostHeaderRequests	對 OpenSearch 叢集發出的 HTTP 請求數，請求中包括無效 (或遺漏) 主機標頭。有效的請求包括網域主機名稱作為主機標頭值。OpenSearch Service 會拒絕沒有限制性存取政策的公有存取網域的無效請求。我們建議將限制存取政策套用到所有網域。 如果您看見此指標有較大值，請確認 OpenSearch 用戶端在其請求中包含網域主機名稱 (而非其 IP 地址)。 相關統計資料：總和
OpenSearchRequests (previously ElasticsearchRequests)	對 OpenSearch 叢集發出的請求數。 相關統計資料：總和
2xx, 3xx, 4xx, 5xx	產生指定 HTTP 回應碼 (2xx、3xx、4xx、5xx) 的網域請求數。 相關統計資料：總和
ThroughputThrottle	指出磁碟是否已調節。當 ReadThroughputMicroBursting 和 的合併輸送量WriteThroughputMicroBursting 高於最大輸送量時，即會發生限流MaxProvisionedThroughput 。MaxProvisionedThroughput 是執行個體輸送量或佈建磁碟區輸送量的較低值。值 1 表示磁碟已調節。0 值表示正常行為。 如需執行個體輸送量的資訊，請參閱 Amazon EBS 最佳化執行個體。如需磁碟區輸送量的資訊，請參閱 Amazon EBS 磁碟區類型。 相關統計資訊：Minimum、Maximum

指標	Description
IopsThrottle	指出是否已調節網域上的每秒輸入/輸出操作數 (IOPS)。當資料節點的 IOPS 超過 EBS 磁碟區或資料節點的 EC2 執行個體允許的上限時，就會發生限流。 如需執行個體 IOPS 的資訊，請參閱 Amazon EBS 最佳化執行個體。如需磁碟區 IOPS 的資訊，請參閱 Amazon EBS 磁碟區類型。 相關統計資訊：Minimum、Maximum
HighSwapUsage	值為 1 表示由於頁面故障而交換可能會導致特定期間內基礎磁碟用量激增。 相關統計資訊：Maximum

專用主節點指標

Amazon OpenSearch Service 提供下列專用主節點指標。

指標	Description
MasterCPUUtilization	專用主節點使用的 CPU 資源的最大百分比。當這項指標達到 60% 時，建議提高執行個體類型的大小。 相關統計資訊：Maximum
MasterFreeStorageSpace	此指標無關，可忽略。此服務不使用主節點做為資料節點。
MasterJVMMemoryPressure	用於叢集中所有專用主節點的 Java heap 的最大百分比。當這項指標達到 85% 時，建議移至較大的執行個體類型。 相關統計資訊：Maximum

指標	Description
	Note 此指標的邏輯在服務軟體 R20220323 中有所變更。如需詳細資訊，請參閱版本備註。
MasterOldGenJVMMemoryPressure	用於每個主節點「舊一代」的 Java 堆積最大百分比。 相關統計資訊：Maximum
MasterCPUCreditBalance	可供叢集中專用主節點使用的剩餘 CPU 額度。一個 CPU 點數提供一分鐘、一個 CPU 核心的完整效能。如需詳細資訊，請參閱 Amazon EC2 開發人員指南中的 CPU 點數。此指標僅適用於 T2 執行個體類型。 相關統計資訊：Minimum
MasterReachableFromNode	MasterNotDiscovered 例外狀況的運作狀態檢查。1 值表示正常行為。0 值表示 /_cluster/health/ 失敗。 失敗表示無法從來源節點連線主節點。通常是網路連線問題或 AWS 相依性問題的結果。 相關統計資訊：Maximum
MasterSysMemoryUtilization	已使用主節點記憶體的百分比。 相關統計資訊：Maximum

專用協調器節點指標

Amazon OpenSearch Service 為專用協調器節點提供下列指標。

指標	Description
CoordinatorCPUUtilization	專用協調器節點使用的 CPU 資源百分比上限。當此指標達到 80% 時，我們建議增加執行個體類型的大小。

指標	Description
	相關統計資訊：Maximum
CoordinatorJVMMemoryPressure	叢集中所有專用協調器節點所使用的 Java 堆積百分比上限。當這項指標達到 85% 時，建議移至較大的執行個體類型。 相關統計資訊：Maximum
CoordinatorOldGenJVMMemoryPressure	用於每個主節點「舊一代」的 Java 堆積最大百分比。 相關統計資訊：Maximum
CoordinatorSysMemoryUtilization	使用中的協調器節點記憶體百分比。 相關統計資訊：Maximum
CoordinatorFreeStorageSpace	此指標表示服務不使用協調器節點做為資料節點。

EBS 磁碟區指標

Amazon OpenSearch Service 提供下列 EBS 磁碟區指標。

指標	Description
ReadLatency	EBS 磁碟區讀取操作的延遲 (以秒為單位)。此指標也適用於個別節點。 相關統計資料：下限、上限、平均數
WriteLatency	EBS 磁碟區寫入操作的延遲 (以秒為單位)。此指標也適用於個別節點。 相關統計資料：下限、上限、平均數
ReadThroughput	EBS 磁碟區讀取操作的傳輸量 (以位元組/秒為單位)。此指標也適用於個別節點。 相關統計資料：下限、上限、平均數

指標	Description
ReadThroughputMicroBursting	考量微爆量時，EBS 磁碟區讀取操作的輸送量，以每秒位元組為單位。此指標也適用於個別節點。當 EBS 磁碟區在明顯較短的時間內（不到一分鐘）爆量高 IOPS 或輸送量時，會發生微爆。 相關統計資料：下限、上限、平均數
WriteThroughput	EBS 磁碟區寫入操作的傳輸量 (以位元組/秒為單位)。此指標也適用於個別節點。 相關統計資料：下限、上限、平均數
WriteThroughputMicroBursting	考量微爆量時，EBS 磁碟區寫入操作的輸送量，以每秒位元組為單位。此指標也適用於個別節點。當 EBS 磁碟區在明顯較短的時間內（不到一分鐘）爆量高 IOPS 或輸送量時，會發生微爆。 相關統計資料：下限、上限、平均數
DiskQueueDepth	等待中的 EBS 磁碟區輸入與輸出 (I/O) 請求數。 相關統計資料：下限、上限、平均數
ReadIOPS	EBS 磁碟區讀取操作的每秒輸入與輸出 (I/O) 操作數。此指標也適用於個別節點。 相關統計資料：下限、上限、平均數
ReadIOPSMicroBursting	考量微爆量時，EBS 磁碟區上讀取操作的每秒輸入和輸出 (I/O) 操作數。此指標也適用於個別節點。當 EBS 磁碟區在明顯較短的時間內（不到一分鐘）爆量高 IOPS 或輸送量時，會發生微爆。 相關統計資料：下限、上限、平均數
WriteIOPS	EBS 磁碟區寫入操作的每秒輸入與輸出 (I/O) 操作數。此指標也適用於個別節點。 相關統計資料：下限、上限、平均數

指標	Description
WriteIOPS MicroBursting	考量微爆量時，EBS 磁碟區上每秒寫入操作的輸入和輸出 (I/O) 操作數目。此指標也適用於個別節點。當 EBS 磁碟區在明顯較短的時間內（不到一分鐘）爆量高 IOPS 或輸送量時，會發生微爆。 相關統計資料：下限、上限、平均數
BurstBalance	EBS 磁碟區的爆量儲存貯體中剩餘的輸入與輸出 (I/O) 點數百分比。值 100 表示磁碟區已累積至最大點數。如果此百分比低於 70%，請參閱 the section called “低 EBS 爆量餘額”。對於具有 gp3 磁碟區類型的網域，以及具有磁碟區大小高於 1000 GiB 之 gp2 磁碟區的網域，爆量餘額會保持在 0。 相關統計資料：下限、上限、平均數
VolumeStalledIOcheck	EBS 磁碟區的狀態，以判斷何時受損。指標是二進位值，根據 EBS 磁碟區是否可以完成輸入和輸出操作，傳回 0（通過）或 1（失敗）狀態。VolumeStalledIOcheck 也可用於個別節點。 相關統計資料：下限、上限、平均數

執行個體指標

Amazon OpenSearch Service 提供網域中每個執行個體的下列指標。OpenSearch Service 也彙總這些執行個體指標，讓您深入了解整體叢集運作狀態。您可以在主控台使用 Sample Count (取樣計數) 統計數字來驗證此行為。請注意，下表中每個指標有節點和叢集的相關統計資料。

> **Important**
>
> 在處理 _index API 的呼叫時，不同版本的 Elasticsearch 會使用不同的執行緒集區。Elasticsearch 1.5 和 2.3 版會使用索引執行緒集區。Elasticsearch 5.x、6.0 和 6.2 版會使用大量執行緒集區。OpenSearch 和 Elasticsearch 6.3 及較新版本會使用寫入執行緒集區。目前，OpenSearch Service 主控台不包含大量執行緒集區的圖形。
> 使用 GET _cluster/settings?include_defaults=true 來檢查叢集的執行緒集區和佇列大小。

指標	Description
FetchLatency	節點中所有碎片擷取操作在分鐘 N 和分鐘 (N - 1) 之間的總時間差異，以毫秒為單位。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限
FetchRate	資料節點上所有碎片每分鐘的碎片擷取操作總數。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限、總和
ScrollTotal	資料節點上所有碎片的每分鐘碎片捲動操作總數。 相關節點統計資料：平均值、最大值 相關叢集統計資訊：平均數、上限、總和
ScrollCurrent	目前正在執行的碎片捲動操作數目。 相關節點統計資料：平均值、最大值 相關叢集統計資訊：平均數、上限、總和
OpenContexts	開放搜尋內容的數量。 相關節點統計資料：平均值、最大值 相關叢集統計資訊：平均數、上限、總和
ThreadCount	OpenSearch 程序目前正在使用的執行緒總數。 相關節點統計資料：平均值、最大值 相關叢集統計資訊：平均數、上限、總和
ShardReactivateCount	從閒置狀態啟動所有碎片的總次數。 相關節點統計資料：總和、最大值

指標	Description
	相關叢集統計資料：總和、最大值
ConcurrentSearchRate	資料節點上所有碎片每分鐘使用並行區段搜尋的搜尋請求總數。對於 _search API 發出的單一呼叫，可能會傳回來自多個不同碎片的結果。如果上述碎片當中有五個是在同一個節點上，則該節點會回報這個指標為 5，即使該用戶端僅發出一次請求。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限、總和
ConcurrentSearchLatency	在分鐘 N 和分鐘 (N-1) 之間的節點中使用並行區段搜尋的所有搜尋所取得的總時間差異，以毫秒為單位。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限
IndexingLatency	節點中所有索引操作在分鐘 N 和分鐘 (N-1) 之間的總時間差異，以毫秒為單位。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限
IndexingRate	每分鐘進行的索引操作次數。對於 _bulk API 發出的單一呼叫，即新增兩份文件並更新兩份文件，此計為可能分散於一個或更多個節點進行的四次操作。如果該索引有一或多個複本，且位於沒有最佳化執行個體的 OpenSearch 網域上，則叢集中的其他節點也會記錄總共四個索引操作。對於具有最佳化執行個體的 OpenSearch 網域，具有複本的其他節點不會記錄任何操作。文件刪除不列入此指標。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限、總和

指標	Description
SearchLatency	節點中所有搜尋在分鐘 N 和分鐘 (N-1) 之間的總時間差異，以毫秒為單位。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限
SearchRate	資料節點上每分鐘對所有碎片發出搜尋請求的總次數。對於 _search API 發出的單一呼叫，可能會傳回來自多個不同碎片的結果。如果上述碎片當中有五個是在同一個節點上，則該節點會回報這個指標為 5，即使該用戶端僅發出一次請求。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限、總和
SegmentCount	資料節點上的區段數。您擁有的區段越多，每個搜尋所需的時間越長。OpenSearch 偶爾會將較小的區段合併成較大的區段。 相關節點統計數字：最大，平均 相關叢集統計資訊：總和、上限、平均數
SysMemoryUtilization	已使用執行個體記憶體的百分比。此測量結果的高值是正常的，通常不代表叢集的問題。如需有關潛在效能和穩定性問題的更佳指標，請參閱 JVMMemoryPressure 指標。 相關節點統計資訊：下限、上限、平均數 相關叢集統計資訊：下限、上限、平均數
JVMGCYoungCollectionCount	「新一代」廢棄項目收集的已執行次數。大量、持續擴增的執行次數是叢集操作的正常情況。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數

指標	Description
JVMGCYoungCollectionTime	叢集已執行「新一代」廢棄項目收集的時間 (單位為毫秒)。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
JVMGCOldCollectionCount	「舊一代」廢棄項目收集的已執行次數。在資源充足的叢集中，這個數字應該很小，而且不常擴增。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
JVMGCOldCollectionTime	叢集已花在執行「舊一代」廢棄項目收集的時間 (單位為毫秒)。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
OpenSearchDashboardsConcurrentConnections	OpenSearch Dashboards 的作用中並行連線的數目。如果此數值持續增加，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
OpenSearchDashboardsHealthyNode	個別 OpenSearch Dashboards 節點的運作狀態檢查。1 值表示正常行為。0 值表示無法存取 Dashboards。 相關節點統計數字：最小 相關叢集統計資訊：下限、上限、平均數
OpenSearchDashboardsHeapTotal	分配給 OpenSearch Dashboards 的堆積記憶體容量 (單位為 MiB)。不同的 EC2 執行個體類型可能會影響精確的記憶體分配。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數

指標	Description
OpenSearchDashboardsHeapUsed	OpenSearch Dashboards 使用的堆積記憶體的絕對容量 (單位為 MiB)。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
OpenSearchDashboardsHeapUtilization	OpenSearch Dashboards 使用的可用堆積記憶體的最大百分比。如果此值超過 80%，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：下限、上限、平均數
OpenSearchDashboardsOS1MinuteLoad	OpenSearch Dashboards 的一分鐘平均 CPU 負載。CPU 負載理想情況下應該保持在 1.00 以下。雖然暫時峰值沒問題，但如果此指標一致高於 1.00，建議您增加執行個體類型的大小。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限
OpenSearchDashboardsRequestTotal	對 OpenSearch Dashboards 發出的 HTTP 請求總數。如果您的系統速度緩慢或您看到大量 Dashboards 請求，請考慮增加執行個體類型的大小。 相關節點統計數字：總和 相關叢集統計資訊：總和
OpenSearchDashboardsResponseTimesMaxInMillis	OpenSearch Dashboards 回應請求所需的時間上限，單位為毫秒。如果請求持續需要很長的時間才能傳回結果，請考慮增加執行個體類型的大小。 相關節點統計資訊：上限 相關叢集統計數字：最大，平均

指標	Description
SearchTaskCancelled	取消協調器節點的數量。 相關節點統計數字：總和 相關叢集統計資訊：總和
SearchShardTaskCancelled	資料節點取消的數量。 相關節點統計數字：總和 相關叢集統計資料：總和、
ThreadpoolForce_mergeQueue	強制合併執行緒集區中的已排入佇列任務數量。如果佇列大小持續高居不下，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
ThreadpoolForce_mergeRejected	強制合併執行緒集區中的已拒絕任務數量。如果這個數量持續增加，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和
ThreadpoolForce_mergeThreads	強制合併執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
ThreadpoolIndexQueue	索引執行緒集區中的已排入佇列任務數量。如果佇列大小持續高居不下，請考慮擴展您的叢集。索引佇列的大小上限為 200。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數

指標	Description
ThreadpoolIndexRejected	索引執行緒集區中的已拒絕任務數量。如果這個數量持續增加，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和
ThreadpoolIndexThreads	索引執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
ThreadpoolSearchQueue	搜尋執行緒集區中的已排入佇列任務數量。如果佇列大小持續高居不下，請考慮擴展您的叢集。搜尋佇列的大小上限為 1,000。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
ThreadpoolSearchRejected	搜尋執行緒集區中的已拒絕任務數量。如果這個數量持續增加，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和
ThreadpoolSearchThreads	搜尋執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
Threadpoolsql-workerQueue	SQL 搜尋執行緒集區中的已排入佇列的任務數量。如果佇列大小持續高居不下，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數

指標	Description
Threadpoolsql-workerRejected	SQL 搜尋執行緒集區中的已拒絕的任務數量。如果這個數量持續增加，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和
Threadpoolsql-workerThreads	SQL 搜尋執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
ThreadpoolBulkQueue	大量執行緒集區中的已排入佇列任務數量。如果佇列大小持續高居不下，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
ThreadpoolBulkRejected	大量執行緒集區中的已拒絕任務數量。如果這個數量持續增加，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和
ThreadpoolBulkThreads	大量執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
ThreadpoolIndexSearcherQueue	索引搜尋器執行緒集區中的佇列任務數目。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數

指標	Description
ThreadpoolIndexSearcherRejected	索引搜尋器執行緒集區中遭拒的任務數量。 相關節點統計資訊：上限 相關叢集統計資訊：總和
ThreadpoolIndexSearcherThreads	索引搜尋器執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
ThreadpoolWriteThreads	寫入執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
ThreadpoolWriteQueue	寫入執行緒集區中的已排入佇列任務數量。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
ThreadpoolWriteRejected	寫入執行緒集區中的已拒絕任務數量。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和 **Note** 由於 7.1 版中的預設寫入佇列大小從 200 增加到 10000，這個指標不再是 OpenSearch Service 拒絕的唯一指標。使用 CoordinatingWriteRejected 、PrimaryWriteRejected 和 ReplicaWriteRejected 指標來監控 7.1 版及更新版本中的拒絕。

指標	Description
CoordinatingWriteRejected	自上次 OpenSearch Service 處理程序啟動後，協調節點上由於索引壓力而發生的拒絕總數。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和 在 7.1 版及更高版本中可使用此指標。
PrimaryWriteRejected	自上次 OpenSearch Service 處理程序啟動後，主碎片上由於索引壓力而發生的拒絕總數。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和 在 7.1 版及更高版本中可使用此指標。
ReplicaWriteRejected	自上次 OpenSearch Service 處理程序啟動後，複本碎片上由於索引壓力而發生的拒絕總數。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和 在 7.1 版及更高版本中可使用此指標。
WorkloadManagementEnabled	指出工作負載管理功能是否已啟用。值 1 表示已啟用，值 0 表示其 monitor_only 或 已停用。 相關節點統計資料：最大值、最小值 相關叢集統計資訊：平均數、總和 在 7.1 版及更高版本中可使用此指標。

指標	Description
SoftQueryGroupCount	網域中軟模式下的查詢群組數量。 相關節點統計資料：平均值、最大值 相關叢集統計資訊：平均數、上限、總和 在 7.1 版及更高版本中可使用此指標。
EnforcedQueryGroupCount	網域中強制執行模式下的查詢群組數目。 相關節點統計資料：平均值、最大值 相關叢集統計資訊：平均數、上限、總和 在 7.1 版及更高版本中可使用此指標。

暖指標

Amazon OpenSearch Service 為多層儲存架構和 UltraWarm 提供下列指標

> **Note**
> 暖索引相關指標僅適用於多層儲存架構

指標	Description
WarmIndexingLatency	在分鐘 N 和分鐘 (N-1) 之間的暖節點中，所有索引操作所取得的總時間差異，以毫秒為單位。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限
WarmIndexingRate	每分鐘暖值的索引操作數目。對於 _bulk API 發出的單一呼叫，即新增兩份文件並更新兩份文件，此計為可能分散於一個或更多個節點進行的四次操作。如果該索引有一或多個複本，且位於沒有最佳化執行個體的

指標	Description
	OpenSearch 網域上，則叢集中的其他節點也會記錄總共四個索引操作。對於具有最佳化執行個體的 OpenSearch 網域，具有複本的其他節點不會記錄任何操作。文件刪除不列入此指標。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限、總和
WarmThreadpoolIndexingQueue	索引執行緒集區中的已排入佇列任務數量。如果佇列大小持續高居不下，請考慮擴展您的叢集。索引佇列的大小上限為 200。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、上限、總和
WarmThreadpoolIndexingRejected	索引執行緒集區中的已拒絕任務數量。如果這個數量持續增加，請考慮擴展您的叢集。 相關節點統計資訊：上限 相關叢集統計資訊：總和
WarmThreadpoolIndexingThreads	索引執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
WarmCPUUtilization	叢集中暖節點的 CPU 用量百分比。上限顯示具有最高 CPU 用量的節點。平均值代表叢集中的所有暖節點。此指標也適用於個別暖節點。 相關統計資訊：Maximum、Average
WarmFreeStorageSpace	可用暖儲存空間量 (以 MiB 為單位)。由於 Warm 使用 Amazon S3 而非連接的磁碟，因此 Sum 是唯一相關的統計資料。您必須保留一分鐘的間隔，才能取得正確的數值。 相關統計資料：總和

指標	Description
WarmSearchableDocuments	叢集中跨所有暖索引的可搜尋文件的總數。您必須保留一分鐘的間隔，才能取得正確的數值。 相關統計資料：總和
WarmSearchLatency	所有搜尋在分鐘 N 到分鐘 (N-1) 之間的暖機時間總時間差異，以毫秒為單位。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限
WarmSearchRate	暖節點上所有碎片的每分鐘搜尋請求總數。對於 _search API 發出的單一呼叫，可能會傳回來自多個不同碎片的結果。如果上述碎片當中有五個是在同一個節點上，則該節點會回報這個指標為 5，即使該用戶端僅發出一次請求。 相關節點統計資訊：平均數 相關叢集統計資訊：平均數、上限、總和
WarmStorageSpaceUtilization	叢集所使用暖儲存空間的總量 (單位為 MiB)。 相關統計資訊：Maximum
HotStorageSpaceUtilization	叢集所使用熱儲存空間的總量。 相關統計資訊：Maximum
WarmSysMemoryUtilization	已使用溫節點記憶體的百分比。 相關統計資訊：Maximum
HotToWarmMigrationQueueSize	目前等待從熱儲存遷移至暖儲存的索引數目。 相關統計資訊：Maximum

指標	Description
WarmToHot Migration QueueSize	目前等待從暖儲存遷移至熱儲存的索引數目。 相關統計資訊：Maximum
HotToWarm Migration FailureCount	從熱儲存遷移至暖儲存的失敗總數。此指標僅適用於 UltraWarm 節點。 相關統計資料：總和
HotToWarm Migration ForceMerg eLatency	遷移程序的強制合併階段的平均延遲。如果此階段持續需要太長的時間，請考慮增加。此指標僅適用於 UltraWarm 節點。index.ult rawarm.migration.force_merge.max_num_segments 。 相關統計資訊：平均數
HotToWarm Migration SnapshotL atency	遷移程序的快照階段的平均延遲。如果此階段持續需要太長時間，請確保您的碎片適當調整大小並分佈在整個叢集中。此指標僅適用於 UltraWarm 節點。 相關統計資訊：平均數
HotToWarm Migration Processin gLatency	從熱儲存成功遷移到暖儲存的平均延遲，不包括佇列中花費的時間。此值是完成遷移程序的強制合併、快照和碎片重新放置階段所需的時間總和。此指標僅適用於 UltraWarm 節點。 相關統計資訊：平均數
HotToWarm Migration SuccessCount	從熱儲存遷移至暖儲存的成功總數。 相關統計資料：總和
HotToWarm Migration SuccessLatency	從熱儲存成功遷移到暖儲存的平均延遲，包括佇列中花費的時間。 相關統計資訊：平均數

指標	Description
WarmThreadpoolSearchThreads	暖搜尋執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資訊：平均數、總和
WarmThreadpoolSearchRejected	暖搜尋執行緒集區中遭拒的任務數量。如果此數字持續增加，請考慮新增更多暖節點。 相關節點統計資訊：上限 相關叢集統計資訊：總和
WarmThreadpoolSearchQueue	暖搜尋執行緒集區中的佇列任務數目。如果佇列大小持續很高，請考慮新增更多暖節點。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
WarmJVMMemoryPressure	用於暖節點的 Java 堆積百分比上限。 相關統計資訊：Maximum **Note** 此指標的邏輯在服務軟體 R20220323 中有所變更。如需詳細資訊，請參閱版本備註。
WarmOldGenJVMMemoryPressure	每個暖節點用於「舊世代」的 Java 堆積百分比上限。 相關統計資訊：Maximum

指標	Description
WarmJVMGCYoungCollectionCount	"young generation" 垃圾回收在暖節點上執行的次數。大量、持續擴增的執行次數是叢集操作的正常情況。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
WarmJVMGCYoungCollectionTime	叢集在暖節點上執行「年輕世代」垃圾回收所花費的時間，以毫秒為單位。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
WarmJVMGCOldCollectionCount	「上一代」垃圾回收在暖節點上執行的次數。在資源充足的叢集中，這個數字應該很小，而且不常擴增。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
WarmConcurrentSearchRate	暖節點上所有碎片使用每分鐘並行區段搜尋的搜尋請求總數。對於 _search API 發出的單一呼叫，可能會傳回來自多個不同碎片的結果。如果上述碎片當中有五個是在同一個節點上，則該節點會回報這個指標為 5，即使該用戶端僅發出一次請求。 相關節點統計資訊：平均數 相關叢集統計資訊：總和、上限、平均數
WarmConcurrentSearchLatency	所有搜尋在分鐘 N 和分鐘 (N-1) 之間的暖節點中使用並行區段搜尋所取得的總時間差異，以毫秒為單位。 相關節點統計資訊：平均數 相關叢集統計數字：最大，平均

指標	Description
WarmThreadpoolIndexSearcherQueue	暖索引搜尋器執行緒集區中的佇列任務數目。 相關節點統計資訊：上限 相關叢集統計資訊：總和、上限、平均數
WarmThreadpoolIndexSearcherRejected	暖索引搜尋器執行緒集區中遭拒的任務數量。 相關節點統計資訊：上限 相關叢集統計資訊：總和
WarmThreadpoolIndexSearcherThreads	暖索引搜尋器執行緒集區的大小。 相關節點統計資訊：上限 相關叢集統計資料：總和、平均

冷儲存指標

Amazon OpenSearch Service 提供下列冷儲存指標。

指標	Description
ColdStorageSpaceUtilization	叢集所使用冷儲存空間的總量 (單位為 MiB)。 相關統計資料：上限
ColdToWarmMigrationFailureCount	從冷儲存遷移至暖儲存的失敗總數。 相關統計資料：總和
ColdToWarmMigrationLatency	成功完成從冷儲存遷移至暖儲存所需的時間量。 相關統計資訊：平均數
ColdToWarmMigrationQueueSize	目前等待從冷儲存遷移至暖儲存的索引數目。 相關統計資訊：Maximum

指標	Description
ColdToWarmMigrationSuccessCount	從冷儲存遷移至暖儲存的成功總數。 相關統計資料：總和
WarmToColdMigrationFailureCount	從暖儲存遷移至冷儲存的失敗總數。 相關統計資料：總和
WarmToColdMigrationLatency	成功完成從暖儲存遷移至冷儲存所需的時間量。 相關統計資訊：平均數
WarmToColdMigrationQueueSize	目前等待從暖儲存遷移至冷儲存的索引數目。 相關統計資訊：Maximum
WarmToColdMigrationSuccessCount	從暖儲存遷移至冷儲存的成功總數。 相關統計資料：總和

OpenSearch Optimized Instances (OR1) 指標

Amazon OpenSearch Service 為 OR1 執行個體提供下列指標。

指標	Description
RemoteStorageUsedSpace	叢集正在使用的 Amazon S3 空間總量，以 MiB 為單位。 相關統計資料：總和
RemoteStorageWriteRejected	主要碎片因遠端儲存和複寫壓力而拒絕的請求總數。這是從上次 OpenSearch Service 程序啟動開始計算的。 相關統計資料：總和
ReplicationLagMaxTime	複本碎片落後於主要碎片的時間量，以毫秒為單位。 相關統計資訊：Maximum

提醒指標

Amazon OpenSearch Service 提供下列提醒指標。

指標	Description
AlertingDegraded	值為 1 表示提醒索引為紅色，或是有一或多個節點不在排程上。0 值表示正常行為。 相關統計資訊：Maximum
AlertingIndexExists	值為 1 表示存在 .opensearch-alerting-config 索引。值為 0 則表示不存在。直到您第一次使用提醒功能為止，這個值都會維持在 0。 相關統計資訊：Maximum
AlertingIndexStatus.green	索引的運作狀態。值為 1 表示綠色。值為 0 表示索引不存在，或是並非綠色。 相關統計資訊：Maximum
AlertingIndexStatus.red	索引的運作狀態。值為 1 表示紅色。值為 0 表示索引不存在，或是並非紅色。 相關統計資訊：Maximum
AlertingIndexStatus.yellow	索引的運作狀態。值為 1 表示黃色。值為 0 表示索引不存在，或是並非黃色。 相關統計資訊：Maximum
AlertingNodesNotOnSchedule	值為 1 表示有些任務並未依照排程執行。值為 0 則表示所有提醒任務都正在依照排程執行 (或是沒有提醒任務)。檢查 OpenSearch Service 主控台或提出 _nodes/stats 請求，以查看是否有任何節點顯示高資源使用量。 相關統計資訊：Maximum
AlertingNodesOnSchedule	值為 1 則表示所有提醒任務都正在依照排程執行 (或是沒有提醒任務)。值為 0 表示有些任務並未依照排程執行。

指標	Description
	相關統計資訊：Maximum
AlertingScheduledJobEnabled	值為 1 表示 opensearch.scheduled_jobs.enabled 叢集設定為 true。值為 0 表示為 false，且已停用排程任務。 相關統計資訊：Maximum

異常偵測指標

Amazon OpenSearch Service 提供下列異常偵測指標。

指標	Description
ADPluginUnhealthy	值 1 表示異常偵測外掛程式無法正常運作，原因是大量失敗或它使用的其中一個索引是紅色。值 0 表示外掛程式如預期般運作。 相關統計資訊：Maximum
ADExecuteRequestCount	偵測異常的請求數。 相關統計資料：總和
ADExecuteFailureCount	偵測異常的失敗請求數。 相關統計資料：總和
ADHCExecuteFailureCount	偵測高基數偵測器異常的失敗請求數。 相關統計資料：總和
ADHCExecuteRequestCount	偵測高基數偵測器異常的請求數。 相關統計資料：總和
ADAnomalyResultsIndexStatusIndexExists	值為 1 表示 .opensearch-anomaly-results 別名指向的索引存在。在您第一次使用異常偵測之前，此值會保持為 0。 相關統計資訊：Maximum

指標	Description
ADAnomaly ResultsIn dexStatus.red	值為 1 表示 .opensearch-anomaly-results 別名指向的索引是紅色。值為 0 則表示它不是紅色。在您第一次使用異常偵測之前，此值會保持為 0。 相關統計資訊：Maximum
ADAnomaly Detectors IndexStat usIndexExists	值為 1 表示 .opensearch-anomaly-detectors 索引存在。值為 0 則表示不存在。在您第一次使用異常偵測之前，此值會保持為 0。 相關統計資訊：Maximum
ADAnomaly Detectors IndexStat us.red	值為 1 表示 .opensearch-anomaly-detectors 索引是紅色。值為 0 則表示它不是紅色。在您第一次使用異常偵測之前，此值會保持為 0。 相關統計資訊：Maximum
ADModelsC heckpoint IndexStat usIndexExists	值為 1 表示 .opensearch-anomaly-checkpoints 索引存在。值為 0 則表示不存在。在您第一次使用異常偵測之前，此值會保持為 0。 相關統計資訊：Maximum
ADModelsC heckpoint IndexStat us.red	值為 1 表示 .opensearch-anomaly-checkpoints 索引是紅色。值為 0 則表示它不是紅色。在您第一次使用異常偵測之前，此值會保持為 0。 相關統計資訊：Maximum

非同步搜尋指標

Amazon OpenSearch Service 提供下列非同步搜尋指標。

非同步搜尋協調器節點統計數字 (每個協調器節點)

指標	Description
AsynchronousSearchSubmissionRate	過去一分鐘內提交的非同步搜尋數量。
AsynchronousSearchInitializedRate	過去一分鐘內初始化的非同步搜尋數量。
AsynchronousSearchRunningCurrent	目前正在執行的非同步搜尋數量。
AsynchronousSearchCompletionRate	過去一分鐘內成功完成的非同步搜尋數量。
AsynchronousSearchFailureRate	過去一分鐘內完成和失敗的非同步搜尋數量。
AsynchronousSearchPersistRate	過去一分鐘內持續的非同步搜尋數量。
AsynchronousSearchPersistFailedRate	過去一分鐘內無法持續的非同步搜尋數量。
AsynchronousSearchRejected	自節點啟動時間以來拒絕的非同步搜尋總數。

指標	Description
AsynchronousSearchCancelled	自節點啟動時間以來已取消的非同步搜尋總數。
AsynchronousSearchMaxRunningTime	在最後一分鐘的節點上執行非同步搜尋的最長持續時間。

非同步搜尋叢集統計數字

指標	Description
AsynchronousSearchStoreHealth	最後一分鐘內持續索引中的存放運作狀態 (紅色/非紅色)。
AsynchronousSearchStoreSize	過去一分鐘內所有碎片的系統索引大小。
AsynchronousSearchStoredResponseCount	過去一分鐘內系統索引中存放的回應數量。

自動調整指標

Amazon OpenSearch Service 為 Auto-Tune 提供下列指標。

指標	Description
AutoTuneChangesHistoryHeapSize	MiB 中堆積大小調校值的變更歷史記錄。

指標	Description
AutoTuneChangesHistoryJVMYoungGenArgs	JVM YongGen 引數的變更歷史記錄。
AutoTuneFailed	指出自動調整變更是否失敗的布林值。
AutoTuneSucceeded	布林值，指出自動調整變更是否成功。
AutoTuneValue	佇列變更歷史記錄（計數）和快取會調校變更歷史記錄（以 MiB 為單位）是否有不中斷的變更。

具有待命指標的異地同步備份

Amazon OpenSearch Service 為具有待命的異地同步備份提供下列指標。

作用中可用區域中資料節點的節點層級指標

指標	Description
CPUUtilization	叢集中資料節點的 CPU 用量百分比。上限顯示具有最高 CPU 用量的節點。平均值代表叢集中的所有節點。此指標也適用於個別節點。
FreeStorageSpace	叢集中資料節點的可用空間。Sum 會顯示叢集的總可用空間，但您必須保留一分鐘的間隔，才能取得正確的值。Minimum 和 Maximum 分別會顯示具有最少和最多可用空間的節點。此指標也適用於個別節點。當此指標達到 0 時，OpenSearch Service 會擲出 ClusterBlockException 。若要恢復，您必須刪除索引、新增更大的執行個體，或為現有執行個體新增 EBS 式儲存。如需詳細資訊，請參閱 the section called “缺少可用儲存空間”。 OpenSearch Service 主控台會以 GiB 為單位顯示此值。Amazon CloudWatch 主控台會以 MiB 為單位顯示它。

指標	Description
JVMMemory Pressure	用於叢集中所有資料節點的 Java heap 的最大百分比。OpenSearch Service 針對 Java 堆積使用執行個體 RAM 的一半，堆積大小最多可達 32 GiB。您可以垂直擴展執行個體高達 64 GiB 的 RAM，屆時便能透過新增執行個體進行水平擴展。請參閱 the section called “建議 CloudWatch 警示”。
SysMemory Utilization	已使用執行個體記憶體的百分比。此測量結果的高值是正常的，通常不代表叢集的問題。如需有關潛在效能和穩定性問題的更佳指標，請參閱 JVMMemoryPressure 指標。
IndexingLatency	節點中所有索引操作在分鐘 N 和分鐘 (N-1) 之間的總時間差異，以毫秒為單位。
IndexingRate	每分鐘進行的索引操作次數。
SearchLatency	節點中所有搜尋在分鐘 N 和分鐘 (N-1) 之間的總時間差異，以毫秒為單位。
SearchRate	資料節點上每分鐘對所有碎片發出搜尋請求的總次數。
ThreadpoolSearchQueue	搜尋執行緒集區中的已排入佇列任務數量。如果佇列大小持續高居不下，請考慮擴展您的叢集。搜尋佇列的大小上限為 1,000。
ThreadpoolWriteQueue	寫入執行緒集區中的已排入佇列任務數量。
ThreadpoolSearchRejected	搜尋執行緒集區中的已拒絕任務數量。如果這個數量持續增加，請考慮擴展您的叢集。
ThreadpoolWriteRejected	寫入執行緒集區中的已拒絕任務數量。

作用中可用區域中叢集的叢集層級指標

指標	Description
DataNodes	作用中和待命碎片的總數。
DataNodes Shards.active	作用中主要碎片和複本碎片的總數。
DataNodes Shards.unassigned	未分配至叢集中節點的碎片數目。
DataNodes Shards.initializing	正在初始化的碎片數量。
DataNodes Shards.relocating	正在重新放置的碎片數量。

可用區域輪換指標

如果為 ActiveReads.*Availability-Zone* = 1，則區域為作用中。如果為 ActiveReads.*Availability-Zone* = 0，則區域處於待命狀態。

時間點指標

Amazon OpenSearch Service 為時間點 (PIT) 搜尋提供下列指標。

PIT 協調器節點統計資料 （每個協調器節點）

指標	Description
CurrentPointInTime	節點中作用中 PIT 搜尋內容的數量。
TotalPointInTime	節點啟動時間之後過期的 PIT 搜尋內容數目。

指標	Description
AvgPointInTimeAliveTime	從節點啟動時間開始，PIT 搜尋內容的平均持續作用。
HasActivePointInTime	值 1 表示自節點啟動時間以來，節點上有作用中的 PIT 內容。值為 0 表示沒有。
HasUsedPointInTime	值 1 表示節點上自節點啟動時間以來有過期的 PIT 內容。值為 0 表示沒有。

SQL 指標

Amazon OpenSearch Service 提供下列 SQL 支援指標。

指標	Description
SQLFailedRequestCountByCusErr	因用戶端問題而失敗的 _sql API 請求數。例如，請求可能會因為 IndexNotFoundException 而傳回 HTTP 狀態碼 400。 相關統計資料：總和
SQLFailedRequestCountBySysErr	因伺服器問題或功能限制而失敗的 _sql API 請求數。例如，請求可能會因為 VerificationException 而傳回 HTTP 狀態碼 503。 相關統計資料：總和
SQLRequestCount	向 _sql API 提出的請求數。 相關統計資料：總和
SQLDefaultCursorRequestCount	類似於 SQLRequestCount ，但只會計算分頁請求。 相關統計資料：總和
SQLUnhealthy	值為 1 表示在特定請求的回應中，SQL 外掛程式正在傳回 5xx 回應碼，或是正在將無效的查詢 DSL 傳遞至 OpenSearch。其他請求應會繼續成功。值為 0 表示最近沒有任何失敗。如果您看到值持續為 1，請針對您用戶端向外掛程式提出的請求進行故障診斷。

指標	Description
	相關統計資訊：Maximum

k-NN 指標

Amazon OpenSearch Service 包含 k 近鄰 (k-NN) 外掛程式的下列指標。

指標	Description
KNNCacheCapacityReached	是否已達到快取容量的每個節點指標。此指標僅與近似的 K-NN 搜尋相關。 相關統計資訊：Maximum
KNNCircuitBreakerTriggered	是否觸發斷路器的每個叢集指標。如果任何節點為 KNNCacheCapacityReached 傳回一個值，此值也將傳回 1。此指標僅與近似的 K-NN 搜尋相關。 相關統計資訊：Maximum
KNNEvictionCount	因記憶體限制或閒置時間而從快取移出的圖形數目的每個節點指標。因索引刪除而發生的明確移出不會計算在內。此指標僅與近似的 K-NN 搜尋相關。 相關統計資料：總和
KNNGraphIndexErrors	將文件的 knn_vector 欄位新增至圖形發生錯誤的請求數的每個節點指標。 相關統計資料：總和
KNNGraphIndexRequests	將文件的 knn_vector 欄位新增至圖形的請求數的每個節點指標。 相關統計資料：總和
KNNGraphMemoryUsage	目前快取大小 (記憶體中所有圖形的總大小) 的每個節點指標 (KB)。此指標僅與近似的 K-NN 搜尋相關。

指標	Description
	相關統計資訊：平均數
KNNGraphQueryErrors	產生錯誤之圖形查詢數目的每個節點指標。 相關統計資料：總和
KNNGraphQueryRequests	圖形查詢數目的每個節點指標。 相關統計資料：總和
KNNHitCount	快取命中次數的每個節點指標。當使用者查詢已載入記憶體的圖形時，就會發生快取命中。此指標僅與近似的 K-NN 搜尋相關。 相關統計資料：總和
KNNLoadExceptionCount	嘗試將圖形載入快取時發生例外狀況的次數的每個節點指標。此指標僅與近似的 K-NN 搜尋相關。 相關統計資料：總和
KNNLoadSuccessCount	外掛程式成功將圖形載入快取的次數的每個節點指標。此指標僅與近似的 K-NN 搜尋相關。 相關統計資料：總和
KNNMissCount	快取遺漏次數的每個節點指標。當使用者查詢尚未載入記憶體的圖形時，就會發生快取遺漏。此指標僅與近似的 K-NN 搜尋相關。 相關統計資料：總和
KNNQueryRequests	K-NN 外掛程式接收之查詢請求數目的每個節點指標。 相關統計資料：總和
KNNRemoteBuildEnabled	指定是否啟用功能的二進位值。 相關統計資料：二進位

指標	Description
KNNRemoteIndexBuildFailureCount	建置失敗的總數。 相關統計資料：總和
KNNRemoteIndexBuildSuccessCount	成功建置的總數。 相關統計資料：總和
KNNScriptCompilationErrors	指令碼編譯期間錯誤數目的每個節點指標。此統計數字僅與 K-NN 分數指令碼搜尋相關。 相關統計資料：總和
KNNScriptCompilations	K-NN 指令碼編譯次數的每個節點指標。此值通常應該為 1 或 0，但是如果包含已編譯指令碼的快取已填滿，K-NN 指令碼可能會被重新編譯。此統計數字僅與 K-NN 分數指令碼搜尋相關。 相關統計資料：總和
KNNScriptQueryErrors	指令碼查詢期間錯誤數目的每個節點指標。此統計數字僅與 K-NN 分數指令碼搜尋相關。 相關統計資料：總和
KNNScriptQueryRequests	指令碼查詢總數的每個節點指標。此統計數字僅與 K-NN 分數指令碼搜尋相關。 相關統計資料：總和
KNNTotalLoadTime	K-NN 將圖形載入到快取所需的時間 (以奈秒為單位)。此指標僅與近似的 K-NN 搜尋相關。 相關統計資料：總和
VectorIndexBuildAccelerationOCU	用於加速向量索引的 OpenSearch 運算單位 (OCU) 數量。 OCUs 相關統計資料：總和

跨叢集搜尋指標

Amazon OpenSearch Service 提供下列跨叢集搜尋指標。

來源網域指標

指標	維度	Description
CrossClusterOutboundConnections	ConnectionId	連線節點數。如果回應包含一或多個略過的網域，請使用此指標以追蹤任何運作狀態不良的連線。如果這個數字掉到 0，則表示連線運作狀態不良。
CrossClusterOutboundRequests	ConnectionId	傳送至目的地網域的搜尋請求數。用來檢查跨叢集搜尋請求的負載是否佔用網域，將此指標中的任何尖峰與任何 JVM/CPU 尖峰相互關聯。

目的地網域指標

指標	維度	Description
CrossClusterInboundRequests	ConnectionId	從來源網域收到的傳入連線請求數。

如果發生意外遺失連線的情況，則新增 CloudWatch 提醒。如需建立提醒的步驟，請參閱根據靜態閾值建立 CloudWatch 提醒。

跨叢集複寫指標

Amazon OpenSearch Service 提供下列跨叢集複寫指標。

指標	Description
ReplicationRate	每秒平均複寫操作速率。此指標類似於 IndexingRate 指標。
LeaderCheckPoint	對於特定連線，為所有複寫索引中領導檢查點值的總和。您可以使用此指標來測量複寫延遲。

指標	Description
FollowerCheckPoint	對於特定連線，為所有複寫索引中追蹤檢查點值的總和。您可以使用此指標來測量複寫延遲。
ReplicationNumSyncingIndices	具有複寫狀態 SYNCING 的索引數目。
ReplicationNumBootstrappingIndices	具有複寫狀態 BOOTSTRAPPING 的索引數目。
ReplicationNumPausedIndices	具有複寫狀態 PAUSED 的索引數目。
ReplicationNumFailedIndices	具有複寫狀態 FAILED 的索引數目。
CrossClusterOutboundReplicationRequests	追隨網域上的複寫傳輸請求數目。傳輸請求是內部的，每次呼叫複寫 API 操作時都會發生。當追隨網域輪詢來自領導網域的變更時，也會發生這種情況。
CrossClusterInboundReplicationRequests	領導網域上的複寫傳輸請求數目。傳輸請求是內部的，每次呼叫複寫 API 操作時都會發生。
AutoFollowNumSuccessStartReplication	針對特定連線，複寫規則成功建立的追蹤索引數目。

指標	Description
AutoFollowNumFailedStartReplication	當有相符模式時，複寫規則無法建立的追蹤索引數目。此問題可能是因為遠端叢集上的網路問題或安全性問題 (也就是說，關聯的角色沒有啟動複寫的許可) 所導致。
AutoFollowLeaderCallFailure	從追蹤索引到領導索引以選取新資料的查詢是否失敗。值 1 意味著在最近一分鐘有 1 個或多個失敗的呼叫。

Learning to Rank 指標

Amazon OpenSearch Service 提供下列 Learning to Rank 指標。

指標	Description
LTRRequestTotalCount	排名請求的總數。
LTRRequestErrorCount	未成功請求的總數。
LTRStatus.red	追蹤需要執行外掛程式的其中一個索引是否為紅色。
LTRMemoryUsage	外掛程式使用的總記憶體。
LTRFeatureMemoryUsageInBytes	Learning to Rank 功能欄位使用的記憶體容量 (以位元組為單位)。
LTRFeaturesetMemoryUsageInBytes	所有 Learning to Rank 功能集使用的記憶體容量 (以位元組為單位)。
LTRModelMemoryUsageInBytes	所有 Learning to Rank 模型使用的記憶體容量 (以位元組為單位)。

Piped Processing Language 指標

Amazon OpenSearch Service 提供下列 Piped Processing Language 指標。

指標	Description
PPLFailedRequestCountByCusErr	因用戶端問題而失敗的 _ppl API 請求數。例如，請求可能會因為 IndexNotFoundException 而傳回 HTTP 狀態碼 400。
PPLFailedRequestCountBySysErr	因伺服器問題或功能限制而失敗的 _ppl API 請求數。例如，請求可能會因為 VerificationException 而傳回 HTTP 狀態碼 503。
PPLRequestCount	向 _ppl API 提出的請求數。

使用 Amazon CloudWatch Logs 監控 OpenSearch 日誌

Amazon OpenSearch Service 透過 Amazon CloudWatch Logs 公開以下 OpenSearch 日誌：

- 錯誤日誌
- 搜尋請求慢速日誌
- 碎片慢速日誌
- 稽核日誌

搜尋碎片慢速日誌、索引碎片慢速日誌和錯誤日誌對於疑難排解效能和穩定性問題非常有用。稽核日誌會追蹤使用者活動，以符合規範。所有日誌都預設為停用狀態。如果已啟用，則套用標準 CloudWatch 定價。

> **Note**
>
> 錯誤日誌僅適用於 OpenSearch 和 Elasticsearch 5.1 版和更高版本。慢速日誌則適用於所有 OpenSearch 和 Elasticsearch 版本。

對於其日誌，OpenSearch 使用 Apache Log4j 2 及其內建的日誌層級 (從最不嚴重到最嚴重)：TRACE、DEBUG、INFO、WARN、ERROR 和 FATAL。

如果啟用錯誤日誌，OpenSearch Service 會將 WARN、ERROR 和 FATAL 的日誌行發佈至 CloudWatch。OpenSearch Service 也會發佈 DEBUG 等級的幾個例外狀況，包括下列項目：

- `org.opensearch.index.mapper.MapperParsingException`
- `org.opensearch.index.query.QueryShardException`
- `org.opensearch.action.search.SearchPhaseExecutionException`
- `org.opensearch.common.util.concurrent.OpenSearchRejectedExecutionException`
- `java.lang.IllegalArgumentException`

在許多情況下，錯誤日誌可以協助排除故障，包括：

- Painless 指令碼編譯問題
- 無效查詢
- 索引問題
- 快照故障
- 索引狀態管理遷移失敗

> **Note**
>
> 並非所有錯誤都會在錯誤日誌中回報。

> **Note**
>
> OpenSearch Service 不會記錄發生的所有錯誤。

主題

- 啟用日誌發佈 (主控台)
- 啟用日誌發佈 (AWS CLI)
- 啟用日誌發佈 (AWS SDKs)
- 啟用日誌發佈 (CloudFormation)
- 設定搜尋請求慢速日誌閾值
- 設定碎片慢速日誌閾值

- 測試慢速日誌
- 檢視日誌

啟用日誌發佈 (主控台)

OpenSearch Service 主控台是將日誌發佈到 CloudWatch 的最簡單方式。

若要將日誌發佈到 CloudWatch (主控台)

1. 前往 https：//aws.amazon.com。
2. 在 Analytics (分析) 下，選擇 Amazon OpenSearch Service。
3. 選取您要更新的網域。
4. 在 Logs (日誌) 索引標籤上，選取日誌類型，然後選擇 Enable (啟用)。
5. 建立新的 CloudWatch 日誌群組或選擇現有的日誌群組。

> **Note**
>
> 您若打算啟用多個日誌，建議發佈每一個到它自己的日誌群組。這個分隔有助於更輕鬆掃描日誌。

6. 選擇包含適當許可權的存取政策，或使用主控台提供的 JSON 建立政策：

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "arn:aws:logs:us-east-1:111122223333:log-
group:cw_log_group_name:*"
```

```
    }
  ]
}
```

建議您新增 aws:SourceAccount 和 aws:SourceArn 條件索引鍵至政策，保護自己免受混淆代理人問題的困擾。來源帳戶是網域的擁有者，且來源 ARN 是網域 ARN。您的網域必須位於服務軟體 R20211203 或更新版本上，才能新增這些條件索引鍵。

例如，您可以將下列條件區塊新增至政策：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

> **Important**
>
> CloudWatch Logs 支援各區域 10 個資源政策。如果您打算為數個 OpenSearch Service 網域啟用日誌，您應該建立和重複使用包含多個日誌群組的更廣泛政策，以避免達到此限制。如需更新政策的相關步驟，請參閱the section called “啟用日誌發佈 (AWS CLI)”。

7. 選擇啟用。

網域變更的狀態會從 Active (作用中) 變成 Processing (處理)。啟用日誌發佈之前，必須使狀態變回 Active (作用中)。這項變更通常需要 30 分鐘，但可能需要更長的時間，視您的網域組態而定。

如果您啟用其中一個碎片慢速日誌，請參閱 the section called “設定碎片慢速日誌閾值”。如果啟用稽核日誌，請參閱the section called “步驟 2：開啟 OpenSearch Dashboards 中的稽核日誌”。如果您只啟用了錯誤日誌，則不需要執行任何其他設定步驟。

啟用日誌發佈 (AWS CLI)

在啟用日誌發佈之前，您需要一個 CloudWatch 日誌群組。如果還沒有日誌群組，您可以使用下列命令建立一個：

```
aws logs create-log-group --log-group-name my-log-group
```

輸入下一個命令來尋找日誌群組的 ARN，然後記下來：

```
aws logs describe-log-groups --log-group-name my-log-group
```

現在您可以提供 OpenSearch Service 許可以寫入日誌群組。您接近命令最後，必須提供日誌群組的 ARN：

```
aws logs put-resource-policy \
  --policy-name my-policy \
  --policy-document '{ "Version": "2012-10-17",        "Statement": [{ "Sid":
 "", "Effect": "Allow", "Principal": { "Service": "es.amazonaws.com"}, "Action":
[ "logs:PutLogEvents","logs:CreateLogStream"],"Resource": "cw_log_group_arn:*"}]}'
```

Important

CloudWatch Logs 支援各區域 10 個資源政策。如果您計劃為數個 OpenSearch Service 網域啟用碎片慢速日誌，您應該建立並重複使用更廣泛的政策，其中包含多個日誌群組，以避免達到此限制。

如果您稍後需要檢閱此政策，請使用 `aws logs describe-resource-policies` 命令。若要更新此政策，請使用新的政策文件發出相同的 `aws logs put-resource-policy` 命令。

最後，您可以使用 `--log-publishing-options` 選項，以啟用發佈。選項的語法對於 `create-domain` 和 `update-domain-config` 命令都是相同的。

參數	有效值
`--log-publishing-options`	`SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`ES_APPLICATION_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`

參數	有效值
	AUDIT_LOGS={CloudWatchLogsLogGroupArn= *cw_log_group_arn* ,Enabled=true|false}

> **Note**
>
> 您若打算啟用多個日誌，建議發佈每一個到它自己的日誌群組。這個分隔有助於更輕鬆掃描日誌。

範例

下列範例會啟用發佈指定網域的搜尋和索引碎片慢速日誌：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --log-publishing-options
 "SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-group:my-log-
group,Enabled=true},INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-group:my-other-log-group,Enabled=true}"
```

若要停用發佈到 CloudWatch，可使用 `Enabled=false` 執行相同命令。

如果您啟用其中一個碎片慢速日誌，請參閱 the section called “設定碎片慢速日誌閾值”。如果啟用稽核日誌，請參閱the section called “步驟 2：開啟 OpenSearch Dashboards 中的稽核日誌”。如果您只啟用了錯誤日誌，則不需要執行任何其他設定步驟。

啟用日誌發佈 (AWS SDKs)

在可以啟用日誌發佈之前，必須先建立一個 CloudWatch 日誌群組，取得其 ARN，並提供 OpenSearch Service 許可以寫入到其中。相關的操作全記錄在 Amazon CloudWatch Logs API 參考中：

- `CreateLogGroup`
- `DescribeLogGroup`
- `PutResourcePolicy`

您可以使用 AWS 開發套件存取這些操作。

AWS SDKs(Android 和 iOS SDKs除外）支援 Amazon OpenSearch Service API 參考中定義的所有操作，包括 CreateDomain和 --log-publishing-options的選項UpdateDomainConfig。

如果您啟用其中一個碎片慢速日誌，請參閱 the section called “設定碎片慢速日誌閾值”。如果您只啟用了錯誤日誌，則不需要執行任何其他設定步驟。

啟用日誌發佈 (CloudFormation)

在此範例中，我們使用 CloudFormation 建立名為 的日誌群組opensearch-logs、指派適當的許可，然後建立已啟用應用程式日誌的日誌發佈、搜尋碎片慢速日誌和索引慢速日誌的網域。

在啟用日誌發佈之前，您需要建立 CloudWatch 日誌群組：

```
Resources:
  OpenSearchLogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      LogGroupName: opensearch-logs
Outputs:
  Arn:
    Value:
      'Fn::GetAtt':
        - OpenSearchLogGroup
        - Arn
```

範本輸出日誌群組的 ARN。在本案例中，ARN 為 arn:aws:logs:us-east-1:123456789012:log-group:opensearch-logs。

使用 ARN 會建立一項資源政策，賦予 OpenSearch Service 許可以寫入日誌群組：

```
Resources:
 OpenSearchLogPolicy:
   Type: AWS::Logs::ResourcePolicy
   Properties:
     PolicyName: my-policy
     PolicyDocument: "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Sid\": \"\",
 \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"es.amazonaws.com\"}, \"Action
\":[ \"logs:PutLogEvents\",\"logs:CreateLogStream\"],\"Resource\": \"arn:aws:logs:us-
east-1:123456789012:log-group:opensearch-logs:*\"}]}"
```

最後，建立以下 CloudFormation 堆疊，其會產生已啟用日誌發佈的 OpenSearch Service 網域。存取政策允許使用者 AWS 帳戶 讓 向網域提出所有 HTTP 請求。

```
Resources:
  OpenSearchServiceDomain:
    Type: "AWS::OpenSearchService::Domain"
    Properties:
      DomainName: my-domain
      EngineVersion: "OpenSearch_1.0"
      ClusterConfig:
        InstanceCount: 2
        InstanceType: "r6g.xlarge.search"
        DedicatedMasterEnabled: true
        DedicatedMasterCount: 3
        DedicatedMasterType: "r6g.xlarge.search"
      EBSOptions:
        EBSEnabled: true
        VolumeSize: 10
        VolumeType: "gp2"
      AccessPolicies:
        Version: "2012-10-17"
        Statement:
            Effect: "Allow"
            Principal:
                AWS: "arn:aws:iam::123456789012:user/es-user"
            Action: "es:*"
            Resource: "arn:aws:es:us-east-1:123456789012:domain/my-domain/*"
      LogPublishingOptions:
        ES_APPLICATION_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        SEARCH_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        INDEX_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
```

如需詳細的語法資訊，請參閱 CloudFormation 使用者指南中的日誌發佈選項。

設定搜尋請求慢速日誌閾值

搜尋請求慢速日誌可用於搜尋在 2.13 版和更新版本上執行的 OpenSearch Service 網域。搜尋請求慢速日誌閾值會針對總請求花費時間進行設定。這與針對個別碎片所花費時間設定的碎片請求慢速日誌不同。

您可以使用叢集設定指定搜尋請求慢速日誌。這與您使用索引設定啟用的碎片慢速日誌不同。例如，您可以透過 OpenSearch REST API 指定下列設定：

```
PUT domain-endpoint/_cluster/settings
{
  "transient": {
    "cluster.search.request.slowlog.threshold.warn": "5s",
    "cluster.search.request.slowlog.threshold.info": "2s"
  }
}
```

> **Note**
>
> PUT /_cluster/settings 端點在 Amazon OpenSearch Service 中受到限制。傳送此請求將傳回錯誤：Your request: '/_cluster/settings' payload is not allowed.若要設定碎片慢速日誌閾值，請改用索引層級設定。例如：
>
> ```
> PUT domain-endpoint/index/_settings
> {
>   "index.search.slowlog.threshold.query.warn": "5s",
>   "index.search.slowlog.threshold.query.info": "2s"
> }
> ```

設定碎片慢速日誌閾值

OpenSearch 預設會停用碎片慢速日誌。啟用碎片慢速日誌發佈至 CloudWatch 之後，您仍然必須為每個 OpenSearch 索引指定記錄閾值。這些閾值精確定義該記錄哪些內容、日誌層級為何等等。

例如，您可以透過 OpenSearch REST API 指定這些設定：

```
PUT domain-endpoint/index/_settings
{
```

```
  "index.search.slowlog.threshold.query.warn": "5s",
  "index.search.slowlog.threshold.query.info": "2s"
}
```

測試慢速日誌

若要測試搜尋請求和碎片慢速日誌是否皆成功發佈，請考慮從非常低的值開始，以驗證日誌是否出現在 CloudWatch 中，然後將閾值提高到更有用的層級。

如果日誌未出現，請檢查下列各項：

- CloudWatch 日誌群組是否存在？ 檢查 CloudWatch 主控台。
- OpenSearch Service 是否擁有寫入到日誌群組的許可？ 檢查 OpenSearch Service 主控台。
- OpenSearch Service 網域是否設定為發佈到日誌群組？ 檢查 OpenSearch Service 主控台、使用 AWS CLI `describe-domain-config`選項，或使用`DescribeDomainConfig`其中一個 SDKs呼叫。
- OpenSearch 記錄閾值是否低到您的請求超出它們？

  若要檢閱網域的搜尋請求慢速日誌閾值，請使用下列命令：

  ```
  GET domain-endpoint/_cluster/settings?flat_settings
  ```

  若要檢閱索引的碎片慢速日誌閾值，請使用下列命令：

  ```
  GET domain-endpoint/index/_settings?pretty
  ```

如果您想要停用索引的慢速日誌，請傳回變更為其預設值 -1 的任何閾值。

使用 OpenSearch Service 主控台停用發佈至 CloudWatch，或 AWS CLI 不會停止 OpenSearch 產生日誌；只會停止發佈這些日誌。如果您不再需要碎片慢速日誌，請務必檢查您的索引設定；如果您不再需要搜尋請求慢速日誌，請務必檢查您的網域設定。

檢視日誌

檢視 CloudWatch 中的應用程式和慢速日誌，就像檢視其他任何 CloudWatch 日誌一樣。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南中的檢視日誌資料。

以下是檢視日誌的一些考量事項：

- OpenSearch Service 只發佈每一行的前 255,000 個字元到 CloudWatch。任何剩餘內容會被截斷。對於稽核日誌，每個訊息為 10,000 個字元。
- 在 CloudWatch 中，日誌串流名稱的尾碼 `-index-slow-logs`、`-search-slow-logs`、`-application-logs` 和 `-audit-logs` 有助於識別其內容。

在 Amazon OpenSearch Service 中監控稽核日誌

如果您的 Amazon OpenSearch Service 網域使用精細存取控制，則您可以針對資料啟用稽核日誌。稽核日誌高度可自訂，可讓您追蹤 OpenSearch 叢集上的使用者活動，包括身分驗證成功與失敗、對 OpenSearch 的請求、索引變更以及傳入的搜尋查詢。預設設定會追蹤一組常用的使用者動作，但我們建議您根據您的確切需求量身打造設定。

就像 OpenSearch 應用程式日誌和慢速日誌一樣，OpenSearch Service 會將稽核日誌發佈到 CloudWatch Logs。如果已啟用，則套用標準 CloudWatch 定價。

> **Note**
>
> 若要啟用稽核日誌，您的使用者角色必須映射至 `security_manager` 角色，讓您可以存取 OpenSearch `plugins/_security` REST API。如需詳細資訊，請參閱 the section called "修改主要使用者"。

主題

- 限制
- 啟用稽核日誌
- 使用 啟用稽核記錄 AWS CLI
- 使用組態 API 啟用稽核日誌記錄
- 稽核日誌層和類別
- 稽核日誌設定
- 稽核日誌範例
- 使用 REST API 設定稽核日誌

限制

稽核日誌具有下列限制：

- 稽核日誌不包含被目的地網域存取政策拒絕的跨叢集搜尋請求。
- 每個稽核日誌訊息的大小上限為 10,000 個字元。如果稽核日誌訊息超過此限制，就會被截斷。

啟用稽核日誌

啟用稽核日誌分為兩個步驟。首先，您要設定網域，以便將稽核日誌發佈到 CloudWatch Logs。然後，您可以在 OpenSearch Dashboards 中啟用稽核日誌，並進行設定以符合您的需求。

> **Important**
>
> 如果您在執行這些步驟時遇到錯誤，請參閱the section called “無法啟用稽核日誌”，瞭解故障診斷資訊。

步驟 1：啟用稽核日誌並設定存取政策

以下步驟說明如何使用主控台啟用稽核日誌。您也可以使用 或 OpenSearch Service API 來啟用它們 AWS CLI。 OpenSearch

啟用 OpenSearch Service 網域的稽核日誌 (主控台)

1. 選擇要打開其組態的網域，然後前往 Logs (日誌) 索引標籤。
2. 選取 Audit logs (稽核日誌)，然後選取 Enable (啟用)。
3. 建立 CloudWatch 日誌群組或選擇現有的日誌群組。
4. 選擇包含適當許可權的存取政策，或使用主控台提供的 JSON 建立政策：

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
```

```
      ],
      "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
opensearch/domains/domain-name/*"
    }
  ]
}
```

建議您新增 aws:SourceAccount 和 aws:SourceArn 條件索引鍵至政策，保護自己免受混淆代理人問題的困擾。來源帳戶是網域的擁有者，且來源 ARN 是網域 ARN。您的網域必須位於服務軟體 R20211203 或更新版本上，才能新增這些條件索引鍵。

例如，您可以將下列條件區塊新增至政策：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

5. 選擇啟用。

步驟 2：開啟 OpenSearch Dashboards 中的稽核日誌

在於 OpenSearch Service 主控台中啟用稽核日誌之後，也必須在 OpenSearch Dashboards 中啟用稽核日誌，並進行設定以符合您的需求。

1. 打開 OpenSearch Dashboards，在左側選單中選擇 Security (安全性)。
2. 選擇 Audit logs (稽核日誌)。
3. 選擇 Enable audit logging (啟用稽核日誌)。

Dashboards UI 在 General settings (一般設定) 和 Compliance settings (合規設定) 下提供稽核日誌設定的完整控制。如需所有組態選項的說明，請參閱稽核日誌設定。

使用 啟用稽核記錄 AWS CLI

下列 AWS CLI 命令會在現有網域上啟用稽核日誌：

```
aws opensearch update-domain-config --domain-name my-domain --log-publishing-options
 "AUDIT_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-
group:my-log-group,Enabled=true}"
```

您也可以在建立網域時啟用稽核日誌。如需詳細資訊，請參閱 AWS CLI 命令參考。

使用組態 API 啟用稽核日誌記錄

下面對組態 API 的請求會啟用現有網域上的稽核日誌：

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "LogPublishingOptions": {
    "AUDIT_LOGS": {
      "CloudWatchLogsLogGroupArn":"arn:aws:logs:us-east-1:123456789012:log-
group1:sample-domain",
      "Enabled":true
    }
  }
}
```

如需詳細資訊，請參閱《Amazon OpenSearch Service API 參考》。

稽核日誌層和類別

叢集通訊發生在兩個不同的層：REST 層和傳輸層。

- REST layer 涵蓋與 HTTP 用戶端的通訊，例如 curl、Logstash、OpenSearch Dashboards、Java 高階 REST 用戶端、Python Requests 程式庫，所有到達叢集的 HTTP 請求。
- 傳輸層涵蓋節點之間的通訊。例如，搜尋請求到達叢集之後 (透過 REST 層)，服務該請求的協調節點會將查詢傳送至其他節點、接收其回應、收集必要的文件，並將它們整理成最終回應。諸如碎片配置和重新平衡等操作也會發生在傳輸層。

您可以啟用或停用整個層的稽核日誌，以及某層的個別稽核類別。下表包含稽核類別及其可用層的摘要。

類別	描述	適用於 REST	適用於傳輸
FAILED_LOGIN	請求包含無效憑證，且身分驗證失敗。	是	是
MISSING_PRIVILEGES	使用者沒有提出請求的權限。	是	是
GRANTED_PRIVILEGES	使用者擁有提出請求的權限。	是	是
OPENSEARCH_SECURITY_INDEX_ATTEMPT	請求嘗試修改 `.opendistro_security` 索引。	否	是
AUTHENTICATED	請求包含有效憑證，且身分驗證成功。	是	是
INDEX_EVENT	請求對索引執行系統管理操作，例如建立索引、設定別名或執行強制合併。如需此類別包含的 `indices:admin/` 動作的完整清單，請參閱 OpenSearch 文件。	否	是

除了這些標準類別之外，精細存取控制還提供數個額外的類別，專為符合資料合規要求而設計。

類別	描述
COMPLIANCE_DOC_READ	請求對索引中的文件執行讀取事件。
COMPLIANCE_DOC_WRITE	請求對索引中的文件執行寫入事件。

類別	描述
COMPLIANCE_INTERNAL_CONFIG_READ	請求對 `.opendistro_security` 索引執行讀取事件。
COMPLIANCE_INTERNAL_CONFIG_WRITE	請求對 `.opendistro_security` 索引執行寫入事件。

您可擁有類別和訊息屬性的任意組合。例如，如果您傳送 REST 請求以索引文件，您可能會在稽核日誌中看到下列幾行：

- REST 層上的 AUTHENTICATED (身分驗證)
- 傳輸層上的 GRANTED_PRIVILEGE (授權)
- COMPLIANCE_DOC_WRITE (文件寫入到索引)

稽核日誌設定

稽核日誌有許多組態選項。

一般設定

使用一般設定可啟用或停用個別類別或整個層。我們強烈建議您將 GRANTED_PRIVILEGES 和 AUTHENTICATED 保留為排除類別。否則，會針對每個有效請求將這些類別記錄至叢集。

名稱	後端設定	Description
REST 層	enable_rest	啟用或停用 REST 層上發生的事件。
REST 已停用類別	disabled_rest_categories	指定要在 REST 層上忽略的稽核類別。修改這些類別會大幅增加稽核日誌的大小。
傳輸層	enable_transport	啟用或停用傳輸層上發生的事件。

名稱	後端設定	Description
傳輸已停用類別	disabled_transport_categories	指定必須在傳輸層上忽略的稽核類別。修改這些類別會大幅增加稽核日誌的大小。

使用屬性設定可自訂每個日誌行中的詳細資訊量。

名稱	後端設定	Description
批量請求	resolve_bulk_requests	啟用此設定會為批量請求中的每個文件產生日誌，這會大幅增加稽核日誌的大小。
請求內文	log_request_body	包含請求的要求主體。
解析索引	resolve_indices	將別名解析為索引。

使用忽略設定來排除一組使用者或 API 路徑：

名稱	後端設定	Description
已忽略的使用者	ignore_users	指定想要排除的使用者。
已忽略的請求	ignore_requests	指定想要排除的請求模式。

合規設定

使用合規設定可針對索引、文件或欄位級存取進行調整。

名稱	後端設定	Description
合規日誌	enable_compliance	啟用或停用合規日誌。

您可以針對讀取和寫入事件日誌指定下列設定。

名稱	後端設定	Description
內部組態日誌	internal_config	啟用或停用 `.opendistro_security` 索引上的事件日誌。

您可以針對讀取事件指定下列設定。

名稱	後端設定	Description
讀取中繼資料	read_metadata_only	僅包含讀取事件的中繼資料。不包含任何文件欄位。
已忽略的使用者	read_ignore_users	請勿針對讀取事件包含特定使用者。
監視的欄位	read_watched_fields	指定針對讀取事件而監視的索引和欄位。新增監視的欄位會依據每個文件存取產生一個日誌，這會大幅增加稽核日誌的大小。監視的欄位支援索引模式和欄位模式： `{ "index-name-pattern": [ "field-name-pattern" ], "logs*": [ "message" ], "twitter": [ "id", "user*" ] }`

您可以針對寫入事件指定下列設定。

名稱	後端設定	Description
寫入中繼資料	write_metadata_only	僅包含寫入事件的中繼資料。不包含任何文件欄位。
日誌差異	write_log_diffs	如果 write_metadata_only 為 false，則只包括寫入事件之間的差異。
已忽略的使用者	write_ignore_users	不包括寫入事件的特定使用者。
監視指數	write_watched_indices	指定針對寫入事件而監視的索引或索引模式。新增監視的欄位會依據每個文件存取產生一個日誌，這會大幅增加稽核日誌的大小。

稽核日誌範例

本節包含索引的所有讀取和寫入事件的範例組態、搜尋請求以及產生的稽核日誌。

步驟 1：設定稽核日誌

啟用稽核日誌發佈至 CloudWatch Logs 群組之後，導覽至 OpenSearch Dashboards 稽核日誌頁面，然後選擇 Enable audit logging (啟用稽核日誌)。

1. 在 General Settings (一般設定) 中，選擇 Configure (設定)，並確保 REST layer (REST 層) 已啟用。
2. 在 Compliance Settings (合規設定) 中，選擇 Configure (設定)。
3. 在 Write (寫入) 下面的 Watched Fields (監視欄位) 中，將所有寫入事件的 `accounts` 新增至此索引。
4. 在 Read (讀取) 下面的 Watched Fields (監視欄位) 中，新增 `accounts` 索引的 `ssn` 和 `id-` 欄位：

```
{
  "accounts-": [
    "ssn",
    "id-"
  ]
```

```
}
```

步驟 2：執行讀取和寫入事件

1. 導覽至 OpenSearch Dashboards，選擇 Dev Tools (開發工具)，並索引範例文件：

```
PUT accounts/_doc/0
{
  "ssn": "123",
  "id-": "456"
}
```

2. 若要測試讀取事件，請傳送下列請求：

```
GET accounts/_search
{
  "query": {
    "match_all": {}
  }
}
```

步驟 3：觀察日誌

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇 Log groups (日誌群組)。
3. 選擇您在啟用稽核日誌時指定的日誌群組。在日誌群組中，OpenSearch Service 會為網域中的每個節點建立日誌串流。
4. 在 Log streams (日誌串流) 中，選擇 Search all (搜尋全部)。
5. 如需讀取和寫入事件，請參閱對應的日誌。在日誌出現之前，預計會有 5 秒延遲。

   寫入稽核日誌範例

```
{
  "audit_compliance_operation": "CREATE",
  "audit_cluster_name": "824471164578:audit-test",
  "audit_node_name": "be217225a0b77c2bd76147d3ed3ff83c",
  "audit_category": "COMPLIANCE_DOC_WRITE",
  "audit_request_origin": "REST",
```

```
  "audit_compliance_doc_version": 1,
  "audit_node_id": "3xNJhm4XS_yTzEgDWcGRjA",
  "@timestamp": "2020-08-23T05:28:02.285+00:00",
  "audit_format_version": 4,
  "audit_request_remote_address": "3.236.145.227",
  "audit_trace_doc_id": "lxnJGXQBqZSlDB91r_uZ",
  "audit_request_effective_user": "admin",
  "audit_trace_shard_id": 8,
  "audit_trace_indices": [
    "accounts"
  ],
  "audit_trace_resolved_indices": [
    "accounts"
  ]
}
```

讀取稽核日誌範例

```
{
  "audit_cluster_name": "824471164578:audit-docs",
  "audit_node_name": "806f6050cb45437e2401b07534a1452f",
  "audit_category": "COMPLIANCE_DOC_READ",
  "audit_request_origin": "REST",
  "audit_node_id": "saSevm9ASte0-pjAtYi2UA",
  "@timestamp": "2020-08-31T17:57:05.015+00:00",
  "audit_format_version": 4,
  "audit_request_remote_address": "54.240.197.228",
  "audit_trace_doc_id": "config:7.7.0",
  "audit_request_effective_user": "admin",
  "audit_trace_shard_id": 0,
  "audit_trace_indices": [
    "accounts"
  ],
  "audit_trace_resolved_indices": [
    "accounts"
  ]
}
```

若要包含要求主體，請回到 OpenSearch Dashboards 的 Compliance settings (合規設定)，然後停用 Write metadata (寫入中繼資料)。若要排除特定使用者的事件，請將使用者新增至 Ignored Users (已忽略的使用者)。

如需每個稽核日誌欄位的說明，請參閱稽核日誌欄位參考。如需搜尋和分析稽核日誌資料的相關資訊，請參閱 Amazon CloudWatch Logs 使用者指南中的使用 CloudWatch Logs Insights 分析日誌資料。

使用 REST API 設定稽核日誌

我們建議您使用 OpenSearch Dashboards 來設定稽核日誌，但您也可以使用精細存取控制 REST API。本節包含請求範例。如需 REST API 的完整文件，請參閱 OpenSearch 文件。

```
PUT _opendistro/_security/api/audit/config
{
  "enabled": true,
  "audit": {
    "enable_rest": true,
    "disabled_rest_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "enable_transport": true,
    "disabled_transport_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "resolve_bulk_requests": true,
    "log_request_body": true,
    "resolve_indices": true,
    "exclude_sensitive_headers": true,
    "ignore_users": [
      "kibanaserver"
    ],
    "ignore_requests": [
      "SearchRequest",
      "indices:data/read/*",
      "/_cluster/health"
    ]
  },
  "compliance": {
    "enabled": true,
    "internal_config": true,
    "external_config": false,
    "read_metadata_only": true,
    "read_watched_fields": {
      "read-index-1": [
        "field-1",
```

```
        "field-2"
      ],
      "read-index-2": [
        "field-3"
      ]
    },
    "read_ignore_users": [
      "read-ignore-1"
    ],
    "write_metadata_only": true,
    "write_log_diffs": false,
    "write_watched_indices": [
      "write-index-1",
      "write-index-2",
      "log-*",
      "*"
    ],
    "write_ignore_users": [
      "write-ignore-1"
    ]
  }
}
```

使用 Amazon EventBridge 監控 OpenSearch Service 事件

Amazon OpenSearch Service 與 Amazon EventBridge 整合，可通知您影響您的網域的特定事件。來自 AWS 服務的事件會以近乎即時的方式交付至 EventBridge。相同的事件也會傳送至 Amazon CloudWatch Events，或 Amazon EventBridge 的前身產品。您可編寫簡單的規則，來指示您在意的事件，以及當事件符合規則時所要自動執行的動作。可以自動觸發的動作如下：

- 叫用 AWS Lambda 函數
- 叫用 Amazon EC2 執行命令
- 將事件轉傳至 Amazon Kinesis Data Streams
- 啟用 AWS Step Functions 狀態機器
- 通知 Amazon SNS 主題或 Amazon SQS 佇列

如需詳細資訊，請參閱 Amazon EventBridge 使用者指南中的 Amazon EventBridge 入門。

主題

- 服務軟體更新事件
- 自動調整事件
- 叢集運作狀態事件
- VPC 端點事件
- 節點淘汰事件
- 已降級的節點淘汰事件
- 網域錯誤事件
- 教學課程：接聽 Amazon OpenSearch Service EventBridge 事件
- 教學課程：可用軟體更新的 Amazon SNS 提醒

服務軟體更新事件

當以下其中一個服務軟體更新事件發生時，OpenSearch Service 會將事件傳送至 EventBridge。

可用的服務軟體更新

OpenSearch Service 會在服務軟體更新可用時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
    "description": "Service software update R20220928 available. Service Software
 Deployment Mechanism:
```

```
                    Blue/Green. For more information on deployment configuration,
 please
                    see: https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
  }
}
```

已排程服務軟體更新

已排程服務軟體更新時，OpenSearch Service 會傳送此事件。對於選用更新，您會在排程日期收到通知，並且可以選擇隨時重新排程。對於必要的更新，您會在排程日期前三天收到通知，而且您可以選擇在強制時段內重新排程。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Scheduled",
    "severity": "High",
    "description": "A new service software update [R20200330-p1] has been scheduled at
 [21st May 2023 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
  }
}
```

已重新排程服務軟體更新

OpenSearch Service 會傳送此事件。如需詳細資訊，請參閱the section called “選用與必要的更新”。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Rescheduled",
    "severity": "High",
    "description": "The service software update [R20200330-p1], which was originally
 scheduled for
                    [21st May 2023 12:40 GMT], has been rescheduled to [23rd May 2023
 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
  }
}
```

服務軟體更新已開始

OpenSearch Service 會在服務軟體更新已開始時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
```

```
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Started",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] started.
  }
}
```

服務軟體更新已完成

OpenSearch Service 會在服務軟體更新已完成時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Completed",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] completed."
  }
}
```

服務軟體更新已取消

OpenSearch Service 會在服務軟體更新取消時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Cancelled",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] has been
 cancelled as a
                    newer update is available. Please schedule the latest update."
  }
}
```

已取消排定的服務軟體更新

OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Cancelled",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] has been
 cancelled."
  }
```

```
}
```

服務軟體更新未執行

當 OpenSearch Service 無法啟動服務軟體更新時，會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Unexecuted",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] cannot be
 started. Reason: [reason]"
  }
}
```

服務軟體更新已失敗

OpenSearch Service 會在服務軟體更新失敗時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
```

```
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Failed",
    "severity": "High",
    "description": "Installation of service software update [R20200330-p1] failed.
 [reason].
  }
}
```

需要服務軟體更新

OpenSearch Service 會在需要服務軟體更新時傳送此事件。如需詳細資訊，請參閱the section called "選用與必要的更新"。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Required",
    "severity": "High",
    "description": "Service software update [R20200330-p1] available. Update
                    will be automatically installed after [21st May 2023] if no
                    action is taken. Service Software Deployment Mechanism: Blue/Green.
                    For more information on deployment configuration, please see:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
  }
}
```

自動調整事件

當以下其中一個自動調整事件發生時，OpenSearch Service 會將事件傳送至 EventBridge。

自動調整待定

當「自動調整」已找到可改善叢集效能和可用性的調整建議時，OpenSearch Service 會傳送此事件。您只能在已停用「自動調整」的網域中看到此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Pending",
    "description": "Auto-Tune recommends the following new settings for your
 domain: { JVM Heap size : 60%}. Enable Auto-Tune to improve cluster stability and
 performance.",
    "scheduleTime": "{iso8601-timestamp}"
  }
}
```

自動調整已開始

當「自動調整」開始將新設定套用至您的網域時，OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
```

```
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Started",
    "scheduleTime": "{iso8601-timestamp}",
    "startTime": "{iso8601-timestamp}",
    "description" : "Auto-Tune is applying the following settings to your domain: { JVM
 Heap size : 60%}."
  }
}
```

自動調整需要排程的藍/綠部署

當「自動調整」已找到需要排程藍/綠部署的調整建議時，OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
    "status": "Pending",
    "startTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has identified the following settings for your domain
 that require a blue/green deployment: { JVM Heap size : 60%}.
```

```
                    You can schedule the deployment for your preferred time."
  }
}
```

自動調整已取消

當「自動調整」排程因為沒有待定調整建議而取消時，OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
    "status": "Cancelled",
    "scheduleTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has cancelled the upcoming blue/green deployment."
  }
}
```

自動調整已完成

當「自動調整」已完成藍/綠部署，且叢集正在使用新 JVM 設定運作時，OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
```

```
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "completionTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has completed the blue/green deployment and successfully
 applied the following settings: { JVM Heap size : 60%}."
  }
}
```

自動調整已停用且變更已還原

當「自動調整」已停用且套用的變更已復原時，OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": [ "arn:aws:es:us-east-1:123456789012:domain/test-domain" ],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "description": "Auto-Tune is now disabled. All settings have been reverted. Auto-
Tune will continue to evaluate
                    cluster performance and provide recommendations.",
    "completionTime": "{iso8601-timestamp}"
  }
```

```
}
```

自動調整已停用且變更已保留

當「自動調整」已停用且套用的變更已保留時，OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "description": "Auto-Tune is now disabled. The most-recent settings by Auto-Tune
 have been retained.
                    Auto-Tune will continue to evaluate cluster performance and provide
 recommendations.",
    "completionTime": "{iso8601-timestamp}"
  }
}
```

叢集運作狀態事件

OpenSearch Service 會在叢集運作狀態受影響時，向 EventBridge 傳送特定事件。

紅色叢集恢復已啟動

OpenSearch Service 會在您的叢集狀態連續為紅色超過一小時後發送此事件。其會嘗試自動從快照中恢復一個或多個紅色索引，以修復叢集狀態。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Started",
      "severity":"High",
      "description":"Your cluster status is red. We have started automatic snapshot
 restore for the red indices.
                    No action is needed from your side. Red indices [red-index-0, red-
index-1]"
   }
}
```

紅色叢集恢復部分完成

當 OpenSearch Service 在嘗試修復紅色叢集狀態時，若只能從快照恢復紅色索引的子集，即會發送此事件。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
```

```
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Partially Restored",
      "severity":"High",
      "description":"Your cluster status is red. We were able to restore the following
 Red indices from
                    snapshot: [red-index-0]. Indices not restored: [red-index-1].
 Please refer https://docs.aws.amazon.com/opensearch-service/latest/developerguide/
handling-errors.html#handling-errors-red-cluster-status for troubleshooting steps."
   }
}
```

紅色叢集恢復失敗

當 OpenSearch Service 在嘗試修復紅色叢集狀態時，若無法恢復任何索引，即會發送此事件。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Failed",
      "severity":"High",
      "description":"Your cluster status is red. We were unable to restore the Red
 indices automatically.
                    Indices not restored: [red-index-0, red-index-1]. Please refer
 https://docs.aws.amazon.com/opensearch-service/latest/developerguide/handling-
errors.html#handling-errors-red-cluster-status for troubleshooting steps."
   }
}
```

待刪除的碎片

當 OpenSearch Service 嘗試自動修復連續 14 天為紅色的紅色叢集狀態後，若仍有一個或多個索引保持紅色狀態，即會發送此事件。再經過 7 天（總共 21 天連續紅色）後，OpenSearch Service 將著手在所有紅色索引進行刪除未指派的碎片。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2022-04-09T10:36:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "severity":"Medium",
      "description":"Your cluster status is red. Please fix the red indices as soon as
 possible.
                    If not fixed by 2022-04-12 01:51:47+00:00, we will delete all
 unassigned shards,
                    the unit of storage and compute, for these red indices to recover
 your domain and make it green.
                    Please refer to https://docs.aws.amazon.com/opensearch-service/
latest/developerguide/handling-errors.html#handling-errors-red-cluster-status for
 troubleshooting steps.
                    test_data, test_data1",
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Shard(s) to be deleted"
   }
}
```

碎片已刪除

OpenSearch Service 會在您的叢集狀態連續為紅色超過 21 天後發送此事件。其會著手刪除所有紅色索引上的未指派碎片（儲存和運算）。如需詳細資訊，請參閱the section called “紅色叢集的自動修復”。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2022-04-09T10:54:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "severity":"High",
      "description":"We have deleted unassinged shards, the unit of storage and
 compute, in
                    red indices: index-1, index-2 because these indices were red for
 more than
                    21 days and could not be restored with the automated restore
 process.
                    Please refer to https://docs.aws.amazon.com/opensearch-service/
latest/developerguide/handling-errors.html#handling-errors-red-cluster-status for
 troubleshooting steps.",
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Shard(s) deleted"
   }
}
```

高碎片計數警告

在熱資料節點的平均碎片計數超過建議預設限制 (1,000) 的 90% 時，OpenSearch Service 便會傳送此事件。雖然較高版本的 Elasticsearch 和 OpenSearch 支援可設定之每節點最大碎片數限制，但我們建議每個節點不要超過 1,000 個碎片。請參閱選擇碎片數。

範例

以下為此類事件的範例：

```
{
```

```
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High Shard Count",
     "status":"Warning",
     "severity":"Low",
     "description":"One or more data nodes have close to 1000 shards. To ensure optimum
 performance and stability of your
                    cluster, please refer to the best practice guidelines - https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/sizing-domains.html#bp-
sharding."
  }
}
```

已超過碎片數量限制

在熱資料節點的平均碎片計數超過建議預設限制 (1,000) 時，OpenSearch Service 便會傳送此事件。雖然較高版本的 Elasticsearch 和 OpenSearch 支援可設定之每節點最大碎片數限制，但我們建議每個節點不要超過 1,000 個碎片。請參閱選擇碎片數。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High Shard Count",
     "status":"Warning",
     "severity":"Medium",
```

```
     "description":"One or more data nodes have more than 1000 shards. To ensure
 optimum performance and stability of your
                    cluster, please refer to the best practice guidelines - https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/sizing-domains.html#bp-
sharding."
  }
}
```

磁碟空間不足

在叢集中一個或多個節點的可用儲存空間少於 25% 或小於 25 GB 時，OpenSearch Service 便會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Low Disk Space",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more data nodes in your cluster has less than 25% of storage
 space or less than 25GB.
                   Your cluster will be blocked for writes at 20% or 20GB. Please refer
 to the documentation for more information - https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/handling-errors.html#troubleshooting-cluster-block"
  }
}
```

低磁碟浮水印違規

在叢集中所有節點的可用儲存空間少於 10% 或小於 10 GB 時，OpenSearch Service 便會傳送此事件。當所有節點都違反低磁碟浮水印時，任何新的索引都會產生黃色叢集，當所有節點都低於高磁碟浮水印時，就會產生紅色叢集。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Low Disk Watermark Breach",
     "status":"Warning",
     "severity":"Medium",
     "description":"Low Disk Watermark threshold is about to be breached. Once the
 threshold is breached, new index creation will be blocked on all
                    nodes to prevent the cluster status from turning red. Please
 increase disk size to suit your storage needs. For more information,
                    see https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#troubleshooting-cluster-block".
  }
}
```

EBS 爆量餘額低於 70%

當一個或多個資料節點上的 EBS 爆量餘額低於 70% 時，OpenSearch Service 會傳送此事件。EBS 爆量餘額損耗可能會導致廣泛的叢集無法使用和 I/O 請求限流，進而導致索引和搜尋請求的高延遲和逾時。如需修正此問題的步驟，請參閱 the section called “低 EBS 爆量餘額”。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
```

```
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"EBS Burst Balance",
     "status":"Warning",
     "severity":"Medium",
     "description":"EBS burst balance on one or more data nodes is below 70%.
                    Follow https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#handling-errors-low-ebs-burst
                    to fix this issue."
  }
}
```

EBS 爆量餘額低於 20%

當一個或多個資料節點上的 EBS 爆量餘額低於 20% 時，OpenSearch Service 會傳送此事件。EBS 爆量餘額損耗可能會導致廣泛的叢集無法使用和 I/O 請求限流，進而導致索引和搜尋請求的高延遲和逾時。如需修正此問題的步驟，請參閱 the section called “低 EBS 爆量餘額”。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"EBS Burst Balance",
     "status":"Warning",
     "severity":"High",
     "description":"EBS burst balance on one or more data nodes is below 20%.
                    Follow https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#handling-errors-low-ebs-burst
                    to fix this issue.
  }
}
```

磁碟輸送量限流

由於 EBS 磁碟區或 EC2 執行個體的輸送量限制，正在調節對網域的讀取和寫入請求時，OpenSearch Service 會傳送此事件。如果您收到此通知，請考慮遵循 AWS 建議的最佳實務來擴展磁碟區或執行個體。如果您的磁碟區類型為 gp2，請增加磁碟區大小。如果您的磁碟區類型為 gp3，請佈建更多輸送量。您也可以檢查執行個體基礎和最大 EBS 輸送量大於或等於佈建的磁碟區輸送量，並據以擴展。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Disk Throughput Throttle",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is experiencing throttling due to instance or volume
 throughput limitations.
                    Please consider scaling your domain to suit your throughput needs.
 In July 2023, we improved
                    the accuracy of throughput throttle calculation by replacing ‘Max
 volume throughput’ with
                    ‘Provisioned volume throughput’. Please refer to the documentation
 for more information."
  }
}
```

大型碎片大小

當叢集中的一或多個碎片超過 50GiB 或 65GiB 時，OpenSearch Service 會傳送此事件。為了確保最佳的叢集效能和穩定性，請減少碎片大小。

如需詳細資訊，請參閱分片最佳實務。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Large Shard Size",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more shards are larger than 65GiB. To ensure optimum cluster
 performance and stability, reduce shard sizes.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-large-shard-
size."
  }
}
```

高 JVM 用量

OpenSearch Service 會傳送此事件。 JVMMemoryPressure如果超過 92% 達 30 分鐘，則會封鎖叢集的所有寫入操作。為了確保最佳的叢集穩定性，請減少叢集的流量或擴展您的網域，為您的工作負載提供足夠的記憶體。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
```

```
     "event":"High JVM Usage",
     "status":"Warning",
     "severity":"High",
     "description":"JVM memory pressure has exceeded 80%. If it exceeds 92% for 30
 minutes, all write operations to your cluster
                    will be blocked. To ensure optimum cluster stability, reduce
 traffic to the cluster or use larger instance types.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-high-jvm."
  }
}
```

GC 不足

OpenSearch Service 會傳送此事件。這可能表示 JVM 無法在工作負載的垃圾回收週期中回收足夠的記憶體。這可能會導致回應越來越慢和延遲更高；在某些情況下，甚至節點會因為運作狀態檢查逾時而下降。為了確保最佳的叢集穩定性，請減少叢集的流量或擴展您的網域，為您的工作負載提供足夠的記憶體。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Insufficient GC",
     "status":"Warning",
     "severity":"Medium",
     "description":"Maximum JVM is above 70% and JVM range is less than 30%. This may
 indicate insufficient garbage collection for your workload.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-insufficient-
gc."
  }
}
```

自訂索引路由警告

OpenSearch Service 會傳送此事件，這可能會導致藍綠部署停滯。確認設定已正確套用。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Custom Index Routing Warning",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is in processing state and contains indice(s) with
 custom index.routing.allocation
                    settings which can cause blue-green deployments to get stuck.
 Verify settings are applied properly.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-index-routing."
  }
}
```

失敗的碎片鎖定

OpenSearch Service 會傳送此事件[ShardLockObtainFailedException]。如需詳細資訊，請參閱[如何解決 Amazon OpenSearch Service 中的記憶體內碎片鎖定例外狀況？]

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
```

```
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Failed Shard Lock",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is unhealthy due to unassigned shards with
 [ShardLockObtainFailedException]. For more information,
                    see https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/monitoring-events.html#monitoring-events-failed-shard-lock."
}
```

VPC 端點事件

OpenSearch Service 會將特定事件傳送至與 AWS PrivateLink 介面端點相關的 EventBridge。

VPC 端點建立失敗

OpenSearch Service 會在無法建立請求的 VPC 端點時傳送此事件。發生此錯誤的原因可能是您已達到一個區域內允許的 VPC 端點數目上限。如果指定的子網路或安全群組不存在，您也會看到這個錯誤。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Create Validation",
      "status":"Failed",
```

```
      "severity":"High",
      "description":"Unable to create VPC endpoint aos-0d4c74c0342343 for domain
                    arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: You've reached the limit on the
                    number of VPC endpoints that you can create in the AWS Region."
   }
}
```

VPC 端點更新失敗

OpenSearch Service 會在無法刪除請求的 VPC 端點時傳送此事件。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Update Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to update VPC endpoint aos-0d4c74c0342343 for domain
                    arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: <failure message>."
   }
}
```

VPC 端點刪除失敗

OpenSearch Service 會在無法刪除請求的 VPC 端點時傳送此事件。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Delete Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to delete VPC endpoint aos-0d4c74c0342343 for domain
                     arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: Specified subnet doesn't exist."
    }
}
```

節點淘汰事件

OpenSearch Service 會將事件傳送至 EventBridge。

已排程節點淘汰

OpenSearch Service 會在排程節點淘汰時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
```

```
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Scheduled",
    "severity": "Medium",
    "description": "An automated action to retire and replace a node has been scheduled
 on your domain.
                    The node will be replaced in the next off-peak window. For more
 information, see
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/monitoring-events.html."
  }
}
```

節點淘汰已完成

OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Completed",
    "severity": "Medium",
    "description": "The node has been retired and replaced with a new node."
  }
}
```

節點淘汰失敗

OpenSearch Service 會在節點淘汰失敗時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Failed",
    "severity": "Medium",
    "description": "Node retirement failed. No actions are required from your end. We
 will automatically
                   retry replacing the node."
  }
}
```

已降級的節點淘汰事件

OpenSearch Service 會傳送這些事件。

已降級的節點淘汰通知

OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"db233454-aad1-7676-3b15-10a84b052baa",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2024-01-11T08:16:06Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
```

```
   ],
   "detail":{
      "severity":"Medium",
      "description":"An automated action to retire and replace a node has
 been scheduled on your domain. For more information, please see https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/monitoring-events.html.",
      "event":"Degraded Node Retirement Notification",
      "status":"Scheduled"
   }
}
```

降級節點淘汰完成

OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"7444215c-90f9-a52d-bcda-e85973a9a762",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2024-01-11T10:20:30Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
   ],
   "detail":{
      "severity":"Medium",
      "description":"The node has been retired and replaced with a new node.",
      "event":"Degraded Node Retirement Notification",
      "status":"Completed"
   }
}
```

降級的節點淘汰失敗

如果降級的節點淘汰失敗，OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"c328e9bb-93b9-c0b2-b17a-df527fdf96b6",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2024-01-11T08:31:38Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
   ],
   "detail":{
      "severity":"Medium",
      "description":"Node retirement failed. No actions are required from your end. We
 will automatically re-try replacing the node.",
      "event":"Degraded Node Retirement Notification",
      "status":"Failed"
   }
}
```

網域錯誤事件

當發生以下其中一個網域錯誤時，OpenSearch Service 會將事件傳送至 EventBridge。

網域更新驗證失敗

OpenSearch Service 在嘗試更新或執行網域上的組態變更時，如果遇到一個或多個驗證失敗，就會傳送此事件。如需解決這些失敗的步驟，請參閱 the section called “對驗證錯誤進行疑難排解”。

範例

以下為此類事件的範例：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Domain Update Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
```

```
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Domain Update Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to perform updates to your domain due to the following
 validation failures: <failures>
                    Please see the documentation for more information https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/managedomains-
configuration-changes.html#validation"
   }
}
```

無法存取 KMS 金鑰

OpenSearch Service 會在無法存取 AWS KMS 金鑰時傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Domain Error Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"KMS Key Inaccessible",
     "status":"Error",
     "severity":"High",
     "description":"The KMS key associated with this domain is inaccessible. You are at
 risk of losing access to your domain.
                    For more information, please refer to https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/encryption-at-rest.html#disabled-key."
  }
}
```

網域隔離

OpenSearch Service 會傳送此事件。

範例

以下為此類事件的範例：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2023-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Domain Isolation Notification",
     "status":"Error",
     "severity":"High",
     "description":"Your OpenSearch Service domain has been isolated. An isolated
 domain is unreachable by network and cannot receive, read, or write requests. For more
 information and assistance, please contact AWS Support at https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/encryption-at-rest.html#disabled-key."
   }
}
```

教學課程：接聽 Amazon OpenSearch Service EventBridge 事件

在本教學課程中，您會設定簡單的 AWS Lambda 函數來接聽 Amazon OpenSearch Service 事件，並將其寫入 CloudWatch Logs 日誌串流。

先決條件

此教學課程假設您擁有現有的 OpenSearch Service 網域。若您尚未建立網域，請依照 建立和管理網域 中的步驟建立一個。

步驟 1：建立 Lambda 函數

在此程序中，您要建立一個簡單的 Lambda 函數，以作為 OpenSearch Service 事件訊息的目標。

若要建立目標 Lambda 函數

1. 在 https：//https://console.aws.amazon.com/lambda/ 開啟 AWS Lambda 主控台。
2. 選擇 Create function (建立函數) 和 Author from scratch (從頭開始撰寫)。
3. 對於 Function name (函數名稱)，輸入 event-handler。
4. 針對執行階段，選擇 Python 3.8。
5. 選擇建立函數。
6. 在 Function code (函數程式碼) 區段中，編輯範本程式碼以符合下列範例：

```
import json

def lambda_handler(event, context):
    if event["source"] != "aws.es":
        raise ValueError("Function only supports input from events with a source
 type of: aws.es")

    print(json.dumps(event))
```

這是一個簡單的 Python 3.8 函數，此函數列印 OpenSearch Service 傳送的事件。如果一切設定正確，則在此教學課程的最後，事件詳細資訊出現在與此 Lambda 函數建立關聯的 CloudWatch Logs 日誌串流中。

7. 選擇部署。

步驟 2：註冊事件規則

在此步驟中，您會建立一個 EventBridge 規則，它可擷取 OpenSearch Service 網域中的事件。此規則會擷取定義它的帳戶中的所有事件。事件訊息本身包含事件來源資訊，包括其來源的網域。您可以使用此資訊以程式設計方式篩選和排序事件。

若要建立 EventBridge 規則

1. 在 https://console.aws.amazon.com/events/ 開啟 EventBridge 主控台。
2. 選擇建立規則。
3. 將規則命名為 event-rule。
4. 選擇下一步。
5. 對於事件模式，選取 services (AWS 服務)、Amazon OpenSearch Service 以及 All Events (所有事件)。此模式適用於所有 OpenSearch Service 網域，以及每個 OpenSearch Service 事件。或者，您可以建立一個更針對性的模式，來篩選掉一些結果。

6. 按下 Next (下一步)。
7. 對於目標，選擇 Lambda function (Lambda 函數)。在函數下拉式選單中，選擇 event-handler。
8. 按下 Next (下一步)。
9. 跳過標籤，然後再按一次 Next (下一步)。
10. 檢閱組態，然後選擇 Create rule (建立規則)。

步驟 3：測試組態

下次您在 OpenSearch Service 主控台的通知區段中收到通知時，若一切均正確設定，將觸發您的 Lambda 函數，它會將事件資料寫入到函數的 CloudWatch Log 日誌串流。

若要測試組態

1. 透過 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇 Logs (日誌)，然後選取 Lambda 函數的日誌群組 (例如，/aws/lambda/event-handler)。
3. 選取日誌串流，以檢視事件資料。

教學課程：可用軟體更新的 Amazon SNS 提醒

在本教學課程中，您會設定 Amazon EventBridge 事件規則，該規則會擷取 Amazon OpenSearch Service 中可用服務軟體更新的通知，並透過 Amazon Simple Notification Service (Amazon SNS) 傳送電子郵件通知給您。

先決條件

此教學課程假設您擁有現有的 OpenSearch Service 網域。若您尚未建立網域，請依照 建立和管理網域 中的步驟建立一個。

步驟 1：建立並訂閱 Amazon SNS 主題

設定 Amazon SNS 主題以作為新事件規則的事件目標。

若要建立 Amazon SNS 目標

1. 在 https://console.aws.amazon.com/sns/v3/home 開啟 Amazon SNS 主控台。

2. 選擇 Topics (主題) 和 Create topic (建立主題)。
3. 對於任務類型，請選擇 Standard (標準)，並將任務命名為 software-update。
4. 請選擇建立主題。
5. 建立主題之後，選擇 Create subscription (建立訂閱)。
6. 對於通訊協定，選擇電子郵件。對於 Endpoint (端點)，輸入您目前能存取的電子郵件地址，並選擇 Create subscription (建立訂閱)。
7. 檢查您的電子郵件帳戶，並等待接收訂閱確認電子郵件訊息。您收到訊息時，請選擇 Confirm subscription (確認訂閱)。

步驟 2：註冊事件規則

接著，註冊僅擷取服務軟體更新事件的事件規則。

若要建立事件規則

1. 在 https://console.aws.amazon.com/events/ 開啟 EventBridge 主控台。
2. 選擇建立規則。
3. 將規則命名為 softwareupdate-rule。
4. 選擇下一步。
5. 針對事件模式，選取AWS 服務、Amazon OpenSearch Service （受管叢集） 和 Amazon OpenSearch Service 軟體更新通知。此模式符合 OpenSearch Service 中的任何服務軟體更新事件。如需有關事件模式的詳細資訊，請參閱 Amazon EventBridge 使用者指南中的 Amazon EventBridge 事件規則。
6. 或著，您可以針對特定的嚴重性進行篩選。有關每個事件的嚴重性，請參閱 the section called “服務軟體更新事件”。
7. 選擇下一步。
8. 對於目標，選擇 SNS topic (SNS 主題)，然後選取 software-update。
9. 選擇下一步。
10. 跳過標籤，然後選擇 Next (下一步)。
11. 檢閱規則組態，然後選擇 Create rule (建立規則)。

下次您收到來自 OpenSearch Service 關於可用服務軟體更新的通知時，若一切均正確設定，Amazon SNS 應向您傳送有關該更新的電子郵件提醒。

使用 監控 Amazon OpenSearch Service API 呼叫 AWS CloudTrail

Amazon OpenSearch Service 與 整合 AWS CloudTrail，此服務提供使用者、角色或 OpenSearch Service 中 AWS 服務所採取動作的記錄。CloudTrail 會擷取 OpenSearch Service 的所有組態 API 呼叫作為事件。

> **Note**
>
> CloudTrail 只會擷取對組態 API 的呼叫，例如 `CreateDomain` 和 `GetUpgradeStatus`。CloudTrail 不會擷取對 OpenSearch API 的呼叫，例如 `_search` 和 `_bulk`。對於這些呼叫，請參閱the section called “監控稽核日誌”。

擷取的呼叫包括來自 OpenSearch Service 主控台 AWS CLI或 AWS SDK 的呼叫。如果您建立追蹤，就可以將 CloudTrail 事件持續交付到 Amazon S3 儲存貯體，包括 OpenSearch Service 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的 Event history (事件歷史記錄) 檢視最新事件。您可以利用 CloudTrail 所收集的資訊來判斷向 OpenSearch Service 發出的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

CloudTrail 中的 Amazon OpenSearch Service 資訊

當您建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。當活動在 OpenSearch Service 中發生時，該活動會與事件歷史記錄中的其他 AWS 服務事件一起記錄在 CloudTrail 事件中。您可以檢視、搜尋和下載 AWS 帳戶 帳戶的最新事件。如需詳細資訊，請參閱使用 CloudTrail 事件歷史記錄檢視事件。

若要持續記錄您 AWS 帳戶 帳戶中的事件，包括 OpenSearch Service 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。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。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- 為您的 建立線索 AWS 帳戶
- AWS 服務與 CloudTrail Logs 整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail 日誌檔案

CloudTrail 會記錄所有 OpenSearch Service 組態 API 動作，並會記錄在 Amazon OpenSearch Service API 參考中。

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：

- 是否使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出請求
- 提出該請求時，是否使用了特定角色或聯合身分使用者的臨時安全憑證
- 請求是否由其他 AWS 服務提出

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 Amazon OpenSearch Service 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的 CloudTrail 日誌項目會示範 `CreateDomain` 操作：

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/test-user",
    "accountId": "123456789012",
    "accessKeyId": "access-key",
    "userName": "test-user",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-08-21T21:59:11Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2018-08-21T22:00:05Z",
  "eventSource": "es.amazonaws.com",
  "eventName": "CreateDomain",
  "awsRegion": "us-west-1",
```

```
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "engineVersion": "OpenSearch_1.0",
    "clusterConfig": {
      "instanceType": "m4.large.search",
      "instanceCount": 1
    },
    "snapshotOptions": {
      "automatedSnapshotStartHour": 0
    },
    "domainName": "test-domain",
    "encryptionAtRestOptions": {},
    "eBSOptions": {
      "eBSEnabled": true,
      "volumeSize": 10,
      "volumeType": "gp2"
    },
    "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}",
    "advancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    }
  },
  "responseElements": {
    "domainStatus": {
      "created": true,
      "clusterConfig": {
        "zoneAwarenessEnabled": false,
        "instanceType": "m4.large.search",
        "dedicatedMasterEnabled": false,
        "instanceCount": 1
      },
      "cognitoOptions": {
        "enabled": false
      },
      "encryptionAtRestOptions": {
        "enabled": false
      },
      "advancedOptions": {
        "rest.action.multi.allow_explicit_index": "true"
      },
      "upgradeProcessing": false,
```

```
      "snapshotOptions": {
        "automatedSnapshotStartHour": 0
      },
      "eBSOptions": {
        "eBSEnabled": true,
        "volumeSize": 10,
        "volumeType": "gp2"
      },
      "engineVersion": "OpenSearch_1.0",
      "processing": true,
      "aRN": "arn:aws:es:us-west-1:123456789012:domain/test-domain",
      "domainId": "123456789012/test-domain",
      "deleted": false,
      "domainName": "test-domain",
      "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":\"es:*\",
\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}"
    }
  },
  "requestID": "12345678-1234-1234-1234-987654321098",
  "eventID": "87654321-4321-4321-4321-987654321098",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

使用 Cluster Insights 進行統一的操作監控

Amazon OpenSearch Service 現在包含 Cluster Insights，這是一種監控解決方案，可透過單一儀表板提供叢集的完整操作可見性。這消除了必須分析和關聯各種日誌和指標的複雜性，以識別叢集可用性或效能的潛在風險。解決方案可自動化跨節點、索引和碎片的關鍵操作資料的整合，將複雜的故障診斷轉換為簡化的程序。您可以偵測大型碎片和低磁碟浮水印等問題、檢視節點、索引和碎片層級的詳細指標，以及存取安全和彈性最佳實務。

> **Note**
>
> 所有執行 OpenSearch 2.17 版或更新版本的使用者都可以透過 OpenSearch Service UI 使用 Cluster Insights，無需額外費用。此外，具有 OpenSearch 2.17 和 2.19 版的網域也需要接受最新的服務軟體版本更新。

優勢

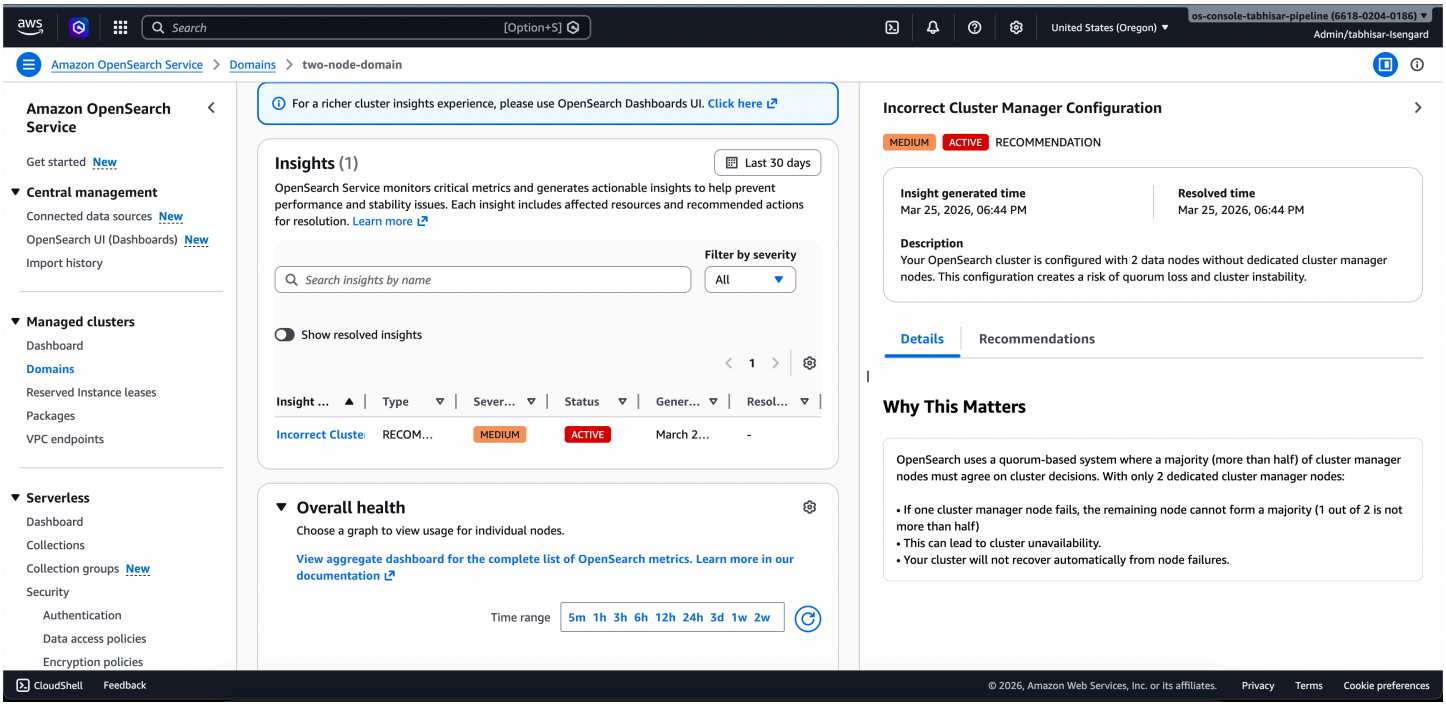
- 主動監控 - 使用所有元件的詳細效能指標主動監控叢集運作狀態 - 從個別節點和索引到碎片和搜尋查詢。
- 統一可見性 - 將監控資料合併到單一儀表板
- 可行的建議 - 取得問題解決的step-by-step指引
- 全面涵蓋 - 監控 OpenSearch 叢集的安全性、穩定性和彈性
- 查詢最佳化 - 識別資源密集型查詢並最佳化效能

使用 Cluster Insights，您可以維持最佳叢集效能、降低營運開銷，並確保 OpenSearch 叢集的最佳實務一致

透過主控台存取 Cluster Insights

檢閱效能和彈性建議，並在相同的主控台中進行必要的組態變更。在主控台的叢集運作狀態索引標籤下，您可以存取列出所有作用中洞見的叢集洞見。按一下任何 Insight 以檢視建議。

Screen-1：Cluster Health 索引標籤下的 Cluster Insights



透過 OpenSearch UI 存取 Cluster Insights 和詳細指標

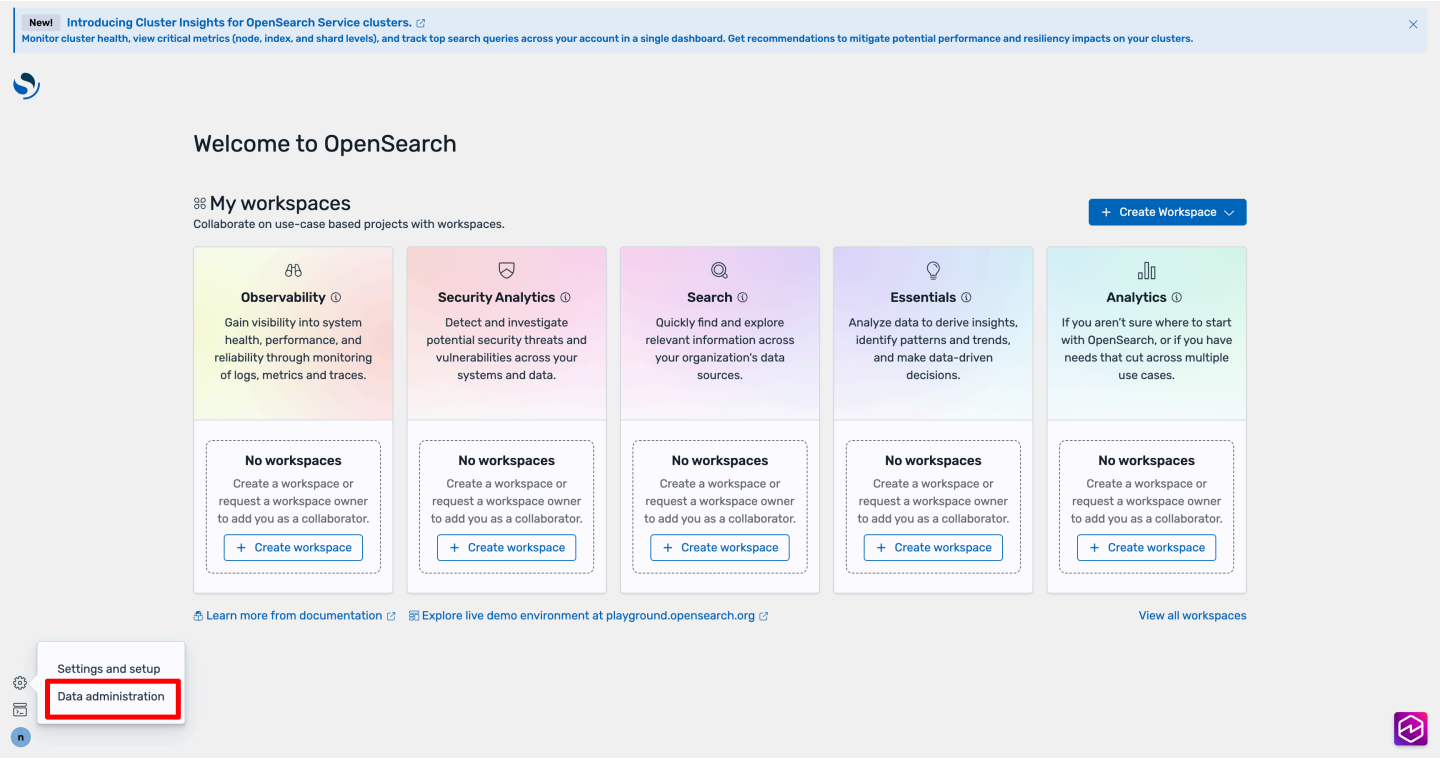
您可以透過 OpenSearch UI （儀表板） 檢視特定 OpenSearch Service 叢集的洞見。在 OpenSearch UI 中，應用程式只是資料夾之類的組織建構。每個應用程式都可以連接到多個 OpenSearch Service 叢集並顯示洞見。存取 Cluster Insights 需要 OpenSearch UI 應用程式中的管理角色。

> **Note**
>
> 存取 Cluster Insights 需要 OpenSearch UI 應用程式中的管理角色。

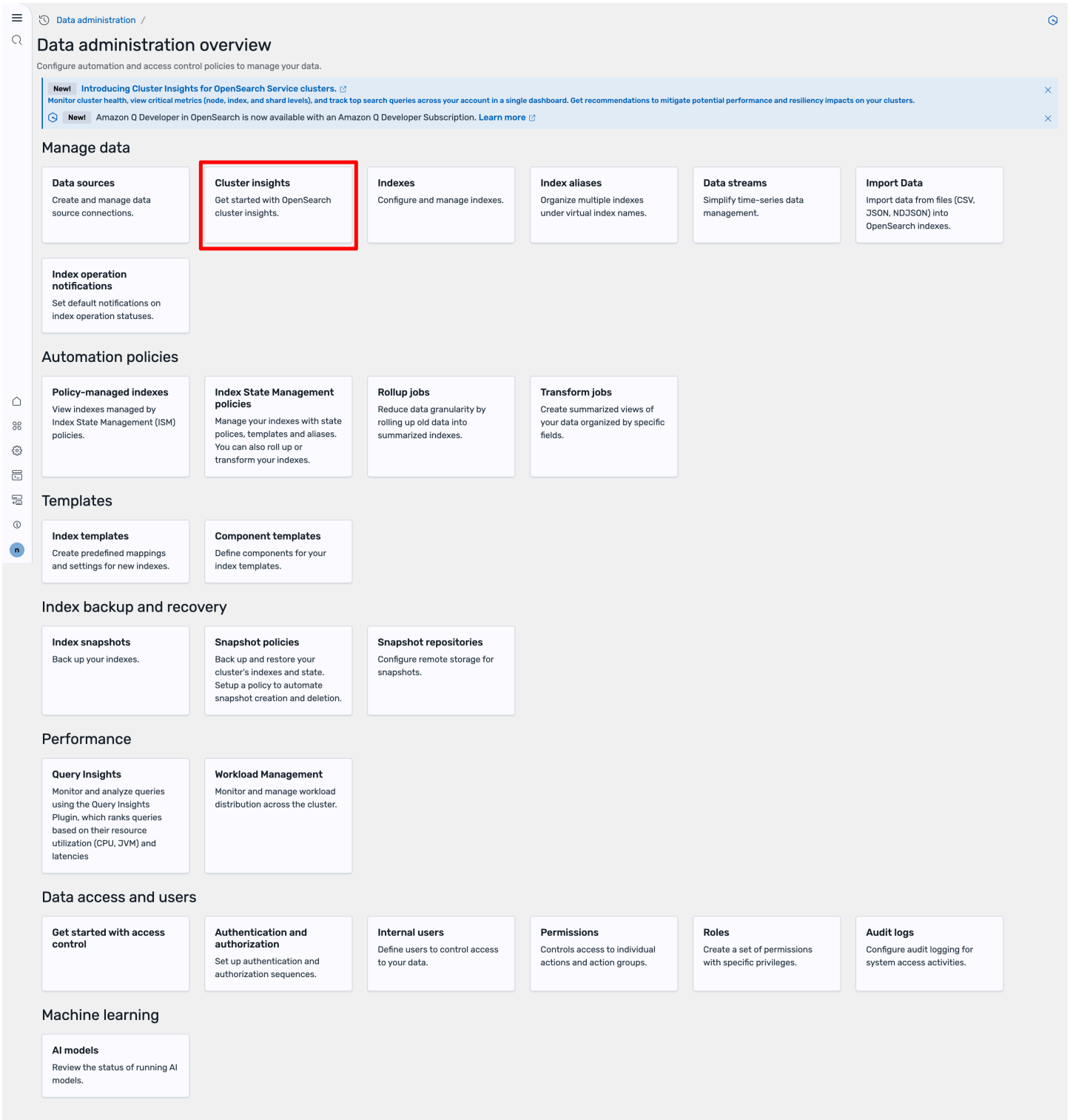
建立和設定應用程式以檢視 Cluster Insights

1. 在 開啟 OpenSearch Service 主控台 https://console.aws.amazon.com/aos/home
2. 從左側導覽中選擇 OpenSearch UI （儀表板）
3. 完成下列步驟以建立和設定應用程式：
    1. 建立 OpenSearch Service 應用程式
    2. 關聯資料來源
4. 完成上述兩個步驟後，您可以在設定 > 資料管理員 > Cluster Insights 區段下的 OpenSearch UI 儀表板中檢視 Cluster Insights。設定圖示位於 OpenSearch UI 畫面的左下角。

Screen-2：從 OpenSearch UI 存取資料管理員



Screen-3：管理資料區段下的叢集洞見



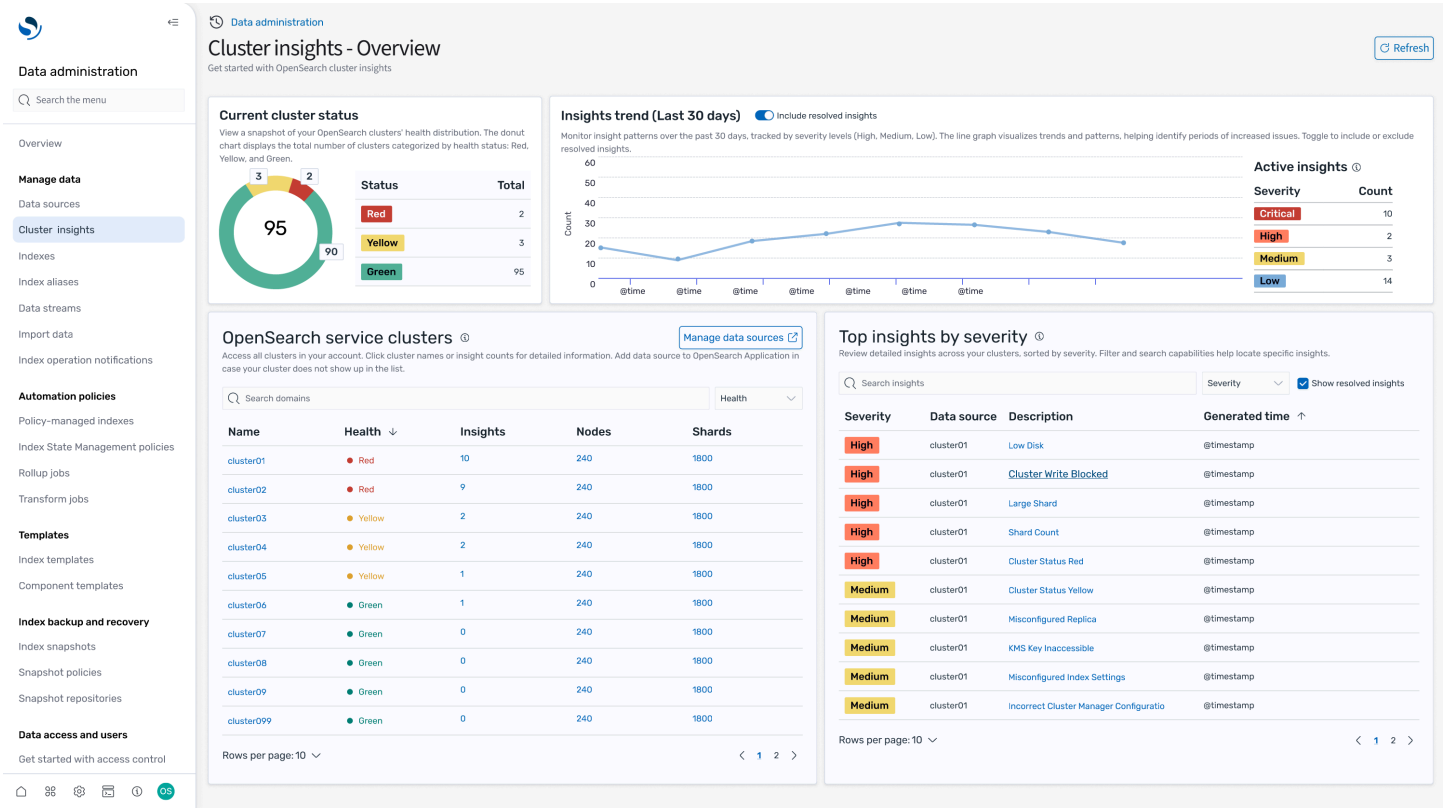
了解 Cluster Insights

本節說明 Cluster Insights 中可用的各種洞見。

概觀儀表板

Cluster Insights 概觀頁面，如下列螢幕擷取畫面所示，在應用程式層級提供叢集運作狀態的高階檢視，並包含下列各節：

Screen-4：OpenSearch UI 應用程式中的 Cluster Insights 登陸頁面。



目前叢集狀態

甜甜圈圖表會顯示您的叢集運作狀態：

- 綠色 - 所有主要碎片和複本都會配置給節點
- 黃色 - 已配置所有主要碎片，但不會配置某些複本
- 紅色 - 至少一個主碎片未配置給任何節點

Insights 趨勢

趨勢圖表會追蹤過去 30 天的問題模式，協助您識別新興問題並監控解決進度。

目前開放的洞見

依過去 30 天內開放洞見嚴重性整理的計數。

OpenSearch Service 叢集

本節列出所有 OpenSearch 叢集的金鑰統計資料，包括節點計數、碎片計數和作用中查詢。

依嚴重性列出的最佳洞見

您可以檢閱應用程式中所有網域的洞見。本節優先考慮需要立即注意的問題（關鍵和高嚴重性）。每個洞見都包含描述和特定建議，可協助您先專注於關鍵問題。

洞見詳細資訊

依嚴重性排名前的洞察區段中的每個洞察都是互動式的，並提供詳細的分析。例如，當您選擇大型碎片大小洞見時：

1. 您可以看到有多少碎片超過閾值，以及哪些索引受到影響。
2. 資源映射會識別每個大型碎片及其索引、ID 和目前大小。
3. 建議索引標籤提供step-by-step修補指引。
4. 歷史記錄索引標籤會顯示資源修復動作的時間軸。

叢集詳細資訊

當您在 OpenSearch Service 叢集區段中選取特定叢集時，OpenSearch 會在下列索引標籤中顯示該叢集的洞見：叢集運作狀態、節點檢視、索引檢視、碎片檢視和查詢檢視。叢集運作狀態索引標籤會顯示下列資訊：

概觀

金鑰資訊包括叢集運作狀態、碎片計數、節點計數、索引計數和文件統計資料。

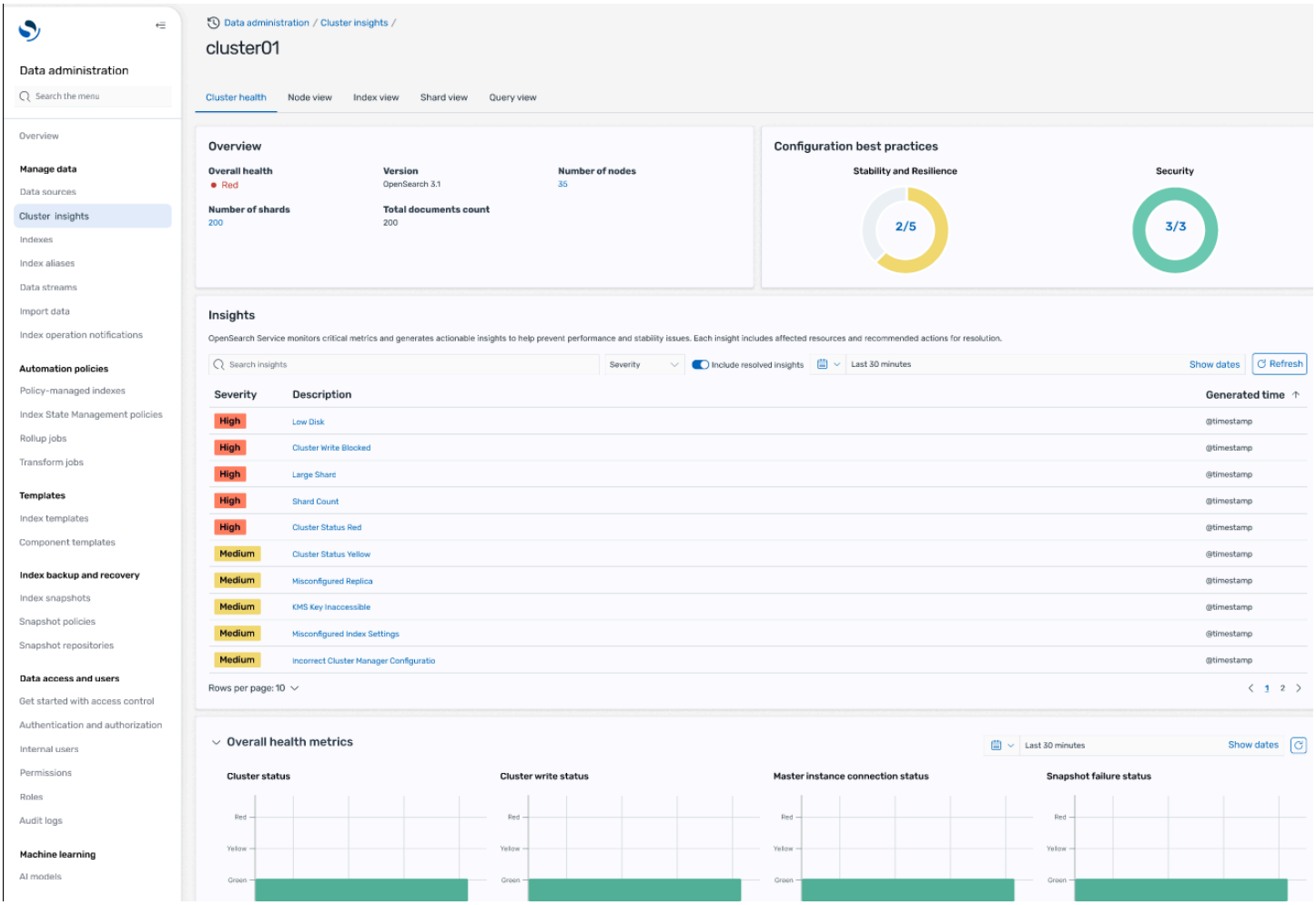
設定最佳實務

甜甜圈圖表顯示符合建議的彈性和安全性設定。

洞見

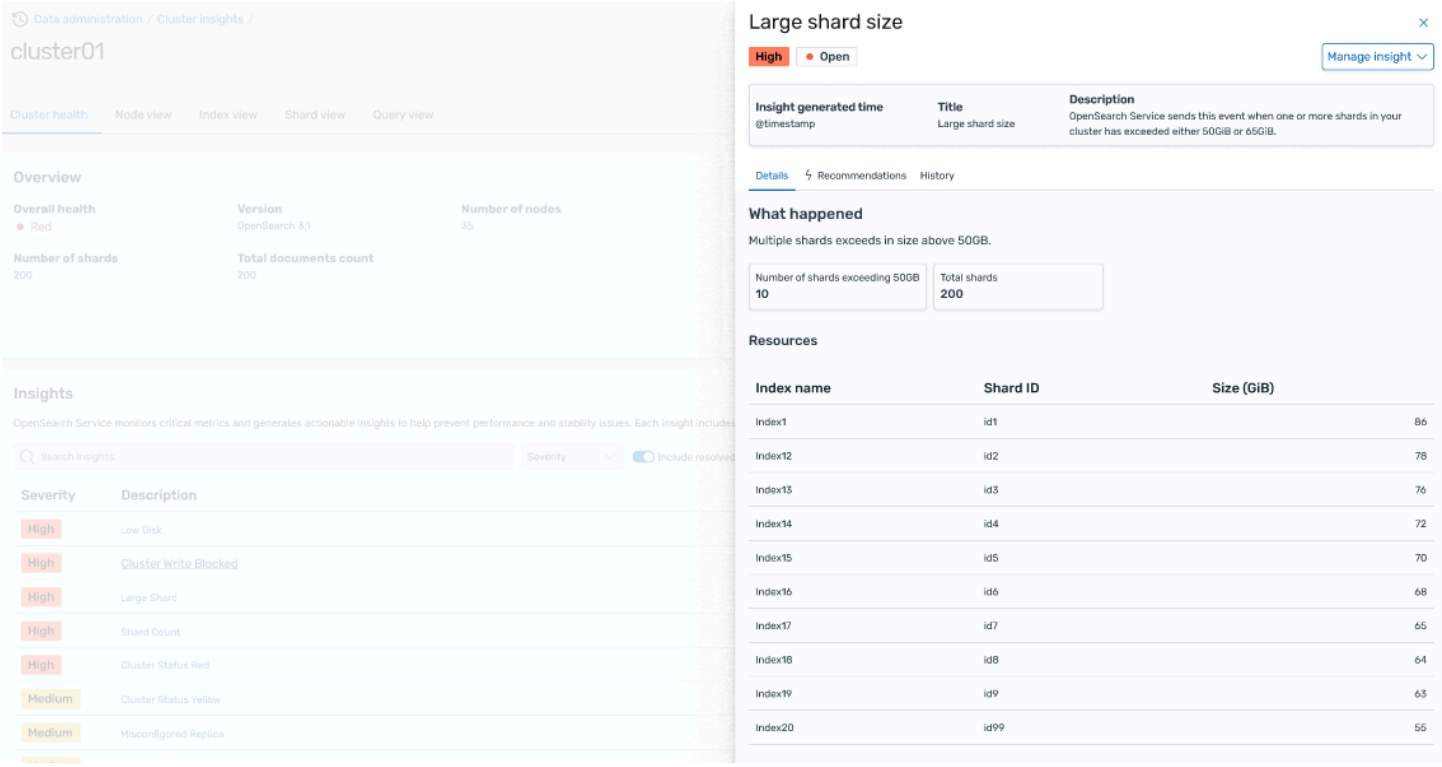
資料表列出為叢集產生的近期洞見，並提供與概觀頁面相同的詳細明細和修復指引。

Screen-5：Cluster Health 概觀提供關鍵指標、最佳實務和 Insights



當您按一下任何洞見時，可以看到詳細資訊和受影響的資源、建議。此外，您也可以查看固定資源的歷史記錄。

Screen-6：洞見詳細資訊。為您提供詳細資訊、建議和歷史時間軸。



指標區段

本節中的互動式圖表會顯示下列叢集指標：

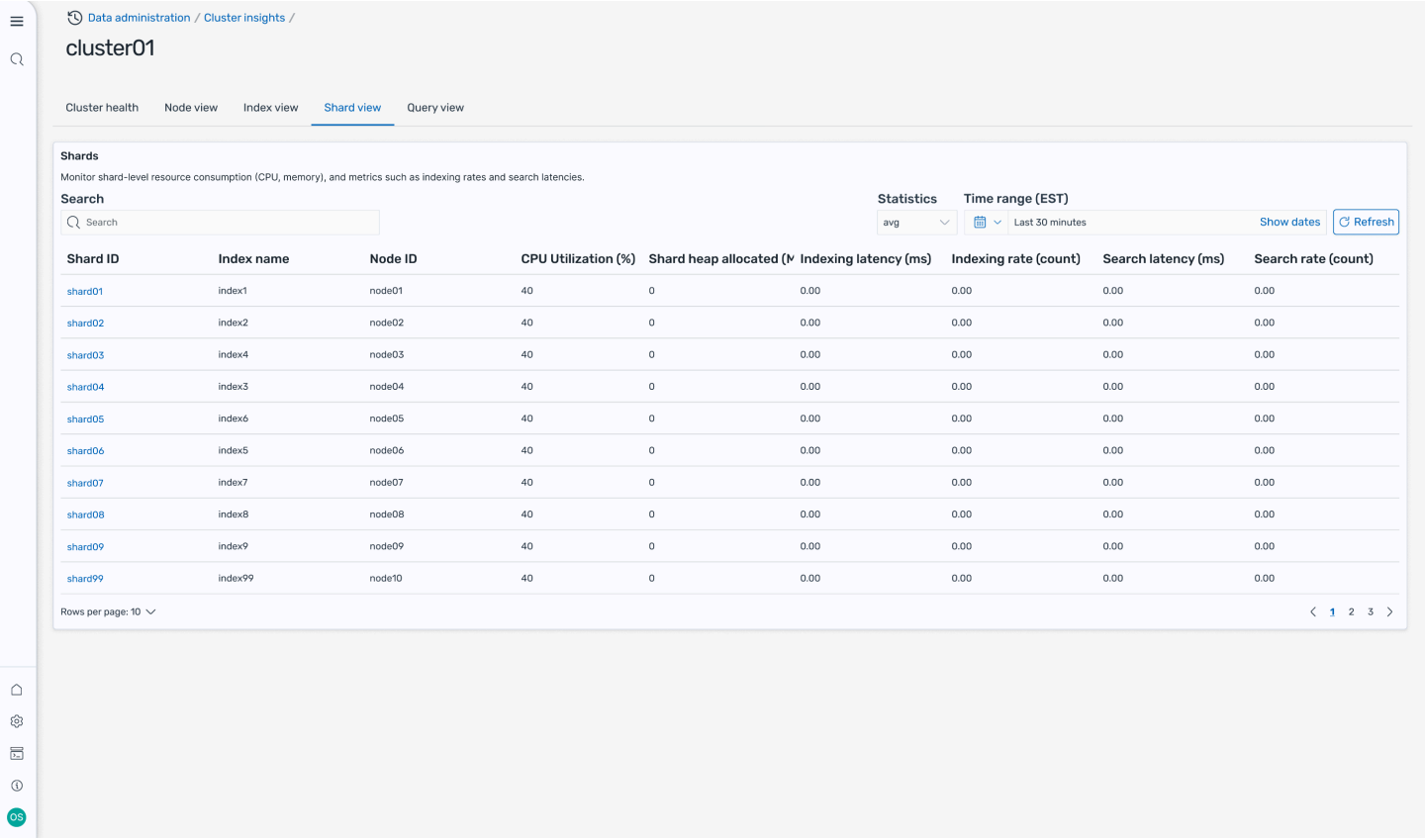
- 整體叢集運作狀態指標，例如叢集狀態、寫入狀態和可搜尋的文件
- KPIs（關鍵績效指標），例如索引和搜尋率和延遲
- JVM 和 CPU 使用率等資源使用率指標

節點、索引和碎片檢視

節點、索引和碎片檢視使用 OpenSearch 統計資料來提供叢集操作的詳細可見性。您可以檢視：

- CPU 使用率和 JVM 記憶體壓力等即時指標
- 搜尋和索引效能資料
- 叢集元件之間的資源熱點
- 精細節點層級診斷
- 配置的頂端碎片堆積

Screen-7：節點、索引和碎片層級指標



查詢檢視

> **Note**
>
> OpenSearch 2.19 版或更新版本支援查詢檢視功能。

查詢檢視頁面可協助您使用下列項目監控資源密集型查詢：

存取設定

檢視前 N 個查詢需要精細存取控制許可。請確定下列事項：

- 在您的網域上啟用精細存取控制。
- 您的 IAM 角色（或內部使用者）會映射至具有查詢洞察所需叢集許可的 OpenSearch 角色。
- 如需完整的管理員存取權，請將您的 IAM 角色 ARN 對應為後端角色，同時映射至 all_access 和 security_manager 角色。您可以在安全 > 角色 > 選取角色 > 映射使用者 > 管理映射，或使用安全 API (PUT _plugins/_security/api/rolesmapping/all_access) 在 OpenSearch Dashboards 中執行此操作。

如果沒有適當的角色映射，使用者在嘗試存取查詢洞見資料時可能會收到 403 禁止的錯誤。如需詳細資訊，請參閱精細存取控制。

即時儀表板

檢視每個查詢的執行統計資料、CPU 和記憶體用量，以及完成進度。

前 N 個查詢

排名資料表顯示最重要的查詢，其中包含詳細資訊，包括：

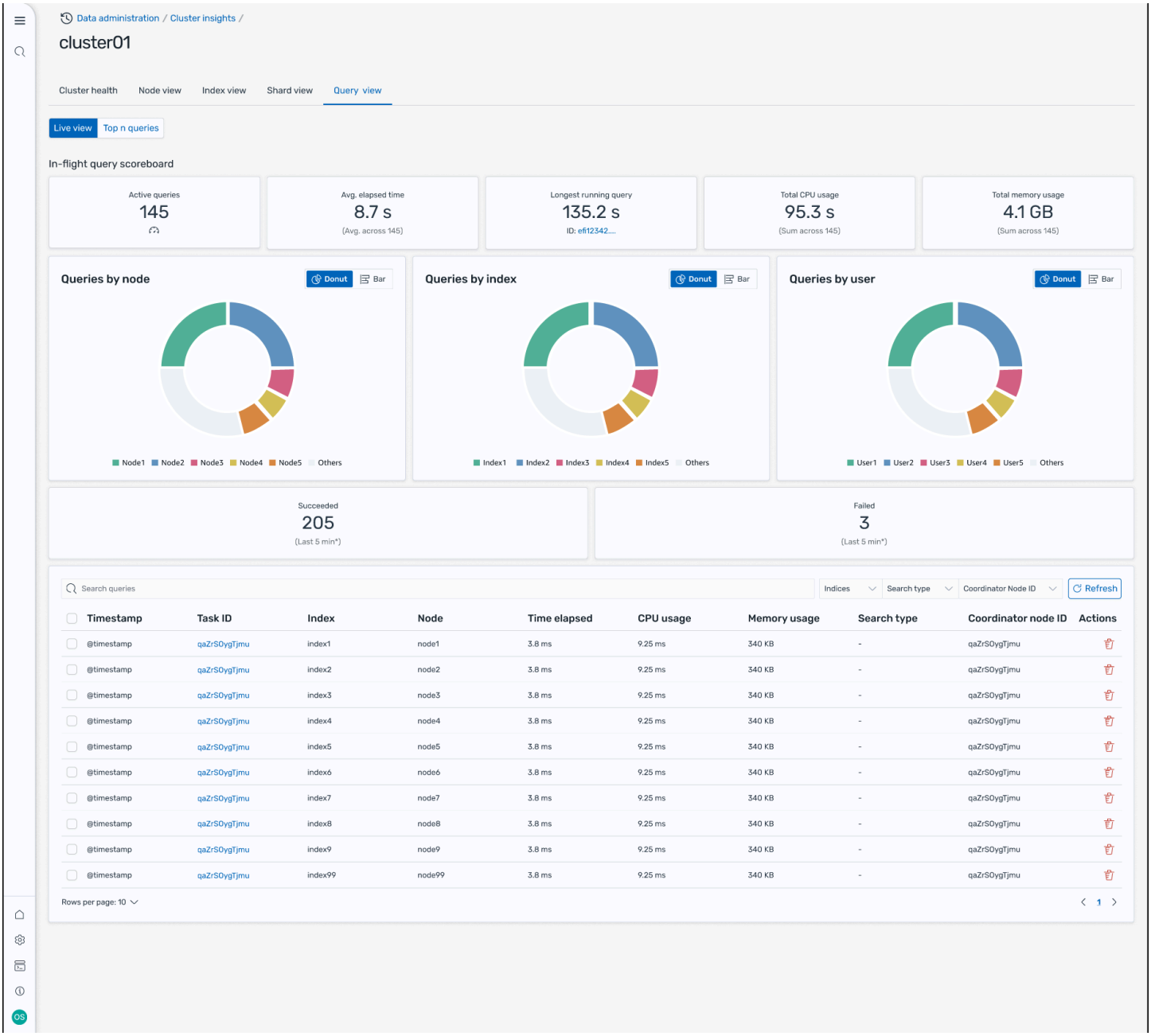
- 查詢計數
- 延遲、CPU 和記憶體用量
- 搜尋類型和協調器節點
- 目標索引和碎片計數

查詢詳細資訊

按兩下任何查詢以查看：

- 確切的查詢承載和執行步驟
- 每個階段的延遲明細（擴展、查詢、擷取）
- 最佳化建議

Screen-8：傳輸中即時檢視。您也可以檢視前 N 個查詢



透過 Amazon EventBridge 事件存取 Insights

您可以透過 Amazon EventBridge 事件監控洞見。如需其他詳細資訊，請檢查通知。

Insights 目錄

Cluster Insights 可在受影響的叢集上提供下列可行的洞見

洞見	Description
低磁碟	當叢集的 FreeStorage 低於總容量的 25% 或 25GB 時，會產生高嚴重性洞見，以先達到閾值為準。
叢集寫入已封鎖	當叢集的 FreeStorage 低於總容量的 20% 或 20GB 時，會產生關鍵嚴重性洞見，以先達到閾值為準。
大碎片	當碎片大小介於 50GB 和 100GB 之間時，會產生嚴重性較低的洞見。當碎片大小介於 100GB 和 200GB 之間時，嚴重性會提升為中，並在碎片大小超過 200GB 時進一步提升為高。
碎片計數	當叢集超過所設定堆積每 GB 的 25 個碎片，或每個節點的碎片數量達到或超過 900 時，就會產生高嚴重性洞見。對於 OpenSearch 2.17 版及更高版本，此洞見將根據叢集管理員組態產生，如本文件所述。
叢集狀態紅色	當替換 OpenSearch 叢集上的降級節點導致紅色叢集狀態時，會產生關鍵嚴重性洞見。這需要立即動作，才能將叢集還原為綠色狀態。
叢集狀態黃色	當替換 OpenSearch 叢集上的降級節點導致黃色叢集狀態時，會產生中等嚴重性洞見。
設定錯誤的複本	當叢集未設定複本，或複本數量超過資料節點總數時，會產生中等嚴重性洞見。
無法存取 KMS 金鑰	當 OpenSearch 叢集無法存取用於靜態加密的 KMS 金鑰時，會產生關鍵嚴重性洞見。這可防止叢集存取加密的資料，並影響叢集操作，需要立即採取動作來還原存取。
錯誤設定的索引設定	當索引設定 "index.routing.allocation.require/include/exclude.<attribute>" 存在於 OpenSearch 叢集中的任何索引時，會產生中等嚴重性洞見。此設定可防止在叢集更新期間成功部署藍綠。
不正確的 Cluster Manager 組態	當 OpenSearch 叢集具有 2 個沒有專用叢集管理員節點的資料節點，或 2 個專用叢集管理員節點時，會產生關鍵嚴重性洞見。這些組態會產生規定人數損失和叢集不穩定的風險。

洞見	Description
次佳碎片策略	(a) 會產生中等嚴重性洞見，指出一或多個索引的碎片計數少於資料節點的 80%。這可能會導致整個叢集的工作負載分佈不佳。 (b) 產生中嚴重性洞見，指出一或多個索引具有比每個節點平均碎片大小多 20% 的資料碎片。這可能會導致整個叢集的工作負載分佈不佳。
叢集超載	(a) 叢集過載的高嚴重性洞見表示叢集中超過 70% 的資料節點在 CPU、記憶體、磁碟 I/O、磁碟輸送量或磁碟使用率方面有更高的使用率。這表示叢集可能接近容量限制。 (b) 重大嚴重性洞見指出，超過 70% 的資料節點在 CPU、記憶體、磁碟 I/O、磁碟輸送量或磁碟使用率中遇到高資源使用率，並觀察到請求調節或拒絕。這需要立即採取動作來擴展叢集。

Amazon OpenSearch Service 中的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 Cloud AWS 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。在 AWS 合規計畫中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon OpenSearch Service 的合規計劃，請參閱合規計劃範圍內的AWS 服務。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件有助於您了解如何在使用 OpenSearch Service 時套用共同責任模型。下列主題將顯示如何設定 OpenSearch Service，以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 OpenSearch Service 資源。

主題

- Amazon OpenSearch Service 中的資料保護
- Amazon OpenSearch Service 中的 Identity and Access Management
- 預防跨服務混淆代理人
- Amazon OpenSearch Service 中的精細存取控制
- Amazon OpenSearch Service 的合規驗證
- Amazon CopenSearFront Service 的復原功能
- Amazon OpenSearch Service 的 JWT 身分驗證和授權
- Amazon OpenSearch Service 中的基礎設施安全性
- OpenSearch Dashboards 的 SAML 身分驗證
- OpenSearch 的 IAM Identity Center 受信任身分傳播支援
- 設定 OpenSearch Dashboards 的 Amazon Cognito 身分驗證
- 使用 Amazon OpenSearch Service 的服務連結角色

Amazon OpenSearch Service 中的資料保護

AWS 共同責任模型適用於 Amazon OpenSearch Service 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 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 共同責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 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 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 OpenSearch Service 或使用主控台 AWS CLI、API 或 AWS SDKs的其他 AWS 服務 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。

Amazon OpenSearch Service 的靜態資料加密

OpenSearch Service 網域提供靜態資料加密，這是一種安全功能，可協助防止未經授權存取您的資料。此功能使用 AWS Key Management Service (AWS KMS) 來存放和管理加密金鑰，並使用 256 位元金鑰 (AES-256) 的進階加密標準演算法來執行加密。如果啟用，此功能會加密網域的以下層面：

- 所有索引 (包括 UltraWarm 儲存中的索引)
- OpenSearch 日誌
- 置換檔案

- 應用程式目錄中的所有其他資料
- 自動快照

當您啟用靜態資料加密時，以下項目不會加密，但您可以採取額外的步驟來保護它們：

- 手動快照：您目前無法使用 AWS KMS 金鑰來加密手動快照。不過，您可以使用伺服器端加密搭配 S3 受管金鑰或 KMS 金鑰來加密您用作快照儲存庫的儲存貯體。如需說明，請參閱the section called "註冊手動快照儲存庫"。
- 慢速日誌和錯誤日誌：如果您發佈日誌並想要加密它們，您可以使用與 OpenSearch Service 網域相同的 AWS KMS 金鑰來加密其 CloudWatch Logs 日誌群組。如需詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中使用 AWS Key Management Service加密 CloudWatch Logs 中的日誌資料。

> **Note**
>
> 如果已在網域上啟用 UltraWarm 或冷儲存，則您無法在現有網域上啟用靜態加密。您必須先停用 UltraWarm 或冷儲存、啟用靜態加密，然後重新啟用 UltraWarm 或冷儲存。如果您想要在 UltraWarm 或冷儲存中保留索引，您必須先將索引移至熱儲存，再停用 UltraWarm 或冷儲存。

OpenSearch Service 僅支援對稱加密 KMS 金鑰，不支援非對稱加密 KMS 金鑰。若要了解如何建立對稱金鑰，請參閱《 AWS Key Management Service 開發人員指南》中的建立 KMS 金鑰。

無論是否啟用靜態加密，所有網域都會使用 AES-256 和 OpenSearch Service 受管密鑰自動加密自訂套件。

許可

若要使用 OpenSearch Service 主控台設定靜態資料的加密，您必須具有 的讀取許可 AWS KMS，例如下列身分型政策：

JSON

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

```
      "Action": [
        "kms:List*",
        "kms:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

如果您想要使用 AWS 擁有金鑰以外的金鑰，您還必須具有為金鑰建立授予的許可。這些許可通常採用以資源為基礎的政策形式，當您建立金鑰時會加以指定。

若您想要讓金鑰保持為 OpenSearch Service 獨有，您可以將 kms:ViaService 條件新增到金鑰政策：

```
"Condition": {
  "StringEquals": {
    "kms:ViaService": "es.us-west-1.amazonaws.com"
  },
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
```

如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的在 AWS KMS中使用金鑰政策。

啟用靜態資料加密

在新網域中進行靜態資料加密需要 OpenSearch 或 Elasticsearch 5.1 版或更高版本。在現有網域上啟用它需要 OpenSearch 或 Elasticsearch 6.7 或更高版本。

啟用靜態資料加密 (主控台)

1. 在 AWS 主控台中開啟網域，然後選擇動作和編輯安全組態。
2. 在 Encryption (加密)，選擇 Enable encryption of data at rest (啟用靜態資料加密)。
3. 選擇要使用的 AWS KMS 金鑰，然後選擇儲存變更。

您也可以透過組態 API 啟用加密。下列請求會啟用現有網域上的靜態資料加密：

```
{
```

```
   "ClusterConfig":{
      "EncryptionAtRestOptions":{
         "Enabled": true,
         "KmsKeyId":"arn:aws:kms:us-east-1:123456789012:alias/my-key"
      }
   }
}
```

禁用或刪除的 KMS 金鑰

如果您停用或刪除用來加密網域的金鑰，網域將無法存取。OpenSearch Service 會傳送通知，通知您其無法存取 KMS 金鑰。立即重新啟用金鑰以存取您的網域。

如果您的金鑰已刪除，OpenSearch Service 團隊無法協助您復原資料。 只會在等待至少七天後 AWS KMS 刪除金鑰。如果您的金鑰仍待刪除，請取消刪除或拍攝網域的手動快照，以避免資料損失。

停用靜態資料加密

在您設定網域以加密靜態資料後，您無法停用設定。相反地，您可以拍攝現有網域的手動快照，建立另一個網域，遷移您的資料和刪除舊的網域。

監控靜態加密資料的網域

加密靜態資料的網域有兩個額外的指標：`KMSKeyError` 和 `KMSKeyInaccessible`。如果網域遇到與您加密金鑰相關的問題，這些指標才會顯示。如需這些指標的完整說明，請參閱the section called “叢集指標”。您可以使用 OpenSearch Service 主控台或 Amazon CloudWatch 主控台來檢視它們。

> **Tip**
>
> 每個指標代表網域的重大問題，因此我們建議您為二者建立 CloudWatch 警示。如需詳細資訊，請參閱the section called “建議 CloudWatch 警示”。

其他考量

- 自動金鑰輪換會保留 AWS KMS 金鑰的屬性，因此輪換不會影響您存取 OpenSearch 資料的能力。已加密的 OpenSearch Service 網域不支援手動金鑰輪換，其中包含建立新金鑰和更新任何舊金鑰的參照。若要進一步了解，請參閱《 AWS Key Management Service 開發人員指南》中的輪換 AWS KMS 金鑰。
- 有些執行個體類型不支援靜態資料的加密。如需詳細資訊，請參閱the section called “支援的執行個體類型”。

- 加密靜態資料的網域對於其自動快照使用不同的儲存庫名稱。如需詳細資訊，請參閱the section called “還原快照”。
- 雖然我們強烈建議啟用靜態加密，但這可能會增加額外的 CPU 負荷和幾毫秒的延遲。不過，大多數使用案例對這些差異並不敏感，且影響程度取決於叢集、用户端和使用設定檔的組態。

Amazon OpenSearch Service 的節點對節點加密

節點對節點加密在 Amazon OpenSearch Service 的預設功能之上提供額外安全層級。

每個 OpenSearch Service 網域 (無論網域是否使用 VPC 存取) 都存在於其自有的專用 VPC 中。此架構可防止潛在的攻擊者攔截 OpenSearch 節點之間的流量，並保護叢集安全。但是根據預設，VPC 內的流量不會加密。節點對節點加密可使該 VPC 內的所有通訊啟用 TLS 1.2 加密。

若您透過 HTTPS 傳送資料到 OpenSearch Service，節點對節點加密會協助確保當 OpenSearch 將您的資料分配 (與重新分配) 到叢集內時能保持加密狀態。若資料透過 HTTP 收到時為未加密狀態，OpenSearch Service 會在它抵達叢集後加密。您可以使用主控台或組態 API AWS CLI，要求透過 HTTPS 到達網域的所有流量。

如果您啟用精細存取控制，則需要節點對節點加密。

啟用節點對節點加密

新網域的節點對節點加密需要任何版本的 OpenSearch 或 Elasticsearch 6.0 或更高版本。在現有網域上啟用節點對節點加密需要任何版本的 OpenSearch 或 Elasticsearch 6.7 或更高版本。選擇 AWS 主控台中現有的網域、Actions (動作)，以及 Edit security configuration (編輯安全組態)。

或者，您可以使用 AWS CLI 或 組態 API。如需詳細資訊，請參閱 AWS CLI 命令參考和 OpenSearch Service API 參考。

停用節點對節點加密

在您設定網域以使用節點對節點加密後，您無法停用設定。相反地，您可以拍攝加密網域的手動快照，建立另一個網域，遷移您的資料和刪除舊的網域。

Amazon OpenSearch Service 中的 Identity and Access Management

Amazon OpenSearch Service 提供數種控制網域存取的方法。本主題涵蓋各種政策類型、它們如何彼此互動，以及如何建立自己的自訂政策。

> **Important**
>
> VPC 支援引入了對 OpenSearch Service 存取控制的一些其他考量。如需詳細資訊，請參閱the section called “關於 VPC 網域上的存取政策”。

政策的類型

OpenSearch Service 支援三種類型的存取政策：

- the section called “資源型政策”
- the section called “身分型政策”
- the section called “以 IP 為基礎的政策”

資源型政策

建立網域時，您可以新增以資源為基礎的政策 (通常稱為網域存取政策)。這些政策指定主體可以對域的子資源執行哪些操作 (跨叢集搜尋除外)。子資源包括 OpenSearch 索引和 API。IAM 中的主體 JSON 政策元素指定允許存取的帳戶、使用者或角色。資源 JSON 政策元素會指定這些委託人可存取的子資源。

例如，以下以資源為基礎的政策向 `test-domain` 上的子資源授予 `test-user` 完整存取權 (`es:*`)：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:*"
      ],
```

```
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

此政策適用兩個重要考量：

- 這些權限只適用於此網域。除非您在其他網域上建立類似的政策，否則 test-user 只能存取 test-domain。
- Resource 元素中的結尾 /* 很重要，並指出以資源為基礎的政策僅適用於網域的子資源，不適用於網域本身。在以資源為基礎的政策中，es:* 動作相當於 es:ESHttp*。

  例如，test-user 可以提出索引請求 (GET https://search-test-domain.us-west-1.es.amazonaws.com/test-index)，但不能更新網域的組態 (POST https://es.us-west-1.amazonaws.com/2021-01-01/opensearch/domain/test-domain/config)。請注意兩個端點之間的差異。存取組態 API 需要身分型政策。

您可以新增萬用字元來指定部分索引名稱。此範例可識別任何以 commerce 開頭的索引：

```
arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce*
```

在此情況下，該萬用字元代表 test-user 可以向 test-domain 中名稱以 commerce 開頭的索引發出請求。

若要進一步限制 test-user，您可以套用以下政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
```

```
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce-
data/_search"
    }
  ]
}
```

現在 test-user 可以執行一項操作：根據 commerce-data 搜尋。其他所有網域內的索引均無法存取，而且無使用 es:ESHttpPut 或 es:ESHttpPost 動作的許可，因此 test-user 無法新增或修改文件。

接著，您可能會決定設定進階使用者角色。這個政策可授予 power-user-role 權限，以利其操作索引中全部 URI 適用的 HTTP GET 和 PUT 方法：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/power-user-role"
        ]
      },
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPut"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce-
data/*"
    }
  ]
}
```

如果您的網域位於 VPC 或使用精細存取控制，您可以使用開放網域存取政策。否則，您的網域存取政策必須包含一些限制 (依委託人或 IP 地址)。

如需有關所有可行動作的詳細資訊，請參閱the section called “政策元素參考”。若要更精細地控制您的資料，請搭配使用開放網域存取政策與精細存取控制。

身分型政策

與屬於每個 OpenSearch Service 網域一部分的資源型政策不同，您可以使用 AWS Identity and Access Management (IAM) 服務將身分型政策連接到使用者或角色。就像以資源為基礎的政策一樣，以身分為基礎的政策會指定誰可以存取服務、可以執行哪些動作，以及可以在哪些資源執行那些動作 (如果適用)。

雖然他們不需要這麼做，但以身分為基礎的政策往往更加通用。它們通常只管理使用者可執行的組態 API 動作。準備好這些政策之後，您可以在 OpenSearch Service 中使用以資源為基礎的政策 (或精細存取控制)，向使用者提供對 OpenSearch 索引和 API 的存取權。

> **Note**
>
> 具有 AWS 受管`AmazonOpenSearchServiceReadOnlyAccess`政策的使用者在主控台上看不到叢集運作狀態。若要允許其查看叢集運作狀態 (以及其他 OpenSearch 資料)，請新增 `es:ESHttpGet` 動作至存取政策並連接至其帳戶或角色。

由於以身分為基礎的政策附加到使用者或角色 (委託人)，JSON 不指定委託人。下列政策授與動作存取權，該動作的開頭是 `Describe` 和 `List`。這組動作提供的唯讀存取權限用於網域組態，而非儲存在網域本身的資料：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:Describe*",
        "es:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

管理員可能擁有 OpenSearch Service 及所有網域中所存放之全部資料的完整存取權：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

身分型政策讓您能使用標籤來控制對組態 API 的存取。例如，如果網域具有 `team:devops` 標籤，下列政策可讓所連接的委託人檢視並更新網域的組態：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Action": [
      "es:UpdateDomainConfig",
      "es:DescribeDomain",
      "es:DescribeDomainConfig"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/team": [
          "devops"
        ]
```

```
      }
    }
  }]
}
```

您也可以使用標籤來控制對 OpenSearch API 的存取。OpenSearch API 的標籤型策略僅適用於 HTTP 方法。例如，如果網域有 `environment:production` 標籤，以下策略可讓連接的主體傳送 GET 和 PUT 請求給 OpenSearch API：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Action": [
      "es:ESHttpGet",
      "es:ESHttpPut"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

若要更精密控制 OpenSearch API，請考慮使用精細存取控制。

> **Note**
>
> 將一或多個 OpenSearch API 新增至任何標籤型政策後，您必須執行單一標籤操作 (例如新增、刪除或修改標籤)，以便變更在網域上生效。您必須使用服務軟體 R20211203 或更新版本，才能在標籤型政策中加入 OpenSearch API 操作。

OpenSearch Service 支援組態 API (而非 OpenSearch API) 的 `RequestTag` 和 `TagKeys` 全域條件索引鍵。這些條件僅適用於在請求中包含標籤的 API 呼叫，例如 `CreateDomain`、`AddTags` 以及 `RemoveTags`。下列政策可讓所連接的委託人建立網域，但前提是他們在請求中包含 `team:it` 標籤：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "es:CreateDomain",
      "es:AddTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/team": [
          "it"
        ]
      }
    }
  }
}
```

如需使用標籤進行存取控制的詳細資訊，以及資源型和身分型政策之間的差異，請參閱《IAM 使用者指南》中的根據具有 ABAC 授權的屬性定義許可。

以 IP 為基礎的政策

以 IP 為基礎的政策，限制存取到一或多個 IP 地址或 CIDR 區塊的網域。從技術層面來看，以 IP 為基礎的政策不是明確的政策類型。反之，它們只是以資源為基礎的政策，指定匿名委託人並包含特殊條件。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

以 IP 為基礎的政策的主要優點是它們允許 OpenSearch Service 網域的未簽署請求，這可讓您使用 curl 和 OpenSearch Dashboards 這類用戶端，或透過代理伺服器存取網域。如需詳細資訊，請參閱 the section called “使用代理伺服器從 Dashboards 中存取 OpenSearch Service”。

> **Note**
>
> 如果為網域啟用 VPC 存取，您無法設定以 IP 為基礎的政策。反之，您可以使用 `security groups` 控制哪些 IP 地址可以存取網域。如需詳細資訊，請參閱下列主題：
>
> - the section called “關於 VPC 網域上的存取政策”
> - 《Amazon VPC 使用者指南》中的使用安全群組控制 AWS 資源的流量

以下政策會將對 `test-domain` 的存取權限授予所有來自指定 IP 範圍的 HTTP 請求：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

如果您的網域具備公有端點且不使用精細存取控制，我們建議將 IAM 委託人與 IP 地址合併。這項政策只有在請求是來自指定 IP 範圍時，才會授予 `test-user` HTTP 存取：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::987654321098:user/test-user"
      ]
    },
    "Action": [
      "es:ESHttp*"
    ],
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24"
        ]
      }
    },
    "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
  }]
}
```

提出並簽署 OpenSearch Service 請求

即使設定完全開放以資源為基礎的存取政策，也必須簽署對 OpenSearch Service 組態 API 的所有請求。如果您的政策指定 IAM 角色或使用者，對 OpenSearch APIs請求也必須使用 AWS Signature 第 4 版簽署。簽署方法因 API 而異：

- 若要呼叫 OpenSearch Service 組態 API，建議您使用其中一個 AWS 開發套件。開發套件已大幅簡化程序，相較於建立和簽署您自己的請求，可為您節省大量時間。組態 API 端點使用下列格式：

  ```
  es.region.amazonaws.com/2021-01-01/
  ```

  例如，下列請求提出對 `movies` 網域進行組態變更，但是您必須自行登入 (不建議)：

  ```
  POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/movies/config
  {
  ```

```
  "ClusterConfig": {
    "InstanceType": "c5.xlarge.search"
  }
}
```

若您使用其中一個開發套件，例如 Boto 3，開發套件會自動處理下列簽署：

```
import boto3

client = boto3.client(es)
response = client.update_domain_config(
  DomainName='movies',
  ClusterConfig={
    'InstanceType': 'c5.xlarge.search'
  }
)
```

如需 Java 程式碼範例，請參閱the section called “使用 AWS SDKs”。

- 若要呼叫 OpenSearch API，您必須簽署您自己的請求。OpenSearch API 採用下列格式：

```
domain-id.region.es.amazonaws.com
```

例如，以下請求將搜尋 thor 的 movies 索引：

```
GET https://my-domain.us-east-1.es.amazonaws.com/movies/_search?q=thor
```

> **Note**
>
> 對於使用 Signature Version 4 簽署的 HTTP POST 請求，此服務會忽略 URL 中傳入的參數。

當政策衝突時

當與政策相違或未明確提及使用者時，事情就變得複雜了。IAM 使用者指南中的 IAM 運作方式提供政策評估邏輯的簡潔摘要：

- 根據預設，所有的請求一律拒絕。
- 明確允許覆寫這個預設值。

- 明確拒絕覆寫任何允許。

例如，假設以資源為基礎的政策允許您存取網域子資源 (OpenSearch 索引或 API)，但以身分為基礎的政策拒絕您存取，您便無法存取。如果以身分為基礎的政策授與存取權，但以資源為基礎的政策未指定您是否有存取權，此時您便可以存取。請參閱下表的相交政策，以了解網域子資源的完整結果摘要。

	以資源為基礎的政策允許	以資源為基礎的政策拒絕	以資源為基礎的政策不允許也不拒絕
Allowed in identity-based policy	允許	Deny	Allow
Denied in identity-based policy	Deny	Deny	Deny
Neither allowed nor denied in identity-based policy	Allow	Deny	Deny

政策元素參考

OpenSearch Service 支援 IAM 政策元素參考中的大多數政策元素，唯獨 `NotPrincipal` 除外。下表顯示最常見的元素。

JSON 政策元素	摘要
`Version`	目前版本的政策語言是 `2012-10-17` 。所有存取政策應該指定這個值。
`Effect`	此元素指定公告內容是否允許或拒絕對指定動作的存取。有效值為 `Allow` 或 `Deny`。
`Principal`	此元素指定允許或拒絕存取資源的 AWS 帳戶 或 IAM 角色或使用者，並且可以採用多種形式： • AWS 帳戶： `"Principal":{"AWS": ["123456789012"]}` 或 `"Principal":{"AWS": ["arn:aws:iam::123456789012:root"]}`

JSON 政策元素	摘要
	• IAM 使用者：`"Principal":{"AWS": ["arn:aws:iam::123456789012:user/test-user"]}` • IAM 角色：`"Principal":{"AWS": ["arn:aws:iam::123456789012:role/test-role"]}`

> **Important**
>
> 指定 * 萬用字元可匿名存取網域，但我們不建議這麼做，除非新增 IP 型條件、使用 VPC 支援，或啟用精細存取控制。此外，請仔細檢查下列政策，以確認它們未授予廣泛存取權：
>
> - 連接至相關聯 AWS 主體 (例如 IAM 角色) 的身分型政策
> - 連接至相關聯 AWS 資源的資源型政策 （ 例如， AWS Key Management Service KMS 金鑰 ）

JSON 政策元素	摘要
Action	OpenSearch Service 針對 OpenSearch HTTP 方法使用 ESHttp* 動作：其餘的動作套用至組態 API。 某些特定 es: 動作支援資源層級的許可。例如，您可以許可使用者刪除一個特定網域，而不是許可使用者刪除任何網域。其他動作只適用於服務本身。例如，es:ListDomainNames 在單一網域的內容細節中不具任何意義，因此需要萬用字元。 如需所有可用動作的清單，以及了解它們是套用至網域子資源 (test-domain/*)、網域組態 (test-domain) 還是僅套用至服務 (*)，請參閱《服務授權參考》中的 Amazon OpenSearch Service 的動作、資源及條件索引鍵 以資源為基礎的政策不同於資源層級的許可。以資源為基礎的政策是附加到網域的完整 JSON 政策。資源層級許可則可讓您將動作限制到特定網域或子資源。在實務層面，您可以將資源層級許可當成選用之資源或身分為基礎的政策的一部分。 由於 es:CreateDomain 的資源層級許可不是直覺式的 - 畢竟為何要許可使用者建立一個已經存在的網域？- 使用萬用字元可為您的網域強制執行簡單的命名機制，例如 "Resource": "arn:aws:es:us-west-1:987654321098:domain/my-team-name-*" 。 當然，您有權利納入一些動作來搭配較無限制性的資源元素，如下所示： JSON `{ "Version":"2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "es:ESHttpGet", "es:DescribeDomain" ], "Resource": "*" }`

JSON 政策元素	摘要
	<pre>] }</pre> 若要進一步了解有關配對動作和資源的詳細資訊，請參閱此表格中的 Resource 元素。

Condition	OpenSearch Service 支援 IAM 使用者指南中 AWS 全域條件內容金鑰中所述的大部分條件。值得注意的例外狀況包括 OpenSearch Service 不支援的aws:PrincipalTag 金鑰。
	
	當設定以 IP 為基礎的政策時，您可指定 IP 地址或 CIDR 區塊當做條件，如下所示：
	
	<pre>"Condition": {
	  "IpAddress": {
	    "aws:SourceIp": [
	      "192.0.2.0/32"
	    ]
	  }
	}</pre>
	正如 the section called “身分型政策” 中所述，aws:ResourceTag 、aws:RequestTag 以及 aws:TagKeys 條件索引鍵適用於組態 API，以及 OpenSearch API。

JSON 政策元素	摘要

Resource	OpenSearch Service 以三種基本方式使用 Resource 元素：
	• 對於適用於 OpenSearch Service 本身的動作，例如 es:ListDomainNames 或是允許完整存取，請使用下列語法：
	`"Resource": "*"`
	• 對於涉及網域組態的動作像是 es:DescribeDomain ，您可以使用以下語法：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name "`
	• 對於適用於網域子資源的動作像是 es:ESHttpGet ，您可以使用以下語法：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /*"`
	您不一定要使用萬用字元。OpenSearch Service 可讓您針對每一個 OpenSearch 索引或 API 定義不同的存取政策。例如，您可以限制使用者的 test-index 索引許可：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index"`
	而不是完整存取 test-index ，因為您可能會希望限制政策為只有搜尋 API：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index/_search"`
	您甚至可以控制存取個別文件：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index/test-type/1"`

JSON 政策元素	摘要
	基本上，如果 OpenSearch 將子資源表示為 URI，您便可以使用存取政策控制對其的存取。如需更進一步控制使用者能夠存取的資源，請參閱 the section called “精細定義存取控制”。 如需有關哪些動作支援資源層級許可的詳細資訊，請參閱此表格中的 Action 元素。

進階選項和 API 考量

OpenSearch Service 有數種進階選項，其中一個可存取控制含意：rest.action.multi.allow_explicit_index。它的預設設定為 true，可讓使用者在特定情況下繞過子資源許可。

例如，請考量以下以資源為基礎的政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/test-index/*",
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/_bulk"
      ]
    },
    {
      "Effect": "Allow",
```

```
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/
restricted-index/*"
    }
  ]
}
```

此政策向 test-user 授予對 test-index 和 OpenSearch 大量 API 的完整存取。它也允許 GET 請求 restricted-index。

以下索引請求因為許可錯誤而失敗：

```
PUT https://search-test-domain.us-west-1.es.amazonaws.com/restricted-index/movie/1
{
  "title": "Your Name",
  "director": "Makoto Shinkai",
  "year": "2016"
}
```

與索引 API 不同的是，大量 API 可讓您在單一呼叫中建立、更新和刪除許多文件。您通常會在請求本文中指定這些操作，而不是在請求 URL。由於 OpenSearch Service 使用 URL 來控制對網域子資源的存取，因此事實上 test-user 可以使用大量 API 來變更 restricted-index。即使使用者缺少索引的 POST 許可，以下請求仍會成功：

```
POST https://search-test-domain.us-west-1.es.amazonaws.com/_bulk
{ "index" : { "_index": "restricted-index", "_type" : "movie", "_id" : "1" } }
{ "title": "Your Name", "director": "Makoto Shinkai", "year": "2016" }
```

在這種情況下，存取政策無法滿足其目的。為了防止使用者繞過這些類型的限制，您可以變更 rest.action.multi.allow_explicit_index 為 false。如果此值為 false，所有對大量、mget、msearch API (其在請求本文中指定索引名稱) 的呼叫，便會停止運作。換言之，呼叫 _bulk 不再運作，但呼叫 test-index/_bulk 則正常運作。第二個端點包含索引名稱，因此您不需要在請求本文中指定一個。

OpenSearch Dashboards 嚴重依賴 mget 和 msearch，因此在此變更後不可能正常運作。若要進行部分補救，您可以保留 `rest.action.multi.allow_explicit_index` 為 true，並且拒絕特定使用者存取一或多個 API。

如需變更此設定的詳細資訊，請參閱the section called “進階叢集設定”。

同樣地，以下以資源為基礎的政策包含兩個細微問題：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    },
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/
restricted-index/*"
    }
  ]
}
```

- 儘管明確拒絕，`test-user` 仍然可以呼叫如 `GET https://search-test-domain.us-west-1.es.amazonaws.com/_all/_search` 和 `GET https://search-test-domain.us-west-1.es.amazonaws.com/*/_search` 以存取 `restricted-index` 中的文件。
- 由於 `Resource` 元素參考 `restricted-index/*`，`test-user` 未獲許可來直接存取索引的文件。不過，使用者有權刪除整個索引。為防止存取和刪除，政策必須指定 `restricted-index*`。

不是廣泛允許和專注於拒絕，最安全的方法是遵循最小特權原則，只授與任務所需的許可。如需控制對個別索引或 OpenSearch 操作的存取詳細資訊，請參閱 the section called “精細定義存取控制”。

> **Important**
>
> 指定 * 萬用字元可匿名存取您的網域。不建議您使用萬用字元 此外，請仔細檢查下列政策，以確認它們未授予廣泛存取權：
>
> - 連接到相關聯 AWS 主體的身分型政策 （ 例如 IAM 角色 ）
> - 連接至相關聯 AWS 資源的資源型政策 （ 例如， AWS Key Management Service KMS 金鑰 ）

設定存取政策

- 如需有關在 OpenSearch Service 中建立或修改以資源為基礎和以 IP 為基礎的政策之說明，請參閱the section called “設定存取政策”。
- 如需在 IAM 中建立或修改身分型政策的說明，請參閱《IAM 使用者指南》中的使用客戶受管政策定義自訂 IAM 許可。

其他範例政策

雖然本章包含許多範例政策， AWS 但存取控制是一個複雜的主題，最好透過範例了解。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM 以身分為基礎的政策範例。

Amazon OpenSearch Service API 許可參考

設定存取控制時，您會撰寫可連接到 IAM 身分的許可政策 （ 以身分為基礎的政策 ）。如需詳細的參考資訊，請參閱《服務授權參考》中的下列主題：

- OpenSearch Service 的動作、資源和條件索引鍵。
- OpenSearch Ingestion 的動作、資源和條件索引鍵。

這項參考包含了可在 IAM 政策中使用哪些 API 操作的相關資訊。它還包含您可以授予許可 AWS 的資源，以及您可以包含用於精細存取控制的條件索引鍵。

您可以在政策的 `Action` 欄位中指定動作、在政策的 `Resource` 欄位中指定資源值，以及在政策的 `Condition` 欄位中指定條件。若要指定 OpenSearch Service 的動作，請使用 `es:`字首，後面接

著 API 操作名稱 （例如，`es:CreateDomain`)。若要指定 OpenSearch Ingestion 的動作，請使用 `osis:`字首，後面接著 API 操作 （例如，`osis:CreatePipeline`)。

AWS Amazon OpenSearch Service 的 受管政策

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住， AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義特定於使用案例的客戶管理政策，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新 AWS 受管政策中定義的許可，則更新會影響政策連接的所有主體身分 （使用者、群組和角色）。 AWS 服務 當新的 啟動或新的 API 操作可用於現有 服務時， AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱 IAM 使用者指南中的 AWS 受管政策。

AmazonOpenSearchDirectQueryGlueCreateAccess

授予 Amazon OpenSearch Service Direct Query Service 對 `CreateDatabase`、`CreatePartition`、`CreateTable`和 的存取權`BatchCreatePartition` AWS Glue API。

您可以在 IAM 主控台中找到 AmazonOpenSearchDirectQueryGlueCreateAccess 政策。

AmazonOpenSearchServiceFullAccess

授予 AWS 帳戶對 OpenSearch Service 組態 API 操作和資源的完整存取權。

您可以在 IAM 主控台中尋找 AmazonOpenSearchServiceFullAccess 政策。

AmazonOpenSearchServiceReadOnlyAccess

授予 所有 OpenSearch Service 資源的唯讀存取權 AWS 帳戶。

您可以在 IAM 主控台中尋找 AmazonOpenSearchServiceReadOnlyAccess 政策。

AmazonOpenSearchServiceRolePolicy

您不得將 `AmazonOpenSearchServiceRolePolicy` 連接到 IAM 實體。此政策會連接到服務連結角色，可讓 OpenSearch Service 存取帳戶資源。如需詳細資訊，請參閱the section called “許可”。

您可以在 IAM 主控台中尋找 AmazonOpenSearchServiceRolePolicy 政策。

AmazonOpenSearchServiceCognitoAccess

提供啟用 Cognito 身分驗證所必需的最低 Amazon Cognito 許可。

您可以在 IAM 主控台中尋找 AmazonOpenSearchServiceCognitoAccess 政策。

AmazonOpenSearchIngestionServiceRolePolicy

您不得將 `AmazonOpenSearchIngestionServiceRolePolicy` 連接到 IAM 實體。此政策會連接到服務連結角色，允許 OpenSearch Ingestion 為擷取管道啟用 VPC 存取、建立標籤，以及將擷取相關 CloudWatch 指標發佈至您的帳戶。如需詳細資訊，請參閱the section called “使用服務連結角色”。

您可以在 IAM 主控台中找到 AmazonOpenSearchIngestionServiceRolePolicy 政策。

OpenSearchIngestionSelfManagedVpcePolicy

您不得將 `OpenSearchIngestionSelfManagedVpcePolicy` 連接到 IAM 實體。此政策會連接到服務連結角色，允許 OpenSearch Ingestion 為擷取管道啟用自我管理 VPC 存取、建立標籤，以及將擷取相關的 CloudWatch 指標發佈至您的帳戶。如需詳細資訊，請參閱the section called “使用服務連結角色”。

您可以在 IAM 主控台中找到 OpenSearchIngestionSelfManagedVpcePolicy 政策。

AmazonOpenSearchIngestionFullAccess

授予 對 OpenSearch Ingestion API 操作和資源的完整存取權 AWS 帳戶。

您可以在 IAM 主控台中找到 AmazonOpenSearchIngestionFullAccess 政策。

AmazonOpenSearchIngestionReadOnlyAccess

授予 所有 OpenSearch Ingestion 資源的唯讀存取權 AWS 帳戶。

您可以在 IAM 主控台中找到 AmazonOpenSearchIngestionReadOnlyAccess 政策。

AmazonOpenSearchServerlessServiceRolePolicy

提供將 OpenSearch Serverless 指標資料傳送至 CloudWatch 所需的最低 Amazon CloudWatch 許可。

您可以在 IAM 主控台中尋找 AmazonOpenSearchServerlessServiceRolePolicy 政策。

AWS 受管政策的 OpenSearch Service 更新

檢視自此服務開始追蹤變更以來OpenSearch Service AWS 受管政策更新的詳細資訊。

變更	描述	Date
已更新 AmazonOpenSearchIngestionServiceRolePolicy	更新提供 OpenSearch Ingestion 許可，以修改 OpenSearch 建立的 VPC 端點，以便在 VPCs 之間共用管道。 如需了解政策 JSON，請參閱 IAM 主控台。	2025 年 8 月 28 日
已更新 AmazonOpenSearchServiceRolePolicy	已將下列陳述式新增至政策。當 Amazon OpenSearch Service 擔任AWSServiceRoleForAmazonOpenSearchService 服務連結角色時，政策中的這個新陳述式可讓 OpenSearch 更新僅由 OpenSearch 管理之任何 AWS IAM Identity Center 應用程式的存取範圍。 `{ "Effect": "Allow", "Action": "sso:PutApplicationAccessScope", "Resource": "arn:aws:sso::*:application/*/*", "Condition": { "StringEquals": {`	2025 年 3 月 31 日

變更	描述	Date
	"aws:Reso urceOrgID": "${aws:Pr incipalOrgID}" } } }	
已更新 AmazonOpenSearchServerlessServiceRolePolicy	已將 Sid AllowAOSSCloudwatchMetrics 新增至政策 AmazonOpenSearchServerlessServiceRolePolicy 。Sid 是做為政策陳述式選用識別符的陳述式 ID。	2024 年 7 月 12 日
已新增 OpenSearchIngestionSelfManagedVpcePolicy	允許 OpenSearch Ingestion 為擷取管道啟用自我管理 VPC 存取、建立標籤，以及將擷取相關 CloudWatch 指標發佈至您的帳戶的新政策。 如需了解政策 JSON，請參閱 IAM 主控台。	2024 年 6 月 12 日
已新增AmazonOpenSearchDirectQueryGlueCreateAccess	授予 Amazon OpenSearch Service Direct Query Service 對 CreateDatabase 、 CreatePartition CreateTable 和 的存取權BatchCreatePartition AWS Glue API。	2024 年 5 月 6 日

變更	描述	Date
更新 AmazonOpenSearchServiceRolePolicy 和 AmazonElasticsearchServiceRolePolicy	新增服務連結角色指派和取消指派 IPv6 地址所需的許可。 已取代的 Elasticsearch 政策也已更新，以確保向後相容性。	2023 年 10 月 18 日
已新增 AmazonOpenSearchIngestionServiceRolePolicy	允許 OpenSearch Ingestion 為擷取管道啟用 VPC 存取、建立標籤，以及將擷取相關 CloudWatch 指標發佈至您帳戶的新政策。 如需了解政策 JSON，請參閱 IAM 主控台。	2023 年 4 月 26 日
已新增 AmazonOpenSearchIngestionFullAccess	新的政策，授予 對 OpenSearch Ingestion API 操作和資源的完整存取權 AWS 帳戶。 如需了解政策 JSON，請參閱 IAM 主控台。	2023 年 4 月 26 日
已新增 AmazonOpenSearchIngestionReadOnlyAccess	授予 所有 OpenSearch Ingestion 資源唯讀存取權的新政策 AWS 帳戶。 如需了解政策 JSON，請參閱 IAM 主控台。	2023 年 4 月 26 日

變更	描述	Date
已新增 `AmazonOpenSearchServerlessServiceRolePolicy`	提供將 OpenSearch Serverless 指標資料傳送至 Amazon CloudWatch所需最低許可的新政策。 如需了解政策 JSON，請參閱 IAM 主控台。	2022 年 11 月 29 日
更新 `AmazonOpenSearchServiceRolePolicy` 和 `AmazonElasticsearchServiceRolePolicy`	已新增服務連結角色建立 OpenSearch Service 受管 VPC 端點所需的許可。某些動作只能在請求包含標籤 `OpenSearchManaged=true` 時執行。 已取代的 Elasticsearch 政策也已更新，以確保向後相容性。	2022 年 11 月 7 日
更新 `AmazonOpenSearchServiceRolePolicy` 和 `AmazonElasticsearchServiceRolePolicy`	已新增對 `PutMetricData` 動作的支援，這是將 OpenSearch 叢集指標發佈至 Amazon CloudWatch 所需的動作。 已取代的 Elasticsearch 政策也已更新，以確保向後相容性。 如需了解政策 JSON，請參閱 IAM 主控台。	2022 年 9 月 12 日

變更	描述	Date
更新 AmazonOpenSearchServiceRolePolicy 和 AmazonElasticsearchServiceRolePolicy	新增對 acm 資源類型的支援。此政策提供服務連結角色驗證和驗證 ACM 資源所需的最低 AWS Certificate Manager (ACM) 唯讀許可，以建立和更新啟用自訂端點的網域。 已取代的 Elasticsearch 政策也已更新，以確保向後相容性。	2022 年 7 月 28 日
更新 AmazonOpenSearchServiceCognitoAccess 和 AmazonESCognitoAccess	新增對 UpdateUserPoolClient 動作的支援，這是在從 Elasticsearch 升級至 OpenSearch 期間設定 Cognito 使用者集區設定所需的動作。 更正 SetIdentityPoolRoles 動作的許可，允許對所有資源的存取。 已取代的 Elasticsearch 政策也已更新，以確保向後相容性。	2021 年 12 月 20 日
已更新 AmazonOpenSearchServiceRolePolicy	新增對 security-group 資源類型的支援。該政策提供服務連結角色所必需的最低 Amazon EC2 和 Elastic Load Balancing 許可，以啟用 VPC 存取。	2021 年 9 月 9 日

變更	描述	Date
• 已新增 AmazonOpenSearchServiceFullAccess • 已取代 AmazonESFullAccess	這項新政策旨在取代舊政策。這兩個原則提供對 OpenSearch Service 組態 API 以及 OpenSearch API 的所有 HTTP 方法的完整存取。精細存取控制和以資源為基礎的政策仍然可以限制存取。	2021 年 9 月 7 日
• 已新增 AmazonOpenSearchServiceReadOnlyAccess • 已取代 AmazonESReadOnlyAccess	這項新政策旨在取代舊政策。這兩個政策提供對 OpenSearch Service 組態 API (es:Describe* 、es:List* 以及 es:Get*) 的唯讀存取，但是不能存取 OpenSearch API 的 HTTP 方法。	2021 年 9 月 7 日
• 已新增 AmazonOpenSearchServiceCognitoAccess • 已取代 AmazonESCognitoAccess	這項新政策旨在取代舊政策。這兩個政策提供啟用 Cognito 身分驗證所必需的最低 Amazon Cognito 許可。	2021 年 9 月 7 日
• 已新增 AmazonOpenSearchServiceRolePolicy • 已取代 AmazonElasticsearchServiceRolePolicy	這項新政策旨在取代舊政策。兩個政策提供服務連結角色所必需的最低 Amazon EC2 和 Elastic Load Balancing 許可，以啟用 VPC 存取。	2021 年 9 月 7 日
開始追蹤變更	Amazon OpenSearch Service 現在會追蹤 AWS受管政策的變更。	2021 年 9 月 7 日

預防跨服務混淆代理人

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在 中 AWS，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況， AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

若要限制 Amazon OpenSearch Service 為資源提供另一項服務的許可，我們推薦在資源政策中使用 aws:SourceArn 和 aws:SourceAccount 全域條件內容索引鍵。如果 aws:SourceArn 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN)，您必須使用這兩個全域條件內容金鑰來限制許可。如果同時使用這兩個全域條件內容金鑰，且 aws:SourceArn 值包含帳戶 ID，則在相同政策陳述式中使用 aws:SourceAccount 值和 aws:SourceArn 值中的帳戶時，必須使用相同的帳戶 ID。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 aws:SourceArn。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 aws:SourceAccount。

aws:SourceArn 的值必須是 OpenSearch Service 網域的 ARN。

防範混淆代理人問題最有效的方法，是使用 aws:SourceArn 全域條件內容金鑰，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 aws:SourceArn 全域條件內容金鑰，同時使用萬用字元 (*) 表示 ARN 的未知部分。例如 arn:aws:*es*:*:*123456789012*:*。

下列範例示範如何使用 OpenSearch Service 中的 aws:SourceArn 和 aws:SourceAccount 全域條件內容金鑰，來預防混淆代理人問題。

JSON

```
{
   "Version":"2012-10-17",
   "Statement":{
      "Sid":"ConfusedDeputyPreventionExamplePolicy",
      "Effect":"Allow",
      "Principal":{
         "Service":"es.amazonaws.com"
      },
      "Action":"sts:AssumeRole",
      "Condition":{
         "StringEquals":{
            "aws:SourceAccount":"111122223333"
```

```
            },
            "ArnLike":{
               "aws:SourceArn":"arn:aws:es:us-east-1:111122223333:domain/my-domain"
            }
         }
      }
}
```

Amazon OpenSearch Service 中的精細存取控制

精細存取控制可提供額外的方式，以控制在 Amazon OpenSearch Service 上對資料的存取。例如，根據提出請求的人員，您可能會希望搜尋只傳回一個索引的結果。您可能會希望隱藏文件中的某些欄位，或是排除特定的文件。

精細存取控制具有以下優勢：

- 角色類型存取控制
- 索引、文件和欄位層級的安全
- OpenSearch Dashboards 多租用
- OpenSearch 和 OpenSearch Dashboards 的 HTTP 基本身分驗證

主題

- 宏觀：精細存取控制和 OpenSearch Service 安全
- 重要概念
- 關於主要使用者
- 啟用精細存取控制
- 以主要使用者身分存取 OpenSearch Dashboards
- 管理許可
- 建議的組態
- 限制
- 修改主要使用者
- 其他主要使用者
- 手動快照

- 整合
- REST API 差異
- 教學課程：使用 IAM 主要使用者和 Amazon Cognito 身分驗證設定網域
- 教學課程：使用內部使用者資料庫和 HTTP 基本身分驗證設定網域

宏觀：精細存取控制和 OpenSearch Service 安全

Amazon OpenSearch Service 安全具有三個主要層次：

網路

第一個安全層次是網路，它決定請求是否可以連線 OpenSearch Service 網域。如果您在建立網域時選擇 Public access (公有存取)，則來自任何網際網路連線用戶端的請求都能連線到網域端點。如果您選擇 VPC access (VPC 存取)，用戶端必須連線至 VPC (且相關聯的安全群組也必須允許)，請求才能連線到端點。如需詳細資訊，請參閱the section called “VPC 支援”。

網域存取政策

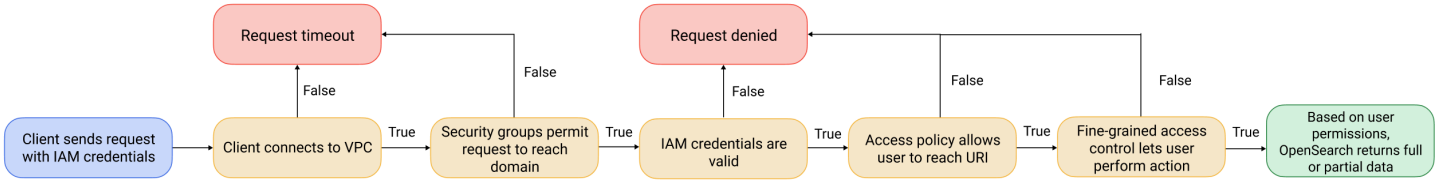
第二個安全層次是網域存取政策。在請求連線到網域端點後，資源類型存取政策會允許或拒絕對指定 URI 的請求存取。存取政策會在請求到達 OpenSearch 本身之前，在網域的「邊緣」接受或拒絕請求。

精細定義存取控制

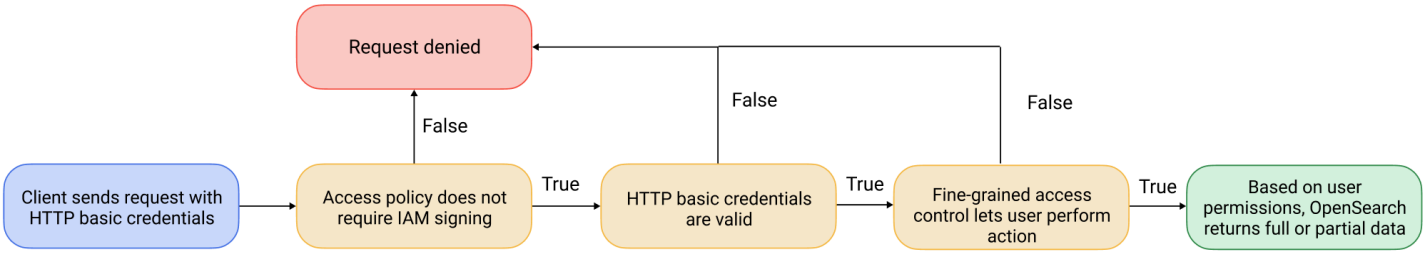
第三個和最後一個安全層次是精細存取控制。在資源類型存取政策允許請求連線到網域端點後，精細存取控制會評估使用者登入資料，並讓使用者通過身分驗證或拒絕請求。如果精細存取控制讓使用者通過身分驗證，則會擷取所有映射到該使用者的角色，並使用完整的許可集合來判斷如何處理請求。

> **Note**
>
> 如果資源型存取政策包含 IAM 角色或使用者，用戶端必須使用 AWS Signature 第 4 版傳送簽署的請求。因此，存取政策可能會和精細存取控制產生衝突，特別是當您使用內部使用者資料庫和 HTTP 基本身分驗證時。您無法使用使用者名稱和密碼和 IAM 登入資料來簽署請求。一般而言，如果您啟用了精細存取控制，我們建議您使用不需要簽章請求的網域存取政策。

下圖說明了常見的組態：啟用精細存取控制的 VPC 存取網域、IAM 型的存取政策以及 IAM 主要使用者。



下圖說明了另一種常見的組態：啟用精細存取控制的公有存取網域，不使用 IAM 委託人的存取政策，以及內部使用者資料庫中的主要使用者。



範例

假設有一個對 `movies/_search?q=thor` 提出的 GET 請求。使用者具備搜尋 `movies` 索引的許可嗎？ 如果具備的話，使用者是否具備查看其中所有文件的許可？ 回應應該要省略或匿名化任何欄位嗎？ 針對主要使用者，回應看起來可能如下：

```
{
  "hits": {
    "total": 7,
    "max_score": 8.772789,
    "hits": [{
        "_index": "movies",
        "_type": "_doc",
        "_id": "tt0800369",
        "_score": 8.772789,
        "_source": {
          "directors": [
            "Kenneth Branagh",
            "Joss Whedon"
          ],
          "release_date": "2011-04-21T00:00:00Z",
          "genres": [
            "Action",
            "Adventure",
            "Fantasy"
```

```
          ],
          "plot": "The powerful but arrogant god Thor is cast out of Asgard to
 live amongst humans in Midgard (Earth), where he soon becomes one of their finest
 defenders.",
          "title": "Thor",
          "actors": [
            "Chris Hemsworth",
            "Anthony Hopkins",
            "Natalie Portman"
          ],
          "year": 2011
        }
      },
      ...
    ]
  }
}
```

如果具備受限性較高許可的使用者提出了完全相同的請求，則回應看起來可能會如下：

```
{
  "hits": {
    "total": 2,
    "max_score": 8.772789,
    "hits": [{
        "_index": "movies",
        "_type": "_doc",
        "_id": "tt0800369",
        "_score": 8.772789,
        "_source": {
          "year": 2011,
          "release_date":
 "3812a72c6dd23eef3c750c2d99e205cbd260389461e19d610406847397ecb357",
          "plot": "The powerful but arrogant god Thor is cast out of Asgard to
 live amongst humans in Midgard (Earth), where he soon becomes one of their finest
 defenders.",
          "title": "Thor"
        }
      },
      ...
    ]
  }
}
```

回應的命中數會較少，且每個命中的欄位數也會比較少。此外，release_date 欄位會進行匿名化。如果沒有具備任何許可的使用者提出相同的請求，則叢集會傳回錯誤：

```
{
  "error": {
    "root_cause": [{
      "type": "security_exception",
      "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
    }],
    "type": "security_exception",
    "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
  },
  "status": 403
}
```

如果使用者提供了無效的登入資料，則叢集會傳回 Unauthorized 例外狀況。

重要概念

當您開始使用精細存取控制時，請考慮下列概念：

- 角色 – 使用精細存取控制的核心方式。在這種情況下，角色與 IAM 角色不同。角色包含了任何許可的組合：全叢集、特定索引、文件層級，以及欄位層級。
- 映射 – 設定角色之後，您會將其映射至一或多個使用者。例如，您可以將三個角色映射到單一使用者：其中一個角色提供存取給 Dashboards，第二個角色提供唯讀存取給 index1，第三個角色則提供寫入存取給 index2。或者，您可以在單一角色中包含這些許可。
- 使用者 – 向 OpenSearch 叢集提出請求的人員或應用程式。使用者擁有 IAM 存取金鑰或使用者名稱和密碼等登入資料，他們在提出請求時指定這些登入資料。

關於主要使用者

OpenSearch Service 中的主要使用者是使用者名稱和密碼組合，或具有基礎 OpenSearch 叢集完整許可的 IAM 主體。如果使用者擁有 OpenSearch 叢集的所有存取權，以及能夠在 OpenSearch Dashboards 中建立內部使用者、角色和角色映射，則會將其視為主要使用者。

在 OpenSearch Service 主控台或透過 CLI 建立的主要使用者會自動對應至兩個預先定義的角色：

- `all_access` – 提供所有叢集整體操作的完整存取權、寫入所有叢集索引的許可，以及寫入所有租用戶的許可。
- `security_manager` – 提供安全外掛程式的存取權，以及使用者和許可的管理。

透過這兩個角色，使用者可以存取 OpenSearch Dashboards 中的安全索引標籤，在其中管理使用者和許可。如果您建立另一個內部使用者，並且只將其映射到`all_access`角色，則該使用者無法存取安全索引標籤。您可以明確地將其他主要使用者映射到 `all_access`和 `security_manager`角色，以建立其他主要使用者。如需說明，請參閱the section called “其他主要使用者”。

當您為網域建立主要使用者時，您可以指定現有的 IAM 主體，或在內部使用者資料庫中建立主要使用者。在決定要使用哪個時，請考慮下列事項：

- IAM 主體 – 如果您為主要使用者選擇 IAM 主體，必須使用 AWS Signature 第 4 版簽署對叢集的所有請求。

  OpenSearch Service 不會考慮 IAM 主體的任何許可。IAM 使用者或角色僅供身分驗證使用。該使用者或角色上的政策與主要使用者的授權無關。授權是透過 OpenSearch Security 外掛程式中的各種許可來處理。

  例如，您可以將零 IAM 許可指派給 IAM 主體，只要機器或人員可以向該使用者或角色進行身分驗證，他們就可以在 OpenSearch Service 中擁有主要使用者的能力。

  如果您希望在多個叢集上使用相同的使用者、希望使用 Amazon Cognito 存取 Dashboards，或是您有支援 Signature 4 版簽署的 OpenSearch 用戶端，則我們建議您使用 IAM。
- 內部使用者資料庫 – 如果您在內部使用者資料庫中建立主伺服器（使用使用者名稱和密碼組合），您可以使用 HTTP 基本身分驗證（以及 IAM 憑證）向叢集提出請求。大多數用戶端都支援基本身分驗證，包括 curl，這也支援 AWS Signature 第 4 版搭配 --aws-sigv4 選項。內部使用者資料庫會存放在 OpenSearch 索引中，因此您無法與其他叢集共享。

  如果您不需要跨多個叢集重複使用使用者、希望使用 HTTP 基本身分驗證來存取 Dashboards (而非 Amazon Cognito)，或是您具備只支援基本身分驗證的用戶端，則我們建議您使用內部使用者資料庫。內部使用者資料庫是開始使用 OpenSearch Service 的最簡單方式。

啟用精細存取控制

使用主控台 AWS CLI或組態 API 啟用精細存取控制。如需這些步驟，請參閱 建立和管理網域。

精細存取控制需要 OpenSearch 或 Elasticsearch 6.7 或更高版本。亦要求所有流向網域的流量皆使用 HTTPS、靜態資料加密，以及節點對節點加密。視您設定精細存取控制進階功能的方式而定，額外處理請求可能需要個別資料節點上的運算和記憶體資源。在您啟用精細存取控制後，您便無法停用此功能。

在現有網域上啟用精細存取控制

您可以在執行 OpenSearch 或 Elasticsearch 6.7 或更高版本的現有網域上，啟用精細存取控制。

在現有網域上啟用精細存取控制 (主控台)

1. 選取網域，並選擇 Actions (動作) 和 Edit security configuration (編輯安全組態)。
2. 選取 Enable fine-grained access control (啟用精細存取控制)。
3. 選擇建立主要使用者的方法：
   - 如果您希望使用 IAM 進行使用者管理，請選擇 Set IAM ARN as master user (將 IAM ARN 設為主要使用者)，並指定 IAM 角色的 ARN。
   - 如果您想要使用內部使用者資料庫，請選擇建立主要使用者，並指定使用者名稱和密碼。
4. (選用) 選取 Enable migration period for open/IP-based access policy (啟用開放/IP 型存取政策的遷移期)。此設定會啟用 30 天的過渡期，在此期間，您目前的使用者可繼續存取網域，不會中斷，且現有的開放和 IP 型存取政策仍可繼續使用您的網域。在此遷移期間，我們建議管理員為網域建立必要角色，並將其映射至使用者。如果您使用以身分為基礎的政策，而非開放或 IP 型存取政策，則您可以停用此設定。

   您也需要更新客戶端，以在遷移期間使用精細存取控制。例如，如果您對應具有精細存取控制的 IAM 角色，則必須更新用戶端以使用 AWS Signature 第 4 版開始簽署請求。如果您使用精細存取控制設定 HTTP 基本身分驗證，則必須更新客戶端，以在請求中提供適當的基本身分驗證憑證。

   在遷移期間，存取網域的 OpenSearch Dashboards 端點的使用者將直接登錄至 Discover (發現) 頁面，而非登入頁面。管理員和主要使用者可選擇 Login (登入)，使用管理員憑據登入並設定角色映射。

   > **Important**
   >
   > OpenSearch Service 會在 30 天後自動停用遷移期。我們建議您在建立必要角色並將其映射至使用者後，立即結束該角色。遷移期結束後，您便無法重新啟用。

5. 選擇儲存變更。

在叢集運作狀態變成紅色期間，變更會觸發藍/綠部署，但所有叢集操作皆不受影響。

在現有網域上啟用精細存取控制 (CLI)

將 AnonymousAuthEnabled 設定為 true，以使用精細存取控制來啟用遷移期：

```
aws opensearch update-domain-config --domain-name test-domain --region us-east-1 \
      --advanced-security-options '{ "Enabled": true,
 "InternalUserDatabaseEnabled":true, "MasterUserOptions": {"MasterUserName":"master-
username","MasterUserPassword":"master-password"},"AnonymousAuthEnabled": true}'
```

關於 default_role

精細存取控制需要角色映射。如果網域使用身分型存取政策，則 OpenSearch Service 會自動將使用者映射至名為 default_role 的新角色，以協助您正確遷移現有使用者。在您建立自有角色映射之前，此臨時映射可確保您的使用者仍可成功傳送由 IAM 簽署之 GET 和 PUT 請求。

該角色不會新增任何安全漏洞或缺陷至您的 OpenSearch Service 網域。我們建議您在設定自有角色後並相應映射它們後，立即刪除預設角色。

遷移案例

下表說明在現有網域上啟用精細存取控制前後的各身分驗證方法的行為，以及若管理員要將其使用者正確映射至角色，必須採取的步驟：

身分驗證方法	在啟用精細存取控制前	在啟用精細存取控制後	管理員任務
以身分為基礎的政策	滿足 IAM 政策的所有使用者都可以存取該網域。	您無需啟用遷移期。 OpenSearch Service 會自動將所有滿足 IAM 政策的 IAM 使用者映射至 default_role，以便他們能夠繼續存取該網域。	1. 在網域上建立自訂角色映射。 2. 刪除 default_role。
以 IP 為基礎的政策	來自受允許 IP 地址或 CIDR 區塊的	在 30 天遷移期內，所有來自受允許 IP 地址	1. 在網域上建立自訂角色映射。

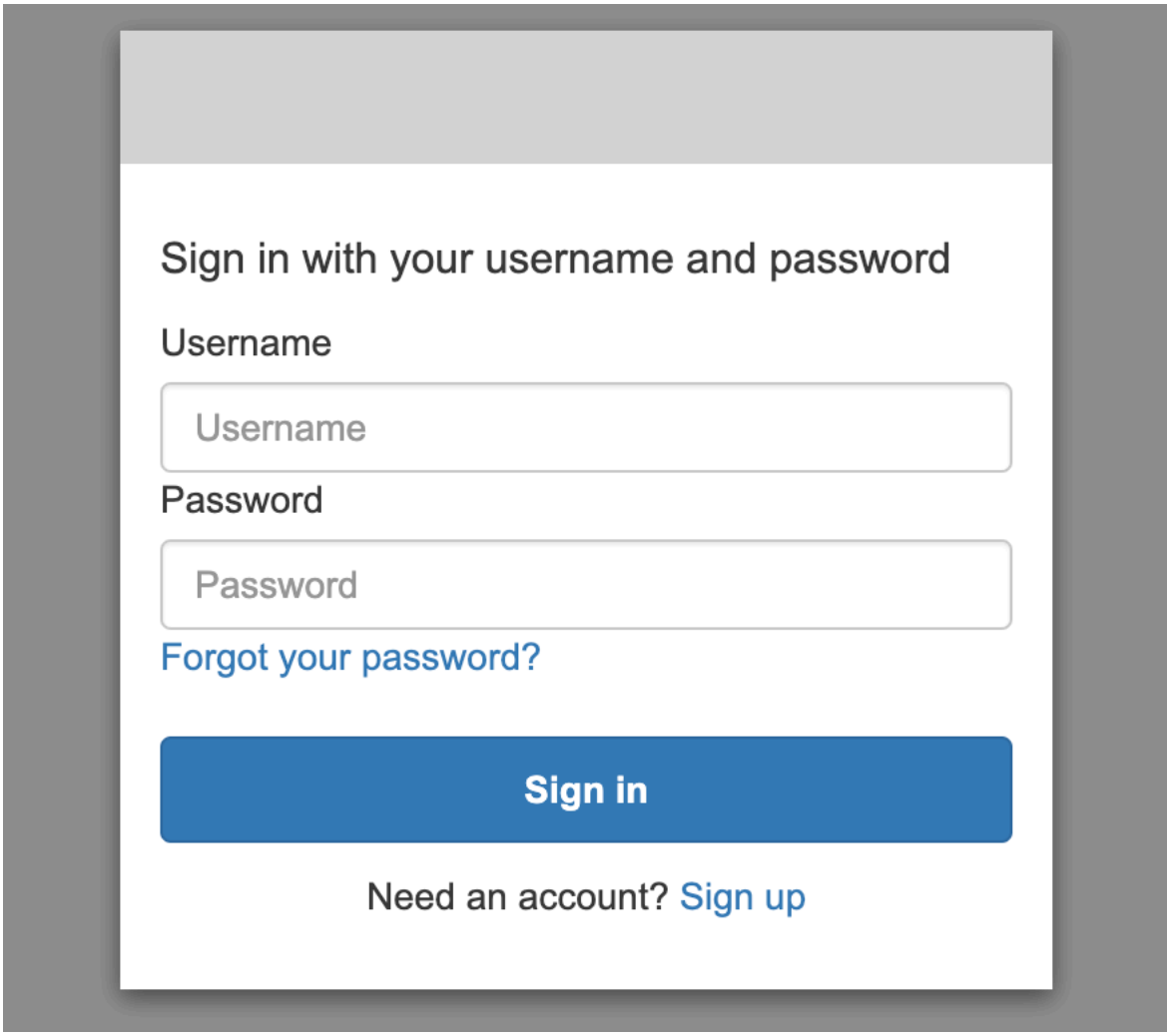
身分驗證方法	在啟用精細存取控制前	在啟用精細存取控制後	管理員任務
	所有使用者皆可存取網域。	或 CIDR 區塊的使用者都可繼續存取網域。	2. 依據您的角色映射組態，更新您的客戶端以提供基本身分驗證憑證或 IAM 憑證。 3. 停用遷移期。若來自受允許的 IP 地址或 CIDR 區塊的使用者，在沒有基本身分驗證或 IAM 憑證的情況下傳送請求，將失去網域的存取權限。
開放存取政策	網際網路上的所有使用者皆可存取網域。	在 30 天的遷移期內，網際網路上的所有使用者皆可繼續存取網域。	1. 在網域上建立角色映射。 2. 依據您的角色映射組態，更新您的客戶端以提供基本身分驗證憑證或 IAM 憑證。 3. 停用遷移期。使用者在沒有基本身分驗證或 IAM 憑證的情況下傳送請求，將失去網域的存取權限。

以主要使用者身分存取 OpenSearch Dashboards

精細存取控制具備 OpenSearch Dashboards 外掛程式，可以簡化管理任務。您可以使用 Dashboards 來管理使用者、角色、映射、動作群組和租用戶。但是 OpenSearch Dashboards 登入頁面和基礎身分驗證方法會有所不同，這取決於您管理使用者及設定網域的方式。

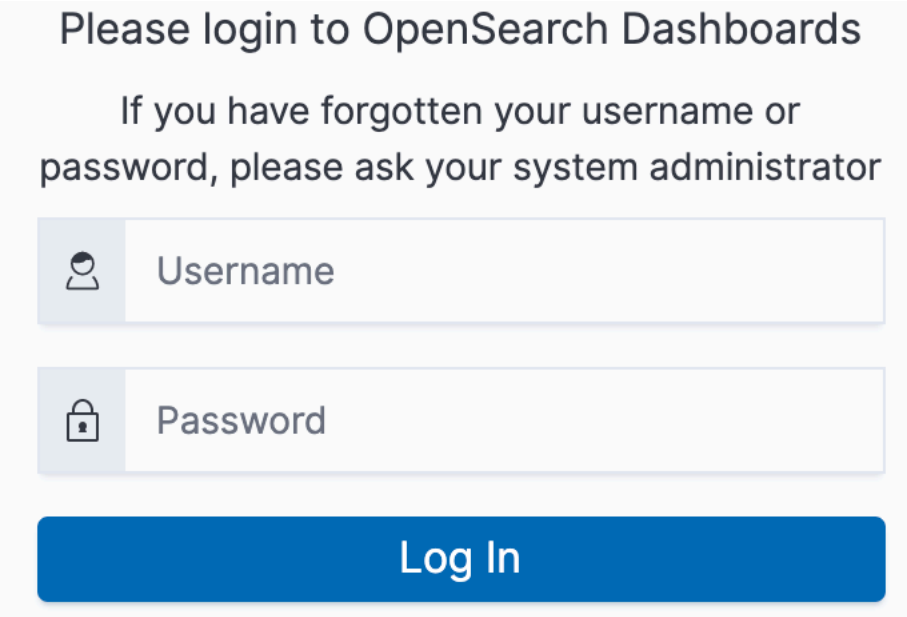
- 如果您想使用 IAM 進行使用者管理，請使用 the section called “OpenSearch Dashboards 的 Amazon Cognito 身分驗證” 來存取 Dashboards。否則，Dashboards 會顯示沒有任何功能的登入頁面。請參閱 the section called “限制”。

  透過 Amazon Cognito 身分驗證，擔任的身分集區中的其中一個角色必須與您為主要使用者指定的 IAM 角色相符。如需此組態的詳細資訊，請參閱 the section called “(選用) 設定精細分級的存取” 和 the section called “教學課程：使用 Cognito 身分驗證進行精細存取控制”。



- 如果您選擇使用內部使用者資料庫，您可以使用主要使用者名稱和密碼登入 Dashboards。您必須透過 HTTPS 存取 Dashboards。適用於 Dashboards 的 Amazon Cognito 和 SAML 身分驗證都會取代此登入畫面。

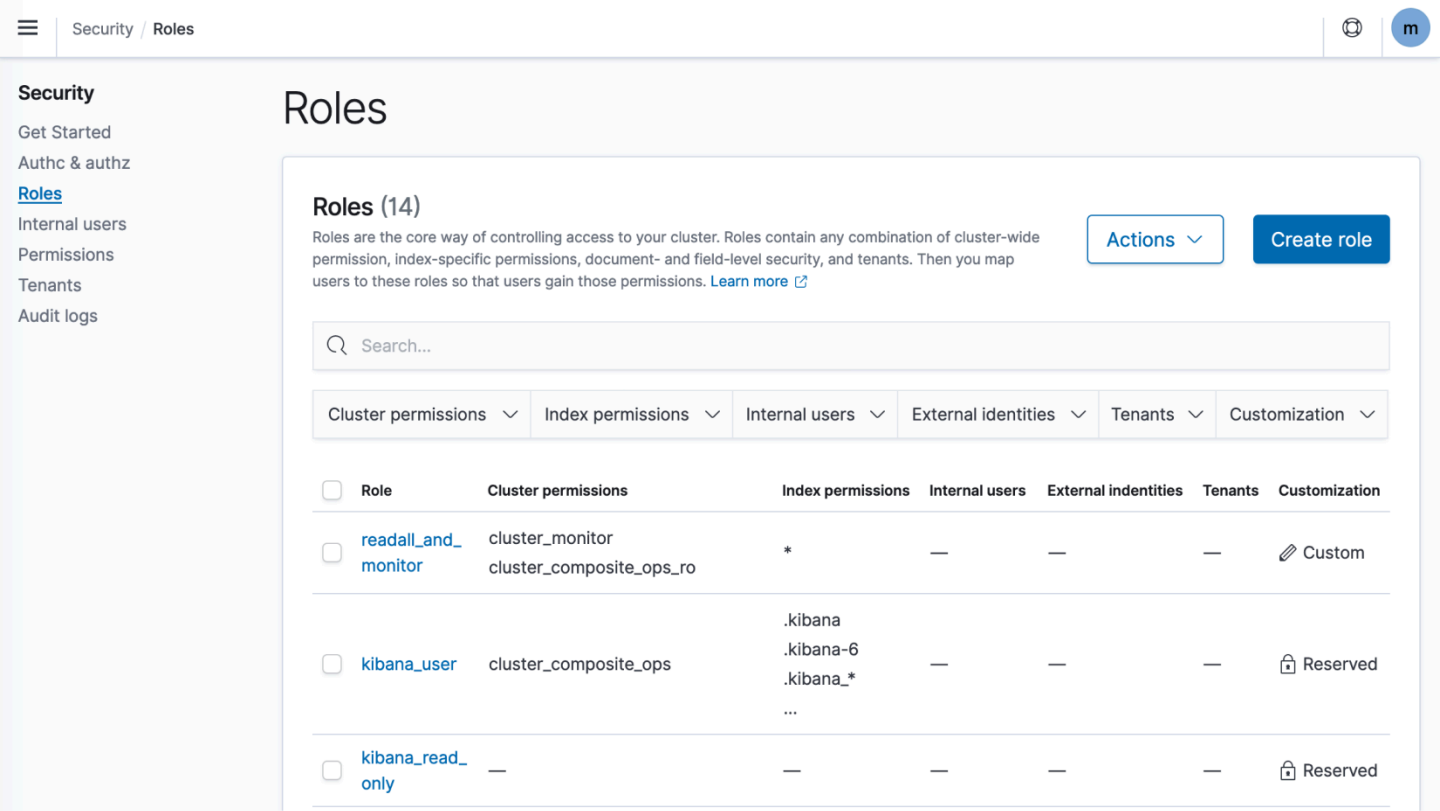
  如需此組態的詳細資訊，請參閱「the section called “教學課程：內部使用者資料庫和基本身分驗證”」。



- 如果您選擇使用 SAML 身分驗證，您可以使用外部身分提供者的憑證登入。如需詳細資訊，請參閱the section called “OpenSearch Dashboards 的 SAML 身分驗證”。

管理許可

如 the section called “重要概念” 中所述，您可以使用角色、使用者和映射來管理精細存取控制許可。本節說明如何建立和套用這些資源。我們建議您以主要使用者身分登入 Dashboards，以執行這些操作。



> **Note**
>
> 您選擇授予使用者的許可，根據使用案例有很大差異。我們無法在本文中涵蓋所有案例。在決定要授予使用者的許可時，請務必參考下列章節中提到的 OpenSearch 叢集和索引許可，並始終遵循最低權限原則。

建立角色

您可以使用 OpenSearch Dashboards 或 REST API 中的 `_plugins/_security` 操作為精細存取控制建立新角色。如需詳細資訊，請參閱建立角色。

精細存取控制也包含了許多預先定義角色。OpenSearch Dashboards 和 Logstash 等用戶端會向 OpenSearch 提出各種請求，使手動建立具備最小許可集合的角色變得難以進行。例如，`opensearch_dashboards_user` 角色包含了使用者使用索引模式、視覺化及儀表板和租用戶所需要的許可。我們建議將其映射至存取 Dashboards 的任何使用者或後端角色，以及允許存取其他索引的其他角色。

Amazon OpenSearch Service 不提供下列 OpenSearch 角色：

- observability_full_access
- observability_read_access
- reports_read_access
- reports_full_access

Amazon OpenSearch Service 提供多個無法與 OpenSearch 搭配使用的角色：

- ultrawarm_manager
- ml_full_access
- cold_manager
- notifications_full_access
- notifications_read_access

叢集層級安全

叢集層級許可包括能夠發出廣泛請求 (例如 _mget、_msearch 以及 _bulk)、監控運作狀態、擷取快照等。請在建立角色時使用 Cluster Permissions (叢集許可) 部分來管理這些許可。如需完整的叢集層級許可清單，請參閱叢集許可。

您通常可以使用預設動作群組組合，而不是個別許可，達到所需的安全狀態。如需叢集層級動作群組的清單，請參閱叢集層級。

索引層級安全

索引層級許可包含建立新索引、搜尋索引、讀取和寫入文件、刪除文件、管理別名等能力。請在建立角色時使用 Index Permissions (索引許可) 部分來管理這些許可。如需完整的索引層級許可清單，請參閱索引許可。

您通常可以使用預設動作群組組合，而不是個別許可，達到所需的安全狀態。如需索引層級動作群組的清單，請參閱索引層級。

文件層級安全

文件層級安全可讓您限制使用者在索引中可看見的文件。請在建立角色時指定索引模式和 OpenSearch 查詢。任何您映射到該角色的使用者都只能看見與查詢相符的文件。文件層級安全會影響您在搜尋時的命中數。

如需詳細資訊，請參閱文件層級安全。

欄位層級安全

欄位層級安全可讓您控制使用者能看見的文件欄位。建立角色時，請新增欄位清單來加入或排除。如果您加入欄位，則您映射到該角色的任何使用者都只能看到那些欄位。如果您排除欄位，則「除了」遭排除的欄位外，使用者可以看見所有欄位。欄位層級安全會影響您在搜尋時包含在命中中的欄位數。

如需詳細資訊，請參閱欄位層級安全。

欄位遮罩

欄位遮罩是欄位層級安全的替代項目，可讓您匿名化欄位中的資料，而非完全移除。請在建立角色時，新增要進行遮罩的欄位清單。欄位遮罩會影響您在搜尋時是否可以看見欄位的內容。

> **Tip**
>
> 如果您將標準遮罩套用至欄位，OpenSearch Service 會使用安全的隨機雜湊，造成彙總結果不精確。若要在遮罩欄位上執行彙總，請改用以模式為基礎的遮罩。

建立 使用者

如果您啟用內部使用者資料庫，則您可以使用 OpenSearch Dashboards 或 REST API 中的 `_plugins/_security` 操作來建立使用者。如需詳細資訊，請參閱建立使用者。

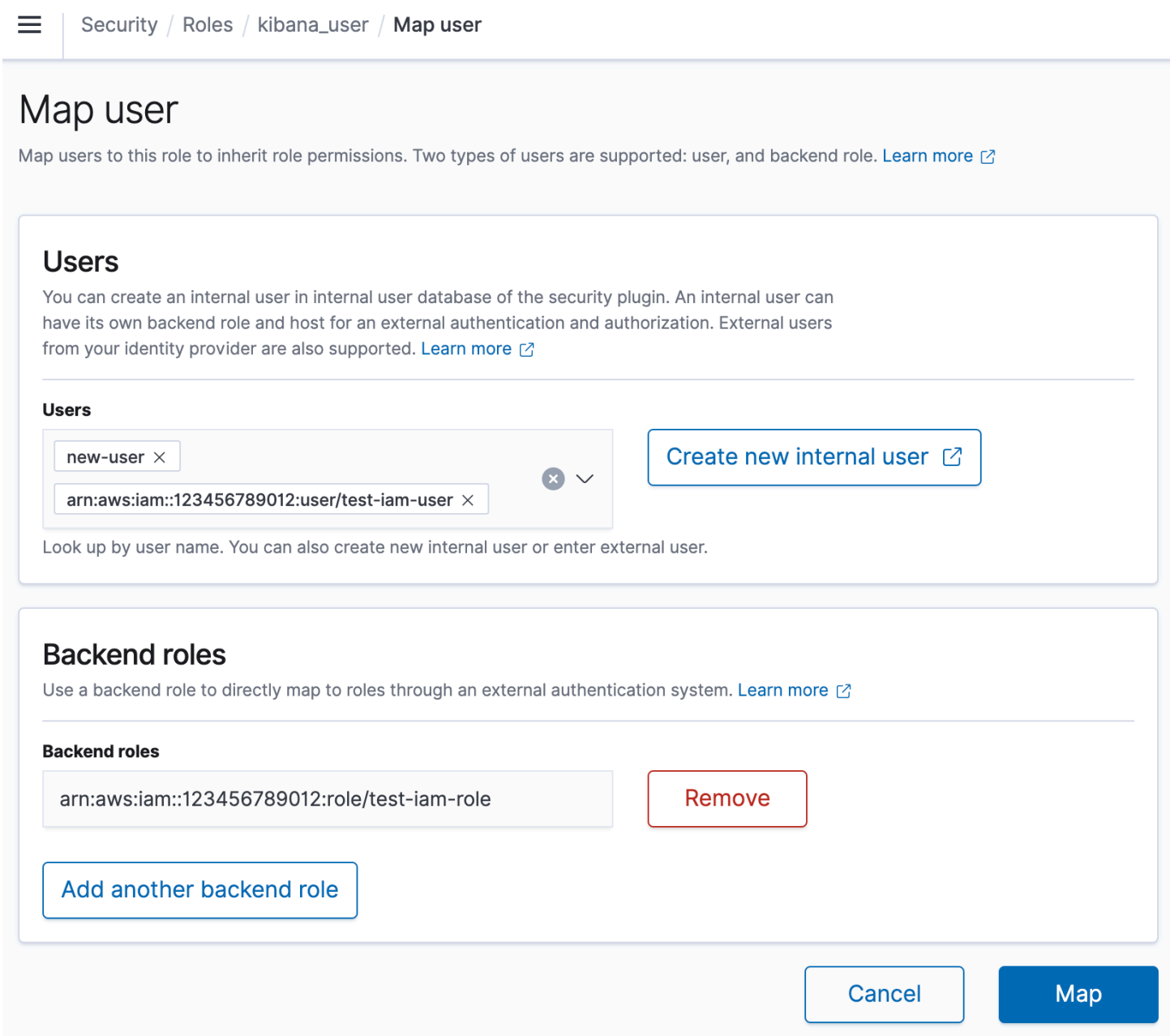
如果您為主要使用者選擇了 IAM，請忽略 Dashboards 的這個部分。請改為建立 IAM 角色。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/。

將角色映射至使用者

角色映射是精細存取控制中最重要的層面。精細存取控制包含了一些預先定義角色，可協助您開始使用，但除非您將這些角色映射到使用者，否則每個向叢集提出的請求都會導致許可錯誤。

後端角色可協助簡化角色映射程序。您可以將角色映射到所有 100 個使用者共用的單一後端角色，而不是將相同的角色映射到 100 個個別使用者。後端角色可以是 IAM 角色或任意字串。

- 在 Users (使用者) 區段中指定使用者、使用者 ARN 和 Amazon Cognito 使用者字串。Cognito 使用者字串的形式為 `Cognito/`*`user-pool-id`*`/`*`username`*。
- 請在 Backend roles (後端角色) 區段中指定後端角色和 IAM 角色 ARN。



您可以使用 OpenSearch Dashboards 或 REST API 中的 `_plugins/_security` 操作將角色映射至使用者。如需詳細資訊，請參閱將使用者映射至角色。

建立動作群組

動作群組是一組許可，可讓您跨不同資源重複使用。您可以使用 OpenSearch Dashboards 或 REST API 中的 `_plugins/_security` 操作建立新的動作群組，雖然預設動作群組已足以適用大多數的使用案例。如需預設動作群組的詳細資訊，請參閱預設動作群組。

OpenSearch Dashboards 多租用

租用戶是儲存索引模式、視覺化、儀表板和其他 Dashboards 物件的空間。Dashboards 多租用戶可讓您安全地與其他 Dashboards 使用者共用您的工作（或保持私有），並動態設定租用戶。您可以控制哪些角色可以存取租用戶，以及這些角色是否具備讀取或寫入存取權限。全域租用戶為預設值。如需進一步了解，請參閱 OpenSearch Dashboards 多租用。

檢視您目前的租用戶或變更租用戶

1. 導覽至 OpenSearch Dashboards 並登入。
2. 選取右上角的使用者圖示，然後選擇 Switch tenants (轉換租用戶)。
3. 在建立視覺效果或儀表板之前驗證您的租用戶。如果您希望與其他所有的 Dashboards 使用者共享您的作品，請選擇 Global (全域)。若要與一部分 Dashboards 使用者共享您的作品，請選擇不同的共享租用戶。否則，請選擇 Private (私有)。

> **Note**
>
> OpenSearch Dashboards 為每個租用戶維護另外的索引，並建立索引範本 `tenant_template`。請勿刪除或修改 `tenant_template` 索引，因為如果租用戶索引映射錯誤設定，可能會導致 OpenSearch Dashboards 故障。

建議的組態

由於精細存取控制與其他安全功能的互動方式，我們建議使用數種精細存取控制組態。這些組態適合大多數的使用案例。

Description	主要使用者	網域存取政策
使用 IAM 憑證來呼叫 OpenSearch API，並使用 SAML 身分驗證存取 Dashboards。使用 Dashboards 或 REST API 管理精細存取控制角色。	IAM 角色或使用者	JSON `{ "Version":"2012-10-17", "Statement": [ { "Effect": "Allow",`

Description	主要使用者	網域存取政策
		"Principal": { "AWS": "*" }, "Action": "es:ESHttp*", "Resource": "arn:aws:es: *us-east-1* :*111122223333* :domain/*domain-name* /*" }] }
使用 IAM 憑證或基本身分驗證來呼叫 OpenSearch API。使用 Dashboards 或 REST API 管理精細存取控制角色。 此組態提供了很多靈活性，尤其是如果您的 OpenSearch 用戶端僅支援基本身分驗證。 如果您擁有現有的身分提供者，請使用 SAML 身分驗證來存取 Dashboards。否則，請管理內部使用者資料庫中的 Dashboards 使用者。	使用者名稱和密碼	JSON { "Version":"2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": "es:ESHttp*", "Resource": "arn:aws:es: *us-east-1* :*111122223333* :domain/*domain-name* /*" }] }

Description	主要使用者	網域存取政策

使用 IAM 憑證來呼叫 OpenSearch API，並使用 Amazon Cognito 存取 Dashboards。使用 Dashboards 或 REST API 管理精細存取控制角色。	IAM 角色或使用者	JSON
		{
		"Version":"2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": "arn:aws:es: *us-east-1* :*111122223333* :domain/*domain-name* /*"
		}
		]
		}

Description	主要使用者	網域存取政策

使用 IAM 憑證來呼叫 OpenSearch API，並阻止對 Dashboards 的大多數存取。使用 REST API 管理精細存取控制角色。	IAM 角色或使用者	JSON
		{
		"Version":"2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": "arn:aws:es: *us-east-1* :*111122223333* :domain/*domain-name* /*"
		},
		{
		"Effect": "Deny",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": "arn:aws:es: *us-east-1* :*111122223333* :domain/*domain-name* /_dashboards*"
		}
		]
		}

限制

精細存取控制有幾個重要的限制：

- 如果網域是位於 VPC 中，則將角色映射到主機名稱或 IP 地址的角色映射 `hosts` 層面會無法正常運作。您仍然可以將角色映射到使用者和後端角色。
- 如果您為主要使用者選擇了 IAM 但並未啟用 Amazon Cognito 或 SAML 身分驗證，則 Dashboards 會顯示沒有任何功能的登入頁面。

- 如果您為主要使用者選擇了 IAM，您仍然可以在內部使用者資料庫中建立使用者。但是，由於沒有在這個組態下啟用 HTTP 基本身分驗證，因此任何使用這些使用者登入資料簽署的請求都會遭到拒絕。
- 如果您使用 SQL 來查詢您無法存取的索引，便會收到「沒有許可」錯誤。如果索引不存在，則您會收到「找不到索引」錯誤。這項錯誤訊息中的差異表示如果您猜測其名稱，您便可以確認該索引是否存在。

  為了將問題降至最低，請不要在索引名稱中包含敏感資訊。如要拒絕所有對 SQL 的存取，請將以下元素新增到您的網域存取政策：

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": [
      "*"
    ]
  },
  "Action": [
    "es:*"
  ],
  "Resource": "arn:aws:es:us-east-1:123456789012:domain/my-domain/_plugins/_sql"
}
```

- 如果您的網域版本是 2.3 或更高版本，且您已啟用精細存取控制，則將 `max_clause_count`設定為 1 會導致網域發生問題。我們建議將此帳戶設定為較高的數字。
- 如果您在未設定精細存取控制的網域中啟用精細存取控制，對於為直接查詢建立的資料來源，您需要自行設定精細存取控制角色。如需如何設定精細存取角色的詳細資訊，請參閱建立與 Amazon S3 的 Amazon OpenSearch Service 資料來源整合。

修改主要使用者

如果您忘記了主要使用者的詳細資訊，您可以使用主控台、 AWS CLI或組態 API 來重新設定。

修改主要使用者 (主控台)

1. 導覽至 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home/。
2. 選取網域，並選擇 Actions (動作)、Edit security configuration (編輯安全組態)。

3. 選擇 Set IAM ARN as master user (將 IAM ARN 設為主要使用者) 或 Create master user (建立主要使用者)。

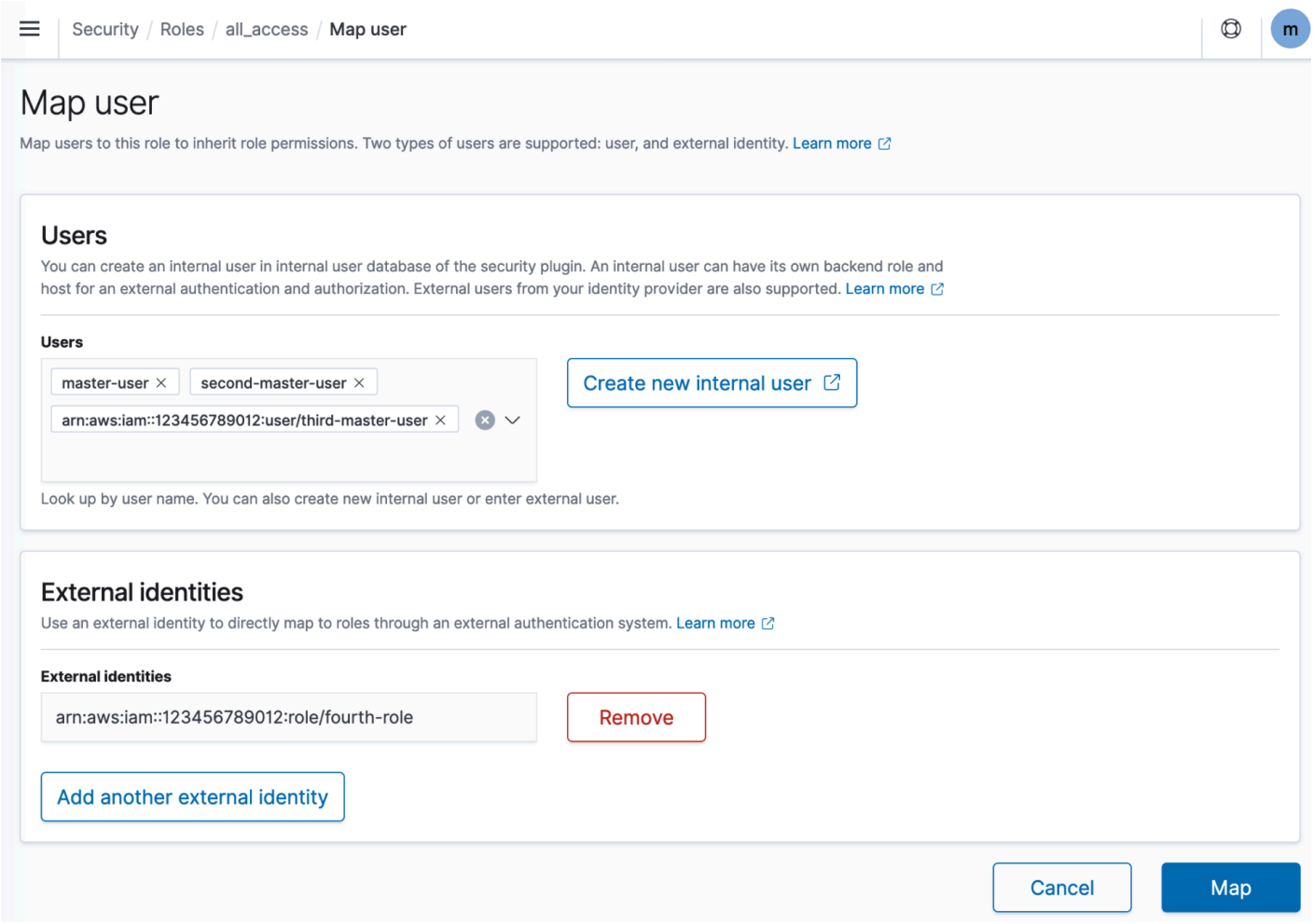
   - 如果您先前使用了 IAM 主要使用者，精細存取控制會將 `all_access` 角色重新映射到您指定的新 IAM ARN。
   - 如果您先前使用了內部使用者資料庫，則精細存取控制會建立新的主要使用者。您可以使用新的主要使用者來刪除舊的主要使用者。
   - 從內部使用者資料庫切換至 IAM 主要使用者不會刪除內部使用者資料庫中的任何使用者。相反，它只是停用 HTTP 基本身分驗證。從內部使用者資料庫手動刪除使用者，或者保留它們，以防您需要重新啟用 HTTP 基本身分驗證。

4. 選擇儲存變更。

其他主要使用者

您可以在建立網域時指定主要使用者，但是如果您希望的話，您可以使用這個主要使用者來建立其他主要使用者。您有兩個選項：OpenSearch Dashboards 或 REST API。

- 在 Dashboards 中，選擇 Security (安全性)、Role (角色)，然後將新的主要使用者映射到 `all_access` 和 `security_manager` 角色。



- 如要使用 REST API，請傳送下列請求：

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

```
PUT _plugins/_security/api/rolesmapping/security_manager
{
```

```
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

這些請求會「取代」目前的角色映射，因此請先執行 GET 請求，讓您可以在 PUT 請求中包含所有目前的角色。在您無法存取 Dashboards 而又希望將 Amazon Cognito 的 IAM 角色映射到 `all_access` 角色時，REST API 特別有用。

手動快照

精細存取控制在擷取手動快照時複雜度較高。若要註冊快照儲存庫 (即使將 HTTP 基本身分驗證用於所有其他用途)，您必須將 `manage_snapshots` 角色映射至具備 `iam:PassRole` 許可能夠擔任 `TheSnapshotRole` 的 IAM 角色，如 the section called “先決條件” 中所定義。

然後使用該 IAM 角色將簽章的請求傳送到網域，如 the section called “註冊手動快照儲存庫” 中所述。

整合

如果您使用其他 AWS 服務搭配 OpenSearch Service，您必須為這些服務提供具有適當許可的 IAM 角色。例如，Firehose 交付串流通常會使用名為 的 IAM 角色`firehose_delivery_role`。在 Dashboards 中，建立精細存取控制的角色，然後將 IAM 角色映射到該角色。在此案例中，新的角色需要下列許可：

```
{
  "cluster_permissions": [
    "cluster_composite_ops",
    "cluster_monitor"
  ],
  "index_permissions": [{
    "index_patterns": [
      "firehose-index*"
    ],
    "allowed_actions": [
      "create_index",
```

```
      "manage",
      "crud"
    ]
  }]
}
```

許可會因每個服務執行的動作而不同。索引資料的 AWS IoT 規則或 AWS Lambda 函數可能需要與 Firehose 類似的許可，而僅執行搜尋的 Lambda 函數可以使用更有限的集合。

REST API 差異

精細存取控制 REST API 會因您的 OpenSearch/Elasticsearch 版本而有些許差異。在提出 PUT 請求前，請先提出 GET 請求以驗證預期的請求主體。例如，向 `_plugins/_security/api/user` 提出的 GET 請求會傳回所有使用者，讓您可以接著進行修改並用來提出有效的 PUT 請求。

在 Elasticsearch 6.x 上，建立使用者的請求如下所示：

```
PUT _opendistro/_security/api/user/new-user
{
  "password": "some-password",
  "roles": ["new-backend-role"]
}
```

在 OpenSearch 或 Elasticsearch 7.x 上，請求則看起來如下 (若使用 Elasticsearch，則將 `_plugins` 變更為`_opendistro`)：

```
PUT _plugins/_security/api/user/new-user
{
  "password": "some-password",
  "backend_roles": ["new-backend-role"]
}
```

此外，租用戶在 Elasticsearch 6.x 中是角色的屬性。

```
GET _opendistro/_security/api/roles/all_access

{
  "all_access": {
    "cluster": ["UNLIMITED"],
    "tenants": {
```

```
      "admin_tenant": "RW"
    },
    "indices": {
      "*": {
        "*": ["UNLIMITED"]
      }
    },
    "readonly": "true"
  }
}
```

在 OpenSearch 和 Elasticsearch 7.x 中，它們是具備自身 URI 的物件 (若使用 Elasticsearch，則將 `_plugins` 變更為 `_opendistro`)：

```
GET _plugins/_security/api/tenants

{
  "global_tenant": {
    "reserved": true,
    "hidden": false,
    "description": "Global tenant",
    "static": false
  }
}
```

如需 OpenSearch REST API 的文件，請參閱安全外掛程式 API 參考。

> **Tip**
>
> 如果您使用內部用戶資料庫，則可以使用 curl 發出請求並測試您的網域。嘗試以下範例命令：
>
> ```
> curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_search'
> curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_plugins/
> _security/api/user'
> ```

教學課程：使用 IAM 主要使用者和 Amazon Cognito 身分驗證設定網域

此本學課程涵蓋適用於精細存取控制的熱門 Amazon OpenSearch Service 使用案例：適用於 OpenSearch Dashboards 的 IAM 主要使用者與 Amazon Cognito 身分驗證。

在本教學課程中，我們將設定主要 IAM 角色和有限的 IAM 角色，然後我們將它們與 Amazon Cognito 中的使用者建立關聯。然後，主要使用者可以登入 OpenSearch Dashboards，將有限的使用者對應至角色，並使用精細存取控制來限制使用者的許可。

雖然這些步驟會使用 Amazon Cognito 使用者集區進行身分驗證，但相同的基本程序也適用於任何可讓您將不同 IAM 角色指派給不同使用者的 Cognito 身分驗證提供者。

在本教學課程中，您會完成下列步驟：

1. 建立主要和有限的 IAM 角色
2. 使用 Cognito 身分驗證建立網域
3. 設定 Cognito 使用者集區和身分集區
4. 在 OpenSearch Dashboards 中對應角色
5. 測試許可

步驟 1：建立主要和有限的 IAM 角色

導覽至 AWS Identity and Access Management (IAM) 主控台並建立兩個不同的角色：

- `MasterUserRole`：具備叢集完整許可並管理角色與角色映射的主要使用者。
- `LimitedUserRole`：更受限制的角色，您將向其授予作為主要使用者的有限存取權。

如需建立角色的指示，請參閱《IAM 使用者指南》中的使用自訂信任政策建立角色。

兩個角色都必須具有下列信任政策，以允許您的 Cognito 身分集區擔任相關角色：

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "cognito-identity.amazonaws.com:aud": "{identity-pool-id}"
      },
      "ForAnyValue:StringLike": {
        "cognito-identity.amazonaws.com:amr": "authenticated"
      }
    }
  }]
}
```

> **Note**
>
> 使用 Amazon Cognito 身分集區的唯一識別符取代 `identity-pool-id`。例如 `us-east-1:0c6cdba7-3c3c-443b-a958-fb9feb207aa6`。

步驟 2：使用 Cognito 身分驗證建立網域

導覽至位於 https：//https://console.aws.amazon.com/aos/home/ 的 Amazon OpenSearch Service 主控台，並使用下列設定建立網域：

- OpenSearch 1.0 或更高版本，或者 Elasticsearch 7.8 或更高版本
- 公開存取
- 以 `MasterUserRole` 作為主要使用者啟用的精細存取控制 (在上一個步驟中建立)
- 已為 OpenSearch Dashboards 啟用 Amazon Cognito 身分驗證。如需啟用 Cognito 身分驗證，以及選取使用者和身分集區的說明，請參閱the section called “設定網域以使用 Amazon Cognito 身分驗證”。
- 以下網域存取政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/{domain-name}/*"
    }
  ]
}
```

- 要求所有前往網域的流量皆使用 HTTPS
- 節點對節點加密
- 靜態資料加密

步驟 3：設定 Cognito 使用者

建立網域時，請依照《Amazon Cognito 開發人員指南》中的 Amazon Cognito建立使用者集區，在 Amazon Cognito 內設定主要和受限使用者。最後，請依照在 Amazon Cognito 中建立身分集區中的步驟來設定身分集區。使用者集區和身分集區必須在相同的 AWS 區域。

步驟 4：在 OpenSearch Dashboards 中映射角色

現在已設定您的使用者，您可以主要使用者身分登入 OpenSearch Dashboards，並將使用者映射至角色。

1. 返回 OpenSearch Service 主控台，然後導覽至您所建立網域的 OpenSearch Dashboards URL。URL 遵循此格式：*domain-endpoint*/_dashboards/。
2. 使用 master-user 憑證登入。
3. 選擇 Add sample data (新增範例資料)，並新增範例航班資料。
4. 在左側導覽窗格中，選擇 Security (安全)、Roles (角色)、Create role (建立角色)。
5. 將角色命名為 new-role。

6. 對於 Index (索引)，指定 `opensearch_dashboards_sample_data_fli*` (Elasticsearch 網域上的 `kibana_sample_data_fli*`)。
7. 對於 Index permissions (索引許可)，請選擇 read (讀取)。
8. 對於 Document level security (文件層級安全)，指定以下查詢：

```
{
  "match": {
    "FlightDelay": true
  }
}
```

9. 如需欄位層級的安全性，請選擇 Exclude (排除)，然後指定 `FlightNum`。
10. 對於 Anonymization (匿名化)，請指定 `Dest`。
11. 選擇建立。
12. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。新增 `LimitedUserRole` 的 Amazon Resource Name (ARN) 作為外部身分，然後選擇 Map (映射)。
13. 傳回角色清單，然後選擇 opensearch_dashboards_user。選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。新增 `LimitedUserRole` 的 ARN 作為後端角色，並選擇 Map (映射)。

步驟 5：測試許可

當您的角色映射正確時，您可以有限使用者身分登入並測試許可。

1. 在新的私有瀏覽器視窗中，導覽至網域的 OpenSearch Dashboards URL，使用 `limited-user` 憑證登入，然後選擇 Explore on my own (自行探索)。
2. 移至 Dev Tools (開發工具) 並執行預設搜尋：

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

   請注意許可錯誤。`limited-user` 沒有執行全叢集搜尋的許可。

3. 執行另一項搜尋：

```
GET opensearch_dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

請注意，所有相符文件的都有值為 `true` 的 `FlightDelay` 欄位、匿名化的 `Dest` 欄位，並不包含 `FlightNum` 欄位。

4. 在原始瀏覽器視窗中，以 `master-user` 的身分登入、選擇 Dev Tools (開發工具)，然後執行相同的搜尋。注意許可、命中數、相符文件以及所包含欄位中的差異。

教學課程：使用內部使用者資料庫和 HTTP 基本身分驗證設定網域

本教學課程涵蓋另一種熱門的精細存取控制使用案例：內部使用者資料庫中的主要使用者，以及適用於 OpenSearch Dashboards 的 HTTP 基本身分驗證。然後，主要使用者可以登入 OpenSearch Dashboards、建立內部使用者、將使用者對應至角色，並使用精細存取控制來限制使用者的許可。

在本教學課程中，您會完成下列步驟：

1. 使用主要使用者建立網域
2. 在 OpenSearch Dashboards 中設定內部使用者
3. 在 OpenSearch Dashboards 中對應角色
4. 測試許可

步驟 1：建立網域

導覽至位於 https：//https://console.aws.amazon.com/aos/home/ 的 Amazon OpenSearch Service 主控台，並使用下列設定建立網域：

- OpenSearch 1.0 或更高版本，或者 Elasticsearch 7.9 或更高版本
- 公開存取
- 使用內部使用者資料庫 (本教學中的其餘部分稱為 `TheMasterUser`) 中的主要使用者進行精細存取控制。
- Dashboards 的 Amazon Cognito 身分驗證已停用
- 以下存取政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/{domain-name}/*"
    }
  ]
}
```

- 要求所有前往網域的流量皆使用 HTTPS
- 節點對節點加密
- 靜態資料加密

步驟 2：在 OpenSearch Dashboards 中建立內部使用者

現在您已擁有網域，您可以登入 OpenSearch Dashboards 並建立內部使用者。

1. 返回 OpenSearch Service 主控台，然後導覽至您所建立網域的 OpenSearch Dashboards URL。URL 遵循此格式：*domain-endpoint*/_dashboards/。
2. 使用 登入TheMasterUser。
3. 選擇 Add sample data (新增範例資料)，並新增範例航班資料。
4. 在左側導覽窗格中，選擇安全性、內部使用者、建立內部使用者。
5. 命名使用者 new-user，然後指定密碼。然後選擇 Create (建立)。

步驟 3：在 OpenSearch Dashboards 中映射角色

現在您的使用者已設定完成，您可以將使用者映射至角色。

1. 停留在 OpenSearch Dashboards 的安全區段，然後選擇角色、建立角色。

2. 將角色命名為 new-role。
3. 對於索引，請指定索引模式的 opensearch_dashboards_sample_data_fli*(kibana_sample_data_fli*在 Elasticsearch 網域上）。
4. 對於動作群組，請選擇 read (讀取)。
5. 對於 Document level security (文件層級安全)，指定以下查詢：

```
{
  "match": {
    "FlightDelay": true
  }
}
```

6. 如需欄位層級的安全性，請選擇 Exclude (排除)，然後指定 FlightNum。
7. 對於 Anonymization (匿名化)，請指定 Dest。
8. 選擇建立。
9. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。然後將 new-user 新增至 Users (使用者)，然後選擇 Map (映射)。
10. 傳回角色清單，然後選擇 opensearch_dashboards_user。選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。然後將 new-user 新增至 Users (使用者)，然後選擇 Map (映射)。

步驟 4：測試許可

當您的角色映射正確時，您可以有限使用者身分登入並測試許可。

1. 在新的私有瀏覽器視窗中，導覽至網域的 OpenSearch Dashboards URL，使用 new-user 憑證登入，然後選擇 Explore on my own (自行探索)。
2. 移至 Dev Tools (開發工具) 並執行預設搜尋：

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

請注意許可錯誤。new-user 沒有執行全叢集搜尋的許可。

3. 執行另一項搜尋：

```
GET dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

請注意，所有相符文件的都有值為 true 的 FlightDelay 欄位、匿名化的 Dest 欄位，並不包含 FlightNum 欄位。

4. 在原始瀏覽器視窗中，以 TheMasterUser 的身分登入、選擇 Dev Tools (開發工具) 並執行相同的搜尋。注意許可、命中數、相符文件以及所包含欄位中的差異。

Amazon OpenSearch Service 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 OpenSearch Service 的安全與合規。這些計劃包括 SOC、PCI 和 HIPAA。

若您有合規性要求，請考慮使用任一版本的 OpenSearch 或 Elasticsearch 6.0 或更高版本。Elasticsearch 的早期版本不提供靜態資料加密和節點對節點加密之組合，因此不可能符合您的需求。若對您的使用案例而言，精細存取控制很重要，也可以考慮使用任一版本的 OpenSearch 或 Elasticsearch 6.7 或更高版本。無論如何，在建立網域時選擇特定的 OpenSearch 或 Elasticsearch 版本，皆無法保證合規性。

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 AWS 安全文件。

Amazon CopenSearFront Service 的復原功能

AWS 全球基礎設施是以 AWS 區域 和 可用區域為基礎建置。 AWS 區域 提供多個實體隔離和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。透過可用區域，您所設計與操作的應

用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

除了 AWS 全球基礎設施之外，OpenSearch Service 還提供數種功能，以協助支援您的資料彈性和備份需求：

- 異地同步備份網域和複本碎片
- 自動和手動快照

Amazon OpenSearch Service 的 JWT 身分驗證和授權

Amazon OpenSearch Service 現在可讓您使用 JSON Web Token (JWTs) 進行身分驗證和授權。JWTs是以 JSON 為基礎的存取字符，用於授予單一登入 (SSO) 存取。您可以在 OpenSearch Service 中使用 JWTs來建立單一登入字符，以驗證對 OpenSearch Service 網域的請求。若要使用 JWTs，您必須啟用精細存取控制，而且必須提供有效的 RSA 或 ECDSA PEM 格式公有金鑰。如需精細存取控制的詳細資訊，請參閱 Amazon OpenSearch Service 中的精細存取控制。

您可以使用 OpenSearch Service 主控台、 AWS Command Line Interface (AWS CLI) 或 AWS SDKs 來設定 JSON Web 權杖。

考量事項

將 JWTs與 Amazon OpenSearch Service 搭配使用之前，您必須考慮下列事項：

- 由於 RSA 公有金鑰的大小採用 PEM 格式，我們建議您使用 AWS 主控台來設定 JWT 身分驗證和授權。
- 指定 JWTs 的主題和角色欄位時，您必須提供有效的使用者和角色，否則請求將被拒絕。
- OpenSearch 2.11 是可用於 JWT 身分驗證的最早相容版本。

修改網域存取政策

您必須先更新網域存取政策，以允許 JWT 使用者存取網域，才能將網域設定為使用 JWT 身分驗證和授權。否則，所有傳入的 JWT 授權請求都會遭到拒絕。提供子資源 (/*) 完整存取權的建議網域存取政策為：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
    }
  ]
}
```

設定 JWT 身分驗證和授權

您可以在網域建立程序期間或透過更新現有網域來啟用 JWT 身分驗證和授權。設定步驟會因您選擇的選項而略有不同。

下列步驟說明如何在 OpenSearch Service 主控台中設定 JWT 身分驗證和授權的現有網域：

1. 在網域組態下，導覽至 OpenSearch 的 JWT 身分驗證和授權，選取啟用 JWT 身分驗證和授權。
2. 設定公有金鑰以用於您的網域。若要這樣做，您可以上傳 PEM 檔案、包含公有金鑰，或手動輸入。

   > **Note**
   >
   > 如果上傳或輸入的金鑰無效，則會在指定問題的文字方塊上方顯示警告。

3. （選用）在其他設定下，您可以設定下列選用欄位

   - 主旨金鑰 — 您可以保留此欄位空白，以使用 JWTs的預設`sub`金鑰。
   - 角色金鑰 — 您可以保留此欄位空白，以使用 JWTs的預設`roles`金鑰。

   完成變更後，請儲存您的網域。

使用 JWT 傳送測試請求

使用指定的主旨和角色對建立新的 JWT 之後，您可以傳送測試請求。若要這樣做，請使用私有金鑰，透過建立 JWT 的工具簽署您的請求。OpenSearch Service 能夠透過驗證此簽章來驗證傳入請求。

> **Note**
>
> 如果您為 JWT 指定了自訂主體金鑰或角色金鑰，則必須為 JWT 使用正確的宣告名稱。

以下是如何使用 JWT 字符透過網域的搜尋端點存取 OpenSearch Service 的範例：

```
curl -XGET "$search_endpoint" -H "Authorization: Bearer <JWT>"
```

設定 JWT 身分驗證和授權 (AWS CLI)

如果網域存在，以下 AWS CLI 命令會啟用 OpenSearch 的 JWT 身分驗證和授權：

```
aws opensearch update-domain-config --domain-name <your_domain_name> --advanced-
security-options '{"JWTOptions":{"Enabled":true, "PublicKey": "<your_public_key>",
 "SubjectKey": "<your_subject_key>", "RolesKey": "<your_roles_key>"}}'
```

設定 JWT 身分驗證和授權 （透過 API 設定）

下列對組態 API 的請求會在現有網域上啟用 OpenSearch 的 JWT 身分驗證和授權：

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "AdvancedSecurityOptions": {
    "JWTOptions": {
      "Enabled": true,
      "PublicKey": "public-key",
      "RolesKey": "optional-roles-key",
      "SubjectKey": "optional-subject-key"
   }
  }
}
```

產生金鑰對

為了為您的 OpenSearch 網域設定 JWTs，您需要以隱私權增強郵件 (PEM) 格式提供公有金鑰。Amazon OpenSearch Service 目前在使用 JWTs時支援兩種非對稱加密演算法：RSA 和 ECDSA。

若要使用通用 openssl 程式庫建立 RSA 金鑰對，請遵循下列步驟：

1. `openssl genrsa -out privatekey.pem 2048`
2. `openssl rsa -in privatekey.pem -pubout -out publickey.pem`

在此範例中， `publickey.pem` 檔案包含用於 Amazon OpenSearch Service 的公有金鑰，而 `privatekey.pem`包含用於簽署傳送至服務的 JWTs的私有金鑰。此外，如果您需要私有金鑰來產生 JWTs，您可以選擇將私有金鑰轉換為常用`pkcs8`格式。

如果您使用上傳按鈕將 PEM 檔案直接新增至主控台，則檔案必須具有`.pem`副檔名、其他副檔名，例如 `.crt.cert`或 目前`.key`不受支援。

Amazon OpenSearch Service 中的基礎設施安全性

作為受管服務， 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 。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。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 OpenSearch Service 組態 API。若要設定接受的最低所需 TLS 版本，請在網域端點選項中指定 `TLSSecurityPolicy` 值：

```
aws opensearch update-domain-config --domain-name my-domain --domain-endpoint-options
 '{"TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"}'
```

如需詳細資訊，請參閱 AWS CLI 命令參考。

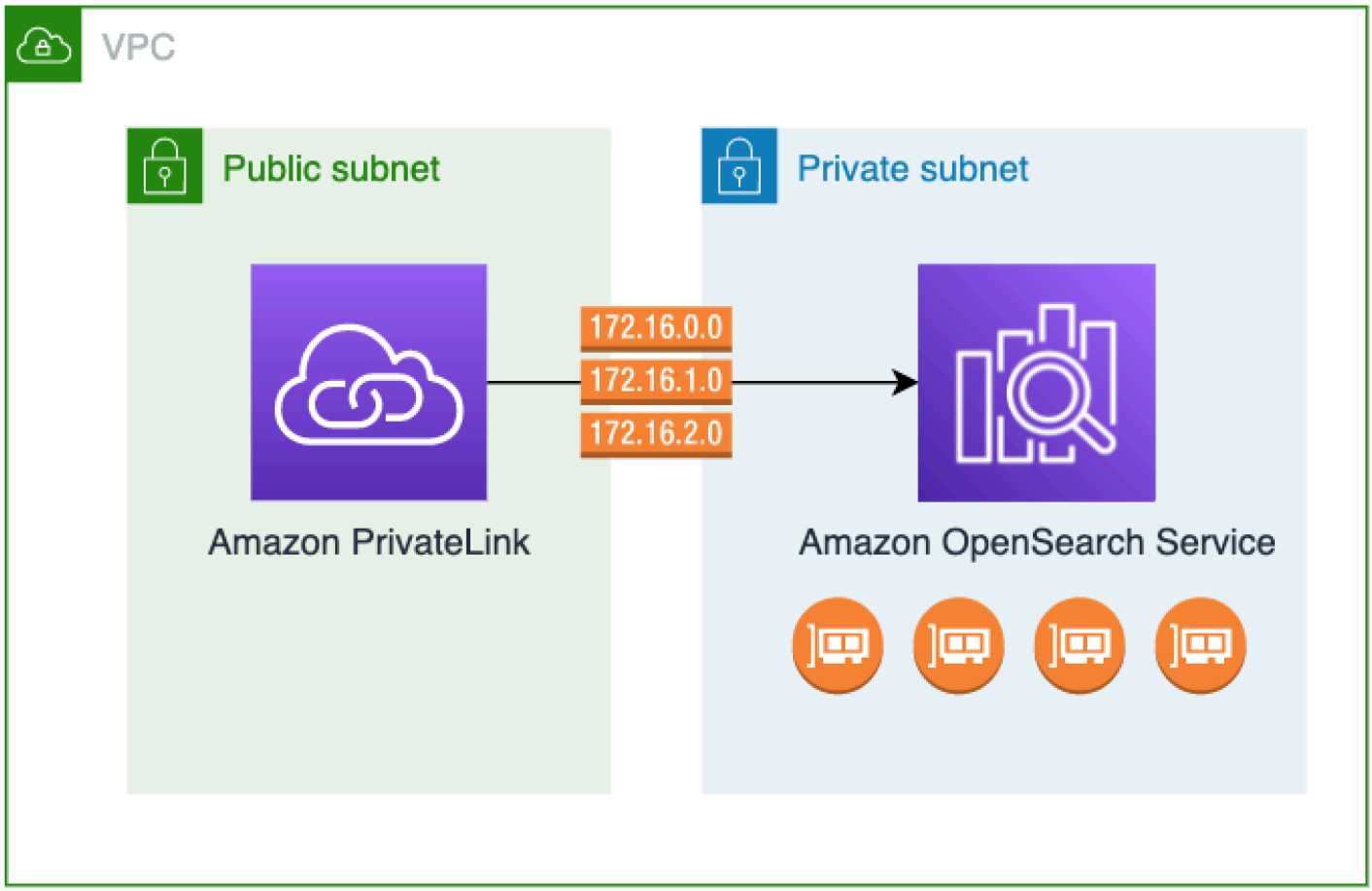
根據您的網域組態，您可能也需要簽署對 OpenSearch API 的請求。如需詳細資訊，請參閱the section called “提出並簽署 OpenSearch Service 請求”。

OpenSearch Service 支援公有存取網域 (可從任何網際網路連線裝置收到請求) 和 VPC 存取網域 (獨立於公有網際網路)。

使用 OpenSearch Service 受管 VPC 端點 (AWS PrivateLink) 存取 Amazon OpenSearch Service

您可以設定 OpenSearch Service 受管 VPC 端點 （ 由 提供支援 ） 來存取 Amazon OpenSearch Service 網域 AWS PrivateLink。這些端點可在 VPC 與 Amazon OpenSearch Service 之間建立私有連線。您可以像在 VPC 中一樣存取 OpenSearch Service VPC 網域，無需使用網際網路閘道、NAT 裝置、VPN 連接或 Direct Connect 連線。VPC 中的執行個體無需公有 IP 地址，即可存取 OpenSearch Service。

您可以將 OpenSearch Service 網域設定為公開在相同 VPC、不同 VPC 或不同 AWS 帳戶內的公有或私有子網路上執行的其他端點。這讓您可以新增額外的安全性層，以存取您的網域 (無論網域在何處執行)，而無需管理基礎架構。下圖說明相同 VPC 內的 OpenSearch Service 受管 VPC 端點：



您可以建立由 AWS PrivateLink提供支援的 OpenSearch Service 受管介面 VPC 端點，以建立此私有連線。我們會在您為介面 VPC 端點啟用的每個子網路中建立端點網路介面。這些是服務受管網路介面，可作為目的地為 OpenSearch Server 之流量的進入點。標準AWS PrivateLink 介面端點定價適用於依 計費的 OpenSearch Service 受管 VPC 端點 AWS PrivateLink。

您可以為執行所有版本的 OpenSearch 和舊版 Elasticsearch 的網域建立 VPC 端點。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的「透過 AWS PrivateLink存取 AWS 服務」。

OpenSearch Service 的考量事項與限制

設定 OpenSearch Service 的介面 VPC 端點之前，請先檢閱《 AWS PrivateLink 指南》中的使用介面 VPC 端點存取 AWS 服務。

使用 OpenSearch Service 受管 VPC 端點時，請考量下列事項：

- 您僅可使用介面 VPC 端點來連線至 VPC 網域。不支援公有網域。
- VPC 端點只能連線至相同 AWS 區域內的網域。
- HTTPS 是 VPC 端點唯一支援的通訊協定。不允許使用 HTTP。
- OpenSearch Server 支援透過介面 VPC 端點呼叫所有支援的 OpenSearch API 操作。
- 每個帳戶最多可設定 50 個端點，每個網域最多可設定 10 個端點。單一網域最多可有 10 個授權的主體。
- 您目前無法使用 AWS CloudFormation 建立介面 VPC 端點。
- 您僅可透過 OpenSearch Service 主控台或使用 OpenSearch Service API 建立介面 VPC 端點。您無法使用 Amazon VPC 主控台為 OpenSearch Service 建立介面 VPC 端點。
- OpenSearch Service 受管 VPC 端點無法從網際網路存取。在路由表和安全群組許可的情況下，OpenSearch Service 受管 VPC 端點只能在佈建端點的 VPC 或與佈建端點之 VPC 對等的任何 VPC 內存取。
- OpenSearch Service 不支援 VPC 端點政策。您可以將安全群組與端點網路介面相關聯，以控制透過介面 VPC 端點傳輸至 OpenSearch Server 的流量。
- 您的服務連結角色必須位於您用來建立 VPC 端點的相同 AWS 帳戶中。
- 若要建立、更新和刪除 OpenSearch Service VPC 端點，除了 Amazon OpenSearch Service 許可之外，您還必須擁有下列 Amazon EC2 許可： OpenSearch
  - `ec2:CreateVpcEndpoint`
  - `ec2:DescribeVpcEndpoints`
  - `ec2:ModifyVpcEndpoint`
  - `ec2:DeleteVpcEndpoints`

- `ec2:CreateTags`
- `ec2:DescribeTags`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeVpcs`

> **Note**
>
> 目前，您無法將 VPC 端點建立限制為 OpenSearch Service。我們正在努力在未來的更新中實現這一點。

提供對網域的存取

如果您想要存取網域的 VPC 位於另一個 VPC 中 AWS 帳戶，您必須先從擁有者的帳戶授權，才能建立介面 VPC 端點。

允許另一個 中的 VPC AWS 帳戶 存取您的網域

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home/ 的 Amazon OpenSearch Service 主控台。
2. 在導覽窗格中選擇 Domains (網域)，然後開啟您要提供存取權的網域。
3. 前往 VPC endpoints (VPC 端點) 索引標籤，其中會顯示可存取您網域的帳戶和對應 VPC。
4. 選擇 Authorize principal (授權主體)。
5. 輸入將存取您的網域的帳戶 AWS 帳戶 ID。此步驟會授權指定帳戶針對網域建立 VPC 端點。
6. 選擇 Authorize (授權)。

為 VPC 網域建立介面 VPC 端點

您可以使用 OpenSearch Service 主控台或 AWS Command Line Interface () 為 OpenSearch Service 建立介面 VPC 端點AWS CLI。

建立 OpenSearch Server 網域的介面 VPC 端點

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home/ 的 Amazon OpenSearch Service 主控台。

2. 在左側導覽窗格中選擇 VPC endpoints (VPC 端點)。
3. 選擇建立端點。
4. 選取是否要連接目前 中的網域 AWS 帳戶 或其他 AWS 帳戶。
5. 選取您與此端點連線的網域。如果網域位於目前的 中 AWS 帳戶，請使用下拉式清單選擇網域。如果網域位於不同的帳戶中，請輸入要連線之網域的 Amazon Resource Name (ARN)。若要選擇不同帳戶中的網域，擁有者需為您提供對網域的存取權。
6. 對於 VPC，選取您將從中存取 OpenSearch Service 的 VPC。
7. 對於 Subnets (子網路)，選取您將從中存取 OpenSearch Service 的一個或多個子網路。
8. 對於 Security group (安全群組)，選取要與端點網路介面建立關聯的安全群組。這是一個關鍵步驟，其中您需限制正授權進入端點之傳入流量的連接埠、通訊協定和來源。安全群組規則必須允許將使用 VPC 端點與 OpenSearch Service 通訊的資源，以便與端點網路介面通訊。
9. 選擇建立端點。端點應會在 2-5 分鐘內進入作用中狀態。

使用 OpenSearch Service 受管 VPC 端點 (使用組態 API)

使用下列 API 操作來建立和管理 OpenSearch Service 受管 VPC 端點。

- CreateVpcEndpoint
- ListVpcEndpoints
- UpdateVpcEndpoint
- DeleteVpcEndpoint

使用下列 API 操作來管理端點對 VPC 網域的存取：

- AuthorizeVpcEndpointAccess
- ListVpcEndpointAccess
- ListVpcEndpointsForDomain
- RevokeVpcEndpointAccess

OpenSearch Dashboards 的 SAML 身分驗證

OpenSearch Dashboards 的 SAML 身分驗證可讓您使用現有身分提供者，以便在執行 OpenSearch 或 Elasticsearch 6.7 或更新版本的 Amazon OpenSearch Service 網域上提供 Dashboards 的單一登入 (SSO)。若要使用 SAML 身分驗證，您必須啟用精細存取控制。

不是透過 Amazon Cognito 或內部使用者資料庫進行身分驗證，OpenSearch Dashboards 的 SAML 身分驗證可讓您使用第三方身分提供者來登入 Dashboards、管理精細存取控制、搜尋您的資料，以及建置視覺效果。OpenSearch Service 支援使用 SAML 2.0 標準的提供者，例如 Okta、Keycloak、Active Directory Federation Services (ADFS)、Auth0 和 AWS IAM Identity Center。

Dashboards 的 SAML 身分驗證僅適用於透過 web 瀏覽器存取 OpenSearch Dashboards。您的 SAML 憑證不能讓您對 OpenSearch 或 Dashboards API 發出直接 HTTP 請求。

SAML 組態概觀

本文件假設您擁有現有的身分提供者並且熟悉它。我們無法為您的確切供應商提供詳細的設定步驟，僅適用於您的 OpenSearch Service 網域。

OpenSearch Dashboards 登入流程可採用以下兩種形式之一：

- 服務供應商 (SP) 已啟動：您瀏覽至 Dashboards (例如 `https://my-domain.us-east-1.es.amazonaws.com/_dashboards`)，它會將您重新引導到登入畫面。登入後，身分提供者會將您重新引導至 Dashboards。
- 身分提供者 (IdP) 已啟動：瀏覽至身分提供者，登入並從應用程式目錄中選擇 OpenSearch Dashboards。

OpenSearch Service 提供兩個單一登入 URL，即 SP 啟動和 IdP 啟動，但您只需要一個，即符合您想要的 OpenSearch Dashboards 登入流程即可。

無論您使用哪種身分驗證類型，目標是透過身分提供者登入，並接收包含您的使用者名稱 (必要) 和任何後端角色 (選用，但建議使用) 的 SAML 聲明。此資訊允許精細存取控制以便將許可指派給 SAML 使用者。在外部身分提供者中，後端角色通常稱為「角色」或「群組」。

考量事項

設定 SAML 身分驗證時請考量下列事項：

- 由於 IdP 中繼資料檔案的大小，強烈建議使用 AWS 主控台來設定 SAML 身分驗證。
- 網域一次只支援一個 Dashboards 身分驗證方法。如果您啟用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證，則必須先將其停用才可啟用 SAML 身分驗證。
- 如果您搭配 SAML 使用網路負載平衡器，您必須先建立自訂端點。如需詳細資訊，請參閱???。
- 在非 IAM 身分的情況下，服務控制政策 (SCP) 將不適用或評估（例如 Amazon OpenSearch Serverless & SAML 中的 SAML 和 Amazon OpenSearch Service 的基本內部使用者授權）。

VPC 網域的 SAML 身分驗證

SAML 不需要身分提供者和服務供應商之間的直接通訊。因此，即使您的 OpenSearch 網域託管在私有 VPC 中，只要您的瀏覽器能夠與您的 OpenSearch 叢集和身分提供者通訊，則您仍可以使用 SAML。您的瀏覽器本質上充當身分提供者和服務供應商之間的媒介。如需有關解釋 SAML 身分驗證流程的實用圖表，請參閱 Okta 文件。

修改網域存取政策

在設定 SAML 身分驗證之前，您必須更新網域存取政策，以允許 SAML 使用者存取網域。否則，您將看到存取遭拒錯誤。

我們建議您採用下列網域存取政策，該政策提供對網域上子資源 (/*) 的完整存取權：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
    }
  ]
}
```

若要讓政策更具限制性，您可以將 IP 地址條件新增至政策。此情況只會限制對指定 IP 地址範圍或子網路的存取。例如，下列政策僅允許從 192.0.2.0/24 子網路進行存取：

JSON

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

```
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
    }
  ]
}
```

> **Note**
>
> 開放網域存取政策要求在您的網域上啟用精細存取控制，否則您會看到下列錯誤：
> `To protect domains with public access, a restrictive policy or fine-grained access control is required.`
> 如果您的主要使用者或內部使用者已設定強式密碼，則從安全角度而言，在使用精細存取控制時保持政策開啟可能是可接受的。如需詳細資訊，請參閱???。

設定 SP 或 IdP 啟動的身分驗證

這些步驟說明如何透過 OpenSearch Dashboards 的 SP 啟動的或 IdP 啟動的身分驗證來啟用 SAML 身分驗證。如需了解啟用兩者所需的額外步驟，請參閱同時啟用 SP 和 IdP 啟動的身分驗證。

步驟 1：啟用 SAML 身分驗證

您可以在網域建立期間啟用 SAML 身分驗證，或在現有網域上選擇 Actions (動作)、Edit security configuration (編輯安全組態)來啟用 SAML 身分驗證。根據您選擇的動作，以下步驟略有不同。

在網域組態中，於 SAML authentication for OpenSearch Dashboards/Kibana (OpenSearch Dashboards/Kibana 的 SAML 身分驗證) 下，選取 Enable SAML authentication (啟用 SAML 身分驗證)。

步驟 2：設定身分提供者

根據設定 SAML 身分驗證的時間，執行下列步驟。

如果您正在建立新網域

如果您正在建立新網域，OpenSearch Service 還無法產生服務提供者實體 ID 或 SSO URL。身分提供者需要這些值才能正確啟用 SAML 身分驗證，但只有在建立網域之後才能產生這些值。若要在網域建立期間處理此相互依存性，您可以在 IdP 組態中提供臨時值，以產生必要的中繼資料，然後在網域處於作用中狀態時加以更新。

如果您使用的是自訂端點，則可以推斷 URL 為何。例如，如果自訂端點是 `www.custom-endpoint.com`，服務提供者實體 ID 將會是 `www.custom-endpoint.com`，IdP 起始的 SSO URL 將會是 `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs/idpinitiated`，並且 SP 起始的 SSO URL 將會是 `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs`。在建立網域之前，您可以使用這些值來設定身分提供者。如需範例，請參閱下一區段。

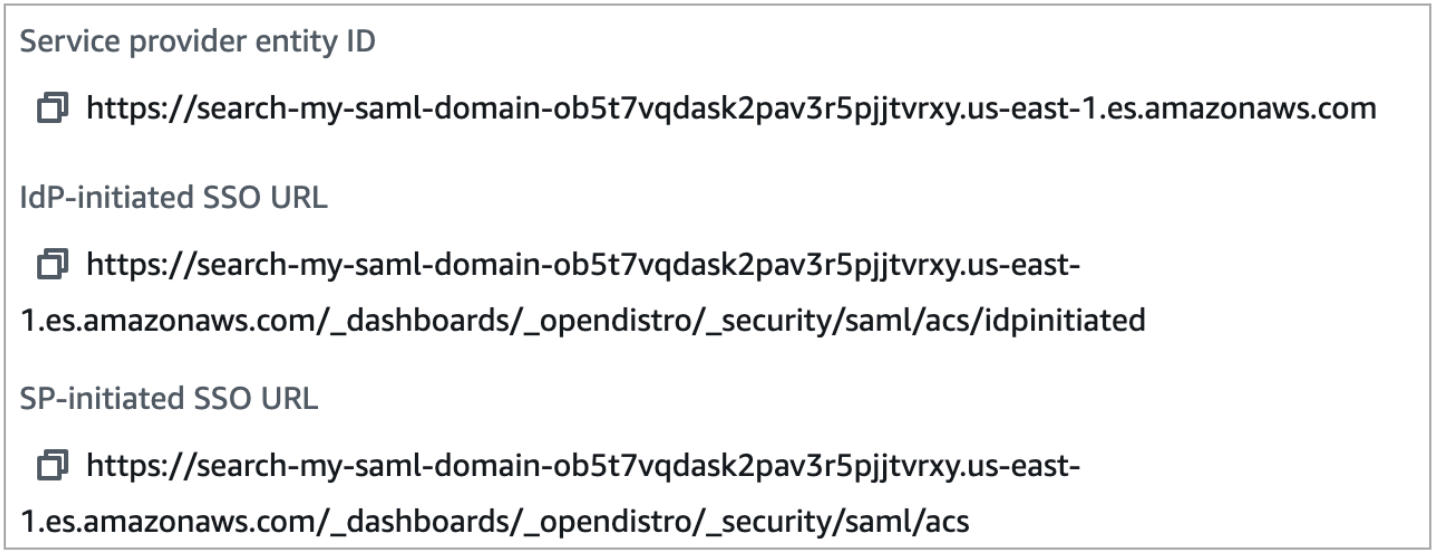
> **Note**
>
> 您不能使用雙堆疊端點登入，因為 HTTP 請求的 FQDN 與 SAML 請求的 FQDN 不同。如果您想要使用雙堆疊端點登入，OpenSearch 管理員將需要設定自訂端點，並將 CNAME 值設定為雙堆疊端點。

如果您未使用自訂端點，可以在 IdP 中輸入臨時值以產生所需的中繼資料，然後在網域處於作用中後加以更新。

例如，在 Okta 中，您可以將 `https://temp-endpoint.amazonaws.com` 輸入 Single sign on URL (單一登入 URL) 和 Audience URI (SP Entity ID) (對象 URI (SP 實體 ID)) 欄位，如此可產生中繼資料。然後，在網域處於作用中狀態後，您可以從 OpenSearch Service 中擷取正確的值並在 Okta 中加以更新。如需說明，請參閱the section called “步驟 6：更新 IdP URL”。

如果您正在編輯現有網域

如果您要在現有網域上啟用 SAML 身分驗證，請複製服務提供者實體 ID 和其中一個 SSO URL。如需有關使用哪個 URL 的指引，請參閱 the section called “SAML 組態概觀”。



使用這些值來設定身分提供者。這是程序中最複雜的部分，不幸的是，術語和步驟因供應商而千差萬別。請咨詢供應商文件。

例如，在 Okta 中，您可以建立 SAML 2.0 Web 應用程式。對於 Single sign on URL (單一登入 URL)，指定 SSO URL。對於對象 URI (SP 實體 ID)，指定 SP 實體 ID。

Okta 擁有使用者和群組，而不是使用者和後端角色。對於 Group Attribute Statements (群組屬性陳述式)，我們建議將 `role` 新增至 Name (名稱) 欄位，將常規表達式 `.+` 新增至 Filter (篩選條件) 欄位。此陳述式會告訴 Okta 身分提供者在使用中身分驗證之後，包含 SAML 聲明 `role` 欄位下的所有使用者群組。

在 IAM Identity Center 中，您將 SP 實體 ID 指定為 Application SAML 對象。您也需要指定下列屬性映射： `Subject=${user:subject}:format=unspecified`和 `Role=${user:groups}:format=uri`。

在 Auth0 中，您可以建立一般 Web 應用程式並啟用 SAML 2.0 附加元件。在 Keycloak 中，您可以建立用戶端。

步驟 3：匯入 IdP 中繼資料

設定身分提供者之後，會產生 IdP 中繼資料檔案。此 XML 檔案包含提供者的相關資訊，例如 TLS 憑證、單一登入端點以及身分提供者的實體 ID。

複製 IdP 中繼資料檔案內容，然後貼至 OpenSearch Service 主控台的 Metadata from IdP (來自 IdP 的中繼資料) 欄位。或者，選擇 Import from XML file (從 XML 檔案匯入)，然後上傳檔案。中繼資料檔案如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
 Location="idp-sso-url"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

步驟 4：設定 SAML 欄位

輸入 IdP 中繼資料之後，請在 OpenSearch Service 主控台中設定下列其他欄位：

- IdP entity ID (IdP 實體 ID) – 從中繼資料檔案中複製 `entityID` 屬性的值，然後貼至此欄位。許多身分提供者也會將此值顯示為組態後摘要的一部分。有些供應商稱之為「發行者」。
- SAML master username (SAML 主要使用者名稱) 和 SAML master backend role (SAML 主要後端角色) – 您指定的使用者和/或後端角色會收到叢集完整許可，相當於新主要使用者，但只能在 OpenSearch Dashboards 中使用這些許可。

  例如，在 Okta 中，您可能擁有屬於群組 `admins` 的使用者 `jdoe`。如果您將 `jdoe` 新增到 SAML 主要使用者名稱欄位，只有該使用者會收到完整許可。如果您將 `admins` 新增到 SAML 主要後端角色欄位中，屬於 `admins` 群組的任何使用者都會收到完整許可。

> **Note**
>
> SAML 聲明的內容必須完全符合您用於 SAML 主要使用者名稱和 SAML 主要角色的字串。某些身分供應商會在其使用者名稱之前新增字首，這可能會導致難以診斷的不相符。在身分

提供者使用者介面中，您可能會看到 jdoe，但 SAML 聲明可能包含 auth0|jdoe。永遠使用 SAML 聲明中的字串。

許多身分提供者可讓您在設定程序期間檢視範例聲明，諸如 SAML-tracer 等工具可以幫助您檢查和疑難排解真實聲明的內容。聲明看起來像這樣：

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion ID="id67229299299259351343340162"
 IssueInstant="2020-09-22T22:03:08.633Z" Version="2.0"
  xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">idp-issuer</
saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">username</saml2:NameID>
    <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <saml2:SubjectConfirmationData NotOnOrAfter="2020-09-22T22:08:08.816Z"
 Recipient="domain-endpoint/_dashboards/_opendistro/_security/saml/acs"/>
    </saml2:SubjectConfirmation>
  </saml2:Subject>
  <saml2:Conditions NotBefore="2020-09-22T21:58:08.816Z"
 NotOnOrAfter="2020-09-22T22:08:08.816Z">
    <saml2:AudienceRestriction>
      <saml2:Audience>domain-endpoint</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml2:Conditions>
  <saml2:AuthnStatement AuthnInstant="2020-09-22T19:54:37.274Z">
    <saml2:AuthnContext>

 <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport<
saml2:AuthnContextClassRef>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
    <saml2:Attribute Name="role" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
      <saml2:AttributeValue
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xs:string">GroupName Match Matches regex ".+" (case-sensitive)
      </saml2:AttributeValue>
    </saml2:Attribute>
```

```
  </saml2:AttributeStatement>
</saml2:Assertion>
```

步驟 5：(選用) 設定其他設定

在 Additional settings (其他設定) 下，設定下列選用欄位：

- Subject key (主體金鑰)– 您可以將此欄位留空，將 SAML 聲明的 `NameID` 元素用於使用者名稱。如果您的聲明不使用此標準元素，而是將使用者名稱作為自訂屬性，請在此處指定該屬性。
- Roles key (角色金鑰)– 如果您想要使用後端角色 (建議使用)，請在此欄位中從聲明中指定屬性，例如 `role` 或 `group`。這是 SAML-tracer 等工具可以提供協助的另一種情況。
- Session time to live (工作階段存留時間) – 依預設，OpenSearch Dashboards 會在 24 小時後將使用者登出。您可以透過指定新值，將此值設定為 60 至 1,440 分鐘 (24 小時) 之間的任何數字。

如果您對於組態感到滿意，請儲存網域。

步驟 6：更新 IdP URL

如果您在建立網域時啟用 SAML 身分驗證，則必須在 IdP 中指定臨時 URL，才能產生 XML 中繼資料檔案。網域狀態變更為 `Active` 後，您可以取得正確的 URL 並修改 IdP。

若要擷取 URL，請選取網域並選擇 Actions (動作)，Edit security configuration (編輯安全組態)。在 SAML authentication for OpenSearch Dashboards/Kibana (OpenSearch Dashboards/Kibana 的 SAML 身分驗證) 下，您可以找到正確的服務提供者實體 ID 和 SSO URL。複製這些值並使用其來設定身分提供者，取代您在步驟 2 中提供的臨時 URL。

步驟 7：將 SAML 使用者映射至角色

當網域狀態為作用中且 IdP 的設定都正確無誤後，請瀏覽至 OpenSearch Dashboards。

- 如果您選擇 SP 啟動的 URL，請導覽至 *`domain-endpoint`*`/_dashboards`。若要直接登入特定租用戶，您可以將 `?security_tenant=`*`tenant-name`* 附加至 URL。
- 如果您選擇 IdP 啟動的 URL，請導覽至身分提供者的應用程式目錄。

在這兩種情況下，請以 SAML 主要使用者或屬於 SAML 主要後端角色的使用者身分登入。若要繼續步驟 7 的範例，請以 `jdoe` 或 `admins` 群組成員身分登入。

OpenSearch Dashboards 載入後，選擇 Security (安全性) 和 Roles (角色)。接著，映射角色以允許其他使用者存取 OpenSearch Dashboards。

例如，您可將信任的同事 `jroe` 映射到 `all_access` 和 `security_manager` 角色。您也可以將後端角色 `analysts` 映射到 `readall` 和 `opensearch_dashboards_user` 角色。

如果您偏好使用 API 而非 OpenSearch Dashboards，請參閱下列範例要求：

```
PATCH _plugins/_security/api/rolesmapping
[
  {
    "op": "add", "path": "/security_manager", "value": { "users": ["master-user",
 "jdoe", "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/all_access", "value": { "users": ["master-user", "jdoe",
 "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/readall", "value": { "backend_roles": ["analysts"] }
  },
  {
    "op": "add", "path": "/opensearch_dashboards_user", "value": { "backend_roles":
 ["analysts"] }
  }
]
```

同時設定 SP 和 IdP 啟動的身分驗證

如果您要設定 SP 和 IdP 啟動的身分驗證，則必須透過身分提供者進行設定。例如，在 Okta 中，您可以執行下列步驟：

1. 在您的 SAML 應用程式中，移至 General (一般)、SAML settings (SAML 設定)。
2. 對於 Single sign on URL (單一登入 URL)，提供 IdP 啟動的 SSO URL。例如 `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs/idpinitiated`。
3. 啟用 Allow this app to request other SSO URLs (允許此應用程式請求其他 SSO URL)。
4. 在 Requestable SSO URLs (可請求的 SSO URL) 中，新增一個或多個 SP 啟動的 SSO URL。例如 `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs`。

設定 SAML 身分驗證 (AWS CLI)

下列 AWS CLI 命令會在現有網域上啟用 OpenSearch Dashboards 的 SAML 身分驗證：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --advanced-security-options '{"SAMLOptions":{"Enabled":true,"MasterUserName":"my-idp-user","MasterBackendRole":"my-idp-group-or-role","Idp":{"EntityId":"entity-id","MetadataContent":"metadata-content-with-quotes-escaped"},"RolesKey":"optional-roles-key","SessionTimeoutMinutes":180,"SubjectKey":"optional-subject-key"}}'
```

您必須轉義中繼資料 XML 中的所有引號和換行符號字元。例如，使用 `<KeyDescriptor use=\"signing\">\n`，而不是 `<KeyDescriptor use="signing">` 和換行符。如需有關使用 AWS CLI的詳細資訊，請參閱 AWS CLI 命令參考。

設定 SAML 身分驗證 (組態 API)

下列對組態 API 的請求會在現有網域上啟用 OpenSearch Dashboards 的 SAML 身分驗證：

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "AdvancedSecurityOptions": {
    "SAMLOptions": {
      "Enabled": true,
      "MasterUserName": "my-idp-user",
      "MasterBackendRole": "my-idp-group-or-role",
      "Idp": {
        "EntityId": "entity-id",
        "MetadataContent": "metadata-content-with-quotes-escaped"
      },
      "RolesKey": "optional-roles-key",
      "SessionTimeoutMinutes": 180,
      "SubjectKey": "optional-subject-key"
    }
  }
}
```

您必須轉義中繼資料 XML 中的所有引號和換行符號字元。例如，使用 `<KeyDescriptor use=\"signing\">\n`，而不是 `<KeyDescriptor use="signing">` 和換行符。如需有關使用組態 API 的詳細資訊，請參閱 OpenSearch Service API reference (OpenSearch Service API 參考)。

SAML 疑難排解

錯誤	詳細資訊
您的請求：'*/some/path* ' 不允許。	確認您為身分提供者提供了正確的 SSO URL (步驟 3)。
請提供有效的身分提供者中繼資料文件以啟用 SAML。	您的 IdP 中繼資料檔案不符合 SAML 2.0 標準。使用驗證工具檢查錯誤。
SAML 組態選項在主控台中不可見。	更新至最新服務軟體。
SAML 組態錯誤：擷取 SAML 組態時發生錯誤，請檢查您的設定。	此一般性錯誤的發生原因很多。 • 檢查您是否為身分提供者提供了正確的 SP 實體 ID 和 SSO URL。 • 重新產生 IdP 中繼資料檔案，並驗證 IdP 實體 ID。在 AWS 主控台中新增任何已更新的中繼資料。 • 請確認您的網域存取政策允許存取 OpenSearch Dashboards 和 `_plugins/_security/*` 。一般而言，對於使用精細存取控制的網域，我們建議使用開放式存取政策。 • 如需設定 SAML 的步驟，請參閱身分提供者的文件。
缺少角色：此使用者沒有可用的角色，請聯絡您的系統管理員。	您已成功進行身分驗證，但 SAML 聲明中的使用者名稱和任何後端角色未映射至任何角色，因此沒有許可。這些映射區分大小寫。 您的系統管理員可以使用 SAML-tracer 等工具驗證 SAML 聲明的內容，然後使用下列請求檢查您的角色映射： `GET _plugins/_security/api/rolesmapping`
當您嘗試存取 OpenSearch Dashboards 時，您的瀏覽器會持續重新引導或收到 HTTP 500 錯誤。	如果您的 SAML 聲明包含大量的角色，總計約 1,500 個字元，就會發生這些錯誤。例如，如果您傳遞 80 個角色，平均長度為 20 個字元，您可能會超過 web 瀏

錯誤	詳細資訊
	覽器中 Cookie 的大小限制。從 OpenSearch 2.7 版開始，SAML 聲明支援最多 5000 個字元的角色。
您無法登出 ADFS。	ADFS 要求簽署所有登出請求，而 OpenSearch Service 不支援這些請求。從 IdP 中繼資料檔案中移除 `<SingleLogoutService />` ，以強制 OpenSearch Service 使用自己的內部登出機制。
`Could not find entity descriptor for __PATH__.`	中繼資料 XML 中提供給 OpenSearch Service 的 IdP 實體 ID 與 SAML 回應中的實體 ID 不同。若要修正此問題，請確定它們相符。在您的網域上啟用 CW 應用程式錯誤日誌，以尋找偵錯 SAML 整合問題的錯誤訊息。
`Signature validation failed. SAML response rejected.`	OpenSearch Service 無法使用中繼資料 XML 中提供的 IdP 憑證來驗證 SAML 回應中的簽章。這可能是手動錯誤，或者您的 IdP 已輪換其憑證。在透過 提供給 OpenSearch Service 的中繼資料 XML 中，從 IdP 更新最新的憑證 AWS 管理主控台。
`__PATH__ is not a valid audience for this response.`	SAML 回應中的對象欄位與網域端點不相符。若要修正此錯誤，請更新 SP 對象欄位以符合您的網域端點。如果您已啟用自訂端點，對象欄位應與自訂端點相符。在您的網域上啟用 CW 應用程式錯誤日誌，以尋找偵錯 SAML 整合問題的錯誤訊息。
您的瀏覽器在回應`Invalid Request Id`中收到 HTTP 400 錯誤與 。	如果您已使用 格式設定 IdP 起始的 URL，則通常會發生此錯誤`<DashboardsURL> /_opendistro/_security/saml/acs` 。反之，請以 格式設定 URL`<DashboardsURL> /_opendistro/_security/saml/acs/idpinitiated` 。

錯誤	詳細資訊
在 收到回應，__PATH__而不是 __PATH__。	SAML 回應中的目的地欄位不符合下列其中一個 URL 格式： • *<DashboardsURL>* /_opendistro/_security/saml/acs • *<DashboardsURL>* /_opendistro/_security/saml/acs/idpinitiated . 根據您使用的登入流程 (SP 起始或 IdP 起始），在符合其中一個 OpenSearch URLs目的地欄位中輸入 。
回應具有 InResponseTo 屬性，但InResponseTo 預期不會。	您正在針對 SP 起始的登入流程使用 IdP 起始的 URL。請改用 SP 起始的 URL。

停用 SAML 身分驗證

若要停用 OpenSearch Dashboards (主控台) 的 SAML 身分驗證

1. 選擇網域、Actions (動作) 和 Edit security configuration (編輯安全組態)。
2. 取消勾選 Enable SAML authentication (啟用 SAML 身分驗證)。
3. 選擇儲存變更。
4. 網域完成處理之後，請使用下列請求確認精細存取控制角色映射：

```
GET _plugins/_security/api/rolesmapping
```

停用 Dashboards 的 SAML 身分驗證不會移除 SAML 主要使用者名稱和/或 SAML 主要後端角色的映射。如果您要移除這些映射，請使用內部使用者資料庫 (如果已啟用) 登入 Dashboards，或使用 API 將其移除：

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "users": [
    "master-user"
  ]
```

```
}
```

OpenSearch 的 IAM Identity Center 受信任身分傳播支援

您現在可以透過信任的身分傳播，使用集中設定的 AWS IAM Identity Center 主體（使用者和群組），透過 OpenSearch Service 應用程式 存取 Amazon OpenSearch Service 網域。若要啟用 OpenSearch 的 IAM Identity Center 支援，您需要啟用 IAM Identity Center 的使用。若要進一步了解如何執行此操作，請參閱什麼是 IAM Identity Center？。如需詳細資訊，請參閱如何在 OpenSearch 應用程式中將 OpenSearch 網域關聯為資料來源？。

您可以使用 OpenSearch Service 主控台、 AWS Command Line Interface (AWS CLI) 或 SDKs 來 AWS 設定 IAM Identity Center。

> **Note**
>
> 透過 Dashboards（與叢集共置）不支援 IAM Identity Center 主體。它們僅透過集中式 OpenSearch 使用者介面（儀表板）支援。

考量事項

將 IAM Identity Center 與 Amazon OpenSearch Service 搭配使用之前，您必須考慮下列事項：

- 帳戶已啟用 IAM Identity Center。
- IAM Identity Center 可在您的 區域使用。
- OpenSearch 網域版本為 1.3 或更新版本。
- 網域上已啟用精細存取控制。
- 網域與 IAM Identity Center 執行個體位於相同的區域。
- 網域和 OpenSearch 應用程式屬於同一個 AWS 帳戶。

修改網域存取政策

設定 IAM Identity Center 之前，您必須更新網域存取政策或在 OpenSearch 應用程式中為受信任身分傳播設定的 IAM 角色許可。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/OpenSearchRole"
            },
            "Action": "es:ESHttp*",
            "Resource": "arn:aws:es:us-east-1:111122223333:domain/example-domain/
*"
        }
    ]
}
```

設定 IAM Identity Center 身分驗證和授權（主控台）

您可以在網域建立程序期間或透過更新現有網域來啟用 IAM Identity Center 身分驗證和授權。設定步驟會因您選擇的選項而略有不同。

下列步驟說明如何在 Amazon OpenSearch Service 主控台中設定 IAM Identity Center 身分驗證和授權的現有網域：

1. 在網域組態下，導覽至安全組態，選擇編輯並導覽至 IAM Identity Center Authentication 區段，然後選取啟用使用 IAM Identity Center 驗證的 API 存取。
2. 選取 SubjectKey 和 Roles 金鑰，如下所示。
   - 主旨金鑰 - 選擇其中一個 UserId（預設）、UserName 和 Email，以使用對應的屬性做為存取網域的主體。
   - 角色金鑰 - 選擇其中一個 GroupId（預設）和 GroupName，針對與 IAM Identity Center 主體相關聯的所有群組，使用對應的屬性值做為fine-grained-access-control的後端角色。

完成變更後，請儲存您的網域。

設定精細存取控制

在 OpenSearch 網域上啟用 IAM Identity Center 選項後，您可以透過建立與後端角色的角色映射來設定對 IAM Identity Center 主體的存取。委託人的後端角色值是以 IAM Identity Center 委託人的群組成員資格和 GroupId 或 GroupName 的 RolesKey 組態為基礎。

> **Note**
>
> Amazon OpenSearch Service 最多可為單一使用者支援 100 個群組。如果您嘗試使用超過允許的執行個體數量，您會遇到與fine-grained-access-control授權處理不一致的情況，並收到 403 錯誤訊息。

設定 IAM Identity Center 身分驗證和授權 (CLI)

```
  aws opensearch update-domain-config \
      --domain-name my-domain \
      --identity-center-options '{"EnabledAPIAccess": true,
 "IdentityCenterInstanceARN": "instance arn",  "SubjectKey": "UserId/UserName/
UserEmail" , "RolesKey": "GroupId/GroupName"}'
```

在網域上停用 IAM Identity Center 身分驗證

若要在 OpenSearch 網域上停用 IAM Identity Center：

1. 選擇網域、Actions (動作) 和 Edit security configuration (編輯安全組態)。
2. 取消勾選啟用透過 IAM Identity Center 驗證的 API 存取。
3. 選擇儲存變更。
4. 網域完成處理後，移除為 IAM Identity Center 主體新增的角色映射

若要透過 CLI 停用 IAM Identity Center，您可以使用下列

```
  aws opensearch update-domain-config \
      --domain-name my-domain \
```

```
    --identity-center-options '{"EnabledAPIAccess": false}'
```

設定 OpenSearch Dashboards 的 Amazon Cognito 身分驗證

您可以使用 Amazon Cognito 進行身分驗證並保護 OpenSearch Dashboards 的 Amazon OpenSearch Service 預設安裝。Amazon Cognito 身分驗證是可選的，並僅適用於使用 OpenSearch 或 Elasticsearch 5.1 或更高版本的網域。如果您不設定 Amazon Cognito 身分驗證，您仍然可以使用以 IP 為基礎的存取政策和代理伺服器、HTTP 基本身分驗證或 SAML 來保護 Dashboards。

大部分身分驗證程序發生在 Amazon Cognito 中，但本節提供設定 Amazon Cognito 資源以使用 OpenSearch Service 網域的準則和要求。標準定價適用於所有 Amazon Cognito 資源。

> **Tip**
>
> 第一次設定網域以便對 OpenSearch Dashboards 使用 Amazon Cognito 身分驗證時，我們建議您使用主控台。Amazon Cognito 資源完全可自訂，而主控台可協助您識別和了解重要的功能。

主題

- 先決條件
- 設定網域以使用 Amazon Cognito 身分驗證
- 允許已經過身分驗證的角色
- 設定身分提供者
- (選用) 設定精細分級的存取
- (選用) 自訂登入頁面
- (選用) 設定進階安全性
- 測試
- 配額
- 常見的設定問題
- 停用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證
- 刪除使用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證的網域

先決條件

您必須滿足幾個先決條件，之後您才能夠為 OpenSearch Dashboards 設定 Amazon Cognito 身分驗證。OpenSearch Service 主控台可協助簡化這些資源的建立，但了解每個資源的目的有助於進行設定和故障診斷。針對 Dashboards 的 Amazon Cognito 身分驗證需要下列資源：

- Amazon Cognito 使用者集區
- Amazon Cognito 身分集區
- 連接 `AmazonOpenSearchServiceCognitoAccess` 政策的 IAM 角色 (`CognitoAccessForAmazonOpenSearch`)

> **Note**
>
> 使用者集區和身分集區必須在相同的 AWS 區域。您可以使用相同的使用者集區、身分集區和 IAM 角色，將 Dashboards 的 Amazon Cognito 身分驗證新增到多個 OpenSearch Service 網域。如需詳細資訊，請參閱 the section called “配額”。

關於使用者集區

使用者集區有兩個主要功能：建立和管理使用者目錄，並讓使用者註冊和登入。如需建立使用者集區的指示，請參閱《Amazon Cognito 開發人員指南》中的使用者集區入門。

當您建立使用者集區以與 OpenSearch Service 搭配使用時，請考慮以下事項：

- 您的 Amazon Cognito 使用者集區必須擁有網域名稱。OpenSearch Service 使用此網域名稱將使用者重新引導至存取 Dashboards 的登入頁面。除了網域名稱，使用者集區不需要任何非預設組態。
- 您必須指定集區所需的標準屬性 - 例如名稱、出生日期、電子郵件地址和電話號碼等屬性。在您建立使用者集區之後您無法變更這些屬性，因此請選擇目前對您重要的屬性。
- 建立您的使用者集區時，選擇使用者是否可以建立自己的帳戶、帳戶的最低密碼強度，以及是否啟用多重要素驗證。如果您計劃使用外部身分提供者，這些設定是無關緊要的。通常，您可以讓使用者集區做為身分提供者並且啟用外部身分提供者，但大多數使用者偏好非此即彼。

使用者集區 ID 採用 *`region_ID`* 格式。如果您計劃使用 AWS CLI 或 AWS SDK 來設定 OpenSearch Service，請記下 ID。

關於身分集區

身分集區可讓您在使用者登入後將臨時受限的許可角色指派給使用者。如需建立身分集區的指示，請參閱《Amazon Cognito 開發人員指南》中的身分集區主控台概觀。當您建立身分集區以與 OpenSearch Service 搭配使用時，請考慮以下事項：

- 如果您使用 Amazon Cognito 主控台，您必須選擇 Enable access to unauthenticated identities (允許存取未經驗證的身分) 核取方塊，以建立身分集區。在您建立身分集區並設定 OpenSearch Service 網域之後，Amazon Cognito 會停用此設定。
- 您不需要新增外部身分提供者到身分集區。當您設定 OpenSearch Service 以使用 Amazon Cognito 身分驗證時，它會設定身分集區以使用您剛建立的使用者集區。
- 建立身分集區之後，您必須選擇未經授權和經過授權的 IAM 角色。這些角色指定使用者登入前後的存取政策。如果您使用 Amazon Cognito 主控台，它可為您建立這些角色。在您建立經過授權的角色後，請記下採用 `arn:aws:iam::`*`123456789012`*`:role/Cognito_`*`identitypoolname`*`Auth_Role` 格式的 ARN。

身分集區 ID 採用 *`region`*`:`*`ID-ID-ID-ID-ID`* 格式。如果您計劃使用 AWS CLI 或 AWS SDK 來設定 OpenSearch Service，請記下 ID。

關於 CognitoAccessForAmazonOpenSearch 角色

OpenSearch Service 需要許可才能設定 Amazon Cognito 使用者集區和身分集區，並使用它們來進行身分驗證。為此，您可以使用 `AmazonOpenSearchServiceCognitoAccess`，這是 AWS 受管政策。 `AmazonESCognitoAccess` 是舊版政策，當服務重新命名為 Amazon OpenSearch Service `AmazonOpenSearchServiceCognitoAccess`時，已由 取代。這兩個政策都提供啟用 Amazon Cognito 身分驗證所需的最低 Amazon Cognito 許可。如需政策詳細資訊，請參閱《 AWS 受管政策參考指南》中的 AmazonOpenSearchServiceCognitoAccess。

如果您使用主控台來建立或設定 OpenSearch Service 網域，它會為您建立 IAM 角色，然後將 `AmazonOpenSearchServiceCognitoAccess` 政策 (或者，若為 Elasticsearch 網域，則為 `AmazonESCognitoAccess` 政策) 連接到角色。此角色的預設名稱為 `CognitoAccessForAmazonOpenSearch`。

角色許可政策 `AmazonOpenSearchServiceCognitoAccess` 和 `AmazonESCognitoAccess` 均允許 OpenSearch Service 在所有身分識別和使用者集區上完成下列動作：

- 動作：`cognito-idp:DescribeUserPool`
- 動作：`cognito-idp:CreateUserPoolClient`

- 動作：cognito-idp:DeleteUserPoolClient
- 動作：cognito-idp:UpdateUserPoolClient
- 動作：cognito-idp:DescribeUserPoolClient
- 動作：cognito-idp:AdminInitiateAuth
- 動作：cognito-idp:AdminUserGlobalSignOut
- 動作：cognito-idp:ListUserPoolClients
- 動作：cognito-identity:DescribeIdentityPool
- 動作：cognito-identity:SetIdentityPoolRoles
- 動作：cognito-identity:GetIdentityPoolRoles

如果您使用 AWS CLI 或其中一個 AWS SDKs，則必須建立自己的角色、連接政策，並在設定 OpenSearch Service 網域時指定此角色的 ARN。角色必須具有下列信任關係：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opensearchservice.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需說明，請參閱《IAM 使用者指南》中的建立角色以將許可委派給 AWS 服務，以及新增和移除 IAM 身分許可。 https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-attach-detach.html

設定網域以使用 Amazon Cognito 身分驗證

在完成先決條件之後，您可以設定 OpenSearch Service 網域以使用適用於 Dashboards 的 Amazon Cognito。

> **Note**
>
> Amazon Cognito 不適用於所有 AWS 區域。如需支援的區域清單，請參閱 Amazon Cognito 的服務端點。您不需要對用於 OpenSearch Service 的 Amazon Cognito 使用相同區域。

設定 Amazon Cognito 身分驗證 (主控台)

由於主控台會為您建立 CognitoAccessForAmazonOpenSearch角色，因此可提供最簡單的組態體驗。除了標準 OpenSearch Service 許可外，您還需要以下一組許可，才能使用主控台來建立網域，該網域對 OpenSearch 儀表板使用 Amazon Cognito 身分驗證。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools",
        "iam:CreateRole",
        "iam:AttachRolePolicy"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/service-
role/CognitoAccessForAmazonOpenSearch"
    }
  ]
}
```

如需有關將許可新增至身分 (使用者、使用者群組或角色) 的說明，請參閱新增 IAM 身分許可 (主控台)。

如果 CognitoAccessForAmazonOpenSearch 已存在，您需要的許可更少：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/service-
role/CognitoAccessForAmazonOpenSearch"
    }
  ]
}
```

若要設定 Dashboards 的 Amazon Cognito 身分驗證 (主控台)

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home/ 的 Amazon OpenSearch Service 主控台。
2. 在 Domains (網域) 下，選取您要設定的網域。
3. 選擇 Actions (動作)、Edit security configuration (編輯安全組態)。
4. 選擇 Enable Amazon Cognito authentication (啟用 Amazon Cognito 身分驗證)。
5. 針對區域，選取 AWS 區域 包含 Amazon Cognito 使用者集區和身分集區的 。

6. 對於 Cognito user pool (Cognito 使用者集區)，選擇使用者集區或建立集區。如需詳細資訊，請參閱the section called “關於使用者集區”。
7. 對於 Cognito identity pool (Cognito 身分集區)，選擇身分集區或建立集區。如需詳細資訊，請參閱the section called “關於身分集區”。

> **Note**
>
> Create user pool (建立使用者集區) 和 Create identity pool (建立身分集區) 連結會將您導向到 Amazon Cognito 主控台，並要求您手動建立這些資源。此程序不是自動的。如需詳細資訊，請參閱the section called “先決條件”。

8. 對於 IAM role name (IAM 角色名稱)，使用 `CognitoAccessForAmazonOpenSearch` 的預設值 (建議) 或輸入新名稱。如需詳細資訊，請參閱the section called “關於 CognitoAccessForAmazonOpenSearch 角色”。
9. 選擇 Save changes (儲存變更)。

在您的網域完成處理之後，請參閱the section called “允許已經過身分驗證的角色”和the section called “設定身分提供者”以取得其他設定步驟。

設定 Amazon Cognito 身分驗證 (AWS CLI)

使用 `--cognito-options` 參數來設定 OpenSearch Service 網域。由 `create-domain` 和 `update-domain-config` 命令使用以下語法：

```
--cognito-options Enabled=true,UserPoolId="user-pool-id",IdentityPoolId="identity-pool-id",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonOpenSearch"
```

範例

以下範例在 `us-east-1` 區域建立網域，該區域使用 `CognitoAccessForAmazonOpenSearch` 角色來啟用 Dashboards 的 Amazon Cognito 身分驗證，並提供對 `Cognito_Auth_Role` 的網域存取：

```
aws opensearch create-domain --domain-name my-domain --
region us-east-1 --access-policies '{ "Version": "2012-10-17",
        "Statement":[{"Effect":"Allow","Principal":{"AWS":
 ["arn:aws:iam::123456789012:role/
Cognito_Auth_Role"]},"Action":"es:ESHttp*","Resource":"arn:aws:es:us-
east-1:123456789012:domain/*" }]}' --engine-version "OpenSearch_1.0"
```

```
 --cluster-config InstanceType=m4.xlarge.search,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeSize=10 --cognito-options
 Enabled=true,UserPoolId="us-east-1_123456789",IdentityPoolId="us-
east-1:12345678-1234-1234-1234-123456789012",RoleArn="arn:aws:iam::123456789012:role/
CognitoAccessForAmazonOpenSearch"
```

在您的網域完成處理之後，請參閱the section called “允許已經過身分驗證的角色”和the section called “設定身分提供者”以取得其他設定步驟。

設定 Amazon Cognito 身分驗證AWS SDKs)

AWS SDKs (Android 和 iOS SDKs除外） 支援 Amazon OpenSearch Service API 參考中定義的所有操作，包括 `CreateDomain`和 `UpdateDomainConfig`操作的 `CognitoOptions` 參數。如需安裝和使用 AWS SDKs 的詳細資訊，請參閱AWS 軟體開發套件。

在您的網域完成處理之後，請參閱the section called “允許已經過身分驗證的角色”和the section called “設定身分提供者”以取得其他設定步驟。

允許已經過身分驗證的角色

在預設情況下，您遵照 the section called “關於身分集區” 中的準則設定的已經過身分驗證的 IAM 角色並不具備存取 OpenSearch Dashboards 所必需的權限。您必須提供該角色額外的許可。

> **Note**
>
> 如果您已設定精細存取控制並使用開放或以 IP 為基礎的存取政策，您可以略過此步驟。

您可以將這些許可包含在身分型政策中，但除非您希望已經過身分驗證的使用者可存取所有 OpenSearch Service 網域，否則連接到單一網域的資源型政策會是更好的方法。

針對 `Principal`，指定您使用 the section called “關於身分集區” 中的準則設定之 Cognito 經身分驗證角色的 ARN。

JSON

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

```
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":[
               "arn:aws:iam::111122223333:role/Cognito_identitypoolname/
Auth_Role"
            ]
         },
         "Action":[
            "es:ESHttp*"
         ],
         "Resource":"arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
      }
   ]
}
```

如需有關將資源型政策新增到 OpenSearch Service 網域的說明，請參閱the section called “設定存取政策”。

設定身分提供者

當您設定網域以使用 Dashboards 的 Amazon Cognito 身分驗證時，OpenSearch Service 會將應用程式用戶端新增到使用者集區，並將使用者集區作為身分驗證供應商新增到身分集區。

> **Warning**
>
> 不重新命名或刪除應用程式用戶端。

根據您設定使用者集區的方式而定，您可能需要手動建立使用者帳戶，或者使用者可以建立自己的帳戶。如果這些設定是可接受的，您便無需做進一步的動作。不過，許多人偏好使用外部身分提供者。

若要啟用 SAML 2.0 身分提供者，您必須提供 SAML 中繼資料文件。若要啟用社交身分提供者，例如，Login with Amazon、Facebook 和 Google，您必須擁有這些供應商提供的應用程式 ID 和應用程式密碼。您可以啟用任意組合的身分提供者。

設定使用者集區的最簡單方式就是使用 Amazon Cognito 主控台。如需說明，請參閱《Amazon Cognito 開發人員指南》中的使用第三方身分提供者的使用者集區登入和使用應用程式用戶端的應用程式特定設定。

(選用) 設定精細分級的存取

您可能已注意到預設身分集區設定會指派登入相同 IAM 角色 (Cognito_*identitypool*Auth_Role) 的每個使用者，這表示每個使用者可以存取相同的 AWS 資源。如果您想要搭配使用精細存取控制與 Amazon Cognito (例如，如果希望您組織的分析師擁有多個索引的唯讀存取權，而開發人員擁有所有索引的寫入存取權)，您有兩個選擇：

- 建立使用者群組並設定您的身分提供者，以根據使用者的身分驗證字符選擇 IAM 角色 (建議)。
- 設定您的身分提供者，以根據一個或多個規則選擇 IAM 角色。

如需包含精細存取控制的演練，請參閱 the section called “教學課程：使用 Cognito 身分驗證進行精細存取控制”。

> **Important**
>
> 與預設角色相似，Amazon Cognito 必須是每個額外角色信任關係的一部分。如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的為角色映射建立角色。

使用者群組和字符

當您建立使用者群組時，您為群組成員選擇 IAM 角色。如需建立群組的相關資訊，請參閱《Amazon Cognito 開發人員指南》中的將群組新增至使用者集區。

在您建立一或多個使用者群組後，您可以設定您的驗證供應商，以指派使用者其群組的角色，而不是身分集區的預設角色。選取 Choose role from token (從字符中選擇角色)，然後選擇 Use default Authenticated role (使用預設已經過身分驗證的角色) 或 DENY (拒絕)，以指定身分集區應該如何處理不屬於群組的使用者。

規則

規則本質上是 Amazon Cognito 依序評估的一系列 `if` 陳述式。例如，如果使用者的電子郵件地址包含 `@corporate`，Amazon Cognito 會為該使用者指派 `Role_A`。如果使用者的電子郵件地址包含 `@subsidiary`，它會指派該使用者 `Role_B`。否則，它會指派使用者預設驗證角色。

若要進一步了解，請參閱《Amazon Cognito 開發人員指南》中的使用規則型映射將角色指派給使用者。

(選用) 自訂登入頁面

您可以使用 Amazon Cognito 主控台上傳自訂標誌，並對登入頁面進行 CSS 變更。如需 CSS 屬性的說明和完整清單，請參閱《Amazon Cognito 開發人員指南》中的自訂託管 UI（傳統）品牌。

(選用) 設定進階安全性

Amazon Cognito 使用者集區支援進階安全功能，例如多重驗證、遭盜用認證檢查和調整式驗證。若要進一步了解，請參閱《Amazon Cognito 開發人員指南》中的使用 Amazon Cognito 使用者集區安全功能。 Amazon Cognito

測試

在您滿意您的設定之後，請驗證使用者體驗是否符合您的期望。

若要存取 OpenSearch Dashboards

1. 在 Web 瀏覽器中導覽至 `https://opensearch-domain/_dashboards`。若要直接登入特定租用戶，請將 `?security_tenant=tenant-name` 附加至 URL。
2. 使用您慣用的登入資料登入。
3. 載入 OpenSearch Dashboards 之後，至少設定一個索引模式。Dashboards 使用這些模式來識別您要分析的索引。輸入 *，選擇 Next step (下一步)，然後選擇 Create index pattern (建立索引模式)。
4. 若要搜尋或探索您的資料，請選擇 Discover (探索)。

如果此程序中的任何步驟失敗，請參閱the section called “常見的設定問題”以取得故障診斷資訊。

配額

Amazon Cognito 對於許多資源有軟性限制。如果您想要為大量的 OpenSearch Service 網域啟用 Dashboards 身分驗證，請視需要檢閱 Amazon Cognito 中的配額並請求提高限制。

每個 OpenSearch Service 網域會新增應用程式用戶端到使用者集區，這會新增身分驗證供應商到身分集區。如果您對 10 個以上的網域啟用 OpenSearch Dashboards 身分驗證，您可能會遇到「每個身分集區的 Amazon Cognito 使用者集區供應商上限」限制。如果超出限制，您嘗試設定以使用 Dashboards 的 Amazon Cognito 身分驗證的任何 OpenSearch Service 網域可能凍結在 Processing (處理) 的設定狀態中。

常見的設定問題

下表列出常見的設定問題和解決方案。

設定 OpenSearch Service

問題	解決方案
`OpenSearch Service can't create the role (主控台)`	您沒有正確的 IAM 許可。新增 the section called “設定 Amazon Cognito 身分驗證 (主控台)” 中指定的許可。
`User is not authorized to perform: iam:PassRole on resource CognitoAccessForAmazonOpenSearch (主控台)`	您沒有 CognitoAccessForAmazonOpenSearch 角色的 `iam:PassRole` 許可。將下列政策連接至您的帳戶： JSON `{ "Version":"2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iam:PassRole" ], "Resource": "arn:aws:iam:: 123456789012:role/service-role/CognitoAccessForAmazonOpenSearch " } ] }` 或者，您可以連接 `IAMFullAccess` 政策。
`User is not authorized to perform: cognito-identity:ListIdentityPools on resource`	您沒有 Amazon Cognito 的讀取許可。將 `AmazonCognitoReadOnly` 政策連接至您的帳戶。
`An error occurred (ValidationException) when calling`	並未在 `CognitoAccessForAmazonOpenSearch` 角色的信任關係中指定 OpenSearch Service。確認您的

問題	解決方案
`the CreateDomain operation : OpenSearch Service must be allowed to use the passed role`	角色使用 the section called “關於 CognitoAccessForAmazonOpenSearch 角色” 中指定的信任關係。或者，使用主控台來設定 Amazon Cognito 身分驗證。主控台會為您建立一個角色。
`An error occurred (ValidationException) when calling the CreateDomain operation : User is not authorized to perform: cognito-idp: action on resource: user pool`	在 `--cognito-options` 中指定的角色沒有存取 Amazon Cognito 的許可。檢查角色是否已連接 AWS 受管`AmazonOpenSearchServiceCognitoAccess` 政策。或者，使用主控台來設定 Amazon Cognito 身分驗證。主控台會為您建立一個角色。
`An error occurred (ValidationException) when calling the CreateDomain operation : User pool does not exist`	OpenSearch Service 找不到使用者集區。確認您已建立一個且具有正確的 ID。若要尋找 ID，您可以使用 Amazon Cognito 主控台或下列 AWS CLI 命令： `aws cognito-idp list-user-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateDomain operation : IdentityPool not found`	OpenSearch Service 找不到身分集區。確認您已建立一個且具有正確的 ID。若要尋找 ID，您可以使用 Amazon Cognito 主控台或下列 AWS CLI 命令： `aws cognito-identity list-identity-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateDomain operation : Domain needs to be specified for user pool`	使用者集區沒有網域名稱。您可以使用 Amazon Cognito 主控台或以下 AWS CLI 命令來設定一個： `aws cognito-idp create-user-pool-domain --domain name --user-pool-id id`

存取 OpenSearch Dashboards

問題	解決方案
登入頁面不會顯示我慣用的身分提供者。	確認您已為 OpenSearch Service 應用程式用戶端啟用 the section called “設定身分提供者” 中指定的身分提供者。
登入頁面看起來不似與我的組織關聯。	請參閱 the section called “(選用) 自訂登入頁面”。
我的登入資料無法運作。	確認您已如 the section called “設定身分提供者” 中所指定，來設定身分提供者。 如果您使用使用者集區做為身分提供者，請檢查帳戶是否存在於 Amazon Cognito 主控台。
OpenSearch Dashboards 完全不載入，或無法正常運作。	Amazon Cognito 經過身分驗證的角色需要網域 (/*) 的 `es:ESHttp*` 許可來存取和使用 Dashboards。如 the section called “允許已經過身分驗證的角色” 中所指定，檢查您新增的存取政策。
當我從一個索引標籤登出 OpenSearch Dashboards 時，剩餘的索引標籤會顯示一則訊息，指出重新整理字符已被撤銷。	當您在使用 Amazon Cognito 身分驗證時登出 OpenSearch Dashboards 工作階段時，OpenSearch Service 會執行 AdminUserGlobalSignOut 操作，該操作會將您登出所有作用中的 OpenSearch Dashboards 工作階段。
`Invalid identity pool configuration. Check assigned IAM roles for this pool.`	Amazon Cognito 無權代表已經過身分驗證的使用者擔任 IAM 角色。修改角色的信任關係，使其包含： JSON `{ "Version":"2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Federated": "cognito-identity.amazonaws.com" },`

問題	解決方案

	"Action": "sts:AssumeRoleWithWebIdentity",
	"Condition": {
	"StringEquals": {
	"cognito-identity.amazonaws.com:aud": " *identity-pool-id* "
	},
	"ForAnyValue:StringLike": {
	"cognito-identity.amazonaws.com:amr": "authenticated"
	}
	}
	}
	]
	}

`Token is not from a supported provider of this identity pool.`	當您從使用者集區移除應用程式用戶端時，可能發生此少見的錯誤。嘗試在新的瀏覽器工作階段中開啟 Dashboards。

停用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證

使用下列步驟來停用 Dashboards 的 Amazon Cognito 身分驗證。

若要停用 Dashboards 的 Amazon Cognito 身分驗證 (主控台)

1. 開啟 Amazon OpenSearch Service 主控台。
2. 在 Domains (網域) 下，選擇您要設定的網域。
3. 選擇 Actions (動作)、Edit security configuration (編輯安全組態)。
4. 取消選擇 Enable Amazon Cognito authentication (啟用 Amazon Cognito 身分驗證)。
5. 選擇儲存變更。

> **Important**
>
> 如果您不再需要 Amazon Cognito 使用者集區和身分集區，請刪除它們。否則，您需繼續負擔費用。

刪除使用 OpenSearch Dashboards 的 Amazon Cognito 身分驗證的網域

為避免使用 Dashboards 的 Amazon Cognito 身分驗證的網域凍結在 Processing (處理) 的設定狀態，請在刪除其相關的 Amazon Cognito 使用者和身分集區之前，刪除 OpenSearch Service 網域。

使用 Amazon OpenSearch Service 的服務連結角色

Amazon OpenSearch Service 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 OpenSearch Service 的一種特殊 IAM 角色類型。服務連結角色是由 OpenSearch Service 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓設定 OpenSearch Service 更為簡單，因為您不必手動新增必要的許可。OpenSearch Service 定義其服務連結角色的許可，除非另有定義，否則僅 OpenSearch Service 可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。如需服務連結角色和許可政策的更新，請參閱 Amazon OpenSearch Service 的文件歷史記錄。

如需有關支援服務連結角色的其他 服務的資訊，請參閱AWS 使用 IAM 的服務，並在服務連結角色欄中尋找具有是的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

主題

- 使用服務連結角色來建立 VPC 網域和直接查詢資料來源
- 使用服務連結角色建立 OpenSearch Serverless 集合
- 使用服務連結角色建立 OpenSearch Ingestion 管道

使用服務連結角色來建立 VPC 網域和直接查詢資料來源

Amazon OpenSearch Service 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 OpenSearch Service 的一種特殊 IAM 角色類型。服務連結角色由 OpenSearch Service 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

OpenSearch Service 使用名為 AWSServiceRoleForAmazonOpenSearchService 的服務連結角色，可提供角色啟用網域或直接查詢資料來源 VPC 存取所需的最低 Amazon EC2 和 Elastic Load Balancing 許可。

舊版 Elasticsearch 角色

Amazon OpenSearch Service 使用名為 `AWSServiceRoleForAmazonOpenSearchService` 的服務連結角色。您的帳戶可能也會包含名為 `AWSServiceRoleForAmazonElasticsearchService` 的舊版服務連結角色，其使用已被淘汰的 Elasticsearch API 端點運作。

如果您的帳戶中沒有舊版的 Elasticsearch 角色，則 OpenSearch Service 會在您首次建立 OpenSearch 網域時，自動建立一個新的 OpenSearch 服務連結角色。否則，您的帳戶將繼續使用 Elasticsearch 角色。若要讓此自動建立作業順利完成，您必須具有 `iam:CreateServiceLinkedRole` 動作的許可。

許可

`AWSServiceRoleForAmazonOpenSearchService` 服務連結角色信任下列服務以擔任角色：

- `opensearchservice.amazonaws.com`

名為 `AmazonOpenSearchServiceRolePolicy` 的角色許可政策允許 OpenSearch Service 對指定資源完成下列動作：

- 動作：* 上的 `acm:DescribeCertificate`
- 動作：* 上的 `cloudwatch:PutMetricData`
- 動作：* 上的 `ec2:CreateNetworkInterface`
- 動作：* 上的 `ec2:DeleteNetworkInterface`
- 動作：* 上的 `ec2:DescribeNetworkInterfaces`
- 動作：* 上的 `ec2:ModifyNetworkInterfaceAttribute`
- 動作：* 上的 `ec2:DescribeSecurityGroups`
- 動作：* 上的 `ec2:DescribeSubnets`
- 動作：* 上的 `ec2:DescribeVpcs`
- 動作：在所有網路介面和 VPC 端點進行 `ec2:CreateTags`
- 動作：* 上的 `ec2:DescribeTags`

- 動作：當請求包含標籤 `OpenSearchManaged=true` 時，在所有 VPC、安全群組、子網路和路由表，以及所有 VPC 端點上進行 `ec2:CreateVpcEndpoint`
- 動作：當請求包含標籤 `OpenSearchManaged=true` 時，在所有 VPC、安全群組、子網路和路由表，以及所有 VPC 端點上進行 `ec2:ModifyVpcEndpoint`
- 動作：當請求包含標籤 `OpenSearchManaged=true` 時，在所有端點上進行 `ec2:DeleteVpcEndpoints`
- 動作：`*` 上的 `ec2:AssignIpv6Addresses`
- 動作：`*` 上的 `ec2:UnAssignIpv6Addresses`
- 動作：`*` 上的 `elasticloadbalancing:AddListenerCertificates`
- 動作：`*` 上的 `elasticloadbalancing:RemoveListenerCertificates`

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

建立 服務連結角色

您不需要手動建立服務連結角色，當您使用 建立已啟用 VPC 的網域或直接查詢資料來源時 AWS 管理主控台，OpenSearch Service 會為您建立服務連結角色。若要讓此自動建立作業順利完成，您必須具有 `iam:CreateServiceLinkedRole` 動作的許可。

您也可以使用 IAM 主控台、IAM CLI 或 IAM API 來手動建立服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的建立服務連結角色。

編輯服務連結角色

OpenSearch Service 不允許您編輯 `AWSServiceRoleForAmazonOpenSearchService` 服務連結角色。因為可能有各種實體會參考服務連結角色，所以您無法在建立角色之後變更其名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的編輯服務連結角色。

刪除 服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，務必清除您的服務連結角色，之後才能以手動方式將其刪除。

清除 服務連結角色

您必須先確認服務連結角色沒有作用中的工作階段，並移除該角色使用的資源，之後才能使用 IAM 將其刪除。

檢查服務連結角色是否於 IAM 主控台有作用中的工作階段

1. 登入 AWS 管理主控台 並開啟位於 https：//https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇角色。然後選擇 `AWSServiceRoleForAmazonOpenSearchService` 角色的名稱 (而非核取方塊)。
3. 在所選角色的 Summary (摘要) 頁面中，選擇 Access Advisor (存取 Advisor) 分頁。
4. 在 Access Advisor (存取 Advisor) 分頁中，檢閱服務連結角色的近期活動。

> **Note**
>
> 如果您不確定 OpenSearch Service 是否正在使用 `AWSServiceRoleForAmazonOpenSearchService` 角色，可嘗試刪除該角色。如果服務正在使用該角色，則刪除會失敗，而您可以檢視正在使用該角色的資源。如果角色正在使用中，則您必須先等到工作階段結束，才能刪除該角色，及/或刪除正在使用該角色的資源。您無法撤銷服務連結角色的工作階段。

手動刪除服務連結角色

從 IAM 主控台、API 或 CLI AWS 刪除服務連結角色。如需相關說明，請參閱 IAM 使用者指南中的刪除服務連結角色。

使用服務連結角色建立 OpenSearch Serverless 集合

OpenSearch Serverless 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 OpenSearch Service 的一種特殊 IAM 角色類型。服務連結角色由 OpenSearch Service 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

OpenSearch Serverless 使用名為 AWSServiceRoleForAmazonOpenSearchServerless 的服務連結角色，提供角色將無伺服器相關 CloudWatch 指標發佈至您的帳戶所需的許可。與 AWSServiceRoleForAmazonOpenSearchServerless 相關聯的角色許可政策名為 `AmazonOpenSearchServerlessServiceRolePolicy`。如需政策的詳細資訊，請參閱《 AWS 受管政策參考指南》中的 AmazonOpenSearchServerlessServiceRolePolicy。

OpenSearch Serverless 的服務連結角色許可

OpenSearch Serverless 使用名為 AWSServiceRoleForAmazonOpenSearchServerless 的服務連結角色，這可讓 OpenSearch Serverless 代表您呼叫 AWS 服務。

AWSServiceRoleForAmazonOpenSearchServerless 服務連結角色信任下列服務擔任該角色：

- observability.aoss.amazonaws.com

名為 AmazonOpenSearchServerlessServiceRolePolicy 的角色許可政策允許 OpenSearch Serverless 對指定資源完成下列動作：

- 動作：cloudwatch:PutMetricData 於所有 AWS 資源。

> **Note**
>
> 政策包括條件條件索引鍵 {"StringEquals": {"cloudwatch:namespace": "AWS/AOSS"}}，而這表示服務連結角色僅可傳送指標資料至 AWS/AOSS CloudWatch 命名空間。

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

建立 OpenSearch Serverless 的服務連結角色

您不需要手動建立服務連結角色，當您在 AWS 管理主控台、 AWS CLI或 AWS API 中建立 OpenSearch Serverless 集合時，OpenSearch Serverless 會為您建立服務連結角色。

> **Note**
>
> 第一次建立集合時，必須在以身分為基礎的政策中指派 iam:CreateServiceLinkedRole 給您。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。建立 OpenSearch Serverless 集合時，OpenSearch Serverless 可再次為您建立服務連結角色。

您也可以使用 IAM 主控台，透過 Amazon OpenSearch Serverless 使用案例建立服務連結角色。在 AWS CLI 或 AWS API 中，使用服務名稱建立observability.aoss.amazonaws.com服務連結角色：

```
aws iam create-service-linked-role --aws-service-name
 "observability.aoss.amazonaws.com"
```

如需詳細資訊，請參閱《IAM 使用者指南》中的「建立服務連結角色」。如果您刪除此服務連結角色，您可以使用此相同的程序以再次建立該角色。

編輯 OpenSearch Serverless 的服務連結角色

OpenSearch Serverless 不允許您編輯 AWSServiceRoleForAmazonOpenSearchServerless 服務連結角色。因為可能有各種實體會參考服務連結角色，所以您無法在建立角色之後變更其名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的編輯服務連結角色。

刪除 OpenSearch Serverless 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。這會防止您擁有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先清除資源。

若要刪除 AWSServiceRoleForAmazonOpenSearchServerless，您必須先刪除 中的所有 OpenSearch Serverless 集合 AWS 帳戶。

> **Note**
>
> 如果 OpenSearch Serverless 在您嘗試刪除資源時正在使用該角色，則刪除動作可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForAmazonOpenSearchServerless 服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

OpenSearch Serverless 服務連結角色的支援區域

OpenSearch Serverless 支援在可使用 OpenSearch Serverless 的每個區域中使用 AWSServiceRoleForAmazonOpenSearchServerless 服務連結角色。 OpenSearch 如需支援區域的清單，請參閱《》中的 Amazon OpenSearch Serverless 端點和配額AWS 一般參考。

使用服務連結角色建立 OpenSearch Ingestion 管道

Amazon OpenSearch Ingestion 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 OpenSearch Ingestion 的唯一 IAM 角色類型。服務連結角色由 OpenSearch Ingestion 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

OpenSearch Ingestion 使用名為 AWSServiceRoleForAmazonOpenSearchIngestionService 的服務連結角色，除非您使用自我管理 VPC，在這種情況下，它使用名為 AWSServiceRoleForOpensearchIngestionSelfManagedVpce 的服務連結角色。連接的政策提供角色在您的帳戶和 OpenSearch Ingestion 之間建立虛擬私有雲端 (VPC)，以及將 CloudWatch 指標發佈至您的帳戶所需的許可。

許可

AWSServiceRoleForAmazonOpenSearchIngestionService 服務連結角色信任下列服務以擔任角色：

- osis.amazonaws.com

名為 的角色許可政策AmazonOpenSearchIngestionServiceRolePolicy允許 OpenSearch Ingestion 對指定的資源完成下列動作：

- 動作：cloudwatch:namespace": "AWS/OSIS" 上的 cloudwatch:PutMetricData
- 動作：arn:aws:ec2:*:*:network-interface/* 上的 ec2:CreateTags
- 動作：* 上的 ec2:CreateVpcEndpoint
- 動作：* 上的 ec2:DeleteVpcEndpoints
- 動作：* 上的 ec2:DescribeSecurityGroups
- 動作：* 上的 ec2:DescribeSubnets
- 動作：* 上的 ec2:DescribeVpcEndpoints
- 動作：* 上的 ec2:ModifyVpcEndpoint

AWSServiceRoleForOpensearchIngestionSelfManagedVpce 服務連結角色信任下列服務以擔任角色：

- self-managed-vpce.osis.amazonaws.com

名為 的角色許可政策OpenSearchIngestionSelfManagedVpcePolicy允許 OpenSearch Ingestion 對指定的資源完成下列動作：

- 動作：* 上的 ec2:DescribeSubnets
- 動作：* 上的 ec2:DescribeSecurityGroups
- 動作：* 上的 ec2:DescribeVpcEndpoints

- 動作：cloudwatch:namespace": "AWS/OSIS" 上的 cloudwatch:PutMetricData

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

建立 OpenSearch Ingestion 的服務連結角色

您不需要手動建立服務連結角色，當您在 AWS 管理主控台、 AWS CLI或 AWS API 中建立 OpenSearch Ingestion 管道時，OpenSearch Ingestion 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立 OpenSearch Ingestion 管道時，OpenSearch Ingestion 會再次為您建立服務連結角色。

編輯 OpenSearch Ingestion 的服務連結角色

OpenSearch Ingestion 不允許您編輯AWSServiceRoleForAmazonOpenSearchIngestionService服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的編輯服務連結角色。

刪除 OpenSearch Ingestion 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先清除資源。

清除服務連結角色

在您使用 IAM 刪除服務連結角色之前，您必須先刪除該角色所使用的任何資源。

> **Note**
>
> 如果 OpenSearch Ingestion 在您嘗試刪除資源時使用角色，則刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

刪除 **AWSServiceRoleForAmazonOpenSearchIngestionService**或 **AWSServiceRoleForOpensearchIngestionSelfManagedVpce**角色使用的 OpenSearch Ingestion 資源

1. 導覽至 Amazon OpenSearch Service 主控台，然後選擇擷取。

2. 刪除所有管道。如需說明，請參閱the section called “刪除管道”。

刪除 OpenSearch Ingestion 的服務連結角色

您可以使用 OpenSearch Ingestion 主控台來刪除服務連結角色。

刪除服務連結角色 (主控台)

1. 導覽至 IAM 主控台。
2. 選擇角色並搜尋 AWSServiceRoleForAmazonOpenSearchIngestionService 或 AWSServiceRoleForOpensearchIngestionSelfManagedVpce 角色。
3. 選取角色，然後選擇刪除。

Amazon OpenSearch Service 的範本程式碼

本章包含使用 Amazon OpenSearch Service 的常見範本程式碼：以各種程式設計語言簽署 HTTP 請求、壓縮 HTTP 請求主體以及使用 AWS 開發套件來建立網域。

主題

- Elasticsearch 用戶端相容性
- 在 Amazon OpenSearch Service 中壓縮 HTTP 請求
- 使用 AWS SDKs與 Amazon OpenSearch Service 互動

Elasticsearch 用戶端相容性

Elasticsearch 用戶端的最新版本可能包含會人為破壞相容性的授權或版本檢查。下表包含有關使用哪些用戶端版本以達到與 OpenSearch Service 最佳相容的建議。

> **Important**
>
> 這些用戶端版本已過時，且不會使用最新相依性 (包括 Log4j) 進行更新。我們強烈建議盡可能使用用戶端的 OpenSearch 版本。

用戶端	建議的版本
Java 低階 REST 用戶端	7.13.4
Java 高階 REST 用戶端	7.13.4
Python Elasticsearch 用戶端	7.13.4
Ruby Elasticsearch 用戶端	7.13.3
Node.js Elasticsearch 用戶端	7.13.0

在 Amazon OpenSearch Service 中壓縮 HTTP 請求

您可以使用 gzip 壓縮來壓縮 Amazon OpenSearch Service 網域中的 HTTP 請求和回應。Gzip 壓縮可以幫助您減少文件的大小，降低頻寬使用率和延遲，進而提高傳輸速度。

執行 OpenSearch 或 Elasticsearch 6.0 或更高版本的所有網域都支援 Gzip 壓縮。有些 OpenSearch 用戶端內建支援 gzip 壓縮，許多程式設計語言都有可簡化該程序的程式庫。

啟用 gzip 壓縮

不要與類似的 OpenSearch 設定混淆，`http_compression.enabled` 專門針對 OpenSearch Service，可以啟用或停用網域上的 gzip 壓縮功能。執行 OpenSearch 或 Elasticsearch 7.x 的網域預設啟用了 gzip 壓縮功能，而執行 Elasticsearch 6.x 的網域預設停用了該功能。

若要啟用 gzip 壓縮，請傳送以下請求：

```
PUT _cluster/settings
{
  "persistent" : {
    "http_compression.enabled": true
  }
}
```

對 `_cluster/settings` 的請求必須解壓縮，因此您可能需要使用單獨的用戶端或標準 HTTP 請求來更新叢集設定。

若要確認您已成功啟用 gzip 壓縮，請傳送下列請求：

```
GET _cluster/settings?include_defaults=true
```

請確定您在回應中看到下列設定：

```
...
"http_compression": {
  "enabled": "true"
}
...
```

必要標頭

當包含 gzip 壓縮的要求主體時，請保留標準 `Content-Type: application/json` 標頭，並新增 `Content-Encoding: gzip` 標頭。若要接受 gzip 壓縮的回應，請也新增 `Accept-Encoding: gzip` 標頭。如果 OpenSearch 用戶端支援 gzip 壓縮，它可能會自動包含這些標頭。

範本程式碼 (Python 3)

下列範例使用 opensearch-py 來執行壓縮並傳送請求。此程式碼使用您的 IAM 憑證來簽署請求。

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3

host = '' # e.g. my-test-domain.us-east-1.es.amazonaws.com
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Create the client.
search = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    http_compress = True, # enables gzip compression for request bodies
    connection_class = RequestsHttpConnection
)

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

# Send the request.
print(search.index(index='movies', id='1', body=document, refresh=True))

# print(search.index(index='movies', doc_type='_doc', id='1', body=document,
 refresh=True))
```

或者，您可以指定適當的標頭，自己壓縮要求主體，並使用標準的 HTTP 程式庫，例如請求。此程式碼使用 HTTP 基本憑證來簽署請求，如果您使用精細存取控制，則您的網域可能會支援。

```
import requests
import gzip
import json

base_url = '' # The domain with https:// and a trailing slash. For example, https://my-test-domain.us-east-1.es.amazonaws.com/
auth = ('master-user', 'master-user-password') # For testing only. Don't store credentials in code.

headers = {'Accept-Encoding': 'gzip', 'Content-Type': 'application/json',
           'Content-Encoding': 'gzip'}

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

# Compress the document.
compressed_document = gzip.compress(json.dumps(document).encode())

# Send the request.
path = 'movies/_doc?refresh=true'
url = base_url + path
response = requests.post(url, auth=auth, headers=headers, data=compressed_document)
print(response.status_code)
print(response.text)
```

使用 AWS SDKs與 Amazon OpenSearch Service 互動

本節包含如何使用 AWS SDKs 與 Amazon OpenSearch Service 組態 API 互動的範例。這些程式碼範例顯示如何建立、更新和刪除 OpenSearch Service 網域。

Java

本節包含 第 1 版和第 2 版的範例 適用於 Java 的 AWS SDK。

Version 2

此範例使用 適用於 Java 的 AWS SDK 版本 2 中的 OpenSearchClientBuilder 構造函數來建立 OpenSearch 網域、更新其組態，然後將其刪除。取消對 `waitForDomainProcessing` 呼叫的註解 (並對 `deleteDomain` 呼叫加上註解)，以允許網域上線並變得可用。

```
package com.example.samples;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.opensearch.OpenSearchClient;
import software.amazon.awssdk.services.opensearch.model.ClusterConfig;
import software.amazon.awssdk.services.opensearch.model.EBSOptions;
import software.amazon.awssdk.services.opensearch.model.CognitoOptions;
import software.amazon.awssdk.services.opensearch.model.NodeToNodeEncryptionOptions;
import software.amazon.awssdk.services.opensearch.model.CreateDomainRequest;
import software.amazon.awssdk.services.opensearch.model.CreateDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainRequest;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainResponse;
import software.amazon.awssdk.services.opensearch.model.OpenSearchException;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */

public class OpenSearchSample {

    public static void main(String[] args) {

     String domainName = "my-test-domain";

     // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.

        OpenSearchClient client = OpenSearchClient.builder()
```

```
          // Unnecessary, but lets you use a region different than your default.
          .region(Region.US_EAST_1)
          // Unnecessary, but if desired, you can use a different provider chain.
          .credentialsProvider(DefaultCredentialsProvider.create())
                    .build();

       // Create a new domain, update its configuration, and delete it.
       createDomain(client, domainName);
       //waitForDomainProcessing(client, domainName);
       updateDomain(client, domainName);
       //waitForDomainProcessing(client, domainName);
       deleteDomain(client, domainName);
   }

   /**
    * Creates an Amazon OpenSearch Service domain with the specified options.
    * Some options require other Amazon Web Services resources, such as an Amazon
Cognito user pool
    * and identity pool, whereas others require just an instance type or instance
    * count.
    *
    * @param client
    *            The client to use for the requests to Amazon OpenSearch Service
    * @param domainName
    *            The name of the domain you want to create
    */

   public static void createDomain(OpenSearchClient client, String domainName) {

    // Create the request and set the desired configuration options

       try {

           ClusterConfig clusterConfig = ClusterConfig.builder()
                   .dedicatedMasterEnabled(true)
                   .dedicatedMasterCount(3)
                   // Small, inexpensive instance types for testing. Not
recommended for production.
                   .dedicatedMasterType("t2.small.search")
                   .instanceType("t2.small.search")
                   .instanceCount(5)
                   .build();

           // Many instance types require EBS storage.
```

```
            EBSOptions ebsOptions = EBSOptions.builder()
                    .ebsEnabled(true)
                    .volumeSize(10)
                    .volumeType("gp2")
                    .build();

            NodeToNodeEncryptionOptions encryptionOptions =
 NodeToNodeEncryptionOptions.builder()
                    .enabled(true)
                    .build();

            CreateDomainRequest createRequest = CreateDomainRequest.builder()
                    .domainName(domainName)
                    .engineVersion("OpenSearch_1.0")
                    .clusterConfig(clusterConfig)
                    .ebsOptions(ebsOptions)
                    .nodeToNodeEncryptionOptions(encryptionOptions)
                    // You can uncomment this line and add your account ID, a
 username, and the
                    // domain name to add an access policy.
                    // .accessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")
                    .build();

            // Make the request.
            System.out.println("Sending domain creation request...");
            CreateDomainResponse createResponse =
 client.createDomain(createRequest);
            System.out.println("Domain status:
 "+createResponse.domainStatus().toString());
            System.out.println("Domain ID:
 "+createResponse.domainStatus().domainId());


        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Updates the configuration of an Amazon OpenSearch Service domain with the
```

```
    * specified options. Some options require other Amazon Web Services resources,
such as an
    * Amazon Cognito user pool and identity pool, whereas others require just an
    * instance type or instance count.
    *
    * @param client
    *            The client to use for the requests to Amazon OpenSearch Service
    * @param domainName
    *            The name of the domain to update
    */

   public static void updateDomain(OpenSearchClient client, String domainName) {

    // Updates the domain to use three data instances instead of five.
       // You can uncomment the Cognito line and fill in the strings to enable
Cognito
       // authentication for OpenSearch Dashboards.

       try {

           ClusterConfig clusterConfig = ClusterConfig.builder()
                   .instanceCount(5)
                   .build();

           CognitoOptions cognitoOptions = CognitoOptions.builder()
                   .enabled(true)
                   .userPoolId("user-pool-id")
                   .identityPoolId("identity-pool-id")
                   .roleArn("role-arn")
                   .build();

           UpdateDomainConfigRequest updateRequest =
UpdateDomainConfigRequest.builder()
                   .domainName(domainName)
                   .clusterConfig(clusterConfig)
                   //.cognitoOptions(cognitoOptions)
                   .build();

           System.out.println("Sending domain update request...");
           UpdateDomainConfigResponse updateResponse =
client.updateDomainConfig(updateRequest);
           System.out.println("Domain config:
"+updateResponse.domainConfig().toString());
```

```
        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
     * several minutes.
     *
     * @param client
     *             The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *             The name of the domain that you want to delete
     */

    public static void deleteDomain(OpenSearchClient client, String domainName) {

        try {

            DeleteDomainRequest deleteRequest = DeleteDomainRequest.builder()
                    .domainName(domainName)
                    .build();

            System.out.println("Sending domain deletion request...");
            DeleteDomainResponse deleteResponse =
client.deleteDomain(deleteRequest);
            System.out.println("Domain status: "+deleteResponse.toString());


        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Waits for the domain to finish processing changes. New domains typically take
15-30 minutes
     * to initialize, but can take longer depending on the configuration. Most
updates to existing domains
     * take a similar amount of time. This method checks every 15 seconds and
finishes only when
```

```
     * the domain's processing status changes to false.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to check
     */

    public static void waitForDomainProcessing(OpenSearchClient client, String
 domainName) {
        // Create a new request to check the domain status.
        DescribeDomainRequest describeRequest = DescribeDomainRequest.builder()
                .domainName(domainName)
                .build();

        // Every 15 seconds, check whether the domain is processing.
        DescribeDomainResponse describeResponse =
 client.describeDomain(describeRequest);
        while (describeResponse.domainStatus().processing()) {
            try {
                System.out.println("Domain still processing...");
                TimeUnit.SECONDS.sleep(15);
                describeResponse = client.describeDomain(describeRequest);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        // Once we exit that loop, the domain is available
        System.out.println("Amazon OpenSearch Service has finished processing
 changes for your domain.");
        System.out.println("Domain description: "+describeResponse.toString());
    }
}
```

Version 1

此範例使用 AWSElasticsearchClientBuilder 構造函數，從 適用於 Java 的 AWS SDK 的版本 1 中建立舊式 Elasticsearch 網域、更新其組態，然後將其刪除。取消對 `waitForDomainProcessing` 呼叫的註解 (並對 `deleteDomain` 呼叫加上註解)，以允許網域上線並變得可用。

```
package com.amazonaws.samples;
```

```
import java.util.concurrent.TimeUnit;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticsearch.AWSElasticsearch;
import com.amazonaws.services.elasticsearch.AWSElasticsearchClientBuilder;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainResult;
import
 com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.EBSOptions;
import com.amazonaws.services.elasticsearch.model.ElasticsearchClusterConfig;
import com.amazonaws.services.elasticsearch.model.ResourceNotFoundException;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigRequest;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigResult;
import com.amazonaws.services.elasticsearch.model.VolumeType;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */

public class OpenSearchSample {

    public static void main(String[] args) {

        final String domainName = "my-test-domain";

        // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.
        final AWSElasticsearch client = AWSElasticsearchClientBuilder
                .standard()
                // Unnecessary, but lets you use a region different than your
 default.
                .withRegion(Regions.US_WEST_2)
                // Unnecessary, but if desired, you can use a different provider
 chain.
```

```
                .withCredentials(new DefaultAWSCredentialsProviderChain())
                .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon OpenSearch Service domain with the specified options.
     * Some options require other Amazon Web Services resources, such as an Amazon
Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain you want to create
     */
    private static void createDomain(final AWSElasticsearch client, final String
domainName) {

        // Create the request and set the desired configuration options
        CreateElasticsearchDomainRequest createRequest = new
CreateElasticsearchDomainRequest()
                .withDomainName(domainName)
                .withElasticsearchVersion("7.10")
                .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                        .withDedicatedMasterEnabled(true)
                        .withDedicatedMasterCount(3)
                        // Small, inexpensive instance types for testing. Not
recommended for production
                        // domains.
                        .withDedicatedMasterType("t2.small.elasticsearch")
                        .withInstanceType("t2.small.elasticsearch")
                        .withInstanceCount(5))
                // Many instance types require EBS storage.
                .withEBSOptions(new EBSOptions()
                        .withEBSEnabled(true)
                        .withVolumeSize(10)
```

```
                        .withVolumeType(VolumeType.Gp2));
                // You can uncomment this line and add your account ID, a username,
 and the
                // domain name to add an access policy.
                // .withAccessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")


        // Make the request.
        System.out.println("Sending domain creation request...");
        CreateElasticsearchDomainResult createResponse =
 client.createElasticsearchDomain(createRequest);
        System.out.println("Domain creation response from Amazon OpenSearch
 Service:");
        System.out.println(createResponse.getDomainStatus().toString());
    }

    /**
     * Updates the configuration of an Amazon OpenSearch Service domain with the
     * specified options. Some options require other Amazon Web Services resources,
 such as an
     * Amazon Cognito user pool and identity pool, whereas others require just an
     * instance type or instance count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain to update
     */
    private static void updateDomain(final AWSElasticsearch client, final String
 domainName) {
        try {
            // Updates the domain to use three data instances instead of five.
            // You can uncomment the Cognito lines and fill in the strings to enable
 Cognito
            // authentication for OpenSearch Dashboards.
            final UpdateElasticsearchDomainConfigRequest updateRequest = new
 UpdateElasticsearchDomainConfigRequest()
                    .withDomainName(domainName)
                    // .withCognitoOptions(new CognitoOptions()
                            // .withEnabled(true)
                            // .withUserPoolId("user-pool-id")
```

```
                        // .withIdentityPoolId("identity-pool-id")
                        // .withRoleArn("role-arn")
                    .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                        .withInstanceCount(3));

            System.out.println("Sending domain update request...");
            final UpdateElasticsearchDomainConfigResult updateResponse = client
                    .updateElasticsearchDomainConfig(updateRequest);
            System.out.println("Domain update response from Amazon OpenSearch
Service:");
            System.out.println(updateResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }

    /**
     * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
     * several minutes.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to delete
     */
    private static void deleteDomain(final AWSElasticsearch client, final String
domainName) {
        try {
            final DeleteElasticsearchDomainRequest deleteRequest = new
DeleteElasticsearchDomainRequest()
                    .withDomainName(domainName);

            System.out.println("Sending domain deletion request...");
            final DeleteElasticsearchDomainResult deleteResponse =
client.deleteElasticsearchDomain(deleteRequest);
            System.out.println("Domain deletion response from Amazon OpenSearch
Service:");
            System.out.println(deleteResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }

    /**
```

```
    * Waits for the domain to finish processing changes. New domains typically take
 15-30 minutes
    * to initialize, but can take longer depending on the configuration. Most
 updates to existing domains
    * take a similar amount of time. This method checks every 15 seconds and
 finishes only when
    * the domain's processing status changes to false.
    *
    * @param client
    *            The client to use for the requests to Amazon OpenSearch Service
    * @param domainName
    *            The name of the domain that you want to check
    */
   private static void waitForDomainProcessing(final AWSElasticsearch client, final
 String domainName) {
       // Create a new request to check the domain status.
       final DescribeElasticsearchDomainRequest describeRequest = new
 DescribeElasticsearchDomainRequest()
               .withDomainName(domainName);

       // Every 15 seconds, check whether the domain is processing.
       DescribeElasticsearchDomainResult describeResponse =
 client.describeElasticsearchDomain(describeRequest);
       while (describeResponse.getDomainStatus().isProcessing()) {
           try {
               System.out.println("Domain still processing...");
               TimeUnit.SECONDS.sleep(15);
               describeResponse =
 client.describeElasticsearchDomain(describeRequest);
           } catch (InterruptedException e) {
               e.printStackTrace();
           }
       }

       // Once we exit that loop, the domain is available
       System.out.println("Amazon OpenSearch Service has finished processing
 changes for your domain.");
       System.out.println("Domain description response from Amazon OpenSearch
 Service:");
       System.out.println(describeResponse.toString());
   }
}
```

Python

此範例使用來自 的 OpenSearchService 低階 Python 用戶端 適用於 Python (Boto) 的 AWS SDK 來建立網域、更新其組態，以及將其刪除。

```
import boto3
import botocore
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-west-2'
)

client = boto3.client('opensearch', config=my_config)

domainName = 'my-test-domain'  # The name of the domain


def createDomain(client, domainName):
    """Creates an Amazon OpenSearch Service domain with the specified options."""
    response = client.create_domain(
        DomainName=domainName,
        EngineVersion='OpenSearch_1.0',
        ClusterConfig={
            'InstanceType': 't2.small.search',
            'InstanceCount': 5,
            'DedicatedMasterEnabled': True,
            'DedicatedMasterType': 't2.small.search',
            'DedicatedMasterCount': 3
        },
        # Many instance types require EBS storage.
        EBSOptions={
            'EBSEnabled': True,
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
```

```
        AccessPolicies="{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-west-2:123456789012:domain/my-test-domain/*
\"}]}",
        NodeToNodeEncryptionOptions={
            'Enabled': True
        }
    )
    print("Creating domain...")
    print(response)


def updateDomain(client, domainName):
    """Updates the domain to use three data nodes instead of five."""
    try:
        response = client.update_domain_config(
            DomainName=domainName,
            ClusterConfig={
                'InstanceCount': 3
            }
        )
        print('Sending domain update request...')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def deleteDomain(client, domainName):
    """Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes."""
    try:
        response = client.delete_domain(
            DomainName=domainName
        )
        print('Sending domain deletion request...')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
```

```
        else:
            raise error


def waitForDomainProcessing(client, domainName):
    """Waits for the domain to finish processing changes."""
    try:
        response = client.describe_domain(
            DomainName=domainName
        )
        # Every 15 seconds, check whether the domain is processing.
        while response["DomainStatus"]["Processing"] == True:
            print('Domain still processing...')
            time.sleep(15)
            response = client.describe_domain(
                DomainName=domainName)

        # Once we exit the loop, the domain is available.
        print('Amazon OpenSearch Service has finished processing changes for your
 domain.')
        print('Domain description:')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def main():
    """Create a new domain, update its configuration, and delete it."""
    createDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    updateDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    deleteDomain(client, domainName)
```

節點

此範例使用適用於 JavaScript in Node.js 的 SDK 版本 3 OpenSearch 用戶端來建立網域、更新其組態並進行刪除。

```
var {
    OpenSearchClient,
    CreateDomainCommand,
    DescribeDomainCommand,
    UpdateDomainConfigCommand,
    DeleteDomainCommand
} = require("@aws-sdk/client-opensearch");
var sleep = require('sleep');

var client = new OpenSearchClient();

var domainName = 'my-test-domain'

// Create a new domain, update its configuration, and delete it.
createDomain(client, domainName)
waitForDomainProcessing(client, domainName)
updateDomain(client, domainName)
waitForDomainProcessing(client, domainName)
deleteDomain(client, domainName)

async function createDomain(client, domainName) {
    // Creates an Amazon OpenSearch Service domain with the specified options.
    var command = new CreateDomainCommand({
        DomainName: domainName,
        EngineVersion: 'OpenSearch_1.0',
        ClusterConfig: {
        'InstanceType': 't2.small.search',
        'InstanceCount': 5,
        'DedicatedMasterEnabled': 'True',
        'DedicatedMasterType': 't2.small.search',
        'DedicatedMasterCount': 3
        },
        EBSOptions:{
            'EBSEnabled': 'True',
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies: "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/my-test-domain/*
\"}]}",
         NodeToNodeEncryptionOptions:{
            'Enabled': 'True'
```

```
        }
    });
    const response = await client.send(command);
    console.log("Creating domain...");
    console.log(response);
}

async function updateDomain(client, domainName) {
    // Updates the domain to use three data nodes instead of five.
    var command = new UpdateDomainConfigCommand({
        DomainName: domainName,
        ClusterConfig: {
        'InstanceCount': 3
        }
    });
    const response = await client.send(command);
    console.log('Sending domain update request...');
    console.log(response);
}

async function deleteDomain(client, domainName) {
    // Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes.
    var command = new DeleteDomainCommand({
        DomainName: domainName
    });
    const response = await client.send(command);
    console.log('Sending domain deletion request...');
    console.log(response);
}

async function waitForDomainProcessing(client, domainName) {
    // Waits for the domain to finish processing changes.
    try {
        var command = new DescribeDomainCommand({
            DomainName: domainName
        });
        var response = await client.send(command);

        while (response.DomainStatus.Processing == true) {
            console.log('Domain still processing...')
            await sleep(15000) // Wait for 15 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
```

```
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        // Once we exit the loop, the domain is available.
        console.log('Amazon OpenSearch Service has finished processing changes for your
 domain.');
        console.log('Domain description:');
        console.log(response);

    } catch (error) {
        if (error.name === 'ResourceNotFoundException') {
            console.log('Domain not found. Please check the domain name.');
            }
    };
}
```

在 Amazon OpenSearch Service 中建立資料索引

由於 Amazon OpenSearch Service 使用 REST API，因此有不少方法都適用於為文件建立索引。您可以使用標準用戶端，例如 Curl 或任何可以傳送 HTTP 請求的程式設計語言。為了進一步簡化互動程序，OpenSearch Service 擁有許多程式設計語言的用戶端。進階使用者可以直接跳到the section called “將資料串流載入至 OpenSearch Service”。

我們強烈建議您使用 Amazon OpenSearch Ingestion 擷取資料，這是在 OpenSearch Service 中建置的全受管資料收集器。如需詳細資訊，請參閱 Amazon OpenSearch Ingestion。

如需索引的簡介，請參閱 OpenSearch 文件。

索引的命名限制

OpenSearch Service 索引具有以下命名限制：

- 所有字母必須小寫。
- 索引名稱的最開頭不可以是 _ 或 -。
- 索引名稱不可以包含空格、逗號、:、"、*、+、/、\、|、?、#、> 或 <。

請勿在索引、類型和文件 ID 名稱中包含敏感資訊。OpenSearch Service 會在其統一資源識別符 (URI) 中使用這些名稱。伺服器和應用程式通常會記錄 HTTP 請求，如果 URI 包含敏感資訊，這會導致不必要的資料暴露：

```
2018-10-03T23:39:43 198.51.100.14 200 "GET https://opensearch-domain/dr-jane-doe/flu-patients-2018/202-555-0100/ HTTP/1.1"
```

即使您沒有檢視相關 JSON 文件的許可，您可以從此仿造日誌列推斷出 Doe 醫生的其中一名病人 (其電話號碼為 202-555-0100) 在 2018 年患有流感。

如果 OpenSearch Service 在索引名稱中偵測到真實或存在的 IP 地址 (例如，my-index-12.34.56.78.91)，則會遮罩 IP 地址。呼叫 _cat/indices 產生下列回應：

```
green open my-index-x.x.x.x.91    soY19tBERoKo71WcEScidw 5 1 0 0   2kb  1kb
```

為防止不必要的混淆，請避免在索引名稱中包含 IP 地址。

縮減回應大小

由 _index 和 _bulk API 而來的回應包含相當多的資訊。這類資訊或許有助於對請求進行故障診斷或實作重試邏輯，但難免會耗用大量頻寬。在此範例中，對 32 個位元組的文件編製索引將產生 339 個位元組的回應 (包括標頭)：

```
PUT opensearch-domain/more-movies/_doc/1
{"title": "Back to the Future"}
```

回應

```
{
  "_index": "more-movies",
  "_type": "_doc",
  "_id": "1",
  "_version": 4,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 2,
    "failed": 0
  },
  "_seq_no": 3,
  "_primary_term": 1
}
```

此回應大小可能看起來很小，但是如果您每天索引 1,000,000 份文件 (大約每秒 11.5 份文件)，則每個回應 339 位元組的運作量達到每月 10.17 GB 的下載流量。

如果您擔心資料傳輸費用，請使用 filter_path 參數縮減 OpenSearch Service 回應的大小，但應留意切勿篩除需要用於識別或重試失敗請求的欄位。這類欄位因用戶端而異。filter_path 參數適用於所有 OpenSearch Service REST API，但對您經常呼叫的 API 尤其實用，例如 _index 和 _bulk API：

```
PUT opensearch-domain/more-movies/_doc/1?filter_path=result,_shards.total
{"title": "Back to the Future"}
```

回應

```
{
```

```
  "result": "updated",
  "_shards": {
    "total": 2
  }
}
```

除了納入欄位，您還可以使用 - 字首排除欄位。filter_path 也支援萬用字元：

```
POST opensearch-domain/_bulk?filter_path=-took,-items.index._*
{ "index": { "_index": "more-movies", "_id": "1" } }
{"title": "Back to the Future"}
{ "index": { "_index": "more-movies", "_id": "2" } }
{"title": "Spirited Away"}
```

回應

```
{
  "errors": false,
  "items": [
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    },
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    }
  ]
}
```

索引轉碼器

索引轉碼器會決定如何壓縮索引上的存放欄位並存放在磁碟上。索引轉碼器是由靜態index.codec設定所控制，其會指定壓縮演算法。此設定會影響索引碎片大小和操作效能。

如需支援的轉碼器及其效能特性的清單，請參閱 OpenSearch 文件中的支援的轉碼器。

當您選擇索引轉碼器時，請考慮下列事項：

- 為了避免變更現有索引的編解碼器設定的挑戰，請在非生產環境中測試代表性工作負載，然後再使用新的編解碼器設定。如需詳細資訊，請參閱變更索引轉碼器。
- 您無法將 Zstandard 壓縮轉碼器 (`"index.codec": "zstd"` 或 `"index.codec": "zstd_no_dict"`) 用於 k-NN 或 Security Analytics 索引。

將資料串流載入至 Amazon OpenSearch Service

您可以使用 OpenSearch Ingestion 直接將串流資料載入 Amazon OpenSearch Service 網域，而不需要使用第三方解決方案。若要將資料傳送至 OpenSearch Ingestion，您可以設定資料生產者，而 服務會自動將資料交付至您指定的網域或集合。若要開始使用 OpenSearch Ingestion，請參閱 the section called “教學課程：將資料擷取至集合”。

您仍然可以使用其他來源載入串流資料，例如 Amazon Data Firehose 和 Amazon CloudWatch Logs，這些資料內建支援 OpenSearch Service。諸如 Amazon S3、Amazon Kinesis Data Streams 和 Amazon DynamoDB 等其他來源使用 AWS Lambda 函數作為事件處理程序。Lambda 函數透過處理新資料並將其串流到您的網域來對其進行回應。

> **Note**
>
> Lambda 支援數種熱門的程式設計語言，並且可用於大部分 AWS 區域。如需詳細資訊，請參閱《 AWS Lambda 開發人員指南》中的 Lambda 入門和《》中的AWS 服務端點AWS 一般參考。

從 OpenSearch Ingestion 載入串流資料

您可以使用 Amazon OpenSearch Ingestion 將資料載入 OpenSearch Service 網域。您可以將資料生產者設定為將資料傳送至 OpenSearch Ingestion，並自動將資料交付至您指定的集合。您也可以設定 OpenSearch Ingestion，在交付資料之前轉換資料。如需詳細資訊，請參閱Amazon OpenSearch Ingestion。

從 Amazon S3 載入串流資料

您可以使用 Lambda 將資料從 Amazon S3 傳送到您的 OpenSearch Service 網域。送達 S3 儲存貯體的新資料會觸發 Lambda 的事件通知，然後執行您的自訂程式碼以執行索引。

這個串流資料方法非常靈活。您可以 索引物件中繼資料 ，若是純文字的話，便可剖析和索引一些物件本體元素。本節包含一些簡單的 Python 範本程式碼，其使用規則表達式來剖析日誌檔案和索引比對。

先決條件

繼續之前，您必須準備好以下資源。

先決條件	Description
Amazon S3 儲存貯體	如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的建立您的第一個 S3 儲存貯體。儲存貯體必須位於與您的 OpenSearch Service 網域相同的區域中。
OpenSearch Service 網域	您的 Lambda 函數處理資料後的資料目的地。如需詳細資訊，請參閱the section called “建立 OpenSearch Service 網域”。

建立 Lambda 部署套件

部署套件是指包含您的程式碼及其相依存項目的 ZIP 或 JAR 檔案。本節包括 Python 範本程式碼。如需其他程式設計語言，請參閱 AWS Lambda 開發人員指南中的 Lambda 部署套件。

1. 建立目錄。在此範例中，我們使用 `s3-to-opensearch` 這個名稱。
2. 在目錄中建立名為 `sample.py` 的檔案：

```
import boto3
import re
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-s3-index'
datatype = '_doc'
url = host + '/' + index + '/' + datatype

headers = { "Content-Type": "application/json" }
```

```
s3 = boto3.client('s3')

# Regular expressions used to parse some simple log lines
ip_pattern = re.compile('(\d+\.\d+\.\d+\.\d+)')
time_pattern = re.compile('\[(\d+\/\w\w\w\/\d\d\d\d:\d\d:\d\d:\d\d\s-\d\d\d\d)\]')
message_pattern = re.compile('\"(.+)\"')

# Lambda execution starts here
def handler(event, context):
    for record in event['Records']:

        # Get the bucket name and key for the new file
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']

        # Get, read, and split the file into lines
        obj = s3.get_object(Bucket=bucket, Key=key)
        body = obj['Body'].read()
        lines = body.splitlines()

        # Match the regular expressions to each line and index the JSON
        for line in lines:
            line = line.decode("utf-8")
            ip = ip_pattern.search(line).group(1)
            timestamp = time_pattern.search(line).group(1)
            message = message_pattern.search(line).group(1)

            document = { "ip": ip, "timestamp": timestamp, "message": message }
            r = requests.post(url, auth=awsauth, json=document, headers=headers)
```

編輯 `region` 和 `host` 的變數。

3. 安裝 pip (如果您尚未安裝的話)，然後將相依項目安裝到新的 `package` 目錄：

```
cd s3-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

所有 Lambda 執行環境均已安裝 Boto3，因此您不需要將其包含在您的部署套件中。

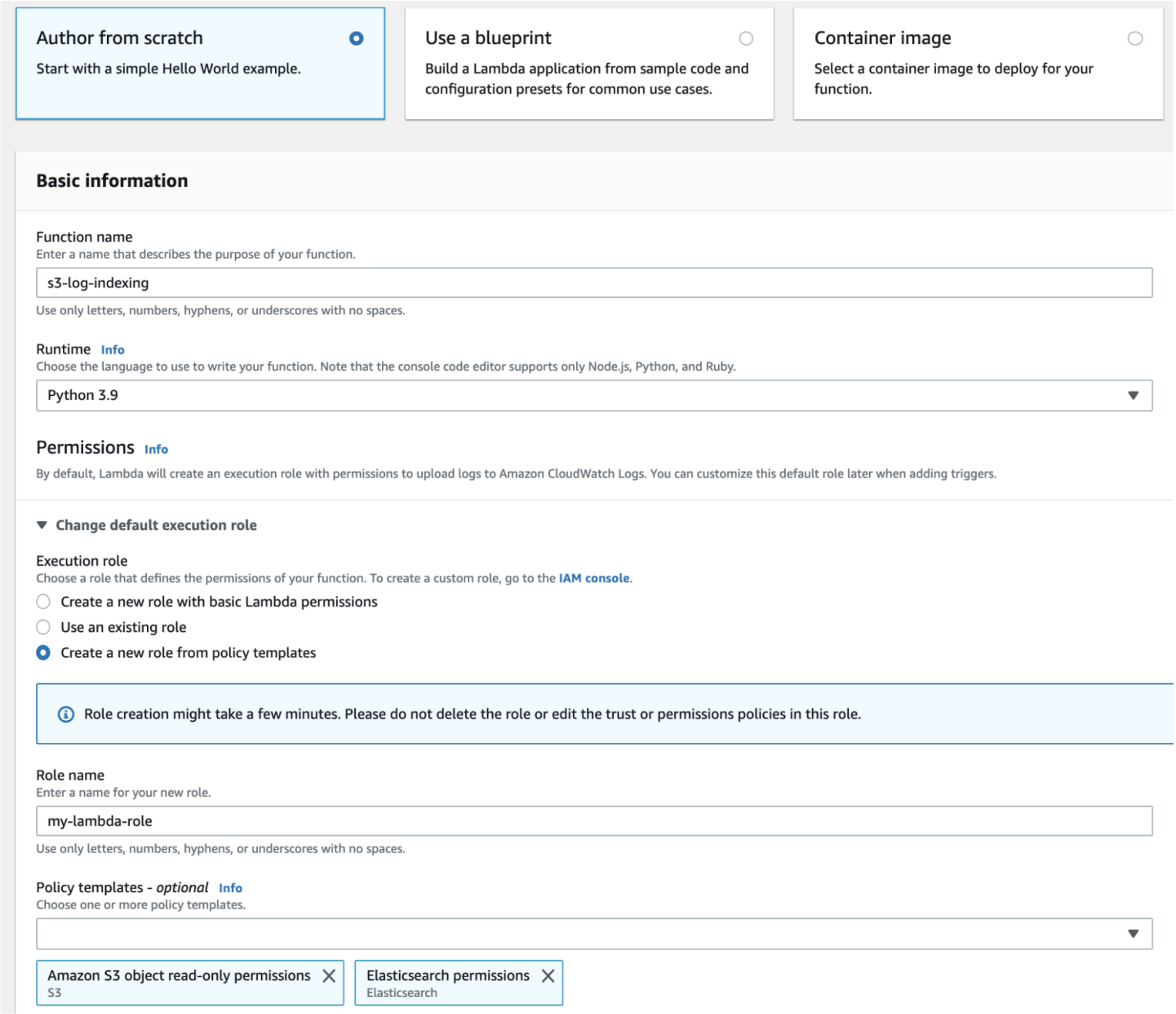
4. 封裝應用程式的程式碼和相依性：

```
cd package
zip -r ../lambda.zip .

cd ..
zip -g lambda.zip sample.py
```

建立 Lambda 函式

建立部署套件後，可建立 Lambda 函數。當您建立函數時，請選擇名稱、執行時間 (例如，Python 3.8) 和 IAM 角色。IAM 角色定義函數的許可。如需詳細說明，請參閱 AWS Lambda 開發人員指南中的使用主控台建立 Lambda 函數。

此範例假設您正在使用主控台。選擇 Python 3.9 和具有 S3 讀取許可和 OpenSearch Service 寫入許可的角色，如以下螢幕擷取畫面所示：



在建立函數，您必須新增觸發。在此範例中，我們希望只要日誌檔案送達 S3 儲存貯體，便執行程式碼：

1. 選擇 Add trigger (新增觸發條件)，然後選取 S3。
2. 選擇您的儲存貯體。
3. 針對 Event type (事件類型) 選擇 PUT。
4. 針對 Prefix (字首) 輸入 `logs/`。
5. 對於 Suffix (尾碼)，輸入 `.log`。
6. 確認遞迴叫用警告，然後選擇 Add (新增)。

最後，您可以上傳部署套件：

1. 選擇 Upload from (上傳自) 和 .zip file (.zip 檔案)，然後依照提示上傳您的部署套件。
2. 上傳完成後，編輯 Runtime settings (執行時間設定)，然後將 Handler (處理常式) 變更為 `sample.handler`。此設定通知 Lambda 應該在觸發後執行的檔案 (`sample.py`) 和方法 (`handler`)。

此時，您有一整組的資源：日誌檔案的儲存貯體、只要日誌檔案新增到儲存貯體便執行的函數、執行剖析和索引的程式碼，以及可搜尋和視覺化的 OpenSearch Service 網域。

測試 Lambda 函數

在建立函數之後，您可以進行測試，做法是上傳檔案到 Amazon S3 儲存貯體。使用下列範例日誌行，建立名為 `sample.log` 的檔案：

```
12.345.678.90 - [10/Oct/2000:13:55:36 -0700] "PUT /some-file.jpg"
12.345.678.91 - [10/Oct/2000:14:56:14 -0700] "GET /some-file.jpg"
```

上傳檔案到您的 S3 儲存貯體的 `logs` 資料夾。如需指示說明，請參閱 Amazon Simple Storage Service 使用者指南中的將物件上傳至您的儲存貯體。

然後使用 OpenSearch Service 主控台或 OpenSearch Dashboards 來確認 `lambda-s3-index` 索引包含兩個文件。您也可以提出標準搜尋請求：

```
GET https://domain-name/lambda-s3-index/_search?pretty
{
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vTYXaWIBJWV_TTkEuSDg",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.91",
          "message" : "GET /some-file.jpg",
          "timestamp" : "10/Oct/2000:14:56:14 -0700"
        }
      },
```

```
      {
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vjYmaWIBJWV_TTkEuCAB",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.90",
          "message" : "PUT /some-file.jpg",
          "timestamp" : "10/Oct/2000:13:55:36 -0700"
        }
      }
    ]
  }
}
```

從 Amazon Kinesis Data Streams 中載入串流資料

您可以將串流資料從 Kinesis Data Streams 中載入至 OpenSearch Service。送達資料串流的新資料會觸發 Lambda 的事件通知，然後執行您的自訂程式碼以執行索引。本節包括一些簡單的 Python 範本程式碼。

先決條件

繼續之前，您必須準備好以下資源。

先決條件	Description
Amazon Kinesis Data Stream	Lambda 函數的事件來源。如需進一步了解，請參閱 Kinesis Data Streams。
OpenSearch Service 網域	您的 Lambda 函數處理資料後的資料目的地。如需詳細資訊，請參閱 the section called “建立 OpenSearch Service 網域”。
IAM 角色	此角色必須擁有基本 OpenSearch Service、Kinesis 和 Lambda 許可，例如下列項目： JSON **{** **"Version":"2012-10-17",**

先決條件	Description

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpPost",
        "es:ESHttpPut",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:DescribeStream",
        "kinesis:ListStreams"
      ],
      "Resource": "*"
    }
  ]
}
```

角色必須具有下列信任關係：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需進一步了解，請參閱 IAM 使用者指南中的建立 IAM 角色。

建立 Lambda 函式

遵循the section called “建立 Lambda 部署套件”中的指示，但要建立名為 kinesis-to-opensearch 的目錄，並使用以下適用於 sample.py 的程式碼：

```
import base64
import boto3
import json
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-kine-index'
datatype = '_doc'
url = host + '/' + index + '/' + datatype + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        id = record['eventID']
        timestamp = record['kinesis']['approximateArrivalTimestamp']

        # Kinesis data is base64-encoded, so decode here
        message = base64.b64decode(record['kinesis']['data'])

        # Create the JSON document
        document = { "id": id, "timestamp": timestamp, "message": message }
        # Index the document
        r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return 'Processed ' + str(count) + ' items.'
```

編輯 region 和 host 的變數。

安裝 pip (如果您尚未安裝的話)，然後使用下列命令安裝相依項目：

```
cd kinesis-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

然後，遵循the section called “建立 Lambda 函式”中的指示，但要從the section called “先決條件”指定 IAM 角色和下列用於觸發的設定：

- Kinesis 串流：您的 Kinesis 串流
- 批次大小：100
- 開始位置：水平修剪

如需進一步了解，請參閱 Amazon Kinesis Data Streams 開發人員指南 中的什麼是 Amazon Kinesis Data Streams？。

此時，您有一整組的資源：Kinesis 資料串流、在串流收到新資料並索引該資料後執行的函數，以及可搜尋和視覺化的 OpenSearch Service 網域。

測試 Lambda 函數

在建立函數後，您可以做測試，方法是將新記錄加入到使用 AWS CLI的資料串流：

```
aws kinesis put-record --stream-name test --data "My test data." --partition-key
 partitionKey1 --region us-west-1
```

然後使用 OpenSearch Service 主控台或 OpenSearch Dashboards 來確認 `lambda-kine-index` 包含一個文件。您也可以使用以下請求：

```
GET https://domain-name/lambda-kine-index/_search
{
  "hits" : [
    {
      "_index": "lambda-kine-index",
      "_type": "_doc",
      "_id":
 "shardId-000000000000:495835116157626994950129608214214566865294366804960870 42",
      "_score": 1,
      "_source": {
```

```
        "timestamp": 1523648740.051,
        "message": "My test data.",
        "id":
 "shardId-000000000000:495835116157626994950129608214214566865294366804960870 42"
      }
    }
  ]
}
```

從 Amazon DynamoDB 中載入串流資料

您可以使用 從 Amazon DynamoDB AWS Lambda 將資料傳送至 OpenSearch Service 網域。送達資料庫資料表的新資料會觸發 Lambda 的事件通知，然後執行您的自訂程式碼以執行索引。

先決條件

繼續之前，您必須準備好以下資源。

先決條件	Description
DynamoDB 表	表格中包含您的來源資料。如需詳細資訊，請參閱 Amazon DynamoDB 開發人員指南中的 DynamoDB 資料表上的基本操作。 資料表必須位於和 OpenSearch Service 網域相同的區域，並將串流設定為 New image (新映像)。如需進一步了解，請參閱啟用串流。
OpenSearch Service 域	您的 Lambda 函數處理資料後的資料目的地。如需詳細資訊，請參閱the section called “建立 OpenSearch Service 網域”。
IAM 角色	此角色必須擁有基本 OpenSearch Service、DynamoDB 和 Lambda 執行許可，例如下列項目： JSON **{** **"Version":"2012-10-17",** **"Statement": [** **{** **"Effect": "Allow",** **"Action": [** **"es:ESHttpPost",**

先決條件	Description

```
        "es:ESHttpPut",
        "dynamodb:DescribeStream",
        "dynamodb:GetRecords",
        "dynamodb:GetShardIterator",
        "dynamodb:ListStreams",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

角色必須具有下列信任關係：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需進一步了解，請參閱 IAM 使用者指南中的建立 IAM 角色。

建立 Lambda 函式

遵循the section called “建立 Lambda 部署套件”中的指示，但要建立名為 `ddb-to-opensearch` 的目錄，並使用以下適用於 `sample.py` 的程式碼：

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-east-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-index'
datatype = '_doc'
url = host + '/' + index + '/' + datatype + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        # Get the primary key for use as the OpenSearch ID
        id = record['dynamodb']['Keys']['id']['S']

        if record['eventName'] == 'REMOVE':
            r = requests.delete(url + id, auth=awsauth)
        else:
            document = record['dynamodb']['NewImage']
            r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return str(count) + ' records processed.'
```

編輯 `region` 和 `host` 的變數。

安裝 pip (如果您尚未安裝的話)，然後使用下列命令安裝相依項目：

```
cd ddb-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

然後，遵循the section called “建立 Lambda 函式”中的指示，但要從the section called “先決條件”指定 IAM 角色和下列用於觸發的設定：

- 資料表：您的 DynamoDB 資料表
- 批次大小：100
- 開始位置：水平修剪

如需進一步了解，請參閱 Amazon DynamoDB 開發人員指南中的使用 DynamoDB Streams 和 Lambda 來處理新項目。

此時，您有一整組的資源：來源資料的 DynamoDB 資料表、資料表的 DynamoDB 串流變更、在您的來源資料變更並索引這些變更後所執行的函數，以及可搜尋和視覺化的 OpenSearch Service 網域。

測試 Lambda 函數

在建立函數後，您可以進行測試，方法是將新項目新增至使用 AWS CLI的 DynamoDB 資料表：

```
aws dynamodb put-item --table-name test --item '{"director": {"S": "Kevin
 Costner"},"id": {"S": "00001"},"title": {"S": "The Postman"}}' --region us-west-1
```

然後使用 OpenSearch Service 主控台或 OpenSearch Dashboards 來確認 `lambda-index` 包含一個文件。您也可以使用以下請求：

```
GET https://domain-name/lambda-index/_doc/00001
{
    "_index": "lambda-index",
    "_type": "_doc",
    "_id": "00001",
    "_version": 1,
    "found": true,
    "_source": {
        "director": {
            "S": "Kevin Costner"
        },
        "id": {
            "S": "00001"
        },
        "title": {
            "S": "The Postman"
        }
```

```
    }
}
```

從 Amazon Data Firehose 載入串流資料

Firehose 支援 OpenSearch Service 作為交付目的地。如需如何將串流資料載入 OpenSearch Service 的指示，請參閱《Amazon Data Firehose 開發人員指南》中的建立 Kinesis Data Firehose 交付串流和為您的目的地選擇 OpenSearch Service。

將資料載入到 OpenSearch Service 之前，您可能需要執行資料轉換。如需進一步了解如何使用 Lambda 函數來執行此任務，請參閱相同指南中的 Amazon Kinesis Data Firehose 資料轉換。

當您設定交付串流時，Firehose 具有「一鍵式」IAM 角色，提供它將資料傳送至 OpenSearch Service、備份 Amazon S3 上的資料，以及使用 Lambda 轉換資料所需的資源存取權。由於手動建立角色涉及複雜度，我們建議您使用所提供角色。

從 Amazon CloudWatch 中載入串流資料

透過使用 CloudWatch Logs 訂閱，您可以將串流資料從 CloudWatch Logs 載入 OpenSearch Service 網域。如需有關 Amazon CloudWatch 訂閱的資訊，請參閱使用訂閱即時處理日誌資料。如需組態資訊，請參閱 Amazon CloudWatch 開發人員指南中的將 CloudWatch Logs 資料串流到 Amazon OpenSearch Service。

從 載入串流資料 AWS IoT

您可以使用 AWS IoT 規則從 傳送資料。如需進一步了解，請參閱 AWS IoT 開發人員指南中的 OpenSearch 動作。

使用 Logstash 將資料載入至 Amazon OpenSearch Service

Logstash 的開放原始碼版本 (Logstash OSS) 提供便利的方法來使用大量 API 將資料上傳到您的 Amazon OpenSearch Service 網域。此服務支援所有標準 Logstash 輸入外掛程式，包括 Amazon S3 輸入外掛程式。OpenSearch Service 可支援 logstash-output-opensearch 輸出外掛程式，該外掛程式支援基本身分驗證和 IAM 憑證。該外掛程式可與 Logstash OSS 的 8.1 版及較低版本搭配使用。

Configuration

Logstash 組態會依據網域使用的身分驗證類型而有所不同。

無論您使用哪種身分驗證方法，都必須在組態檔案的輸出部分中將 `ecs_compatibility` 設定為 `disabled`。Logstash 8.0 引入了一個突破性的變更，其中所有外掛程式都以預設 ECS 相容模式執行。您必須覆寫預設值才能保持舊式行為。

精細存取控制組態

如果您的 OpenSearch Service 網域搭配使用精細存取控制與 HTTP 基本身分驗證，則組態類似於任何其他 OpenSearch 叢集。此組態檔案範例會從 Filebeat 的開源版本 (Filebeat OSS) 取得輸入：

```
input {
  beats  {
    port => 5044
  }
}

output {
  opensearch {
    hosts       => "https://domain-endpoint:443"
    user        => "my-username"
    password    => "my-password"
    index       => "logstash-logs-%{+YYYY.MM.dd}"
    ecs_compatibility => disabled
    ssl_certificate_verification => false
  }
}
```

組態因 Beats 應用程式和使用案例而異，但您的 Filebeat OSS 組態可能如下所示：

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /path/to/logs/dir/*.log
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
setup.ilm.enabled: false
setup.ilm.check_exists: false
setup.template.settings:
  index.number_of_shards: 1
output.logstash:
  hosts: ["logstash-host:5044"]
```

IAM 組態

如果您的網域搭配主要使用者使用 IAM 型網域存取政策或精細存取控制，您必須使用 IAM 憑證簽署對 OpenSearch Service 的所有請求。下列身分型政策會將所有 HTTP 請求授予網域的子資源。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
    }
  ]
}
```

若要設定 Logstash 組態，請將您的組態檔案變更為使用 外掛程式進行輸出。此組態檔案範例會從 S3 儲存貯體中的檔案取得輸入：

```
input {
  s3 {
    bucket => "amzn-s3-demo-"
    region => "us-east-1"
  }
}

output {
  opensearch {
    hosts => ["domain-endpoint:443"]
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      }
      index  => "logstash-logs-%{+YYYY.MM.dd}"
      ecs_compatibility => disabled
```

```
  }
}
```

如果您不想在組態檔案中提供 IAM 憑證，則您可以將其匯出 (或執行 `aws configure`)：

```
export AWS_ACCESS_KEY_ID="your-access-key"
export AWS_SECRET_ACCESS_KEY="your-secret-key"
export AWS_SESSION_TOKEN="your-session-token"
```

如果您的 OpenSearch Service 網域位於 VPC 中，則 Logstash OSS 機器必須能夠連線到 VPC 並可透過 VPC 安全群組存取該網域。如需詳細資訊，請參閱the section called “關於 VPC 網域上的存取政策”。

在 Amazon OpenSearch Service 中搜尋資料

有幾種在 Amazon OpenSearch Service 中搜尋文件的常用方法，包含 URI 搜尋與要求主體搜尋。OpenSearch Service 提供額外的功能，可改善搜尋體驗，例如自訂套件、SQL 支援和非同步搜尋。如需完整 OpenSearch 搜尋 API 參考，請參閱 OpenSearch 文件。

> **Note**
>
> 下列範例請求可與 OpenSearch API 搭配使用。部分請求可能無法與較舊的 Elasticsearch 版本搭配使用。

主題

- URI 搜尋
- 要求主體搜尋
- 對搜尋結果進行分頁
- 儀表板查詢語言
- 在 Amazon OpenSearch Service 中匯入和管理套件
- 使用 SQL 查詢您的 Amazon OpenSearch Service 資料
- Amazon OpenSearch Service 中的跨叢集搜尋
- 適用於 Amazon OpenSearch Service 的 Learning to Rank
- Amazon OpenSearch Service 中的非同步搜尋
- Amazon OpenSearch Service 中的時間點搜尋
- Amazon OpenSearch Service 中的代理程式搜尋
- Amazon OpenSearch Service 中的語意搜尋
- Amazon OpenSearch Service 中的並行區段搜尋
- 在 Amazon OpenSearch Service 中產生自然語言查詢

URI 搜尋

通用資源識別碼 (URI) 搜尋是最簡易的搜尋方式。在 URI 搜尋中，您將查詢指定為 HTTP 請求參數：

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/_search?q=house
```

範例回應看起來類似如下：


```
{
  "took": 25,
  "timed_out": false,
  "_shards": {
    "total": 10,
    "successful": 10,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 85,
      "relation": "eq",
    },
    "max_score": 6.6137657,
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "tt0077975",
        "_score": 6.6137657,
        "_source": {
          "directors": [
            "John Landis"
          ],
          "release_date": "1978-07-27T00:00:00Z",
          "rating": 7.5,
          "genres": [
            "Comedy",
            "Romance"
          ],
          "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTY2OTQxNTc1OF5BMl5BanBnXkFtZTYwNjA3NjI5._V1_SX400_.jpg",
          "plot": "At a 1962 College, Dean Vernon Wormer is determined to expel the
 entire Delta Tau Chi Fraternity, but those troublemakers have other plans for him.",
          "title": "Animal House",
          "rank": 527,
          "running_time_secs": 6540,
          "actors": [
            "John Belushi",
            "Karen Allen",
            "Tom Hulce"
```

```
          ],
          "year": 1978,
          "id": "tt0077975"
        }
      },
      ...
    ]
  }
}
```

根據預設，此查詢在所有索引中的所有欄位裡搜尋含有 house 的詞語。 若要縮小搜尋，請在 URI 中指定索引 (movies) 與文件欄位 (title)。

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?q=title:house
```

您可以在請求中加入其他參數，但是支援的參數只會提供 OpenSearch 搜尋選項的小型子集。下列請求傳回 20 個結果 (而非預設的 10 個)，並根據年份排序 (而非依據 _score)：

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?
q=title:house&size=20&sort=year:desc
```

要求主體搜尋

若要執行更複雜的搜尋，查詢時請使用 HTTP 要求主體以及 OpenSearch 網域專用語言 (DSL)。查詢 DSL 可讓您指定 OpenSearch 搜尋選項的完整範圍。

> **Note**
>
> 您不能在文字欄位值中包含 Unicode 特殊字元，否則會將值剖析為由特殊字元分隔的多個值。這種不正確的剖析可能會導致意外對文件進行篩選，並可能影響對其存取權的控制。如需詳細資訊，請參閱 OpenSearch 文件中文字欄位中 Unicode 特殊字元的注意事項。

下列 match 查詢類似於最終 URI 搜尋範例：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "sort": {
    "year": {
```

```
      "order": "desc"
    }
  },
  "query": {
    "query_string": {
      "default_field": "title",
      "query": "house"
    }
  }
}
```

> **Note**
>
> _search API 接受 HTTP GET 與POST 用於請求本文搜尋，但是所有 HTTP 用戶端支援新增請求本文到 GET 請求。POST 是更為通用的選擇。

在許多情況下，您可能想要搜尋多個欄位，但並非所有欄位。使用 multi_match 查詢：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title", "plot", "actors", "directors"]
    }
  }
}
```

增加欄位

您可以透過「增加」特定欄位來提升搜尋相關性。增加功能為運用倍數在一個欄位中加重配對數，使之較其他欄位中的配對數更高。在下面的例子中，匹配 title 欄位中的 John 對_score 的影響是在 plot 欄位中匹配的兩倍，以及在 actors 或 directors 欄位中匹配的四倍。結果就是 John Wick 和 John Carter 之類的影片會在搜尋結果的頂部，且 John Travolta 主演的影片會出現在底部。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
```

```
    "multi_match": {
      "query": "john",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

搜尋結果反白呈現

highlight 選項會通知 OpenSearch 在 hits 陣列內傳回額外的物件 (如果查詢符合一個或多個欄位)。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    }
  }
}
```

如果查詢符合 plot 欄位的內容，則命中看起來會如下：

```
{
  "_index": "movies",
  "_type": "movie",
  "_id": "tt0091541",
  "_score": 11.276199,
  "_source": {
    "directors": [
      "Richard Benjamin"
    ],
    "release_date": "1986-03-26T00:00:00Z",
    "rating": 6,
    "genres": [
```

```
      "Comedy",
      "Music"
    ],
    "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTIzODEzODE2OF5BMl5BanBnXkFtZTcwNjQ3ODcyMQ@@._V1_SX400_.jpg",
    "plot": "A young couple struggles to repair a hopelessly dilapidated house.",
    "title": "The Money Pit",
    "rank": 4095,
    "running_time_secs": 5460,
    "actors": [
      "Tom Hanks",
      "Shelley Long",
      "Alexander Godunov"
    ],
    "year": 1986,
    "id": "tt0091541"
  },
  "highlight": {
    "plot": [
      "A young couple struggles to repair a hopelessly dilapidated <em>house</em>."
    ]
  }
}
```

依預設，OpenSearch 會在 標籤中包裝相符的字串，提供高達 100 個相符項目細節的字元，並透過辨識標點符號、空格、標籤和換行來將內容分為多個句子。所有設定皆可自訂：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    },
    "pre_tags": "<strong>",
    "post_tags": "</strong>",
    "fragment_size": 200,
```

```
    "boundary_chars": ".,!? "
  }
}
```

計數 API

如果您對文件的內容不感興趣，且只想知道相符項目的數量，您可以使用 _count API (而不是 _search API)。下列請求使用 query_string 查詢來識別浪漫喜劇：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_count
{
  "query": {
    "query_string": {
      "default_field": "genres",
      "query": "romance AND comedy"
    }
  }
}
```

範例回應看起來類似如下：

```
{
  "count": 564,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  }
}
```

對搜尋結果進行分頁

如果您需要顯示大量搜尋結果，您可以使用數種不同的方法來實作分頁。

時間點

時間點 (PIT) 功能是一種搜尋類型，可讓您針對固定時間的資料集執行不同的查詢。這是 OpenSearch 中偏好的分頁方法，特別是針對深度分頁。您可以搭配 OpenSearch Service 2.5 版及更新版本使用 PIT。如需 PIT 的詳細資訊，請參閱 ???。

from 和 size 參數

分頁最簡單的方法是使用 from和 size 參數。以下請求傳回結果 20-39 項以零為索引的搜尋結果清單：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "from": 20,
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

如需搜尋分頁的詳細資訊，請參閱 OpenSearch 文件中的分頁結果。

儀表板查詢語言

您可以使用儀表板查詢語言 (DQL)，在 OpenSearch Dashboards 中搜尋資料和視覺化效果。DQL 使用四種主要查詢類型：術語、布林值、日期和範圍、以及巢狀欄位。

術語查詢

術語查詢要求您指定要搜尋的術語。

若要執行術語查詢，請輸入下列內容：

```
host:www.example.com
```

布林值查詢

您可以使用布林運算子 AND、or 以及 not 以合併多個查詢。

要執行布林值查詢，請貼上以下內容：

```
host.keyword:www.example.com and response.keyword:200
```

日期和範圍查詢

您可以使用日期和範圍查詢來尋找查詢之前或之後的日期。

- > 表示搜尋指定日期之後的日期。
- < 表示搜尋指定日期之前的日期。

```
@timestamp > "2020-12-14T09:35:33"
```

巢狀欄位查詢

如果您擁有一個帶巢狀欄位的文件，則必須指定要擷取文件的哪些部分。以下是包含巢狀欄位的範例文件：

```
{"NBA players":[
    {"player-name": "Lebron James",
      "player-position": "Power forward",
      "points-per-game": "30.3"
    },
    {"player-name": "Kevin Durant",
      "player-position": "Power forward",
      "points-per-game": "27.1"
    },
    {"player-name": "Anthony Davis",
      "player-position": "Power forward",
      "points-per-game": "23.2"
    },
    {"player-name": "Giannis Antetokounmpo",
      "player-position": "Power forward",
      "points-per-game":"29.9"
    }
  ]
}
```

若要使用 DQL 擷取特定欄位，請貼上下列內容：

```
NBA players: {player-name: Lebron James}
```

若要從巢狀文件中擷取多個物件，請貼入下列內容：

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo}
```

若要在某個範圍內搜尋，請貼上以下內容：

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo and < 30}
```

如果您的文件在另一個物件中有巢狀物件，您仍然可以透過指定所有層級來擷取資料。如要執行此操作，請貼上以下內容：

```
Top-Power-forwards.NBA players: {player-name:Lebron James}
```

在 Amazon OpenSearch Service 中匯入和管理套件

Amazon OpenSearch Service 可讓您上傳自訂字典檔案，例如停止單字和同義詞，並將外掛程式與您的網域建立關聯。這些外掛程式可以是預先封裝、自訂或第三方，可讓您彈性擴展網域的功能。所有這些類型檔案的一般術語都是套件。

- 字典檔案會指示 OpenSearch 忽略常見的高頻率單字，或將類似字詞視為同等字詞，例如「凍結卡士達」、「Gelato」和「冰淇淋」，以協助精簡搜尋結果。它們也可以改善幹線，如日文 (kuromoji) 分析外掛程式所示。
- 預先封裝的外掛程式提供內建功能，例如用於個人化搜尋結果的 Amazon Personalize 外掛程式。這些外掛程式使用 `ZIP-PLUGIN`套件類型。如需詳細資訊，請參閱the section called “依引擎版本分類的外掛程式”。
- 自訂和第三方外掛程式可讓您新增量身打造的功能，或與外部系統整合，為您的網域提供更多彈性。如同預先封裝的外掛程式，您可以將自訂外掛程式上傳為`ZIP-PLUGIN`套件。對於第三方外掛程式，您還必須將外掛程式授權和組態檔案匯入為單獨的套件，然後將它們全部與網域建立關聯。

  如需詳細資訊，請參閱下列主題：

  - the section called “自訂外掛程式”
  - the section called “第三方外掛程式”

> **Note**
>
> 您最多可以將 20 個外掛程式與單一網域建立關聯。此限制包含所有外掛程式類型 - 選用、第三方和自訂外掛程式。

主題

- 所需的許可
- 將套件上傳至 Amazon S3
- 匯入和關聯套件
- 搭配 OpenSearch 使用套件
- 更新套件
- 使用新的字典手動更新索引
- 解除關聯並移除套件
- 在 Amazon OpenSearch Service 中管理自訂外掛程式
- 在 Amazon OpenSearch Service 中安裝第三方外掛程式

所需的許可

沒有管理員存取權的使用者需要特定 AWS Identity and Access Management (IAM) 動作，才能管理套件：

- `es:CreatePackage` – 建立套件
- `es:DeletePackage` – 刪除套件
- `es:AssociatePackage` – 將套件與網域建立關聯
- `es:DissociatePackage` – 取消套件與網域的關聯

您也需要自訂套件所在的 Amazon S3 儲存貯體路徑或物件的許可。

授予 IAM 內的所有許可，而不是在網域存取政策中。如需詳細資訊，請參閱the section called “身分和存取權管理”。

將套件上傳至 Amazon S3

本節說明如何上傳自訂字典套件，因為已安裝預先封裝的外掛程式套件。您必須先將自訂字典上傳至 Amazon S3 儲存貯體，才能將自訂字典與您的網域建立關聯。如需指示說明，請參閱 Amazon Simple Storage Service 使用者指南中的上傳物件。不需要上傳支援的外掛程式。

如果您的字典包含敏感資訊，請在上傳時以 S3-managed金鑰指定伺服器端加密。OpenSearch Service 無法存取您使用 AWS KMS 金鑰保護的 S3 中的檔案。

上傳檔案後，請記下其 S3 路徑。路徑格式為 `s3://amzn-s3-demo-bucket/file-path/file-name`。

您可以使用下列同義字檔案進行測試。另存為 synonyms.txt。

```
danish, croissant, pastry
ice cream, gelato, frozen custard
sneaker, tennis shoe, running shoe
basketball shoe, hightop
```

某些字典 (例如 Hunspell 字典) 會使用多個檔案並要求在檔案系統上使用自己的字典。目前，OpenSearch Service 僅支援單一檔案字典。

匯入和關聯套件

主控台是將自訂字典匯入 OpenSearch Service 的最簡單方法。當您從 Amazon S3 匯入字典時，OpenSearch Service 會儲存自己的套件複本，並使用 AES-256 搭配 OpenSearch Service 受管金鑰自動加密該複本。

選用外掛程式已預先安裝在 OpenSearch Service 中，因此您不需要自行上傳，但您需要將外掛程式與網域建立關聯。可用的外掛程式會列在 主控台的套件畫面上。

匯入套件並將其與網域建立關聯

1. 在 Amazon OpenSearch Service 主控台中，選擇 Packages (套件)。
2. 選擇 Import package (匯入套件)。
3. 為套件提供描述性名稱。
4. 提供檔案的 S3 路徑，然後選擇 Import (匯入)。
5. 返回 Packages (套件) 畫面。
6. 當封裝狀態為 Available (可用) 時，請加以選取。
7. 選擇與網域建立關聯。
8. 選取網域，然後選擇下一步。檢閱套件，然後選擇關聯。
9. 在導覽窗格中，選擇您的網域，然後移至 Packages (套件) 索引標籤。
10. 如果套件是自訂字典，請在套件變成可用時記下 ID。使用 `analyzers/id` 作為對 OpenSearch 的請求中的檔案路徑。

搭配 OpenSearch 使用套件

本節說明如何使用兩種類型的套件：自訂字典和預先封裝的外掛程式。

使用自訂字典

將檔案與網域建立關聯後，您可以在建立權杖化程式和權杖篩選條件user_dictionary時synonyms_path，將其用於參數，例如 stopwords_path、 和 。確切的參數因物件而異。多個物件可支援 synonyms_path 和 stopwords_path，但 user_dictionary 專屬於 kuromoji 外掛程式。

對於 IK (中文) 分析外掛程式，您可以將自訂字典檔案做為自定套件上傳，並將其與一個網域相關聯，且外掛程式會自動將其掛載，不需要 user_dictionary 參數。如果您的檔案是同義詞檔案，請使用 synonyms_path 參數。

以下範例會將同義字檔案新增至新索引：

```
PUT my-index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "my_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["my_filter"]
          }
        },
        "filter": {
          "my_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F111111111",
            "updateable": true
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "standard",
        "search_analyzer": "my_analyzer"
      }
    }
```

```
  }
}
```

此請求建立使用標準分詞器和同義詞字符篩選條件索引的自訂分析器。

- 分詞器會根據一組規則將字符分解為字符 (通常是字詞)。最簡單的例子是空格分詞器，每次遇到一個空白字符時，便會將前面的字元分解為字符。一個更複雜的例子是標準分詞器，它會使用一組基於語法的規則跨多種語言作業。
- 字符篩選條件會新增、修改或刪除字符。例如，同義字字符篩選條件會在同義字清單中找到單字時加入字符。停止字符篩選條件會在停止字詞清單中找到字詞時移除標記。

此請求還會將一個文字欄位 (`description`) 新增至映射，並讓 OpenSearch 使用新分析器作為其搜尋分析器。您可以看到它仍然使用標準分析器作為其索引分析器。

最後，請注意字符篩選器中的 `"updateable": true` 行。此欄位僅適用於搜尋分析器，不適用於索引分析器，如果您稍後想要自動更新搜尋分析器，此欄位就非常重要。

為了測試，我們加入了一些文件索引：

```
POST _bulk
{ "index": { "_index": "my-index", "_id": "1" } }
{ "description": "ice cream" }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "description": "croissant" }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "description": "tennis shoe" }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "description": "hightop" }
```

然後使用同義詞來搜尋它們：

```
GET my-index/_search
{
  "query": {
    "match": {
      "description": "gelato"
    }
  }
}
```

在這種情況下，OpenSearch 會傳回以下回應：

```
{
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 0.99463606,
    "hits": [{
      "_index": "my-index",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.99463606,
      "_source": {
        "description": "ice cream"
      }
    }]
  }
}
```

> **Tip**
>
> 字典檔案使用與其大小呈正比的 Java 堆積空間。例如，2 GiB 的字典檔案可能會在節點上使用 2 GiB 的堆積空間。如果您使用大型檔案，請確保節點有足夠的堆積空間可容納這些檔案。監控 `JVMMemoryPressure` 指標，並視需要擴展您的叢集。

使用預先封裝的外掛程式

OpenSearch Service 可讓您將預先安裝的選用 OpenSearch 外掛程式與網域建立關聯。預先封裝的外掛程式套件與特定 OpenSearch 版本相容，而且只能與該版本的網域建立關聯。您網域的可用套件清單包含與您的網域版本相容的所有支援外掛程式。將外掛程式與網域建立關聯後，網域上的安裝程序就會開始。然後，您可以在向 OpenSearch Service 提出請求時參考和使用外掛程式。

關聯和取消關聯外掛程式需要藍/綠部署。如需詳細資訊，請參閱the section called “通常會導致藍/綠部署的變更”。

選用的外掛程式包括語言分析器和自訂的搜尋結果。例如，Amazon Personalize Search Ranking 外掛程式使用機器學習來為您的客戶個人化搜尋結果。如需此外掛程式的詳細資訊，請參閱從 OpenSearch

個人化搜尋結果。如需所有支援的外掛程式清單，請參閱 the section called “依引擎版本分類的外掛程式”。

Sudachi 外掛程式

對於 Sudachi 外掛程式，當您重新關聯字典檔案時，它不會立即反映在網域上。當下一個藍/綠部署在網域上執行時，字典會重新整理，作為組態變更或其他更新的一部分。或者，您可以使用更新的資料建立新的套件、使用此新套件建立新的索引、將現有索引重新索引為新索引，然後刪除舊索引。如果您偏好使用重新索引方法，請使用索引別名，以免流量中斷。

此外，Sudachi 外掛程式僅支援二進位 Sudachi 字典，您可以使用 CreatePackage API 操作來上傳。如需有關預先建置的系統字典和編譯使用者字典程序的資訊，請參閱 Sudachi 文件。

> **Note**
>
> 將二進位字典檔案上傳至 Amazon S3 時，您必須將 S3 物件的內容類型設定為 `binary/octet-stream`。使用 `application/octet-stream` 會導致套件匯入失敗。

下列範例示範如何使用系統和使用者字典搭配 Sudachi 權杖化工具。您必須將這些字典上傳為具有 類型的自訂套件，TXT-DICTIONARY並在其他設定中提供其套件 IDs。

```
PUT sudachi_sample
{
  "settings": {
    "index": {
      "analysis": {
        "tokenizer": {
          "sudachi_tokenizer": {
            "type": "sudachi_tokenizer",
            "additional_settings": "{\"systemDict\": \"<system-dictionary-package-id>\",\"userDict\": [\"<user-dictionary-package-id>\"]}"
        }
        },
        "analyzer": {
          "sudachi_analyzer": {
            "filter": ["my_searchfilter"],
            "tokenizer": "sudachi_tokenizer",
            "type": "custom"
          }
        },
        "filter":{
```

```
          "my_searchfilter": {
            "type": "sudachi_split",
            "mode": "search"
          }
        }
      }
    }
  }
}
```

更新套件

本節僅涵蓋如何更新自訂字典套件，因為預先封裝的外掛程式套件已為您更新。將新版本的字典上傳至 Amazon S3 並不會自動更新 Amazon OpenSearch Service 上的套件。OpenSearch Service 會儲存自己的檔案副本，因此如果您將新版本上傳到 S3，則必須手動更新。

每個關聯網域也都會儲存其自己的檔案副本。為了保持搜尋行為可預測，網域會繼續使用其目前的套件版本，直到您明確更新它們為止。若要更新自訂套件，請在 中修改 檔案 Amazon S3 Control、在 OpenSearch Service 中更新套件，然後套用更新。

主控台

1. 在 OpenSearch Service 主控台中，選擇 Packages (套件)。
2. 選擇套件和 Update (更新)。
3. 提供檔案的新 S3 路徑，然後選擇更新套件。
4. 返回 Packages (套件) 畫面。
5. 當套件狀態變為 Available (可用) 時，請加以選取。然後選擇一個或多個關聯網域，選擇 Apply update (套用更新) 並確認。等待關聯狀態變為 Active (作用中)。
6. 後續步驟取決於您設定索引的方式：

   - 如果您的網域執行的是 OpenSearch 或 Elasticsearch 7.8 或更新版本，且僅使用可更新欄位設定為 true 的搜尋分析器，則您不需要採取任何進一步動作。OpenSearch Service 會使用 _plugins/_refresh_search_analyzers API 自動更新您的索引。
   - 如果您的網域正在執行 Elasticsearch 7.7 或更早版本、使用索引分析器，或不使用 `updateable` 欄位，請參閱 the section called “使用新的字典手動更新索引”。

雖然主控台是最簡單的方法，但您也可以使用 AWS CLI、 SDKs或組態 API 來更新 OpenSearch Service 套件。如需詳細資訊，請參閱 AWS CLI 命令參考和 Amazon OpenSearch Service API 參考。

AWS 開發套件

您可以使用開發套件來自動化更新程序，而不是在主控台中手動更新套件。以下範例 Python 指令碼會將新的套件檔案上傳到 Amazon S3，在 OpenSearch Service 中更新套件，並將新套件套用到指定網域。確認更新成功後，它會對 OpenSearch 進行範例呼叫，以證明已套用新的同義詞。

您必須提供
host、region、file_name、bucket_name、s3_key、package_id、domain_name 和 query 的值。

```
from requests_aws4auth import AWS4Auth
import boto3
import requests
import time
import json
import sys

host = ''  # The OpenSearch domain endpoint with https:// and a trailing slash. For
 example, https://my-test-domain.us-east-1.es.amazonaws.com/
region = ''  # For example, us-east-1
file_name = ''  # The path to the file to upload
bucket_name = ''  # The name of the S3 bucket to upload to
s3_key = ''  # The name of the S3 key (file name) to upload to
package_id = ''  # The unique identifier of the OpenSearch package to update
domain_name = ''  # The domain to associate the package with
query = ''  # A test query to confirm the package has been successfully updated

service = 'es'
credentials = boto3.Session().get_credentials()
client = boto3.client('opensearch')
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def upload_to_s3(file_name, bucket_name, s3_key):
    """Uploads file to S3"""
    s3 = boto3.client('s3')
    try:
        s3.upload_file(file_name, bucket_name, s3_key)
        print('Upload successful')
        return True
    except FileNotFoundError:
        sys.exit('File not found. Make sure you specified the correct file path.')
```

```
def update_package(package_id, bucket_name, s3_key):
    """Updates the package in OpenSearch Service"""
    print(package_id, bucket_name, s3_key)
    response = client.update_package(
        PackageID=package_id,
        PackageSource={
            'S3BucketName': bucket_name,
            'S3Key': s3_key
        }
    )
    print(response)


def associate_package(package_id, domain_name):
    """Associates the package to the domain"""
    response = client.associate_package(
        PackageID=package_id, DomainName=domain_name)
    print(response)
    print('Associating...')


def wait_for_update(domain_name, package_id):
    """Waits for the package to be updated"""
    response = client.list_packages_for_domain(DomainName=domain_name)
    package_details = response['DomainPackageDetailsList']
    for package in package_details:
        if package['PackageID'] == package_id:
            status = package['DomainPackageStatus']
            if status == 'ACTIVE':
                print('Association successful.')
                return
            elif status == 'ASSOCIATION_FAILED':
                sys.exit('Association failed. Please try again.')
            else:
                time.sleep(10)  # Wait 10 seconds before rechecking the status
                wait_for_update(domain_name, package_id)


def sample_search(query):
    """Makes a sample search call to OpenSearch"""
    path = '_search'
    params = {'q': query}
```

```
url = host + path
response = requests.get(url, params=params, auth=awsauth)
print('Searching for ' + '"' + query + '"')
print(response.text)
```

> **Note**
>
> 如果您在使用 執行指令碼時收到「找不到套件」錯誤 AWS CLI，可能表示 Boto3 正在使用 ~/.aws/config 中指定的區域，這不是 S3 儲存貯體所在的區域。執行 `aws configure` 並指定正確的區域，或者明確地將區域新增至用戶端：
>
> ```
> client = boto3.client('opensearch', region_name='us-east-1')
> ```

使用新的字典手動更新索引

手動索引更新僅適用於自訂字典，不適用於預先封裝的外掛程式。若要使用更新的字典，如果您符合下列任何條件，則必須手動更新索引：

- 您的網域執行 Elasticsearch 7.7 或更早版本。
- 您使用自訂套件作為索引分析器。
- 您使用自訂套件作為搜索分析器，但不包含 updateable (可更新) 欄位。

若要使用新套件檔案更新分析器，您有兩個選項：

- 關閉並開啟您要更新的任何索引：

```
POST my-index/_close
POST my-index/_open
```

- 對索引進行重新索引。首先，建立使用更新同義詞檔案（或全新的檔案）的索引。請注意，僅支援 UTF-8。

```
PUT my-new-index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
```

```
          "synonym_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["synonym_filter"]
          }
        },
        "filter": {
          "synonym_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F222222222"
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "synonym_analyzer"
      }
    }
  }
}
```

然後將舊索引重新索引到該新索引：

```
POST _reindex
{
  "source": {
    "index": "my-index"
  },
  "dest": {
    "index": "my-new-index"
  }
}
```

如果您經常更新索引分析器，請使用索引別名來維持最新索引的一致路徑：

```
POST _aliases
{
  "actions": [
```

```
    {
      "remove": {
        "index": "my-index",
        "alias": "latest-index"
      }
    },
    {
      "add": {
        "index": "my-new-index",
        "alias": "latest-index"
      }
    }
  ]
}
```

如果您不需要舊索引，請將其刪除：

```
DELETE my-index
```

解除關聯並移除套件

從網域取消套件的關聯，無論是自訂字典或預先封裝的外掛程式，都表示您在建立新索引時無法再使用該套件。取消套件的關聯後，使用套件的現有索引將無法再使用它。您必須從任何索引移除套件，才能將其取消關聯，否則取消關聯會失敗。

主控台是將套件與網域解除關聯並將其從 OpenSearch Service 中移除的最簡單方法。從 OpenSearch Service 中移除套件並不會將其從 Amazon S3 上的原始位置中移除。

取消套件與網域的關聯

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在導覽窗格中，選擇網域。
3. 選擇網域，然後導覽至套件索引標籤。
4. 選擇套件，Actions (動作)，然後選擇 Dissociate (解除關聯)。確認您的選擇。
5. 等待套件從清單中消失。您可能需要重新整理瀏覽器。
6. 如果您想要將套件與其他網域搭配使用，請在這裡停止。若要繼續移除套件 （ 如果它是自訂字典 ），請在導覽窗格中選擇套件。

7. 選取套件並選擇 Delete (刪除)。

或者，使用 、 AWS CLI SDKs或組態 API 來取消關聯和移除套件。如需詳細資訊，請參閱 AWS CLI 命令參考和 Amazon OpenSearch Service API 參考。

在 Amazon OpenSearch Service 中管理自訂外掛程式

使用 OpenSearch Service 的自訂外掛程式，您可以在語言分析、自訂篩選、排名等領域擴展 OpenSearch 功能，讓您可以打造個人化的搜尋體驗。OpenSearch 的自訂外掛程式可以透過擴展 `org.opensearch.plugins.Plugin`類別，然後將其封裝在 `.zip` 檔案中來開發。

Amazon OpenSearch Service 目前支援下列外掛程式延伸：

- AnalysisPlugin – 透過新增自訂分析器、角色字符化器或用於文字處理的篩選條件來擴展分析功能。
- SearchPlugin – 使用自訂查詢類型、相似性演算法、建議選項和彙總增強搜尋功能。
- MapperPlugin – 可讓您在 OpenSearch 中建立自訂欄位類型及其映射組態，讓您定義應如何存放和編製索引不同類型的資料。
- ScriptPlugin – 可讓您將自訂指令碼功能新增至 OpenSearch，例如，在搜尋或編製索引期間進行評分、排序和欄位值轉換等操作的自訂指令碼。

您可以使用 OpenSearch Service 主控台或現有 API 命令來上傳自訂套件，並將外掛程式與 Amazon OpenSearch Service 叢集建立關聯。您也可以使用 DescribePackages 命令來描述您帳戶中的所有套件，以及檢視 OpenSearch 版本和錯誤詳細資訊等詳細資訊。OpenSearch Service 會驗證外掛程式套件的版本相容性、安全性漏洞和允許的外掛程式操作。如需自訂套件的詳細資訊，請參閱 the section called “套件”。

OpenSearch 版本和 AWS 區域 支援

執行 OpenSearch 2.15 版的 OpenSearch Service 網域支援自訂外掛程式，如下所示 AWS 區域：

- 美國東部 (俄亥俄) (us-east-2)
- 美國東部 (維吉尼亞北部) (us-east-1)
- 美國西部 (奧勒岡) (us-west-2)
- 亞太區域 (孟買) (ap-south-1)
- 亞太區域 (首爾) (ap-northeast-2)
- 亞太地區 (新加坡) (ap-southeast-1)

- 亞太地區 (雪梨) (ap-southeast-2)
- 亞太地區 (東京) (ap-northeast-1)
- 加拿大 (中部) (ca-central-1)
- 歐洲 (法蘭克福) (eu-central-1)
- 歐洲 (愛爾蘭) (eu-west-1)
- 歐洲 (倫敦) (eu-west-2)
- 歐洲 (巴黎) (eu-west-3)
- 南美洲 (聖保羅) (sa-east-1)

> **Note**
>
> 自訂外掛程式包含使用者開發的程式碼。任何由使用者開發的程式碼引起的問題，包括 SLA 違規，都不符合 SLA 點數的資格。如需詳細資訊，請參閱 Amazon OpenSearch Service - 服務水準協議。

主題

- 外掛程式配額
- 先決條件
- 疑難排解
- 使用主控台安裝自訂外掛程式
- 使用 管理自訂外掛程式 AWS CLI
- Amazon OpenSearch Service 自訂套件 AWS KMS 整合

外掛程式配額

- 每個區域每個帳戶最多可以建立 25 個自訂外掛程式。
- 外掛程式的未壓縮大小上限為 1 GB。
- 可以與單一網域相關聯的外掛程式數目上限為 20。此配額適用於所有合併的外掛程式類型：選用、第三方和自訂。
- 執行 OpenSearch 2.15 版或更新版本的網域支援自訂外掛程式。
- 外掛程式`descriptor.properties`的檔案必須支援類似 2.15.0 或任何 2.x.x 版本的引擎版本，其中修補程式版本設定為零。

先決條件

安裝自訂外掛程式並將其與網域建立關聯之前，請確定您符合下列要求：

- `descriptor.properties` 檔案中外掛程式支援的引擎版本應類似於 `2.15.0`或 `2.x.0`。也就是說，修補程式版本必須為零。
- 您的網域必須啟用下列功能：
  - 節點對節點加密
  - 靜態加密
  - `EnforceHTTPS` 設定為 'true'

    另請參閱《 AWS Config 開發人員指南》中的 opensearch-https-required。
  - 用戶端必須支援 Policy-Min-TLS-1-2-PFS-2023-10。您可以使用下列命令指定此支援。將*####*取代為您自己的資訊：

    ```
    aws opensearch update-domain-config \
        --domain-name domain-name \
        --domain-endpoint-options '{"TLSSecurityPolicy":"Policy-Min-TLS-1-2-
    PFS-2023-10" }'
    ```

    如需詳細資訊，請參閱《Amazon OpenSearch Service API 參考》中的 DomainEndpointOptions。

疑難排解

如果系統傳回錯誤 `PluginValidationFailureReason : The provided plugin could not be loaded`，請參閱 the section called “由於版本相容性，自訂外掛程式安裝失敗” 以取得故障診斷資訊。

使用主控台安裝自訂外掛程式

若要將第三方外掛程式與網域建立關聯，請先將外掛程式授權和組態匯入為套件。

安裝自訂外掛程式

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇套件。

3. 選擇 Import package (匯入套件)。
4. 在名稱中，輸入唯一且易於識別的外掛程式名稱。
5. （選用）針對描述，提供套件或其用途的任何實用詳細資訊。
6. 針對套件類型，選擇外掛程式。
7. 針對套件來源，輸入路徑或瀏覽至 Amazon S3 中的外掛程式 ZIP 檔案。
8. 針對 OpenSearch 引擎版本，選擇外掛程式支援的 OpenSearch 版本。
9. 針對套件加密，選擇是否要自訂套件的加密金鑰。根據預設，OpenSearch Service 會使用 加密外掛程式套件 AWS 擁有的金鑰。您可以改為使用客戶受管金鑰。
10. 選擇匯入。

匯入外掛程式套件之後，請將其與網域建立關聯。如需說明，請參閱the section called “匯入套件並將其與網域建立關聯”。

使用 管理自訂外掛程式 AWS CLI

您可以使用 AWS CLI 來管理許多自訂外掛程式任務。

任務

- 使用 安裝自訂外掛程式 AWS CLI
- 使用 更新自訂外掛程式 AWS CLI
- 建立或更新具有 AWS KMS 金鑰安全性的自訂外掛程式
- 使用 將具有自訂外掛程式的 OpenSearch Service 網域升級至較新版本的 OpenSearch AWS CLI
- 解除安裝和檢視自訂外掛程式的取消關聯狀態

使用 安裝自訂外掛程式 AWS CLI

開始之前

您必須先將自訂外掛程式上傳至 Amazon Simple Storage Service (Amazon S3) 儲存貯體，才能將自訂外掛程式與網域建立關聯。儲存貯體必須位於您打算使用外掛程式的相同 AWS 區域 位置。如需有關將物件新增至 S3 儲存貯體的資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的上傳物件。

如果您的外掛程式包含敏感資訊，請在上傳時以 S3-managed金鑰指定伺服器端加密。上傳檔案後，請記下其 S3 路徑。路徑格式為 `s3://amzn-s3-demo-bucket/file-path/file-name`。

> **Note**
>
> 建立外掛程式時，您可以指定 AWS Key Management Service (AWS KMS) 金鑰，選擇性地保護自訂外掛程式。如需相關資訊，請參閱the section called “建立或更新具有 AWS KMS 金鑰安全性的自訂外掛程式”。

使用 安裝自訂外掛程式 AWS CLI

1. 執行下列 create-package 命令來為您的自訂外掛程式建立新的套件，確保符合下列要求：

   - 儲存貯體和金鑰位置必須指向您執行命令之帳戶中 S3 儲存貯體中的外掛程式.zip檔案。
   - S3 儲存貯體必須位於建立套件的相同區域中。
   - ZIP-PLUGIN 套件僅支援 .zip 檔案。
   - .zip 檔案的內容必須遵循外掛程式預期的目錄結構。
   - 的值格式--engine-version必須為 OpenSearch_*{MAJOR}*.*{MINOR}*。例如：**OpenSearch_2.17**。

   將*#####*取代為您自己的資訊：

   ```
   aws opensearch create-package \
       --package-name package-name \
       --region region \
       --package-type ZIP-PLUGIN \
       --package-source S3BucketName=amzn-s3-demo-bucket,S3Key=s3-key \
       --engine-version opensearch-version
   ```

2. （選用）使用 describe-packages 命令檢視create-package操作的狀態，包括任何驗證和安全性漏洞問題清單。將*#####*取代為您自己的資訊：

   ```
   aws opensearch describe-packages \
       --region region  \
       --filters '[{"Name": "PackageType","Value": ["ZIP-PLUGIN"]}, {"Name":
    "PackageName","Value": ["package-name"]}]'
   ```

   此命令會傳回與以下內容相似的資訊：

   ```
   {
   ```

```
    "PackageDetailsList": [{
        "PackageID": "pkg-identifier",
        "PackageName": "package-name",
        "PackageType": "ZIP-PLUGIN",
        "PackageStatus": "VALIDATION_FAILED",
        "CreatedAt": "2024-11-11T13:07:18.297000-08:00",
        "LastUpdatedAt": "2024-11-11T13:10:13.843000-08:00",
        "ErrorDetails": {
            "ErrorType": "",
            "ErrorMessage": "PluginValidationFailureReason : Dependency Scan
 reported 3 vulnerabilities for the plugin: CVE-2022-23307, CVE-2019-17571,
 CVE-2022-23305"
        },
        "EngineVersion": "OpenSearch_2.15",
        "AllowListedUserList": [],
        "PackageOwner": "OWNER-XXXX"
    }]
}
```

> **Note**
>
> 在create-package操作期間，Amazon OpenSearch Service 會檢查版本相容性、支援的外掛程式延伸模組和安全性漏洞ZIP-PLUGIN的值。使用 Amazon Inspector 服務掃描安全漏洞。這些檢查的結果會顯示在 API 回應的 ErrorDetails 欄位中。

3. 使用 associate-package 命令，使用上一個步驟中建立之套件的套件 ID，將外掛程式與您選擇的 OpenSearch Service 網域建立關聯。

> **Tip**
>
> 如果您有多個外掛程式，您可以改為使用 associate-packages 命令，在單一操作中將多個套件與網域建立關聯。

將*#####*取代為您自己的資訊：

```
aws opensearch associate-package \
    --domain-name domain-name \
    --region region \
    --package-id package-id
```

> **Note**
>
> 外掛程式是使用藍/綠部署程序進行安裝和解除安裝。

4. （選用）使用 list-packages-for-domain 命令來檢視關聯的狀態。關聯狀態會隨著工作流程從 進展ASSOCIATING至 而變更ACTIVE。外掛程式安裝完成且外掛程式已就緒可供使用後，關聯狀態會變更為 ACTIVE。

   使用您的資訊取代######。

```
aws opensearch list-packages-for-domain \
    --region region \
    --domain-name domain-name
```

使用 更新自訂外掛程式 AWS CLI

使用 update-package 命令來變更外掛程式。

> **Note**
>
> 當您指定 AWS Key Management Service (AWS KMS) 金鑰更新外掛程式時，您可以選擇保護自訂外掛程式。如需相關資訊，請參閱the section called “建立或更新具有 AWS KMS 金鑰安全性的自訂外掛程式”。

使用 更新自訂外掛程式 AWS CLI

- 執行下列命令。使用您的資訊取代######。

```
aws opensearch update-package \
    --region region \
    --package-id package-id \
    --package-source S3BucketName=amzn-s3-demo-bucket,S3Key=s3-key \
    --package-description description
```

更新套件後，您可以使用 associate-package 或 associate-packages 命令，將套件更新套用至網域。

> **Note**
>
> 您可以使用 稽核、建立、更新、關聯和取消關聯外掛程式上的操作 AWS CloudTrail。如需詳細資訊，請參閱使用 監控 Amazon OpenSearch Service API 呼叫 AWS CloudTrail。

建立或更新具有 AWS KMS 金鑰安全性的自訂外掛程式

您可以在建立或更新外掛程式時，透過指定 AWS KMS 金鑰來保護自訂外掛程式。若要達成此目的，請將 `PackageEncryptionOptions` 設定為 ，`true`並指定金鑰的 Amazon Resource Name (ARN)，如下列範例所示。

範例：建立具有 AWS KMS 金鑰安全性的自訂外掛程式

```
aws opensearch create-package \
    --region us-east-2  --package-name my-custom-package \
    --package-type ZIP-PLUGIN \
    --package-source S3BucketName=amzn-s3-demo-bucket,S3Key=my-s3-key
    --engine-version OpenSearch_2.15
"PackageConfigOptions": {
     ...
  }
  "PackageEncryptionOptions": {
    "Enabled": true,
    "KmsKeyId":"arn:aws:kms:us-east-2:111222333444:key/2ba228d5-1d09-456c-ash9-
daf42EXAMPLE"
  }
```

範例：使用 AWS KMS 金鑰安全性更新自訂外掛程式

```
aws opensearch update-package \
    --region us-east-2  --package-name my-custom-package \
    --package-type ZIP-PLUGIN \
    --package-source S3BucketName=amzn-s3-demo-bucket,S3Key=my-s3-key
    --engine-version OpenSearch_2.15
"PackageConfigOptions": {
     ...
  }
  "PackageEncryptionOptions": {
    "Enabled": true,
    "KmsKeyId":"arn:aws:kms:us-east-2:111222333444:key/2ba228d5-1d09-456c-ash9-
daf42EXAMPLE"
```

```
}
```

> **Important**
>
> 如果您指定的 AWS KMS 金鑰已停用或刪除，可能會讓相關聯的叢集無法運作。

如需與自訂套件 AWS KMS 整合的詳細資訊，請參閱the section called “Amazon OpenSearch Service 自訂套件 AWS KMS 整合”。

使用 將具有自訂外掛程式的 OpenSearch Service 網域升級至較新版本的 OpenSearch AWS CLI

當您需要將使用自訂外掛程式的 OpenSearch Service 網域升級至較新版本的 OpenSearch 時，請完成下列程序。

使用 將具有自訂外掛程式的 OpenSearch Service 網域升級至較新版本的 OpenSearch AWS CLI

1. 使用 create-package 命令為您的外掛程式建立新的套件，以指定新的 OpenSearch 版本。

   確保所有引擎版本的外掛程式的套件名稱相同。變更套件名稱會導致網域升級程序在藍/綠部署期間失敗。
2. 遵循中的步驟，將您的網域升級至更高版本the section called “升級網域”。

   在此過程中，Amazon OpenSearch Service 會取消與舊版外掛程式套件的關聯，並使用藍/綠部署安裝新版本。

解除安裝和檢視自訂外掛程式的取消關聯狀態

若要從任何網域解除安裝外掛程式，您可以使用 dissociate-package 命令。執行此命令也會移除任何相關的組態或授權套件。然後，您可以使用 list-packages-for-domain 命令來檢視取消關聯的狀態。

> **Tip**
>
> 您也可以使用 dissociate-packages 命令，在單一操作中從網域解除安裝多個外掛程式。

解除安裝並檢視自訂外掛程式的取消關聯狀態

1. 在每個索引中停用外掛程式。這必須在取消外掛程式套件的關聯之前完成。

如果您在從每個索引停用外掛程式之前嘗試解除安裝外掛程式，藍/綠部署程序會維持在 `Processing` 狀態。

2. 執行下列命令來解除安裝外掛程式。使用您的資訊取代*######*。

```
aws opensearch dissociate-package \
    --region region \
    --package-id plugin-package-id \
    --domain-name domain name
```

3. （選用）執行 list-packages-for-domain 命令，以檢視取消關聯的狀態。

Amazon OpenSearch Service 自訂套件 AWS KMS 整合

Amazon OpenSearch Service 自訂套件預設提供加密，以使用 保護靜態`ZIP-PLUGIN`套件 AWS 受管金鑰。

- AWS 擁有的金鑰 – Amazon OpenSearch Service 自訂套件預設使用這些金鑰來自動加密`ZIP-PLUGIN`套件。您無法檢視、管理或使用 AWS 擁有的金鑰 或稽核其使用方式。但是，您不需要採取任何動作或變更任何程式，即可保護加密您資料的金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 AWS 擁有的金鑰。
- 客戶受管金鑰 – 您可以在建立`ZIP-PLUGIN`自訂套件時選擇客戶受管金鑰， AWS 擁有的金鑰 在現有的 上新增第二層加密。

  Amazon OpenSearch Service 自訂套件支援使用您建立、擁有和管理的對稱客戶受管金鑰，以透過現有 AWS 擁有的加密新增第二層加密。由於您可以完全控制此加密層，因此您可以執行下列任務：

  - 建立和維護金鑰政策
  - 建立和維護 AWS Identity and Access Management (IAM) 政策和授予
  - 啟用和停用金鑰政策
  - 輪換金鑰密碼編譯資料
  - 新增標籤
  - 建立金鑰別名
  - 排程金鑰刪除

如需更多資訊，請參閱 AWS Key Management Service 開發人員指南中的客戶受管金鑰。

> **Note**
>
> Amazon OpenSearch Service 自訂套件會自動使用 AWS 擁有的金鑰 免費啟用靜態加密。不過，當您使用客戶受管金鑰時會產生 AWS KMS 費用。如需定價的詳細資訊，請參閱 AWS Key Management Service 定價。

Amazon OpenSearch Service 自訂套件服務如何在 中使用授予 AWS KMS

OpenSearch Service 自訂套件需要授予才能使用客戶受管金鑰。

當您建立使用客戶受管金鑰加密的`ZIP-PLUGIN`套件時，Amazon OpenSearch Service 自訂套件服務會透過傳送 CreateGrant 請求至 來代表您建立授權 AWS KMS。授予 AWS KMS OpenSearch Service 存取您帳戶中的 AWS KMS 金鑰。OpenSearch Service 自訂套件建立的授予具有限制條件，僅在請求包含具有自訂套件 ID 的加密內容時，才允許操作。

Amazon OpenSearch Service 自訂套件需要授予 ，才能將客戶受管金鑰用於下列內部操作：

作業	Description
`DescribeKey`	傳送`DescribeKey` 請求至 AWS KMS ，以驗證在建立外掛程式套件時輸入的對稱客戶受管金鑰 ID 是否有效。
`GenerateDataKeyWithoutPlaintext`	將`GenerateDataKeyWithoutPlaintext` 請求傳送至 AWS KMS ，以產生由客戶受管金鑰加密的資料金鑰。
`GenerateDataKey`	將`GenerateDataKey` 請求傳送至 AWS KMS 以產生資料金鑰，以在內部複製套件時加密套件。
`Decrypt`	將`Decrypt`請求傳送至 AWS KMS 以解密加密的資料金鑰，以便用來解密您的資料。

您可以隨時撤銷授予的存取權，或移除服務對客戶受管金鑰的存取權。如果您這麼做，OpenSearch Service 將無法存取客戶受管金鑰加密的任何資料，這會影響依賴該資料的操作。例如，如果您嘗試關聯 OpenSearch Service 無法存取的外掛程式套件，操作會傳回`AccessDeniedException`錯誤。

建立客戶自管金鑰

您可以使用 AWS 管理主控台 或 AWS KMS APIs 來建立對稱客戶受管金鑰。

建立對稱客戶受管金鑰

- 請遵循《 AWS Key Management Service 開發人員指南》中的建立 KMS 金鑰中的步驟。

金鑰政策

金鑰政策會控制客戶受管金鑰的存取權限。每個客戶受管金鑰都必須只有一個金鑰政策，其中包含決定誰可以使用金鑰及其使用方式的陳述式。在建立客戶受管金鑰時，可以指定金鑰政策。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的在 AWS KMS中使用金鑰政策。

若要將客戶受管金鑰與外掛程式資源搭配使用，您必須在金鑰政策中允許下列 API 操作：

- `kms:CreateGrant`：新增客戶受管金鑰的授權。授予控制對指定 AWS KMS 金鑰的存取權，允許存取 OpenSearch Service 自訂套件所需的操作。如需使用授予的詳細資訊，請參閱 AWS KMS 開發人員指南。

  這可讓 OpenSearch Service 執行下列動作：

  - 呼叫 `GenerateDataKeyWithoutPlainText` 以產生加密的資料金鑰，並將它存放以供進一步驗證。
  - 呼叫 `GenerateDataKey` 在內部複製外掛程式套件。
  - 呼叫 `Decrypt` 以在內部存取外掛程式套件。
  - 設定淘汰主體，以允許服務執行 `RetireGrant`。
- `kms:DescribeKey` – 提供客戶受管金鑰詳細資訊，以允許 OpenSearch Service 驗證金鑰。
- `kms:GenerateDataKey`、 `kms:GenerateDataKeyWithoutPlaintextkms:Decrypt`- 授予 OpenSearch Service 自訂套件在授予中使用這些操作的存取權。

以下是您可以為 OpenSearch Service 自訂套件新增的政策陳述式範例：

```
"Statement" : [
  {
    "Sid" : "Allow access to principals authorized to use OpenSearch Service custom
 packages",
    "Effect" : "Allow",
    "Principal" : {
```

```
      "AWS" : "*"
    },
    "Action" : [
      "kms:CreateGrant",
      "kms:GenerateDataKey",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:Decrypt"
    ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "custom-packages.region.amazonaws.com"
      },
      "StringEquals" : {
        "kms:EncryptionContext:packageId": "Id of the package"
      }
    }
  },
  {
    "Sid" : "Allow access to principals authorized to use Amazon OpenSearch Service
 custom packages",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [
      "kms:DescribeKey"
    ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "custom-packages.region.amazonaws.com"
      }
    }
  }
]
```

如需在政策中指定許可的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的 中的金鑰政策 AWS KMS。

如需對金鑰存取進行故障診斷的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的故障診斷 AWS KMS 許可。

指定 Amazon OpenSearch Service 自訂套件的客戶受管金鑰

您可以將客戶受管金鑰指定為ZIP-PLUGIN套件的第二層加密。

當您建立外掛程式套件時，您可以輸入金鑰 ID 來指定資料 AWS KMS 金鑰，OpenSearch Service 自訂套件會使用該 ID 來加密外掛程式套件。

AWS KMS 金鑰 ID — AWS KMS 客戶受管金鑰的金鑰識別符。輸入金鑰 ID、金鑰 ARN、別名名稱或別名 ARN。

Amazon OpenSearch Service 自訂套件加密內容

加密內容是一組選用的金鑰值對，包含資料的其他相關內容資訊。

AWS KMS 使用加密內容做為額外的已驗證資料，以支援已驗證的加密。當您在加密資料的請求中包含加密內容時， 會將加密內容 AWS KMS 繫結至加密的資料。若要解密資料，您必須在請求中包含相同的加密內容。

Amazon OpenSearch Service 自訂套件加密內容

Amazon OpenSearch Service 自訂套件會在所有 AWS KMS 密碼編譯操作中使用相同的加密內容，其中金鑰為 `packageId` ，而值為外掛程式套件`package-id`的 。

使用加密內容進行監控

當您使用對稱客戶受管金鑰來加密外掛程式套件時，您可以使用稽核記錄和日誌中的加密內容來識別客戶受管金鑰的使用方式。加密內容也會出現在 AWS CloudTrail 或 Amazon CloudWatch Logs 產生的日誌中。

使用加密內容控制對客戶受管金鑰的存取

您也可以在金鑰政策和 IAM 政策中，使用加密內容來控制對對稱客戶受管金鑰的存取。您也可以在授予中使用加密內容條件。

OpenSearch Service 自訂套件在授予中使用加密內容限制，以控制對您帳戶或區域中客戶受管金鑰的存取。授予條件會要求授予允許的操作使用指定的加密內容。

以下是授予特定加密內容之客戶受管金鑰存取權的金鑰政策陳述式範例。此政策陳述式中的條件會要求具有指定加密內容的加密內容條件。

```
{
    "Sid": "Enable DescribeKey",
```

```
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
},
{
    "Sid": "Enable OpenSearch Service custom packages to use the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
    },
    "Action" : [
         "kms:CreateGrant",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals" : {
            "kms:EncryptionContext:packageId": "ID of the package"
         }
    }
}
```

監控 OpenSearch 自訂套件服務的加密金鑰

當您使用 AWS KMS 客戶受管金鑰搭配 OpenSearch Service 自訂套件服務資源時，您可以使用 CloudTrail 或 CloudWatch Logs 來追蹤 OpenSearch 自訂套件傳送至 的請求 AWS KMS。

進一步了解

下列資源會提供有關靜態資料加密的詳細資訊。

- 如需 AWS KMS 基本概念的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南AWS KMS keys》中的 。
- 如需 安全最佳實務的詳細資訊 AWS KMS，請參閱 AWS Key Management Service 最佳實務AWS 規範指南。

在 Amazon OpenSearch Service 中安裝第三方外掛程式

Amazon OpenSearch Service 支援所選合作夥伴的第三方外掛程式。這些外掛程式可以透過自訂分析器、字符化器或加密功能等其他功能來增強 OpenSearch 設定。遵循第三方開發人員提供的特定安裝和組態指示，以確保與您的 OpenSearch Service 網域適當整合。

> **Note**
>
> 您必須直接從第三方開發人員取得和維護有效的授權。有些供應商可能不會在所有 中啟用其外掛程式 AWS 區域，因此請向外掛程式供應商確認可用性。

下列第三方外掛程式可與 OpenSearch Service 搭配使用：

- Portal26 加密外掛程式 (Titanium-lockbox) – 使用 NIST FIPS 140-2 認證的加密，在資料編製索引時加密資料。它包含自帶金鑰 (BYOK) 支援，可讓您管理加密金鑰以增強安全性。外掛程式由 Portal26 提供，且需要 OpenSearch 2.15 版或更新版本。
- 名稱比對 (RNI) – 以超過 24 種語言比對名稱、組織、地址和日期，以改善安全性和合規性。外掛程式由 Babel Street 提供，需要 OpenSearch 2.15 版或更新版本。

主題

- 先決條件
- 安裝第三方外掛程式
- 後續步驟

先決條件

安裝第三方外掛程式之前，請執行下列步驟：

- 取得外掛程式組態和授權檔案，並將其上傳至 Amazon S3 儲存貯體。儲存貯體必須與網域 AWS 區域 位於相同的 。
- 第三方外掛程式是一種自訂外掛程式。請確定網域符合自訂外掛程式的先決條件。

安裝第三方外掛程式

若要將第三方外掛程式與 OpenSearch Service 網域建立關聯，您必須先上傳三個不同的套件：授權套件、組態套件和外掛程式套件。

- 授權套件包含與外掛程式相關聯的授權資訊或中繼資料，格式為 .json 或 .xml。
- 組態套件包含外掛程式組態檔案，以及支援資產和設定。這些檔案會定義外掛程式與 OpenSearch 的行為或整合方式。
- 外掛程式套件包含編譯的外掛程式二進位檔，這是 OpenSearch 執行的可執行程式碼。這是外掛程式功能的核心。

上傳這兩個套件之後，您可以將外掛程式和授權與相容的網域建立關聯。

主控台

若要將第三方外掛程式與網域建立關聯，請先將外掛程式授權和組態匯入為套件。

安裝第三方外掛程式

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇套件。
3. 首先，匯入授權套件。選擇 Import package (匯入套件)。
4. 針對套件類型，選擇授權。
5. 針對套件來源，輸入 Amazon S3 中授權 JSON 或 XML 檔案的路徑。
6. 選擇匯入。套件會出現在套件頁面的授權索引標籤上。
7. 現在，匯入外掛程式組態。再次選擇匯入套件。
8. 針對套件類型，選擇組態。
9. 針對套件來源，輸入 Amazon S3 中外掛程式組態 ZIP 檔案的路徑。
10. 選擇匯入。
11. 最後，匯入外掛程式本身。選擇 Import package (匯入套件)。
12. 針對套件類型，選擇外掛程式。
13. 在套件來源中，輸入 Amazon S3 中外掛程式 ZIP 檔案的路徑。
14. 選取外掛程式支援的 OpenSearch 引擎版本。
15. 選擇匯入。

將第三方外掛程式與網域建立關聯

1. 現在，將外掛程式授權和組態與網域建立關聯。在左側導覽窗格中選擇 Domains (網域)。
2. 選擇網域名稱以開啟其叢集組態。
3. 導覽至外掛程式索引標籤。
4. 選擇關聯套件，然後選取您剛匯入的外掛程式、授權和組態套件。
5. 選擇選取。
6. 選擇下一步。檢閱要關聯的套件，然後選擇關聯。

CLI

首先，使用 create-package 命令來建立新的套件，其中包含外掛程式授權。必須S3Key指向 Amazon S3 中的 .json 或 .xml 檔案，其中包含授權文字或中繼資料。

```
aws opensearch create-package \
  --package-name plugin-license-package \
  --package-type PACKAGE-LICENSE \
  --package-source S3BucketName=my-bucket,S3Key=licenses/my-plugin-license.json
```

再次使用 create-package 命令來建立包含外掛程式組態的套件。必須S3Key指向 Amazon S3 中遵循外掛程式預期目錄結構的 .zip 檔案。

```
aws opensearch create-package \
  --package-name plugin-config-package \
  --package-type PACKAGE-CONFIG \
  --package-source S3BucketName=my-bucket,S3Key=path/to/package.zip
```

再次使用 create-package 命令來建立包含外掛程式本身的套件。必須S3Key指向 Amazon S3 中的外掛程式 .zip 檔案。

```
aws opensearch create-package \
  --package-name plugin-package \
  --package-type ZIP-PLUGIN \
  --package-source S3BucketName=my-bucket,S3Key=path/to/package.zip
```

最後，使用 associate-package 命令，透過指定每個 的套件 IDs，將合作夥伴外掛程式、授權和組態連結至相容的網域。指定外掛程式 ID 做為其他套件的先決條件，這表示它必須在其他套件之前與網域建立關聯。

```
aws opensearch associate-packages \
  --domain-name my-domain \
  --package-list '[{"PackageID": "plugin-package-id"},{"PackageID": "license-package-id","PrerequisitePackageIDList":["plugin-package-id"]},{"PackageID":"config-package-id","PrerequisitePackageIDList":["plugin-package-id"]}]'
```

後續步驟

當關聯完成時，您可以在特定索引上啟用外掛程式，或根據您的需求視需要進行設定。若要將第三方外掛程式功能套用至特定索引，請在索引建立期間修改索引設定，或更新現有的索引。例如，如果您的第三方外掛程式包含自訂分析器，請在索引設定中參考它。

若要一致地跨多個索引套用外掛程式功能，請使用包含外掛程式組態的索引範本。請務必參閱外掛程式文件，了解如何為 OpenSearch 設定設定其功能。

使用 SQL 查詢您的 Amazon OpenSearch Service 資料

您可以使用 SQL 來查詢您的 Amazon OpenSearch Service，而不是使用基於 JSON 的 OpenSearch 查詢 DSL。如果您已熟悉 SQL，或想要透過使用它的應用程式來整合您的網域，則使用 SQL 進行查詢非常實用。執行 OpenSearch 或 Elasticsearch 6.5 或更高版本的網域支援 SQL。

> **Note**
>
> 本文件說明 OpenSearch Service 與各種 SQL 外掛程式版本，以及 JDBC 和 ODBC 驅動程式之間的版本相容性。如需基本和複雜查詢、函數、中繼資料查詢和彙總函數語法的相關資訊，請參閱開放原始碼 OpenSearch 文件。

請使用下表來查找每個 OpenSearch 和 Elasticsearch 版本所支援的 SQL 外掛程式版本。

OpenSearch

OpenSearch 版本	SQL 外掛程式版本	值得注意的功能
2.19.0	2.19.0.0	
2.18.0	2.18.0.0	

OpenSearch 版本	SQL 外掛程式版本	值得注意的功能
2.17.0	2.17.0.0	
2.15.0	2.15.0.0	
2.13.0	2.13.0.0	
2.11.0	2.11.0.0	新增 PPL 語言和查詢的支援
2.9.0	2.9.0.0	新增 Spark 連接器，並支援資料表和 PromQL 函數
2.7.0	2.7.0.0	新增 `datasource` API
2.5.0	2.5.0.0	
2.3.0	2.3.0.0	新增 `maketime` 和 `makedate` 日期時間函數
1.3.0	1.3.0.0	支援預設查詢限制大小和 IN 子句，以便從值清單中進行選擇
1.2.0	1.2.0.0	新增用於視覺化效果回應格式的新協議
1.1.0	1.1.0.0	支援比對功能，作為 SQL 和 PPL 中的篩選條件
1.0.0	1.0.0.0	支援查詢資料串流

Open Distro for Elasticsearch

Elasticsearch 版本	SQL 外掛程式版本	值得注意的功能
7.10	1.13.0	視窗函數的 NULL FIRST 和 LAST，CAST() 函數，SHOW 和 DESCRIBE 命令
7.9	1.11.0	新增其他日期/時間函數，ORDER BY 關鍵字
7.8	1.9.0	

Elasticsearch 版本	SQL 外掛程式版本	值得注意的功能
7.7	1.8.0	
7.3	1.3.0	多個字串和數字運算子
7.1	1.1.0	

範例呼叫

若要使用 SQL 查詢資料，請使用以下格式傳送 HTTP 請求到 `_sql`：

```
POST domain-endpoint/_plugins/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

> **Note**
>
> 如果您的網域正在執行 Elasticsearch 而不是 OpenSearch，則格式為 `_opendistro/_sql`。

備註和差異

對 `_plugins/_sql` 的呼叫會在請求內文中包含索引名稱，因此有相同的大量、mget、和 msearch 操作的存取政策考量。一如以往，授權許可到 API 操作時請遵循最低權限的原則。

如需了解搭配使用 SQL 與精細存取控制的相關安全考量，請參閱 the section called “精細定義存取控制”。

OpenSearch SQL 外掛程式包含許多可調整設定。在 OpenSearch Service 中，使用 `_cluster/settings` 路徑，而不是外掛程式設定路徑 (`_plugins/_query/settings`)：

```
PUT _cluster/settings
{
  "transient" : {
    "plugins.sql.enabled" : true
  }
```

```
}
```

對於舊版 Elasticsearch 網域，請將 `plugins` 取代為 `opendistro`：

```
PUT _cluster/settings
{
  "transient" : {
    "opendistro.sql.enabled" : true
  }
}
```

SQL Workbench

SQL Workbench 是一個 OpenSearch Dashboards 使用者界面，可讓您執行隨需 SQL 查詢、將 SQL 轉譯為其 REST 同等項目，然後檢視結果並儲存為文字、JSON、JDBC 或 CSV。如需詳細資訊，請參閱 Query Workbench。

SQL CLI

SQL CLI 是可透過 `opensearchsql` 命令啟動的獨立 Python 應用程式。如需安裝、設定和使用的步驟，請參閱 SQL CLI。

JDBC 驅動程式

Java 資料庫連接 (JDBC) 驅動程式可讓您整合 OpenSearch Service 網域與您喜好的商業智慧 (BI) 應用程式。若要下載驅動程式，請按一下這裡。如需更多資訊，請參閱 GitHub 儲存庫。

ODBC 驅動程式

開放式資料庫連線 (ODBC) 驅動程式是適用於 Windows 和 macOS 的唯讀 ODBC 驅動程式，可讓您將 Microsoft Excel 等商業智慧和資料視覺化應用程式連線至 SQL 外掛程式。

如需安裝驅動程式的詳細資訊，請參閱 GitHub 上的 SQL 儲存庫。

Amazon OpenSearch Service 中的跨叢集搜尋

Amazon OpenSearch Service 中的跨叢集搜尋可讓您跨多個已連線網域執行查詢和彙總。通常使用多個較小的網域 (而不是單一大型網域) 會比較有意義，尤其是在執行不同類型工作負載的情況下。

透過工作負載特定網域可執行以下任務：

- 選擇特定工作負載的執行個體類型以優化每個網域。
- 建立多個工作負載間的故障隔離界限。也就是說，如果其中一個工作負載失敗，此故障會包含在該特定網域內，而且不會影響其他工作負載。
- 更輕鬆地跨網域進行擴展。

跨叢集搜尋可支援 OpenSearch Dashboards，因此您可以跨所有網域建立視覺效果和儀表板。對於在網域之間傳輸的搜尋結果，您需要支付標準 AWS 資料傳輸費用。

> **Note**
>
> 開放原始碼 OpenSearch 也有跨叢集搜尋的文件。相較於 受管 Amazon OpenSearch Service 網域，開放原始碼叢集的設定有很大差異。最值得注意的是，在 OpenSearch Service 中，您使用 AWS 管理主控台 而不是透過 cURL 來設定跨叢集連線。此外，除了精細存取控制之外，受管服務還會使用 AWS Identity and Access Management (IAM) 進行跨叢集身分驗證。因此，我們建議您使用本文件而非開放原始碼 OpenSearch 文件，為您的網域設定跨叢集搜尋。

主題

- 限制
- 跨叢集搜尋先決條件
- 跨叢集搜尋定價
- 設定連線
- 移除連線
- 設定安全性和範例演練
- OpenSearch Dashboards

限制

跨叢集搜尋有多項重要限制：

- 您無法將 Elasticsearch 網域連線至 OpenSearch 網域。
- 您無法連線至自我管理的 OpenSearch/Elasticsearch 叢集。
- 若要連線區域間的網域，這兩個網域都必須位於 Elasticsearch 7.10 或更高版本或 OpenSearch 上。

- 一個網域最多可以有 20 個外寄連線。同樣地，一個網域最多可以有 20 個傳入連線。換句話說，一個網域最多可連線到 20 個其他網域。
- 來源網域的版本必須與目的地網域相同或更高。如果您在兩個網域之間設定雙向連線，並且想要升級其中一個或兩個網域，您必須先刪除其中一個連線。
- 您無法將自訂字典或 SQL 使用跨叢集搜尋搭配使用。
- 您無法使用 CloudFormation 來連接網域。
- 您無法在 M3 或高載 (T2 和 T3) 執行個體上使用跨叢集搜尋。

跨叢集搜尋先決條件

設定跨叢集搜尋前，請確認網域符合下列需求：

- 兩個 OpenSearch 網域，或 6.7 版或更新版本的 Elasticsearch 網域
- 已啟用精細存取控制
- 已啟用節點對節點加密
- 如果任一網域位於 VPC 中，則必須透過 VPC 對等互連或傳輸閘道來連接網域，且安全群組必須允許它們之間的流量。

跨叢集搜尋定價

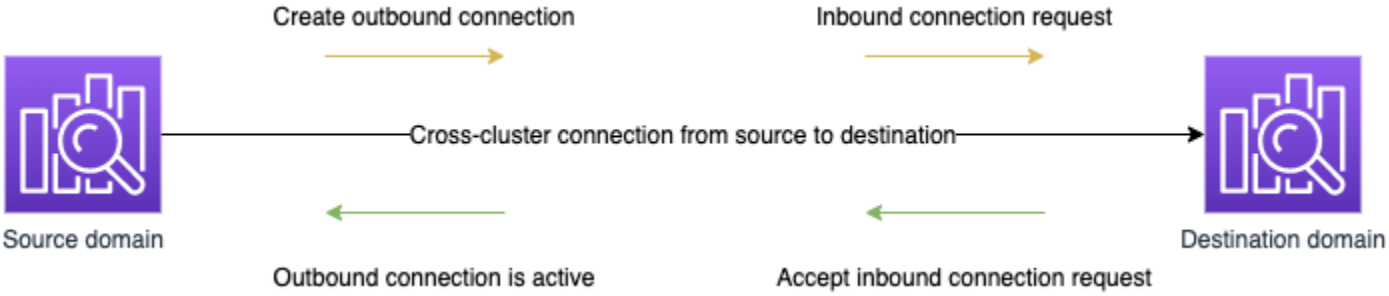
跨網域搜尋無須額外付費。

設定連線

「來源」網域是指跨叢集搜尋請求的來源位置。換句話說，來源網域是您傳送初始搜尋請求的目標位置。

「目的地」網域是來源網域查詢的網域。

跨叢集連線為單向 (從來源到目的地網域)。也就是說，目的地網域無法查詢來源網域。不過，您可以設定相反方向的其他連線。



來源網域會建立「傳出」至目的地網域的連線。目的地網域會接收來自來源網域的「傳入」連線請求。

如何設定連線

1. 在網域儀表板上，選擇網域並移至 Connections (連線) 索引標籤。
2. 在 Outbound connections (傳出連線) 區段中，選擇 Request (請求)。
3. 對於 Connection alias (連線別名)，輸入連線的名稱。
4. 選擇連線到您 AWS 帳戶 和 區域中的網域，或另一個帳戶或區域中的網域。
   - 若要連線到 AWS 帳戶 和 區域中的叢集，請從下拉式功能表中選取網域，然後選擇請求。
   - 若要連線到另一個 AWS 帳戶 或區域中的叢集，請選取遠端網域的 ARN，然後選擇請求。若要連線區域間的網域，這兩個網域必須執行 Elasticsearch 7.10 或更高版本或 OpenSearch。
5. 若要略過叢集查詢無法使用的叢集，請選取略過無法使用。此設定可確保跨叢集查詢即使一或多個遠端叢集發生故障，仍會傳回部分結果。
6. 跨叢集搜尋會先驗證連線請求，確認是否符合先決條件。如果發現網域不相容，連線請求則會進入 `Validation failed` 狀態。
7. 連線請求驗證成功後，則會傳送至目的地網域，需要在這裡進行核准。在此核准發生前，連線會保持在 `Pending acceptance` 狀態。在目的地網域接受連線請求後，狀態會變更為 `Active`，而目的地網域會變為可供查詢。
   - 網域頁面會顯示目的地網域的整體網域運作狀態和執行個體運作狀態詳細資訊。只有網域擁有者才能靈活建立、檢視、移除和監控與其網域的連線。

建立連線後，在連線網域節點之間流動的任何流量都會加密。如果您將 VPC 網域連接至非 VPC 網域，且該非 VPC 網域是可從網際網路接收流量的公有端點，網域之間的跨叢集流量仍為加密且安全的狀態。

移除連線

移除連線會停止其索引上的任何跨叢集操作。

1. 在網域儀表板上，前往 Connections (連線) 索引標籤。
2. 選取您要移除的網域連線並選擇 Delete (刪除)，然後確認刪除。

您可以在來源或目的地網域上執行這些步驟，以移除連線。移除連線後，在 15 天內仍可見，狀態為 `Deleted`。

您無法刪除具有作用中跨叢集連線的網域。若要刪除網域，請先從該網域中移除所有傳入和傳出連線。這可確保您在刪除網域前，將跨叢集網域使用者納入考量。

設定安全性和範例演練

1. 您會將跨叢集搜尋請求傳送至來源網域。
2. 來源網域會根據其網域存取政策評估請求。由於跨叢集搜尋需要精細存取控制，建議您在來源網域上使用開放式存取政策。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/src-domain/*"
    }
  ]
}
```

> **Note**
>
> 如果您在路徑中包含遠端索引，則必須對網域 ARN 中的 URI 進行 URL 編碼。例如，使用 `arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst%3Aremote_index` 而非 `arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst:remote_index`。

除了精細存取控制以外，如果您還選擇使用限制性存取政策，政策則必須至少允許對 `es:ESHttpGet` 的存取權。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/test-user"
        ]
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/src-domain/*"
    }
  ]
}
```

3. 來源網域上的精細存取控制會評估請求：

   - 請求是否以有效的 IAM 或 HTTP 基本登入資料簽署？
   - 如果是這樣，使用者是否具備執行搜尋和存取資料的許可？

   如果請求只搜尋目的地網域上的資料 (例如 `dest-alias:dest-index/_search`)，您只需要目的地網域上的許可。

如果請求同時搜尋兩個網域上的資料 (例如 source-index,dest-alias:dest-index/_search)，您需要兩個網域上的許可。

在精細存取控制中，除了相關索引的標準read或indices:admin/shards/search_shards許可之外，使用者還必須擁有 search許可。

4. 來源網域會將請求傳遞至目的地網域。目的地網域會根據其網域存取政策評估此請求。您必須在目的地網域上包含 es:ESCrossClusterGet 許可：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/dst-domain"
    }
  ]
}
```

確認已針對 /dst-domain (而不是 /dst-domain/*) 套用 es:ESCrossClusterGet 許可。

不過，此最低政策只允許跨叢集搜尋。若執行其他操作 (例如為文件編製索引和執行標準搜尋)，則需要額外的許可。建議在目的地網域上使用下列政策：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
```

```
            "*"
          ]
        },
        "Action": [
          "es:ESHttp*"
        ],
        "Resource": "arn:aws:es:us-east-1:111122223333:domain/dst-domain/*"
      },
      {
        "Effect": "Allow",
        "Principal": {
          "AWS": "*"
        },
        "Action": "es:ESCrossClusterGet",
        "Resource": "arn:aws:es:us-east-1:111122223333:domain/dst-domain"
      }
    ]
}
```

> **Note**
>
> 根據預設，網域之間的所有跨叢集搜尋請求在傳輸中都會經過加密，做為節點對節點加密的一部分。

5. 目的地網域會執行搜尋並將結果傳回至來源網域。
6. 來源網域會將其自身結果 (如果有的話) 與來自目的地網域的結果加以結合，然後將其傳回給您。
7. 建議針對測試請求使用 Postman：

   - 在目的地網域上，為文件編製索引：

   ```
   POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1

   {
     "Dracula": "Bram Stoker"
   }
   ```

   - 若要從來源網域查詢此索引，請在查詢內包含目的地網域的連線別名。

   ```
   GET https://src-domain.us-east-1.es.amazonaws.com/<connection_alias>:books/
   _search
   ```

```
{
    ...
  "hits": [
    {
      "_index": "source-destination:books",
      "_type": "_doc",
      "_id": "1",
      "_score": 1,
      "_source": {
        "Dracula": "Bram Stoker"
      }
    }
  ]
}
```

您可以在網域儀表板的 Connections (連線) 索引標籤上找到連線別名。

- 如果在具有連線別名 cluster_b 的 domain-a -> domain-b 與具有連線別名 cluster_c 的 domain-a -> domain-c 之間設定連線，請搜尋 domain-a、domain-b 和 domain-c，如下所示：

```
GET https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_search
{
  "query": {
    "match": {
      "user": "domino"
    }
  }
}
```

回應

```
{
  "took": 150,
  "timed_out": false,
  "_shards": {
    "total": 3,
    "successful": 3,
    "failed": 0,
    "skipped": 0
```

```
  },
  "_clusters": {
    "total": 3,
    "successful": 3,
    "skipped": 0
  },
  "hits": {
    "total": 3,
    "max_score": 1,
    "hits": [
      {
        "_index": "local_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 1,
        "_source": {
          "user": "domino",
          "message": "This is message 1",
          "likes": 0
        }
      },
      {
        "_index": "cluster_b:b_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 2,
        "_source": {
          "user": "domino",
          "message": "This is message 2",
          "likes": 0
        }
      },
      {
        "_index": "cluster_c:c_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 3,
        "_source": {
          "user": "domino",
          "message": "This is message 3",
          "likes": 0
        }
      }
    ]
```

```
  }
}
```

如果您未選擇在連線設定中略過無法使用的叢集，則您搜尋的所有目的地叢集都必須可供搜尋請求成功執行。否則，整個請求會失敗，即使其中一個網域無法使用，也不會傳回任何搜尋結果。

OpenSearch Dashboards

您可以透過與從單一網域相同的方式，從多個連線網域視覺化資料，但您必須使用 `connection-alias:index` 存取遠端索引時除外。因此，索引模式必須符合 `connection-alias:index`。

適用於 Amazon OpenSearch Service 的 Learning to Rank

OpenSearch 使用名為 BM-25 的概率排名架構來計算相關性分數。如果特殊關鍵字更頻繁地出現在文件中，BM-25 會為該文件指派較高的相關性分數。然而，此架構並不考慮像點選資料這樣的使用者行為，這樣可以進一步改善相關性。

Learning to Rank 是一個開放原始碼外掛程式，可讓您使用機器學習和行為資料來調整文件的相關性。其會使用 XGBoost 和 Ranklib 程式庫中的模型來重新評分搜尋結果。Elasticsearch LTR 外掛程式最初是由 OpenSource Connections 開發，且維基媒體基金會、Snagajob Engineering、Bonsai 和 Yelp Engineering 亦提供重大協助。該外掛程式的 OpenSearch 版本源自 Elasticsearch LTR 外掛程式。

Learning to Rank 需要 OpenSearch 或 Elasticsearch 7.7 或更高版本。若要使用 Learning to Rank 外掛程式，您必須擁有完整的管理員許可。如需詳細資訊，請參閱 the section called “修改主要使用者”。

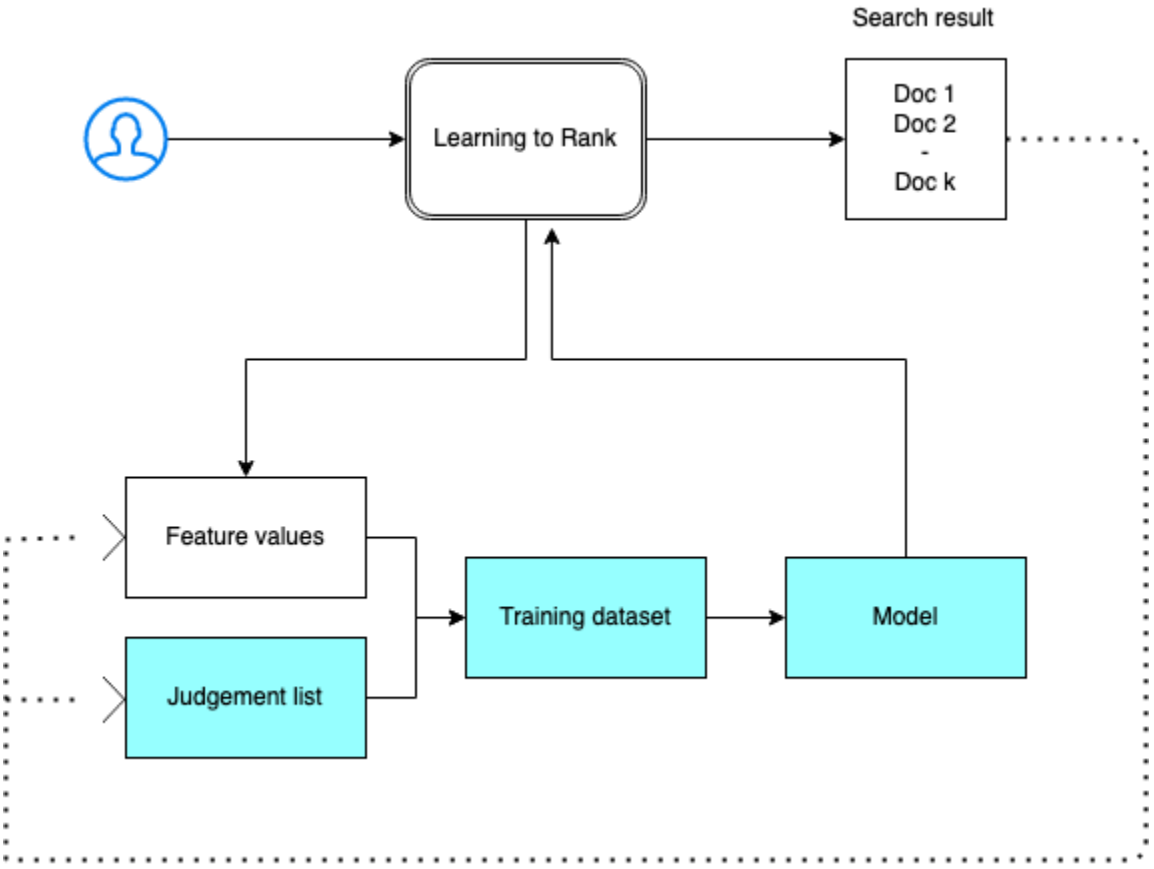
> **Note**
>
> 本文件提供 Learning to Rank 外掛程式的一般概觀，並協助您開始使用它。如需完整文件，包括詳細步驟和 API 說明，請參閱 Learning to Rank 文件。

主題

- Learning to Rank 入門
- Learning to Rank API

Learning to Rank 入門

您需要提供判斷清單，準備訓練資料集，以及在 Amazon OpenSearch Service 之外訓練模型。藍色零件出現在 OpenSearch Service 以外：



步驟 1：初始化外掛程式

若要初始化 Learning to Rank 外掛程式，請將下列請求傳送至您的 OpenSearch Service 網域：

```
PUT _ltr
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : ".ltrstore"
}
```

此命令會建立一個隱藏的 `.ltrstore` 索引，它會存放中繼資料資訊，例如功能集和模型。

步驟 2：建立判斷清單

> **Note**
>
> 您必須在 OpenSearch Service 之外執行此步驟。

判斷清單是機器學習模型從中學習的範例集合。您的判斷清單應包含對您很重要的關鍵字，以及每個關鍵字的一組分級文件。

在此例中，我們有一個影片資料集的判斷清單。等級 4 表示完美匹配。等級 0 表示最差匹配。

等級	關鍵字	文件 ID	影片名稱
4	rambo	7555	Rambo
3	rambo	1370	Rambo III
3	rambo	1369	Rambo: First Blood Part II
3	rambo	1368	First Blood

請按下列格式準備判斷清單：

```
4 qid:1 # 7555 Rambo
3 qid:1 # 1370 Rambo III
3 qid:1 # 1369 Rambo: First Blood Part II
3 qid:1 # 1368 First Blood

where qid:1 represents "rambo"
```

如需判斷清單的更完整範例，請參閱影片判斷。

您可以在人類註釋器的幫助下手動建立此判斷清單，或者從分析資料中以程式設計方式推斷它。

步驟 3：建置功能集

功能是對應於文件相關性的欄位，例如 `title`、`overview`、`popularity score` (視圖數) 等等。

使用每個功能的 Mustache 範本建置一個功能集。如需有關功能的詳細資訊，請參閱使用功能。

在此範例中，我們使用 title 和 overview 欄位來建置 movie_features 功能集：

```
POST _ltr/_featureset/movie_features
{
  "featureset" : {
      "name" : "movie_features",
      "features" : [
        {
          "name" : "1",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "title" : "{{keywords}}"
            }
          }
        },
        {
          "name" : "2",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "overview" : "{{keywords}}"
            }
          }
        }
      ]
    }
}
```

如果您查詢原始 .ltrstore 索引，可以返回您的功能集：

```
GET _ltr/_featureset
```

步驟 4：記錄功能值

功能值是 BM-25 針對每個功能計算的相關性分數。

結合功能集和判斷清單來記錄功能值。如需有關記錄功能的詳細資訊，請參閱記錄功能分數。

在此範例中，bool 查詢會使用篩選器來擷取已分級的文件，然後使用 sltr 查詢選取功能集。ltr_log 查詢結合文件和功能來記錄相應的功能值：

```
POST tmdb/_search
{
  "_source": {
    "includes": [
      "title",
      "overview"
    ]
  },
  "query": {
    "bool": {
      "filter": [
        {
          "terms": {
            "_id": [
              "7555",
              "1370",
              "1369",
              "1368"
            ]
          }
        },
        {
          "sltr": {
            "_name": "logged_featureset",
            "featureset": "movie_features",
            "params": {
              "keywords": "rambo"
            }
          }
        }
      ]
    }
  },
  "ext": {
    "ltr_log": {
      "log_specs": {
        "name": "log_entry1",
        "named_query": "logged_featureset"
      }
```

```
    }
  }
}
```

範例回應看起來類似如下：

```
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4,
      "relation" : "eq"
    },
    "max_score" : 0.0,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1368",
        "_score" : 0.0,
        "_source" : {
          "overview" : "When former Green Beret John Rambo is harassed by local law
 enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
 rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
 sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1"
                },
                {
                  "name" : "2",
```

```
                "value" : 10.558305
              }
            ]
          }
        ]
      },
      "matched_queries" : [
        "logged_featureset"
      ]
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "7555",
      "_score" : 0.0,
      "_source" : {
        "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
        "title" : "Rambo"
      },
      "fields" : {
        "_ltrlog" : [
          {
            "log_entry1" : [
              {
                "name" : "1",
                "value" : 11.2569065
              },
              {
                "name" : "2",
                "value" : 9.936821
              }
            ]
          }
        ]
      },
      "matched_queries" : [
        "logged_featureset"
      ]
    },
    {
```

```
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 0.0,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1",
                  "value" : 6.334839
                },
                {
                  "name" : "2",
                  "value" : 10.558305
                }
              ]
            }
          ]
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 0.0,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
          "title" : "Rambo III"
        },
        "fields" : {
```

```
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1",
                  "value" : 9.425955
                },
                {
                  "name" : "2",
                  "value" : 11.262714
                }
              ]
            }
          ]
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      }
    ]
  }
}
```

在上一個範例中，第一個功能沒有功能值，因為關鍵字「rambo」不會出現在 ID 等於 1368 的文件的標題欄位中。這是訓練資料中缺少的功能值。

步驟 5：建立訓練資料集

> **Note**
>
> 您必須在 OpenSearch Service 之外執行此步驟。

下一步是結合判斷清單和功能值來建立訓練資料集。如果您的原始判斷清單如下所示：

```
4 qid:1 # 7555 Rambo
3 qid:1 # 1370 Rambo III
3 qid:1 # 1369 Rambo: First Blood Part II
3 qid:1 # 1368 First Blood
```

將它轉換為最終訓練資料集，如下所示：

```
4 qid:1 1:12.318474 2:10.573917 # 7555 rambo
3 qid:1 1:10.357875 2:11.950391 # 1370 rambo
3 qid:1 1:7.010513 2:11.220095 # 1369 rambo
3 qid:1 1:0.0 2:11.220095 # 1368 rambo
```

您可以手動執行此步驟或編寫程式來自動執行此步驟。

第 6 步：選擇一個演算法並建置模型

> **Note**
>
> 您必須在 OpenSearch Service 之外執行此步驟。

訓練資料集到位之後，下一步是使用 XGBoost 或 Ranklib 程式庫來建置一個模型。使用 XGBoost 和 Ranklib 程式庫可建置熱門模型，例如 LambdaMART、Random Forests 等。

有關使用 XGBoost 和 Ranklib 建置模型的步驟，請分別參閱 XGBoost 和 RankLib 文件。若要使用 Amazon SageMaker 來建置 XGBoost 模型，請參閱 XGBoost 演算法。

步驟 7：部署模型

建置模型之後，請將其部署到 Learning to Rank 外掛程式中。如需有關部署模型的詳細資訊，請參閱上傳訓練模型。

在此範例中，我們使用 Ranklib 程式庫建置 `my_ranklib_model` 模型：

```
POST _ltr/_featureset/movie_features/_createmodel?pretty
{
  "model": {
    "name": "my_ranklib_model",
    "model": {
      "type": "model/ranklib",
      "definition": """## LambdaMART
## No. of trees = 10
## No. of leaves = 10
## No. of threshold candidates = 256
## Learning rate = 0.1
## Stop early = 100

<ensemble>
   <tree id="1" weight="0.1">
```

```
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-2.0</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-2.0</output>
               </split>
               <split pos="right">
                  <output>-2.0</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>2.0</output>
         </split>
      </split>
   </tree>
   <tree id="2" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.67031991481781</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.67031991481781</output>
               </split>
               <split pos="right">
                  <output>-1.6703200340270996</output>
               </split>
```

```
            </split>
         </split>
         <split pos="right">
            <output>1.6703201532363892</output>
         </split>
      </split>
   </tree>
   <tree id="3" weight="0.1">
      <split>
         <feature>2</feature>
         <threshold>10.573917</threshold>
         <split pos="left">
            <output>1.479954481124878</output>
         </split>
         <split pos="right">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.4799546003341675</output>
               </split>
               <split pos="right">
                  <output>-1.479954481124878</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.479954481124878</output>
            </split>
         </split>
      </split>
   </tree>
   <tree id="4" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.3569872379302979</output>
            </split>
            <split pos="right">
```

```
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <output>-1.3569872379302979</output>
            </split>
            <split pos="right">
               <output>-1.3569872379302979</output>
            </split>
         </split>
      </split>
      <split pos="right">
         <output>1.3569873571395874</output>
      </split>
   </split>
</tree>
<tree id="5" weight="0.1">
   <split>
      <feature>1</feature>
      <threshold>10.357875</threshold>
      <split pos="left">
         <feature>1</feature>
         <threshold>0.0</threshold>
         <split pos="left">
            <output>-1.2721362113952637</output>
         </split>
         <split pos="right">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <output>-1.2721363306045532</output>
            </split>
            <split pos="right">
               <output>-1.2721363306045532</output>
            </split>
         </split>
      </split>
      <split pos="right">
         <output>1.2721362113952637</output>
      </split>
   </split>
</tree>
<tree id="6" weight="0.1">
   <split>
      <feature>1</feature>
```

```
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.2110036611557007</output>
               </split>
               <split pos="right">
                  <output>-1.2110036611557007</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.2110037803649902</output>
            </split>
         </split>
         <split pos="right">
            <output>1.2110037803649902</output>
         </split>
      </split>
   </tree>
   <tree id="7" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.165616512298584</output>
               </split>
               <split pos="right">
                  <output>-1.165616512298584</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.165616512298584</output>
            </split>
         </split>
```

```
         <split pos="right">
            <output>1.165616512298584</output>
         </split>
      </split>
   </tree>
   <tree id="8" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.131177544593811</output>
               </split>
               <split pos="right">
                  <output>-1.131177544593811</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.131177544593811</output>
            </split>
         </split>
         <split pos="right">
            <output>1.131177544593811</output>
         </split>
      </split>
   </tree>
   <tree id="9" weight="0.1">
      <split>
         <feature>2</feature>
         <threshold>10.573917</threshold>
         <split pos="left">
            <output>1.1046180725097656</output>
         </split>
         <split pos="right">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
```

```
               <split pos="left">
                  <output>-1.1046180725097656</output>
               </split>
               <split pos="right">
                  <output>-1.1046180725097656</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.1046180725097656</output>
            </split>
         </split>
      </split>
   </tree>
   <tree id="10" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.0838804244995117</output>
               </split>
               <split pos="right">
                  <output>-1.0838804244995117</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.0838804244995117</output>
            </split>
         </split>
         <split pos="right">
            <output>1.0838804244995117</output>
         </split>
      </split>
   </tree>
</ensemble>
"""
    }
  }
```

```
}
```

若要查看模型，請傳送下列請求：

```
GET _ltr/_model/my_ranklib_model
```

步驟 8：使用 Learning to Rank 進行搜尋

部署模型之後，您準備好進行搜尋。

透過您正在使用的功能和您想要執行的模型名稱來執行 `sltr` 查詢：

```
POST tmdb/_search
{
  "_source": {
    "includes": ["title", "overview"]
  },
  "query": {
    "multi_match": {
      "query": "rambo",
      "fields": ["title", "overview"]
    }
  },
  "rescore": {
    "query": {
      "rescore_query": {
        "sltr": {
          "params": {
            "keywords": "rambo"
          },
          "model": "my_ranklib_model"
        }
      }
    }
  }
}
```

透過 Learning to Rank，您會看到「Rambo」作為第一個結果，因為我們已將其分配在判斷清單中的最高等級：

```
{
  "took" : 12,
```

```
 "timed_out" : false,
 "_shards" : {
   "total" : 1,
   "successful" : 1,
   "skipped" : 0,
   "failed" : 0
 },
 "hits" : {
   "total" : {
     "value" : 7,
     "relation" : "eq"
   },
   "max_score" : 13.096414,
   "hits" : [
     {
       "_index" : "tmdb",
       "_type" : "movie",
       "_id" : "7555",
       "_score" : 13.096414,
       "_source" : {
         "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
         "title" : "Rambo"
       }
     },
     {
       "_index" : "tmdb",
       "_type" : "movie",
       "_id" : "1370",
       "_score" : 11.17245,
       "_source" : {
         "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
         "title" : "Rambo III"
       }
     },
     {
       "_index" : "tmdb",
       "_type" : "movie",
```

```
        "_id" : "1368",
        "_score" : 10.442155,
        "_source" : {
          "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 10.442155,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "31362",
        "_score" : 7.424202,
        "_source" : {
          "overview" : "It is 1985, and a small, tranquil Florida town is being rocked
by a wave of vicious serial murders and bank robberies. Particularly sickening to the
authorities is the gratuitous use of violence by two “Rambo” like killers who dress
themselves in military garb. Based on actual events taken from FBI files, the movie
depicts the Bureau’s efforts to track down these renegades.",
          "title" : "In the Line of Duty: The F.B.I. Murders"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "13258",
        "_score" : 6.43182,
        "_source" : {
```

```
          "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
 his family's stifling home life when he encounters Lee Carter (Will Poulter), the
 school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
 to make cinematic history by filming his own action-packed video epic. Together, these
 two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
 – and sometimes mishap-filled – cinematic adventure has begun to take on a life of its
 own!""",
          "title" : "Son of Rambow"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "61410",
        "_score" : 3.9719706,
        "_source" : {
          "overview" : "It's South Africa 1990. Two major events are about to happen:
 The release of Nelson Mandela and, more importantly, it's Spud Milton's first year
 at an elite boys only private boarding school. John Milton is a boy from an ordinary
 background who wins a scholarship to a private school in Kwazulu-Natal, South Africa.
 Surrounded by boys with nicknames like Gecko, Rambo, Rain Man and Mad Dog, Spud has
 his hands full trying to adapt to his new home. Along the way Spud takes his first
 tentative steps along the path to manhood. (The path it seems could be a rather long
 road). Spud is an only child. He is cursed with parents from well beyond the lunatic
 fringe and a senile granny. His dad is a fervent anti-communist who is paranoid that
 the family domestic worker is running a shebeen from her room at the back of the
 family home. His mom is a free spirit and a teenager's worst nightmare, whether it's
 shopping for Spud's underwear in the local supermarket",
          "title" : "Spud"
        }
      }
    ]
  }
}
```

如果您不使用 Learning to Rank 外掛程式進行搜尋，OpenSearch 會傳回不同的結果：

```
POST tmdb/_search
{
  "_source": {
    "includes": ["title", "overview"]
  },
  "query": {
```

```
    "multi_match": {
      "query": "Rambo",
      "fields": ["title", "overview"]
    }
  }
}
```


```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 5,
      "relation" : "eq"
    },
    "max_score" : 11.262714,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 11.262714,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
 find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
 his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
 when Trautman is captured.",
          "title" : "Rambo III"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "7555",
        "_score" : 11.2569065,
        "_source" : {
```

```
        "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
        "title" : "Rambo"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1368",
      "_score" : 10.558305,
      "_source" : {
        "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
        "title" : "First Blood"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1369",
      "_score" : 10.558305,
      "_source" : {
        "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
        "title" : "Rambo: First Blood Part II"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "13258",
      "_score" : 6.4600153,
      "_source" : {
        "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
his family's stifling home life when he encounters Lee Carter (Will Poulter), the
school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
to make cinematic history by filming his own action-packed video epic. Together, these
```

```
 two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
 – and sometimes mishap-filled – cinematic adventure has begun to take on a life of its
 own!""",
          "title" : "Son of Rambow"
        }
      }
    ]
  }
}
```

根據您認為的模型執行情況，調整判斷清單和功能。然後，重複步驟 2 - 8，以隨著時間改善排名結果。

Learning to Rank API

使用 Learning to Rank 操作，以程式設計方式處理功能集和模型。

建立商店

建立一個隱藏的 `.ltrstore` 索引，它會存放中繼資料資訊，例如功能集和模型。

```
PUT _ltr
```

刪除商店

刪除隱藏的 `.ltrstore` 索引並重置外掛程式。

```
DELETE _ltr
```

建立功能集

建立功能集。

```
POST _ltr/_featureset/<name_of_features>
```

刪除功能集

刪除功能集。

```
DELETE _ltr/_featureset/<name_of_feature_set>
```

取得功能集

擷取功能集。

```
GET _ltr/_featureset/<name_of_feature_set>
```

建立模型

建立模型。

```
POST _ltr/_featureset/<name_of_feature_set>/_createmodel
```

刪除模型

刪除模型。

```
DELETE _ltr/_model/<name_of_model>
```

取得模型

擷取模型。

```
GET _ltr/_model/<name_of_model>
```

取得統計資料

提供外掛程式如何運作的相關資訊。

```
GET _ltr/_stats
```

您也可以使用篩選條件來擷取單一統計資料：

```
GET _ltr/_stats/<stat>
```

此外，您可以將資訊限制在叢集中的單一節點：

```
GET _ltr/_stats/<stat>/nodes/<nodeId>

{
  "_nodes" : {
    "total" : 1,
```

```
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "873043598401:ltr-77",
  "stores" : {
    ".ltrstore" : {
      "model_count" : 1,
      "featureset_count" : 1,
      "feature_count" : 2,
      "status" : "green"
    }
  },
  "status" : "green",
  "nodes" : {
    "DjelK-_ZSfyzstO5dhGGQA" : {
      "cache" : {
        "feature" : {
          "eviction_count" : 0,
          "miss_count" : 0,
          "entry_count" : 0,
          "memory_usage_in_bytes" : 0,
          "hit_count" : 0
        },
        "featureset" : {
          "eviction_count" : 2,
          "miss_count" : 2,
          "entry_count" : 0,
          "memory_usage_in_bytes" : 0,
          "hit_count" : 0
        },
        "model" : {
          "eviction_count" : 2,
          "miss_count" : 3,
          "entry_count" : 1,
          "memory_usage_in_bytes" : 3204,
          "hit_count" : 1
        }
      },
      "request_total_count" : 6,
      "request_error_count" : 0
    }
  }
}
```

在兩個層級 (節點和叢集) 提供統計資料，如下表所指定：

節點級統計資料

欄位名稱	Description
request_total_count	排名請求的總數。
request_error_count	未成功請求的總數。
快取	所有快取 (功能、功能集、模型) 的統計資料。當使用者查詢外掛程式且模型已載入記憶體時，就會發生快取命中。
cache.eviction_count	快取移出次數。
cache.hit_count	快取命中次數。
cache.miss_count	快取遺漏次數。當使用者查詢外掛程式並且模型尚未載入到記憶體時，會發生快取遺漏。
cache.entry_count	快取中的項目數。
cache.memory_usage_in_bytes	使用的總記憶體 (以位元組為單位)。
cache.cache_capacity_reached	指示是否已達到快取限制。

叢集層級統計資料

欄位名稱	Description
存放	指出存放功能集和模型中繼資料的位置。(預設值是「.ltrstore」。否則，它的字首為「.ltrstore_」，後面為使用者提供的名稱)。
stores.status	索引狀態。
stores.feature_sets	功能集的數目。
stores.features_count	功能數目。

欄位名稱	Description
stores.model_count	模型數目。
status	以特徵存放區索引 (紅色、黃色或綠色) 和斷路器狀態 (開啟或關閉) 為基礎的外掛程式狀態。
cache.cache_capacity_reached	指示是否已達到快取限制。

取得快取統計資料

傳回快取和記憶體用量的相關統計資料。

```
GET _ltr/_cachestats

{
    "_nodes": {
        "total": 2,
        "successful": 2,
        "failed": 0
    },
    "cluster_name": "opensearch-cluster",
    "all": {
        "total": {
            "ram": 612,
            "count": 1
        },
        "features": {
            "ram": 0,
            "count": 0
        },
        "featuresets": {
            "ram": 612,
            "count": 1
        },
        "models": {
            "ram": 0,
            "count": 0
        }
    },
    "stores": {
        ".ltrstore": {
```

```
            "total": {
                "ram": 612,
                "count": 1
            },
            "features": {
                "ram": 0,
                "count": 0
            },
            "featuresets": {
                "ram": 612,
                "count": 1
            },
            "models": {
                "ram": 0,
                "count": 0
            }
        }
    },
    "nodes": {
        "ejF6uutERF20wOFNOXB61A": {
            "name": "opensearch1",
            "hostname": "172.18.0.4",
            "stats": {
                "total": {
                    "ram": 612,
                    "count": 1
                },
                "features": {
                    "ram": 0,
                    "count": 0
                },
                "featuresets": {
                    "ram": 612,
                    "count": 1
                },
                "models": {
                    "ram": 0,
                    "count": 0
                }
            }
        },
        "Z2RZNWRLSveVcz2c6lHf5A": {
            "name": "opensearch2",
            "hostname": "172.18.0.2",
```

```
            "stats": {
                ...
            }
        }
    }
}
```

清除快取

清除外掛程式快取。使用它來重新整理模型。

```
POST _ltr/_clearcache
```

Amazon OpenSearch Service 中的非同步搜尋

透過 Amazon OpenSearch Service 的非同步搜尋，您可以提交在背景中執行的搜尋查詢、監控請求進度以及在稍後階段擷取結果。在搜尋完成之前，您可以在部分結果變為可用時進行擷取。搜尋完成之後，儲存結果以供日後擷取和分析。

非同步搜尋需要 OpenSearch 1.0 或更高版本，或 Elasticsearch 7.10 或更高版本。

本文件提供非同步搜尋的簡短概觀。它還討論使用非同步搜尋搭配受管 Amazon OpenSearch Service 網域而非開放原始碼 OpenSearch 叢集的限制。如需非同步搜尋的完整文件，包括可用的設定、許可和完整的 API 參考，請參閱 OpenSearch 文件中的非同步搜尋。

搜尋呼叫範例

若要執行非同步搜尋，請使用下列格式將 HTTP 請求傳送至 `_plugins/_asynchronous_search`：

```
POST opensearch-domain/_plugins/_asynchronous_search
```

> **Note**
>
> 如果您使用的是 Elasticsearch 7.10 而不是 OpenSearch 版本，請在所有非同步搜尋請求中將 `_plugins` 替換為 `_opendistro`。

您可以指定下列非同步搜尋選項：

選項	Description	預設值	必要
wait_for_completion_timeout	指定您計劃等待結果的時間量。您可以看到在這個時間內得到的任何結果與一般搜尋一樣。您可以根據 ID 來輪詢剩餘結果。最高值為 300 秒。	1 秒	否
keep_on_completion	指定搜尋完成後是否要將結果儲存在叢集中。您可以在稍後時間檢查儲存的結果。	false	否
keep_alive	指定在叢集中儲存結果的時間長度。例如：2d 表示結果在叢集中儲存 48 小時。儲存的搜尋結果會在此時段之後或取消搜尋時刪除。請注意，這包括查詢執行時間。如果此次查詢超限，處理程序會自動取消此查詢。	12 小時	否

請求範例

```
POST _plugins/_asynchronous_search/?
pretty&size=10&wait_for_completion_timeout=1ms&keep_on_completion=true&request_cache=false
{
  "aggs": {
    "city": {
      "terms": {
        "field": "city",
        "size": 10
      }
    }
  }
}
```

> **Note**
>
> 支援適用於標準 _search 查詢的所有請求參數。如果您使用的是 Elasticsearch 7.10 而不是 OpenSearch 版本，請將 _plugins 替換為 _opendistro。

非同步搜尋許可

非同步搜尋支援精細存取控制。如需混合和匹配許可以符合您的使用案例的詳細資訊，請參閱非同步搜尋安全性。

對於已啟用精細存取控制的網域，您需要角色的下列最低許可：

```
# Allows users to use all asynchronous search functionality
asynchronous_search_full_access:
  reserved: true
  cluster_permissions:
    - 'cluster:admin/opensearch/asynchronous-search/*'
  index_permissions:
    - index_patterns:
        - '*'
      allowed_actions:
        - 'indices:data/read/search*'

# Allows users to read stored asynchronous search results
asynchronous_search_read_access:
  reserved: true
  cluster_permissions:
    - 'cluster:admin/opensearch/asynchronous-search/get'
```

對於已停用精細存取控制的網域，請使用您的 IAM 存取權和秘密金鑰來簽署所有請求。您可以使用非同步搜尋 ID 來存取結果。

非同步搜尋設定

使用 `_cluster/settings` API，OpenSearch 可讓您變更所有可用的非同步搜尋設定。在 OpenSearch Service 中，您只能變更下列設定：

- `plugins.asynchronous_search.node_concurrent_running_searches`
- `plugins.asynchronous_search.persist_search_failures`

跨叢集搜尋

您可以在具有下列次要限制的叢集間執行非同步搜尋：

- 您只能在來源網域上執行非同步搜尋。

- 您無法將網路往返次數降到最低，作為跨叢集搜尋查詢的一部分。

如果在具有連線別名 cluster_b 的 domain-a -> domain-b 與具有連線別名 cluster_c 的 domain-a -> domain-c 之間設定連線，請非同步搜尋 domain-a、domain-b 和 domain-c，如下所示：

```
POST https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_plugins/_asynchronous_search/?
pretty&size=10&wait_for_completion_timeout=500ms&keep_on_completion=true&request_cache=false
{
  "size": 0,
  "_source": {
    "excludes": []
  },
  "aggs": {
    "2": {
      "terms": {
        "field": "clientip",
        "size": 50,
        "order": {
          "_count": "desc"
        }
      }
    }
  },
  "stored_fields": [
    "*"
  ],
  "script_fields": {},
  "docvalue_fields": [
    "@timestamp"
  ],
  "query": {
    "bool": {
      "must": [
        {
          "query_string": {
            "query": "status:404",
            "analyze_wildcard": true,
            "default_field": "*"
          }
        },
```

```
        {
          "range": {
            "@timestamp": {
              "gte": 1483747200000,
              "lte": 1488326400000,
              "format": "epoch_millis"
            }
          }
        }
      ],
      "filter": [],
      "should": [],
      "must_not": []
    }
  }
}
```

回應

```
{
  "id" :
 "Fm9pYzJyVG91U19xb0hIQUJnMHJfRFEAAAAAAAAknghQ1OWVBczNZQjVEa2dMYTBXaTdEagAAAAAAAAAAB",
  "state" : "RUNNING",
  "start_time_in_millis" : 1609329314796,
  "expiration_time_in_millis" : 1609761314796
}
```

如需詳細資訊，請參閱the section called “跨叢集搜尋”。

UltraWarm

使用 UltraWarm 索引的非同步搜尋繼續運作。如需詳細資訊，請參閱the section called “UltraWarm 儲存”。

> **Note**
>
> 您可以在 CloudWatch 中監控非同步搜尋統計數字。如需指標的完整清單，請參閱 the section called “非同步搜尋指標”。

Amazon OpenSearch Service 中的時間點搜尋

時間點 (PIT) 是一種搜尋類型，可讓您針對固定時間的資料集執行不同的查詢。一般而言，當您在不同時間點對相同索引執行相同的查詢時，會收到不同的結果，因為文件會持續編製索引、更新和刪除。使用 PIT，您可以查詢資料集的恆定狀態。

PIT 搜尋的主要用途是將其與 `search_after`功能結合。這是 OpenSearch 中偏好的分頁方法，特別是深度分頁，因為它在及時凍結的資料集上運作，它不繫結至查詢，並且支援向前和向後一致的分頁。您可以將 PIT 與執行 OpenSearch 2.5 版的網域搭配使用。

> **Note**
>
> 本主題提供 PIT 的概觀，以及在受管 Amazon OpenSearch Service 網域上使用它時需要考慮的一些事項，而非自我管理的 OpenSearch 叢集。如需 PIT 的完整文件，包括完整的 API 參考，請參閱開放原始碼 OpenSearch 文件中的時間點。

考量事項

當您設定 PIT 搜尋時，請考慮下列事項：

- 如果您要從執行 OpenSearch 2.3 版的網域升級，且需要 PIT 動作的精細存取控制，則需要手動新增這些動作和角色。
- PIT 沒有彈性。節點重新啟動、節點終止、藍/綠部署和 OpenSearch 程序重新啟動會導致所有 PIT 資料遺失。
- 如果碎片在藍/綠部署期間重新定位，則只會將即時資料區段傳輸到新節點。PIT 持有的碎片區段（僅限 和與即時資料共用的碎片）會保留在舊節點上。
- PIT 搜尋目前不適用於非同步搜尋。

建立 PIT

若要執行 PIT 查詢，`_search/point_in_time`請使用下列格式將 HTTP 請求傳送至 ：

```
POST opensearch-domain/my-index/_search/point_in_time?keep_alive=time
```

您可以指定下列 PIT 選項：

選項	Description	預設值	必要
keep_alive	保留 PIT 的時間量。每次使用搜尋請求存取 PIT 時，PIT 生命週期都會延長等於 keep_alive 參數的時間量。建立 PIT 時需要此查詢參數，但在搜尋請求中為選用。		是
preference	字串，指定用於執行搜尋的節點或碎片。	隨機	否
routing	指定將搜尋請求路由到特定碎片的字串。	文件的 _id	否
expand_wildcards	指定可符合萬用字元模式之索引類型的字串。支援逗號分隔值。有效值如下： • all：比對任何索引或資料串流，包括隱藏的索引或資料串流。 • open：比對開啟、非隱藏的索引或非隱藏的資料串流。 • closed：比對已關閉、非隱藏的索引或非隱藏的資料串流。 • hidden：比對隱藏的索引或資料串流。必須結合開啟、關閉或開啟和關閉。 • none：不接受萬用字元模式。	open	否
allow_partial_pit_creation	布林值，指定是否建立具有部分失敗的 PIT。	true	否

回應範例

```
{
    "pit_id":
 "o463QQEPbXktaW5kZXgtMDAwMDAxFnNOWU43ckt3U3IyaFVpbGE1UWEtMncAFjFyeXBsRGJmVFM2RTB6eVg1aVVqQncAA
    "_shards": {
        "total": 1,
        "successful": 1,
```

```
        "skipped": 0,
        "failed": 0
    },
    "creation_time": 1658146050064
}
```

當您建立 PIT 時，您會在回應中收到 PIT ID。這是您使用 PIT 執行搜尋的 ID。

時間點許可

PIT 支援精細存取控制。如果您要升級至 OpenSearch 2.5 版網域且需要精細存取控制，則需要手動建立具有下列許可的角色：

```
# Allows users to use all point in time search search functionality
point_in_time_full_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - '*'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/point_in_time/readall"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"


# Allows users to use point in time search search functionality for specific index
# All type operations like list all PITs, delete all PITs are not supported in this
 case

point_in_time_index_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - 'my-index-1'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"
```

對於具有 OpenSearch 2.5 版及更高版本的網域，您可以使用內建`point_in_time_full_access`角色。如需詳細資訊，請參閱 OpenSearch 文件中的安全模型。

PIT 設定

OpenSearch 可讓您使用 `_cluster/settings` API 變更所有可用的 PIT 設定。在 OpenSearch Service 中，您目前無法修改設定。

跨叢集搜尋

您可以建立 PITs、使用 PIT IDs搜尋、列出 PITs，以及刪除具有下列次要限制的叢集之間的 PITs：

- 您只能列出來源網域上的所有 和刪除所有 PITs。
- 您無法將網路往返次數降到最低，作為跨叢集搜尋查詢的一部分。

如需詳細資訊，請參閱the section called “跨叢集搜尋”。

UltraWarm

使用 UltraWarm 索引的 PIT 搜尋會繼續運作。如需詳細資訊，請參閱the section called “UltraWarm 儲存”。

> **Note**
>
> 您可以在 CloudWatch 中監控 PIT 搜尋統計資料。如需指標的完整清單，請參閱 the section called “時間點指標”。

Amazon OpenSearch Service 中的代理程式搜尋

從 OpenSearch 3.3 版開始，代理程式搜尋可讓採用 AI 的程序使用自動代理程式代表您執行複雜的搜尋。

客服人員搜尋引進智慧型客服人員系統，了解使用者意圖、協調正確的工具，並產生最佳化的查詢。它透過自然語言界面提供其決策的透明摘要。

使用 OpenSearch Service，您可以為 和外部服務設定 AI 連接器 AWS 服務。使用 主控台，您也可以使用可用於建置代理程式的 CloudFormation 範本來建立 ML 模型。如需詳細資訊，請參閱使用 Bedrock Claude 設定代理搜尋。

如需完整的文件和step-by-step實作，請參閱 OpenSearch 文件中的客服人員搜尋。

Amazon OpenSearch Service 中的語意搜尋

從 2.19 版開始，您可以使用自動語意擴充，以最少的設定工作實作語意搜尋。此功能會自動為您的文字欄位產生稀疏編碼，無需手動擷取管道設定。如需詳細資訊，請參閱 OpenSearch 文件中的自動語意擴充。

從 OpenSearch 2.9 版開始，您可以使用語意搜尋來協助您了解搜尋查詢並改善搜尋相關性。您可以透過兩種方式之一使用語意搜尋：搭配神經搜尋和 k 近鄰 (k-NN) 搜尋。

使用 OpenSearch Service，您可以為 和外部服務設定 AI 連接器 AWS 服務。使用 主控台，您也可以使用 CloudFormation 範本建立 ML 模型。如需詳細資訊，請參閱the section called “CloudFormation 範本整合”。

如需語意搜尋的完整文件，包括使用語意搜尋step-by-step指南，請參閱開放原始碼 OpenSearch 文件中的語意搜尋。

Amazon OpenSearch Service 中的並行區段搜尋

從 OpenSearch 2.17 版開始，並行區段搜尋會使用新的設定來控制並行搜尋行為。

- 在 2xl 或更高版本的節點上，使用 2.17 版建立的新網域預設會將並行區段搜尋設定為自動模式。
- 升級至 2.17 的現有網域預設並行區段搜尋會根據所有 2xl 或更高節點的執行個體類型，以及過去 1 週內叢集的整體 CPU 使用率低於 45% 時，設定為自動。
- 如需詳細資訊，請參閱並行區段搜尋 2.17 版。

從 OpenSearch 2.13 版開始，您可以使用並行區段搜尋，協助您在查詢階段平行搜尋區段。如需並行區段搜尋的完整文件，請參閱開放原始碼 OpenSearch 文件中的並行區段搜尋。如需並行區段搜尋相關 Amazon CloudWatch 指標的資訊，請參閱執行個體指標和 UltraWarm 指標。

當您搭配 Amazon OpenSearch Service 使用目前的客群搜尋時，會套用一些額外的限制：

- 根據預設，OpenSearch Service 會使用具有最大配量計數機制的 2 個配量計數。

在 Amazon OpenSearch Service 中產生自然語言查詢

Amazon OpenSearch Service 中的自然語言查詢產生功能可讓您透過自然語言查詢操作和安全性日誌資料。OpenSearch 是探索日誌資料的理想選項，因為它是高度可擴展且高效能的日誌分析和搜尋引擎，現在您可以使用自然語言來探索這些日誌。此功能可讓您在不依賴 OpenSearch 管道處理語言 (PPL) 的情況下識別問題，或在建置查詢時查詢資料定義。您可以在具有 2.13 版和更新版本的 OpenSearch Service 網域上使用自然語言查詢產生功能。您必須啟用精細存取控制。

此功能是使用 OpenSearch Assistant Toolkit 建置。如果您想要建立連線至大型語言模型的類似功能，您可以使用 工具組來設定自己的客服人員和工具。

先決條件

您的網域必須具有下列項目，才能使用自然語言查詢產生功能：

- 2.13 版或更新版本。
- 服務軟體 R20240520-P4 或更高版本。
- 已啟用精細存取控制。如需詳細資訊，請參閱the section called “啟用精細存取控制”。

開始使用

在已啟用精細存取控制的 2.13 版或更新版本建立的所有網域上，預設會啟用自然語言查詢產生。

對於其他網域，請選取啟用自然語言查詢產生和 Amazon Q Developer 功能來啟用它。

啟用後，導覽至 OpenSearch Dashboards 中的日誌頁面。選擇事件總管，並向查詢助理提出問題。

設定 許可

如果您在預先存在的 OpenSearch Service 網域上啟用自然語言查詢產生，則可能不會在網域上定義 query_assistant_access 角色。非系統管理員使用者必須映射至此角色，以便在使用精細存取控制的網域上管理暖索引。若要手動建立 query_assistant_access 角色，請執行下列步驟：

1. 在 OpenSearch Dashboards 中，前往安全性並選擇角色。
2. 選擇建立角色並設定下列叢集許可：
    - `cluster:admin/opensearch/ml/config/get`
    - `cluster:admin/opensearch/ml/execute`

- `cluster:admin/opensearch/ml/predict`
- `cluster:admin/opensearch/ppl`

3. 命名角色 query_assistant_access。
4. 選擇建立角色。query_assistant_access 角色現已可用。

> **Note**
>
> 您還必須擁有要使用自然語言問題之索引的 `indices:admin/mappings/get`和 `read` 索引許可。

組態自動化

Flow Framework 是一種 OpenSearch 外掛程式，提供自動化 OpenSearch 組態的方法，適用於查詢產生和對話聊天等使用案例。由於外掛程式會追蹤啟用自然語言查詢產生功能的資源，因此流程架構索引會為每個使用查詢協助的網域存放範本。

流程架構可讓您從一組預先定義的範本中選取，或為機器學習連接器、工具、代理程式和其他元件建立自己的自動化，以準備 OpenSearch 做為生成模型的後端。

向量搜尋

Amazon OpenSearch Service 中的向量搜尋可讓您使用機器學習內嵌而非傳統關鍵字比對來搜尋語意相似的內容。向量搜尋會將您的資料（文字、影像、音訊等）轉換為擷取內容語意意義的高維度數值向量（內嵌）。當您執行搜尋時，OpenSearch 會將查詢的向量表示法與儲存的向量進行比較，以尋找最相似的項目。

向量搜尋包含下列重要元件。

向量欄位

OpenSearch 支援 `knn_vector` 欄位類型，以存放具有可設定維度的密集向量（最多 16，000 個）。

搜尋方法

- k-NN (k 近鄰）：尋找 k 最相似的向量
- 大約 k-NN：使用 HNSW (Hierarchical Navigable Small World) 等演算法，在大型資料集上更快速地搜尋

距離指標

支援各種相似性計算，包括：

- 歐幾里德距離
- 餘弦相似度
- 點產品

常用案例

向量搜尋支援下列常見使用案例。

- 語意搜尋：尋找具有類似意義的文件，而不只是比對關鍵字
- 建議系統：建議類似的產品、內容或使用者
- 影像搜尋：尋找視覺效果相似的影像
- 異常偵測：識別資料模式中的極端值
- RAG（擷取增強生成）：使用相關內容增強 LLM 回應

與機器學習整合

OpenSearch 與下列機器學習服務和模型整合：

- Amazon Bedrock：用於使用基礎模型產生內嵌
- Amazon SageMaker AI：適用於自訂 ML 模型部署
- Hugging Face 模型：預先訓練的內嵌模型
- 自訂模型：您自己的訓練內嵌模型

透過向量搜尋，您可以建置複雜的 AI 支援應用程式，了解內容和意義，遠遠超過傳統的文字比對功能。

從 Amazon S3 向量匯入 OpenSearch Serverless

Amazon S3 Vectors 提供第一個具有原生支援的雲端物件存放區，以存放和查詢向量。S3 Vectors 提供經濟實惠、彈性且耐用的向量儲存，可根據語意意義和相似性進行查詢。它提供一秒的查詢回應時間，並降低上傳、儲存和查詢向量的成本高達 90%。

Amazon S3 Vectors 引入 S3 向量儲存貯體，您可以用來儲存、存取和查詢向量資料，而無需佈建任何基礎設施。在向量儲存貯體中，您可以在向量索引中組織向量資料。您的向量儲存貯體可以有多個向量索引，每個向量索引可以容納數百萬個向量。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的使用 Amazon S3 向量和向量儲存貯體。 Amazon S3

每個向量包含：

- 唯一金鑰
- 向量資料
- JSON 格式的選用中繼資料

向量索引支援歐幾里得和餘弦距離函數，以進行相似性搜尋操作。

> **Note**
>
> 向量儲存貯體的主要優點是能夠以極低的成本存放大量資料集，同時提供向量操作的直接 API 存取。

如需 Amazon S3 向量儲存貯體的詳細資訊，包括如何建立一個，請參閱《Amazon S3 使用者指南》中的使用 Amazon S3 向量和向量儲存貯體。 Amazon S3 如需本主題所述範圍之外與 OpenSearch Service 整合的詳細資訊，請參閱搭配使用 S3 向量與 OpenSearch Service

您可以將 S3 Vectors 與 Amazon OpenSearch Service 搭配使用，以降低查詢頻率較低的向量儲存成本，然後隨著需求增加或增強搜尋功能，將這些資料集快速移至 OpenSearch。

OpenSearch Service 與 Amazon S3 Vectors 整合，提供超越 Amazon S3 向量儲存貯體本身的增強效能和功能。當您需要時，請考慮此整合：

- 較高的查詢輸送量
- 低於秒的搜尋延遲
- 進階分析功能，例如彙總
- 結合文字和向量資料的混合式搜尋

當多個應用程式使用具有不同效能需求的相同向量資料時，此整合特別有用。您可以讓某些應用程式直接與 Amazon S3 向量儲存貯體互動，處理成本敏感的使用案例，而其他應用程式則利用 OpenSearch 整合進行效能關鍵的操作。

整合架構

整合使用 Amazon OpenSearch Ingestion (OSI) 作為 Amazon S3 向量索引和 Amazon OpenSearch Serverless 向量集合之間的資料管道。OpenSearch Ingestion 會自動從指定的向量索引匯出向量資料，並將其擷取至 OpenSearch Serverless 向量集合，以進行高效能搜尋操作。

> **Note**
>
> 匯出之後，您的資料仍然存在於 S3 向量索引中。您有兩份資料副本。

每個向量索引都會對應至 OpenSearch Serverless 集合中的對應索引。整合：

- 保留向量維度
- 保留中繼資料
- 最佳化 OpenSearch 向量搜尋功能的資料結構

組態之後，OpenSearch Ingestion 會使用 Amazon S3 ListVectors API 從指定的向量索引取用向量，以開始資料匯出程序。服務會平行處理向量，以最佳化擷取速度，同時遵守 OpenSearch Ingestion 和 Amazon OpenSearch Serverless 的擴展限制。

在擷取期間，服務會：

- 轉換向量資料以符合 OpenSearch Service 的預期格式
- 保留重要資訊，包括向量值、中繼資料和距離指標
- 透過智慧型重試機制處理失敗案例
- 將有問題的記錄放在用作無效字母佇列的 Amazon S3 儲存貯體中，以供日後分析

整合可有效處理大量資料集，效能取決於向量維度、資料集大小和設定的擴展限制。OSI 每個管道最多可擴展 16 個工作者，而 OpenSearch Serverless 會根據擷取需求自動調整容量。根據預設，OpenSearch 會將 `maxSearch` OpenSearch Serverless 端的 OpenSearch 運算單位 (OCU) 增加到 100。

> **Note**
>
> 整合透過以下方式排定成本效益的優先順序：
>
> - 匯出完成後自動關閉管道
> - OpenSearch Serverless 集合擴展
> - Pay-per-use資源模型

所需的 IAM 許可

整合需要仔細設定 IAM 許可，才能在服務之間進行安全通訊。OpenSearch Ingestion 需要許可，才能從 Amazon S3 向量索引讀取、寫入 OpenSearch Service 向量集合，以及管理相關聯的安全政策。

當您使用本主題稍後的程序啟用整合時，您可以選擇下列其中一個選項進行許可管理：

- 允許系統自動建立具有必要許可的服務角色
- 提供符合要求的現有角色

自動建立的角色包含下列政策：

- 存取 Amazon S3 向量索引 APIs
- 管理 OpenSearch Service 集合操作
- 處理失敗擷取嘗試的無效字母佇列操作

如果您選擇指定現有角色，請確認該角色具有下列 IAM 許可：

（必要）：OpenSearch Ingestion 和 OpenSearch Serverless 之間的資料管道許可

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "allowAPIs",
            "Effect": "Allow",
            "Action": [ "aoss:APIAccessAll", "aoss:BatchGetCollection" ],
            "Resource": [ "arn:aws:aoss:*:111122223333:collection/collection-
id" ]
        },
        {
            "Sid": "allowSecurityPolicy",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:UpdateSecurityPolicy",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*",
            "Condition":{
               "StringLike":{
                  "aoss:collection": [ "collection-name" ]
                },
               "StringEquals": {
                  "aws:ResourceAccount": [ "111122223333" ]
               }
            }
        }
    ]
}
```

（必要）：OpenSearch Ingestion 和 Amazon S3 無效字母佇列之間的資料擷取許可

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "s3Access",
            "Effect": "Allow",
            "Action": [
              "s3:PutObject"
            ],
            "Resource": [ "arn:aws:s3:::bucket/*" ]
        }
    ]
}
```

（必要）：OpenSearch Ingestion 和 Amazon S3 Vectors 之間的資料擷取許可

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3VectorIndexAccess",
            "Effect": "Allow",
            "Action": [
               "s3vectors:ListVectors",
               "s3vectors:GetVectors"
            ],
            "Resource": [
                "arn:aws:s3vectors:us-east-1:111122223333:bucket/bucket-name/
index/index-name"
            ]
        }
    ]
}
```

（啟用 AWS KMS 加密時必填）：OpenSearch Ingestion 和 Amazon S3 Vectors 之間通訊的解密許可 JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "allowS3VectorDecryptionOfCustomManagedKey",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [ "arn:aws:kms:us-east-1:111122223333:key/key-id" ],
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3vectors.us-east-1.amazonaws.com",
                    "kms:EncryptionContext:aws:s3vectors:arn": [
                        "arn:aws:s3vectors:us-east-1:111122223333:bucket/example-
bucket",
                        "arn:aws:s3vectors:us-east-1:111122223333:bucket/example-
bucket/index/example-index"
                        ]
                 }
             }
        }
    ]
}
```

設定 Amazon S3 Vectors 與 OpenSearch 的整合

使用下列程序來設定 Amazon S3 Vectors 與 OpenSearch Serverless 的整合。

> **Note**
>
> 如果您在 Vector 儲存貯體頁面中選擇匯出至 OpenSearch 選項，開始從 Amazon S3 主控台設定整合的程序，則下列程序中的某些步驟不適用，如程序中所述。

設定 Amazon S3 Vectors 與 OpenSearch Serverless 的整合

1. 在 Amazon OpenSearch Service 主控台中開啟將 S3 向量索引匯入 OpenSearch 向量引擎頁面。OpenSearch 如果您在 Amazon S3 主控台中按一下匯出至 OpenSearch，頁面會自動顯示。如果您是從 OpenSearch 主控台開始，請選擇左側導覽中的整合，然後選擇匯入 S3 向量索引。
2. 在來源區段中，如果您在 Amazon S3 主控台中啟動 ，請確認已指定向量索引的名稱及其 Amazon Resource Name (ARN)。如果您在 OpenSearch 主控台中啟動 ，請在 S3 向量索引 ARN 欄位中輸入索引 ARN。
3. 在服務存取區段中，選擇選項。如果您選擇現有的角色，請確認它具有整合所需的所有必要許可，如中所述所需的 IAM 許可。
4. (選用) 展開 Additional settings (其他設定)。針對啟用備援 （主動式複本），我們建議針對生產環境保留此選項。建立第一個集合時，OpenSearch Serverless 會執行個體化兩個 OCU，一個用於索引編製，另一個用於搜尋。為確保高可用性，它也會啟動另一個可用區域中的待命節點集。為了開發和測試目的，您可以停用集合的啟用備援設定，這會消除兩個待命複本，並且只會執行個體化兩個 OCUs。根據預設，會啟用備援作用中複本，這表示帳戶的第一個集合總共會執行個體化四個 OCUs。

   針對新增 Amazon OpenSearch Serverless 向量的客戶受管 AWS KMS 金鑰，選擇此選項以使用客戶受管金鑰加密向量集合中的資料。根據預設，OpenSearch 會使用 AWS 受管金鑰。
5. 如果您在 Amazon S3 主控台中按一下匯出至 OpenSearch 選項來啟動此程序，匯出詳細資訊區段會列出 OpenSearch 接下來將採取的步驟。當您準備好時，請選擇匯出。

   如果您在 OpenSearch Service 主控台中啟動此程序，匯入詳細資訊區段會列出 OpenSearch 接下來將採取的步驟。當您準備好時，請選擇匯入。

   OpenSearch 會開啟歷史記錄頁面，以顯示 Amazon S3 向量索引的所有匯出/匯入至 OpenSearch Serverless 索引。

成功擷取後，OSI 會自動停止管道，以防止不必要的成本，同時在 OpenSearch 中維護匯出的資料。您可以透過 CloudWatch 指標監控整合進度，並存取詳細的日誌以進行故障診斷。

OpenSearch 集合會保持作用中狀態，並在初始擷取完成後可供查詢使用。您可以執行：

- 相似性搜尋
- 彙總
- 分析操作

使用 Amazon S3 向量引擎的進階搜尋功能

Amazon OpenSearch Service 提供使用 Amazon S3 作為向量索引向量引擎的功能。此功能可讓您將向量資料卸載至 Amazon S3，同時以低成本維持低於一秒的向量搜尋功能。

透過此功能，OpenSearch 會將向量內嵌存放在 Amazon S3 向量索引中，同時將其他文件欄位保留在 OpenSearch 叢集的儲存中。此架構提供下列優點：

- 耐用性：寫入 S3 Vectors 的資料存放在 S3 上，專為 11 9 秒的資料耐用性而設計。
- 可擴展性：將大型向量資料集卸載至 S3，而不耗用叢集儲存。
- 成本效益：最佳化大量向量工作負載的儲存成本。

OpenSearch 對於使用 S3 向量索引有下列需求：

- OpenSearch 2.19 版或更新版本
- OpenSearch Optimized 執行個體
- OpenSearch 版本的最新修補程式版本

啟用 S3 向量

建立新網域或更新現有網域時，您可以在進階功能區段中選擇啟用 S3 向量作為引擎選項。當您利用 S3 向量做為引擎時，此設定可讓 OpenSearch 建立 S3 向量儲存貯體。當您啟用此選項時，OpenSearch 會透過下列方式為您的網域設定 S3 向量：

1. 在以您的網域設定的 AWS KMS 金鑰上建立新的兩個授權：
   - 授予具有解密權限的 S3 Vectors 背景索引任務
   - 授予 OpenSearch 以建立具有`GenerateDataKey`許可的 S3 向量儲存貯體
2. 將 OpenSearch 網域使用的 KMS 金鑰設定為所有向量索引資料其餘部分的加密 CMK。

使用 S3 向量引擎建立索引

設定網域之後，您可以使用 `s3vector`做為索引映射中的後端向量引擎，以 欄位建立一或多個 k-NN 索引。您可以根據您的使用案例，使用不同的引擎類型來設定不同的向量欄位。

> **Important**
>
> 您只能在建立索引期間使用 s3vector 引擎來映射欄位定義。建立索引後，您無法使用s3vector引擎新增或更新映射。

以下是一些建立 S3 向量引擎索引的範例。

範例：使用 S3 向量引擎建立 k-NN 索引

```
PUT my-first-s3vector-index
{
  "settings": {
    "index": {
      "knn": true
    }
  },
  "mappings": {
    "properties": {
        "my_vector_1": {
          "type": "knn_vector",
          "dimension": 2,
          "space_type": "l2",
          "method": {
            "engine": "s3vector"
          }
        },
        "price": {
          "type": "float"
        }
    }
  }
}
```

範例：使用 S3 向量和 FAISS 引擎建立 k-NN 索引

此範例強調您可以在同一索引內使用多個向量引擎的事實。

```
PUT my-vector-index
{
  "settings": {
    "index": {
```

```
      "knn": true
    }
  },
  "mappings": {
    "properties": {
        "my_vector_1": {
          "type": "knn_vector",
          "dimension": 2,
          "space_type": "l2",
          "method": {
            "engine": "s3vector"
          }
        },
        "price": {
          "type": "float"
        },
        "my_vector_2": {
            "type": "knn_vector",
            "dimension": 2,
            "space_type": "cosine",
            "method": {
                "name": "hnsw",
                "engine": "faiss",
                "parameters": {
                    "ef_construction": 128,
                    "m": 24
                }
            }
        }
    }
  }
}
```

不支援的範例：建立索引後新增 S3 向量引擎

不支援下列方法，且 將會失敗。

```
PUT my-first-s3vector-index
{
  "settings": {
    "index": {
      "knn": true
    }
  }
```

```
}

PUT my-first-s3vector-index/_mapping
{
  "properties": {
        "my_vector_1": {
          "type": "knn_vector",
          "dimension": 2,
          "space_type": "l2",
          "method": {
            "engine": "s3vector"
          }
        },
        "price": {
          "type": "float"
        }
    }
}
```

功能限制

在索引中使用s3vector引擎之前，請考慮下列限制：

s3vector 引擎不支援的功能和行為

功能	Behavior (行為)
Split/Shrink/Clone索引	這些 APIs與 knn_vector 欄位中以s3vector引擎設定的索引搭配使用時失敗。
快照	使用 s3vector 引擎的索引不支援快照。對於受管網域： • 自動化快照僅包含未使用s3vector引擎的索引。 • s3vector 索引的手動快照請求失敗。 **Note** 雖然快照不支援point-in-time復原，但s3vector引擎以及 OpenSearch Optimized 執行個體可提供 11 九個持久性。

功能	Behavior (行為)
UltraWarm 層	使用s3vector引擎設定的索引無法遷移至 UltraWarm 層。
跨叢集複寫	使用s3vector引擎設定的索引不支援跨叢集複寫。
意外刪除保護	由於使用 s3vector引擎的索引不支援快照，因此意外刪除保護無法使用。您仍然可以還原網域中的其他索引。
放射搜尋	使用s3vector引擎的欄位不支援具有放射搜尋的查詢。

索引文件

使用 S3 向量引擎建立索引後，您可以使用標準 _bulk API 擷取文件。OpenSearch 會自動使用s3vector引擎將knn_vector欄位的向量資料即時卸載至 S3 向量索引。OpenSearch 會在自己的儲存層中保留屬於其他欄位或使用不同引擎knn_vector的欄位的資料。

對於所有已確認的大量請求，OpenSearch 保證所有資料（向量和非向量）都耐用。如果請求收到負面確認，則無法保證該大量請求中文件的耐久性。建議您在刪除先前失敗的請求後，使用文件 ID 重試此類請求，以避免在這些極少數情況下重複的文件。

大量索引的範例

```
POST _bulk
{ "index": { "_index": "my-first-s3vector-index", "_id": "1" } }
{ "my_vector_1": [1.5, 2.5], "price": 12.2 }
{ "index": { "_index": "my-first-s3vector-index", "_id": "2" } }
{ "my_vector_1": [2.5, 3.5], "price": 7.1 }
{ "index": { "_index": "my-first-s3vector-index", "_id": "3" } }
{ "my_vector_1": [3.5, 4.5], "price": 12.9 }
{ "index": { "_index": "my-first-s3vector-index", "_id": "4" } }
{ "my_vector_1": [5.5, 6.5], "price": 1.2 }
{ "index": { "_index": "my-first-s3vector-index", "_id": "5" } }
{ "my_vector_1": [4.5, 5.5], "price": 3.7 }
```

搜尋文件

您可以使用標準 _search API 來搜尋索引，以執行文字、k-NN 或混合式查詢。對於使用s3vector引擎設定knn_vector之欄位的查詢，OpenSearch 會自動將查詢卸載至對應的 S3 向量索引。

> **Note**
>
> 使用 s3vector 引擎時，k-NN 搜尋查詢支援k最大值 100。這表示搜尋結果中最多可傳回 100 個最近的鄰。

搜尋查詢範例

```
GET my-first-s3vector-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector_1": {
        "vector": [2.5, 3.5],
        "k": 2
      }
    }
  }
}
```

您可以使用 s3vector 引擎在 OpenSearch kNN 索引上執行篩選過的向量搜尋。OpenSearch 會將篩選條件套用為後置篩選條件，並使用根據特定啟發式的超取樣機制來平衡召回與延遲。

具有篩選條件的搜尋查詢範例：

```
GET my-index/_search
{
  "size": 10,
  "query": {
    "knn": {
      "my_vector_field": {
        "vector": [2.5, 3.5, 1.2, 4.8],
        "k": 10,
        "filter": {
          "range": {
            "price": {
              "gte": 10,
              "lte": 100
            }
          }
        }
```

```
      }
    }
  }
}
```

支援的映射參數

使用s3vector引擎時， knn_vector 欄位支援映射中的下列參數。

向量欄位參數

參數	必要	Description	支援的值
type	是	文件中存在的欄位類型。	knn_vector
dimension	是	將擷取至索引的每個向量維度。	>0、<=4096
space_type	否	用於計算向量之間距離的向量空間。	l2, cosinesimil
method.engine	是	用於編製索引和搜尋的大約 k-NN 引擎。	s3vector
method.name	否	最近的鄰近方法	""
store	N/A	啟用或停用此映射參數是無操作的，因為 knn_vector 資料不會儲存在 OpenSearch 中。	不支援
doc_values	N/A	啟用或停用此映射參數是無操作的，因為 knn_vector 資料不會儲存在 OpenSearch 中。	不支援

Important

使用s3vector引擎不支援巢狀knn_vector欄位類型

計量和計費

如需此功能計量和計費的相關資訊，請參閱 Amazon OpenSearch Service 定價。

停用 s3vector 引擎

停用s3vector引擎之前，請刪除目前正在使用的所有索引。如果沒有，任何停用引擎的嘗試都會失敗。

另請注意，啟用或停用s3vector引擎會在您的網域上觸發藍/綠部署。

若要停用s3vector引擎，請編輯您的網域組態並設定 S3VectorsEngine.Enabled: false。

Amazon OpenSearch Service 中的 k 近鄰 (k-NN) 搜尋

這是其相關聯的 k 近鄰演算法的簡稱，Amazon OpenSearch Service 的 KNN 可讓您搜尋向量空間中的點，並通過歐幾里德距離或餘弦相似度找出這些點的「近鄰」。使用案例包括建議 (例如，音樂應用程式中「其他您可能喜歡的歌曲」功能)、影像辨識和詐騙偵測。

> **Note**
>
> 本文件提供 k-NN 外掛程式的簡短概觀，以及將外掛程式與受管 OpenSearch Service 搭配使用時的限制。如需 k-NN 外掛程式的完整文件，包括簡單和複雜的範例、參數參考和完整的 API 參考，請參閱開放原始碼 OpenSearch 文件。開放原始碼文件也涵蓋效能調校和 k-NN-specific叢集設定。

k-NN 入門

若要使用 k-NN，您必須使用 index.knn 設定建立索引，並新增資料類型為 knn_vector 的一個或多個欄位。

```
PUT my-index

{
  "settings": {
    "index.knn": true
  },
  "mappings": {
    "properties": {
```

```
      "my_vector1": {
        "type": "knn_vector",
        "dimension": 2
      },
      "my_vector2": {
        "type": "knn_vector",
        "dimension": 4
      }
    }
  }
}
```

knn_vector 資料類型支援最多 10,000 個浮點數的單一清單，其中包含由所需 dimension 參數定義的浮點數目。建立索引之後，將一些資料新增至其中。

```
POST _bulk

{ "index": { "_index": "my-index", "_id": "1" } }
{ "my_vector1": [1.5, 2.5], "price": 12.2 }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "my_vector1": [2.5, 3.5], "price": 7.1 }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "my_vector1": [3.5, 4.5], "price": 12.9 }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "my_vector1": [5.5, 6.5], "price": 1.2 }
{ "index": { "_index": "my-index", "_id": "5" } }
{ "my_vector1": [4.5, 5.5], "price": 3.7 }
{ "index": { "_index": "my-index", "_id": "6" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 10.3 }
{ "index": { "_index": "my-index", "_id": "7" } }
{ "my_vector2": [2.5, 3.5, 5.6, 6.7], "price": 5.5 }
{ "index": { "_index": "my-index", "_id": "8" } }
{ "my_vector2": [4.5, 5.5, 6.7, 3.7], "price": 4.4 }
{ "index": { "_index": "my-index", "_id": "9" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 8.9 }
```

然後，您可以使用 knn 查詢類型搜尋資料。

```
GET my-index/_search
{
  "size": 2,
  "query": {
    "knn": {
```

```
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  }
}
```

在此情況下，k 是您想要查詢傳回的近鄰數目，但您亦須包含 size 選項。否則，您會獲得每個碎片 (和每個區段) 的 k 結果，而不是整個查詢的 k 結果。KNN 支援的最大 k 值為 10,000。

如果您將 knn 查詢與其他子句混合使用，則可能會收到少於 k 個的結果。在此範例中，post_filter 子句會將結果的數目從 2 減少為 1。

```
GET my-index/_search

{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  },
  "post_filter": {
    "range": {
      "price": {
        "gte": 6,
        "lte": 10
      }
    }
  }
}
```

如果您需要在維持最佳效能的同時處理大量查詢，您可以使用 _msearch API 建構 JSON 的大量搜尋，並傳送單一請求來執行多個搜尋：

```
GET _msearch

{ "index": "my-index"}
```

```
{ "query": { "knn": {"my_vector2":{"vector": [2, 3, 5, 6],"k":2 }} } }
{ "index": "my-index", "search_type": "dfs_query_then_fetch"}
{ "query": { "knn": {"my_vector1":{"vector": [2, 3],"k":2 }} } }
```

下列影片示範如何設定大量向量搜尋 K-NN 查詢。

k-NN 差異、調校和限制

OpenSearch 可讓您使用 `_cluster/settings` API 來修改所有 k-NN 設定。在 OpenSearch Service 上，您可以變更除了 `knn.memory.circuit_breaker.enabled` 和 `knn.circuit_breaker.triggered` 之外的所有設定。包含 k-NN 統計數字作為 Amazon CloudWatch 指標。

尤其應依據 `knn.memory.circuit_breaker.limit` 統計數字和執行個體類型的可用 RAM，來檢查每個資料節點的 `KNNGraphMemoryUsage` 指標。OpenSearch Service 針對 Java 堆積使用執行個體 RAM 的一半 (堆積大小最多可達 32 GiB)。根據預設，k-NN 最高會使用剩下一半的 50%，這樣具有 32 GiB 的 RAM 的執行個體類型就能容納 8 GiB 的圖形 (32 * 0.5 * 0.5)。如果圖形記憶體用量超過此值，效能可能會受到影響。

您可以將 2.x 版或更新版本上建立的 k-NN 索引遷移至 2.17 版或更新版本網域上的 UltraWarm 或冷儲存。

清除快取 API 和 k-NN 索引的暖索引會封鎖。當索引啟動第一個查詢時，它會從 Amazon S3 下載圖形檔案，並將圖形載入記憶體。同樣地，當圖形的 TTL 過期時，檔案會自動從記憶體移出。

向量擷取

向量擷取可協助您快速擷取和索引 OpenSearch 網域和 OpenSearch Serverless 集合。此服務會檢查您的網域或集合，並代表您建立擷取管道，將您的資料載入 OpenSearch。您的網域或集合的擷取和索引是由 Vector 擷取為您管理。

您可以透過啟用 和 自動最佳化功能來加速向量索引的 GPU 加速和最佳化索引程序。使用向量擷取，您不需要管理基礎基礎設施、修補程式軟體或擴展叢集，以支援向量資料庫索引和擷取。這可讓您快速建置向量資料庫，以符合您的需求。

運作方式

向量擷取會檢查您的網域或集合及其索引。您可以手動設定向量索引欄位，或允許 OpenSearch 使用自動組態。

向量擷取使用 OpenSearch Ingestion (OSI) 作為 Amazon S3 和 OpenSearch 之間的資料管道。服務會平行處理向量，以最佳化擷取速度，同時遵守 OSI 和 OpenSearch 的擴展限制。

OpenSearch Vector 擷取定價

在任何特定時間，您只需支付配置給管道的向量擷取 OCUs 數量，無論是否有資料流經管道。OpenSearch 向量擷取會根據用量向上或向下擴展管道容量，立即容納您的工作負載。

如需完整定價詳細資訊，請參閱 Amazon OpenSearch Service 定價。

先決條件

使用向量擷取之前，請確定您有下列資源：

- Amazon S3 儲存貯體，其中包含 Parquet 或 JSONL 格式的 OpenSearch JSON 文件
- OpenSearch 資源 - 網域或集合
- OpenSearch 版本 2.19 或更新版本 （ 自動最佳化整合時需要 ）

建立向量資料庫

使用向量擷取任務建立工作流程來設定自動向量索引調校，並加速大規模索引建置。

> **Note**
>
> 隨著使用者介面定案，本節中的程序內容可能會有所變更。工作流程可能會在未來的版本中更新，以反映最新的主控台體驗。

建立向量擷取任務

1. 在向量擷取任務詳細資訊區段中，針對名稱輸入擷取任務的名稱。
2. 在資料來源區段中，設定下列項目：
    a. 針對 Amazon S3 URI，輸入包含 OpenSearch Service JSON 文件的 Amazon S3 儲存貯體位置。
    b. 選擇瀏覽 Amazon S3 從可用的儲存貯體中選取，或選擇檢視以預覽儲存貯體內容。
    c. 針對內容類型，選取下列其中一項：
        - 向量 - 文件已包含向量，且不需要進一步產生向量內嵌。

- 文字、影像或音訊 - 文件包含需要編碼為向量內嵌的內容，例如文字、影像或音訊位元組。

3. 在資料來源許可區段中，設定存取許可：

   a. 針對 IAM 角色，選擇下列其中一項：

      - Create a new role (建立新角色)
      - 使用現有角色

   b. 針對 IAM 角色名稱，輸入角色的名稱。

4. 在目的地區段中，設定 OpenSearch Service 端點：

   a. 針對端點，選擇從目前區域中的相容網域或集合中選擇選項。

   b. 選擇下一步以繼續選取的端點。

5. 選擇下一步以繼續下一個步驟，或選擇取消以退出而不儲存。

相關功能

向量擷取可與下列 Amazon OpenSearch Service 功能搭配使用，以最佳化向量資料庫效能：

向量索引的 GPU 加速

GPU 加速可減少建立、更新和刪除向量索引所需的時間。與向量擷取搭配使用時，您可以大幅加速大規模向量資料庫的擷取和索引程序。

自動最佳化

自動最佳化會自動探索搜尋延遲、品質和記憶體需求之間的最佳權衡。向量擷取可以在擷取程序期間套用自動最佳化建議，以確保您的向量索引已進行最佳設定。

為了獲得最佳結果，請考慮在使用向量擷取建置大規模向量資料庫時啟用 GPU 加速和自動最佳化。

將 Amazon S3 向量索引匯出至 OpenSearch Service 向量引擎

您所選 Amazon S3 向量索引的point-in-time匯出至 OpenSearch Service。OpenSearch Service 向量引擎提供簡單且可擴展的向量存放區，具有進階搜尋功能。

將 Amazon S3 向量索引匯出至 OpenSearch Service 向量引擎

1. 在來源區段中，驗證 Amazon S3 向量索引詳細資訊：

- Amazon S3 向量索引 - 來源索引的名稱
- Amazon S3 向量索引 ARN - 索引的 Amazon Resource Name

2. 在服務存取區段中，設定 OpenSearch Service 授權：

   a. 針對選擇授權 OpenSearch Service 的方法，選取下列其中一項：

      - 建立和使用新的服務角色
      - 使用現有的服務角色

   b. 在服務角色名稱中，輸入服務角色的名稱。

      > **Note**
      >
      > 服務角色名稱必須為 1 到 64 個字元。有效字元為 a-z、A-Z、0-9 和句點 (.)。

   c. 選擇檢視許可詳細資訊以檢閱所需的許可。

3. 展開其他設定 - 視需要選擇性設定進階選項。
4. 在匯出詳細資訊區段中，設定下列選項：

   - 自動化 OpenSearch Service 向量收集建立 - OpenSearch Service 收集用於儲存向量資料。無伺服器運算容量是以 OpenSearch Service Compute Units (OCUs) 測量，根據預設，最大 OCU 容量為 50。
   - 自動化服務存取的 IAM 角色建立 - OpenSearch Service 使用此角色來讀取 Amazon S3 向量索引並寫入 OpenSearch Service 集合。
   - 自動化 OpenSearch Service 擷取管道建立 - OpenSearch Service 擷取管道用於擷取資料。建立 Amazon S3 儲存貯體做為最佳實務，以擷取故障事件並將其存放在 Amazon S3 儲存貯體無效字母佇列 (DLQ)，以便輕鬆存取以進行故障診斷和分析。

5. 選擇匯出以開始匯出程序，或選擇取消以退出而不匯出。

將 Amazon S3 向量命名空間匯入 OpenSearch Service 向量引擎

使用 OpenSearch Service 分析向量資料需要一次性 OpenSearch Service 集合和 IAM 許可設定。

將 Amazon S3 向量命名空間匯入 OpenSearch Service 向量引擎

1. 在來源區段中，設定 Amazon S3 向量索引：

- 針對 Amazon S3 向量索引 ARN，輸入 Amazon S3 向量索引的 ARN。

> **Note**
>
> 必須為 arn：aws：iam：：account-id：vector-bucket-name/*：index 格式

2. 在服務存取區段中，設定 OpenSearch Service 授權：

   a. 針對選擇授權 OpenSearch Service 的方法，選取下列其中一項：

      - 建立和使用新的服務角色
      - 使用現有的服務角色

   b. 針對服務角色名稱，輸入服務角色的名稱。

> **Note**
>
> 服務角色名稱必須為 1 到 64 個字元。有效字元為 a-z、A-Z、0-9 和句點 (.)。

   c. 選擇檢視許可詳細資訊以檢閱所需的許可。

3. 展開其他設定 - 視需要選擇性設定進階選項。
4. 在匯入步驟區段中，設定下列自動化選項：

   - 自動化 OpenSearch Service 向量收集建立 - OpenSearch Service 收集用於儲存向量資料。無伺服器運算容量是以 OpenSearch Service Compute Units (OCUs) 測量，預設情況下，最大 OCU 容量為 50。
   - 自動化服務存取的 IAM 角色建立 - OpenSearch Service 使用此角色來讀取 Amazon S3 向量索引並寫入 OpenSearch Service 集合。
   - 自動化 OpenSearch Service 擷取管道建立 - OpenSearch Service 擷取管道用於擷取資料。建立 Amazon S3 儲存貯體做為最佳實務，以擷取故障事件並將其儲存在 Amazon S3 儲存貯體無效字母佇列 (DLQ) 中，以便輕鬆存取以進行故障診斷和分析。

5. 選擇匯入以開始匯入程序，或選擇取消以退出而不匯入。

檢視向量擷取任務和匯入歷史記錄

向量擷取任務會建立管道，用於對資料集進行向量化、自動化向量索引調校，以及加速大規模索引建置。

檢視向量擷取任務

1. 在向量擷取任務區段中，檢視摘要資訊：
   - 任務 - 擷取任務的總數
   - 選擇建立向量資料庫以建立新的擷取任務
2. 在 Amazon S3 向量匯入區段中，檢視匯入摘要：
   - 匯入總數 - 已完成的匯入數目
   - 選擇匯入 Amazon S3 向量以開始新的匯入
3. 在向量擷取任務表中，使用下列資訊監控作用中任務：
   - 名稱 - 任務名稱
   - 狀態 - 目前任務狀態（例如作用中）
   - 資料來源 - 來源位置（例如 s3：//location)
   - 目的地 - 目標目的地
   - 上次更新 - 最近的更新時間戳記
4. 使用搜尋方塊尋找向量擷取任務來尋找特定任務。
5. 若要管理任務，請從下列動作中選擇：
   - 選擇刪除以移除選取的任務
   - 選擇建立向量資料庫以建立其他任務
6. 在 Amazon S3 向量匯入歷史記錄區段中，追蹤匯入事件：
   a. 使用日期範圍篩選條件來指定匯入歷史記錄的期間。
   b. 使用狀態下拉式清單依匯入狀態（例如任何狀態）篩選。
   c. 使用搜尋方塊依 Amazon S3 向量索引 na... 尋找匯入，以尋找特定匯入。
   d. 檢視匯入詳細資訊，包括：
      - 匯入起始於 (UTC+5：30) - 匯入開始時
      - 匯入狀態 - 目前狀態（進行中、完成、失敗、部分完成）
      - Amazon S3 向量索引 ARN - 來源索引識別符
      - OpenSearch Service 向量集合 - 目的地集合
7. 選擇匯入 Amazon S3 向量以啟動新的匯入程序。

自動最佳化

自動最佳化是一項服務，可自動化向量索引最佳化，讓使用者在搜尋品質、速度和成本之間取得平衡，而不需要數週的手動專家調校。它會根據使用者定義的延遲和召回需求評估索引組態，並產生最佳化建議，因此需要最少的專業知識。建議通常會在 30-60 分鐘內交付。

傳統向量索引組態需要豐富的專業知識和實驗，才能達到最佳效能。`ef_construction`（控制索引建置品質）、`m`（決定圖形連線數）、`ef_search`（控制 HNSW 搜尋）和量化方法（二進位量化 (32x、16x、8x)、純量量化 (4x)) 等參數會大幅影響搜尋準確性和資源使用率。自動最佳化使用超參數最佳化演算法來探索索引組態，這些組態在您定義的延遲和召回需求內，最適合您的資料集。

優勢

OpenSearch 的自動最佳化提供下列優點：

- 自動化參數調校 - 消除對演算法 (HNSW)、量化、重新評分和引擎參數的手動實驗，節省時間並減少向量搜尋最佳化的學習曲線。
- 最佳化搜尋速度 - 根據預設，OpenSearch 已設定為記憶體內效能。自動最佳化可探索有利的權衡，改善搜尋品質並節省成本，同時維持可接受的搜尋速度。
- 成本最佳化 - 透過尋找可降低索引記憶體需求的選項來降低成本，同時將搜尋品質和速度權衡降至最低。
- 最佳化搜尋品質 - 可能提供比預設設定更高的召回率，或發現有利的權衡，可大幅節省成本並降低召回損失。

自動最佳化可與其他 OpenSearch 功能搭配使用，例如 向量索引的 GPU 加速，為向量搜尋工作負載提供全面的效能最佳化。

運作方式

自動最佳化透過以任務為基礎的架構運作，可分析向量資料並提供最佳化建議。重點：

- 使用者在 Amazon S3 儲存貯體中以 Parquet 或 JSONL 格式共用其資料集。
- 他們透過設定可接受的召回和延遲閾值來設定無伺服器自動最佳化任務。更寬鬆的閾值可讓服務探索更顯著的成本最佳化。
- 在由 Amazon OpenSearch Service 完全管理的基礎設施上執行的自動最佳化任務。任務不會在您的網域或集合上使用資源。工作者會平行執行以評估索引組態，並在大型資料集上使用取樣，通常在 30-60 分鐘內交付結果。

- 每個任務都會以可預測的固定費率計費。如需定價資訊，請參閱 Amazon OpenSearch Service 定價。

先決條件

- 資料集格式和許可 - 您的資料集必須是 Amazon S3 儲存貯體資料夾中的一或多個 Parquet 或 JSONL 檔案。例如：
  - Parquet： `s3://dataset-bucket-us-east-1/dataset_folder/first_half.parquet` 和 `s3://dataset-bucket-us-east-1/dataset_folder/second_half.parquet`
  - JSONL： `s3://dataset-bucket-us-east-1/dataset_folder/data.jsonl`

  提供封裝資料夾 URI（例如 `s3://dataset-bucket-us-east-1/dataset_folder/`）。資料夾必須包含單一格式的檔案 — 請勿在相同的資料夾中混合 Parquet 和 JSONL 檔案。此資料集將用於產生建議。確保您的聯合角色對該資源具有下列 Amazon S3 許可：`"s3:Get*", "s3:List*", "s3:Describe*"`。
- 指定正確的資料集中繼資料 - 提供的資料集必須包含浮點數列。每個資料欄的名稱和每個向量的維度必須符合 主控台中提供的選項。例如，如果資料集包含名為`train_data`每個768維度的向量，則這些值必須符合自動最佳化主控台。
- （如果使用向量擷取功能）- 如果您計劃使用擷取功能（接受自動最佳化建議以自動建立索引和擷取資料），您必須設定 OpenSearch 叢集，以授予自動最佳化許可，以將資料集擷取至 OpenSearch 叢集。對於具有網域存取政策的 OpenSearch 網域，請透過該政策授予新建立的角色存取。對於具有精細存取控制的 OpenSearch 網域，請將管道角色新增為後端角色。對於 OpenSearch Serverless 集合，將管道角色新增至資料存取政策。
- IAM 許可 - 您需要下列 IAM 許可才能使用自動最佳化：
  - `opensearch:SubmitAutoOptimizeJob`
  - `opensearch:GetAutoOptimizeJob`
  - `opensearch:DeleteAutoOptimizeJob`
  - `opensearch:CancelAutoOptimizeJob`
  - `opensearch:ListAutoOptimizeJobs`

> **Note**
>
> 這些是身分型政策。自動最佳化不支援以資源為基礎的政策。

- 登入資料過期 - 將聯合身分使用者工作階段設定為至少 1 小時的登入資料過期時間。對於非常大型的資料集或高維度，請考慮將過期持續時間增加至 3 小時。

自動最佳化的使用案例

在下列案例中，自動最佳化特別有價值：

初始組態最佳化

第一次部署向量搜尋應用程式時，判斷最佳 HNSW 參數通常需要廣泛的測試和領域專業知識。自動最佳化可分析您的資料和工作負載特性，以建議生產就緒組態，藉此消除此trial-and-error程序。

此使用案例非常適合初次搜尋向量的團隊，或從需要快速建立基準組態的其他向量資料庫平台遷移的團隊。

擴展最佳化

隨著向量資料集從數千個向量成長到數百萬個向量，一開始運作良好的參數可能會變得不理想。自動最佳化建議調整以大規模維持效能。

降低成本

向量索引可能會耗用大量的運算和儲存資源，尤其是在高維度內嵌的情況下。透過尋找更有效率的參數組態來維持所需的效能等級，同時使用較少的資源，自動最佳化可找出降低成本的機會。

例如，自動最佳化可能會發現您目前的m值（圖形連線能力）高於準確性需求所需的值，可讓您減少索引時間和儲存體，而不會影響搜尋品質。

效能故障診斷

在向量搜尋操作中遇到緩慢的查詢效能或高延遲時，自動最佳化可以分析資料集並識別更理想的組態。此服務提供特定建議來解決效能瓶頸，例如調整圖形連線或搜尋參數。

限制

- 區域可用性 - 自動最佳化僅適用於下列 AWS 區域：
  - ap-south-1
  - eu-west-1
  - us-west-2

  - us-east-2
  - us-east-1
  - eu-central-1
  - ap-southeast-2
  - ap-northeast-1
  - ap-southeast-1
- 集合類型 - 僅向量搜尋集合和 OpenSearch 網域 (2.19、3.1 和 3.3) 支援自動最佳化。
- 引擎支援

  依部署類型的引擎支援

引擎	無伺服器	OpenSearch 受管
盧塞內	否	是
Faiss	是	是
Nmslib	否	否

- 演算法支援 - 自動最佳化僅支援 HNSW 型向量索引。
- 並行任務 - 每個區域每個帳戶最多可以執行 10 個並行最佳化任務。如果達到限制，則無法接受任何新任務。
- 任務持續時間 - 最佳化任務可能需要 15 分鐘到數小時，具體取決於資料集大小、維度和所需的效能指標。
- 建議 - 自動最佳化建議最多 3 個建議。
- 資料集
  - 支援的格式：Parquet、JSONL
  - 資料存放區：Amazon S3

帳單與成本

無論資料集大小和最佳化組態為何，自動最佳化都會使用每項工作的定價模型，讓您為每項成功的最佳化任務付費。您不需要為失敗或取消的任務付費。此外，自動最佳化會在與受管或無伺服器 OpenSearch 叢集不同的基礎設施上執行，因此不會影響現有叢集的資源使用率。

定價方式

自動最佳化成本會與標準 OpenSearch Serverless 或 OpenSearch 受管網域運算和儲存成本分開計費。

如需定價資訊，請參閱 Amazon OpenSearch Service 定價。

支援的資料格式

自動最佳化支援下列存放在 Amazon S3 中的向量資料集資料格式：

Parquet 格式

Parquet 是針對分析工作負載最佳化的單欄式儲存格式。每個 Parquet 檔案都應包含代表向量資料的浮點數陣列資料欄。

Parquet 檔案結構範例 （檢視為資料表）：

```
| id  | train_data                      |
|-----|---------------------------------|
| 1   | [0.12, 0.45, 0.78, ..., 0.33] |
| 2   | [0.56, 0.89, 0.12, ..., 0.67] |
| 3   | [0.34, 0.67, 0.90, ..., 0.11] |
```

JSONL 格式

JSONL (JSON Lines) 是文字格式，其中每一行都是有效的 JSON 物件。每一行都應包含一個欄位，其中包含代表向量資料的浮點數陣列。

範例 JSONL 檔案：

```
{"id": 1, "train_data": [0.12, 0.45, 0.78, 0.33]}
{"id": 2, "train_data": [0.56, 0.89, 0.12, 0.67]}
{"id": 3, "train_data": [0.34, 0.67, 0.90, 0.11]}
```

在格式之間轉換

如果您的資料格式不同，您可以使用下列 Python 指令碼進行轉換。

將 JSON 或 JSONL 轉換為 Parquet

```
#!/usr/bin/env python3
import json
import pyarrow as pa
import pyarrow.parquet as pq
from pathlib import Path
from typing import Any, Dict, List


def load_json_any(path: Path) -> List[Dict[str, Any]]:
    """
    Load JSON that can be:
      - a list of objects
      - a single object
      - JSON Lines (one object per line)
    Returns list[dict].
    """
    text = path.read_text().strip()

    # Try full JSON file
    try:
        obj = json.loads(text)
        if isinstance(obj, list):
            return obj
        if isinstance(obj, dict):
            return [obj]
    except json.JSONDecodeError:
        pass

    # Fallback # JSON Lines
    records = []
    for i, line in enumerate(text.splitlines(), start=1):
        line = line.strip()
        if not line:
            continue
        try:
            rec = json.loads(line)
        except json.JSONDecodeError as e:
            raise ValueError(f"Invalid JSON on line {i}: {e}")
        if not isinstance(rec, dict):
            raise ValueError(f"Line {i} must contain a JSON object")
        records.append(rec)

    return records
```

```
def json_to_parquet(json_path: str, parquet_path: str, compression: str = "snappy"):
    """Convert ANY JSON to Parquet (schema inferred)."""
    records = load_json_any(Path(json_path))
    table = pa.Table.from_pylist(records)
    pq.write_table(table, parquet_path, compression=compression)
    print(f"Wrote {len(records)} rows to {parquet_path}")


if __name__ == "__main__":
    INPUT_JSON = "vectors.jsonl"
    OUTPUT_PARQUET = "vectors.parquet"
    json_to_parquet(INPUT_JSON, OUTPUT_PARQUET)
```

將 Parquet 轉換為 JSONL

```
#!/usr/bin/env python3
import json
import pyarrow.parquet as pq


def parquet_to_jsonl(parquet_path: str, jsonl_path: str):
    """Convert a Parquet file to JSONL format."""
    table = pq.read_table(parquet_path)
    rows = table.to_pylist()
    with open(jsonl_path, "w") as f:
        for row in rows:
            f.write(json.dumps(row) + "\n")
    print(f"Wrote {len(rows)} rows to {jsonl_path}")


if __name__ == "__main__":
    INPUT_PARQUET = "vectors.parquet"
    OUTPUT_JSONL = "vectors.jsonl"
    parquet_to_jsonl(INPUT_PARQUET, OUTPUT_JSONL)
```

在主控台中使用自動最佳化

您可以使用 Amazon OpenSearch Service 主控台來建立向量擷取任務、監控其進度、檢視最佳化建議，並根據這些建議建置索引。

先決條件

在主控台中使用自動最佳化之前，您必須具有下列項目：

- 可存取 OpenSearch 主控台的作用中 AWS 帳戶。
- 現有的 OpenSearch Serverless 類型向量搜尋集合或受管 OpenSearch 網域。
- 下列動作的 IAM 許可：
  - `opensearch:SubmitAutoOptimizeJob`
  - `opensearch:GetAutoOptimizeJob`
  - `opensearch:DeleteAutoOptimizeJob`
  - `opensearch:CancelAutoOptimizeJob`
  - `opensearch:ListAutoOptimizeJobs`

> **Note**
>
> 這些是以身分為基礎的政策。 AWS 不支援以資源為基礎的政策來自動最佳化資源。

- 將聯合身分使用者工作階段設定為具有至少 1 小時的最短憑證過期時間。對於非常大型的資料集或高維度，請考慮將過期持續時間增加至 3 小時。

建立向量擷取任務

向量擷取任務會分析向量資料，並提供索引組態的最佳化建議。

建立向量擷取任務

1. 在 登入 Amazon OpenSearch Service 主控台AWS 管理主控台。
2. 在導覽窗格中，選擇自動最佳化。
3. 選擇建立向量擷取任務。
4. 在任務詳細資訊下，輸入向量擷取任務的名稱。此名稱可協助您在 主控台中識別任務。
5. 在資料來源下，設定下列項目：

   a. 針對 Amazon S3 URI，輸入包含資料檔案 (Parquet 或 JSONL) 之資料夾的 Amazon S3 URI。URI 必須指向封裝資料夾，而不是個別檔案。例如，如果您的檔案位於 `s3://my-bucket/my-folder/file1.parquet`或 `s3://my-bucket/my-folder/data.jsonl`，請輸入 `s3://my-bucket/my-folder/`。

> **Note**
>
> 資料夾必須包含單一格式的檔案。請勿在相同的資料夾中混合 Parquet 和 JSONL 檔案。

b. 針對區域，選取 Amazon S3 儲存貯體所在的 AWS 區域。區域必須符合儲存貯體位置。

6. 在 OpenSearch 網域下，選取現有的網域或集合，或選擇建立新網域或集合。

> **Note**
>
> 您可以指定 OpenSearch Managed 網域或 OpenSearch Serverless Serverless 集合。

7. 在資料來源許可下，指定具有存取 Amazon S3 儲存貯體和 OpenSearch 網域或集合許可的 IAM 角色。該角色必須根據您的網域或集合組態具有必要的許可：
   - 對於具有網域存取政策的 OpenSearch 網域，請透過該政策授予角色存取權。
   - 對於具有精細存取控制的 OpenSearch 網域，請將角色新增為後端角色。
   - 對於 OpenSearch Serverless 集合，請將 角色新增至資料存取政策。
8. 選擇下一步。
9. 在設定索引下，指定下列項目：
   a. 對於欄位名稱，輸入包含向量資料之資料集的欄位名稱。
   b. 針對空格類型，選取用於計算向量之間距離的距離指標：
      - l2 - 歐幾里得距離
      - cosinesimil - 餘弦相似性
      - 內部產品 - 內部產品
   c. 針對維度，輸入每個向量中的浮點數。
10. 在效能需求下，設定下列項目：
    a. 對於召回，將所需的搜尋品質指定為介於 0 到 1 之間的十進位值。較高的回收值會傳回更相關的結果。例如：
       - 0.95 表示平均傳回 20 個最接近查詢向量的文件向量中的 19 個
       - 0.9 表示 9/10
       - 0.8 表示 10 中有 8 個

b. 針對搜尋延遲需求，選取您的延遲容錯能力。中等要求允許透過降低記憶體需求的壓縮方法來節省更多成本。

11. 選擇下一步。
12. 檢閱您的組態，然後選擇建立。

任務開始處理。您可以在向量擷取任務表格中監控其進度。

監控最佳化任務

您可以從自動最佳化登陸頁面監控向量擷取任務的狀態。

監控最佳化任務

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在導覽窗格中，選擇自動最佳化。
3. 向量擷取任務資料表會顯示具有其目前狀態的所有任務。重新整理頁面以查看更新的狀態資訊。

> **Note**
>
> 沒有自動重新整理或通知機制。您必須手動重新整理主控台，以查看任務何時完成。

了解任務狀態

自動最佳化任務可以有下列狀態值：

待定

任務已排入佇列並等待啟動。

執行中

自動最佳化任務正在主動分析您的資料並產生建議。

已完成

自動最佳化任務已成功完成。所有分析、評估和建議都已完成，可供檢視。

失敗

任務發生錯誤。檢視任務詳細資訊頁面中的錯誤詳細資訊，以判斷原因。

作用中

已在連接的叢集中建立索引，並已擷取資料。

任務持續時間主要取決於資料集大小和目前的服務負載。一般任務會在 15 分鐘到數小時內完成。

檢視任務詳細資訊

您可以檢視特定最佳化任務的詳細資訊，包括其組態和狀態。

檢視任務詳細資訊

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在導覽窗格中，選擇自動最佳化。
3. 在向量擷取任務表格中，選擇任務名稱。
4. 任務詳細資訊頁面會顯示下列資訊：
   - 任務名稱和狀態
   - 資料來源組態 (Amazon S3 URI 和區域）
   - OpenSearch 網域或集合
   - 索引組態（欄位名稱、空格類型、維度）
   - 效能需求（召回和延遲）
   - 錯誤訊息（如果任務失敗）

檢視和了解結果

任務成功完成後，您可以檢視最佳化建議。

檢視最佳化結果

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在導覽窗格中，選擇自動最佳化。
3. 在向量擷取任務表格中，選擇狀態為已完成的任務。
4. 結果頁面會顯示下列區段：

- 結果概觀 - 顯示與您的需求相比的預估搜尋品質回收，以及與最佳建議組態相比的索引記憶體使用量。
- 建議 - 列出最多三個最佳化建議，並依最佳建議排序，做為最適合您組態的建議。每個建議包括：
  - 索引組態參數
  - 搜尋組態參數
  - 預期的效能指標
  - 記憶體使用量預估

> **Note**
>
> 雖然建議是依最佳相符項目排序，但您可以選擇任何更符合您特定使用案例的建議。自動最佳化嘗試尋找最接近您所選取回條件的相符項目。

從建議建立索引

檢閱最佳化建議後，您可以使用建議的組態手動建立索引，或使用選取的建議自動建置索引。

自動建置索引

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在導覽窗格中，選擇自動最佳化。
3. 在向量擷取任務表格中，選擇狀態為已完成的任務。
4. 檢閱建議，然後選取您要使用的建議。
5. 選擇建置索引。
6. 系統會使用選取的建議在您的叢集中自動建立索引，並從資料集擷取向量資料。

手動建置索引

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在導覽窗格中，選擇自動最佳化。

3. 在向量擷取任務表格中，選擇狀態為已完成的任務。
4. 檢閱建議，並記下所選建議的索引組態和搜尋組態參數。
5. 使用 OpenSearch API 或主控台，以建議的參數手動建立索引。

相關功能

自動最佳化可與其他 Amazon OpenSearch Service 功能搭配使用，協助您建置和最佳化向量搜尋應用程式：

- 向量索引的 GPU 加速 - 使用 GPU 加速加速向量索引建置，以減少索引時間和成本。
- 向量擷取 - 快速將向量資料從 Amazon S3 擷取並編製索引到您的網域或集合。

向量索引的 GPU 加速

GPU 加速可協助您更快、更有效率地建置大規模向量資料庫。您可以在新的或現有的 OpenSearch 網域和 OpenSearch Serverless 集合上啟用此功能。此功能使用 GPU 加速來減少將資料索引為向量索引所需的時間。

透過 GPU 加速，您可以提高向量索引速度高達四分之一的索引成本 10X。

先決條件

執行 OpenSearch 版本 `3.1` 或更新版本的 OpenSearch 網域和 OpenSearch Serverless 集合支援 GPU 加速。如需詳細資訊，請參閱 升級 Amazon OpenSearch Service 網域、UpdateDomainConfig 和 UpdateCollection APIs。

運作方式

向量索引需要大量的運算資源來建置資料結構，例如階層式導航小型世界 (HNSW) 圖形。當您在網域或集合上啟用 GPU 加速時，OpenSearch 會自動偵測加速索引建置的機會，並將索引建置卸載至 GPU 執行個體。OpenSearch Service 會代表您管理 GPU 執行個體，並在需要時將其指派給您的網域或集合。這表示您不管理使用率或支付閒置時間。

您只需透過運算單位 (OCU) - 向量加速支付有用的處理費用。每個向量加速 OCU 是大約 8 GiB CPU 記憶體、2 vCPUs 和 6 GiB GPU 記憶體的組合。如需詳細資訊，請參閱the section called “GPU 加速定價”。

若要為您的網域或集合啟用 GPU 加速，請參閱 啟用 GPU 加速。

GPU 加速定價

AWS 當 OpenSearch 偵測到加速網域或集合索引建置工作負載的機會時， 會向您收取費用。每個向量加速 OCU 是大約 8 GiB CPU 記憶體、2 vCPUs 和 6 GiB GPU 記憶體的組合。

AWS 以第二級精細程度向 OCU 收費。在您的帳戶陳述式中，您會看到以 OCU 小時為單位的運算項目。

例如，當您使用 GPU 加速一小時來建立索引時，使用 2 個 vCPU 和 1 GiB 的 GPU 記憶體，則會向您收取 1 個 OCU。如果您在使用 GPU 加速時使用 9 GiB 的 CPU 記憶體，則會向您收取 2 個 OCU 的費用。

OpenSearch Serverless 會根據支援集合所需的運算能力和儲存體， OCUs 以 1 個 OCU 的增量新增額外的 OCU。您可以為您的帳戶設定 OCU 數量上限，以控制成本。

> **Note**
>
> 隨時佈建OCUs 數量可能不同，也不確切。隨著時間的推移，OpenSearch 和 OpenSearch Serverless 使用的演算法將繼續改進，以更好地最大限度地減少系統使用量。

如需完整定價詳細資訊，請參閱 Amazon OpenSearch Service 定價。

GPU 加速和寫入操作

當 OpenSearch 的向量擷取速率 (MB/秒） 在範圍內時，會啟用 GPU 加速。在 OpenSearch 網域上，您可以靈活地透過 `index.knn.remote_index_build.size.min`和 設定此範圍`index.knn.remote_index_build.size.max`。例如，在較低的範圍預設值為 50 MB 的情況下，寫入 15，000 個完整精確度向量，且重新整理間隔之間具有 768 個維度，預設會觸發 GPU 加速。

使用下列 API 操作寫入資料：

- 排清
- 大批
- 重新索引
- 索引
- 更新
- 刪除

- 強制合併

GPU 加速可透過自動和手動區段合併來啟用。

支援的索引組態

Faiss 引擎支援 GPU 加速。

下列組態不支援 GPU 加速：

- Faiss 產品量化
- 反轉檔案索引 (IVF)
- 非指標空間程式庫
- Lucene 引擎

支援的 AWS 區域

GPU 加速可在下列內容中使用 AWS 區域：

- 美國東部 (維吉尼亞北部)
- 美國西部 (奧勒岡)
- 亞太地區 (雪梨)
- 亞太地區 (東京)
- 歐洲 (愛爾蘭)

啟用 GPU 加速

您可以在使用 AWS 管理主控台或 AWS CLI AWS SDK 建立或更新 OpenSearch 網域或 OpenSearch Serverless 集合時啟用 GPU 加速。

在網域或集合上啟用 GPU 加速後，此功能會在所有索引上預設為啟用。如果您需要在索引層級停用此功能，請參閱 建立 GPU 加速向量索引。

主控台

下列程序使用 OpenSearch Serverless 管理主控台啟用 OpenSearch 網域和 OpenSearch Serverless 集合的 GPU 加速。

Create new domain

若要建立已啟用 GPU 加速的 OpenSearch 網域，請參閱 建立 OpenSearch Service 網域。

Edit existing domain

1. 開啟 OpenSearch Service 管理主控台。
2. 在導覽窗格中，選擇網域。
3. 選擇您的網域名稱以開啟網域詳細資訊頁面。
4. 選擇動作，然後選擇編輯網域。
5. 在進階功能區段中，選取啟用 GPU 加速。啟用此功能後，您的向量索引操作就會加速。
6. 選擇儲存變更。

Create new collection

若要建立已啟用 GPU 加速的 OpenSearch Serverless 集合，請參閱 教學課程：Amazon OpenSearch Serverless 入門。在建立集合期間，請確定您選取向量搜尋集合類型，並在向量搜尋組態中啟用 GPU 加速。

Edit existing collection

1. 開啟 OpenSearch Service 管理主控台。
2. 在導覽窗格中，選擇集合。
3. 選擇您的集合名稱以開啟集合詳細資訊頁面。
4. 在部署選項區段中，編輯向量 GPU 加速。
5. 停用或啟用 GPU 加速。
6. 選擇儲存變更。

AWS CLI

Create new domain

下列 AWS CLI 範例會建立在美國東部（維吉尼亞北部）啟用 GPU 加速的 OpenSearch 網域。將##取代為您自己的組態文字。

```
aws opensearch create-domain \
    --domain-name my-domain \
```

```
    --engine-version OpenSearch_3.1 \
    --cluster-config InstanceType=r6g.xlarge.search,\
        InstanceCount=1,\
        DedicatedMasterEnabled=true,\
        DedicatedMasterCount=3,\
        DedicatedMasterType=m6g.large.search \
    --ebs-options "EBSEnabled=true,\
        VolumeType=gp3,\
        VolumeSize=2000" \
    --encryption-at-rest-options '{"Enabled":true}' \
    --aiml-options '{"ServerlessVectorAcceleration": {"Enabled": true}}' \
    --node-to-node-encryption-options '{"Enabled":true}' \
    --domain-endpoint-options '{"EnforceHTTPS":true,\
        "TLSSecurityPolicy":"Policy-Min-TLS-1-0-2019-07"}' \
    --access-policies '{"Version": "2012-10-17",
        "Statement": [{
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": "es:*",
            "Resource": "arn:aws:es:us-east-1:123456789012:domain/my-domain/*"
        }]}' \
    --advanced-security-options '{
        "Enabled":true,
        "InternalUserDatabaseEnabled":true,
        "MasterUserOptions": {
            "MasterUserName":"USER_NAME",
            "MasterUserPassword":"PASSWORD"
        }}' \
    --region us-east-1
```

Edit existing domain

下列 AWS CLI 範例會啟用現有 OpenSearch 網域的 GPU 加速。將##取代為您自己的組態文字。

```
aws opensearch update-domain-config \
    --domain-name my-domain \
    --cluster-config InstanceType=r7g.16xlarge.search,InstanceCount=3 \
    --aiml-options '{"ServerlessVectorAcceleration": {"Enabled": true}}'
```

Create new collection

下列 AWS CLI 範例會建立在美國東部（維吉尼亞北部）啟用 GPU 加速的 OpenSearch Serverless 集合。將##取代為您自己的組態文字。

```
aws opensearchserverless create-collection \
    --name "my-collection" \
    --type "VECTORSEARCH" \
    --description "My vector collection with GPU acceleration" \
    --vector-options '{"ServerlessVectorAcceleration": "ENABLED"}' \
    --region us-east-1
```

Edit existing collection

下列 AWS CLI 範例會啟用現有 OpenSearch Serverless 集合的 GPU 加速。將##取代為您自己的組態文字。

```
aws opensearchserverless update-collection \
    --id 07tjusf2h91cunochc \
    --vector-options '{"ServerlessVectorAcceleration": "ENABLED"}' \
    --region us-east-1
```

建立 GPU 加速向量索引

在網域或集合上啟用 GPU 加速之後，請建立向量索引，以利用 GPU 處理。

> **Note**
>
> 當您建立已啟用 GPU 加速的網域時，`true`預設為`index.knn.remote_index_build.enabled`設定。建立索引時，您不需要明確設定此設定。對於集合，您必須明確指定此設定的值。

Creating index with GPU-acceleration

下列範例會建立針對 GPU 處理最佳化的向量索引。此索引存放 768 維向量（常見於文字內嵌）。

```
PUT my-vector-index
{
  "settings": {
    "index.knn": true,
    "index.knn.remote_index_build.enabled": true
  },
  "mappings": {
```

```
    "properties": {
      "vector_field": {
        "type": "knn_vector",
        "dimension": 768
      },
      "text": {
        "type": "text"
      }
    }
  }
}
```

金鑰組態元素：

- "index.knn": true - 啟用 k 近鄰功能
- "index.knn.remote_index_build.enabled": true - 啟用此索引的 GPU 處理。當網域已啟用 GPU 加速時，未指定此設定預設為 true。對於集合，您必須明確指定此設定的值。
- "dimension": 768 - 指定向量大小（根據您的內嵌模型調整）

Creating index without GPU-acceleration

下列範例會建立停用 GPU 處理的向量索引。此索引存放 768 維向量（常見於文字內嵌）。

```
PUT my-vector-index
{
  "settings": {
    "index.knn": true,
    "index.knn.remote_index_build.enabled": false
  },
  "mappings": {
    "properties": {
      "vector_field": {
        "type": "knn_vector",
        "dimension": 768
      },
      "text": {
        "type": "text"
      }
    }
  }
}
```

索引向量資料和強制合併

在網域或集合上建立 GPU 加速向量索引後，您可以使用標準 OpenSearch 操作新增向量資料並最佳化索引。GPU 加速會自動增強索引效能和強制合併操作，讓您更快速地建置和維護大規模向量搜尋應用程式，而不需要變更現有的工作流程。

索引向量資料

像往常一樣索引向量資料。GPU 加速會自動套用至索引和強制合併操作。下列範例示範如何使用大量 API 將向量文件新增至索引。每個文件都包含一個向量欄位，其中包含數值和相關文字內容：

```
POST _bulk
{"index": {"_index": "my-vector-index"}}
{"vector_field": [0.1, 0.2, 0.3, ...], "text": "Sample document 1"}
{"index": {"_index": "my-vector-index"}}
{"vector_field": [0.4, 0.5, 0.6, ...], "text": "Sample document 2"}
```

強制合併操作

GPU 加速也適用於強制合併操作，可大幅縮短最佳化向量索引所需的時間。請注意，集合不支援強制合併操作。下列範例示範如何透過將所有區段合併為單一區段來最佳化向量索引：

```
POST my-vector-index/_forcemerge?max_num_segments=1
```

最佳實務

遵循下列最佳實務，將向量搜尋工作負載的 GPU 加速優勢發揮到最大：

- 增加索引用戶端 - 若要在索引建置期間充分利用 GPUs，請增加將資料擷取至 OpenSearch 的索引用戶端數量。這可提高 GPU 資源的平行處理和使用率。
- 調整近似閾值 - 變更 `index.knn.advanced.approximate_threshold` 設定，以確保不會發生較小的區段索引組建，從而改善擷取的整體速度。值 10，000 是很好的起點。對於集合，您必須明確指定此設定的值。
- 最佳化碎片大小 - 嘗試建立至少具有 100 萬份文件的碎片。少於此數量的文件碎片可能無法從 GPU 加速中看到整體優勢。

搭配使用 OpenSearch Dashboards 與 Amazon OpenSearch Service

OpenSearch Dashboards 是一種開放原始碼視覺化工具，專為與 OpenSearch 搭配使用而設計。Amazon OpenSearch Service 提供每個 OpenSearch Service 網域的 Dashboards 安裝。儀表板會在網域中的常用資料節點上執行。

OpenSearch Dashboards 是一種視覺化工具，用於探索和分析單一 OpenSearch 網域中的資料。相反地，集中式 OpenSearch 使用者介面（也稱為 OpenSearch 應用程式）是雲端型 UI，可連線至多個 OpenSearch 網域、OpenSearch Serverless 集合和 AWS 資料來源。它包含特定使用案例的工作區，例如可觀測性和安全性分析，並提供跨資料集的統一體驗。雖然 Dashboards 繫結至個別網域，但集中式 UI 可啟用跨網域資料整合和分析。如需詳細資訊，請參閱OpenSearch UI。

您可以在 OpenSearch Service 主控台的網域儀表板上找到 OpenSearch Dashboards 的連結。對於執行 OpenSearch 的網域，URL 為 *domain-endpoint*`/_dashboards/`。對於執行舊版 Elasticsearch 的網域，URL 為 *domain-endpoint*`/_plugin/kibana`。

使用此預設 Dashboards 安裝的查詢具有 300 秒的逾時。

> **Note**
>
> 本文件討論 Amazon OpenSearch Service 內容中的 OpenSearch Dashboards，包括不同的連線方式。如需完整的文件，包括入門指南、建立儀表板的說明、儀表板管理和儀表板查詢語言 (DQL)，請參閱開放原始碼 OpenSearch 文件中的 OpenSearch Dashboards。 OpenSearch

控制對 Dashboards 的存取

Dashboards 原本不支援 IAM 使用者和角色，但 OpenSearch Service 提供多種解決方案以控制 Dashboards 的存取：

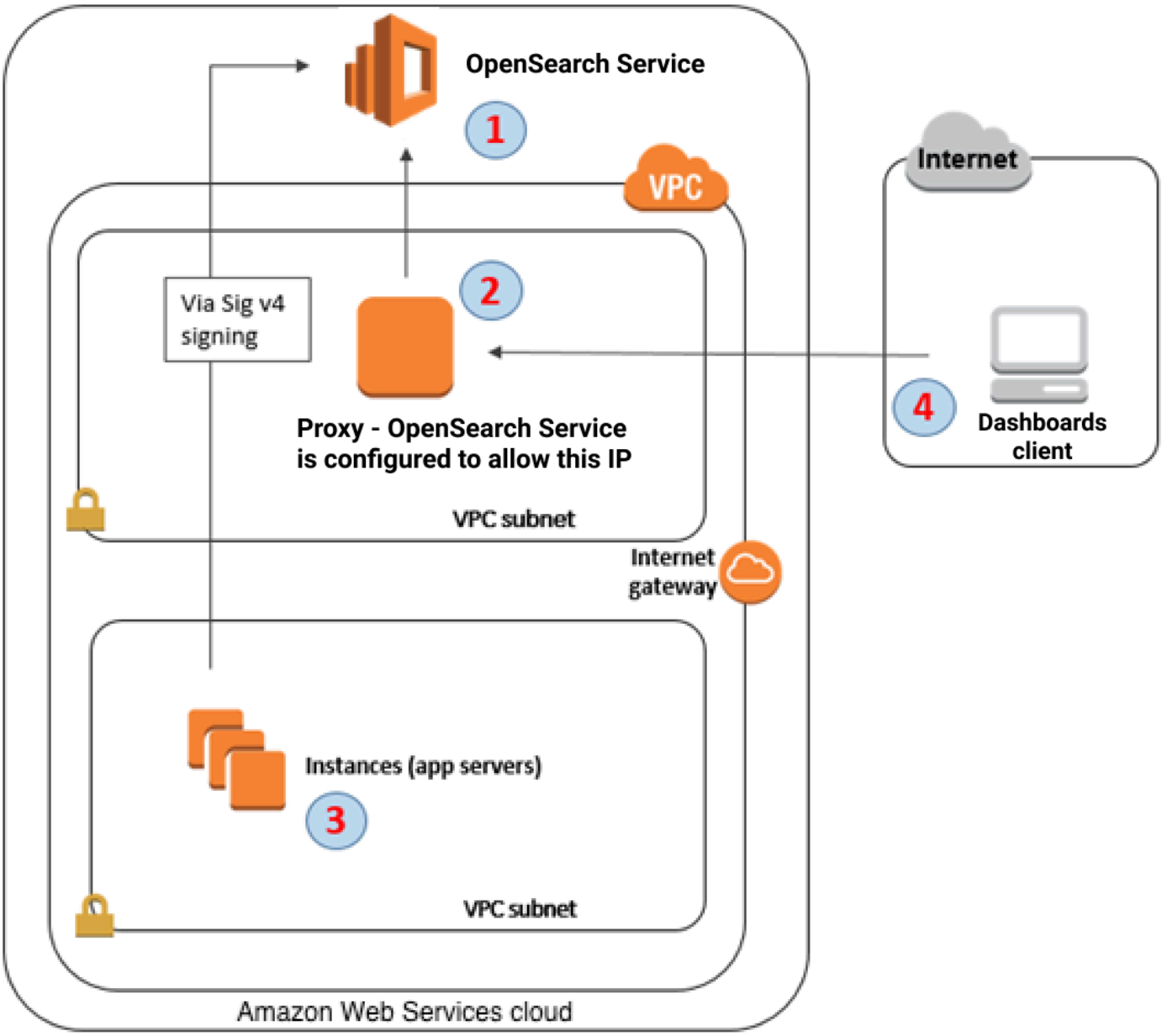
- 啟用 Dashboards 的 SAML 身分驗證。
- 搭配使用精細存取控制與 HTTP 基本身分驗證。
- 設定儀表板的 Cognito 身分驗證。
- 對於公有存取網域，設定 IP 型存取政策 (使用或不使用代理伺服器)。

- 對於 VPC 存取網域，使用開放存取政策 (使用或不使用代理伺服器) 和安全群組來控制存取。如需詳細資訊，請參閱 the section called “關於 VPC 網域上的存取政策”。

使用代理伺服器從 Dashboards 中存取 OpenSearch Service

> **Note**
>
> 此程序僅當您的網域使用公有存取而您不想使用 Cognito 身分驗證時才適用。請參閱 the section called “控制對 Dashboards 的存取”。

由於 Dashboards 是一種 JavaScript 應用程式，請求會來自使用者的 IP 地址。以 IP 為基礎的存取控制可能不切實際，因為您為了每個使用者可以存取 Dashboards，而需要允許的 IP 地址相當龐大。解決方法是在 Dashboards 和 OpenSearch Service 之間放置代理伺服器。然後，您可以新增 IP 為基礎的存取政策，允許僅從一個 IP 地址 (代理的) 發出請求。下圖顯示此組態。



1. 這是您的 OpenSearch Service 網域。IAM 提供此網域的授權存取。額外的 IP 為基礎的存取政策提供代理伺服器的存取。
2. 這是在 Amazon EC2 執行個體上執行的代理伺服器。
3. 其他應用程式可以使用 Signature 4 版簽署程序，將已經過身分驗證的請求傳送至 OpenSearch Service。
4. Dashboards 用戶端會透過代理連線到您的 OpenSearch Service 網域。

若要啟用此排序的組態，您需要以資源為基礎的政策，來指定角色和 IP 地址。以下是範例政策：

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Resource":"arn:aws:es:us-west-2:111111111111:domain/my-domain/*",
         "Principal":{
            "AWS":"arn:aws:iam::111111111111:role/allowedrole1"
         },
         "Action":[
            "es:ESHttpGet"
         ],
         "Effect":"Allow"
      },
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":"*"
         },
         "Action":"es:*",
         "Condition":{
            "IpAddress":{
               "aws:SourceIp":[
                  "203.0.113.0/24",
                  "2001:DB8:1234:5678::/64"
               ]
            }
         },
         "Resource":"arn:aws:es:us-west-2:111111111111:domain/my-domain/*"
      }
   ]
}
```

我們建議您設定執行代理伺服器 (含彈性 IP 地址) 的 EC2 執行個體。如此，您可以在必要時更換執行個體，同時連接相同的公有 IP 地址。若要進一步了解，請參閱《Amazon EC2 使用者指南》中的彈性 IP 地址。

如果您使用代理伺服器和 Cognito 身分驗證，您可能需要新增 Dashboards 和 Amazon Cognito 的設定，以避免發生 `redirect_mismatch` 錯誤。請參閱以下 `nginx.conf` 範例：

```
server {
    listen 443;
    server_name $host;
    rewrite ^/$ https://$host/_plugin/_dashboards redirect;

    ssl_certificate           /etc/nginx/cert.crt;
    ssl_certificate_key       /etc/nginx/cert.key;

    ssl on;
    ssl_session_cache  builtin:1000  shared:SSL:10m;
    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
    ssl_prefer_server_ciphers on;

    location /_plugin/_dashboards {
        # Forward requests to Dashboards
        proxy_pass https://$dashboards_host/_plugin/_dashboards;

        # Handle redirects to Cognito
        proxy_redirect https://$cognito_host https://$host;

        # Update cookie domain and path
        proxy_cookie_domain $dashboards_host $host;
        proxy_cookie_path / /_plugin/_dashboards/;

        # Response buffer settings
        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;
    }

    location ~ \/(log|sign|fav|forgot|change|saml|oauth2) {
        # Forward requests to Cognito
        proxy_pass https://$cognito_host;

        # Handle redirects to Dashboards
        proxy_redirect https://$dashboards_host https://$host;

        # Update cookie domain
        proxy_cookie_domain $cognito_host $host;
    }
}
```

> **Note**
>
> （選用）如果您選擇佈建專用協調器節點，則會自動開始託管 OpenSearch Dashboards。因此，CPU 和記憶體等資料節點資源的可用性會增加。增加資料節點資源的可用性有助於改善網域的整體彈性。

設定儀表板以使用 WMS 地圖伺服器

適用於 OpenSearch Service 的 Dashboards 預設安裝包含地圖服務，印度和中國區域中的網域除外。地圖服務最多支援 10 個縮放比例。

無論您的區域為何，您皆可設定 Dashboards，以將不同的 Web Map Service (WMS) 伺服器用於座標地圖視覺化。區域地圖視覺化僅支援預設地圖服務。

若要設定 Dashboards 以使用 WMS 地圖伺服器：

1. 開啟 Dashboards。
2. 選擇 Stack Management (堆疊管理)。
3. 選擇 Advanced Settings (進階設定)。
4. 尋找 visualization:tileMap:WMSdefaults。
5. 將 `enabled` 變更為 `true`，並將 `url` 變更為有效 WMS 地圖伺服器的 URL：

```
{
  "enabled": true,
  "url": "wms-server-url",
  "options": {
    "format": "image/png",
    "transparent": true
  }
}
```

6. 選擇儲存變更。

若要將新的預設值套用到視覺化，您可能需要重新載入 Dashboards。如果您已儲存視覺化，請在開啟視覺化後選擇 Options (選項)。確認 WMS map server (WMS 地圖伺服器) 已啟用且 WMS url 包含您偏好使用的地圖伺服器，然後選擇 Apply changes (套用變更)。

> **Note**
>
> 地圖服務通常需要授權費或限制。您要對您指定在任何地圖伺服器負責所有這類考量。您可能會發現美國地質調查局的地圖服務適用於測試。

將本機 Dashboards 伺服器連線至 OpenSearch Service

如果您已投入大量時間來設定自己的 Dashboards 執行個體，您可以使用它，而不是（或除了）OpenSearch Service 提供的預設 Dashboards 執行個體。下列程序適用於搭配使用精細的存取控制與開放存取政策的網域。

將本機 Dashboards 伺服器連線至 OpenSearch Service

1. 在您的 OpenSearch Service 網域上，建立具有適當權限的使用者：
   a. 在 Dashboards 中，移至 Security (安全性)、Internal users (內部使用者)，然後選擇 Create internal user (建立內部使用者)。
   b. 提供使用者名稱和密碼，然後選擇 Create (建立)。
   c. 移至 Roles (角色)，然後選取角色。
   d. 選取 Mapped users (映射的使用者)，然後選擇 Manage mapping (管理映射)。
   e. 在 Users (使用者) 中，新增您的使用者名稱並選擇 Map (映射)。
2. 下載並在您的自我管理的 Dashboards OSS 安裝上安裝適當版本的 OpenSearch 安全外掛程式。
3. 在本機 Dashboards 伺服器上，開啟 `config/opensearch_dashboards.yml` 檔案，然後使用您之前建立的使用者名稱和密碼來新增您的 OpenSearch Service 端點：

   ```
   opensearch.hosts: ['https://domain-endpoint']
   opensearch.username: 'username'
   opensearch.password: 'password'
   ```

   您可以使用以下範例 `opensearch_dashboards.yml` 檔案：

   ```
   server.host: '0.0.0.0'

   opensearch.hosts: ['https://domain-endpoint']

   opensearchDashboards.index: ".username"
   ```

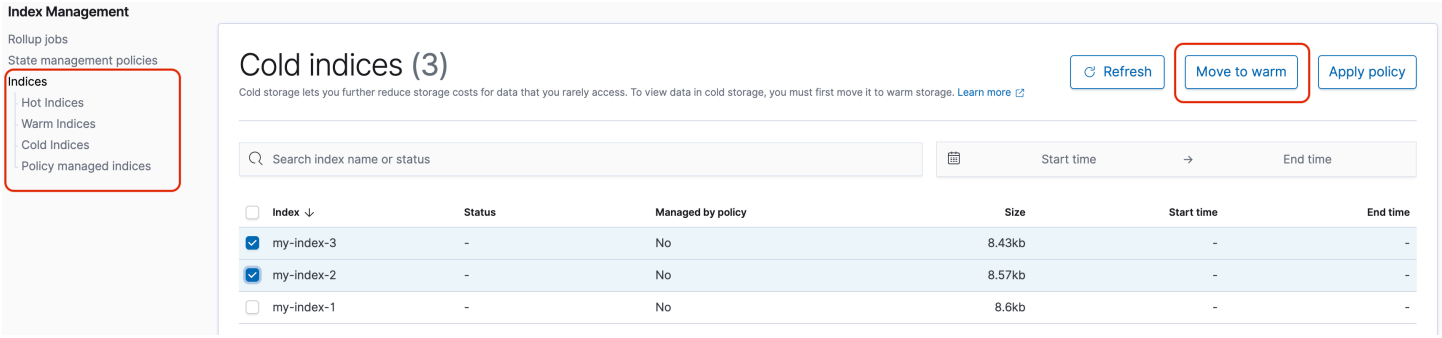
```
opensearch.ssl.verificationMode: none # if not using HTTPS

opensearch_security.auth.type: basicauth
opensearch_security.auth.anonymous_auth_enabled: false
opensearch_security.cookie.secure: false # set to true when using HTTPS
opensearch_security.cookie.ttl: 3600000
opensearch_security.session.ttl: 3600000
opensearch_security.session.keepalive: false
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ['opensearch_dashboards_read_only']
opensearch_security.auth.unauthenticated_routes: []
opensearch_security.basicauth.login.title: 'Please log in using your username and
 password'

opensearch.username: 'username'
opensearch.password: 'password'
opensearch.requestHeadersWhitelist: [authorization, securitytenant,
 security_tenant]
```

若要查看您的 OpenSearch Service 索引，請啟動本機 Dashboards 伺服器，前往開發工具並執行下列命令：

```
GET _cat/indices
```

在儀表板中管理索引

在 OpenSearch Service 網域上安裝的 Dashboards 提供有用的 UI，用於管理網域上不同儲存層的索引。從 Dashboards 主選單中選擇 Index Management (索引管理)，以檢視熱儲存、UltraWarm 以及冷儲存中的所有索引，以及索引狀態管理 (ISM) 政策管理的索引。使用索引管理在暖儲存和冷儲存之間移動索引，並監控三個層級之間的遷移。



請注意，除非啟用了 UltraWarm 及/或冷儲存，否則您將不會看到熱索引、暖索引和冷索引選項。

額外功能

在每個 OpenSearch Service 網域上安裝的預設儀表板具有一些其他功能：

- 各種 OpenSearch 外掛程式的使用者介面
- 租用戶
- 報告

  使用 Reporting (報告) 選單，從 Discover (探索) 頁面中產生隨需 CSV 報告，以及儀表板或視覺效果的 PDF 或 PNG 報告。CSV 報告的資料列限制為 10,000。
- 甘特圖
- 筆記本

在 Amazon OpenSearch Service 中使用 OpenSearch UI

OpenSearch UI（使用者介面）是 Amazon OpenSearch Service 的現代化操作分析體驗，可提供統一檢視，讓您跨多個來源與資料互動。與僅與託管它的一個網域或集合搭配使用的 OpenSearch Dashboards 不同，OpenSearch UI 託管在 中 AWS 雲端。這可讓 OpenSearch UI 在叢集升級期間實現高可用性並保持正常運作，以及原生與多個資料來源連線。如需 OpenSearch Dashboards 的詳細資訊，請參閱 OpenSearch Dashboards。

以下是 OpenSearch UI 的主要功能：

- 多個資料來源支援 – OpenSearch UI 可以與多個資料來源連線，以建立全面的檢視。這包括 OpenSearch 網域和無伺服器集合，以及整合 AWS 資料來源，例如 Amazon CloudWatch、Amazon Security Lake 和 Amazon Simple Storage Service (Amazon S3)。
- 升級期間零停機時間 – OpenSearch UI 託管在 中 AWS 雲端。這表示 OpenSearch 保持運作，並且可以在升級程序期間從叢集擷取資料。
- 工作區 – 為各種工作流程的團隊協作預留空間，例如可觀測性、安全性分析和搜尋。您可以在工作區中定義協作者的隱私權設定和管理許可。
- 單一登入 – OpenSearch UI 使用 AWS IAM Identity Center 和 SAML through AWS Identity and Access Management (IAM) 聯合，與您的身分提供者整合，為您的最終使用者建立單一登入體驗。
- GenAI 支援的分析 – OpenSearch UI 支援自然語言查詢產生，以協助為您的分析產生正確的查詢。OpenSearch UI 也與 Amazon Q Developer 搭配使用，以提供 Amazon Q 聊天，並協助產生視覺化、提醒摘要、洞見和建議的異常偵測器。
- 多查詢語言支援 – OpenSearch UI 支援管道處理語言 (PPL)、SQL、Lucene 和儀表板查詢語言 (DQL)。
- 跨區域和跨帳戶支援 – OpenSearch UI 可以使用跨叢集搜尋功能來連接不同帳戶和不同區域中的 OpenSearch 網域，以進行彙總分析和視覺化。

若要開始使用並建立您的第一個 OpenSearch UI，請遵循 中的指示the section called “開始使用”。

如需針對 OpenSearch UI 發行的最新功能的相關資訊，請參閱 the section called “版本歷史記錄”。

主題

- Amazon OpenSearch Service 使用者介面發行歷史記錄
- Amazon OpenSearch Service 中的 OpenSearch 使用者介面入門

- Amazon OpenSearch Service 中的代理式 AI
- 使用客戶受管金鑰加密 OpenSearch UI 應用程式中繼資料
- 使用 啟用 SAML 聯合 AWS Identity and Access Management
- 管理資料來源關聯和虛擬私有雲端存取許可
- 使用 Amazon OpenSearch Service 工作區
- 跨區域和跨帳戶資料存取
- 從 VPC 端點管理對 OpenSearch UI 的存取
- 將儲存的物件從 OpenSearch Dashboards 遷移至 OpenSearch UI
- OpenSearch UI 端點和配額

Amazon OpenSearch Service 使用者介面發行歷史記錄

下表列出 OpenSearch UI 的所有 Amazon OpenSearch Service 支援版本，以及每個版本隨附的功能和增強功能。

變更	版本日期	Description
新功能	2025-03-10	OpenSearch UI 現在支援跨帳戶資料存取 OpenSearch 網域。您可以針對公有和 VPC 網域，將一個帳戶中的應用程式設定為使用 IAM 或 AWS IAM Identity Center 身分驗證來存取不同帳戶中的網域。如需詳細資訊，請參閱the section called “OpenSearch 網域的跨帳戶資料存取”。
新功能	2025-12-29	OpenSearch UI 現在支援客戶受管金鑰 (CMK) 來加密應用程式中繼資料。此功能可讓您完全控制加密金鑰，協助您符合法規和合規要求。此啟動也會增加 OpenSearch UI 中已儲存物件的中繼資料大小限制，讓您能夠建立和存放複雜的查詢、廣泛的視覺化效果和大規模工作區。如需詳細資訊，請參閱the section called “加密和客戶受管金鑰”。
新功能	2025-04-16	OpenSearch UI 現在適用於跨叢集搜尋。這可讓您在一個 中使用 AWS 區域 OpenSearch UI 來存取不同區域中的叢集。方法是將其設定為與相同區域中的叢集連線的遠端叢集。如需詳細資訊，請參閱the section called “跨叢集搜尋”。

變更	版本日期	Description
新功能	2025-03-31	Amazon Q Developer 現在已全面推出 Amazon OpenSearch Service。如需相關資訊，請參閱AI 助理。
新功能	2025-02-05	安全聲明標記語言 2.0) (SAML) 到 AWS Identity and Access Management (IAM) 聯合現在可搭配 OpenSearch UI 使用。這可讓您為最終使用者建立身分提供者起始的單一登入 (SSO) 體驗。如需詳細資訊，請參閱the section called “使用 IAM 啟用 SAML 聯合”。
新的整合	2024-12-01	與 Amazon CloudWatch 的零 ETL 整合可簡化日誌資料分析和視覺化，降低技術開銷和營運成本。如需詳細資訊，請參閱AWS 新聞部落格上的 New Amazon CloudWatch 和 Amazon OpenSearch Service 啟動整合式分析體驗。
新的整合	2024-12-01	與 Amazon Security Lake 的零 ETL 整合可讓組織有效率地搜尋、分析並從其安全資料中獲得可行的洞見。如需詳細資訊，請參閱AWS 新聞部落格上的簡介 Amazon OpenSearch Service 和 Amazon Security Lake 整合，以簡化安全分析。
初始版本	2024-11-07	OpenSearch UI 的初始公有版本。如需詳細資訊，請參閱AWS 大數據部落格上的 Amazon OpenSearch Service 啟動新一代 OpenSearch UI。

Amazon OpenSearch Service 中的 OpenSearch 使用者介面入門

在 Amazon OpenSearch Service 中，應用程式是 OpenSearch 使用者介面 (OpenSearch UI) 的執行個體。每個應用程式都可以與多個資料來源建立關聯，而單一來源可以與多個應用程式建立關聯。您可以使用不同的支援身分驗證選項，為不同的管理員建立多個應用程式。

使用本主題中的資訊，引導您使用 AWS 管理主控台 或 建立 OpenSearch UI 應用程式。 AWS CLI

主題

- 建立 Amazon OpenSearch Service 應用程式所需的許可
- 建立 OpenSearch UI 應用程式
- 管理應用程式管理員

建立 Amazon OpenSearch Service 應用程式所需的許可

建立應用程式之前，請確認您已獲得任務的必要許可。如有需要，請聯絡帳戶管理員尋求協助。

一般許可

若要在 OpenSearch Service 中使用應用程式，您需要下列政策中顯示的許可。許可的用途如下：

- 建立和管理應用程式需要這五個`es:*Application`許可。
- 需要這三個`es:*Tags`許可才能從應用程式新增、列出和移除標籤。
- 關聯資料來源需要 `aoss:BatchGetCollection`、 `es:DescribeDomain`和 `es:GetDirectQueryDataSource`許可。
- 存取資料來源需要 `es:ESHttp*`、 `aoss:APIAccessAll`和 4 `opensearch:*DirectQuery*`許可。
- 為 Amazon OpenSearch Service `iam:CreateServiceLinkedRole`提供在您的帳戶中建立服務連結角色 (SLR) 的許可。使用此角色，可讓 OpenSearch UI 應用程式在您的帳戶中發佈 Amazon CloudWatch 指標。如需詳細資訊，請參閱 使用服務連結角色來建立 VPC 網域和直接查詢資料來源主題中的 the section called “許可”。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "es:CreateApplication",
                "es:DeleteApplication",
                "es:GetApplication",
                "es:ListApplications",
                "es:UpdateApplication",
                "es:AddTags",
                "es:ListTags",
                "es:RemoveTags",
                "aoss:APIAccessAll",
                "es:ESHttp*",
                "opensearch:StartDirectQuery",
```

```
                "opensearch:GetDirectQuery",
                "opensearch:CancelDirectQuery",
                "opensearch:GetDirectQueryResult",
                "aoss:BatchGetCollection",
                "aoss:ListCollections",
                "es:DescribeDomain",
                "es:DescribeDomains",
                "es:ListDomainNames",
                "es:GetDirectQueryDataSource",
                "es:ListDirectQueryDataSources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService"
        }
    ]
}
```

建立使用 IAM Identity Center 身分驗證之應用程式的許可 （選用）

根據預設，儀表板應用程式會使用 AWS Identity and Access Management (IAM) 進行身分驗證，以管理 AWS 資源使用者的許可。不過，您可以選擇使用 IAM Identity Center 提供單一登入體驗，這可讓您使用現有的身分提供者來登入 OpenSearch UI 應用程式。在此情況下，您將在本主題稍後的程序中選取使用 IAM Identity Center 進行身分驗證選項，然後授予 IAM Identity Center 使用者存取 OpenSearch UI 應用程式所需的許可。）

若要建立使用 IAM Identity Center 身分驗證的應用程式，您需要下列許可。使用您的資訊取代*#####*。如有需要，請聯絡帳戶管理員尋求協助。

JSON

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

```
            "Effect": "Allow",
            "Action": [
                "es:CreateApplication",
                "es:DeleteApplication",
                "es:GetApplication",
                "es:ListApplications",
                "es:UpdateApplication",
                "es:AddTags",
                "es:ListTags",
                "es:RemoveTags",
                "aoss:BatchGetCollection",
                "aoss:ListCollections",
                "es:DescribeDomain",
                "es:DescribeDomains",
                "es:ListDomainNames",
                "es:GetDirectQueryDataSource",
                "es:ListDirectQueryDataSources",
                "sso:CreateApplication",
                "sso:DeleteApplication",
                "sso:PutApplicationGrant",
                "sso:PutApplicationAccessScope",
                "sso:PutApplicationAuthenticationMethod",
                "sso:ListInstances",
                "sso:DescribeApplicationAssignment",
                "sso:DescribeApplication",
                "sso:CreateApplicationAssignment",
                "sso:ListApplicationAssignments",
                "sso:DeleteApplicationAssignment",
                "sso-directory:SearchGroups",
                "sso-directory:SearchUsers",
                "sso:ListDirectoryAssociations",
                "identitystore:DescribeUser",
                "identitystore:DescribeGroup",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SLRPermission",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService"
        },
```

```
        {
            "Sid": "PassRolePermission",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/iam-role-for-identity-
center"
        }
    ]
}
```

建立 OpenSearch UI 應用程式

使用以下其中一個程序建立應用程式，指定 和應用程式名稱、身分驗證方法和管理員。

主題

- 在主控台中建立使用 IAM 身分驗證的 OpenSearch UI 應用程式
- 建立在主控台中使用 AWS IAM Identity Center 身分驗證的 OpenSearch UI 應用程式
- 建立使用 身分 AWS IAM Identity Center 驗證的 OpenSearch UI 應用程式 AWS CLI

在主控台中建立使用 IAM 身分驗證的 OpenSearch UI 應用程式

在主控台中建立使用 IAM 身分驗證的 OpenSearch UI 應用程式

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇 OpenSearch UI （儀表板）。
3. 選擇建立應用程式。
4. 在應用程式名稱中，輸入應用程式的名稱。
5. 請勿選取使用 IAM Identity Center 進行身分驗證核取方塊。如需透過 驗證建立應用程式的相關資訊 AWS IAM Identity Center，請參閱本主題the section called “建立在主控台中使用 AWS IAM Identity Center 身分驗證的 OpenSearch UI 應用程式”稍後的 。
6. （選用）您會自動新增為所建立應用程式的管理員。在 OpenSearch 應用程式管理員管理區域中，您可以將管理員許可授予其他使用者。

> **Note**
>
> OpenSearch UI 應用程式管理員角色會授予編輯和刪除 OpenSearch UI 應用程式的權限。應用程式管理員也可以在 OpenSearch UI 應用程式中建立、編輯和刪除工作區。

若要將管理員許可授予其他使用者，請選擇下列其中一項：

- 授予管理員對特定使用者的許可 (在 OpenSearch 應用程式管理員) 欄位中，在屬性快顯清單中，選取 IAM 使用者或

  AWS IAM Identity Center 使用者，然後選擇要授予管理員許可的個別使用者。
- 將管理員許可授予所有使用者 – 您組織或帳戶中的所有使用者都會獲得管理員許可。

7. （選用）設定加密設定。根據預設，OpenSearch UI 中繼資料會使用 AWS 擁有的金鑰加密。若要使用您自己的客戶受管金鑰 (CMK) 進行加密，請參閱 the section called “加密和客戶受管金鑰”。
8. （選用）在標籤區域中，將一或多個標籤索引鍵名稱/值對套用至應用程式。

   標籤是您指派給資源的選用性中繼資料。標籤允許您以不同的方式 (例如用途、擁有者或環境) 將資源分類。
9. 選擇建立。

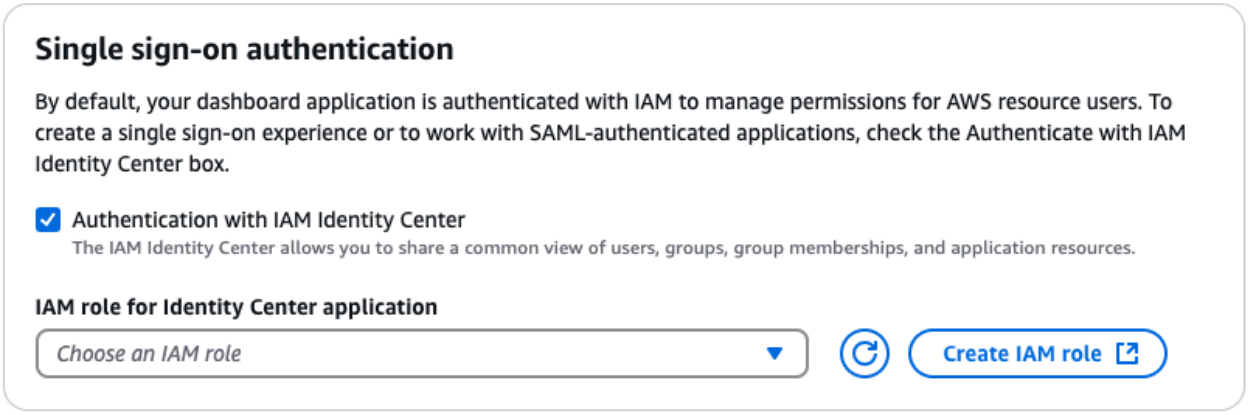
建立在主控台中使用 AWS IAM Identity Center 身分驗證的 OpenSearch UI 應用程式

若要建立使用 AWS IAM Identity Center 身分驗證的 OpenSearch UI 應用程式，您必須擁有 中本主題先前所述的 IAM 許可the section called “建立使用 IAM Identity Center 身分驗證之應用程式的許可（選用）”。

在主控台中建立使用 AWS IAM Identity Center 身分驗證的 OpenSearch UI 應用程式

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇 OpenSearch UI （儀表板）。
3. 選擇建立應用程式。
4. 在應用程式名稱中，輸入應用程式的名稱。
5. （選用）若要為您的組織或帳戶啟用單一登入，請執行下列動作：

a. 選取使用 IAM Identity Center 進行身分驗證核取方塊，如下圖所示：



b. 執行以下任意一項：

- 在 Identity Center 的 IAM 角色應用程式清單中，選擇現有的 IAM 角色，為 IAM Identity Center 提供存取 OpenSearch UI 和相關聯資料來源所需的許可。如需角色必須擁有的許可，請參閱下一個項目符號中的政策。
- 建立具有必要許可的新角色。使用 IAM 使用者指南中的下列程序搭配指定的選項來建立新的角色，以及必要的許可政策和信任政策。
  - 程序：建立 IAM 政策（主控台）

    當您遵循此程序中的步驟時，請將下列政策貼入政策編輯器 JSON 欄位：

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "IdentityStoreOpenSearchDomainConnectivity",
            "Effect": "Allow",
            "Action": [
                "identitystore:DescribeUser",
                "identitystore:ListGroupMembershipsForMember",
                "identitystore:DescribeGroup"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledViaLast": "es.amazonaws.com"
```

```
                }
            }
        },
        {
            "Sid": "OpenSearchDomain",
            "Effect": "Allow",
            "Action": [
                "es:ESHttp*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "OpenSearchServerless",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll"
            ],
            "Resource": "*"
        }
    ]
}
```

- 程序：使用自訂信任政策建立角色

  當您遵循此程序中的步驟時，請將自訂信任政策方塊中的預留位置 JSON 取代為下列項目：

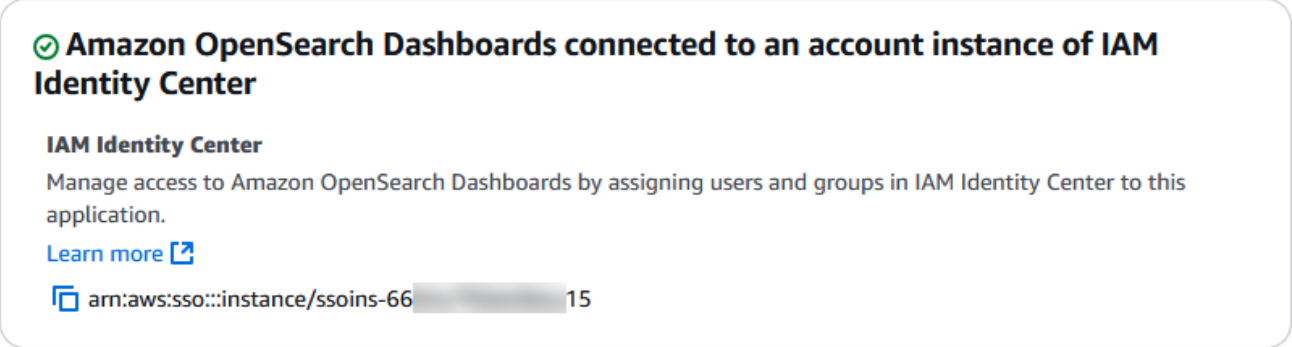
  > **Tip**
  >
  > 如果您要將信任政策新增至現有角色，請在角色的信任關係索引標籤上新增政策。

  JSON

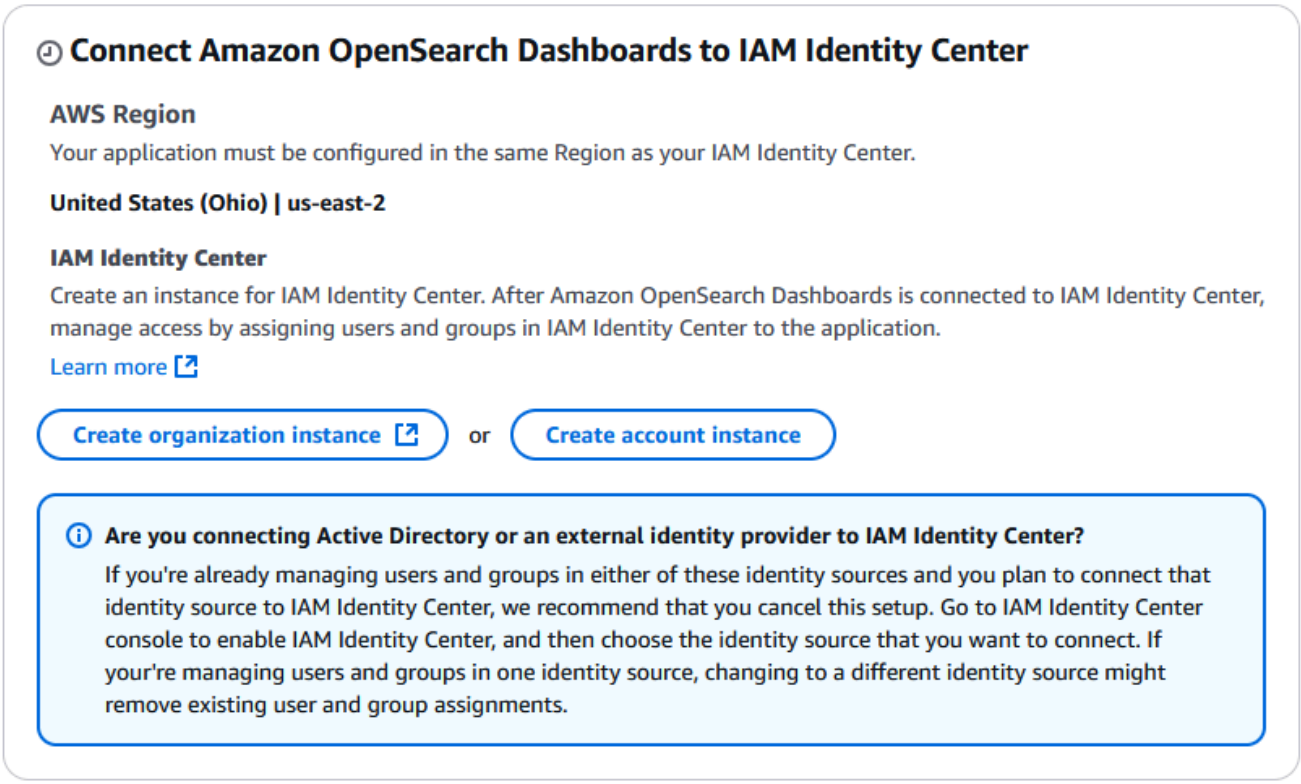
```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "Service":
 "application.opensearchservice.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service":
 "application.opensearchservice.amazonaws.com"
            },
            "Action": "sts:SetContext",
            "Condition": {
                "ForAllValues:ArnEquals": {
                "sts:RequestContextProviders":
 "arn:aws:iam::123456789012:oidc-provider/portal.sso.us-
east-1.amazonaws.com/apl/application-id"
                }
            }
        }
    ]
}
```

c. 如果您的組織或帳戶中已建立 IAM Identity Center 執行個體，主控台會報告 Amazon OpenSearch Dashboards 已連線至 IAM Identity Center 的組織執行個體，如下圖所示。



如果您的組織或帳戶中尚無法使用 IAM Identity Center，則您或具有必要許可的管理員可以建立組織執行個體或帳戶執行個體。將 Amazon OpenSearch Dashboards 連線至 IAM Identity Center 區域為兩者提供選項，如下圖所示：

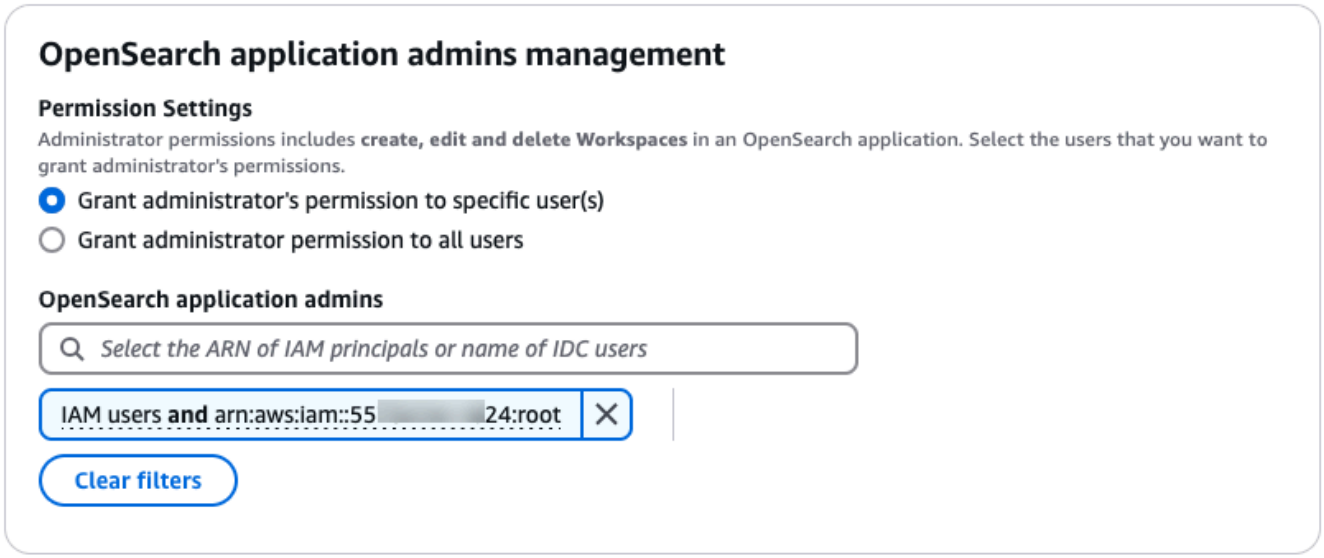


在這種情況下，您可以在 IAM Identity Center 中建立帳戶執行個體進行測試，或請求管理員在 IAM Identity Center 中建立組織執行個體。如需詳細資訊，請參閱《AWS IAM Identity Center 使用者指南》中的以下主題：

> **Note**
>
> 如果您想要在 AWS 區域 與 IAM Identity Center 組織執行個體不同的 中建立 OpenSearch UI 應用程式，請參閱跨多個 使用 IAM Identity Center AWS 區域。

- IAM Identity Center 的組織執行個體
- IAM Identity Center 的帳戶執行個體。
- 啟用 AWS IAM Identity Center

6. （選用）您會自動新增為所建立應用程式的管理員。在 OpenSearch 應用程式管理員管理區域中，您可以將管理員許可授予其他使用者，如下圖所示：



> **Note**
>
> OpenSearch UI 應用程式管理員角色會授予編輯和刪除 OpenSearch UI 應用程式的權限。應用程式管理員也可以在 OpenSearch UI 應用程式中建立、編輯和刪除工作區。

若要將管理員許可授予其他使用者，請選擇下列其中一項：

- 授予管理員對特定使用者的許可 (在 OpenSearch 應用程式管理員欄位中，在屬性快顯清單中，選取 IAM 使用者或

  AWS IAM Identity Center 使用者，然後選擇要授予管理員許可的個別使用者。
- 將管理員許可授予所有使用者 – 您組織或帳戶中的所有使用者都會獲得管理員許可。

7. （選用）在標籤區域中，將一或多個標籤索引鍵名稱/值對套用至應用程式。

   標籤是您指派給資源的選用性中繼資料。標籤允許您以不同的方式 (例如用途、擁有者或環境) 將資源分類。
8. 選擇建立。

建立使用 身分 AWS IAM Identity Center 驗證的 OpenSearch UI 應用程式 AWS CLI

若要建立使用 身分 AWS IAM Identity Center 驗證的 OpenSearch UI 應用程式 AWS CLI，請使用 create-application 命令搭配下列選項：

- `--name` – 應用程式的名稱。
- `--iam-identity-center-options` – （選用） OpenSearch 將用於身分驗證和存取控制的 IAM Identity Center 執行個體和 IAM 角色。

使用您的資訊取代*######*。

```
aws opensearch create-application \
    --name application-name \
    --iam-identity-center-options "
          {
          \"enabled\":true,
          \"iamIdentityCenterInstanceArn\":\"arn:aws:sso:::instance/sso-instance\",
          \"iamRoleForIdentityCenterApplicationArn\":\"arn:aws:iam::account-
id:role/role-name\"
          }
    "
```

管理應用程式管理員

OpenSearch UI 應用程式管理員是已定義的角色，具有編輯和刪除 OpenSearch UI 應用程式的權限。

根據預設，身為 OpenSearch UI 應用程式的建立者，您是 OpenSearch UI 應用程式的第一個管理員。

使用主控台管理 OpenSearch UI 管理員

您可以在應用程式建立工作流程期間或在應用程式建立之後的編輯頁面中 AWS 管理主控台，將其他管理員新增至 中的 OpenSearch UI 應用程式。

OpenSearch UI 應用程式管理員角色會授予編輯和刪除 OpenSearch UI 應用程式的權限。應用程式管理員也可以在 OpenSearch UI 應用程式中建立、編輯和刪除工作區。

在應用程式詳細資訊頁面上，您可以搜尋 IAM 主體的 Amazon Resource Name (ARN)，或搜尋 IAM Identity Center 使用者的名稱。

使用主控台管理 OpenSearch UI 管理員

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇 OpenSearch UI （儀表板）。
3. 在 OpenSearch 應用程式區域中，選擇現有應用程式的名稱。

4. 選擇 Edit (編輯)
5. 若要將管理員許可授予其他使用者，請選擇下列其中一項：

   - 授予管理員對特定使用者的許可 (在 OpenSearch 應用程式管理員) 欄位中，在屬性快顯清單中，選取 IAM 使用者或

     AWS IAM Identity Center 使用者，然後選擇要授予管理員許可的個別使用者。
   - 將管理員許可授予所有使用者 – 您組織或帳戶中的所有使用者都會獲得管理員許可。
6. 選擇更新。

您可以移除其他管理員，但每個 OpenSearch UI 應用程式必須至少保留一個管理員。

使用 管理 OpenSearch UI 管理員 AWS CLI

您可以使用 建立和更新 OpenSearch UI 應用程式管理員 AWS CLI。

使用 建立 OpenSearch UI 管理員 AWS CLI

以下是在建立 OpenSearch UI 應用程式時，將 IAM 主體和 IAM Identity Center 使用者新增為管理員的範例。

範例 1：建立 OpenSearch UI 應用程式，將 IAM 使用者新增為管理員

執行下列命令來建立 OpenSearch UI 應用程式，將 IAM 使用者新增為管理員。使用您的資訊取代*######*。

```
aws opensearch create-application \
    --name application-name \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"arn:aws:iam::account-id:user/user-id\"
        }
    "
```

範例 2：建立 OpenSearch UI 應用程式以啟用 IAM Identity Center，並將 IAM Identity Center 使用者 ID 新增為 OpenSearch UI 應用程式管理員

執行下列命令來建立 OpenSearch UI 應用程式，以啟用 IAM Identity Center，並將 IAM Identity Center 使用者 ID 新增為 OpenSearch UI 應用程式管理員。使用您的資訊取代*######*。

key 指定要設定的組態項目，例如 OpenSearch UI 應用程式的管理員角色。有效值包括 opensearchDashboards.dashboardAdmin.users 與 opensearchDashboards.dashboardAdmin.groups。

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 代表指派給金鑰的值，例如 IAM 使用者的 Amazon Resource Name (ARN)。

```
aws opensearch create-application \
    --name myapplication \
    --iam-identity-center-options "
        {
        \"enabled\":true,
        \"iamIdentityCenterInstanceArn\":\"arn:aws:sso:::instance/ssoins-instance-id\",
        \"iamRoleForIdentityCenterApplicationArn\":\"arn:aws:iam::account-id:role/role-name\"
        }
    " \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"
        }
    "
```

使用 更新 OpenSearch UI 管理員 AWS CLI

以下是更新指派為現有 OpenSearch 應用程式管理員的 IAM 主體和 IAM Identity Center 使用者的範例。

範例 1：將 IAM 使用者新增為現有 OpenSearch 應用程式的管理員

執行下列命令來更新 OpenSearch UI 應用程式，以將 IAM 使用者新增為管理員。使用您的資訊取代*########*。

```
aws opensearch update-application \
    --id myapplication \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"arn:aws:iam::account-id:user/user-id\"
        }
    "
```

範例 2：更新 OpenSearch UI 應用程式，將 IAM Identity Center 使用者 ID 新增為 OpenSearch UI 應用程式管理員

執行下列命令以更新 OpenSearch UI 應用程式，將 IAM Identity Center 使用者 ID 新增為 OpenSearch UI 應用程式管理員。使用您的資訊取代*######*。

`key` 指定要設定的組態項目，例如 OpenSearch UI 應用程式的管理員角色。有效值包括 `opensearchDashboards.dashboardAdmin.users` 與 `opensearchDashboards.dashboardAdmin.groups`。

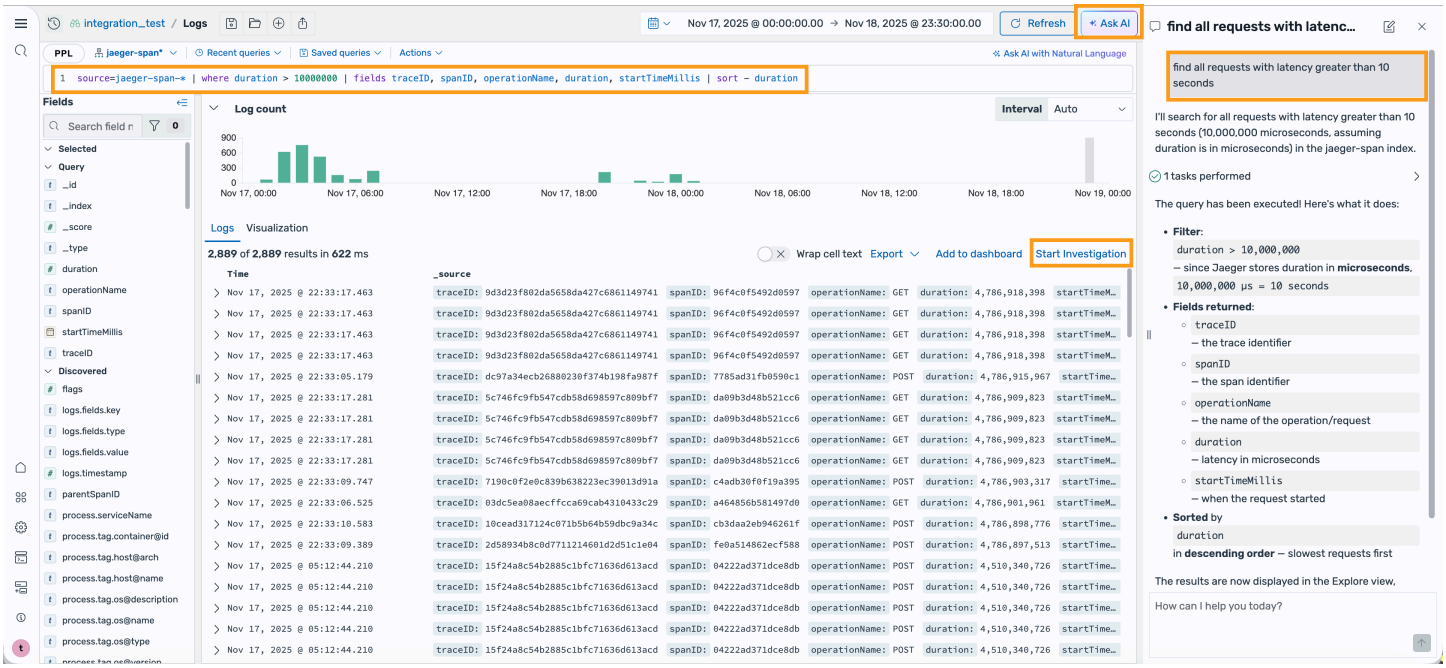
`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` 代表指派給金鑰的值，例如 IAM 使用者的 Amazon Resource Name (ARN)。

```
aws opensearch update-application \
    --id myapplication \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"
        }
    "
```

Amazon OpenSearch Service 中的代理式 AI

OpenSearch UI 包含一組代理式 AI 功能，旨在簡化操作分析並加速工程和支援團隊的事件調查。

在 OpenSearch UI 應用程式的每個頁面上，選擇詢問 AI 按鈕以開啟客服人員聊天。客服人員聊天了解您正在檢視的頁面內容、使用客服人員工具來分析基礎資料、更新查詢以在探索中顯示結果，以及以清晰且準確的方式回答問題。當複雜的根本原因分析需要更深入的調查時，您可以觸發調查代理程式自動規劃、執行和反映多步驟工作流程，提供由資料證據支援的結構化假設。支援這兩個代理程式功能是代理程式記憶體，這是在對話或調查中保留內容的記憶體層，因此您可以繼續在不同的 Web 工作階段中使用相同的一致內容來詢問後續問題。



關鍵功能

- 客服人員聊天 – 客服人員聊天在 OpenSearch UI 每個頁面上的詢問 AI 功能中內嵌，了解您目前的內容並呼叫工具來分析資料，以最佳地回答問題。在探索上，在聊天界面中輸入自然語言來產生和重複管道處理語言 (PPL) 查詢，因此您不需要是 PPL 中的專家，就能取得可行的洞見。您也可以直接從視覺效果開始對話，其中客服人員聊天可以識別圖形中的異常、與基礎資料相互關聯，並產生分析。需要複雜的根本原因分析時，請使用/investigate斜線命令，根據持續對話或新指示啟動調查代理程式。
- 調查代理程式 – 您透過在聊天/investigate中輸入或選擇功能頁面上的開始調查按鈕來啟用的目標驅動型深入研究代理程式。輸入調查目標，例如「尋找此延遲尖峰的根本原因」。Investigation Agent 會使用您的資料和陳述的目標自動規劃、執行查詢和分析，並透過多步驟工作流程進行反映。當調查完成時，通常在幾分鐘內，它會產生按可能性排名的結構化假設，每個假設都由資料證據提供支援。它提供其推理的每個步驟的完整透明度，以便您可以驗證和信任結果。
- 代理程式記憶體 – 一種服務管理的記憶體層，可在對話或調查中保留內容，讓您可以繼續在一致的內容中提出問題或調查。代理程式記憶體可讓代理程式聊天和調查代理程式在整個工作階段中維持連續性。

定價和用量限制

代理式 AI 功能可免費使用。以字符為基礎的用量限制適用於每個帳戶，以防止濫用。

代理式 AI 功能在 Amazon Bedrock 中使用基礎模型。所有 Amazon Bedrock 安全和資料控管組態都適用於代理式 AI 功能。如需詳細資訊，請參閱 Amazon Bedrock 中的資料保護。

安全

代理式 AI 功能會在您現有的安全架構中運作，不會略過或覆寫任何設定的存取控制。當您與客服人員聊天互動或啟動調查客服人員時，AI 功能會使用您目前的 IAM 或 IAM Identity Center 登入資料進行驗證，並繼承所有相關許可。AI 助理只能查詢您已透過現有角色型存取控制 (RBAC) 政策明確存取的資料來源、索引和視覺效果。

如果您的 OpenSearch 網域實作具有欄位層級或文件層級安全性的精細存取控制，則 AI 會遵守這些限制，且無法從超出您許可範圍的限制欄位或文件顯示資料。同樣地，代理聊天所產生的任何 PPL 查詢或在調查期間執行的資料擷取都會使用您的使用者內容執行，以確保所有設定的安全政策都保持完全強制執行。

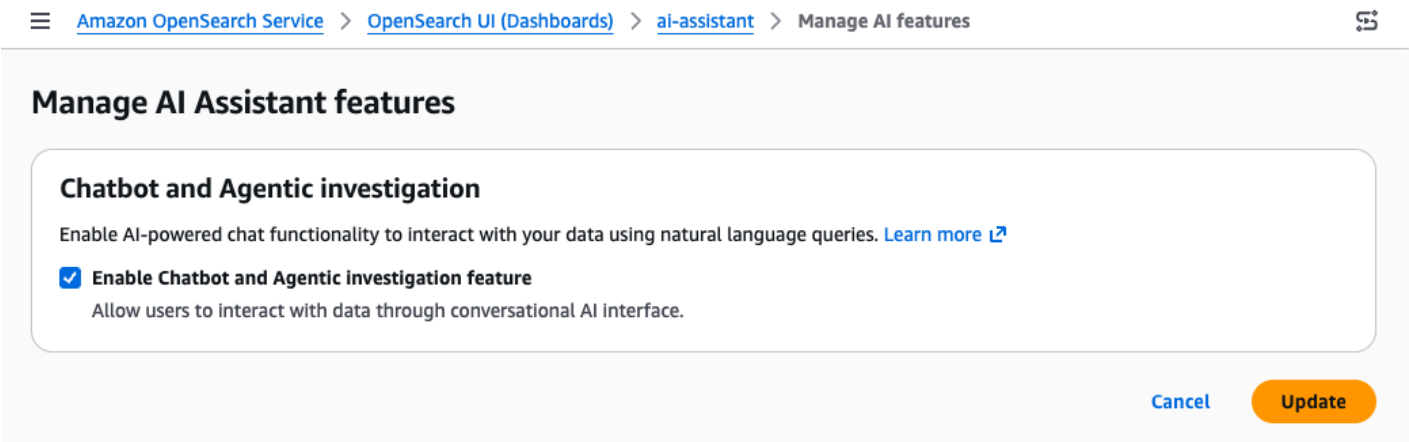
此許可繼承模型可確保啟用 AI 功能不會產生新的安全漏洞，或要求管理員管理個別的 AI 互動許可結構。

啟用和停用代理式 AI 功能

除非您已在任何 OpenSearch 網域中明確停用 AI 功能，否則代理程式 AI 功能會自動為新的 OpenSearch UI 應用程式和現有應用程式啟用。

若要啟用或停用 AI 功能，您可以使用 AWS 管理主控台 或 API：

- 主控台 – 導覽至 OpenSearch UI 應用程式的詳細資訊頁面，並從該處更新 AI 功能。



- API – 使用 RegisterCapability API 來啟用 AI 功能，或使用 DeregisterCapability API 來停用它們。

區域可用性

代理式 AI 功能提供如下 AWS 區域：

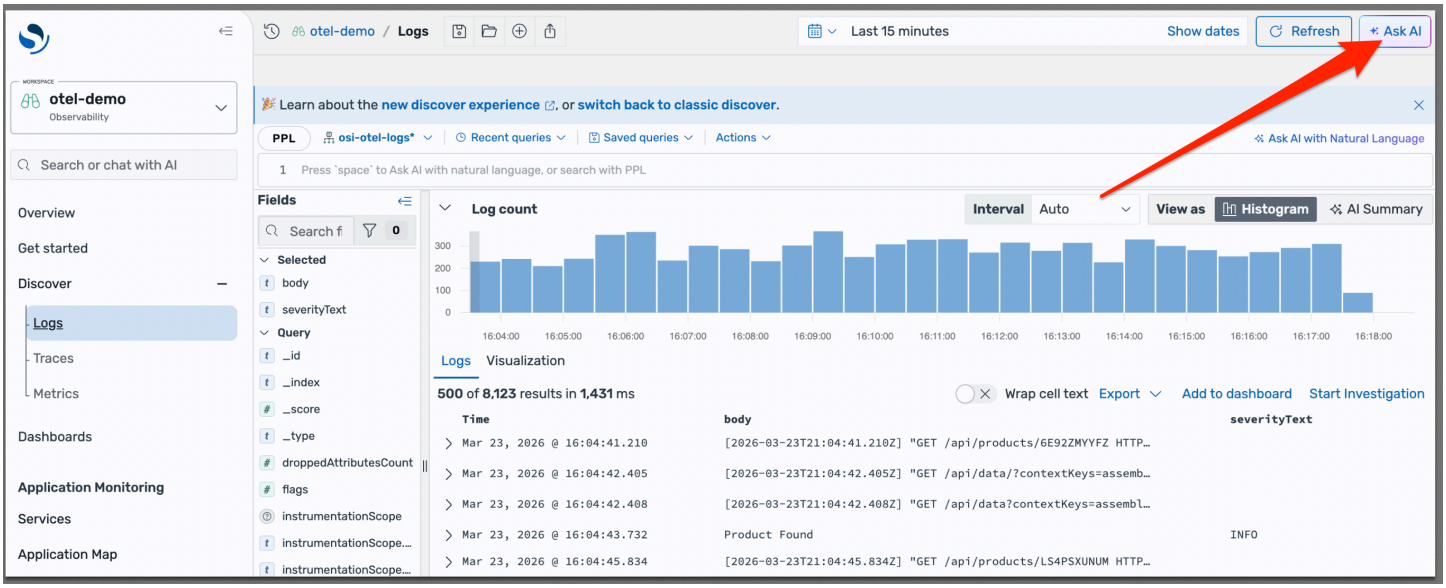
- 亞太區域 (東京) – ap-northeast-1
- 亞太區域 (雪梨) – ap-southeast-2
- 歐洲 (法蘭克福) – eu-central-1
- 歐洲 (斯德哥爾摩) – eu-north-1
- 歐洲 (西班牙) – eu-south-2
- 歐洲 (愛爾蘭) – eu-west-1
- 美國東部 (維吉尼亞北部) – us-east-1
- 美國東部 (俄亥俄) – us-east-2
- 美國西部 (奧勒岡) – us-west-2

主題

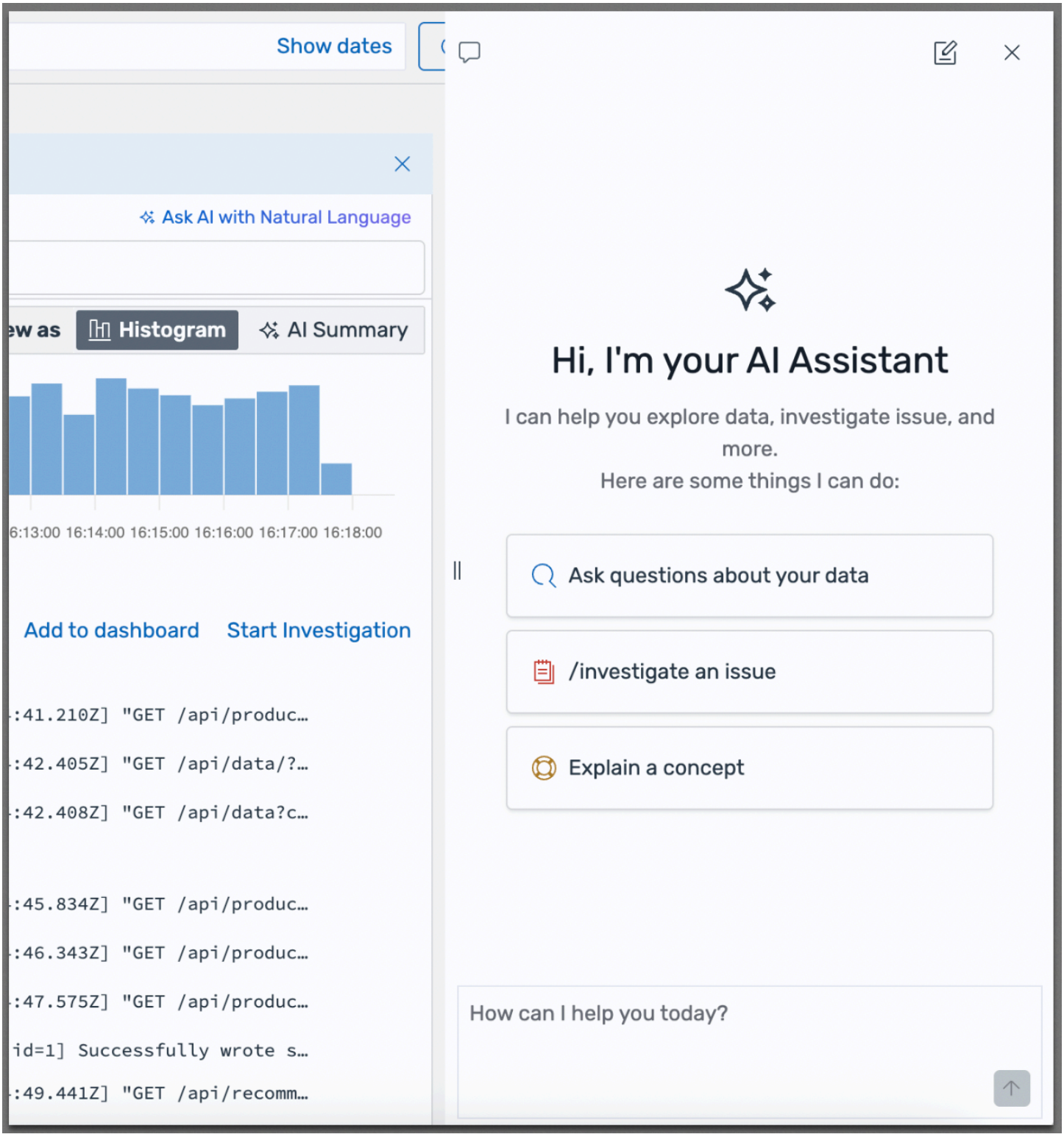
- Amazon OpenSearch Service 中的客服人員聊天
- Amazon OpenSearch Service 中的調查代理程式
- Amazon OpenSearch Service 中的代理程式記憶體

Amazon OpenSearch Service 中的客服人員聊天

Agentic Chat 是內嵌在 OpenSearch UI 每個頁面中的 AI 助理。選擇詢問 AI 按鈕以開啟聊天面板，您可以在其中詢問有關資料的問題、產生查詢，以及啟動調查。客服人員聊天了解您在探索和調查中檢視的頁面內容，並使用客服人員工具來分析基礎資料。



當您開啟聊天面板時，客服人員聊天會提供可協助您入門的選項：詢問有關資料的問題、調查問題或解釋概念。如果您之前已開始對話，當您在頁面之間導覽時，它仍會在聊天面板中顯示，以便您可以從您離開的地方繼續。或者，選擇右上角的新聊天按鈕以開始新的對話。


Show dates
Ask AI with Natural Language
ew as
Histogram
AI Summary
6:13:00 16:14:00 16:15:00 16:16:00 16:17:00 16:18:00
Add to dashboard
Start Investigation
:41.210Z] "GET /api/produc…
:42.405Z] "GET /api/data/?…
:42.408Z] "GET /api/data?c…
:45.834Z] "GET /api/produc…
:46.343Z] "GET /api/produc…
:47.575Z] "GET /api/produc…
id=1] Successfully wrote s…
:49.441Z] "GET /api/recomm…
Hi, I'm your AI Assistant
I can help you explore data, investigate issue, and more.
Here are some things I can do:
Ask questions about your data
/investigate an issue
Explain a concept
How can I help you today?

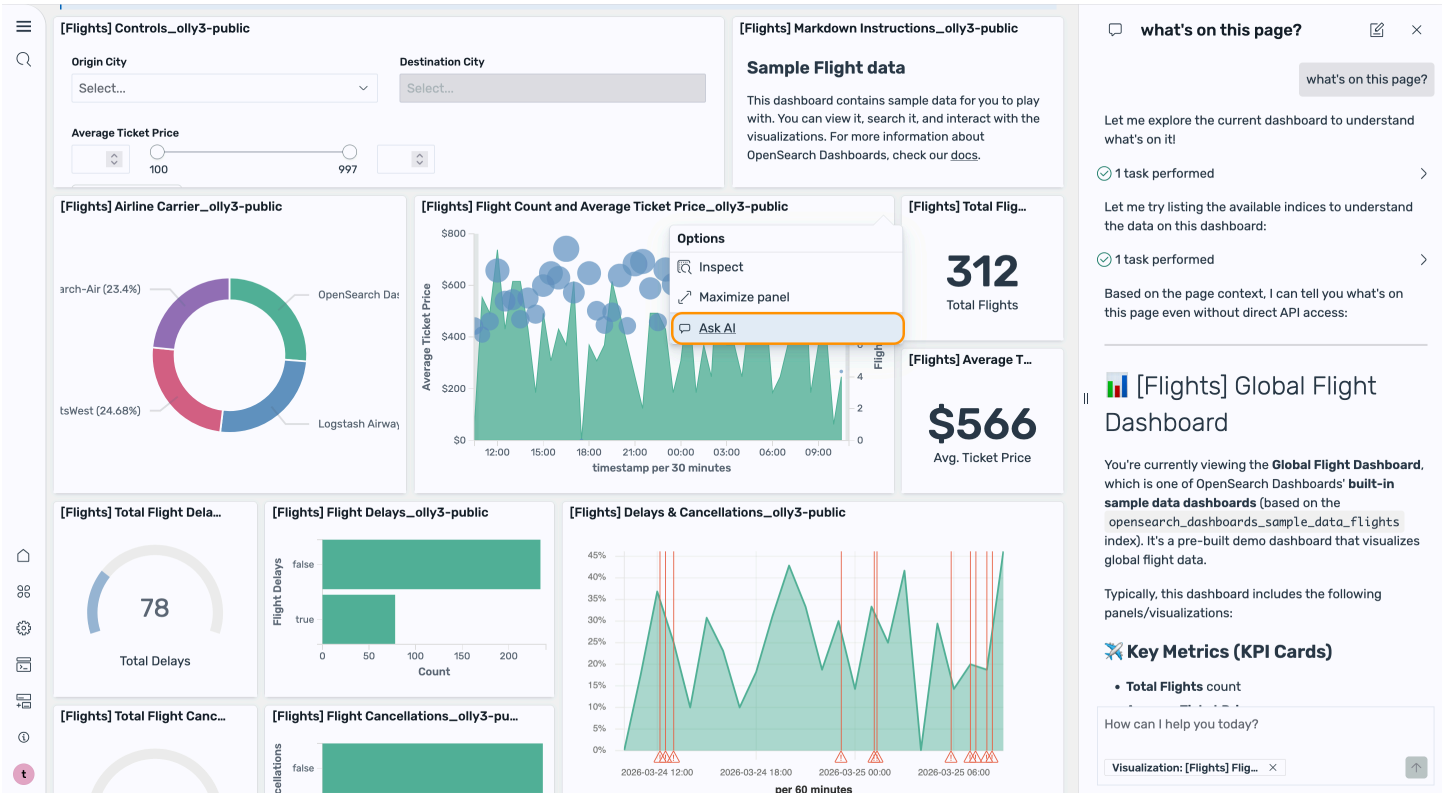
搭配 Discover 使用客服人員聊天

在可觀測性工作區的探索頁面上，您可以在聊天介面中輸入自然語言來產生 PPL 查詢。客服人員聊天會將您的問題翻譯為 PPL、執行查詢，並直接在探索檢視中顯示結果。您不需要是 PPL 的專家，即可從資料中取得可行的洞見。

若要精簡產生的查詢，請以自然語言詢問後續問題，例如「新增狀態碼 500 的篩選條件」。客服人員聊天了解目前查詢的內容，並據此進行修改。您也可以要求 調整彙總、變更時間範圍，或將其他欄位新增至結果。每個反覆運算都會使用新的查詢結果更新探索檢視。

將客服人員聊天與視覺化搭配使用

您可以直接從視覺化開始與客服人員聊天對話。在視覺化面板上開啟內容選單，然後選擇詢問 AI。客服人員聊天會分析視覺化效果、識別圖形中的異常、與基礎資料建立關聯，並產生分析。



從聊天開始調查

需要複雜的根本原因分析時，您可以直接從客服人員聊天啟動調查客服人員。在聊天輸入中使用`/investigate`斜線命令，或在功能頁面上選擇開始調查按鈕。

如需調查代理程式的詳細資訊，請參閱 the section called “調查代理程式”。

支援的工具

客服人員聊天使用下列工具來分析您的資料並回答問題。若要查看最新可用工具up-to-date清單，請在聊天界面中輸入「您可以使用哪些工具」。

前端工具

這些工具會更新 OpenSearch UI：

- `create_investigation` – 建立新的客服人員調查筆記本，其中包含目標、症狀、索引和時間範圍等詳細資訊。
- `execute_ppl_query` – 針對目前的資料集執行 PPL 查詢，並在探索頁面中顯示結果。
- `update_time_range` – 更新目前探索頁面上的全域時間範圍篩選條件（例如，「過去 24 小時」或「上週」)。

後端工具

這些工具會直接與 OpenSearch 資料和 APIs互動：

- `SearchIndexTool` – 使用 DSL 查詢搜尋索引。
- `MsearchTool` – 在單一請求中執行多個搜尋操作。
- `CountTool` – 傳回符合查詢的文件數量。
- `ExplainTool` – 說明文件符合或不符合查詢的原因。
- `ListIndexTool` – 列出具有選用詳細資訊的叢集中的索引。
- `IndexMappingTool` – 擷取索引映射和設定。
- `GetShardsTool` – 取得索引的碎片資訊。
- `ClusterHealthTool` – 傳回叢集運作狀態資訊。
- `LogPatternAnalysisTool` – 分析日誌模式、比較時間範圍或執行追蹤序列分析。
- `MetricChangeAnalysisTool` – 比較兩個時間範圍之間的數值欄位百分位數分佈。
- `DataDistributionTool` – 分析目標時間範圍中的欄位值分佈，選擇性與基準比較。
- `GenericOpenSearchApiTool` – 直接呼叫任何 OpenSearch API 端點的彈性工具。

Amazon OpenSearch Service 中的調查代理程式

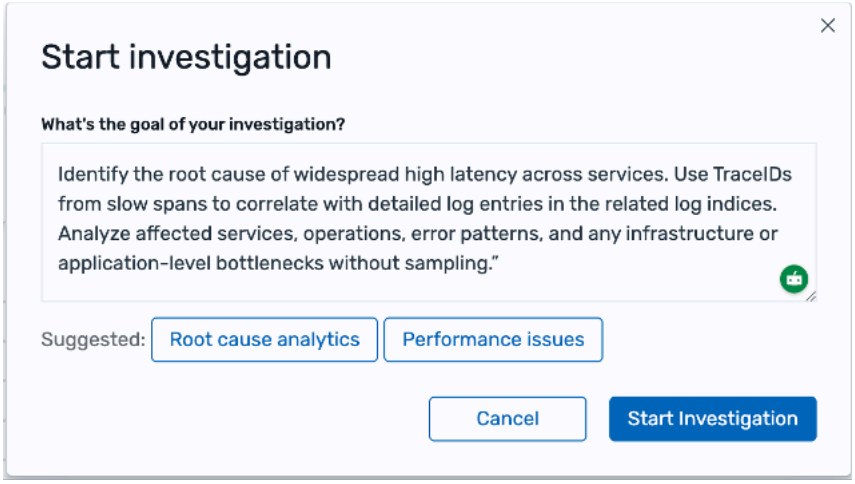
調查代理程式是 OpenSearch UI 中的目標驅動型深入研究代理程式，可代表您自動調查複雜的問題。它計劃使用您的資料和陳述的目標，執行查詢和分析，並透過多步驟工作流程進行反射。當調查完成

時，通常在幾分鐘內，它會產生按可能性排名的結構化假設，每個假設都由資料證據提供支援。它提供其推理的每個步驟的完整透明度，以便您可以驗證和信任結果。

開始調查

您可以透過兩種方式開始調查：

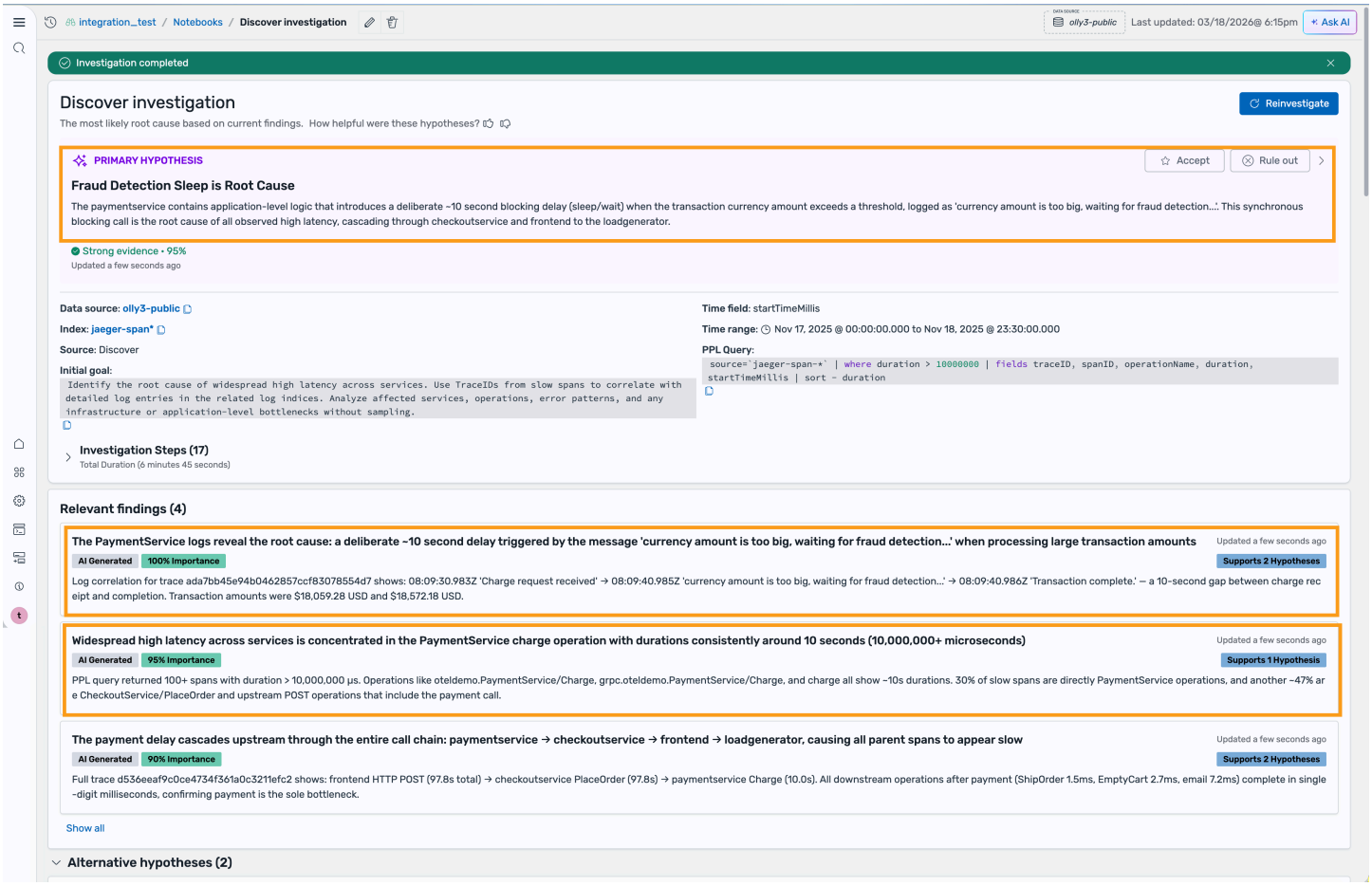
- 在探索、視覺化或其他支援的功能頁面中，選擇開始調查按鈕。出現一個對話方塊，您可以在其中輸入您的調查目標，並從建議的範本中選擇，例如「根本原因分析」或「效能問題」。
- 在客服人員聊天中，輸入包含調查目標的聊天輸入中的`/investigate`斜線命令。



如需客服人員聊天的詳細資訊，請參閱 the section called “客服人員聊天”。

檢閱調查結果

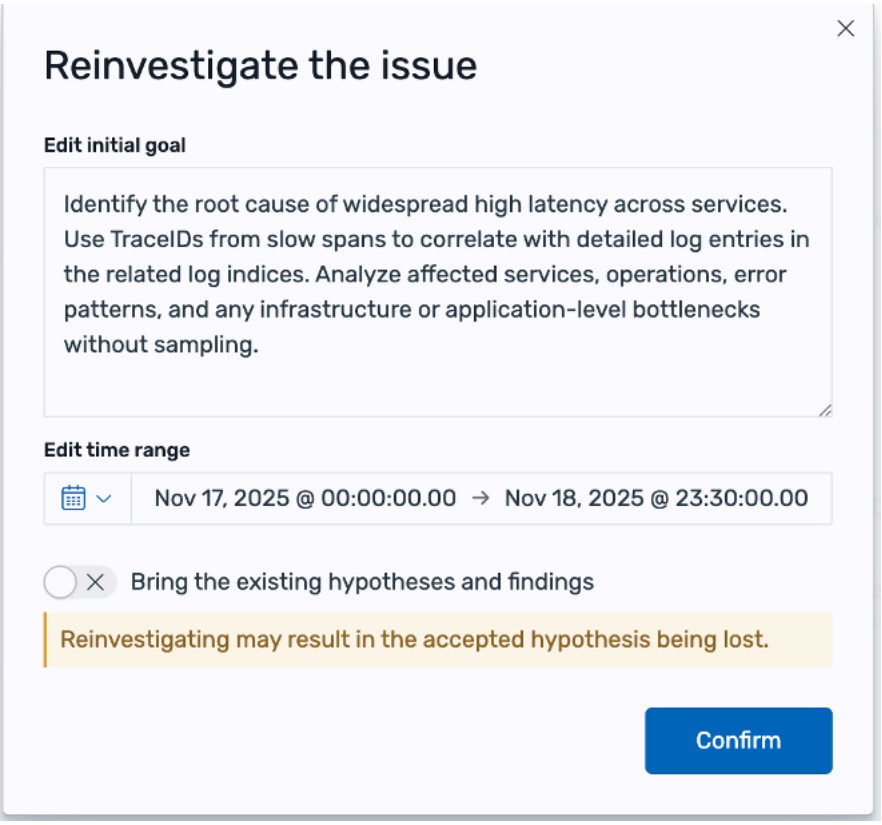
調查完成時，調查客服人員會呈現具有可信度層級和支援證據的主要假設。結果頁面會顯示採取的調查步驟、依重要性排名的相關調查結果，以及替代假設。



您可以檢閱每個假設背後的問題清單，然後選擇接受來確認假設，或選擇排除來拒絕。也有可能性較低的替代假設可供檢閱。如果符合您的評估，您可以選擇任何替代假設做為最終結論。

重新調查

如果調查結果需要更多釐清，或者如果調查客服人員判斷可用資料集無法回答調查問題，您可以使用重新調查選項來調整和重新執行調查。選擇重新調查以編輯初始目標、調整時間範圍，並選擇性地將現有的假設和調查結果帶入新的調查中。



Amazon OpenSearch Service 中的代理程式記憶體

Agentic Memory 是一種服務受管的記憶體層，可支援 Agentic Chat 和 Investigation Agent。它會在您的對話或調查中保留內容，以便您在不同的功能頁面、瀏覽器索引標籤和頁面重新整理之間擁有一致的體驗。代理程式記憶體會自動運作，不需要使用者組態。

代理程式記憶體建置在 OpenSearch 代理程式記憶體架構上。記憶體儲存體會依使用者 ID 隔離以進行隱私權保護。

資料保護

代理程式記憶體可免費使用。存放在代理記憶體中的客戶資料會使用服務受管金鑰加密。如果您為 OpenSearch UI 應用程式啟用客戶受管金鑰 (CMK) 加密，您的記憶體資料將改為使用 CMK 加密。記憶體存放在服務管理的 Amazon OpenSearch Serverless 集合中。

如需 CMK 加密的詳細資訊，請參閱 the section called “加密和客戶受管金鑰”。

限制

代理記憶體無法保留不同對話執行緒的內容。

使用客戶受管金鑰加密 OpenSearch UI 應用程式中繼資料

視覺化資產和組態會儲存為 OpenSearch UI 應用程式的中繼資料。這包括已儲存的查詢、視覺化和儀表板。來自相關聯資料來源的資料不會存放在中繼資料中。如需有關加密資料來源中資料的資訊，請參閱 Amazon OpenSearch Service for OpenSearch 網域中的資料保護和 Amazon OpenSearch Serverless 中的無伺服器集合中的加密。 OpenSearch

您的 OpenSearch UI 中繼資料受到靜態加密保護。這可防止未經授權的存取。加密會使用 AWS Key Management Service (AWS KMS) 來存放和管理加密金鑰。根據預設，OpenSearch UI 中繼資料會使用 AWS 擁有的金鑰加密。

您也可以使用客戶受管金鑰 (CMK) 功能來管理您自己的加密金鑰。這可協助您符合法規和合規要求。若要使用 CMK，您必須建立新的 OpenSearch UI 應用程式，並在建立程序中啟用 CMK。目前不支援將現有的 OpenSearch UI 應用程式從 AWS 擁有的金鑰更新為 CMK。

何時使用客戶受管金鑰：

- 您的組織具有金鑰管理的法規合規要求
- 您需要加密金鑰使用的稽核線索
- 您想要控制金鑰輪換排程
- 您需要與現有的金鑰管理工作流程整合

當您使用客戶受管金鑰時，您可以完全控制金鑰。這包括能夠：

- 建立和維護金鑰政策
- 建立和維護 IAM 政策和授予
- 啟用和停用金鑰
- 輪換金鑰的密碼編譯材料
- 將標籤新增至金鑰
- 建立金鑰別名
- 排程要刪除的金鑰

> **Note**
>
> 客戶受管金鑰必須與 OpenSearch UI 應用程式位於相同的 AWS 區域 中。您無法使用不同區域的金鑰。

主題

- 使用客戶受管金鑰的先決條件
- 使用 主控台建立具有客戶受管金鑰加密的應用程式
- 使用 建立具有客戶受管金鑰加密的應用程式 AWS CLI
- 監控客戶受管金鑰用量
- 更新加密設定

使用客戶受管金鑰的先決條件

您必須先在其中建立對稱加密金鑰，才能使用客戶受管金鑰來加密 OpenSearch UI 應用程式中繼資料 AWS KMS。如需建立金鑰的指示，請參閱《 AWS KMS 開發人員指南》中的建立金鑰。

客戶受管金鑰的金鑰政策必須授予 OpenSearch UI 許可才能使用金鑰。使用下列金鑰政策，將####取代為您自己的資訊：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOpenSearchUIToUseKey",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "application.opensearchservice.amazonaws.com"
                ]
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:DescribeKey"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "AllowKeyAdministration",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account-id:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

政策包含兩個陳述式：

- 第一個陳述式允許 OpenSearch UI 使用 金鑰進行加密操作。
- 第二個陳述式可讓 中的使用者 AWS 帳戶 管理金鑰。這包括更新金鑰政策、啟用或停用金鑰，以及排程刪除金鑰的許可。您可以將根主體取代為特定的 IAM 使用者或角色，以進一步限制這些許可。

如需金鑰政策的詳細資訊，請參閱《 AWS KMS 開發人員指南》中的在 中使用金鑰政策 AWS KMS。

使用 主控台建立具有客戶受管金鑰加密的應用程式

當您在主控台中建立 OpenSearch UI 應用程式時，您可以指定客戶受管金鑰來加密應用程式的中繼資料。

使用主控台建立具有客戶受管金鑰加密的 OpenSearch UI 應用程式

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇 OpenSearch UI （儀表板）。
3. 選擇建立應用程式。
4. 在應用程式名稱中，輸入應用程式的名稱。
5. 視需要設定身分驗證和管理員設定。如需詳細資訊，請參閱the section called “開始使用”。
6. 在加密區段中，針對靜態加密，選擇使用客戶受管金鑰。
7. 從清單中選擇現有的客戶受管金鑰，或選擇建立金鑰以建立新的金鑰 AWS KMS。

> **Note**
>
> 金鑰必須與您建立的應用程式 AWS 區域 位於相同的 中。

8. （選用）將標籤新增至應用程式。
9. 選擇建立。

使用 建立具有客戶受管金鑰加密的應用程式 AWS CLI

若要使用 建立具有客戶受管金鑰加密的 OpenSearch UI 應用程式 AWS CLI，請使用 create-application 命令搭配 `--kms-key-arn` 參數。

使用您的資訊取代*######*。

```
aws opensearch create-application \
    --name my-application \
    --kms-key-arn arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012
```

如果您未指定 `--kms-key-arn` 參數，OpenSearch 會使用 AWS受管金鑰來加密應用程式的中繼資料。

監控客戶受管金鑰用量

當您搭配 OpenSearch UI 應用程式使用客戶受管金鑰時， 會在 AWS CloudTrail 日誌中 AWS KMS 記錄金鑰的每次使用。您可以使用這些日誌來監控金鑰的使用方式和時間。日誌會顯示哪些使用者或服務存取了 金鑰。

AWS AWS KMS 每年會自動輪換客戶受管金鑰。您也可以視需要手動輪換金鑰。如需金鑰輪換的詳細資訊，請參閱《 AWS KMS 開發人員指南》中的輪換 KMS 金鑰。

如需監控金鑰用量的詳細資訊，請參閱《 AWS KMS 開發人員指南》中的使用 記錄 AWS KMS API 呼叫 AWS CloudTrail。

> **Note**
>
> 使用客戶受管金鑰會產生 AWS KMS 費用。費用是根據儲存的 API 請求和金鑰數量而定。如需定價詳細資訊，請參閱 AWS Key Management Service 定價。

更新加密設定

建立 OpenSearch UI 應用程式後，您無法變更其加密設定。如果您需要使用不同的客戶受管金鑰，您必須建立新的應用程式。如果您需要在 AWS受管金鑰和客戶受管金鑰之間切換，您還必須使用所需的加密設定建立新的應用程式。

> **Important**
>
> 在您停用或刪除客戶受管金鑰之前，請考慮下列事項：
>
> - 如果您停用金鑰，應用程式將無法存取其加密的中繼資料。您必須重新啟用相同的金鑰才能還原存取權。
> - 如果您刪除金鑰，應用程式儲存的物件將永久無法存取。這包括查詢、視覺化和儀表板。無法刪除已刪除的金鑰。
> - 我們建議您在對金鑰狀態進行任何變更之前，先記錄您的金鑰 ARN。

後續步驟

為您的應用程式設定 CMK 加密之後，您可以：

- 將資料來源與您的應用程式建立關聯。如需詳細資訊，請參閱the section called “管理資料來源關聯和 VPC 存取許可”。
- 為您的團隊建立工作區。如需詳細資訊，請參閱the section called “使用 Amazon OpenSearch Service 工作區”。
- 設定 AWS CloudTrail 監控金鑰用量。如需詳細資訊，請參閱the section called “監控客戶受管金鑰用量”。

使用 啟用 SAML 聯合 AWS Identity and Access Management

OpenSearch UI 支援安全性聲明標記語言 2.0 (SAML)，這是許多身分提供者使用的開放標準。這會啟用與 AWS Identity and Access Management (IAM) 的聯合身分。透過此支援，您帳戶或組織中的使用者可以透過擔任 IAM 角色直接存取 OpenSearch UI。您可以為您的最終使用者建立身分提供者起始 (IdP) 單一登入體驗，他們可以在外部身分提供者中進行身分驗證，並直接路由到 OpenSearch UI 中定義的頁面。您也可以透過設定最終使用者或群組來擔任具有不同許可的不同 IAM 角色，以存取 OpenSearch UI 和相關聯的資料來源，來實作精細存取控制。

本主題提供設定搭配 OpenSearch UI 使用 SAML 的step-by-step說明。在這些程序中，我們使用設定 Okta 身分和存取管理應用程式的步驟做為範例。其他身分提供者的組態步驟類似，例如 Azure Active Directory 和 Ping。

主題

- 步驟 1：設定身分提供者應用程式 (Okta)
- 步驟 2：設定 Okta 的 AWS 組態
- 步驟 3：在 IAM 中建立 Amazon OpenSearch Service 存取政策
- 步驟 4：使用 SAML 驗證身分提供者起始的單一登入體驗
- 步驟 5：設定以 SAML 屬性為基礎的精細存取控制

步驟 1：設定身分提供者應用程式 (Okta)

若要搭配 OpenSearch UI 使用 SAML，第一步是設定您的身分提供者。

任務 1：建立 Okta 使用者

1. 以具有管理權限的使用者身分登入您的 Okta 組織，網址為 https：//https://login.okta.com/。
2. 在管理員主控台的導覽窗格中的目錄下，選擇人物。
3. 選擇新增人員。
4. 針對名字，輸入使用者的名字。
5. 針對姓氏，輸入使用者的姓氏。
6. 針對使用者名稱，以電子郵件格式輸入使用者的使用者名稱。
7. 選擇我將會設定密碼並輸入密碼
8. （選用）如果您不希望使用者在第一次登入時變更密碼，請清除使用者必須在第一次登入時變更密碼方塊。
9. 選擇儲存。

任務 2：建立和指派群組

1. 以具有管理權限的使用者身分登入您的 Okta 組織，網址為 https：//https://login.okta.com/。
2. 在管理員主控台的導覽窗格中的目錄下，選擇群組。
3. 選擇 Add group (新增群組)。
4. 輸入群組名稱，然後選擇儲存。

5. 選擇新建立的群組，然後選擇指派人員。
6. 選擇加號 (+)，然後選擇完成。
7. （選用）重複步驟 1–6 以新增更多群組。

任務 3：建立 Okta 應用程式

1. 以具有管理權限的使用者身分登入您的 Okta 組織，網址為 https：//https://login.okta.com/。
2. 在管理員主控台的導覽窗格中的應用程式下，選擇應用程式。
3. 選擇 Create App Integration (建立應用程式整合)。
4. 選擇 SAML 2.0 作為登入方法，然後選擇下一步。
5. 輸入應用程式整合的名稱（例如 **OpenSearch_UI**），然後選擇下一步。
6. 在應用程式中輸入下列值；您不需要變更其他值：
   a. 1. 在單一登入 URL 中，輸入 **https://signin.aws.amazon.com/saml** 表示商業 AWS 區域，或輸入您區域特定的 URL。
   b. 2. 針對對象 URI (SP 實體 ID)，輸入 **urn:amazon:webservices**。
   c. 3. 針對名稱 ID 格式，輸入 **EmailAddress**。
7. 選擇下一步。
8. 選擇我是新增內部應用程式的 Okta 客戶，然後選擇這是我們建立的內部應用程式。
9. 選擇完成。
10. 選擇指派，然後選擇指派。
11. 選擇指派給群組，然後選取您要新增的群組旁的指派。
12. 選擇完成。

任務 4：設定 Okta 進階組態

建立自訂 SAML 應用程式後，請完成下列步驟：

1. 以具有管理權限的使用者身分登入您的 Okta 組織，網址為 https：//https://login.okta.com/。

   在管理員主控台的一般區域中，選擇 SAML 設定下的編輯。
2. 選擇下一步。
3. 使用格式將預設轉送狀態設定為 OpenSearch UI 端點：

https://*region*.console.aws.amazon.com/aos/home?region=*region*#opensearch/applications/*application-id*/redirectToDashboardURL.

以下是範例：

https://us-east-2.console.aws.amazon.com/aos/home?region=us-east-2#opensearch/applications/abc123def4567EXAMPLE/redirectToDashboardURL

4. 在屬性陳述式（選用）下，新增下列屬性：

   a. 使用角色屬性，以逗號分隔格式提供 IAM 角色和身分提供者。在稍後的步驟中，您將在設定 AWS 組態時使用相同的 IAM 角色和身分提供者。

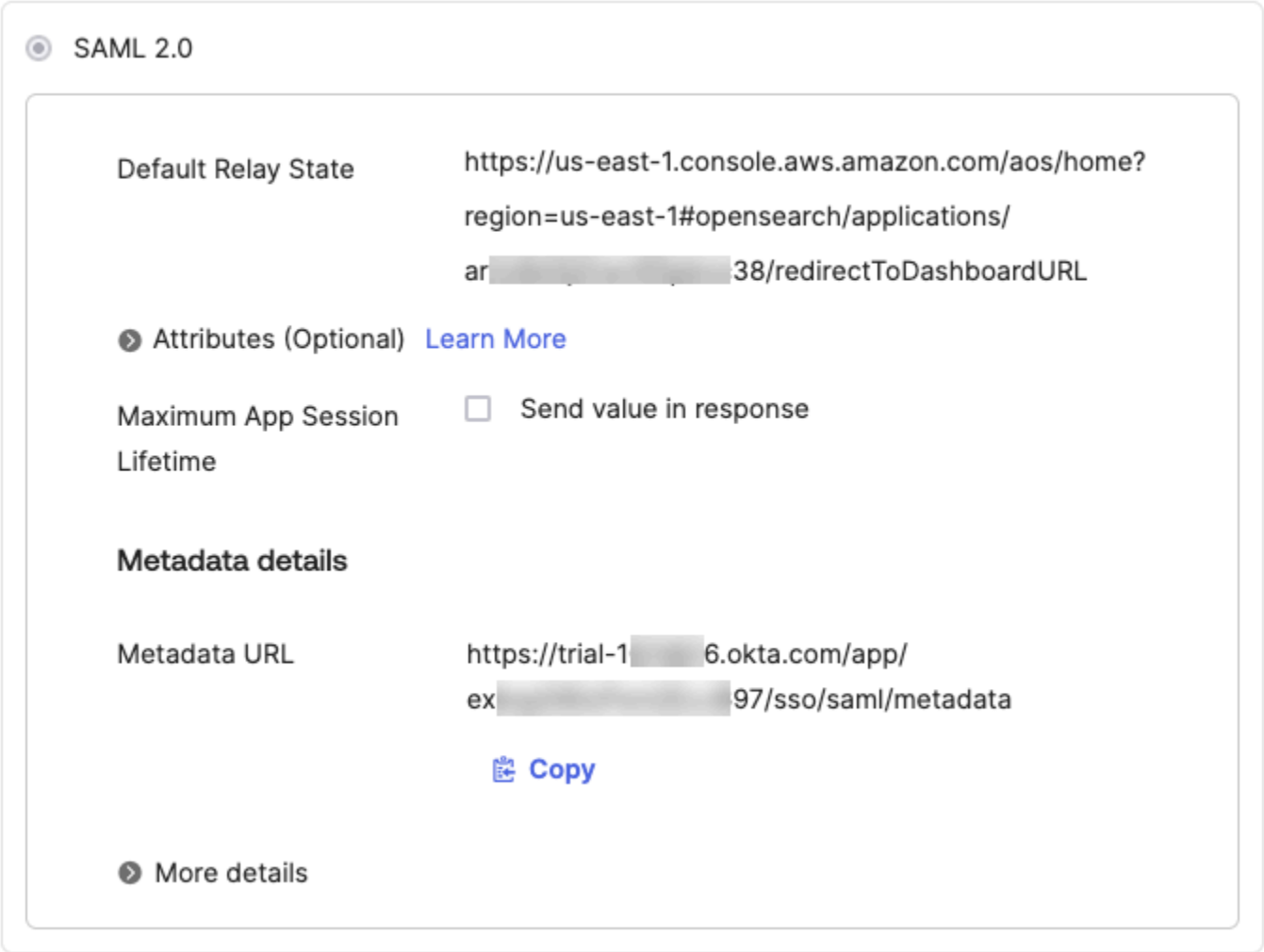
   b. 設定 RoleSessionName 的 user.login。這是做為擔任角色時所發出臨時登入資料的識別符。

   如需參考：

名稱	名稱格式	格式	範例
https://aws.amazon.com/SAML/Attributes/Role	未指定	arn:aws:iam:: *aws-account-id* :role/role-name,arn:aws:iam:: *aws-account-id* :saml-provider/ *provider-name*	arn:aws:iam::111222333444:role/oktarole,arn:aws:iam::111222333444:saml-provider/oktaidp
https://aws.amazon.com/SAML/Attributes/RoleSessionName	未指定	user.login	user.login

5. 新增屬性屬性之後，請選擇下一步，然後選擇完成。

您的屬性格式應類似於下圖所示的屬性。預設轉送狀態值是在您帳戶或組織中最終使用者完成 Okta 的單一登入驗證後定義登陸頁面的 URL。您可以將其設定為 OpenSearch UI 中的任何頁面，然後將該 URL 提供給其預期的最終使用者。



步驟 2：設定 Okta 的 AWS 組態

完成下列任務以設定 Okta 的 AWS 組態。

任務 1：收集 Okta 資訊

在此步驟中，您將需要收集 Okta 資訊，以便稍後進行設定 AWS。

1. 以具有管理權限的使用者身分登入您的 Okta 組織，網址為 https：//https://login.okta.com/。
2. 在登入索引標籤的頁面右下角，選擇檢視 SAML 設定指示。

3. 請記下身分提供者單一登入 URL 的值。您可以在連線至 SQL Workbench/J 等任何第三方 SQL 用戶端時使用此 URL。
4. 在區塊 4 中使用身分提供者中繼資料，然後將中繼資料檔案儲存為 .xml 格式 （ 例如 `metadata.xml`)。

任務 2：建立 IAM 提供者

若要建立 IAM 提供者，請完成下列步驟：

1. 登入 AWS 管理主控台 ，並在 https：//https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格的存取管理 下，選擇身分提供者。
3. 選擇 Add provider (新增提供者)。
4. 針對提供者類型選取 SAML。
5. 針對提供者名稱輸入名稱。
6. 針對中繼資料文件，選擇選擇檔案並上傳您先前下載的中繼資料檔案 (.xml)。
7. 選擇 Add provider (新增提供者)。

任務 3：建立 IAM 角色

若要建立您的 AWS Identity and Access Management 角色，請完成下列步驟：

1. 登入 AWS 管理主控台 ，並在 https：//https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在導覽窗格的存取管理 下，選擇角色。
3. 選擇建立角色。
4. 針對信任的實體類型，選取 SAML 2.0 聯合。
5. 針對以 SAML 2.0 為基礎的提供者，選擇您先前建立的身分提供者。
6. 選取允許程式設計和 AWS 管理主控台 存取。
7. 選擇下一步。
8. 在許可政策清單中，選取授予 OpenSearch Service 許可之政策的核取方塊，例如 AWS ，受管政策 AmazonOpenSearchServiceFullAccess。
9. 選擇下一步。
10. 在檢閱區域中，針對角色名稱輸入角色的名稱，例如 **`oktarole`**。
11. （選用 ） 針對描述，輸入角色用途的簡短描述。

12. 選擇建立角色。
13. 導覽至您剛建立的角色，選擇信任關係索引標籤，然後選擇編輯信任政策。
14. 在編輯陳述式窗格的新增 STS 動作下，選取 TagSession 的方塊。
15. 選擇更新政策。

步驟 3：在 IAM 中建立 Amazon OpenSearch Service 存取政策

了解如何為 OpenSearch 存取控制設定 IAM 角色。透過 IAM 角色，您可以為 Okta 使用者群組實作精細存取控制，以存取 OpenSearch 資源。本主題示範使用兩個範例群組的 IAM 角色型組態。

Sample group: Alice

要求:

```
GET _plugins/_security/api/roles/alice-group
```

結果：

```
{
  "alice-group": {
    "reserved": false,
    "hidden": false,
    "cluster_permissions": [
      "unlimited"
    ],
    "index_permissions": [
      {
        "index_patterns": [
          "alice*"
        ],
        "dls": "",
        "fls": [],
        "masked_fields": [],
        "allowed_actions": [
          "indices_all"
        ]
      }
    ],
    "tenant_permissions": [
      {
```

```
        "tenant_patterns": [
          "global_tenant"
        ],
        "allowed_actions": [
          "kibana_all_write"
        ]
      }
    ],
    "static": false
  }
}
```

Sample group: Bob

要求:

```
GET _plugins/_security/api/roles/bob-group
```

結果：

```
{
  "bob-group": {
    "reserved": false,
    "hidden": false,
    "cluster_permissions": [
      "unlimited"
    ],
    "index_permissions": [
      {
        "index_patterns": [
          "bob*"
        ],
        "dls": "",
        "fls": [],
        "masked_fields": [],
        "allowed_actions": [
          "indices_all"
        ]
      }
    ],
    "tenant_permissions": [
      {
        "tenant_patterns": [
```

```
          "global_tenant"
        ],
        "allowed_actions": [
          "kibana_all_write"
        ]
      }
    ],
    "static": false
  }
}
```

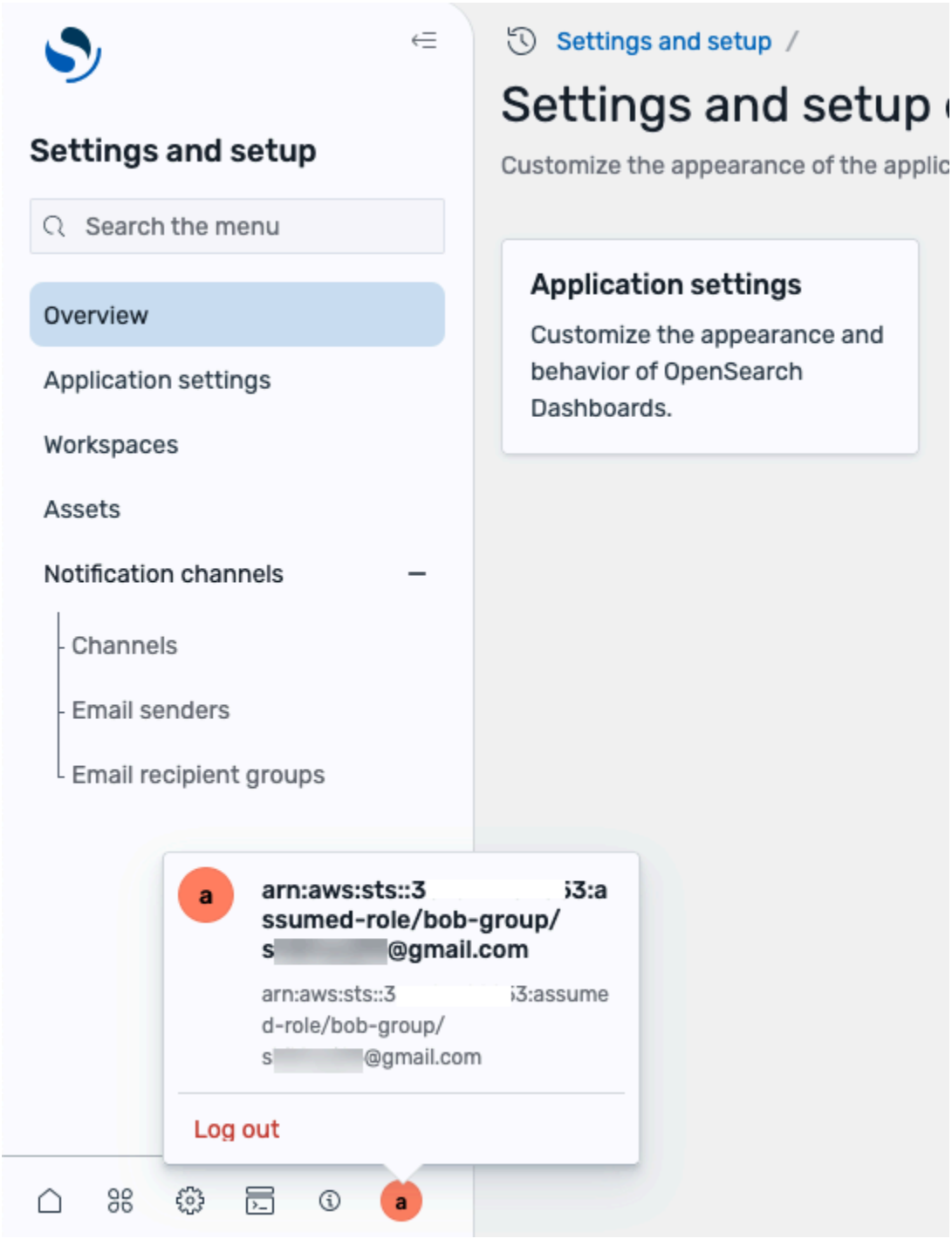
您可以使用後端角色映射將 Amazon OpenSearch Service 網域角色映射至 IAM 角色，如下列範例所示：

```
{
  "bob-group": {
    "hosts": [],
    "users": [],
    "reserved": false,
    "hidden": false,
    "backend_roles": [
      "arn:aws:iam::111222333444:role/bob-group"
    ],
    "and_backend_roles": []
  },
  "alice-group": {
    "hosts": [],
    "users": [],
    "reserved": false,
    "hidden": false,
    "backend_roles": [
      "arn:aws:iam::111222333444:role/alice-group"
    ],
    "and_backend_roles": []
  }
}
```

步驟 4：使用 SAML 驗證身分提供者起始的單一登入體驗

開啟預設轉送狀態的 URL 以開啟 Okta 身分驗證頁面。輸入最終使用者的登入資料。您會自動重新導向至 OpenSearch UI。

您可以選擇導覽面板底部的使用者圖示來檢查目前的登入資料，如下圖所示：



您也可以存取導覽面板底部的開發人員工具，並在主控台中執行查詢，來驗證使用者的精細存取控制許可。以下是範例查詢。

Example 1: Displays information about the current user

要求:

```
GET _plugins/_security/api/account
```

結果：

```
{
  "user_name": "arn:aws:iam::XXXXXXXXXXXX:role/bob-group",
  "is_reserved": false,
  "is_hidden": false,
  "is_internal_user": false,
  "user_requested_tenant": null,
  "backend_roles": [
    "arn:aws:iam::XXXXXXXXXXXX:role/bob-group"
  ],
  "custom_attribute_names": [],
  "tenants": {
    "global_tenant": true,
    "arn:aws:iam::XXXXXXXXXXXX:role/bob-group": true
  },
  "roles": [
    "bob-group"
  ]
}
```

Example 2: Displays actions permitted for a user

要求:

```
GET bob-test/_search
```

結果：

```
{
  "took": 390,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
```

```
      "value": 1,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "bob-test",
        "_id": "ui01N5UBCIHpjO8Jlvfy",
        "_score": 1,
        "_source": {
          "title": "Your Name",
          "year": "2016"
        }
      }
    ]
  }
}
```

Example 3: Displays actions not permitted for a user

要求:

```
GET alice-test
```

結果：

```
{
  "error": {
    "root_cause": [
      {
        "type": "security_exception",
        "reason": "no permissions for [indices:admin/get]
 and User [name=arn:aws:iam::111222333444:role/bob-group,
 backend_roles=[arn:aws:iam::111222333444:role/bob-group], requestedTenant=null]"
      }
    ],
    "type": "security_exception",
    "reason": "no permissions for [indices:admin/get]
 and User [name=arn:aws:iam::111222333444:role/bob-group,
 backend_roles=[arn:aws:iam::111222333444:role/bob-group], requestedTenant=null]"
  },
  "status": 403
}
```

步驟 5：設定以 SAML 屬性為基礎的精細存取控制

透過 Amazon OpenSearch Service，您可以將精細存取控制與 SAML 搭配使用，以動態方式將身分提供者的使用者和群組映射至 OpenSearch 精細存取控制使用者和角色。您可以將這些角色範圍限定在特定 OpenSearch 網域和無伺服器集合，並定義索引層級許可和文件層級安全性。

> **Note**
>
> 如需精細存取控制的詳細資訊，請參閱 the section called “精細定義存取控制”。

主題

- 精細存取控制的 SAML 屬性
- 任務 1：設定 Okta 進行精細存取控制
- 任務 2：在 OpenSearch 網域中設定 SAML
- 任務 3：在 OpenSearch Serverless 集合中設定 SAML

精細存取控制的 SAML 屬性

subjectKey

映射至唯一使用者屬性，例如電子郵件或使用者名稱，以識別要進行身分驗證的使用者。

rolesKey

映射至 IdP 中的群組或角色屬性，以決定角色或授權許可。

任務 1：設定 Okta 進行精細存取控制

設定 Okta 進行精細存取控制

1. 在屬性陳述式區段中為 OpenSearch 使用者主體新增屬性：

   - 名稱：`UserName`
   - 值：`${user-email}`

   此屬性用作 OpenSearch 精細存取控制組態中用於身分驗證的主體金鑰。

2. 在群組屬性陳述式區段中新增角色的群組屬性：

   - 名稱：groups
   - 篩選條件： OpenSearch_xxx

   此屬性用作將群組映射至 OpenSearch 精細存取控制角色以進行授權的角色金鑰。 OpenSearch

任務 2：在 OpenSearch 網域中設定 SAML

在 OpenSearch 網域中設定 SAML

1. 在 AWS 管理主控台中，識別您要為 OpenSearch UI 使用者啟用精細分級存取控制的 OpenSearch Service 網域。
2. 導覽至特定網域的詳細資訊頁面。
3. 選取安全組態索引標籤，然後按一下編輯。
4. 透過 IAM Federate 展開 SAML。
5. 輸入您在 Okta 中定義的 roleKey subjectKey和 。
6. SelectSave 變更。

您也可以使用 設定精細存取控制 AWS CLI。

```
aws opensearch create-domain \
--domain-name testDomain \
--engine-version OpenSearch_1.3 \
--cluster-config
 InstanceType=r5.xlarge.search,InstanceCount=1,DedicatedMasterEnabled=false,ZoneAwarenessEnable
 \
--access-policies '{"Version": "2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":{"AWS":"*"},"Action":"es:*","Resource":"arn:aws:es:us-
east-1:12345678901:domain/neosaml10/*"}]}' \
--domain-endpoint-options '{"EnforceHTTPS":true,"TLSSecurityPolicy":"Policy-Min-
TLS-1-2-2019-07"}' \
--node-to-node-encryption-options '{"Enabled":true}' \
--encryption-at-rest-options '{"Enabled":true}' \
--advanced-security-options
 '{"Enabled":true,"InternalUserDatabaseEnabled":true,"MasterUserOptions":
{"MasterUserName":"********","MasterUserPassword":"********"}, "IAMFederationOptions":
{"Enabled": true,"SubjectKey":"TestSubjectKey","RolesKey":"TestRolesKey"}}' \
```

```
--ebs-options "EBSEnabled=true,VolumeType=gp2,VolumeSize=300" \
--no-verify-ssl \
--endpoint-url https://es.us-east-1.amazonaws.com \
--region us-east-1
```

若要更新現有的網域：

```
aws opensearch update-domain-config \
--domain-name testDomain \
--advanced-security-options
 '{"Enabled":true,"InternalUserDatabaseEnabled":true,"MasterUserOptions":
{"MasterUserName":"********","MasterUserPassword":"********"}, "IAMFederationOptions":
{"Enabled": true,"SubjectKey":"TestSubjectKey","RolesKey":"TestRolesKey"}}' \
--ebs-options "EBSEnabled=true,VolumeType=gp2,VolumeSize=300" \
--no-verify-ssl \
--endpoint-url https://es.us-east-1.amazonaws.com \
--region us-east-1
```

任務 3：在 OpenSearch Serverless 集合中設定 SAML

在 OpenSearch Serverless 中設定以 SAML 為基礎的精細存取控制

1. 開啟 AWS 管理主控台 並導覽至 Amazon OpenSearch Service。
2. 在導覽窗格的無伺服器下，選擇安全性，然後選擇身分驗證。
3. 在 IAM 聯合區段中，選取編輯。

   您可以使用此組態控制以 SAML 屬性為基礎的精細存取控制。IAM 聯合預設為停用。
4. 選取啟用 IAM 聯合。
5. 輸入您在 Okta 中定義的 `subjectKey`和 `roleKey`值。

   如需詳細資訊，請參閱the section called “精細存取控制的 SAML 屬性”。
6. 選取儲存。
7. 在無伺服器下的導覽窗格中，選擇資料存取政策。
8. 更新現有的政策或建立新的政策。
9. 展開規則，選擇新增主體，然後選擇 IAM 聯合身分使用者和群組。
10. 新增必要的委託人，然後選擇儲存。
11. 選擇 Grant (授予)。

12. 在此規則下，執行下列動作：

    - 選取您要為所選委託人定義的許可。
    - 指定您要套用許可的集合。
    - 或者，定義索引層級許可。

    > **Note**
    >
    > 您可以建立多個規則，將不同的許可指派給不同的委託人群組。

13. 完成時，選擇儲存。
14. 選擇建立。

或者，您可以使用 CLI 來建立集合的安全組態，如下所示：

```
aws opensearchserverless create-security-config --region "region"  --type iamfederation
 --name "configuration_name" --description "description" --iam-federation-options
 '{"groupAttribute":"GroupKey","userAttribute":"UserKey"}'
```

管理資料來源關聯和虛擬私有雲端存取許可

使用本節中的程序來管理資料來源關聯，並設定虛擬私有雲端 (VPC) 所需的任何存取許可。

主題

- 將資料來源與 OpenSearch UI 應用程式建立關聯
- 管理對 VPC 中網域的存取
- 設定對 VPC 中 OpenSearch Serverless 集合的存取

將資料來源與 OpenSearch UI 應用程式建立關聯

建立 OpenSearch UI 應用程式之後，您可以使用 主控台 AWS CLI ，或將其與一或多個資料來源建立關聯。之後，最終使用者可以從這些資料來源擷取資料以進行搜尋、使用儀表板等。

將資料來源與 OpenSearch UI 應用程式建立關聯（主控台）

使用主控台將資料來源與 OpenSearch UI 應用程式建立關聯

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 選擇 OpenSearch UI（儀表板），然後選擇 OpenSearch UI 應用程式的名稱。
3. 在關聯的資料來源區域中，選擇管理資料來源。
4. 從您要與應用程式建立關聯的 OpenSearch 網域和集合中進行選擇。

   > **Tip**
   >
   > 如果您找不到要尋找的資料來源，請聯絡您的管理員以授予您必要的許可。如需詳細資訊，請參閱the section called “建立使用 IAM Identity Center 身分驗證之應用程式的許可（選用）”。

5. 選擇下一步，然後選擇儲存。

將資料來源與應用程式建立關聯後，應用程式詳細資訊頁面上會啟用啟動應用程式按鈕。您可以選擇啟動應用程式以開啟歡迎使用 OpenSearch 頁面，您可以在其中建立和管理工作區。

如需使用工作區的詳細資訊，請參閱 the section called “使用 Amazon OpenSearch Service 工作區”。

管理對 VPC 中網域的存取

如果 VPC 中的 OpenSearch 網域與應用程式相關聯，VPC 管理員必須使用 主控台或 授權 OpenSearch UI 和 VPC 之間的存取 AWS CLI。

管理對 VPC 中網域的存取（主控台）

若要使用 設定對 VPC 網域的存取 AWS 管理主控台：

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇網域，然後選擇 VPC 網域的名稱。

   -或-

   選擇建立網域，然後設定網域的詳細資訊。

3. 選擇 VPC 端點索引標籤，然後選擇授權委託人。
4. 在授權委託人對話方塊中，選取從其他 AWS 服務授權委託人，然後從清單中選擇 OpenSearch 應用程式 （儀表板）。
5. 選擇 Authorize (授權)。

管理對 VPC 中網域的存取 (AWS CLI)

使用 授權 VPC 網域 AWS CLI

若要使用 授權 VPC 網域 AWS CLI，請執行下列命令。使用您的資訊取代######。

```
aws opensearch authorize-vpc-endpoint-access \
 --domain-name domain-name \
 --service application.opensearchservice.amazonaws.com \
 --region region-id
```

使用主控台撤銷 VPC 網域關聯

當不再需要關聯時，VPC 網域擁有者可以使用下列程序撤銷存取權。

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽窗格中，選擇網域，然後選擇 VPC 網域的名稱。
3. 選擇 VPC 端點索引標籤，然後選取 OpenSearch 應用程式 （儀表板） 列的按鈕。
4. 選擇撤銷存取權。

使用 撤銷 VPC 網域關聯 AWS CLI

若要撤銷與 OpenSearch UI 應用程式的 VPC 網域關聯，請執行下列命令。使用您的資訊取代######。

```
aws opensearch revoke-vpc-endpoint-access \
    --domain-name domain-name \
    --service application.opensearchservice.amazonaws.com \
    --region region-id
```

設定對 VPC 中 OpenSearch Serverless 集合的存取

如果 VPC 中的 Amazon OpenSearch Serverless 集合與應用程式相關聯，VPC 管理員可以透過建立新的網路政策並將其連接到集合來授權存取。

在 VPC 中設定 OpenSearch Serverless 集合的存取權 （主控台）

使用主控台設定對 VPC 中 OpenSearch Serverless 集合的存取

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽中，選擇網路政策，選擇網路政策的名稱，然後選擇編輯。

   -或-

   選擇建立網路政策，然後設定政策的詳細資訊。
3. 在存取類型區域中，選擇私有 （建議），然後選擇AWS 服務私有存取。
4. 在搜尋欄位中，選擇服務，然後選擇 `application.opensearchservice.amazonaws.com`。
5. 在資源類型區域中，選取啟用對 OpenSearch 端點的存取方塊。
6. 在搜尋集合 （搜尋） 或輸入特定字首字詞 （搜尋） 欄位中，選取集合名稱，然後輸入或選取要與網路政策建立關聯的集合名稱。
7. 選擇建立新網路政策或更新現有網路政策。

設定對 VPC 中 OpenSearch Serverless 集合的存取 (AWS CLI)

使用 設定對 VPC 中 OpenSearch Serverless 集合的存取 AWS CLI

1. 建立類似以下內容的 .json 檔案。使用您的資訊取代*######*。

```
{
    "Description" : "policy-description",
    "Rules": [{
       "ResourceType" : "collection",
        "Resource" : ["collection/collection_name"]
     }],
    "SourceServices" : [
          "application.opensearchservice.amazonaws.com"
      ],
```

```
      "AllowFromPublic" : false
}
```

2. 為 VPC 中的集合建立或更新網路政策，以使用 OpenSearch UI 應用程式。

   Create a network policy

   執行下列命令。使用您的資訊取代*######*。

```
aws opensearchserverless create-security-policy \
    --type network  \
    --region region \
    --endpoint-url endpoint-url \
    --name network-policy-name \
    --policy file:/path_to_network_policy_json_file
```

   此命令會傳回與以下內容相似的資訊：

```
{
    "securityPolicyDetail": {
        "createdDate": ******,
        "lastModifiedDate": ******,
        "name": "network-policy-name",
        "policy": [
            {
                "SourceVPCEs": [],
                "AllowFromPublic": false,
                "Description": "",
                "Rules": [
                    {
                        "Resource": [
                            "collection/network-policy-name"
                        ],
                        "ResourceType": "collection"
                    }
                ],
                "SourceServices": [
                    "application.opensearchservice.amazonaws.com"
                ]
            }
        ],
        "policyVersion": "******",
        "type": "network"
```

```
    }
}
```

Update a network policy

執行下列命令。使用您的資訊取代######。

```
aws opensearchserverless update-security-policy \
    --type network  \
    --region region \
    --endpoint-url endpoint-url \
    --name network-policy-name \
    --policy-version "policy_version_from_output_of_network_policy_creation" \
    --policy file:/path_to_network_policy_json_file
```

此命令會傳回與以下內容相似的資訊：

```
{
    "securityPolicyDetail": {
        "createdDate": ******,
        "lastModifiedDate": ******,
        "name": "network-policy-name",
        "policy": [
            {
                "SourceVPCEs": [],
                "AllowFromPublic": false,
                "Description": "",
                "Rules": [
                    {
                        "Resource": [
                            "collection/network-policy-name"
                        ],
                        "ResourceType": "collection"
                    }
                ],
                "SourceServices": [
                    "application.opensearchservice.amazonaws.com"
                ]
            }
        ],
        "policyVersion": "******",
        "type": "network"
```

```
    }
}
```

使用 Amazon OpenSearch Service 工作區

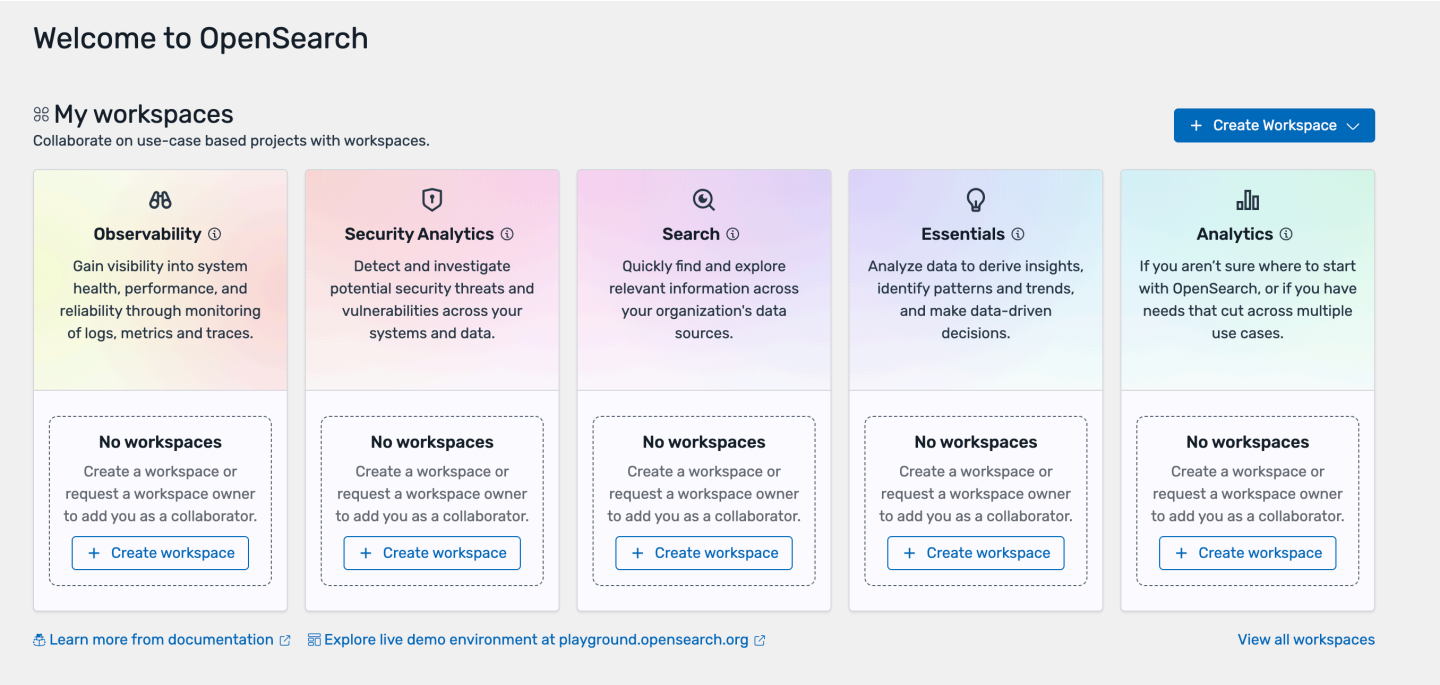
Amazon OpenSearch Service 支援建立多個使用案例特定的工作區。每個工作區都為熱門的使用案例提供精心策劃的體驗，例如可觀測性、安全性分析和搜尋。工作區也支援協作者管理，因此您只能與預期的協作者共用工作區，並管理每個工作區的許可。

建立 OpenSearch UI 應用程式工作區

在建立 OpenSearch UI 應用程式並與資料來源建立關聯，並且已為應用程式設定使用者許可之後，您可以啟動 OpenSearch UI 應用程式來建立工作區。

若要開始建立工作區，您可以在應用程式詳細資訊頁面上選取啟動應用程式按鈕，或使用 OpenSearch UI 應用程式 URL 在新的瀏覽器視窗中開啟 OpenSearch UI 應用程式首頁。

OpenSearch UI 應用程式提供建立工作區的選項，並列出首頁中依使用案例分類的所有現有工作區。



如需支援的工作區類型的詳細資訊，請參閱 the section called “工作區類型”。

工作區隱私權和協作者

您可以將工作區的隱私權設定定義為所有使用者的預設許可層級。您可以在建立工作區或修改現有工作區時執行此操作（在工作區協作者索引標籤上）。有三種隱私權選項可供選擇：

- 私有到協作者 – 只有您明確新增至工作區的協作者才能存取工作區。您可以為每個協作者定義許可層級。
- 任何人都可以檢視 – 有權存取 OpenSearch UI 應用程式的任何人都可以存取工作區並檢視其資產，但無法在工作區中進行任何變更。
- 任何人都可以編輯 – 有權存取 OpenSearch UI 應用程式的任何人都可以存取工作區、檢視其中的資產，以及變更工作區中的資產。

在工作區協作者索引標籤上，您可以將 IAM 使用者或角色和 AWS IAM Identity Center 使用者新增為工作區中的協作者。協作者有三個層級的許可可供選擇：

- 唯讀 – 協作者只能檢視工作區中的資產。如果工作區設定為使用任何人都可以編輯隱私權設定，則會覆寫此設定。
- 讀取和寫入 – 協作者可以檢視和編輯工作區中的資產。如果設定為使用任何人的工作區可以檢視隱私權設定，則協作者仍然可以編輯。
- Admin – 協作者可以更新設定並刪除工作區。協作者也可以變更工作區隱私權設定和管理協作者。建立工作區的使用者會自動指派為工作區管理員。

工作區類型

Amazon OpenSearch Service 提供五種工作區類型，每種類型都有不同的使用案例功能：

- 可觀測性工作區旨在透過監控日誌、指標和追蹤，取得系統運作狀態、效能和可靠性的可見性。
- Security Analytics 工作區旨在偵測和調查系統和資料的潛在安全威脅和漏洞。
- 搜尋工作區旨在快速尋找和探索整個組織資料來源的相關資訊。
- Essentials 工作區專為 OpenSearch Serverless 設計，做為資料來源，可讓分析資料以衍生洞見、識別模式和趨勢，並快速做出資料驅動型決策。您可以在 Essentials 工作區中找到並探索組織資料來源的相關資訊。
- Analytics （所有功能） 工作區專為多用途使用案例而設計，並支援 OpenSearch Service UI （儀表板） 中提供的所有功能。

跨區域和跨帳戶資料存取

OpenSearch UI 支援從不同 AWS 帳戶和 AWS 區域的 OpenSearch 網域存取資料。根據您的需求，您可以選擇兩種方法。下表比較這兩種方法。

> **Note**
>
> 跨帳戶資料存取和跨叢集搜尋都僅適用於 OpenSearch 網域。這兩種方法都不支援 OpenSearch Serverless 集合。

面向	跨帳戶資料存取	跨叢集搜尋
功能	在 OpenSearch UI 中將來自其他帳戶的網域關聯為直接資料來源	使用跨叢集搜尋連線跨連線網域查詢資料
Mechanism	直接存取 – OpenSearch UI 直接連接到另一個帳戶中的目標網域	間接存取 – 需要與 OpenSearch UI 相同帳戶中的本機網域，才能將請求轉送至遠端網域
跨帳戶支援	是	是
跨區域支援	否 – 來源和目標網域必須位於相同的 AWS 區域	是 – 來源和目的地網域可以位於不同的 AWS 區域中
跨網域的聯集資料	否 – 個別查詢每個網域做為個別資料來源	是 – 單一查詢可以彙總多個連線網域的結果
身分驗證方法	IAM 和 AWS IAM Identity Center	IAM （具有精細存取控制）
設定複雜性	較低 – 需要跨帳戶 IAM 角色進行驗證	較高 – 需要跨叢集連線、兩個網域上的存取政策，以及精細的存取控制
OpenSearch UI 中的資料來源可見性	每個跨帳戶網域會顯示為個別的資料來源	透過本機來源網域的連線別名存取遠端網域
遠端網域的寫入存取權	是 – 由目標網域的存取政策控制	否 – 跨叢集搜尋提供對遠端網域的唯讀存取

主題

- OpenSearch 網域的跨帳戶資料存取
- 跨叢集搜尋

OpenSearch 網域的跨帳戶資料存取

您可以在一個帳戶中設定 OpenSearch UI 應用程式，以存取不同帳戶中的 OpenSearch 網域。當您使用跨帳戶資料來源建立 OpenSearch UI 應用程式時，您會提供`iamRoleForDataSourceArn`指向目標帳戶中 IAM 角色的 。OpenSearch UI 透過擔任此角色並呼叫 `es:DescribeDomain` 來驗證網域可存取性來驗證請求。跨帳戶角色僅用於控制平面驗證。資料平面存取是由目標網域的存取政策單獨控制。

> **⚠ 範本程式碼**
>
> 本主題中的程式碼範例僅供說明之用。它們示範了基本功能，並且可能不包括錯誤處理、安全最佳實務或生產就緒功能。在生產環境中使用範本程式碼之前，請檢閱並修改範本以符合您的特定需求，並在您的環境中徹底測試。

重要概念

來源帳戶

AWS 帳戶 託管 OpenSearch UI 應用程式的 。

目標帳戶

AWS 帳戶 OpenSearch 網域所在的 。

跨帳戶角色

目標帳戶中的 IAM 角色，僅用於控制平面驗證。此角色只需要 `es:DescribeDomain`許可。

IAM Identity Center 應用程式角色

來源帳戶中的 IAM 角色，用於 IAM Identity Center 使用者資料平面存取。

先決條件

設定跨帳戶資料存取之前，請確定您有下列項目：

- AWS CLI 已安裝和設定
- 同時存取來源和目標 AWS 帳戶
- 對於 IAM Identity Center 流程： AWS IAM Identity Center 組織執行個體

案例

選擇符合您身分驗證方法和網域組態的案例：

- the section called “案例 1：存取公有網域的 IAM 使用者”
- the section called “案例 2：存取公有網域的 IAM Identity Center 使用者”
- the section called “案例 3：存取 VPC 網域的 IAM 使用者”
- the section called “案例 4：存取 VPC 網域的 IAM Identity Center 使用者”

案例 1：存取公有網域的 IAM 使用者

步驟 1：建立跨帳戶 IAM 角色（目標帳戶）

在目標帳戶中建立 IAM 角色，允許來源帳戶擔任該角色進行網域驗證。

建立跨帳戶角色

1. 建立允許來源帳戶擔任角色的信任政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::source-account-id:root"
    },
    "Action": "sts:AssumeRole"
  }]
}
```

2. 建立角色：

```
aws iam create-role \
  --role-name OpenSearchUIAccessRole \
  --assume-role-policy-document file://trust-policy.json
```

3. 建立僅具有 es:DescribeDomain動作的許可政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "es:DescribeDomain",
    "Resource": "arn:aws:es:region:target-account-id:domain/*"
  }]
}
```

4. 將許可政策連接至角色：

```
aws iam put-role-policy \
  --role-name OpenSearchUIAccessRole \
  --policy-name ValidationOnly \
  --policy-document file://permissions-policy.json
```

步驟 2：建立 OpenSearch 網域（目標帳戶）

在已啟用精細存取控制和加密的目標帳戶中建立 OpenSearch 網域：

```
aws opensearch create-domain \
  --domain-name domain-name \
  --engine-version OpenSearch_2.19 \
  --cluster-config InstanceType=m5.large.search,InstanceCount=1 \
  --ebs-options "EBSEnabled=true,VolumeType=gp3,VolumeSize=100" \
  --advanced-security-options
 '{"Enabled":true,"InternalUserDatabaseEnabled":true,"MasterUserOptions":
{"MasterUserName":"admin","MasterUserPassword":"master-password"}}' \
  --node-to-node-encryption-options '{"Enabled":true}' \
  --encryption-at-rest-options '{"Enabled":true}' \
  --domain-endpoint-options '{"EnforceHTTPS":true,"TLSSecurityPolicy":"Policy-Min-
TLS-1-2-2019-07"}' \
  --access-policies '{"Version":"2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":
{"AWS":"*"},"Action":"es:ESHttp*","Resource":"arn:aws:es:region:target-account-
id:domain/domain-name/*"}]}' \
  --region region
```

等待網域狀態變為 ，Active再繼續。

步驟 3：建立 OpenSearch UI 應用程式（來源帳戶）

使用跨帳戶資料來源在來源帳戶中建立應用程式：

```
aws opensearch create-application \
  --region region \
  --name "cross-account-iam-app" \
  --data-sources '[{
    "dataSourceArn":"arn:aws:es:region:target-account-id:domain/domain-name",
    "dataSourceDescription":"Cross-account domain",
    "iamRoleForDataSourceArn":"arn:aws:iam::target-account-
id:role/OpenSearchUIAccessRole"
  }]' \
  --app-configs '[{"key":"opensearchDashboards.dashboardAdmin.users","value":"[\"*
\"]"}]'
```

步驟 4：驗證和存取

擷取應用程式詳細資訊以取得端點 URL：

```
aws opensearch get-application \
  --region region \
  --id application-id
```

- 從回應導覽至應用程式端點 URL。
- 使用 IAM 登入資料登入。
- IAM 使用者使用自己的登入資料簽署資料平面請求。
- 目標網域存取政策控制使用者可以存取哪些資料。

案例 2：存取公有網域的 IAM Identity Center 使用者

步驟 1：建立跨帳戶 IAM 角色（目標帳戶）

在目標帳戶中建立 IAM 角色，允許來源帳戶擔任該角色進行網域驗證。

建立跨帳戶角色

1. 建立允許來源帳戶擔任角色的信任政策：

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::source-account-id:root"
    },
    "Action": "sts:AssumeRole"
  }]
}
```

2. 建立角色：

```
aws iam create-role \
  --role-name OpenSearchUIAccessRole \
  --assume-role-policy-document file://trust-policy.json
```

3. 建立僅具有 es:DescribeDomain動作的許可政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "es:DescribeDomain",
    "Resource": "arn:aws:es:region:target-account-id:domain/*"
  }]
}
```

4. 將許可政策連接至角色：

```
aws iam put-role-policy \
  --role-name OpenSearchUIAccessRole \
  --policy-name ValidationOnly \
  --policy-document file://permissions-policy.json
```

步驟 2：建立 OpenSearch 網域（目標帳戶）

在目標帳戶中建立 OpenSearch 網域。使用與 相同的命令the section called “步驟 2：建立 OpenSearch 網域（目標帳戶）”，但更新存取政策以允許來源帳戶的 IAM Identity Center 應用程式角色：

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::source-account-id:role/NeoIdCAppRole"
    },
    "Action": "es:ESHttp*",
    "Resource": "arn:aws:es:region:target-account-id:domain/domain-name/*"
  }]
}
```

等待網域狀態變為 ，Active再繼續。

步驟 3：建立 IAM Identity Center 應用程式的 IAM 角色 （來源帳戶）

在 OpenSearch UI 用於 IAM Identity Center 使用者資料平面存取的來源帳戶中建立 IAM 角色。

建立 IAM Identity Center 應用程式角色

1. 建立信任政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "application.opensearchservice.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "application.opensearchservice.amazonaws.com"
      },
      "Action": "sts:SetContext",
      "Condition": {
        "ForAllValues:ArnEquals": {
          "sts:RequestContextProviders": "arn:aws:iam::source-account-id:oidc-
provider/portal.sso.region.amazonaws.com/apl/application-id"
        }
      }
    }
```

```
  ]
}
```

2. 建立許可政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "OpenSearchDomain",
    "Effect": "Allow",
    "Action": ["es:ESHttp*"],
    "Resource": "*"
  }]
}
```

3. 建立角色並連接政策：

```
aws iam create-role \
  --role-name NeoIdCAppRole \
  --assume-role-policy-document file://neoidc-trust-policy.json

aws iam put-role-policy \
  --role-name NeoIdCAppRole \
  --policy-name NeoIdCAppPermissions \
  --policy-document file://neoidc-permissions-policy.json
```

步驟 4：使用 IAM Identity Center （來源帳戶） 建立 OpenSearch UI 應用程式

```
aws opensearch create-application \
  --region region \
  --name "cross-account-idc-app" \
  --iam-identity-center-options '{
    "enabled":true,
    "iamIdentityCenterInstanceArn":"arn:aws:sso:::instance/ssoins-instance-id",
    "iamRoleForIdentityCenterApplicationArn":"arn:aws:iam::source-account-
id:role/NeoIdCAppRole"
  }' \
  --data-sources '[{
    "dataSourceArn":"arn:aws:es:region:target-account-id:domain/domain-name",
    "dataSourceDescription":"Cross-account domain",
    "iamRoleForDataSourceArn":"arn:aws:iam::target-account-
id:role/OpenSearchUIAccessRole"
```

```
  }]' \
  --app-configs '[{"key":"opensearchDashboards.dashboardAdmin.users","value":"[\"*
\"]"}]'
```

步驟 5：建立和指派 IAM Identity Center 使用者和群組

建立 IAM Identity Center 使用者

執行下列命令。使用您的資訊取代*######*。

```
aws identitystore create-user \
  --identity-store-id d-directory-id \
  --user-name user-email \
  --display-name "display-name" \
  --name Formatted=string,FamilyName=last-name,GivenName=first-name \
  --emails Value=user-email,Type=work,Primary=true
```

建立 IAM Identity Center 群組並新增使用者

執行下列命令：

```
aws identitystore create-group \
  --identity-store-id d-directory-id \
  --display-name "OpenSearchUsers" \
  --description "Users with OpenSearch access"

aws identitystore create-group-membership \
  --identity-store-id d-directory-id \
  --group-id group-id \
  --member-id UserId=user-id
```

將使用者或群組指派給應用程式

執行以下命令：

```
aws sso-admin create-application-assignment \
  --application-arn "arn:aws:sso:::source-account-id:application/ssoins-instance-id/
apl-application-id" \
  --principal-id user-id-or-group-id \
  --principal-type USER
```

在目標網域上設定後端角色映射

將 IAM Identity Center 群組映射至目標網域上的 OpenSearch 安全角色：

```
curl -XPUT "https://domain-endpoint/_plugins/_security/api/rolesmapping/all_access" \
  -u admin:master-password \
  -H 'Content-Type: application/json' \
  -d '{
    "backend_roles": ["group-id"],
    "hosts": [],
    "users": []
  }'
```

步驟 6：驗證和存取

```
aws opensearch get-application \
  --region region \
  --id application-id
```

- 導覽至應用程式端點 URL。
- 使用 IAM Identity Center 使用者憑證登入。
- IAM Identity Center 使用者的資料請求會使用 IAM Identity Center 應用程式角色簽署，而不是跨帳戶角色。
- 網域控制資料存取許可上的後端角色映射。

案例 3：存取 VPC 網域的 IAM 使用者

步驟 1：建立跨帳戶 IAM 角色（目標帳戶）

在目標帳戶中建立 IAM 角色，允許來源帳戶擔任該角色進行網域驗證。

建立跨帳戶角色

1. 建立允許來源帳戶擔任角色的信任政策：

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

```
      "AWS": "arn:aws:iam::source-account-id:root"
    },
    "Action": "sts:AssumeRole"
  }]
}
```

2. 建立角色：

```
aws iam create-role \
  --role-name OpenSearchUIAccessRole \
  --assume-role-policy-document file://trust-policy.json
```

3. 建立僅具有 es:DescribeDomain動作的許可政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "es:DescribeDomain",
    "Resource": "arn:aws:es:region:target-account-id:domain/*"
  }]
}
```

4. 將許可政策連接至角色：

```
aws iam put-role-policy \
  --role-name OpenSearchUIAccessRole \
  --policy-name ValidationOnly \
  --policy-document file://permissions-policy.json
```

步驟 2：設定 VPC （目標帳戶）

如果目標帳戶中已存在 VPC，請略過此步驟。

```
# Create VPC
aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --region region

# Create subnet
aws ec2 create-subnet \
  --vpc-id vpc-id \
```

```
  --cidr-block 10.0.1.0/24 \
  --availability-zone regiona \
  --region region

# Create security group
aws ec2 create-security-group \
  --group-name opensearch-vpc-sg \
  --description "Security group for OpenSearch VPC domain" \
  --vpc-id vpc-id \
  --region region

# Allow inbound HTTPS
aws ec2 authorize-security-group-ingress \
  --group-id security-group-id \
  --protocol tcp \
  --port 443 \
  --cidr 10.0.0.0/16 \
  --region region
```

進一步了解 VPC 網域建立。

步驟 3：建立 VPC 網域（目標帳戶）

```
aws opensearch create-domain \
  --domain-name vpc-domain-name \
  --engine-version OpenSearch_2.19 \
  --cluster-config InstanceType=m5.large.search,InstanceCount=1 \
  --ebs-options "EBSEnabled=true,VolumeType=gp3,VolumeSize=100" \
  --vpc-options "SubnetIds=subnet-id,SecurityGroupIds=security-group-id" \
  --advanced-security-options
 '{"Enabled":true,"InternalUserDatabaseEnabled":true,"MasterUserOptions":
{"MasterUserName":"admin","MasterUserPassword":"master-password"}}' \
  --node-to-node-encryption-options '{"Enabled":true}' \
  --encryption-at-rest-options '{"Enabled":true}' \
  --domain-endpoint-options '{"EnforceHTTPS":true,"TLSSecurityPolicy":"Policy-Min-
TLS-1-2-2019-07"}' \
  --access-policies '{"Version":"2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":
{"AWS":"*"},"Action":"es:ESHttp*","Resource":"arn:aws:es:region:target-account-
id:domain/vpc-domain-name/*"}]}' \
  --region region
```

等待網域狀態變為 ，Active再繼續。

步驟 4：授權 OpenSearch UI 服務主體的 VPC 端點（目標帳戶）

> **Important**
>
> 這是 VPC 網域獨有的關鍵步驟。OpenSearch UI 服務必須明確授權才能存取 VPC 端點。

```
# Authorize the service principal
aws opensearch authorize-vpc-endpoint-access \
  --domain-name vpc-domain-name \
  --service "application.opensearchservice.amazonaws.com" \
  --region region

# Verify authorization
aws opensearch list-vpc-endpoint-access \
  --domain-name vpc-domain-name \
  --region region
```

預期回應：

```
{
  "AuthorizedPrincipalList": [
    {
      "PrincipalType": "AWS_SERVICE",
      "Principal": "application.opensearchservice.amazonaws.com"
    }
  ]
}
```

步驟 5：建立 OpenSearch UI 應用程式（來源帳戶）

```
aws opensearch create-application \
  --region region \
  --name "cross-account-vpc-iam-app" \
  --data-sources '[{
    "dataSourceArn":"arn:aws:es:region:target-account-id:domain/vpc-domain-name",
    "dataSourceDescription":"Cross-account VPC domain",
    "iamRoleForDataSourceArn":"arn:aws:iam::target-account-
id:role/OpenSearchUIAccessRole"
  }]' \
```

```
  --app-configs '[{"key":"opensearchDashboards.dashboardAdmin.users","value":"[\"*
\"]"}]'
```

步驟 6：驗證和存取

擷取應用程式詳細資訊以取得端點 URL：

```
aws opensearch get-application \
  --region region \
  --id application-id
```

- 從回應導覽至應用程式端點 URL。
- 使用 IAM 登入資料登入。
- IAM 使用者使用自己的登入資料簽署資料平面請求。
- 目標網域存取政策控制使用者可以存取哪些資料。

案例 4：存取 VPC 網域的 IAM Identity Center 使用者

步驟 1：建立跨帳戶 IAM 角色（目標帳戶）

在目標帳戶中建立 IAM 角色，允許來源帳戶擔任該角色進行網域驗證。

建立跨帳戶角色

1. 建立允許來源帳戶擔任角色的信任政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::source-account-id:root"
    },
    "Action": "sts:AssumeRole"
  }]
}
```

2. 建立角色：

```
aws iam create-role \
```

```
  --role-name OpenSearchUIAccessRole \
  --assume-role-policy-document file://trust-policy.json
```

3. 建立僅具有 es:DescribeDomain動作的許可政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "es:DescribeDomain",
    "Resource": "arn:aws:es:region:target-account-id:domain/*"
  }]
}
```

4. 將許可政策連接至角色：

```
aws iam put-role-policy \
  --role-name OpenSearchUIAccessRole \
  --policy-name ValidationOnly \
  --policy-document file://permissions-policy.json
```

步驟 2：設定 VPC （目標帳戶）

如果目標帳戶中已存在 VPC，請略過此步驟。

```
# Create VPC
aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --region region

# Create subnet
aws ec2 create-subnet \
  --vpc-id vpc-id \
  --cidr-block 10.0.1.0/24 \
  --availability-zone regiona \
  --region region

# Create security group
aws ec2 create-security-group \
  --group-name opensearch-vpc-sg \
  --description "Security group for OpenSearch VPC domain" \
  --vpc-id vpc-id \
```

```
  --region region

# Allow inbound HTTPS
aws ec2 authorize-security-group-ingress \
  --group-id security-group-id \
  --protocol tcp \
  --port 443 \
  --cidr 10.0.0.0/16 \
  --region region
```

進一步了解 VPC 網域建立。

步驟 3：建立 VPC 網域（目標帳戶）

使用與 相同的命令the section called “步驟 3：建立 VPC 網域（目標帳戶）”，但更新存取政策以允許來源帳戶的 IAM Identity Center 應用程式角色：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::source-account-id:role/NeoIdCAppRole"
    },
    "Action": "es:ESHttp*",
    "Resource": "arn:aws:es:region:target-account-id:domain/vpc-domain-name/*"
  }]
}
```

等待網域狀態變為 ，Active再繼續。

步驟 4：授權 OpenSearch UI 服務主體的 VPC 端點（目標帳戶）

> **Important**
>
> 這是 VPC 網域獨有的關鍵步驟。OpenSearch UI 服務必須明確授權才能存取 VPC 端點。

```
# Authorize the service principal
aws opensearch authorize-vpc-endpoint-access \
  --domain-name vpc-domain-name \
  --service "application.opensearchservice.amazonaws.com" \
```

```
  --region region

# Verify authorization
aws opensearch list-vpc-endpoint-access \
  --domain-name vpc-domain-name \
  --region region
```

預期回應：

```
{
  "AuthorizedPrincipalList": [
    {
      "PrincipalType": "AWS_SERVICE",
      "Principal": "application.opensearchservice.amazonaws.com"
    }
  ]
}
```

步驟 5：建立 IAM Identity Center 應用程式的 IAM 角色（來源帳戶）

在 OpenSearch UI 用於 IAM Identity Center 使用者資料平面存取的來源帳戶中建立 IAM 角色。

建立 IAM Identity Center 應用程式角色

1. 建立信任政策：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "application.opensearchservice.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "application.opensearchservice.amazonaws.com"
      },
      "Action": "sts:SetContext",
      "Condition": {
```

```
        "ForAllValues:ArnEquals": {
          "sts:RequestContextProviders": "arn:aws:iam::source-account-id:oidc-
provider/portal.sso.region.amazonaws.com/apl/application-id"
        }
      }
    }
  ]
}
```

2. 建立許可政策：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "OpenSearchDomain",
    "Effect": "Allow",
    "Action": ["es:ESHttp*"],
    "Resource": "*"
  }]
}
```

3. 建立角色並連接政策：

```
aws iam create-role \
  --role-name NeoIdCAppRole \
  --assume-role-policy-document file://neoidc-trust-policy.json

aws iam put-role-policy \
  --role-name NeoIdCAppRole \
  --policy-name NeoIdCAppPermissions \
  --policy-document file://neoidc-permissions-policy.json
```

步驟 6：使用 IAM Identity Center （來源帳戶） 建立 OpenSearch UI 應用程式

```
aws opensearch create-application \
  --region region \
  --name "cross-account-vpc-idc-app" \
  --iam-identity-center-options '{
    "enabled":true,
    "iamIdentityCenterInstanceArn":"arn:aws:sso:::instance/ssoins-instance-id",
    "iamRoleForIdentityCenterApplicationArn":"arn:aws:iam::source-account-
id:role/NeoIdCAppRole"
```

```
  }' \
  --data-sources '[{
    "dataSourceArn":"arn:aws:es:region:target-account-id:domain/vpc-domain-name",
    "dataSourceDescription":"Cross-account VPC domain",
    "iamRoleForDataSourceArn":"arn:aws:iam::target-account-
id:role/OpenSearchUIAccessRole"
  }]' \
  --app-configs '[{"key":"opensearchDashboards.dashboardAdmin.users","value":"[\"*
\"]"}]'
```

步驟 7：建立和指派 IAM Identity Center 使用者和群組

建立 IAM Identity Center 使用者

執行下列命令。使用您的資訊取代*######*。

```
aws identitystore create-user \
  --identity-store-id d-directory-id \
  --user-name user-email \
  --display-name "display-name" \
  --name Formatted=string,FamilyName=last-name,GivenName=first-name \
  --emails Value=user-email,Type=work,Primary=true
```

建立 IAM Identity Center 群組並新增使用者

執行下列命令：

```
aws identitystore create-group \
  --identity-store-id d-directory-id \
  --display-name "OpenSearchUsers" \
  --description "Users with OpenSearch access"

aws identitystore create-group-membership \
  --identity-store-id d-directory-id \
  --group-id group-id \
  --member-id UserId=user-id
```

將使用者或群組指派給應用程式

執行以下命令：

```
aws sso-admin create-application-assignment \
```

```
  --application-arn "arn:aws:sso:::source-account-id:application/ssoins-instance-id/
apl-application-id" \
  --principal-id user-id-or-group-id \
  --principal-type USER
```

在目標網域上設定後端角色映射

將 IAM Identity Center 群組對應至目標網域上的 OpenSearch 安全角色：

```
curl -XPUT "https://domain-endpoint/_plugins/_security/api/rolesmapping/all_access" \
  -u admin:master-password \
  -H 'Content-Type: application/json' \
  -d '{
    "backend_roles": ["group-id"],
    "hosts": [],
    "users": []
  }'
```

步驟 8：驗證和存取

```
aws opensearch get-application \
  --region region \
  --id application-id
```

- 導覽至應用程式端點 URL。
- 使用 IAM Identity Center 使用者憑證登入。
- IAM Identity Center 使用者的資料請求會使用 IAM Identity Center 應用程式角色簽署，而不是跨帳戶角色。
- 網域控制資料存取許可上的後端角色映射。

管理 應用程式

使用跨帳戶資料來源更新應用程式

執行下列命令。使用您的資訊取代######。

```
aws opensearch update-application \
  --region region \
```

```
  --id application-id \
  --data-sources '[{
    "dataSourceArn":"arn:aws:es:region:target-account-id:domain/domain-1",
    "dataSourceDescription":"First cross-account domain",
    "iamRoleForDataSourceArn":"arn:aws:iam::target-account-
id:role/OpenSearchUIAccessRole"
  },{
    "dataSourceArn":"arn:aws:es:region:target-account-id:domain/domain-2",
    "dataSourceDescription":"Second cross-account domain",
    "iamRoleForDataSourceArn":"arn:aws:iam::target-account-
id:role/OpenSearchUIAccessRole"
  }]'
```

Important

更新操作會取代整個資料來源陣列。包含您要保留的所有資料來源。

列出應用程式

執行以下命令：

```
aws opensearch list-applications \
  --region region
```

刪除應用程式

執行以下命令：

```
aws opensearch delete-application \
  --region region \
  --id application-id
```

撤銷 VPC 端點存取

執行以下命令：

```
aws opensearch revoke-vpc-endpoint-access \
  --domain-name vpc-domain-name \
  --service "application.opensearchservice.amazonaws.com" \
```

```
  --region region
```

快速參考

下表摘要說明網域類型和身分驗證方法之間的主要差異。

公有網域與 VPC 網域的比較

面向	公有網域	VPC 網域
VPC 端點授權	非必要	必要 – 必須授權 application.opensearchservice.amazonaws.com
網路設定	無	VPC、子網路、具有 HTTPS (443) 傳入的安全群組
IAM 存取政策	必要	必要
跨帳戶角色	跨帳戶的必要項目	跨帳戶的必要項目

IAM 使用者相較於 IAM Identity Center 使用者

面向	IAM 使用者	IAM Identity Center 使用者
資料平面登入資料	使用者自己的 IAM 登入資料	IAM Identity Center 應用程式角色
存取控制	網域存取政策	網域存取政策和後端角色映射
其他設定	無	IAM Identity Center 應用程式角色、使用者/群組建立、應用程式指派、後端角色映射
OpenSearch UI 應用程式組態	沒有 IAM Identity Center 選項	`--iam-identity-center-options` 必要

重要說明

- `iamRoleForDataSourceArn` 必須與 位於相同的 帳戶中`dataSourceArn`。

- 只有跨帳戶資料來源`iamRoleForDataSourceArn`需要 。為相同帳戶資料來源省略它。
- 跨帳戶角色只需要 `es:DescribeDomain`許可。它永遠不會用於資料平面存取。
- 對於 VPC 網域，必須同時設定 IAM 政策和 VPC 端點授權。
- 支援的引擎版本：OpenSearch 1.3 及更高版本。

疑難排解

問題	Resolution
應用程式建立失敗，並顯示「無法存取網域」	確認跨帳戶角色具有 `es:DescribeDomain` 許可，且信任政策允許來源帳戶。
VPC 網域關聯失敗	確定 VPC 端點已獲得 的授權`application.opensearchservice.amazonaws.com` 。
IAM 使用者的資料平面存取遭拒	檢查目標網域存取政策是否允許 IAM 使用者或角色主體。
IAM Identity Center 使用者的資料平面存取遭拒	確認後端角色映射包含 IAM Identity Center 群組 ID，且網域政策允許 IAM Identity Center 應用程式角色。
帳戶不相符錯誤	確定 `iamRoleForDataSourceArn` 位於與 中網域相同的 帳戶中`dataSourceArn` 。

跨叢集搜尋

在 Amazon OpenSearch Serverless 中使用跨叢集搜尋，您可以跨多個連線網域執行查詢和彙總。

Amazon OpenSearch Serverless 中的跨叢集搜尋會使用來源網域和目的地網域的概念。跨叢集搜尋請求來自來源網域。要查詢來源網域的目的地網域可以位於不同的 AWS 帳戶 或 AWS 區域 （或兩者）。使用跨叢集搜尋，您可以設定來源網域，以與相同帳戶中的 OpenSearch UI 建立關聯，然後建立目的地網域的連線。因此，您可以使用 OpenSearch UI 搭配來自目的地網域的資料，即使它們位於不同的帳戶或區域。

對於傳入和傳出 Amazon OpenSearch Service 的資料，您需要支付標準 AWS 資料傳輸費用。您無需為 OpenSearch Service 網域內節點之間的資料傳輸付費。如需「輸入」和「輸出」費用的詳細資訊，請參閱 Amazon EC2 隨需定價頁面上的資料傳輸。

您可以使用跨叢集搜尋作為 OpenSearch UI 與不同帳戶或不同區域中叢集相關聯的機制。根據預設，網域之間的請求會在傳輸中加密，作為node-to-node加密的一部分。

> **Note**
>
> 開放原始碼 OpenSearch 工具也會記錄跨叢集搜尋。請注意，開放原始碼工具的設定與 受管 Amazon OpenSearch Serverless 網域相比，開放原始碼叢集有很大的差異。
> 最值得注意的是，在 Amazon OpenSearch Serverless 中，您使用 AWS 管理主控台 而不是使用 cURL 請求來設定跨叢集連線。除了精細存取控制之外， 受管服務還使用 AWS Identity and Access Management (IAM) 進行跨叢集身分驗證。
> 因此，我們建議您使用本主題中的內容來設定網域的跨叢集搜尋，而不是開放原始碼 OpenSearch 文件。

使用跨叢集搜尋時的功能差異

相較於一般網域，使用跨叢集搜尋建立的目的地網域具有下列功能差異和要求：

- 您無法寫入或執行PUT命令至遠端叢集。您對遠端叢集的存取是唯讀的。
- 來源和目的地網域都必須是 OpenSearch 網域。您無法為 OpenSearch UI 連接 Elasticsearch 網域或自我管理的 OpenSearch/Elasticsearch 叢集。 OpenSearch
- 網域最多可以有 20 個與其他網域的連線。這包括傳出和傳入連線。
- 來源網域的 OpenSearch 版本必須與目的地網域相同或更高。如果您想要在兩個網域之間設定雙向連線，這兩個網域應該位於相同的版本中。我們建議您先將兩個網域升級至最新版本，再進行連線。如果您在設定雙向連線後需要更新網域，您必須先刪除連線，然後在之後重新建立。
- 您無法搭配遠端叢集使用自訂字典或 SQL。
- 您無法使用 CloudFormation 來連接網域。
- 您無法在 M3 或高載 (T2 和 T3) 執行個體上使用跨叢集搜尋。
- 跨叢集搜尋不適用於 Amazon OpenSearch Serverless 集合。

OpenSearch UI 的跨叢集搜尋先決條件

使用兩個 OpenSearch 網域設定跨叢集搜尋之前，請確定您的網域符合下列要求：

- 兩個網域都已啟用精細存取控制
- 兩個網域都已啟用Node-to-node加密

主題

- 使用跨叢集搜尋設定跨區域和跨帳戶資料存取的存取許可
- 在網域之間建立連線
- 使用跨叢集搜尋測試跨區域和跨帳戶資料存取的安全設定
- 刪除連線

使用跨叢集搜尋設定跨區域和跨帳戶資料存取的存取許可

當您將跨叢集搜尋請求傳送至來源網域時，網域會根據其網域存取政策評估該請求。跨叢集搜尋需要精細存取控制。以下是來源網域上具有開放存取政策的範例。

JSON

```
{
"Version":"2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
          "AWS": [
            "*"
          ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-1:111222333444:domain/src-domain/*"
    }
  ]
}
```

> **Note**
>
> 如果您在路徑中包含遠端索引，則必須對網域 ARN 中的 URI 進行 URL 編碼。
> 例如，使用下列 ARN 格式：
> `:arn:aws:es:us-east-1:111222333444:domain/my-domain/local_index,dst%3Aremote_index`

請勿使用下列 ARN 格式：
`arn:aws:es:us-east-1:111222333444:domain/my-domain/local_index,dst:remote_index.`

如果您選擇在精細存取控制之外使用限制性存取政策，則您的政策必須至少允許存取 `es:ESHttpGet`。以下是範例：

JSON

```
{
"Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111222333444:user/john-doe"
        ]
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/my-domain/*"
    }
  ]
}
```

來源網域上的精細存取控制會評估請求，以判斷該請求是否使用有效的 IAM 或 HTTP 基本登入資料簽署。如果是，精細存取控制接下來會評估使用者是否具有執行搜尋和存取資料的許可。

以下是搜尋的許可要求：

- 如果請求只搜尋目的地網域上的資料（例如 `dest-alias:dest-index/_search`），則只需要目的地網域上的許可。
- 如果請求搜尋兩個網域上的資料（例如 `source-index,dest-alias:dest-index/_search`），則兩個網域都需要許可。
- 若要使用精細存取控制，除了相關索引的標準讀取或搜尋許可之外，`indices:admin/shards/search_shards`還需要 許可。

來源網域會將請求傳遞至目的地網域。目的地網域會根據其網域存取政策評估此請求。若要支援 OpenSearch UI 中的所有功能，例如編製文件索引和執行標準搜尋，必須設定完整許可。以下是我們在目的地網域上建議的政策範例：

JSON

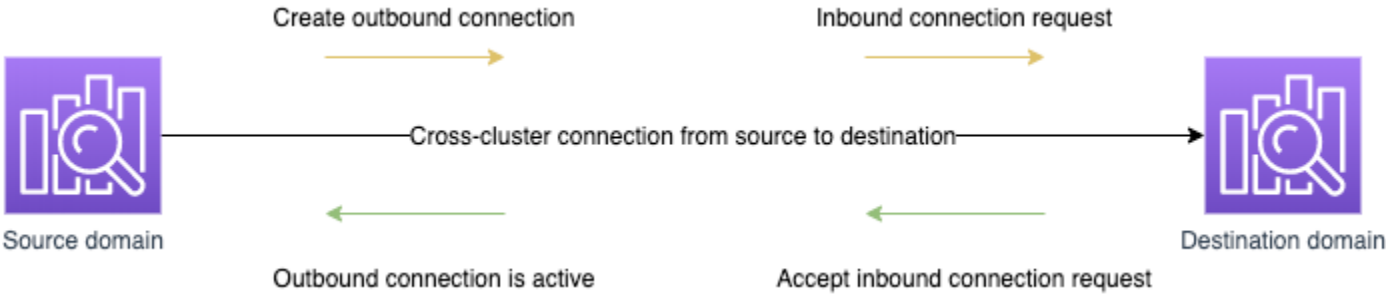
```
{
"Version":"2012-10-17",
 "Statement": [
    {
       "Effect": "Allow",
       "Principal": {
         "AWS": [
           "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-2:111222333444:domain/my-destination-
domain/*"
    },
    {
    "Effect": "Allow",
        "Principal": {
    "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:us-east-2:111222333444:domain/"
    }
  ]
}
```

如果您只想要執行基本搜尋，則最低政策要求是在沒有萬用字元支援的情況下套用`es:ESCrossClusterGet`許可到目的地網域。例如，在上述政策中，您將網域名稱指定為 *`/my-destination-domain`*，而不是 *`/my-destination-domain/*`*。

在此情況下，目的地網域會執行搜尋，並將結果傳回來源網域。來源網域會將其自身結果 (如果有的話) 與來自目的地網域的結果加以結合，然後將其傳回給您。

在網域之間建立連線

跨叢集搜尋連線從來源網域到目的地網域是單向的。這表示目的地網域（在不同帳戶或區域中）無法查詢來源網域，這是 OpenSearch UI 的本機網域。來源網域會建立與目的地網域的傳出連線。目的地網域會從來源網域接收傳入連線請求。



在網域之間建立連線

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽中，選擇網域。
3. 選擇要做為來源網域的網域名稱，然後選擇連線索引標籤。
4. 在傳出連線區域中，選擇請求。
5. 對於 Connection alias (連線別名)，輸入連線的名稱。在 OpenSearch UI 中使用連線別名來選取目的地網域。
6. 針對連線模式，選擇跨叢集搜尋或複寫的直接。
7. 若要指定連線應該在搜尋期間略過無法使用的叢集，請選取略過無法使用的叢集方塊。選擇此選項可確保您的跨叢集查詢傳回部分結果，無論一或多個遠端叢集是否失敗。
8. 對於目的地叢集，請在連線至此 中的叢集 AWS 帳戶和連線至另一個叢集 AWS 帳戶之間進行選擇。
9. 針對遠端網域 ARN，輸入叢集的 Amazon Resource Name (ARN)。網域 ARN 可以位於網域詳細資訊頁面的一般資訊區域中。

   網域必須符合下列要求：

   - ARN 格式必須為 `arn:`*`partition`*`:es:`*`regionaccount-id`*`:`*`type`*`/`*`domain-id`*。例如：

     ```
     arn:aws:es:us-east-2:111222333444:domain/my-domain
     ```
   - 網域必須設定為使用 OpenSearch 1.0 版（或更新版本）或 Elasticsearch 6.7 版（或更新版本）。

- 必須在網域上啟用精細存取控制。
- 網域必須執行 OpenSearch。

10. 選擇請求。

跨叢集搜尋會先驗證連線請求，確認是否符合先決條件。如果網域不相容，連線請求會進入 `Validation failed` 狀態。

如果成功驗證連線請求，則會將其傳送至必須核准的目的地網域。在此核准之前，連線會保持 `Pending acceptance` 狀態。在目的地網域接受連線請求後，狀態會變更為 `Active`，而目的地網域會變為可供查詢。

網域頁面會顯示目的地網域的整體網域運作狀態和執行個體運作狀態詳細資訊。只有網域擁有者才能靈活建立、檢視、移除和監控與其網域的連線。

建立連線後，在連線網域節點之間流動的任何流量都會加密。當您將 VPC 網域連線至非 VPC 網域，且非 VPC 網域是可從網際網路接收流量的公有端點時，網域之間的跨叢集流量仍會加密且安全。

使用跨叢集搜尋測試跨區域和跨帳戶資料存取的安全設定

使用跨叢集搜尋設定跨區域和跨帳戶資料存取的存取許可後，建議您使用 測試設定Postman，這是協作 API 開發的第三方平台。

使用 設定您的安全設定 Postman

1. 在目的地網域上，為文件編製索引。以下是範例請求：

```
POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1
{
"Dracula": "Bram Stoker"
}
```

2. 若要從來源網域查詢此索引，請在查詢內包含目的地網域的連線別名。您可以在網域儀表板的 Connections (連線) 索引標籤上找到連線別名。以下是範例請求和截斷的回應：

```
GET https://src-domain.us-east-1.es.amazonaws.com/connection_alias:books/_search
{
...
  "hits": [
{
"_index": "source-destination:books",
```

```
  "_type": "_doc",
  "_id": "1",
  "_score": 1,
  "_source": {
"Dracula": "Bram Stoker"
  }
}
  ]
}
```

3. （選用）您可以在單一搜尋中建立包含多個網域的組態。例如，假設您設定了下列項目：

   domain-a 與 之間的連線domain-b，具有名為 的連線別名 cluster_b

   domain-a 與 之間的連線domain-c，具有名為 的連線別名 cluster_c

   在此情況下，您的搜尋會包含內容 domain-a、 domain-b和 domain-c。以下是範例請求和回應：

   請求

```
GET https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_search
{
  "query": {
"match": {
  "user": "domino"
}
  }
}
```

   回應：

```
{
"took": 150,
  "timed_out": false,
  "_shards": {
"total": 3,
"successful": 3,
"failed": 0,
"skipped": 0
  },
  "_clusters": {
```

```
"total": 3,
"successful": 3,
"skipped": 0
  },
  "hits": {
"total": 3,
"max_score": 1,
"hits": [
  {
"_index": "local_index",
    "_type": "_doc",
    "_id": "0",
    "_score": 1,
    "_source": {
"user": "domino",
      "message": "This is message 1",
      "likes": 0
    }
  },
  {
"_index": "cluster_b:b_index",
    "_type": "_doc",
    "_id": "0",
    "_score": 2,
    "_source": {
"user": "domino",
      "message": "This is message 2",
      "likes": 0
    }
  },
  {
"_index": "cluster_c:c_index",
    "_type": "_doc",
    "_id": "0",
    "_score": 3,
    "_source": {
"user": "domino",
      "message": "This is message 3",
      "likes": 0
    }
  }
]
  }
```

```
}
```

如果您未選擇在連線設定中略過無法使用的叢集，則您搜尋的所有目的地叢集都必須可供搜尋請求成功執行。否則，整個請求會失敗，即使其中一個網域無法使用，也不會傳回任何搜尋結果。

刪除連線

刪除連線會停止目的地網域上的任何跨叢集搜尋操作。

您可以在來源或目的地網域上執行下列程序，以移除連線。移除連線後，狀態為 的連線會保持可見 `Deleted` 15 天。

您無法刪除具有作用中跨叢集連線的網域。若要刪除網域，請先從該網域中移除所有傳入和傳出連線。這可確保您在刪除網域前，將跨叢集網域使用者納入考量。

刪除連線

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽中，選擇網域。
3. 選擇要刪除的網域名稱，然後選擇連線索引標籤。
4. 選取要刪除的連線名稱。
5. 選擇刪除，然後確認刪除。

從 VPC 端點管理對 OpenSearch UI 的存取

您可以使用 在 VPC 和 OpenSearch UI 之間建立私有連線 AWS PrivateLink。使用此連線，您可以存取 OpenSearch UI 應用程式，就像在相同的 VPC 中一樣。如此一來，您就不需要設定網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 建立連線。VPC 中的執行個體不需要公有 IP 地址即可存取 OpenSearch UI。

若要建立此私有連線，您必須先建立採用 的介面端點 AWS PrivateLink。端點網路介面會自動在您為介面端點指定的每個子網路中建立。這些是請求者管理的網路介面，可做為目的地為 OpenSearch UI 應用程式之流量的進入點。

在 VPC 和 OpenSearch UI 之間建立私有連線

您可以使用 AWS 管理主控台 或 建立私有連線，以從 VPC 存取 OpenSearch UI AWS CLI。

在 VPC 和 OpenSearch UI 之間建立私有連線 （主控台）

使用主控台在 VPC 和 OpenSearch UI 之間建立私有連線

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 在左側導覽的無伺服器下，選擇 VPC 端點。
3. 選擇 Create VPC endpoint (建立 VPC 端點)。
4. 在名稱中，輸入端點的名稱。
5. 針對 VPC，選取您將從中存取 OpenSearch UI 應用程式的 VPC。
6. 針對子網路，選取您要從中存取 OpenSearch UI 應用程式的子網路。

> **Note**
>
> 端點的 IP 地址和 DNS 類型是以子網路類型為基礎：
>
> - 雙堆疊：如果所有子網路同時具有 IPv4 和 IPv6 地址範圍。
> - IPv6：如果所有子網路都是僅限 IPv6 的子網路。
> - IPv4：如果所有子網路都有 IPv4 地址範圍。

7. 針對安全群組，選取要與端點網路介面建立關聯的一或多個安全群組。

> **Note**
>
> 在此步驟中，您要限制您授權至端點之傳入流量的連接埠、通訊協定和來源。確保安全群組規則允許將使用 VPC 端點與 OpenSearch UI 應用程式通訊的資源，也可以與端點網路介面通訊。

8. 8. 選擇建立端點。

在 VPC 和 OpenSearch UI 之間建立私有連線 (AWS CLI)

使用 在 VPC 和 OpenSearch UI 之間建立私有連線 AWS CLI

執行下列命令。使用您的資訊取代######。

```
aws opensearchserverless create-vpc-endpoint \
    --region region \
```

```
    --endpoint endpoint \
    --name vpc_endpoint_name \
    --vpc-id vpc_id \
    --subnet-ids subnet_ids
```

更新 VPC 端點政策以允許存取 OpenSearch UI 應用程式

建立私有連線後，請更新 VPC 端點政策，以透過指定應用程式 ID 來允許存取 VPC 端點政策中的 OpenSearch UI 應用程式。

如需有關更新 VPC 端點政策的資訊，請參閱《 AWS PrivateLink 指南》中的更新 VPC 端點政策。

確定 VPC 端點政策包含下列陳述式。將*#####*取代為您自己的資訊。

```
{
    "Statement": [{
        "Action": ["opensearch:*"],
        "Effect": "Allow",
        "Principal": "*",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "opensearch:ApplicationId": ["opensearch-ui-application-id"]
            }
        }
    }]
}
```

在 VPC 端點政策中撤銷對 OpenSearch UI 的存取

OpenSearch UI 需要 VPC 端點政策中的明確許可，以允許使用者從 VPC 存取應用程式。如果您不再希望使用者從 VPC 存取 OpenSearch UI，您可以在端點政策中移除 許可。之後，使用者在嘗試存取 OpenSearch UI 時遇到`403 forbidden`錯誤訊息。

如需有關更新 VPC 端點政策的資訊，請參閱《 AWS PrivateLink 指南》中的更新 VPC 端點政策。

以下是拒絕從 VPC 存取 UI 應用程式的 VPC 端點政策範例：

```
{
    "Statement": [{
        "Action": ["opensearch:*"],
```

```
        "Effect": "Allow",
        "Principal": "*",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "opensearch:ApplicationId": [""]
            }
        }
    }]
}
```

將儲存的物件從 OpenSearch Dashboards 遷移至 OpenSearch UI

如果您在 OpenSearch Dashboards 中有現有的儀表板、視覺化效果、索引模式和其他已儲存的物件，您可以在 OpenSearch UI 中遷移和重複使用它們。

遷移至 OpenSearch UI 的優點：

- 高可用性 – OpenSearch UI 託管在 中 AWS 雲端 ，並在網域升級和維護期間保持可用，而 OpenSearch Dashboards 託管在網域中，將暫時無法使用。
- 多個資料來源 – OpenSearch UI 可以跨多個資料來源提供合併的單一窗格，包括 OpenSearch 網域、無伺服器集合，以及與 Amazon S3 和 Amazon CloudWatch 的資料連線；而每個 OpenSearch Dashboards 只能使用一個網域或集合。
- OpenSearch UI 提供 AI Assistant 和 Workspaces 等其他功能。進一步了解：OpenSearch UI。

主題

- 遷移概觀
- 先決條件
- 步驟 1：從 OpenSearch Dashboards 匯出儲存的物件
- 步驟 2：將儲存的物件匯入 OpenSearch UI

遷移概觀

遷移程序包含下列高階步驟：

1. 從 OpenSearch Dashboards 匯出儲存的物件 – 使用 OpenSearch Dashboards 儲存的物件管理 UI 或匯出 API，將儀表板、視覺效果、索引模式和其他物件下載為換行分隔 JSON (NDJSON) 檔案。

2. 建立 OpenSearch UI 應用程式和工作區 – 如果您尚未這麼做，請建立 OpenSearch UI 應用程式和工作區來接收匯入的物件。
3. 在 OpenSearch UI 中註冊資料來源 – 將您的 OpenSearch 網域與 OpenSearch UI 應用程式建立關聯，並將其註冊為工作區內的資料來源。匯入物件中的索引模式會參考此資料來源。
4. 將儲存的物件匯入 OpenSearch UI – 使用 OpenSearch UI 儲存的物件管理 UI 或匯入 API，將 NDJSON 檔案上傳到目標工作區。
5. 驗證匯入的物件 – 在 OpenSearch UI 中開啟儀表板和視覺化，以驗證它們是否正確顯示，以及資料是否從相關聯的網域或集合中流動。

先決條件

在遷移之前，請確認下列先決條件：

- 您擁有呼叫 Amazon OpenSearch Service 和 OpenSearch APIs所需的必要 IAM 許可。如需詳細資訊，請參閱the section called “建立 Amazon OpenSearch Service 應用程式所需的許可”。
- 您可以存取要遷移的網域或集合和 OpenSearch Dashboards。
- 您已建立 OpenSearch UI 應用程式。如需建立應用程式和工作區的資訊，請參閱 the section called “開始使用”。
- 您已將相同的網域或集合關聯至 OpenSearch UI 應用程式。如需建立資料來源關聯的資訊，請參閱 the section called “管理資料來源關聯和 VPC 存取許可”。

> **Note**
>
> OpenSearch UI 僅支援 OpenSearch 1.3 版和更新版本。在嘗試遷移儲存的物件之前，請確認您的 OpenSearch 網域正在執行 1.3 版或更新版本。

步驟 1：從 OpenSearch Dashboards 匯出儲存的物件

使用管理 UI 或匯出 API 從 OpenSearch Dashboards 匯出您儲存的物件。匯出會產生以換行分隔的 JSON (NDJSON) 檔案，其中包含所有選取的已儲存物件類型及其相依性。

主題

- 在 OpenSearch Dashboards 上手動匯出
- 透過 API 匯出

在 OpenSearch Dashboards 上手動匯出

使用 OpenSearch Dashboards 管理 UI 匯出儲存的物件

1. 開啟您的 OpenSearch Dashboards 執行個體。
2. 在左側導覽面板中，選擇管理。
3. 在儀表板管理下，選擇已儲存的物件。
4. 選取您要匯出的已儲存物件。若要匯出特定類型的所有物件，請使用搜尋列依類型篩選。若要匯出所有物件，請選取資料表標頭中的核取方塊。
5. 選擇 Export (匯出)。
6. 在匯出儲存的物件對話方塊中，確定已選取包含相關物件。此選項包含所選已儲存物件所依賴的所有物件，例如視覺化所參考的索引模式。只有在您想要手動管理相依性時，才清除此選項。
7. 選擇匯出，將`.ndjson`檔案下載到本機電腦。

> **Tip**
>
> 當您選擇包含相關物件時，匯出的 NDJSON 檔案會包含轉譯所選儀表板和視覺化所需的所有已儲存物件，包括其相依索引模式、視覺化和搜尋物件。這可簡化匯入步驟，並避免遺漏參考錯誤。

透過 API 匯出

您可以使用 OpenSearch Dashboards 已儲存物件匯出 API，以程式設計方式匯出已儲存物件。這對於自動化遷移或將匯出步驟整合到 CI/CD 管道非常有用。

> **Note**
>
> 如果您的 OpenSearch 網域已啟用精細存取控制，則必須在匯出請求中提供身分驗證憑證。使用您的使用者名稱和密碼新增 `-u`旗標，以使用 HTTP 基本身分驗證。如需有關身分驗證選項的詳細資訊，請參閱 the section called “精細定義存取控制”。

下列範例會匯出具有其相關物件的所有儀表板。使用您的資訊取代`######`。

```
curl -X POST \
```

```
    "https://dashboards-endpoint/_dashboards/api/saved_objects/_export" \
    -u 'master-username:master-password' \
    -H "Content-Type: application/json" \
    -H "osd-xsrf: true" \
    -d '{
        "type": ["dashboard", "visualization", "index-pattern", "search"],
        "includeReferencesDeep": true,
        "excludeExportDetails": false
    }' \
    -o saved-objects-export.ndjson
```

如果您的網域未啟用精細存取控制，您可以省略 -u旗標。

若要依 ID 匯出特定的已儲存物件，請使用 objects 參數，而非 type：

```
curl -X POST \
    "https://dashboards-endpoint/_dashboards/api/saved_objects/_export" \
    -u 'master-username:master-password' \
    -H "Content-Type: application/json" \
    -H "osd-xsrf: true" \
    -d '{
        "objects": [
            {"type": "dashboard", "id": "dashboard-id"},
            {"type": "visualization", "id": "visualization-id"}
        ],
        "includeReferencesDeep": true
    }' \
    -o saved-objects-export.ndjson
```

Note

若要尋找已儲存IDs，您可以使用已儲存物件 API 列出特定類型的所有物件。下列範例列出所有儀表板：

```
curl -X GET \
    "https://dashboards-endpoint/_dashboards/api/saved_objects/_find?
type=dashboard" \
    -u 'master-username:master-password'
```

回應包含每個已儲存物件的 ID。在 OpenSearch Dashboards 中檢視物件時，您也可以在瀏覽器 URL 中找到 ID。

步驟 2：將儲存的物件匯入 OpenSearch UI

匯出儲存的物件之後，您可以手動或透過 API 將 NDJSON 檔案匯入 OpenSearch UI。

主題

- 在 OpenSearch UI 上手動匯入
- 透過 API 匯入

在 OpenSearch UI 上手動匯入

使用 OpenSearch UI 管理 UI 匯入儲存的物件

1. 開啟您的 OpenSearch UI 應用程式並導覽至目標工作區。
2. 在工作區中，從頂端導覽中選擇資產，或前往工作區資產頁面。
3. 選擇匯入以開啟匯入資產對話方塊。
4. 選擇選取檔案，然後選取您從 OpenSearch Dashboards 匯出`.ndjson`的檔案。
5. 針對衝突管理，選擇下列其中一項：
    - 使用唯一 IDs（預設）– 為所有匯入的物件產生新 IDs，避免與現有資產衝突。
    - 檢查現有資產 – 檢查與現有物件的衝突。選取時，請選擇下列其中一個子選項：
        - 自動覆寫衝突 – 自動取代具有相同 ID 的現有資產。
        - 請求對衝突採取動作 – 系統會提示您個別解決每個衝突。
6. 選擇匯入。
7. 檢閱匯入摘要。

透過 API 匯入

若要使用具有 AWS 簽章第 4 版身分驗證的 API 匯入已儲存的物件，您必須先取得資料來源 ID，然後在匯入請求中使用它。使用您的資訊取代*######*。

步驟 1：取得工作區的資料來源 ID：

```
curl -X GET \
    "https://opensearch-ui-endpoint/w/workspace-id/api/saved_objects/_find?type=data-
source" \
    --aws-sigv4 "aws:amz:region:opensearch" \
    --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
```

```
    -H "x-amz-security-token: $AWS_SESSION_TOKEN" \
    -H "osd-xsrf: true"
```

> **Note**
>
> 回應包含資料來源 ID。在 OpenSearch UI 中檢視資料來源時，您也可以在瀏覽器 URL 中找到資料來源 ID。

步驟 2：使用步驟 1 的資料來源 ID 匯入儲存的物件：

```
curl -X POST \
    "https://opensearch-ui-endpoint/w/workspace-id/api/saved_objects/_import?
overwrite=true&dataSourceId=data-source-id" \
    --aws-sigv4 "aws:amz:region:opensearch" \
    --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
    -H "x-amz-security-token: $AWS_SESSION_TOKEN" \
    -H "osd-xsrf: true" \
    -F "file=@saved-objects-export.ndjson"
```

> **Note**
>
> 這些範例使用 curl 的內建`--aws-sigv4`選項（提供 curl 7.75 或更新版本）來簽署請求。在執行命令之前，將您的 AWS 登入資料設定為環境變數：`AWS_ACCESS_KEY_ID`、`AWS_SECRET_ACCESS_KEY`和 `AWS_SESSION_TOKEN`（如果使用臨時登入資料）。

OpenSearch UI 端點和配額

若要以程式設計方式連線至 AWS 服務，您可以使用 端點。服務配額也稱為限制，是您的帳戶 AWS 的服務資源或操作數量上限。

Amazon OpenSearch UI 是 OpenSearch Dashboards 的新一代 OpenSearch 使用者介面。它提供端點以存取您的 OpenSearch 儀表板。使用此主題來尋找 OpenSearch UI 的服務端點和服務配額。

如需其他 OpenSearch 服務的詳細資訊，請參閱服務端點和配額。

OpenSearch UI 端點

OpenSearch UI 適用於下列區域：

區域名稱	區域	端點	通訊協定
亞太區域 (孟買)	ap-south-1	opensearch.ap-south-1.amazonaws.com es.ap-south-1.amazonaws.com	HTTPS HTTPS
Europe (Paris)	eu-west-3	opensearch.eu-west-3.amazonaws.com es.eu-west-3.amazonaws.com	HTTPS HTTPS
美國東部 (俄亥俄)	us-east-2	opensearch.us-east-2.amazonaws.com es.us-east-2.amazonaws.com	HTTPS HTTPS
歐洲 (愛爾蘭)	eu-west-1	opensearch.eu-west-1.amazonaws.com es.eu-west-1.amazonaws.com	HTTPS HTTPS
歐洲 (法蘭克福)	eu-central-1	opensearch.eu-central-1.amazonaws.com es.eu-central-1.amazonaws.com	HTTPS HTTPS
南美洲 (聖保羅)	sa-east-1	opensearch.sa-east-1.amazonaws.com es.sa-east-1.amazonaws.com	HTTPS HTTPS
美國東部 (維吉尼亞北部)	us-east-1	opensearch.us-east-1.amazonaws.com es.us-east-1.amazonaws.com	HTTPS HTTPS
歐洲 (倫敦)	eu-west-2	opensearch.eu-west-2.amazonaws.com es.eu-west-2.amazonaws.com	HTTPS HTTPS

區域名稱	區域	端點	通訊協定
亞太地區 (東京)	ap-northeast-1	opensearch.ap-northeast-1.amazonaws.com es.ap-northeast-1.amazonaws.com	HTTPS HTTPS
美國西部 (奧勒岡)	us-west-2	opensearch.us-west-2.amazonaws.com es.us-west-2.amazonaws.com	HTTPS HTTPS
亞太區域 (新加坡)	ap-southeast-1	opensearch.ap-southeast-1.amazonaws.com es.ap-southeast-1.amazonaws.com	HTTPS HTTPS
亞太地區 (悉尼)	ap-southeast-2	opensearch.ap-southeast-2.amazonaws.com es.ap-southeast-2.amazonaws.com	HTTPS HTTPS
加拿大 (中部)	ca-central-1	opensearch.ca-central-1.amazonaws.com es.ca-central-1.amazonaws.com	HTTPS HTTPS
Europe (Stockholm)	eu-north-1	opensearch.eu-north-1.amazonaws.com es.eu-north-1.amazonaws.com	HTTPS HTTPS
亞太地區 (香港)	ap-east-1	opensearch.ap-east-1.amazonaws.com es.ap-east-1.amazonaws.com	HTTPS HTTPS
亞太地區 (首爾)	ap-northeast-2	opensearch.ap-northeast-2.amazonaws.com es.ap-northeast-2.amazonaws.com	HTTPS HTTPS

區域名稱	區域	端點	通訊協定
亞太地區 (大阪)	ap-northeast-3	opensearch.ap-northeast-3.amazonaws.com es.ap-northeast-3.amazonaws.com	HTTPS HTTPS
亞太地區 (海德拉巴)	ap-south-2	opensearch.ap-south-2.amazonaws.com es.ap-south-2.amazonaws.com	HTTPS HTTPS
歐洲 (西班牙)	eu-south-2	opensearch.eu-south-2.amazonaws.com es.eu-south-2.amazonaws.com	HTTPS HTTPS
美國西部 (加利佛尼亞北部)	us-west-1	opensearch.us-west-1.amazonaws.com es.us-west-1.amazonaws.com	HTTPS HTTPS
歐洲 (蘇黎世)	eu-central-2	opensearch.eu-central-2.amazonaws.com es.eu-central-2.amazonaws.com	HTTPS HTTPS
歐洲 (米蘭)	eu-south-1	opensearch.eu-south-1.amazonaws.com es.eu-south-1.amazonaws.com	HTTPS HTTPS

OpenSearch UI 服務配額

AWS 您的帳戶具有下列與 OpenSearch UI 資源相關的配額。

名稱	預設	可調整	備註
每個區域每個帳戶的 OpenSearch UI 應用程式	30	是	每個區域每個帳戶可建立的 OpenSearch UI 應用程式數量上限。 您可以使用服務配額將限制提高到 50，並自動獲得核准。若要請求更高的限制，請提交支援票證。

管理 Amazon OpenSearch Service 中的索引

將資料新增至 Amazon OpenSearch Service 之後，您經常必須重新編製該資料的索引、使用索引別名、將索引移至更經濟實惠的儲存體，或將其完全刪除。本章涵蓋儲存分層解決方案、索引狀態管理和其他索引操作。如需 OpenSearsearch 索引 API 的資訊，請參閱 OpenSearch 文件。

主題

- Amazon OpenSearch Service 網域的 OpenSearch 最佳化執行個體
- Amazon OpenSearch Service 的多層儲存
- Amazon OpenSearch Service 的 UltraWarm 儲存
- Amazon OpenSearch Service 冷儲存
- Amazon OpenSearch Service 中的索引狀態管理
- 使用索引彙總來匯總 Amazon OpenSearch Service 中的索引
- 在 Amazon OpenSearch Service 中轉換索引
- Amazon OpenSearch Service 的跨叢集複寫
- 使用遠端重新索引遷移 Amazon OpenSearch Service 索引
- 使用資料串流管理 Amazon OpenSearch Service 中的時間序列資料

Amazon OpenSearch Service 網域的 OpenSearch 最佳化執行個體

Amazon OpenSearch Service 的 OpenSearch 最佳化執行個體系列是經濟實惠的解決方案，用於儲存大量資料。具有 OpenSearch 最佳化執行個體的網域會使用本機儲存做為主要儲存體，並在資料送達時同步複製到 Amazon S3。此儲存結構提供更高的索引輸送量和高耐用性。OR1, OR2, OM2本機使用 Amazon Elastic Block Store (Amazon EBS) gp3或磁碟io1區，而 OI2 執行個體使用本機 NVMe 磁碟。OpenSearch 最佳化執行個體系列也支援在發生故障時自動資料復原。如需 OpenSearch 最佳化執行個體類型選項的相關資訊，請參閱 the section called “最新一代執行個體類型”。

如果您正在對日誌分析、可觀測性或安全性分析等繁重操作分析工作負載執行索引，您可以受益於 OpenSearch 最佳化執行個體改善的效能和運算效率。此外，OpenSearch 最佳化執行個體提供的自動資料復原可改善網域的整體可靠性。

OpenSearch Service 會將儲存體相關的 OpenSearch 最佳化執行個體指標傳送至 Amazon CloudWatch。如需可用指標的清單，請參閱 ???。

OpenSearch 最佳化執行個體可隨需使用，也可以搭配預留執行個體定價使用，其每小時費率適用於 Amazon EBS 和 Amazon S3 中佈建的執行個體和儲存體。

主題

- 限制
- 調校以獲得更好的擷取輸送量
- OpenSearch 最佳化執行個體與其他執行個體的差異
- OpenSearch 最佳化執行個體與 UltraWarm 執行個體有何不同
- 使用 OpenSearch 最佳化執行個體佈建網域

限制

為您的網域使用 OpenSearch 最佳化執行個體時，請考慮下列限制。

- 新建立的網域必須執行 OpenSearch 2.11 版或更新版本。
- 現有網域必須執行 OpenSearch 2.15 版或更新版本。
- 您的網域必須啟用靜態加密。如需詳細資訊，請參閱???。
- 如果您的網域使用專用主節點，則必須使用 Graviton 執行個體。如需專用主節點的詳細資訊，請參閱 ???。
- OpenSearch 最佳化執行個體上索引的重新整理間隔必須為 10 秒或更高。OpenSearch 最佳化執行個體的預設重新整理間隔為 10 秒。

調校以獲得更好的擷取輸送量

若要從 OpenSearch 最佳化執行個體取得最佳索引輸送量，建議您執行下列動作：

- 使用大量大小來改善緩衝區使用率。建議的大小為 10 MB。
- 使用多個用戶端來改善平行處理效能。
- 設定作用中主要碎片的數量，以符合資料節點的數量，以最大化資源使用率。

OpenSearch 最佳化執行個體與其他執行個體的差異

OpenSearch 最佳化執行個體與非最佳化執行個體的差別如下：

- 對於 OpenSearch 最佳化執行個體，僅對主要碎片執行索引。
- 如果 OpenSearch 最佳化執行個體使用複本設定，索引速率可能會低於實際速率。例如，如果有一個主要碎片和一個複本碎片，則當實際索引率為 2000 時，索引率可能會顯示 1000 的速率。
- OpenSearch 最佳化執行個體會在傳送至遠端來源之前執行緩衝區操作。這會導致較高的擷取延遲。

> **Note**
>
> `IndexingLatency` 指標不受影響，因為它不包含同步 Translog 的時間。

- 複本碎片可能比主要碎片落後幾秒鐘。您可以使用 `ReplicationLagMaxTime`Amazon CloudWatch 指標監控延遲

OpenSearch 最佳化執行個體與 UltraWarm 執行個體有何不同

OpenSearch Service 提供 UltraWarm 執行個體，這是一種經濟實惠的方式來存放大量唯讀資料。OpenSearch 最佳化和 UltraWarm 執行個體都會在 Amazon EBS 本機存放資料，在 Amazon S3 遠端存放資料。不過，OpenSearch 最佳化和 UltraWarm 執行個體有幾個重要方面不同：

- OpenSearch 最佳化執行個體會保留本機和遠端存放區中的資料副本。在 UltraWarm 執行個體中，資料主要保留在遠端存放區中，以降低儲存成本。根據您的使用模式，資料可以移至本機儲存體。
- OpenSearch 最佳化執行個體處於作用中狀態，可接受讀取和寫入操作，而 UltraWarm 執行個體上的資料是唯讀的，直到您手動將其移回熱儲存體為止。
- UltraWarm 倚賴索引快照提供資料耐久性。OpenSearch 最佳化執行個體會在幕後執行複寫和復原。如果發生紅色索引，OpenSearch 最佳化執行個體會自動從 Amazon S3 中的遠端儲存體還原缺少的碎片。復原時間會根據要復原的資料量而有所不同。

如需 UltraWarm 儲存體的詳細資訊，請參閱 ???。

使用 OpenSearch 最佳化執行個體佈建網域

當您使用 AWS 管理主控台 或 AWS Command Line Interface () 建立新網域時，您可以為資料節點選取 OpenSearch 最佳化執行個體AWS CLI。然後，您可以使用現有的工具編製索引和查詢資料。

主控台

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇 Domains (網域)。

3. 選擇建立網域。
4. 在資料節點數量區段中，展開執行個體系列功能表，然後選擇 OpenSearch 最佳化。
5. 選擇執行個體類型和其他儲存體設定。
6. 在加密區段中，確定已選取啟用靜態資料的加密。
7. 設定網域的其餘部分，然後選擇建立。

AWS CLI

若要使用 佈建使用 OpenSearch 最佳化儲存的網域 AWS CLI，您必須在 中提供特定執行個體類型大小 （ 例如 OR1, OR2, OM2或 OI2) 的值`InstanceType`。

下列範例會建立具有 OR1 執行個體大小的網域，`2xlarge`並啟用靜態加密。

```
aws opensearch create-domain \
  --domain-name test-domain \
  --engine-version OpenSearch_2.11 \
  --cluster-config
 "InstanceType=or1.2xlarge.search,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedMasterTy
 \
  --ebs-options "EBSEnabled=true,VolumeType=gp3,VolumeSize=200" \
  --encryption-at-rest-options Enabled=true \
  --advanced-security-options
 "Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions={MasterUserName=test-
user,MasterUserPassword=test-password}" \
  --node-to-node-encryption-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true \
  --access-policies '{"Version": "2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":{"AWS":"*"},"Action":"es:*","Resource":"arn:aws:es:us-
east-1:account-id:domain/test-domain/*"}]}'
```

下列範例會建立 OI2 執行個體大小的網域，`large`並啟用靜態加密。請注意，OI2 執行個體不需要 EBS 組態，因為它們使用本機 NVMe 儲存體。

```
aws opensearch create-domain \
  --domain-name test-domain-oi2 \
  --engine-version OpenSearch_2.11 \
  --cluster-config
 "InstanceType=oi2.2xlarge.search,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedMasterTy
 \
```

```
  --encryption-at-rest-options Enabled=true \
  --advanced-security-options
 "Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions={MasterUserName=test-
user,MasterUserPassword=test-password}" \
  --node-to-node-encryption-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true \
  --access-policies '{"Version": "2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":{"AWS":"*"},"Action":"es:*","Resource":"arn:aws:es:us-
east-1:account-id:domain/test-domain-oi2/*"}]}'
```

Amazon OpenSearch Service 的多層儲存

Amazon OpenSearch Service 的多層儲存是一種智慧型資料管理解決方案，可透過管理不同儲存層的資料來最佳化效能和成本。此架構可讓組織將經常存取的資料保留在高效能熱儲存中，同時將較不經常存取的資料移至更具成本效益的暖儲存，從而有效率地平衡效能和成本之間的權衡。

Amazon OpenSearch Service 為熱/暖儲存層提供兩種架構選項：

- OpenSearch 多層儲存架構
  - 結合 Amazon S3 與本機執行個體儲存體
  - 採用 OpenSearch Optimized Instances 技術
  - 支援暖層中的寫入操作
  - 支援在熱層和暖層之間無縫遷移資料
  - 適用於 OpenSearch 3.3 及更高版本
  - 不支援冷方案
- UltraWarm 型架構
  - 結合 Amazon S3 與本機執行個體儲存體
  - 採用 UltraWarm 執行個體
  - 針對唯讀暖層工作負載進行最佳化
  - 適用於 Elasticsearch 6.8 版及更新版本，以及所有 OpenSearch 版本
  - 支援冷方案

> **Note**
>
> 本文件僅著重於多層架構。如需 Ultrawarm 儲存架構，請參閱 Ultrawarm 和 Cold Storage

多層儲存架構

主題

- 主要優點
- 先決條件
- 限制
- 須知事項
- 建立多層網域
- 管理層遷移
- 安全組態
- 最佳實務
- 監控指標

主要優點

- 可寫入暖：支援暖索引的寫入操作
- 無縫遷移：跨儲存層無縫移動資料
- 成本最佳化：將非作用中的資料移至經濟實惠的暖儲存，以降低儲存成本
- 效能增強：維持熱層中經常存取資料的高效能
- 彈性資料管理：選擇最符合您工作負載需求的架構
- 自動化管理：跨儲存層簡化資料生命週期管理

先決條件

- 引擎版本：OpenSearch 3.3 或更新版本
- 執行個體系列：
  - 熱節點：OR1, OR2, OM2或 OI2
  - 暖節點：OI2
- 安全性：Node-to-node加密、靜態加密、HTTPS 強制執行

限制

- 適用於所有具有 OpenSearch 最佳化執行個體且尚未啟用 Ultrawarm 的網域
- 不支援冷方案

須知事項

- 熱遷移至暖遷移不會觸發 mult-tier 架構中的強制合併。如有需要，使用者仍然可以使用索引管理政策協調強制合併。
- 除了編製索引之外，暖節點現在也會執行背景合併操作 （類似熱）。
- 暖索引的所有搜尋請求都會路由至主要碎片，而複本只會在主要碎片關閉時提供讀取。
- 此架構也支援暖索引的自動化快照
- 只有熱索引才支援跨叢集複寫
- Shrink、Split 和 Clone 等索引 APIs 不適用於暖索引。

建立多層網域

步驟 1：建立網域

```
aws opensearch create-domain \
      --domain-name my-domain \
      --engine-version OpenSearch_3.3 \
      --cluster-config
 InstanceCount=3,InstanceType=or2.large.search,DedicatedMasterEnabled=true,DedicatedMasterType=
 \
      --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
      --node-to-node-encryption-options Enabled=true \
      --encryption-at-rest-options Enabled=true \
      --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
      --advanced-security-options
 '{"Enabled":true,"InternalUserDatabaseEnabled":true,"MasterUserOptions":
{"MasterUserName":"user_name","MasterUserPassword":"your_pass"}}' \
      --access-policies '{"Version": "2012-10-17",        "Statement":
[{"Effect":"Allow","Principal":"*","Action":"es:*","Resource":"*"}]}' \
      --region us-east-1
```

步驟 2：驗證暖節點

```
aws opensearch describe-domain-nodes --domain-name my-domain --region us-east-1
```

回應範例 （摘錄）：

```
{
      "NodeType": "Warm",
      "InstanceType": "oi2.large.search",
      "NodeStatus": "Active"
    }
```

管理層遷移

多層網域支援：

- 新的分層 APIs可簡化體驗
- 舊版 UltraWarm APIs以實現相容性

新的分層 APIs

將索引遷移至暖機：

```
curl -XPOST 'https://localhost:9200/index-name/_tier/warm'
```

回應：

```
{"acknowledged": true}
```

將索引遷移至熱索引：

```
curl -XPOST 'https://localhost:9200/index-name/_tier/hot'
```

回應：

```
{"acknowledged": true}
```

檢查分層狀態：

```
curl -XGET 'https://localhost:9200/index-name/_tier'
```

回應範例：

```
{
      "tiering_status": {
         "index": "index-name",
         "state": "RUNNING_SHARD_RELOCATION",
         "source": "HOT",
         "target": "WARM",
         "start_time": 1745836500563,
         "shard_level_status": {
           "running": 0,
           "total": 100,
           "pending": 100,
           "succeeded": 0
         }
      }
    }
```

詳細碎片檢視：

```
curl 'https://localhost:9200/index1/_tier?detailed=true'
```

列出所有進行中的遷移（文字）：

```
curl 'https://localhost:9200/_tier/all'
```

列出所有進行中的遷移 (JSON)：

```
curl 'https://localhost:9200/_tier/all?format=json'
```

依目標層篩選：

```
curl 'https://localhost:9200/_tier/all?target=_warm'
```

舊版 UltraWarm APIs以實現相容性

遷移至暖機：

```
curl -XPOST localhost:9200/_ultrawarm/migration/index2/_warm
```

遷移至熱：

```
curl -XPOST localhost:9200/_ultrawarm/migration/index2/_hot
```

檢查狀態：

```
curl -XGET localhost:9200/_ultrawarm/migration/index2/_status
```

安全組態

如果您在預先存在的 Amazon OpenSearch Service 網域上啟用多層儲存，則可能不會在網域上定義`storage_tiering_manager`角色。非系統管理員使用者必須映射至此角色，以便在使用精細存取控制的網域上管理暖索引。若要手動建立 `storage_tiering_manager` 角色，請執行以下步驟：

1. 在 OpenSearch Dashboards 中，移至 Security (安全性)，然後選擇 Permissions (許可)。
2. 選擇 Create action group (建立動作群組) 並設定下列群組：

群組名稱	許可
storage_tiering_cluster	indices：admin/_tier/all
storage_tiering_index_read	indices：admin/_tier/get、indices：admin/get
storage_tiering_index_write	indices：admin/_tier/hot_to_warm、indices：admin/_tier/warm_to_hot

3. 選擇 Roles (角色)，然後選擇 Create role (建立角色)。
4. 將角色命名為 `storage_tiering_manager`。
5. 對於 Cluster permissions (叢集許可)，選取 `storage_tiering_cluster` 和 `cluster_monitor`。
6. 對於 Index (索引)，輸入 *。
7. 針對索引許可，選取 `storage_tiering_index_readstorage_tiering_index_write`和 `indices_monitor`。

8. 選擇建立。
9. 建立角色之後，請將其映射到將管理多層索引的任何使用者或後端角色。

最佳實務

當您在 Amazon OpenSearch Service 網域中實作多層儲存時，請考慮下列最佳實務：

- 定期檢閱您的資料存取模式，以最佳化層配置
- 監控效能指標，以確保有效率的資源使用率
- 使用新的分層 APIs 來精細控制資料遷移

監控指標

多層儲存網域提供額外的指標來監控暖層效能。這些指標包括現有的 UltraWarm 指標和 OpenSearch Optimized Instances 架構特有的新指標：

新指標

指標名稱	節點層級統計資料	叢集層級統計資料	精細程度
WarmIndexingLatency	平均數	平均數	1 分鐘
WarmIndexingRate	平均數	平均值、最大值、總和	1 分鐘
WarmThreadpoolIndexingQueue	上限	總和、最大值、平均值	1 分鐘
WarmThreadpoolIndexingRejected	上限	總和	1 分鐘
WarmThreadpoolIndexingThreads	上限	總和，平均	1 分鐘

Amazon OpenSearch Service 的 UltraWarm 儲存

UltraWarm 提供經濟有效的方法，可在 Amazon OpenSearch Service 上儲存大量唯讀資料。標準資料節點使用「熱」儲存，它採用將執行個體存放區或 Amazon EBS 磁碟區連接到各個節點的形式。熱儲存可盡可能提供最快速的效能，以編製索引和搜尋新資料。

UltraWarm 節點使用 Simple Storage Service (Amazon S3) 和精密的快取解決方案來改善效能，而不是使用連接的儲存空間。對於您未積極寫入、較不常查詢且不需要相同效能的索引，UltraWarm 大幅降低了每 GiB 資料的成本。由於暖索引為唯讀狀態 (除非您將它們還原到熱儲存)，因此 UltraWarm 最適合不可變資料，例如日誌。

在 OpenSearch 中，這些暖索引的運作方式就像任何其他索引一樣。您可以使用相同的 API 來查詢索引，或使用索引在 OpenSearch Dashboards 中建立視覺效果。

主題

- 先決條件
- UltraWarm 儲存要求和效能考量
- UltraWarm 定價
- 啟用 UltraWarm
- 將索引遷移到 UltraWarm 儲存
- 自動化遷移
- 遷移調整
- 取消遷移
- 列出熱索引和暖索引
- 將暖索引傳回熱儲存區
- 從快照還原暖索引
- 暖索引的手動快照
- 將暖索引遷移到冷儲存
- KNN 索引的最佳實務
- 停用 UltraWarm

先決條件

UltraWarm 有幾個重要的先決條件：

- UltraWarm 需要 OpenSearch 或 Elasticsearch 6.8 或更高版本。
- 若要使用暖儲存區，網域必須具有專用的主節點。
- 將多可用區域與待命網域搭配使用時，暖節點的數量必須是正在使用的可用區域的倍數。
- 如果您的網域將 T2 或 T3 執行個體類型用於資料節點，您則無法使用暖儲存。
- 如果您的索引使用大約 k-NN (`"index.knn":true`)，您可以將其從 2.17 版和更新版本移至暖儲存。2.17 之前的版本上的網域可以升級至 2.17 以使用此功能，但 2.x 之前的版本上建立的 KNN 索引無法遷移至 UltraWarm。
- 如果網域使用精細存取控制，使用者必須在 OpenSearch Dashboards 中映射至 `ultrawarm_manager` 角色，以進行 UltraWarm API 呼叫。

> **Note**
>
> 某些預先存在的 OpenSearch Service 網域上可能未定義 `ultrawarm_manager` 角色。如果在 Dashboards 中沒有看到該角色，您需要手動建立它。

設定許可

如果在預先存在的 OpenSearch Service 網域上啟用 UltraWarm，可能未在網域上定義 `ultrawarm_manager` 角色。非系統管理員使用者必須映射至此角色，以便在使用精細存取控制的網域上管理暖索引。若要手動建立 `ultrawarm_manager` 角色，請執行以下步驟：

1. 在 OpenSearch Dashboards 中，移至 Security (安全性)，然後選擇 Permissions (許可)。
2. 選擇 Create action group (建立動作群組) 並設定下列群組：

Group name (群組名稱)	許可
`ultrawarm_cluster`	• `cluster:admin/ultrawarm/migration/list` • `cluster:monitor/nodes/stats`
`ultrawarm_index_read`	• `indices:admin/ultrawarm/migration/get` • `indices:admin/get`
`ultrawarm_index_write`	• `indices:admin/ultrawarm/migration/warm`

Group name (群組名稱)	許可
	• `indices:admin/ultrawarm/migration/hot` • `indices:monitor/stats` • `indices:admin/ultrawarm/migration/cancel`

3. 選擇 Roles (角色)，然後選擇 Create role (建立角色)。
4. 將角色命名為 ultrawarm_manager。
5. 對於 Cluster permissions (叢集許可)，選取 `ultrawarm_cluster` 和 `cluster_monitor`。
6. 對於 Index (索引)，輸入 *。
7. 對於 Index permissions (索引許可)，選取 `ultrawarm_index_read`、`ultrawarm_index_write` 以及 `indices_monitor`。
8. 選擇建立。
9. 建立角色後，將它映射到將管理 UltraWarm 索引的任何使用者或後端角色。

UltraWarm 儲存要求和效能考量

如 the section called “計算儲存需求” 中所述，熱儲存中的資料會產生大量開銷：複本、Linux 保留空間和 OpenSearch Service 保留空間。例如，含有一個複本碎片的 20 GiB 的主要碎片大約需要 58 GiB 的熱儲存空間。

由於它使用 Simple Storage Service (Amazon S3)，UltraWarm 不會產生任何此類開銷。計算 UltraWarm 儲存需求時，您只需考慮主要碎片的大小。S3 中的資料耐久性不需要複本，且 S3 消除了任何作業系統或服務考量事項。相同的 20 GiB 碎片需要 20 GiB 的暖儲存。如果您佈建了 `ultrawarm1.large.search` 執行個體，您可將其所有 20 TiB 的儲存空間上限用於主要碎片。請參閱the section called “UltraWarm 儲存配額”，了解執行個體類型摘要，以及每個類型可處理的儲存量上限。

針對 UltraWarm，仍建議使用 50 GiB 的最大碎片大小。CPU 核心數量和配置給每個 UltraWarm 執行個體類型的 RAM 數量可讓您了解它們可以同時搜尋的碎片數量。請注意，雖然只有主要分片會計入 S3 中的 UltraWarm 儲存空間，但 OpenSearch Dashboards 和 `_cat/indices` 仍會將 UltraWarm 索引大小報告為所有主碎片和複本碎片的總計。

例如，每個 `ultrawarm1.medium.search` 執行個體具有兩個 CPU 核心，並且可處理 S3 上最多 1.5 TiB 的儲存。其中兩個執行個體具有 3 TiB 的組合儲存，如果每個碎片為 50 GiB，則可達到大約

62 個碎片。如果對叢集的請求只搜索這些碎片中的四個，則效能可能會很好。如果請求範圍廣泛，並且搜尋所有 62 個碎片，則四個 CPU 核心可能很難執行操作。監控 `WarmCPUUtilization` 和 `WarmJVMMemoryPressure` UltraWarm 指標，以了解執行個體如何處理您的工作負載。

如果您的搜尋範圍廣泛或頻繁，請考慮將索引留在熱儲存中。如同任何其他 OpenSearch 工作負載一樣，判斷 UltraWarm 是否符合您需求的最重要步驟是使用實際的資料集執行代表性的用戶端測試。

UltraWarm 定價

使用熱儲存時，您需要按實際佈建內容付費。有些執行個體需要連接的 Amazon EBS 磁碟區，有些則包含執行個體存放區。無論該儲存空間是空的還是已滿，都是支付相同的價格。

使用 UltraWarm 儲存時，您需要按實際用量付費。一個 `ultrawarm1.large.search` 執行個體可在 S3 上處理高達 20 TiB 的儲存空間，但如果您只儲存 1 TiB 的資料，則只需支付 1 TiB 的資料費用。就像所有其他節點類型一樣，您也需要按照小時費率為每個 UltraWarm 節點支付費用。如需詳細資訊，請參閱the section called “定價”。

啟用 UltraWarm

主控台是建立使用暖儲存之網域最簡單的方法。建立網域時，請選擇啟用暖資料節點和您想要的暖節點數量。同樣的基本程序適用於現有的網域，只要它們符合 先決條件即可。即使網域狀態從 Processing (處理) 變更為 Active (作用中)，仍可能會長達數小時無法使用 UltraWarm。

將多可用區域與待命網域搭配使用時，暖節點的數量必須是可用區域的倍數。如需詳細資訊，請參閱the section called “具有待命的異地同步備份”。

您也可以使用 AWS CLI 或組態 API 來啟用 UltraWarm，具體而言就是 `ClusterConfig` 中的 `WarmEnabled`、`WarmCount` 和 `WarmType` 選項。

> **Note**
>
> 網域支援暖節點數量上限。如需詳細資訊，請參閱the section called “配額”。

CLI 命令範例

下列 AWS CLI 命令會建立已啟用三個資料節點、三個專用主節點、六個暖節點和精細存取控制的網域：

```
aws opensearch create-domain \
  --domain-name my-domain \
  --engine-version Opensearch_1.0 \
  --cluster-config
 InstanceCount=3,InstanceType=r6g.large.search,DedicatedMasterEnabled=true,DedicatedMasterType=
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --access-policies '{"Version": "2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":{"AWS":["123456789012"]},"Action":
["es:*"],"Resource":"arn:aws:es:us-west-1:123456789012:domain/my-domain/*"}]}' \
  --region us-east-1
```

如需詳細資訊，請參閱 AWS CLI 命令參考。

組態 API 請求範例

以下對組態 API 的請求會建立一個網域，它具有三個資料節點、三個專用主節點、六個已啟用精細存取控制的暖節點、以及一個限制性存取政策：

```
POST https://es.us-east-2.amazonaws.com/2021-01-01/opensearch/domain
{
  "ClusterConfig": {
    "InstanceCount": 3,
    "InstanceType": "r6g.large.search",
    "DedicatedMasterEnabled": true,
    "DedicatedMasterType": "r6g.large.search",
    "DedicatedMasterCount": 3,
    "ZoneAwarenessEnabled": true,
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": 3
    },
    "WarmEnabled": true,
    "WarmCount": 6,
    "WarmType": "ultrawarm1.medium.search"
  },
  "EBSOptions": {
```

```
    "EBSEnabled": true,
    "VolumeType": "gp2",
    "VolumeSize": 11
  },
  "EncryptionAtRestOptions": {
    "Enabled": true
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"
  },
   "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true,
    "MasterUserOptions": {
      "MasterUserName": "master-user",
      "MasterUserPassword": "master-password"
    }
  },
  "EngineVersion": "Opensearch_1.0",
  "DomainName": "my-domain",
  "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/my-domain/*\"}]}"
}
```

如需詳細資訊，請參閱 Amazon OpenSearch Service API 參考。

將索引遷移到 UltraWarm 儲存

如果您已完成索引寫入操作，且不再需要盡可能最快速的搜尋效能，請將其從熱儲存遷移到 UltraWarm：

```
POST _ultrawarm/migration/my-index/_warm
```

接著，檢查遷移狀態：

```
GET _ultrawarm/migration/my-index/_status
```

```
{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_SHARD_RELOCATION",
    "migration_type": "HOT_TO_WARM",
    "shard_level_status": {
      "running": 0,
      "total": 5,
      "pending": 3,
      "failed": 0,
      "succeeded": 2
    }
  }
}
```

索引運作狀態必須為綠色才能執行遷移。如果您快速地連續遷移多個索引，您可以取得純文字格式的所有遷移摘要，與 _cat API 類似：

```
GET _ultrawarm/migration/_status?v

index    migration_type state
my-index HOT_TO_WARM    RUNNING_SHARD_RELOCATION
```

OpenSearch Service 一次將一個索引遷移至 UltraWarm。您最多可以在佇列中進行 200 個遷移操作。任何超出限制的請求都會被拒絕。若要檢查佇列中的目前遷移數目，請監控 HotToWarmMigrationQueueSize 指標。在整個遷移過程中，索引仍然可用，無需停機。

遷移程序具有下列狀態：

```
PENDING_INCREMENTAL_SNAPSHOT
RUNNING_INCREMENTAL_SNAPSHOT
FAILED_INCREMENTAL_SNAPSHOT
PENDING_FORCE_MERGE
RUNNING_FORCE_MERGE
FAILED_FORCE_MERGE
PENDING_FULL_SNAPSHOT
RUNNING_FULL_SNAPSHOT
FAILED_FULL_SNAPSHOT
PENDING_SHARD_RELOCATION
RUNNING_SHARD_RELOCATION
FINISHED_SHARD_RELOCATION
```

如這些狀態所示，遷移可能在快照、碎片重新配置或強制合併期間失敗。快照或碎片重新配置期間的失敗通常是因為節點故障或 S3 連線問題。磁碟空間不足通常是強制合併失敗的根本原因。

遷移完成後，同一個 _status 請求會傳回錯誤。如果您在當下立即檢查索引，您可能會看到暖索引獨有的某些設定：

```
GET my-index/_settings

{
  "my-index": {
    "settings": {
      "index": {
        "refresh_interval": "-1",
        "auto_expand_replicas": "false",
        "provided_name": "my-index",
        "creation_date": "1599241458998",
        "unassigned": {
          "node_left": {
            "delayed_timeout": "5m"
          }
        },
        "number_of_replicas": "1",
        "uuid": "GswyCdR0RSq0SJYmzsIpiw",
        "version": {
          "created": "7070099"
        },
        "routing": {
          "allocation": {
            "require": {
              "box_type": "warm"
            }
          }
        },
        "number_of_shards": "5",
        "merge": {
          "policy": {
            "max_merge_at_once_explicit": "50"
          }
        }
      }
    }
  }
}
```

- 在此情況下，number_of_replicas 則為被動複本的數量，這不會耗用磁碟空間。
- routing.allocation.require.box_type 指定索引應使用暖節點，而不是標準資料節點。
- merge.policy.max_merge_at_once_explicit 指定遷移期間要同時合併的區段數目。

暖儲存中的索引為唯讀狀態 (除非您將它們還原到熱儲存)，因此 UltraWarm 最適合不可變資料，例如日誌。您可以查詢索引並刪除它們，但無法新增、更新或刪除個別文件。如果您嘗試，可能會遇到下列錯誤：

```
{
  "error" : {
    "root_cause" : [
      {
        "type" : "cluster_block_exception",
        "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage
 exceeded flood-stage watermark, index has read-only-allow-delete block];"
      }
    ],
    "type" : "cluster_block_exception",
    "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage exceeded
 flood-stage watermark, index has read-only-allow-delete block];"
  },
  "status" : 429
}
```

自動化遷移

建議您在索引達到特定存在時間或符合其他條件之後，使用 the section called “索引狀態管理” 自動化移轉程序。請參閱示範此工作流程的範例政策。

遷移調整

索引遷移至 UltraWarm 儲存需要強制合併。每個 OpenSearch 索引由一定數量的碎片組成，每個碎片都由一定數量的 Lucene 區段組成。強制合併操作會清除標示為刪除的文件，並節省磁碟空間。根據預設，UltraWarm 會將索引合併為一個區段，但使用預設值 20 的 kNN 索引除外。

您可以使用 index.ultrawarm.migration.force_merge.max_num_segments 設定，將此值變更為最多 1,000 個區段。較高的值會加快遷移程序，但會在移轉完成後增加暖索引的查詢延遲。若要變更設定，請提出下列請求：

```
PUT my-index/_settings
```

```
{
  "index": {
    "ultrawarm": {
      "migration": {
        "force_merge": {
          "max_num_segments": 1
        }
      }
    }
  }
}
```

若要檢查遷移程序的此階段需要多長時間，請監控 HotToWarmMigrationForceMergeLatency 指標。

取消遷移

UltraWarm 在佇列中按順序處理遷移。如果遷移位於佇列中，但尚未啟動，您可以使用下列請求從佇列中移除它：

```
POST _ultrawarm/migration/_cancel/my-index
```

如果您的網域使用精細存取控制，則您必須擁有 indices:admin/ultrawarm/migration/cancel 許可才能提出此請求。

列出熱索引和暖索引

UltraWarm 新增了兩個額外的選項 (與 _all 類似)，協助您管理熱索引和暖索引。如需所有暖索引或熱索引的清單，請提出下列請求：

```
GET _warm
GET _hot
```

您可以在指定索引的其他請求中使用這些選項，例如：

```
_cat/indices/_warm
_cluster/state/_all/_hot
```

將暖索引傳回熱儲存區

如果您需要再次寫入索引，請將其移轉回熱存儲區：

```
POST _ultrawarm/migration/my-index/_hot
```

您一次最多可以有 10 個從暖儲存到熱儲存的佇列遷移。OpenSearch Service 會依佇列順序一次處理一個遷移請求。若要檢查目前的數目，請監控 WarmToHotMigrationQueueSize 指標。

遷移完成後，請檢查索引設定以確定符合您的需求。索引會使用一個複本傳回熱儲存區。

從快照還原暖索引

除了適用於自動化快照的標準儲存庫以外，UltraWarm 還新增了適用於暖索引的第二個儲存庫：cs-ultrawarm。此儲存庫中的每個快照只包含一個索引。如果您刪除暖索引，其快照會保留在 cs-ultrawarm 儲存庫 14 天，就像任何其他自動快照一樣。

當您從 cs-ultrawarm 還原快照時，快照會還原至暖儲存區，而非熱儲存區。cs-automated 和 cs-automated-enc 儲存庫中的快照會還原為熱儲存。

將 UltraWarm 暖快照還原至熱儲存區

1. 識別包含您要還原之索引的最新快照：

```
GET _snapshot/cs-ultrawarm/_all?verbose=false

{
  "snapshots": [{
    "snapshot": "snapshot-name",
    "version": "1.0",
    "indices": [
      "my-index"
    ]
  }]
}
```

> **Note**
>
> 根據預設，GET _snapshot/<repo>操作會顯示儲存庫中每個快照的詳細資料資訊，例如開始時間、結束時間和持續時間。GET _snapshot/<repo> 操作會從儲存庫中包含的每個快照檔案擷取資訊。如果您不需要開始時間、結束時間和持續時間，而且只需要快照的名稱和索引資訊，建議您在列出快照時使用 verbose=false 參數，以將處理時間降至最低並防止逾時。

2. 如果索引已經存在，請將其刪除：

```
DELETE my-index
```

   如果您不想刪除索引，將其還原至熱儲存並重新索引它。

3. 還原快照：

```
POST _snapshot/cs-ultrawarm/snapshot-name/_restore
```

   UltraWarm 會忽略您在此還原請求中指定的任何索引設定，但您可以指定諸如 `rename_pattern` 和 `rename_replacement` 等選項。如需 OpenSearch 快照還原選項的摘要，請參閱 OpenSearch 文件。

暖索引的手動快照

您可以拍攝暖索引的手動快照，但我們不建議這樣做。自動化 `cs-ultrawarm` 儲存庫已包含每個暖索引的快照 (在遷移期間取得)，不需額外付費。

依預設，OpenSearch Service 不會在手動快照中包含暖索引。例如，以下呼叫只包含熱索引：

```
PUT _snapshot/my-repository/my-snapshot
```

如果您選擇拍攝暖索引的手動快照，則需要考慮幾個重要因素。

- 您不能混合熱索引和暖索引。例如，下列請求會失敗：

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1,hot-index-1",
  "include_global_state": false
}
```

  如果它們包含熱索引和暖索引的混合，則萬用字元 (*) 陳述式也會失敗。

- 每個快照只能包含一個暖索引。例如，下列請求會失敗：

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1,warm-index-2,other-warm-indices-*",
```

```
  "include_global_state": false
}
```

此請求成功：

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1",
  "include_global_state": false
}
```

- 手動快照一律還原至熱儲存，即使它們原本包含暖索引。

將暖索引遷移到冷儲存

如果在 UltraWarm 中有不常查詢的資料，請考慮將資料遷移至冷儲存。冷儲存適合您偶爾才存取的資料，或不再為使用中狀態的資料。您無法讀取或寫入冷索引，但無論何時需要查詢，都可以免費將它們遷移回暖儲存。如需說明，請參閱將索引遷移至冷儲存。

KNN 索引的最佳實務

- Ultrawarm/Cold 層適用於所有 KNN 索引引擎類型。對於使用 Lucene 引擎和磁碟最佳化向量搜尋的 KNN 索引，建議使用它，這不需要在堆積外記憶體中完全載入圖形資料。將其與 FAISS 和 NMSLIB 等原生記憶體內引擎搭配使用時，您必須考慮要主動搜尋的碎片圖形大小，並相應地佈建 UltraWarm 執行個體，最好是`uw.large`執行個體類型。例如，如果客戶已設定 2 個`uw.large`執行個體，則每個執行個體都會有大約 `knn.memory.circuit_breaker.limit * 61` GiB 的可用非堆積記憶體。如果您的所有暖查詢都以累積圖形大小不超過可用堆積外記憶體的碎片為目標，您可以獲得最佳效能。如果可用記憶體因移出而低於載入圖形所需的記憶體，並等待離散記憶體變成可用，則延遲會受到影響。因此，無論引擎為何，我們不建議將`uw.medium`執行個體用於使用記憶體內引擎的使用案例，或用於更高的搜尋輸送量案例。
- 遷移至 UltraWarm 的 KNN 索引不會強制合併至單一區段。這可避免因圖形大小對記憶體內引擎太大而對執行至 OOM 問題的熱節點和暖節點造成任何影響。由於每個碎片的區段數量增加，這可能會導致耗用更多本機快取空間，並允許較少的索引遷移至暖層。您可以選擇使用現有設定強制合併索引至單一區段，並在將索引遷移至暖層之前將其覆寫。如需詳細資訊，請參閱the section called “遷移調整”。
- 如果您有不常搜尋索引且不提供延遲敏感工作負載的使用案例，您可以選擇將這些索引遷移到 UltraWarm 層。這將協助您縮減熱層運算執行個體，並讓 UltraWarm 層運算處理這類低優先順序索

引上的查詢。這也有助於隔離低優先順序和高優先順序索引查詢之間消耗的資源，使其不會互相影響。

停用 UltraWarm

透過主控台是停用 UltraWarm 的最簡單方法。選擇網域、Actions (動作) 和 Edit cluster configuration (編輯叢集組態)。取消選取啟用暖資料節點，然後選擇儲存變更。您也可以使用 AWS CLI 和配置 API 中的 `WarmEnabled` 選項。

停用 UltraWarm 之前，您必須刪除所有暖索引，或將它們遷移回熱儲存區。暖儲存區變空後，請等待五分鐘，然後再嘗試停用 UltraWarm。

Amazon OpenSearch Service 冷儲存

冷儲存可讓您在 Amazon OpenSearch Service 網域中儲存任何數量不常存取的資料或歷史資料，並以比其他儲存層更低的成本依需求進行分析。如果您需要對舊資料進行定期研究或鑑識分析，則冷儲存適合。適合冷儲存的實際資料範例包括不常存取的日誌、必須保留以滿足合規要求的資料，或是具有歷史價值的日誌。

與 UltraWarm 儲存類似，冷儲存由 Simple Storage Service (Amazon S3) 支持。當您需要查詢冷資料時，可以選擇性地將其連接到現有的 UltraWarm 節點。您可以手動或使用索引狀態管理政策來管理冷資料的遷移和生命週期。

主題

- 先決條件
- 冷儲存要求和效能考量
- 冷儲存定價
- 啟用冷儲存
- 管理 OpenSearch Dashboards 中的冷索引
- 將索引遷移至冷儲存
- 自動移轉至冷儲存
- 取消遷移至冷儲存
- 列出冷索引
- 將冷索引遷移至暖儲存

- 從快照中還原冷索引
- 取消從冷儲存遷移至暖儲存
- 更新冷索引中繼資料
- 刪除冷索引
- 停用冷儲存

先決條件

冷儲存具有以下先決條件：

- 冷儲存需要 OpenSearch 或 Elasticsearch 版本 7.9 或更高版本。
- 若要在 OpenSearch Service 網域上啟用冷儲存，您還必須在相同的網域上啟用暖儲存。
- 若要使用冷儲存，網域必須具有專用的主節點。
- 如果您的網域將 T2 或 T3 執行個體類型用於資料節點，則無法使用冷儲存。
- 如果您的索引使用大約 k-NN (`"index.knn":true`)，您可以將其從 2.17 版和更新版本移至冷儲存。2.17 之前的版本上的網域可以升級至 2.17 以使用此功能，但 2.x 之前的版本上建立的 KNN 索引無法遷移至 Cold。
- 如果網域使用精細存取控制，則非管理員使用者必須對應至 OpenSearch Dashboards 中`cold_manager`的角色，才能管理冷索引。

> **Note**
>
> 某些預先存在的 OpenSearch Service 網域上可能不存在 `cold_manager` 角色。如果在 Dashboards 中沒有看到該角色，您需要手動建立它。

設定許可

如果在預先存在的 OpenSearch Service 網域上啟用冷儲存，可能未在網域上定義 `cold_manager` 角色。如果網域使用精細存取控制，則非管理員使用者必須對應至此角色，才能管理冷索引。若要手動建立 `cold_manager` 角色，請執行以下步驟：

1. 在 OpenSearch Dashboards 中，移至 Security (安全性)，然後選擇 Permissions (許可)。
2. 選擇 Create action group (建立動作群組) 並設定下列群組：

Group name (群組名稱)	許可
cold_cluster	• cluster:monitor/nodes/stats • cluster:admin/ultrawarm* • cluster:admin/cold/*
cold_index	• indices:monitor/stats • indices:data/read/minmax • indices:admin/ultrawarm/migration/get • indices:admin/ultrawarm/migration/cancel

3. 選擇 Roles (角色)，然後選擇 Create role (建立角色)。
4. 將角色命名為 cold_manager。
5. 對於 Cluster permissions (叢集許可)，選擇您建立的 cold_cluster 群組。
6. 對於 Index (索引)，輸入 *。
7. 對於 Index permissions (索引許可)，選擇您建立的 cold_index 群組。
8. 選擇建立。
9. 建立角色之後，請將其對應至管理冷索引的任何使用者或後端角色。

冷儲存要求和效能考量

由於冷儲存使用 Simple Storage Service (Amazon S3)，因此不會產生任何熱儲存負荷，例如複本、Linux 保留空間和 OpenSearch Service 保留空間。冷儲存沒有特定執行個體類型，因為它沒有連接任何運算容量。您可以在冷儲存中儲存任意數量的資料。監控 Amazon CloudWatch 中的 ColdStorageSpaceUtilization 指標，以查看您正在使用多少冷儲存空間。

冷儲存定價

與 UltraWarm 儲存類似，使用冷儲存時，您只需支付資料儲存費用。冷資料沒有運算成本，如果冷儲存中沒有資料，您將無需支付費用。

在冷儲存和暖儲存之間移動資料時，不會產生任何傳輸費用。雖然正在暖儲存和冷儲存之間遷移索引，但您仍然只需支付一份索引副本的費用。遷移完成後，會根據索引遷移到的儲存層計費。如需有關冷儲存定價的詳細資訊，請參閱 Amazon OpenSearch Service 定價。

啟用冷儲存

主控台是建立使用冷儲存之網域的最簡單方法。建立網域時，請先選擇啟用暖資料節點，因為您必須在相同的網域上啟用暖儲存。然後，選擇啟用冷儲存。

只要滿足先決條件，相同程序也適用於現有網域。即使網域狀態從 Processing (正在處理) 變更為 Active (作用中)，冷儲存可能會長達數小時無法使用。

您也可以使用 AWS CLI 或組態 API 來啟用冷儲存。

CLI 命令範例

下列 AWS CLI 命令會建立已啟用三個資料節點、三個專用主節點、啟用冷儲存和精細存取控制的網域：

```
aws opensearch create-domain \
  --domain-name my-domain \
  --engine-version Opensearch_1.0 \
  --cluster-
config ColdStorageOptions={Enabled=true},WarmEnabled=true,WarmCount=4,WarmType=ultrawarm1.mediu
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --region us-east-2
```

如需詳細資訊，請參閱 AWS CLI 命令參考。

組態 API 請求範例

對組態 API 的以下請求會建立一個網域，它具有三個資料節點、三個專用主節點、已啟用冷儲存，以及已啟用精細存取控制：

```
POST https://es.us-east-2.amazonaws.com/2021-01-01/opensearch/domain
{
  "ClusterConfig": {
    "InstanceCount": 3,
```

```
    "InstanceType": "r6g.large.search",
    "DedicatedMasterEnabled": true,
    "DedicatedMasterType": "r6g.large.search",
    "DedicatedMasterCount": 3,
    "ZoneAwarenessEnabled": true,
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": 3
     },
    "WarmEnabled": true,
    "WarmCount": 4,
    "WarmType": "ultrawarm1.medium.search",
    "ColdStorageOptions": {
       "Enabled": true
     }
  },
  "EBSOptions": {
    "EBSEnabled": true,
    "VolumeType": "gp2",
    "VolumeSize": 11
  },
  "EncryptionAtRestOptions": {
    "Enabled": true
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"
  },
   "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true,
    "MasterUserOptions": {
      "MasterUserName": "master-user",
      "MasterUserPassword": "master-password"
    }
  },
  "EngineVersion": "Opensearch_1.0",
  "DomainName": "my-domain"
}
```

如需詳細資訊，請參閱 Amazon OpenSearch Service API 參考。

管理 OpenSearch Dashboards 中的冷索引

您可以使用 OpenSearch Service 網域中現有的 Dashboards 介面來管理熱索引、暖索引和冷索引。Dashboards 可讓您在暖儲存和冷儲存之間遷移索引並監控索引遷移狀態，而不需要使用 CLI 或組態 API。如需詳細資訊，請參閱管理 OpenSearch Dashboards 中的索引。

將索引遷移至冷儲存

當您將索引遷移至冷儲存時，會為資料提供一個時間範圍，以便更輕鬆地探索。您可以根據索引中的資料選取時間戳記欄位，手動提供開始和結束時間戳記，或選擇不指定時間戳記。

參數	支援的值	Description
`timestamp_field`	來自索引映射的日期/時間欄位。	會運算提供的欄位的最小值和最大值並存儲為冷索引的 `start_time` 和 `end_time` 中繼資料。
`start_time` 和 `end_time`	下列其中一個格式： • strict_date_optional_time。例如：`yyyy-MM-dd'T'HH:mm:ss.SSSZ` 或 `yyyy-MM-dd` • Epoch 時間 (以毫秒為單位)	提供的值存儲為冷索引的 `start_time` 和 `end_time` 中繼資料。

如果您不希望指定時間戳記，請將 `?ignore=timestamp` 新增到請求。

下列請求會將暖索引遷移至冷儲存，並提供在該索引中資料的開始和結束時間：

```
POST _ultrawarm/migration/my-index/_cold
  {
    "start_time": "2020-03-09",
    "end_time": "2020-03-09T23:00:00Z"
  }
```

接著，檢查遷移狀態：

```
GET _ultrawarm/migration/my-index/_status
```

```
{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_METADATA_RELOCATION",
    "migration_type": "WARM_TO_COLD"
  }
}
```

OpenSearch Service 一次將一個索引遷移至冷儲存。您最多可以在佇列中進行 100 個遷移操作。任何超出限制的請求都會被拒絕。若要檢查佇列中的目前遷移數目，請監控 `WarmToColdMigrationQueueSize` 指標。遷移程序具有下列狀態：

```
ACCEPTED_COLD_MIGRATION - Migration request is accepted and queued.
RUNNING_METADATA_MIGRATION - The migration request was selected for execution and
 metadata is migrating to cold storage.
FAILED_METADATA_MIGRATION - The attempt to add index metadata has failed and all
 retries are exhausted.
PENDING_INDEX_DETACH - Index metadata migration to cold storage is completed. Preparing
 to detach the warm index state from the local cluster.
RUNNING_INDEX_DETACH - Local warm index state from the cluster is being removed. Upon
 success, the migration request will be completed.
FAILED_INDEX_DETACH - The index detach process failed and all retries are exhausted.
```

自動移轉至冷儲存

在索引達到特定時間或滿足其他條件時，您可使用索引狀態管理自動化遷移程序。請參閱範例政策，它示範如何自動將索引從熱儲存遷移至 UltraWarm 儲存，再遷移至冷儲存。

> **Note**
>
> 需要明確定義 `timestamp_field`，才能使用索引狀態管理政策將索引移至冷儲存。

取消遷移至冷儲存

如果遷移至冷儲存已排入佇列或處於失敗狀態，您可以使用下列請求取消遷移：

```
POST _ultrawarm/migration/_cancel/my-index

{
```

```
  "acknowledged" : true
}
```

如果您的網域使用精細存取控制，您需要 indices:admin/ultrawarm/migration/cancel 許可來提出此請求。

列出冷索引

在查詢之前，您可以列出冷儲存中的索引，以決定哪些要遷移至 UltraWarm 以進一步分析。下列請求會列出所有冷索引，依索引名稱排序：

```
GET _cold/indices/_search
```

回應範例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 3,
  "indices" : [
    {
      "index" : "my-index-1",
      "index_cold_uuid" : "hjEoh26mRRCFxRIMdgvLmg",
      "size" : 10339,
      "creation_date" : "2021-06-28T20:23:31.206Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-2",
      "index_cold_uuid" : "0vIS2n-oROmOWDFmwFIgdw",
      "size" : 6068,
      "creation_date" : "2021-07-15T19:41:18.046Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-3",
      "index_cold_uuid" : "EaeXOBodTLiDYcivKsXVLQ",
      "size" : 32403,
      "creation_date" : "2021-07-08T00:12:01.523Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
```

```
    }
  ]
}
```

篩選

您可以根據以字首為基礎的索引模式和時間範圍偏移來篩選冷索引。

以下請求列出了與 event-* 的字首模式匹配的索引：

```
GET _cold/indices/_search
 {
   "filters":{
      "index_pattern": "event-*"
   }
 }
```

回應範例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
  "indices" : [
    {
      "index" : "events-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

以下請求傳回的索引具有介於 2019-03-01 到 2020-03-01 之間的 start_time 和 end_time 中繼資料欄位：

```
GET _cold/indices/_search
{
  "filters": {
    "time_range": {
      "start_time": "2019-03-01",
```

```
      "end_time": "2020-03-01"
    }
  }
}
```

回應範例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
  "indices" : [
    {
      "index" : "my-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2019-05-09T00:00Z",
      "end_time" : "2019-09-09T23:00Z"
    }
  ]
}
```

排序

您可以按照中繼資料欄位 (例如索引名稱或大小) 來排序冷索引。以下請求列出按照大小排序 (降序) 的所有索引：

```
GET _cold/indices/_search
 {
 "sort_key": "size:desc"
 }
```

回應範例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 5,
  "indices" : [
    {
      "index" : "my-index-6",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
```

```
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-9",
      "index_cold_uuid" : "mbD3ZRVDRI6ONqgEOsJyUA",
      "size" : 57922,
      "creation_date" : "2021-07-07T23:41:35.640Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-5",
      "index_cold_uuid" : "EaeXOBodTLiDYcivKsXVLQ",
      "size" : 32403,
      "creation_date" : "2021-07-08T00:12:01.523Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

其他有效的排序金鑰為 `start_time:asc/desc`、`end_time:asc/desc` 和 `index_name:asc/desc`。

分頁

您可以分頁冷索引清單。使用 `page_size` 參數來設定每頁要傳回的索引數目 (預設值為 10)。針對冷索引的每個 `_search` 請求傳回一個 `pagination_id`，您可以將其用於後續呼叫。

以下請求會對冷索引的 `_search` 請求結果進行分頁，並顯示接下來的 100 個結果：

```
GET _cold/indices/_search?page_size=100
{
"pagination_id": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
}
```

將冷索引遷移至暖儲存

使用前一節中的篩選條件縮小冷索引清單範圍之後，請將它們遷移回 UltraWarm，您可以在其中查詢資料並使用它來建立視覺效果。

以下請求會將兩個冷索引遷移回暖儲存：

```
POST _cold/migration/_warm
 {
 "indices": "my-index1,my-index2"
 }


{
  "acknowledged" : true
}
```

若要檢查遷移狀態並擷取遷移 ID，請傳送下列請求：

```
GET _cold/migration/_status
```

回應範例

```
{
  "cold_to_warm_migration_status" : [
    {
      "migration_id" : "tyLjXCA-S76zPQbPVHkOKA",
      "indices" : [
        "my-index1,my-index2"
      ],
      "state" : "RUNNING_INDEX_CREATION"
    }
  ]
}
```

若要取得索引特定的遷移資訊，請包含索引名稱：

```
GET _cold/migration/my-index/_status
```

您可以按照當前遷移狀態列出索引，而不是指定索引。有效值為 _failed、_accepted 和 _all。

以下命令會取得單一遷移請求中所有索引的狀態：

```
GET _cold/migration/_status?migration_id=my-migration-id
```

使用狀態請求擷取遷移 ID。如需詳細的遷移資訊，請新增 &verbose=true。

您可以分 10 個或更少批次，將索引從冷儲存遷移到暖儲存，最多可同時遷移 100 個索引。任何超出限制的請求都會被拒絕。若要檢查佇列中的目前遷移數目，請監控 ColdToWarmMigrationQueueSize 指標。遷移程序具有下列狀態：

```
ACCEPTED_MIGRATION_REQUEST - Migration request is accepted and queued.
RUNNING_INDEX_CREATION - Migration request is picked up for processing and will create
 warm indexes in the cluster.
PENDING_COLD_METADATA_CLEANUP - Warm index is created and the migration service will
 attempt to clean up cold metadata.
RUNNING_COLD_METADATA_CLEANUP - Cleaning up cold metadata from the indexes migrated to
 warm storage.
FAILED_COLD_METADATA_CLEANUP - Failed to clean up metadata in the cold tier.
FAILED_INDEX_CREATION - Failed to create an index in the warm tier.
```

從快照中還原冷索引

如果您需要還原已刪除的冷索引，您可以依照 中的指示將其還原至暖層，the section called “從快照還原暖索引”然後再次將索引遷移回冷層。您無法將已刪除的冷索引直接還原回冷層。OpenSearch Service 會在刪除冷索引後保留 14 天。

取消從冷儲存遷移至暖儲存

如果從冷儲存到暖儲存的索引遷移已排入佇列或處於失敗狀態，您可以使用以下請求取消它：

```
POST _cold/migration/my-index/_cancel

{
  "acknowledged" : true
}
```

若要取消批次索引的遷移 (一次最多 10 個)，請指定遷移 ID：

```
POST _cold/migration/_cancel?migration_id=my-migration-id

{
  "acknowledged" : true
}
```

使用狀態請求擷取遷移 ID。

更新冷索引中繼資料

您可以更新冷索引的 `start_time` 和 `end_time` 欄位：

```
PATCH _cold/my-index
 {
 "start_time": "2020-01-01",
 "end_time": "2020-02-01"
 }
```

您無法更新冷儲存中的索引的 `timestamp_field`。

> **Note**
>
> OpenSearch Dashboards 不支援 PATCH 方法。使用 curl、Postman 或其他方法來更新冷中繼資料。

刪除冷索引

如果您未使用 ISM 政策，您可以手動刪除冷索引。以下要求會刪除冷索引：

```
DELETE _cold/my-index

{
  "acknowledged" : true
}
```

停用冷儲存

OpenSearch Service 主控台是停用冷儲存的最簡單方法。選取網域並選擇 Actions (動作)、Edit cluster configuration (編輯叢集組態)，然後取消選取 Enable cold storage (啟用冷儲存)。

若要使用 AWS CLI 或組態 API，請在 下`ColdStorageOptions`設定 `"Enabled"="false"`。

停用冷儲存之前，您必須刪除所有冷索引，或將它們遷移回暖儲存，否則停用動作會失敗。

Amazon OpenSearch Service 中的索引狀態管理

Amazon OpenSearch Service 中的索引狀態管理 (ISM) 可讓您定義自訂管理政策，這些政策能夠自動化例行任務並將它們套用至索引和索引模式。您不再需要設定和管理外部程序來執行索引操作。

政策包含預設狀態和狀態清單，供索引進行相互轉換。在每種狀態下，您可以定義要執行的操作清單和觸發這些轉換的條件。典型的使用案例是在一段時間後定期刪除舊的索引。例如，您可以定義一個政策，在 30 天後將索引移至 `read_only` 狀態，然後最終在 90 天後將索引刪除。

在您將政策連接至索引之後，ISM 會建立一個任務，每 5 至 8 分鐘 (或預先 1.3 個叢集為每 30 到 48 分鐘) 執行一次，以執行政策動作、檢查條件，然後將索引轉換為不同的狀態。執行此任務的基準時間是每 5 分鐘一次，再加上隨機的 0-60% 抖動，以確保您不會同時看到來自所有索引的活動突增。如果叢集狀態為紅色，ISM 就不會執行任務。

ISM 需要 OpenSearch 或 Elasticsearch 6.8 或更新版本。

> **Note**
>
> 本文件提供 ISM 和數個範例政策的簡短概觀。它還說明 Amazon OpenSearch Service 網域的 ISM 與自我管理 OpenSearch 叢集上的 ISM 有何不同。如需 ISM 的完整文件，包括完整的參數參考、每個設定的描述和 API 參考，請參閱 OpenSearch 文件中的索引狀態管理。

> **Important**
>
> 您無法再使用索引範本將 ISM 政策套用至新建立的索引。使用 ISM 範本欄位，您可以繼續自動管理新建立的索引。此更新引入了重大變更，它會影響使用此設定的現有 CloudFormation 範本。

建立 ISM 政策。

開始使用索引狀態管理

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 選取您要為其建立 ISM 政策的網域。

3. 從網域的儀表板，導覽至 OpenSearch Dashboards URL，並使用主要使用者名稱和密碼登入。URL 遵循以下格式：

```
domain-endpoint/_dashboards/
```

4. 開啟 OpenSearch Dashboards 的左側導覽面板，然後依序選擇 Index Management (索引管理) 和 Create policy (建立政策)。
5. 使用視覺化編輯器或 JSON 編輯器來建立政策。我們推薦使用視覺化編輯器，因為它提供一種更加結構化的政策定義方式。如需建立政策的協助，請參閱下方的範例政策。
6. 建立政策之後，請將其連接到一或多個索引：

```
POST _plugins/_ism/add/my-index
{
  "policy_id": "my-policy-id"
}
```

> **Note**
>
> 如果您的網域正在執行舊版 Elasticsearch 版本，請使用 `_opendistro` 而非 `_plugins`。

或者，在 OpenSearch Dashboards 中選取索引，然後選擇 Apply policy (套用政策)。

範例政策

下列範例政策示範如何自動化常見的 ISM 使用案例。

熱至暖再至冷儲存

此範例政策會將索引從熱儲存移至 UltraWarm，最後再移至冷儲存。然後，它會刪除索引。

該索引最初位於 `hot` 狀態。十天之後，ISM 將其移至 `warm` 狀態。80 天後，索引已經超過 90 天，ISM會將索引移至 `cold` 狀態。一年之後，服務會將通知傳送到正在刪除索引的 Amazon Chime 空間，然後永久將其刪除。

請注意，冷索引需要 `cold_delete` 操作，而不是正常的 `delete` 操作。另請注意，資料中必須有明確的 `timestamp_field`，以便使用 ISM 管理冷索引。

```
{
  "policy": {
    "description": "Demonstrate a hot-warm-cold-delete workflow.",
    "default_state": "hot",
    "schema_version": 1,
    "states": [{
        "name": "hot",
        "actions": [],
        "transitions": [{
          "state_name": "warm",
          "conditions": {
            "min_index_age": "10d"
          }
        }]
      },
      {
        "name": "warm",
        "actions": [{
          "warm_migration": {},
          "retry": {
            "count": 5,
            "delay": "1h"
          }
        }],
        "transitions": [{
          "state_name": "cold",
          "conditions": {
            "min_index_age": "90d"
          }
        }]
      },
      {
        "name": "cold",
        "actions": [{
            "cold_migration": {
              "timestamp_field": "<your timestamp field>"
            }
          }
        ],
        "transitions": [{
          "state_name": "delete",
          "conditions": {
             "min_index_age": "365d"
```

```
          }
        }]
      },
      {
        "name": "delete",
        "actions": [{
          "notification": {
            "destination": {
              "chime": {
                "url": "<URL>"
              }
            },
            "message_template": {
              "source": "The index {{ctx.index}} is being deleted."
            }
          }
        },
        {
          "cold_delete": {}
        }]
      }
    ]
  }
}
```

減少複本計數

此範例政策會在七天後將複本計數縮減至零以保留磁碟空間，然後在 21 天後刪除索引。此政策會假設索引不重要且不再接收寫入請求；擁有零個複本會伴隨某種程度的資料遺失風險。

```
{
  "policy": {
    "description": "Changes replica count and deletes.",
    "schema_version": 1,
    "default_state": "current",
    "states": [{
        "name": "current",
        "actions": [],
        "transitions": [{
          "state_name": "old",
          "conditions": {
            "min_index_age": "7d"
          }
```

```
        }]
      },
      {
        "name": "old",
        "actions": [{
          "replica_count": {
            "number_of_replicas": 0
          }
        }],
        "transitions": [{
          "state_name": "delete",
          "conditions": {
            "min_index_age": "21d"
          }
        }]
      },
      {
        "name": "delete",
        "actions": [{
          "delete": {}
        }],
        "transitions": []
      }
    ]
  }
}
```

拍攝索引快照

此範例政策會使用 snapshot 操作在索引包含至少一個文件時立即拍攝快照。repository 是您在 Simple Storage Service (Amazon S3) 中註冊的手動快照儲存庫的名稱。snapshot 是快照的名稱。如需快照必要條件和註冊儲存庫的步驟，請參閱the section called “建立索引快照”。

```
{
  "policy": {
    "description": "Takes an index snapshot.",
    "schema_version": 1,
    "default_state": "empty",
    "states": [{
        "name": "empty",
        "actions": [],
        "transitions": [{
          "state_name": "occupied",
```

```
          "conditions": {
            "min_doc_count": 1
          }
        }]
      },
      {
        "name": "occupied",
        "actions": [{
          "snapshot": {
            "repository": "<my-repository>",
            "snapshot": "<my-snapshot>"
            }
          }],
          "transitions": []
      }
    ]
  }
}
```

ISM 範本

您可以在政策中設定 `ism_template` 欄位，以便當您建立符合範本模式的索引時，該政策可自動連接至該索引。在此範例中，您建立的名稱以 "log" 開頭的任何索引都會自動匹配 ISM 政策 `my-policy-id`：

```
PUT _plugins/_ism/policies/my-policy-id
{
  "policy": {
    "description": "Example policy.",
    "default_state": "...",
    "states": [...],
    "ism_template": {
      "index_patterns": ["log*"],
      "priority": 100
    }
  }
}
```

如需更詳細的範例，請參閱使用 ISM 範本進行自動變換的範例政策。

差異

與 Open Search 和 Elasticsearch 相比，Amazon OpenSearch Service 的 ISM 有幾個不同之處。

ISM 操作

- OpenSearch Service 支援三個獨特的 ISM 操作，`warm_migration`、`cold_migration` 以及 `cold_delete`。
  - 如果您的網域已啟用 UltraWarm，則 `warm_migration` 動作會將索引轉換為暖儲存。
  - 如果您的網域啟用了冷儲存，`cold_migration` 動作會將索引轉換為冷儲存，而 `cold_delete` 動作則會從冷存儲中刪除索引。

  即使其中一個動作沒有在設定的逾時期限內完成，仍會繼續遷移至熱索引。針對上述其中一個動作設定一個 error_notification，如果動作未在逾時期限內完成，會通知您該動作失敗，但該通知僅供您自己參考。實際操作沒有固有的逾時，會在最終成功或失敗之前持續執行。
- 如果您的網域執行 OpenSearch 或 Elasticsearch 7.4 或更新版本，則 OpenSearch Service 支援 ISM `open` 和 `close` 操作。
- 如果您的網域執行 OpenSearch 或 Elasticsearch 7.7 或更新版本，則 OpenSearch Service 支援 ISM `snapshot` 操作。

冷儲存 ISM 操作

對於冷索引，當您使用下列 ISM API 時，必須指定 `?type=_cold` 參數：

- 新增政策
- 移除政策
- 更新政策
- 重試失敗的索引
- 解釋索引

這些適用於冷索引的 API 具有以下其他不同之處：

- 不支援萬用字元運算子，除非您在結尾處使用。例如，支援 `_plugins/_ism/<add, remove, change_policy, retry, explain>/logstash-*|`，但不支援 `_plugins/_ism/<add, remove, change_policy, retry, explain>/iad-*-prod`。

- 不支援多行索引名稱和模式。例如，支援 `_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs|`，但不支援 `_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs,sample-data`。

ISM 設定

OpenSearch 和 Elasticsearch 可讓您使用 `_cluster/settings` API 變更所有可用的 ISM 設定。在 Amazon OpenSearch Service 中，您只能變更下列 ISM 設定：

- 叢集層級設定：
  - `plugins.index_state_management.enabled`
  - `plugins.index_state_management.history.enabled`
- 索引層級設定：
  - `plugins.index_state_management.rollover_alias`

教學課程：自動化索引狀態管理程序

本教學課程示範如何實作 ISM 政策，該政策可自動化例行索引管理任務並將它們套用至索引和索引模式。

Amazon OpenSearch Service 中的索引狀態管理 (ISM) 可讓您自動化週期性索引管理活動，因此您可以避免使用其他工具來管理索引生命週期。您可以建立一個政策，根據索引期限、大小和其他條件，全部從 Amazon OpenSearch Service 網域內自動化這些操作。

OpenSearch Service 支援三種儲存層：用於主動寫入和低延遲分析的預設「熱」狀態、用於高達 3 PB 唯讀資料的 UltraWarm，以及用於無限長期存檔的冷儲存。

本教學課程提供在每日索引中處理時間序列資料的範例使用案例。在本教學課程中，您可設定政策，以在 24 小時後拍攝每個所連接索引的自動快照。然後，它會在兩天後將索引從預設的熱狀態遷移至 UltraWarm 儲存，在 30 天後遷移至冷儲存，最後在 60 天後刪除索引。

先決條件

- 您的 OpenSearch Service 網域必須執行 Elasticsearch 6.8 或更新版本。
- 您的網域必須已啟用 UltraWarm 和冷儲存。
- 必須為您的網域註冊手動快照儲存庫。

- 您的使用者角色需要足夠的許可才能存取 OpenSearch Service 主控台。如有必要，請驗證並設定網域的存取權。

步驟 1：設定 ISM 政策

首先，在 OpenSearch Dashboards 中設定 ISM 政策。

1. 從 OpenSearch Service 主控台的網域儀表板中，導覽至 OpenSearch Dashboards URL，並使用主要使用者名稱和密碼登入。URL 遵循此格式：*domain-endpoint*/_dashboards/。
2. 在 OpenSearch Dashboards 中，選擇 Add sample data (新增範例資料)，然後新增一個或多個範例索引至您的網域。
3. 開啟左側導覽面板，然後依序選擇 Index Management (索引管理) 和 Create policy (建立政策)。
4. 將政策命名為 `ism-policy-example`。
5. 將預設政策取代為以下政策：

```
{
  "policy": {
    "description": "Move indexes between storage tiers",
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [],
        "transitions": [
          {
            "state_name": "snapshot",
            "conditions": {
              "min_index_age": "24h"
            }
          }
        ]
      },
      {
        "name": "snapshot",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
              "delay": "30m"
```

```
            },
            "snapshot": {
              "repository": "snapshot-repo",
              "snapshot": "ism-snapshot"
            }
          }
        ],
        "transitions": [
          {
            "state_name": "warm",
            "conditions": {
              "min_index_age": "2d"
            }
          }
        ]
      },
      {
        "name": "warm",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
              "delay": "1h"
            },
            "warm_migration": {}
          }
        ],
        "transitions": [
          {
            "state_name": "cold",
            "conditions": {
              "min_index_age": "30d"
            }
          }
        ]
      },
      {
        "name": "cold",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
```

```
              "delay": "1h"
            },
            "cold_migration": {
              "start_time": null,
              "end_time": null,
              "timestamp_field": "@timestamp",
              "ignore": "none"
            }
          }
        ],
        "transitions": [
          {
            "state_name": "delete",
            "conditions": {
              "min_index_age": "60d"
            }
          }
        ]
      },
      {
        "name": "delete",
        "actions": [
          {
            "cold_delete": {}
          }
        ],
        "transitions": []
      }
    ],
    "ism_template": [
      {
        "index_patterns": [
          "index-*"
        ],
        "priority": 100
      }
    ]
  }
}
```

> **Note**
>
> `ism_template` 欄位會自動將政策連接至符合其中一個指定 `index_patterns` 的任何新建立的索引。在此案例中，則為所有以 `index-` 開頭的索引。您可以修改此欄位以符合環境中的索引格式。如需詳細資訊，請參閱 ISM 範本。

6. 在政策的 `snapshot` 區段中,將 *`snapshot-repo`* 取代為您為網域註冊的快照儲存庫的名稱。您也可以選擇取代 *`ism-snapshot`*，這將是建立快照時的名稱。
7. 選擇建立。現在可以在 State management policies (狀態管理政策) 頁面上看到該政策。

步驟 2：將政策連接至一個或多個索引

既然您已建立政策，請將其連接至叢集中的一個或多個索引。

1. 轉至 Hot indicies (熱索引) 標籤並搜尋 `opensearch_dashboards_sample`，這會列出您在步驟 1 中新增的所有範例索引。
2. 選取所有索引並選擇 Apply policy (套用政策)，然後選擇您剛剛建立的 ism-policy-example 政策。
3. 選擇套用。

您可以在 Policy managed indices (政策管理的索引) 頁面上監控索引在各種狀態間的移動。

使用索引彙總來匯總 Amazon OpenSearch Service 中的索引

Amazon OpenSearch Service 中的索引彙總可讓您定期將舊資料彙總至摘要索引，以降低儲存成本。

您可以選擇感興趣的欄位，並使用索引彙總來建立一個新的索引，其中只有這些欄位彙總到更粗糙的時間儲存貯體。您可以很低的成本存放幾個月或幾年的歷史資料，且擁有相同的查詢效能。

索引彙總需要 OpenSearch 或 Elasticsearch 7.9 或更高版本。

> **Note**
>
> 本文件可協助您開始在 Amazon OpenSearch Service 中建立索引彙總任務。如需完整的文件，包括所有可用設定的清單和完整的 API 參考，請參閱 OpenSearch 文件中的索引彙總。

建立索引彙總任務

若要開始使用，請在 OpenSearch Dashboards 中選擇 Index Management (索引管理)。選取 Rollup Jobs (彙總任務)，然後選擇 Create rollup job (建立彙總任務)。

步驟 1：設定索引

設定來源和目標索引。來源索引是您想要彙總的索引。目標索引是儲存索引彙總結果的位置。

建立索引彙總任務之後，您無法變更索引選擇。

步驟 2：定義彙總和指標

選取含有您要總計之彙總 (詞彙和直方圖) 和指標 (平均、總和、最大值、最小值和值計數) 的屬性。確保沒有新增很多高精細屬性，因為不會節省太多空間。

步驟 3：指定排程

指定排程，以便在擷取索引時進行彙總。預設啟用索引彙總任務。

步驟 4：檢閱和建立

檢閱您的組態，然後選取 Create (建立)。

步驟 5：搜尋目標索引

您可以使用標準 `_search` API 來搜尋目標索引。您無法存取目標索引中資料的內部結構，因為外掛程式會自動在後台重寫查詢以適應目標索引。這是為了確保您可以對來源索引和目標索引使用相同的查詢。

若要查詢目標索引，請將 `size` 設定為 0：

```
GET target_index/_search
{
  "size": 0,
  "query": {
    "match_all": {}
  },
  "aggs": {
    "avg_cpu": {
      "avg": {
```

```
        "field": "cpu_usage"
      }
    }
  }
}
```

> **Note**
>
> OpenSearch 2.2 版及更新版本支援在一個請求中搜尋多個彙總索引。2.2 版之前的 OpenSearch 版本和舊版 Elasticsearch OSS 版本僅在每次搜尋時支援一個彙總索引。

在 Amazon OpenSearch Service 中轉換索引

雖然索引彙總任務可讓您透過將舊資料彙總為壓縮索引來減少資料精細度，但轉換任務可讓您建立以特定欄位為中心的不同摘要檢視，以便以不同的方式視覺化或分析資料。

索引轉換具有 OpenSearch Dashboards 使用者介面和 REST API。該功能需要 OpenSearch 1.0 或更高版本。

> **Note**
>
> 本文件提供索引轉換的簡短概觀，協助您在 Amazon OpenSearch Service 網域上開始使用它。如需完整文件和 REST API 參考，請參閱開放原始碼 OpenSearch 文件中的索引轉換。

建立索引轉換任務

如果叢集中沒有任何資料，請使用 OpenSearch Dashboards 中的範例航班資料，以嘗試轉換任務。新增資料後，啟動 OpenSearch Dashboards。然後依次選擇 Index Management (索引管理)、Transform Jobs (轉換任務) 以及 Create Transform Job (建立轉換任務)。

步驟 1：選擇索引

在 Indices (索引) 區段中，選取來源索引和目標索引。您可以選取現有目標索引，也可以輸入目標索引的名稱來建立新的目標索引。

如果您只想轉換來源索引的子集，請選擇 Add Data Filter (新增資料篩選)，並使用 OpenSearch 查詢 DSL 來指定來源索引的子集。

步驟 2：選擇欄位

選擇索引後，請選擇您要在轉換任務中使用的欄位，以及是否使用分組或彙總。

- 您可以使用分組將資料放入轉換索引中的單獨儲存貯體中。例如，如果您想要對範例航班資料中的所有機場目的地進行分組，請將 `DestAirportID` 欄位分組到 `DestAirportID_terms` 欄位的目標欄位中，在轉換任務完成後，您可以在已轉換的索引中找到已分組的機場 ID。
- 另一方面，彙總可讓您執行簡單的計算。例如，您可能會在轉換任務中包含彙總，以定義計算所有機票總和的新欄位 `sum_of_total_ticket_price`。然後，您可以分析已轉換索引中的新資料。

步驟 3：指定排程

轉換任務預設為啟用，並依排程執行。若要轉換執行間隔，請以分鐘、小時或天數來指定間隔。

步驟 4：檢閱和監控

檢閱您的組態，然後選取 Create (建立)。然後監控 Transform job status (轉換任務狀態) 欄。

步驟 5：搜尋目標索引

任務完成後，您可以使用標準 `_search` API 來搜尋目標索引。

例如，在執行根據 `DestAirportID` 欄位轉換航班資料的轉換任務後，您可以執行下列請求以傳回值為 `SFO` 的所有欄位：

```
GET target_index/_search
{
  "query": {
    "match": {
      "DestAirportID_terms" : "SFO"
    }
  }
}
```

Amazon OpenSearch Service 的跨叢集複寫

透過 Amazon OpenSearch Service 中的跨叢集複寫，您可以將使用者索引、映射和中繼資料從一個 OpenSearch Service 網域複寫到另一個網域。使用跨叢集複寫有助於確保災難復原是否會中斷，並可讓您跨遠距資料中心複寫資料，以減少延遲。您需要為網域之間傳輸的資料支付標準 AWS 資料傳輸費用。

跨叢集複寫遵循主動-被動複寫模型，其中本機或跟隨者索引會從遠端或領導索引提取資料。領導索引是指資料來源，或是您要從中複寫資料的索引。跟隨者索引是指資料的目標，或是您想要複寫資料的索引。

跨叢集複寫適用於執行 Elasticsearch 7.10 或 OpenSearch 1.1 或更高版本的網域。

> **Note**
>
> 本文件說明如何從 Amazon OpenSearch Service 的角度設定跨叢集複寫。這包括使用 AWS 管理主控台 來設定跨叢集連線，這無法在自我管理的 OpenSearch 叢集上進行。如需完整文件，包括設定參考和完整的 API 參考，請參閱 OpenSearch 文件中的跨叢集複寫。

主題

- 限制
- 先決條件
- 許可需求
- 設定跨叢集連線
- 開始複寫
- 確認複寫
- 暫停和繼續複寫
- 停止複寫
- 自動追蹤
- 升級連線的網域

限制

跨叢集複寫有下列限制：

- 您無法在 Amazon OpenSearch Service 網域或自我管理的 OpenSearch 或 Elasticsearch 叢集之間複寫資料。
- 您無法將索引從追隨網域複寫到另一個追隨網域。如果您想要將索引複寫到多個跟隨網域，您只能從單一領導網域複寫索引。
- 一個網域可以透過傳入和傳出連線的組合，連線到最多 20 個其他網域。
- 當您最初設定跨叢集連線時，領導網域的版本必須與追隨網域相同或更高。

- 您無法使用 CloudFormation 來連接網域。
- 您無法在 M3 或高載 (T2 和 T3) 執行個體上使用跨叢集複寫。
- 您無法在 UltraWarm 索引或冷索引之間複寫資料。兩個索引都必須位於熱儲存中。
- 當您從領導網域刪除索引時，跟隨網域上的對應索引不會自動刪除。
- 預設 和選擇加入 之間不支援跨叢集複寫。兩個網域都必須位於預設區域或選擇加入區域。

先決條件

設定跨叢集複寫前，請確認網域符合下列需求：

- Elasticsearch 7.10 或 OpenSearch 1.1 或更高版本
- 已啟用精細存取控制
- 已啟用節點對節點加密
- 領導者索引必須`index.soft_deletes.enabled`設定為 `true`。在 Elasticsearch 7.0 或 OpenSearch 1.0 及更新版本中建立的索引預設會啟用此設定。不過，在 Elasticsearch 6.x 中建立，然後升級的索引會保留 。 `soft_deletes=false`若要複寫這類索引，您必須先重新編製索引。

  若要檢查索引是否已啟用軟刪除：

  ```
  GET <index-name>/_settings?
  include_defaults=true&flat_settings=true&filter_path=*.settings.index.soft_deletes.enabled
  ```

  如果 `soft_deletes`是 `false`，請在開始複寫之前將資料重新索引為新索引。

許可需求

若要開始複寫，您必須在遠端 (領導) 網域上提供 `es:ESCrossClusterGet` 許可。我們建議在遠端網域上使用下列 IAM 政策。此政策也可讓您執行其他操作，例如編製文件索引和執行標準搜尋：

JSON

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

```
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/leader-domain/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/leader-domain"
    }
  ]
}
```

確認已針對 `/leader-domain` (而不是 `/leader-domain/*`) 套用 `es:ESCrossClusterGet` 許可。

為了讓非管理員使用者執行複寫活動，他們也必須對應至適當的許可。大多數許可均對應特定的 REST API 操作。例如，`indices:admin/plugins/replication/index/_resume` 許可能讓您恢復索引複寫。如需完整的許可清單，請參閱 OpenSearch 文件中的複寫許可。

> **Note**
>
> 開始複寫和建立複寫規則的命令屬於特殊情況。因為這兩者會叫用領導和追蹤網域上的背景處理程序，所以您必須在請求中傳遞 `leader_cluster_role` 和 `follower_cluster_role`。OpenSearch Service 在所有後端複寫任務中會使用這些角色。如需有關對應和使用這些角色的詳細資訊，請參閱 OpenSearch 文件中的對應領導和追蹤叢集角色。

設定跨叢集連線

若要將索引從一個網域複寫到另一個網域，您需要設定網域之間的跨叢集連線。連線網域最簡單的方法是透過網域儀表板的 Connections (連線) 索引標籤。您也可以使用組態 API 或 AWS CLI。因為跨叢集複寫遵循「提取」模型，您會從追蹤網域啟動連線。

> **Note**
>
> 如果您先前已連線兩個網域以執行跨叢集搜尋，則無法使用相同的連線進行複寫。在主控台中，連線將標記為 SEARCH_ONLY。若要在兩個之前連線的網域之間執行複寫，您必須刪除連線，再重新建立連線。完成此操作後，該連線可用於跨叢集搜尋和跨叢集複寫。

如何設定連線

1. Amazon OpenSearch Service 主控台中，選取追蹤網域，前往 Connections (連線) 索引標籤，然後選擇 Request (請求)。
2. 對於 Connection alias (連線別名)，輸入連線的名稱。
3. 選擇連線到您 AWS 帳戶 和 區域中的網域，或另一個帳戶或區域中的網域。
   - 若要連線到 AWS 帳戶 和 區域中的網域，請選取網域，然後選擇請求。
   - 若要連線到另一個 AWS 帳戶 或區域中的網域，請指定遠端網域的 ARN，然後選擇請求。

OpenSearch Service 會驗證連線請求。如果網域不相容，連線會失敗。如果驗證成功，則會將請求傳送至目的地網域進行核准。當目的地網域核准請求時，您就可以開始複寫。

跨叢集複寫支援雙向複寫。這表示您可以建立從網域 A 到網域 B 的傳出連線，以及從網域 B 到網域 A 的另一個傳出連線。然後，您可以設定複寫，讓網域 A 遵循網域 B 中的索引，而網域 B 遵循網域 A 中的索引。

開始複寫

建立跨叢集連線之後，您就可以開始複寫資料。首先，在主導領域上建立要複寫的索引：

```
PUT leader-01
```

若要複寫該索引，請將此命令傳送至追蹤網域：

```
PUT _plugins/_replication/follower-01/_start
{
   "leader_alias": "connection-alias",
   "leader_index": "leader-01",
   "use_roles":{
      "leader_cluster_role": "all_access",
      "follower_cluster_role": "all_access"
   }
}
```

您可以在網域儀表板的 Connections (連線) 索引標籤上找到連線別名。

為了簡單起見，此範例假設管理員發出請求，並針對 `leader_cluster_role` 和 `follower_cluster_role` 使用 `all_access`。不過，在生產環境中，我們建議您同時在領導和追蹤索引上建立複寫使用者，並加以對應。使用者名稱必須相同。如需有關這些角色及其對應方法的詳細資訊，請參閱 OpenSearch 文件中的對應領導和追蹤叢集角色。

確認複寫

若要確認正在進行複寫，請取得複寫狀態：

```
GET _plugins/_replication/follower-01/_status

{
  "status" : "SYNCING",
  "reason" : "User initiated",
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01",
  "syncing_details" : {
    "leader_checkpoint" : -5,
    "follower_checkpoint" : -5,
    "seq_no" : 0
  }
}
```

領導和追蹤檢查點的值以負整數開始，並反映您擁有的碎片數量 (-1 代表一個碎片，-5 代表五個碎片，依此類推)。這些值會隨著您所做的每項變更而遞增為正整數。如果值是相同的，表示索引已完全同步。您可以使用這些檢查點的值來測量跨網域的複寫延遲。

若要進一步驗證複寫，請將文件新增至領導索引：

```
PUT leader-01/_doc/1
{
   "Doctor Sleep":"Stephen King"
}
```

同時確認文件顯示在追蹤索引上：

```
GET follower-01/_search

{
    ...
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "follower-01",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "Doctor Sleep" : "Stephen King"
        }
      }
    ]
  }
}
```

暫停和繼續複寫

如果您需要修復問題或減少領導網域的負載，可以暫時停止複寫。將此請求傳送至追蹤網域。請務必包含空的請求主體：

```
POST _plugins/_replication/follower-01/_pause
{}
```

隨後獲取狀態，確保複寫已暫停：

```
GET _plugins/_replication/follower-01/_status

{
  "status" : "PAUSED",
  "reason" : "User initiated",
```

```
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01"
}
```

當您完成變更後，請繼續複寫。將此請求傳送至追蹤網域。請務必包含空的請求主體：

```
POST _plugins/_replication/follower-01/_resume
{}
```

您無法在已暫停超過 12 小時後繼續複寫。您必須停止複寫、刪除追蹤者索引，然後重新啟動領導者的複寫。

停止複寫

當您完全停止複寫時，追蹤索引會取消追蹤領導索引，並成為標準索引。停止複寫後，就無法重新開始複寫。

從追蹤網域停止複寫。請務必包含空的請求主體：

```
POST _plugins/_replication/follower-01/_stop
{}
```

自動追蹤

您可以針對單一領導領域定義一組複寫規則，以自動複寫符合指定模式的索引。當您在領導網域上建立與其中一個模式相符的索引時 (例如，books*)，會在追隨網域上建立相符的追隨索引。OpenSearch Service 會複寫與模式相符的所有現有索引以及您建立的新索引。不會複寫追隨網域上已經存在的索引。

若要複寫所有索引 (系統建立的索引以及追隨網域上已存在的索引除外)，應使用萬用字元 (*) 模式。

建立複寫規則

在追蹤網域上建立複寫規則，並指定跨叢集連線的名稱：

```
POST _plugins/_replication/_autofollow
{
   "leader_alias" : "connection-alias",
   "name": "rule-name",
```

```
   "pattern": "books*",
   "use_roles":{
      "leader_cluster_role": "all_access",
      "follower_cluster_role": "all_access"
   }
}
```

您可以在網域儀表板的 Connections (連線) 索引標籤上找到連線別名。

為了簡單起見，此範例假設管理員發出請求，並使用 all_access 作為領導和追蹤網域角色。不過，在生產環境中，我們建議您同時在領導和追蹤索引上建立複寫使用者，並加以對應。使用者名稱必須相同。如需有關這些角色及其對應方法的詳細資訊，請參閱 OpenSearch 文件中的對應領導和追蹤叢集角色。

若要擷取網域上現有複寫規則的清單，請使用自動追蹤統計 API 操作。

若要測試規則，請建立與領導網域上模式相符的索引：

```
PUT books-are-fun
```

同時檢查索引副本是否出現在追蹤網域上：

```
GET _cat/indices

health status index          uuid                   pri rep docs.count docs.deleted
 store.size pri.store.size
green  open   books-are-fun  ldfHO78xYYdxRMULuiTvSQ     1   1          0            0
     208b           208b
```

刪除複寫規則

當您刪除複寫規則時，OpenSearch Service 會停止複寫與模式相符的新索引，但會繼續現有的複寫活動，直到您停止複寫這些索引為止。

從追蹤網域刪除複寫規則：

```
DELETE _plugins/_replication/_autofollow
{
   "leader_alias" : "connection-alias",
   "name": "rule-name"
}
```

升級連線的網域

若要升級具有跨叢集連線之兩個網域的引擎版本，請先升級追隨網域，然後再升級領導網域。請勿刪除它們之間的連線，否則複寫會暫停，您將無法繼續。

使用遠端重新索引遷移 Amazon OpenSearch Service 索引

遠端重新索引可讓您將索引從一個 Amazon OpenSearch Service 網域複製到另一個網域。您可以從任何 OpenSearch Service 網域或自我管理的 OpenSearch 和 Elasticsearch 叢集遷移索引。

遠端網域和索引是指資料來源，或是您要從中複製資料的網域和索引。本機網域和索引是指資料的目標，或是您要複製資料的網域和索引。

遠端重新索引需要在本機網域上使用 OpenSearch 1.0 或更新版本，或 Elasticsearch 6.7 或更新版本。遠端網域必須與本機網域較低或相同的主要版本。Elasticsearch 版本被認為低於 OpenSearch 版本，這表示您可以將資料從 Elasticsearch 網域重新索引至 OpenSearch 網域。在相同的主要版本中，遠端網域可以是任何次要版本。例如，支援從 Elasticsearch 7.10.x 到 7.9 的遠端重新索引，但不支援 OpenSearch 1.0 到 Elasticsearch 7.10.x。

> **Note**
>
> 本文件說明如何在 Amazon OpenSearch Service 網域之間重新索引資料。如需 `reindex`操作的完整文件，包括詳細步驟和支援的選項，請參閱 OpenSearch 文件中的 Reindex 文件。

主題

- 先決條件
- 在 OpenSearch Service 網際網路網域之間重新索引資料
- 當遠端位於 VPC 中時，在 OpenSearch Service 網域之間重新索引資料
- 在非 OpenSearch Service 網域之間重新建立資料索引
- 重新索引大型資料集
- 遠端重新索引設定

先決條件

遠端重新索引有以下要求：

- 遠端網域必須可從本機網域存取。對於位於 VPC 中的遠端網域，本機網域必須具有 VPC 的存取權。此程序會因網路組態而有所不同，但可能涉及連線至 VPN 或受管網路，或使用原生 VPC 端點連線。如需詳細資訊，請參閱 the section called “VPC 支援”。
- 請求必須由遠端網域授權，就像任何其他 REST 請求一樣。如果遠端網域已啟用精細存取控制，您必須具有在遠端網域上執行重新索引並讀取本機網域索引的許可。如需更多安全性考量，請參閱 the section called “精細定義存取控制”。
- 建議您先在本機網域上建立具有所需設定的索引，再開始重新索引程序。
- 如果您的網域使用資料節點的 T2 或 T3 執行個體類型，則無法使用遠端重新索引。

在 OpenSearch Service 網際網路網域之間重新索引資料

最基本的案例是遠端索引與具有可公開存取端點的 AWS 區域 本機網域位於相同位置，而且您已簽署 IAM 憑證。

從遠端網域中，指定要重新索引的遠端索引，以及要重新索引的本機索引：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

您必須在遠端網域端點結尾新增 443，以進行驗證檢查。

若要確認索引已複製到本機網域，請將此請求傳送至本機網域：

```
GET local_index/_search
```

如果遠端索引所在的區域與本機網域不同，請傳遞其區域名稱，例如在此範例請求中：

```
POST _reindex
```

```
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "region": "eu-west-1"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

如果發生隔離的區域，例如 AWS GovCloud (US) 或中國區域，端點可能無法存取，因為在這些區域中無法辨識您的 IAM 使用者。

如果遠端網域使用基本身分驗證進行保護，請指定使用者名稱和密碼：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

當遠端位於 VPC 中時，在 OpenSearch Service 網域之間重新索引資料

每個 OpenSearch Service 網域都由自己的內部虛擬私有雲端 (VPC) 基礎設施組成。當您在現有的 OpenSearch Service VPC 中建立新網域時，會為 VPC 中的每個資料節點建立彈性網路界面。

由於遠端重新索引操作是從遠端 OpenSearch Service 網域執行，因此在自己的私有 VPC 內，您需要一種方法來存取本機網域的 VPC。您可以使用內建 VPC 端點連線功能透過 建立連線 AWS PrivateLink，或設定代理來達成此目的。

如果您的本機網域使用 OpenSearch 1.0 版或更新版本，您可以使用 主控台或 AWS CLI 來建立 AWS PrivateLink 連線。 AWS PrivateLink 連線可讓本機 VPC 中的資源在相同 內私下連線至遠端 VPC 中的資源 AWS 區域。

若要建立 VPC 端點連線，要重新編製索引的來源網域必須位於本機 VPC 中，而來源和目的地網域都必須位於相同的 中 AWS 區域。

使用 重新索引資料 AWS 管理主控台

您可以搭配 主控台使用遠端重新索引，在共用 VPC 端點連線的兩個網域之間複製索引。

1. 在 https://console.aws.amazon.com/aos/ 瀏覽至 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中選擇 Domains (網域)。
3. 選取本機網域或您要複製資料的網域。這會開啟網域詳細資訊頁面。選擇一般資訊下方的連線索引標籤，然後選擇請求。
4. 在請求連線頁面上，為您的連線模式選取 VPC 端點連線，然後輸入其他相關詳細資訊。這些詳細資訊包括遠端網域，這是您要從中複製資料的網域。然後，選擇 Request (請求)。
5. 導覽至遠端網域的詳細資訊頁面，選擇連線索引標籤，然後尋找傳入連線資料表。選取剛剛從中建立連線的網域名稱 (本機網域) 旁的核取方塊。選擇 Approve (核准)。
6. 瀏覽至本機網域，選擇 Connections (連線) 標籤，然後找到 Outbound connections (傳出連線) 資料表。兩個網域之間的連線處於作用中狀態之後，資料表 Endpoint (端點) 欄中的端點就會變為可用。複製端點。
7. 開啟本機網域的儀表板，然後在左側導覽中選擇 Dev Tools (開發工具)。若要確認遠端網域索引尚未存在於您的本機網域上，請執行下列 GET 請求。以您自己的索引名稱取代 *remote-domain-index-name*。

   ```
   GET remote-domain-index-name/_search
   {
      "query":{
         "match_all":{}
      }
   }
   ```

   在輸出中，您應該會看到錯誤，指出找不到索引。
8. 如下所示，在 GET 請求下方，建立 POST 要求並使用端點作為遠端主機。

   ```
   POST _reindex
   {
   ```

```
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

執行此請求。

9. 再次執行 GET 請求。輸出現在應指出本機索引的存在。您可以查詢此索引，以確認 OpenSearch 從遠端索引複製所有資料。

使用 OpenSearch Service API 操作重新索引資料

您可以搭配 API 使用遠端重新索引，在共用 VPC 端點連線的兩個網域之間複製索引。

1. 使用 CreateOutboundConnection API 操作，請求從本機網域到遠端網域的新連線。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/cc/outboundConnection

{
   "ConnectionAlias": "remote-reindex-example",
   "ConnectionMode": "VPC_ENDPOINT",
   "LocalDomainInfo": {
      "AWSDomainInformation": {
         "DomainName": "local-domain-name",
         "OwnerId": "aws-account-id",
         "Region": "region"
      }
   },
   "RemoteDomainInfo": {
      "AWSDomainInformation": {
         "DomainName": "remote-domain-name",
         "OwnerId": "aws-account-id",
         "Region": "region"
      }
   }
```

```
}
```

您會在回應ConnectionId中收到 。儲存此 ID 以用於下一個步驟。

2. 使用 AcceptInboundConnection API 操作搭配您的連線 ID 來核准來自本機網域的請求。

```
PUT https://es.region.amazonaws.com/2021-01-01/opensearch/cc/
inboundConnection/ConnectionId/accept
```

3. 使用 DescribeOutboundConnections API 操作來擷取遠端網域的端點。

```
{
    "Connections": [
        {
            "ConnectionAlias": "remote-reindex-example",
            "ConnectionId": "connection-id",
            "ConnectionMode": "VPC_ENDPOINT",
            "ConnectionProperties": {
                "Endpoint": "connection-endpoint"
            },
            ...
        }
    ]
}
```

儲存要在步驟 5 中使用的####。

4. 若要確認遠端網域索引尚未存在於您的本機網域上，請執行下列 GET 請求。以您自己的索引名稱取代 *remote-domain-index-name*。

```
GET local-domain-endpoint/remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

在輸出中，您應該會看到錯誤，指出找不到索引。

5. 建立 POST 請求並使用端點做為遠端主機，如下所示。

```
POST local-domain-endpoint/_reindex
{
```

```
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

執行此請求。

6. 再次執行 GET 請求。輸出現在應指出本機索引的存在。您可以查詢此索引，以確認 OpenSearch 從遠端索引複製所有資料。

如果遠端網域託管在 VPC 內，而您不想使用 VPC 端點連線功能，您必須使用可公開存取的端點設定代理。在此情況下，OpenSearch Service 需要公有端點，因為它無法將流量傳送到您的 VPC。

當您在 VPC 模式下執行網域時，一或多個端點會放置在 VPC 中。不過，這些端點僅適用於傳入 VPC 內網域的流量，而且不允許傳入 VPC 本身的流量。

遠端重新索引命令是從本機網域執行，因此來源流量無法使用這些端點存取遠端網域。因此，在此使用案例中需要代理。代理網域必須擁有由公有憑證授權單位 (CA) 所簽署的憑證。不支援自我簽署或私有 CA 簽署憑證。

在非 OpenSearch Service 網域之間重新建立資料索引

如果遠端索引託管在 OpenSearch Service 外部，例如在自我管理的 EC2 執行個體中，請將 `external` 參數設定為 `true`：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password",
      "external": true
    },
```

```
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

在此情況下，僅支援具有使用者名稱和密碼的基本身分驗證。遠端網域必須具有可公開存取的端點（即使與本機 OpenSearch Service 網域位於相同的 VPC 中），以及由公有 CA 簽署的憑證。不支援自我簽署或私有 CA 簽署憑證。

重新索引大型資料集

遠端重新索引會使用下列預設值，將捲動請求傳送至遠端網域：

- 搜尋 5 分鐘內容
- 通訊端逾時 30 秒
- 批次大小為 1,000

我們建議您調整這些參數以容納您的資料。對於大型文件，請考慮較小的批次大小和/或較長的逾時。如需詳細資訊，請參閱分頁結果。

```
POST _reindex?pretty=true&scroll=10h&wait_for_completion=false
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "socket_timeout": "60m"
    },
    "size": 100,
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

我們也建議您將下列設定新增至本機索引，以獲得更好的效能：

```
PUT local_index
```

```
{
  "settings": {
    "refresh_interval": -1,
    "number_of_replicas": 0
  }
}
```

重新索引程序完成後，您可以設定想要的複本數，並移除重新整理間隔設定。

若要僅重新索引您透過查詢選取的文件子集，請將此請求傳送至本機網域：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index",
    "query": {
      "match": {
        "field_name": "text"
      }
    }
  },
  "dest": {
    "index": "local_index"
  }
}
```

遠端重新索引不支援分割，因此您無法為同一個請求並行執行多個捲動操作。

遠端重新索引設定

除了標準重新索引選項，OpenSearch Service 還支援以下選項：

選項	有效值	Description	必要
外部	Boolean	如果遠端網域不是 OpenSearch Service 網域，或者如果您要在兩個 VPC 網域之間	否

選項	有效值	Description	必要
		重新編製索引，請指定 為 `true`。	
region	String	如果遠端網域位於不同的區域，請指定區域名稱。	否

使用資料串流管理 Amazon OpenSearch Service 中的時間序列資料

管理時間序列資料的典型工作流程包含多個步驟，例如建立變換索引別名、定義寫入索引，以及定義備份索引的通用映射和設定。

Amazon OpenSearch Service 中的資料串流有助於簡化此初始設定程序。資料串流可立即用於以時間為基礎的資料，例如通常只附加的應用程式日誌。

資料串流需要 OpenSearch 1.0 版或更新版本。

> **Note**
>
> 本文件提供基本步驟，協助您開始使用 Amazon OpenSearch Service 網域上的資料串流。如需完整文件，請參閱 OpenSearch 文件中的資料串流。

資料串流入門

資料串流內部由多個後備索引組成。搜尋請求會路由至所有後備索引，而索引請求則會路由至最新的寫入索引。

步驟 1：建立索引範本

若要建立資料串流，首先需要建立一個索引範本，設定一組索引作為資料串流。`data_stream` 物件表示它是一個資料串流，而不是一個常規的索引範本。索引模式與資料串流的名稱匹配：

```
PUT _index_template/logs-template
{
  "index_patterns": [
```

```
    "my-data-stream",
    "logs-*"
  ],
  "data_stream": {},
  "priority": 100
}
```

在這種情況下，每個擷取的文件都必須有 @timestamp 欄位。您也可以將自己的自訂時間戳記欄位定義為 data_stream 物件中的一個屬性：

```
PUT _index_template/logs-template
{
  "index_patterns": "my-data-stream",
  "data_stream": {
    "timestamp_field": {
      "name": "request_time"
    }
  }
}
```

步驟 2：建立資料串流

建立索引範本之後，您可以直接開始擷取資料，而不需要建立資料串流。

因為我們有一個與 data_stream 物件匹配的索引範本，所以 OpenSearch 會自動建立資料串流：

```
POST logs-staging/_doc
{
  "message": "login attempt failed",
  "@timestamp": "2013-03-01T00:00:00"
}
```

步驟 3：將資料擷取至資料串流

若要將資料擷取至資料串流，您可以使用一般索引 API。確保您索引的每個文件都有一個時間戳記欄位。如果您嘗試擷取沒有時間戳記欄位的文件，會收到錯誤。

```
POST logs-redis/_doc
{
  "message": "login attempt",
  "@timestamp": "2013-03-01T00:00:00"
```

```
}
```

步驟 4：搜尋資料串流

您可以像搜尋一般索引或索引別名一樣搜尋資料串流。搜尋操作適用於所有後備索引 (串流中存在的所有資料)。

```
GET logs-redis/_search
{
  "query": {
    "match": {
      "message": "login"
    }
  }
}
```

步驟 5：變換資料串流

您可以設定索引狀態管理 (ISM)政策來自動化資料串流的變換程序。ISM 政策會在建立後備索引時便套用至它們。當您將政策與資料串流產生關聯時，它只會影響該資料串流的未來後備索引。您也不需要提供 `rollover_alias` 設定，因為 ISM 政策會從後備索引中推斷此資訊。

> **Note**
>
> 如果您將備份索引遷移至冷儲存，OpenSearch 會從資料串流中移除此索引。即使您將索引移回 UltraWarm，索引會保持獨立，而不是原始資料串流的一部分。從資料串流移除索引後，搜尋串流不會傳回索引中的任何資料。

> **Warning**
>
> 資料串流的寫入索引無法遷移至冷儲存。如果您想要將資料串流中的資料遷移至冷儲存，則必須在遷移之前輪換資料串流。

步驟 6：管理 OpenSearch Dashboards 中的資料串流

若要在 OpenSearch Dashboards 中管理資料串流，請開啟 OpenSearch Dashboards，選擇 Index Management (索引管理)，選取 Indices (索引) 或 Policy managed indices (政策管理的索引)。

步驟 7：刪除資料串流

刪除操作首先刪除資料串流的後備索引，然後刪除資料串流本身。

若要刪除資料串流及其所有隱藏的後備索引：

```
DELETE _data_stream/name_of_data_stream
```

在 Amazon OpenSearch Service 中監控資料

透過提醒和異常偵測，在 Amazon OpenSearch Service 中主動監控您的資料。設定警示以便在資料超過特定閾值時接收通知。異常偵測會使用機器學習來自動偵測串流資料中的任何極端值。您可以將異常偵測與提醒配對，以確保在偵測到異常時立即通知您。

主題

- 在 Amazon OpenSearch Service 中設定提醒
- Amazon OpenSearch Service 中的異常偵測

在 Amazon OpenSearch Service 中設定提醒

在 Amazon OpenSearch Service 中設定提醒，以便在一或多個索引的資料符合特定條件時收到通知。例如，您可能希望在應用程式於 1 個小時內記錄超過 5 個以上的 HTTP 503 錯誤時收到電子郵件，或者在過去 20 分鐘內未有任何新文件編製成索引時通知開發人員。

提醒需要 OpenSearch 或 Elasticsearch 6.2 或更新版本。

> **Note**
>
> 本文件提供提醒的簡短概觀，並強調 Amazon OpenSearch Service 網域上的提醒與開放原始碼 OpenSearch 叢集上的提醒有何不同。如需完整的提醒文件，包括完整的 API 參考、複合監視器的可用請求欄位清單，以及可用觸發和動作變數的說明，請參閱 OpenSearch 文件中的提醒。

主題

- 提醒許可
- 提醒入門
- 通知
- 差異

提醒許可

提醒支援精細存取控制。如需混合和匹配許可以符合您的使用案例的詳細資訊，請參閱 OpenSearch 文件中的提醒安全。

若要存取 OpenSearch Dashboards 中的提醒頁面，您必須至少對應至`alerting_read_access`預先定義的角色，或獲得同等許可。此角色授予檢視提醒、目的地和監視器的許可，但不認可提醒或修改目的地或監視器。

提醒入門

若要建立提醒，您可以設定 monitor (監控)，這是按照定義的排程執行並查詢 OpenSearch 索引的任務。您也可以設定一個或多個觸發條件，這些觸發條件會定義產生事件的條件。最後，您可以設定 action (動作)，這是觸發提醒之後發生的動作。

開始使用提醒

1. 從 OpenSearch Dashboards 主選單中選擇 Alerting (提醒)，並選擇 Choose Alerting (建立提醒)。
2. 建立每個查詢、每個儲存貯體、每個叢集指標或每個文件監控。如需相關說明，請參閱 Create a monitor (建立監控)。
3. 針對 Trigger (觸發條件)，建立一個或多個觸發條件。如需相關說明，請參閱 Create triggers (建立觸發條件)。
4. 針對 Action (動作)，設定提醒的 notification channel (通知頻道)。在 Slack、Amazon Chime、自訂 Webhook 或 Amazon SNS 之間進行選擇。如您所想，通知需要連線到相關頻道。例如，若要通知 Slack 頻道或將自訂 Webhook 傳送至第三方伺服器，您的 OpenSearch Service 網域需要適當的網路連線。自訂 Webhook 必須具有公有 IP 地址，以便 OpenSearch Service 網域向其傳送提醒。

> **Tip**
>
> 當動作成功傳送簡訊之後，您就必須負責保護該簡訊的存取權 (例如，存取 Slack 頻道)。如果您的網域包含敏感資料，請考慮在沒有動作的情況下使用觸發程序，並定期檢查 Dashboards 是否有提醒。

通知

提醒與通知整合，這是 OpenSearch 通知的統一系統。通知可讓您設定要使用的通訊服務，並查看相關統計資料和疑難排解資訊。如需完整文件，請參閱 OpenSearch 文件中的通知。

您的網域必須執行 OpenSearch 2.3 版或更新版本才能使用通知。

> **Note**
>
> OpenSearch 通知獨立於 OpenSearch 服務通知，它提供有關服務軟體更新、自動調整增強功能以及其他重要網域層級資訊的詳細資訊。OpenSearch 通知針對特定的外掛程式。

從 OpenSearch 2.0 版開始，通知頻道取代了提醒目的地。目的地已被正式取代，並且所有提醒通知將透過未來的頻道進行管理。

當您將網域升級至 2.3 版或更新版本時 (由於 2.x 版的 OpenSearch Service 支援從 2.3 版開始)，您現有的目的地會自動移轉至通知頻道。如果目的地無法移轉，監控將繼續使用它，直到監控移轉到通知頻道為止。如需更多資訊，請參閱《OpenSearch 文件》中關於目的地的問題。

若要開始使用通知，請登入 OpenSearch Dashboards，然後選擇 Notifications (通知)、Channels (頻道) 和 Create channel (建立頻道)。

Amazon Simple Notification Service (Amazon SNS) 是支援的通知頻道類型。為了驗證使用者，您需要為使用者提供對 Amazon SNS 的完整存取權限，或讓他們擔任具有存取 Amazon SNS 許可的 IAM 角色。如需相關指示，請參閱《Amazon SNS 作為頻道類型》。

差異

Amazon OpenSearch Service 中的提醒與 OpenSearch 的開放原始碼版本相比，有一些顯著差異。

提醒設定

使用 OpenSearch Service 可修改下列提醒設定：

- `plugins.scheduled_jobs.enabled`
- `plugins.alerting.alert_history_enabled`
- `plugins.alerting.alert_history_max_age`

- plugins.alerting.alert_history_max_docs
- plugins.alerting.alert_history_retention_period
- plugins.alerting.alert_history_rollover_period
- plugins.alerting.filter_by_backend_roles

所有其他設定都會使用您無法變更的預設值。

若要停用提醒，請傳送下列請求：

```
PUT _cluster/settings
{
  "persistent" : {
    "plugins.scheduled_jobs.enabled" : false
  }
}
```

下列請求會將提醒設定為在七天後自動刪除歷史記錄索引，而不是預設的 30 天：

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.alerting.alert_history_retention_period": "7d"
  }
}
```

如果您先前已建立監視器，並想要停止建立每日提醒索引，請刪除所有提醒歷史記錄索引：

```
DELETE .plugins-alerting-alert-history-*
```

若要減少歷史記錄索引的碎片計數，請建立索引範本。下列請求將提醒的歷史記錄索引設定為一個碎片和一個複本：

```
PUT _index_template/template-name
{
  "index_patterns": [".opendistro-alerting-alert-history-*"],
  "template": {
    "settings": {
      "number_of_shards": 1,
```

```
      "number_of_replicas": 1
    }
  }
}
```

根據您對資料丟失的容忍度，您甚至可以考慮使用零複本。如需建立和管理索引範本的詳細資訊，請參閱 OpenSearch 文件中的索引範本。

Amazon OpenSearch Service 中的異常偵測

Amazon OpenSearch Service 中的異常偵測功能會使用 Random Cut Forest (隨機分割森林，簡稱 RCF) 演算法，以近乎即時的速度自動偵測 OpenSearch 資料中的異常。RCF 是一種非監督式的機器學習演算法，它會為傳入資料串流的草圖建立模型。該演算法會為每個傳入資料點計算 `anomaly grade` 和 `confidence score` 值。異常偵測使用這些值來區分資料中的異常與正常變化。

您可以將異常偵測外掛程式與警示外掛程式配對，以便在偵測到異常後立即通知您。

在執行任何 OpenSearch 版本或 Elasticsearch 7.4 或更新版本的網域上均可使用異常偵測。除了 `t2.micro` 和 `t2.small` 外，所有執行個體類型都支援異常偵測。

> **Note**
>
> 本文件提供 Amazon OpenSearch Service 內容中異常偵測的簡短概觀。如需完整的文件，包括詳細步驟、API 參考、所有可用設定的參考，以及建立視覺化和儀表板的步驟，請參閱開放原始碼 OpenSearch 文件中的異常偵測。

先決條件

異常偵測具有以下先決條件：

- 異常偵測需要 OpenSearch 或 Elasticsearch 7.4 或更新版本。
- 異常偵測在 Elasticsearch 版本 7.9 和更新版本以及所有版本的 OpenSearch 上僅支援精細存取控制。在 Elasticsearch 7.9 之前，只有管理員使用者可以建立、檢視和管理偵測器。
- 如果您的網域使用精細存取控制，則非系統管理員使用者必須在 OpenSearch Dashboards 中映射到 `anomaly_read_access` 角色來檢視偵測器，或映射到 `anomaly_full_access` 來建立和管理偵測器。

開始使用異常偵測

若要開始使用，請在 OpenSearch Dashboards 中選擇異常偵測。

步驟 1：建立偵測器

偵測器是個別的異常偵測任務。您可以建立多個偵測器，且所有偵測器皆可同時執行，而每個偵測器會分析來自不同來源的資料。

步驟 2：將功能新增至偵測器

功能是您檢查異常之索引中的欄位。偵測器可在一或多個功能中探索異常。您必須針對每個功能選擇以下其中一個彙總：`average()`、`sum()`、`count()`、`min()` 或 `max()`。

> **Note**
>
> `count()` 彙總方法只能在 OpenSearch 和 Elasticsearch 7.7 或更新版本中使用。對於 Elasticsearch 7.4，請使用如下所示的自訂表達式：
>
> ```
> {
>   "aggregation_name": {
>      "value_count": {
>         "field": "field_name"
>      }
>   }
> }
> ```

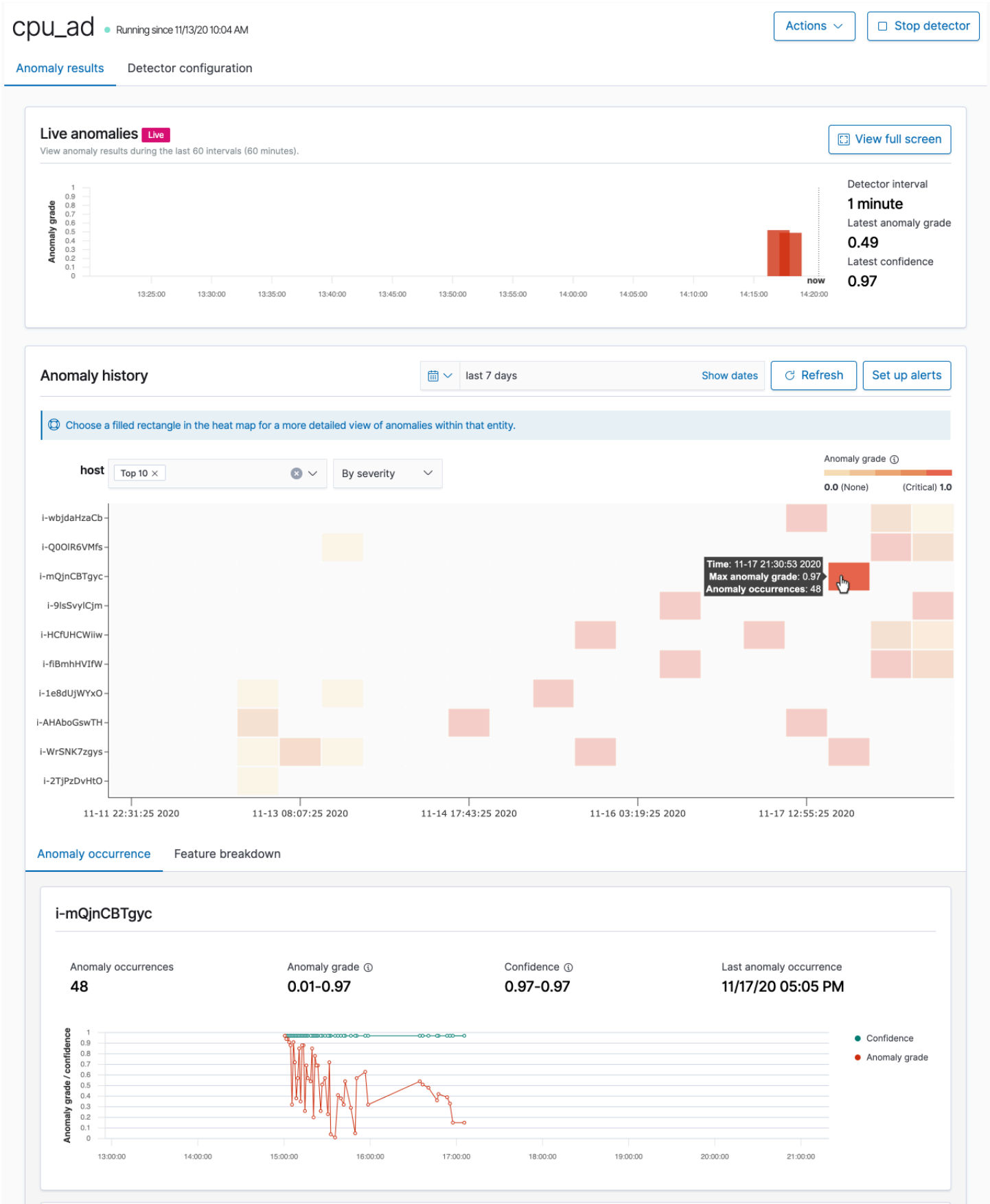
彙總方法會決定構成異常的內容。例如，如果您選擇 `min()`，偵測器會專注於根據功能的最小值來尋找異常。如果您選擇 `average()`，偵測器會根據功能的平均值來尋找異常。每個偵測器最多可以新增五個功能。

您可以配置下列選用設定 (在 Elastisearch 7.7 及更新版本中可用)：

- 類別欄位-使用 IP 地址、產品 ID、國家/地區代碼等維度來分類或分割資料。
- 時段大小-設定要在偵測時段中考量的資料串流彙總間隔數。

設定功能後，請預覽範例異常，並視需要調整功能設定。

步驟 3：觀察結果


cpu_ad
Running since 11/13/20 10:04 AM
Actions
Stop detector
Anomaly results
Detector configuration
Live anomalies
Live
View anomaly results during the last 60 intervals (60 minutes).
View full screen
Anomaly grade
Detector interval
1 minute
Latest anomaly grade
0.49
Latest confidence
0.97
now
Anomaly history
last 7 days
Show dates
Refresh
Set up alerts
Choose a filled rectangle in the heat map for a more detailed view of anomalies within that entity.
host
Top 10
By severity
Anomaly grade
0.0 (None)
(Critical) 1.0
i-wbjdaHzaCb
i-Q0OlR6VMfs
i-mQjnCBTgyc
i-9lsSvylCjm
i-HCfUHCWiiw
i-fiBmhHVIfW
i-1e8dUjWYxO
i-AHAboGswTH
i-WrSNK7zgys
i-2TjPzDvHtO
Time: 11-17 21:30:53 2020
Max anomaly grade: 0.97
Anomaly occurrences: 48
11-11 22:31:25 2020
11-13 08:07:25 2020
11-14 17:43:25 2020
11-16 03:19:25 2020
11-17 12:55:25 2020
Anomaly occurrence
Feature breakdown
i-mQjnCBTgyc
Anomaly occurrences
48
Anomaly grade
0.01-0.97
Confidence
0.97-0.97
Last anomaly occurrence
11/17/20 05:05 PM
Anomaly grade / confidence
Confidence
Anomaly grade
Anomaly occurrences (48)
Start time
End time
Entity
Data confidence
Anomaly grade
11/17/20 5:04 PM
11/17/20 5:05 PM
i-mQjnCBTgyc
0.97
0.15

- Live anomalies (即時異常) - 顯示最後 60 個間隔的即時異常結果。例如，如果間隔設為 10，則會顯示最後 600 分鐘的結果。此圖表每隔 30 秒重新整理一次。
- Anomaly history (異常歷程記錄) - 會使用對應可信度測量來繪製異常分數。
- Feature breakdown (功能明細) - 根據彙總方法來繪製功能。您可以呈現偵測器的日期時間範圍差異。
- Anomaly occurrence (異常出現次數) - 顯示每個偵測到的異常的 `Start time`、`End time`、`Data confidence` 和 `Anomaly grade`。

  如果您設定類別欄位，您會看到額外的熱度圖圖表，它會關聯異常實體的結果。選擇填滿的矩形以查看異常的更詳細檢視。

步驟 4：設定提醒

若要建立監控以在偵測到任何異常時傳送通知，請選擇 Set up alerts (設定提醒)。外掛程式會將您重新導向至 Add monitor (新增監控) 頁面，您可在此設定提醒。

教學課程：使用異常偵測來偵測高 CPU 使用率

此教學課程展示如何在 Amazon OpenSearch Service 中建立異常偵測器，以偵測高 CPU 使用率。您將使用 OpenSearch Dashboards 來設定偵測器以監控 CPU 使用率，並在 CPU 使用率上升超過指定閾值時產生警示。

> **Note**
>
> 這些步驟適用於最新版本的 OpenSearch，對於過去的版本，則可能會略有不同。

先決條件

- 您必須具有執行 Elasticsearch 7.4 或更新版本或者任何 OpenSearch 版本的 OpenSearch Service 網域。
- 您必須將應用程式日誌檔案擷取到包含 CPU 使用率資料的叢集中。

步驟 1：建立偵測器

首先，建立偵測器來識別 CPU 使用率資料中的異常情況。

1. 在 OpenSearch Dashboards 中開啟左側面板選單，選擇 Anomaly Detection (異常偵測)，然後選擇 Create detector (建立偵測器)。
2. 將偵測器命名為 **high-cpu-usage**。
3. 對於您的資料來源，請選擇包含 CPU 使用率日誌檔案 (您要在其中識別異常情況) 的索引。
4. 從您的資料中選擇 Timestamp field (時間戳記欄位)。您可以選擇性地新增資料篩選條件。此資料篩選條件僅分析資料來源的子集，並減少不相關資料的雜訊。
5. 將 Detector interval (偵測器間隔) 設為 2 分鐘。此間隔定義偵測器收集資料的時間 (按分鐘間隔)。
6. 在 Window delay (時段延遲) 中，新增 1-minute (1 分鐘) 延遲。此延遲會增加額外的處理時間，以確保該時段內的所有資料都存在。
7. 選擇下一步。在異常偵測儀表板的偵測器名稱下，選擇 Configure model (設定模型)。
8. 針對 Feature name (功能名稱)，輸入 **max_cpu_usage**。針對 Feature state (功能狀態)，選取 Enable feature (啟用功能)。
9. 針對 Find anomalies based on (尋找異常的依據)，選擇 Field value (欄位值)。
10. 針對 Aggregation method (彙總方法)，選擇 **max()**。
11. 針對 Field (欄位)，選取資料中的欄位以檢查異常。例如，它可能會稱為 `cpu_usage_percentage`。
12. 將其他所有設定保留為預設值，然後選擇 Next (下一步)。
13. 忽略偵測器任務設定並選擇 Next (下一步)。
14. 在彈出式視窗中，選擇何時啟動偵測器 (自動或手動)，然後選擇 Confirm (確認)。

既然已設定偵測器，在它初始化後，您將能夠在偵測器面板的 Real-time results (即時結果) 區段中看到 CPU 使用率的即時結果。Live anomalies (即時異常) 區段會顯示即時擷取資料時發生的任何異常。

步驟 2：設定警示

既然您已建立偵測器，請建立一個監控器，以在偵測到符合偵測器設定中指定條件的 CPU 使用率時，呼叫警示以便將訊息傳送至 Slack。當來自一個或多個索引的資料符合呼叫警示的條件時，您會收到 Slack 通知。

1. 在 OpenSearch Dashboards 中打開左側面板選單，選擇 Alerting (警示)，然後選擇 Create monitor (建立監控器)。
2. 提供監控器的名稱。
3. 針對 Monitor type (監控器類型)，選擇 Per-query monitor (每個查詢監控器)。每個查詢監控器會執行指定的查詢並定義觸發條件。

4. 針對 Monitor defining method (監控器定義方法)，選擇 Anomaly detector (異常偵測器)，然後從 Detector (偵測器) 下拉式選單中選取您在上一個區段中建立的偵測器。
5. 針對 Schedule (排程)，選擇監控器收集資料的頻率，以及您接收警示的頻率。為了實現本教學課程的目的，請將排程設定為每 7 分鐘執行一次。
6. 在 Triggers (觸發條件) 區段中，選擇 Add trigger (新增觸發條件)。針對 Trigger name (觸發條件名稱)，輸入 **`High CPU usage`**。在本教學課程中，針對 Severity level (嚴重性等級)，選擇 1 (最高嚴重性等級)。
7. 針對 Anomaly grade threshold (異常等級閾值)，選擇 IS ABOVE (超過)。在其下面的選單中，選擇要套用的等級閾值。針對本教學課程，請將 Anomaly grade (異常等級) 設為 0.7。
8. 針對 Anomaly confidence threshold (異常可信度閾值)，選擇 IS ABOVE (超過)。在其下面的選單中，輸入與異常等級相同的數字。針對本教學課程，請將 Anomaly confidence threshold (異常可信度閾值) 設為 0.7。
9. 在 Actions (動作) 區段中，請選擇 Destination (目的地)。在 Name (名稱) 欄位中，請選擇目的地名稱。在 Type (類型) 選單中，請選擇 Slack。在 Webhook URL 欄位中，請輸入要接收警示的 Webhook URL。如需詳細資訊，請參閱 Sending messages using incoming webhooks (使用傳入的 Webhook 傳送訊息)。

10選擇建立。

相關資源

- the section called “提醒”

- the section called “異常偵測”
- 異常偵測 API

Amazon OpenSearch Service 的 AI 助理

AI Assistant 是生成式 AI 輔助，可協助開發人員和 IT 專業人員在整個軟體開發生命週期中執行各種任務。AI Assistant 可 AWS 協助客戶編碼、測試、部署、疑難排解和最佳化在 中執行的應用程式 AWS。

AI Assistant 與 Amazon OpenSearch Service 整合提供下列生成功能：

- the section called “使用自然語言產生視覺化效果”
- the section called “檢視提醒摘要和洞見”
- the section called “在探索頁面上檢視 AI 助理產生的查詢結果摘要”
- the section called “檢視建議的異常偵測器”
- the section called “存取適用於 OpenSearch Service 問題的 AI 助理聊天”

若要存取 OpenSearch Service 中與您的任務相關的 AI Assistant 功能，請在提醒、探索和建立視覺化頁面上尋找下列內容內圖示。按一下 圖示以提示 協助，並使用本節所述的功能。

您也可以按一下 OpenSearch Service 主控台右上角的 圖示，直接與 AI Assistant 聊天。AI 助理聊天支援 OpenSearch Service 功能相關問題的解答。

支援的 AWS 區域

以下提供 OpenSearch Service 的 AI 助理支援 AWS 區域：

- 美國東部 (維吉尼亞北部)
- 美國西部 (奧勒岡)
- 亞太地區 (孟買)
- 亞太地區 (雪梨)
- 亞太區域 (東京)
- 加拿大 (中部)
- 歐洲 (法蘭克福)
- 歐洲 (倫敦)
- Europe (Paris)
- 南美洲 (聖保羅)

設定適用於 OpenSearch Service 的 AI 助理

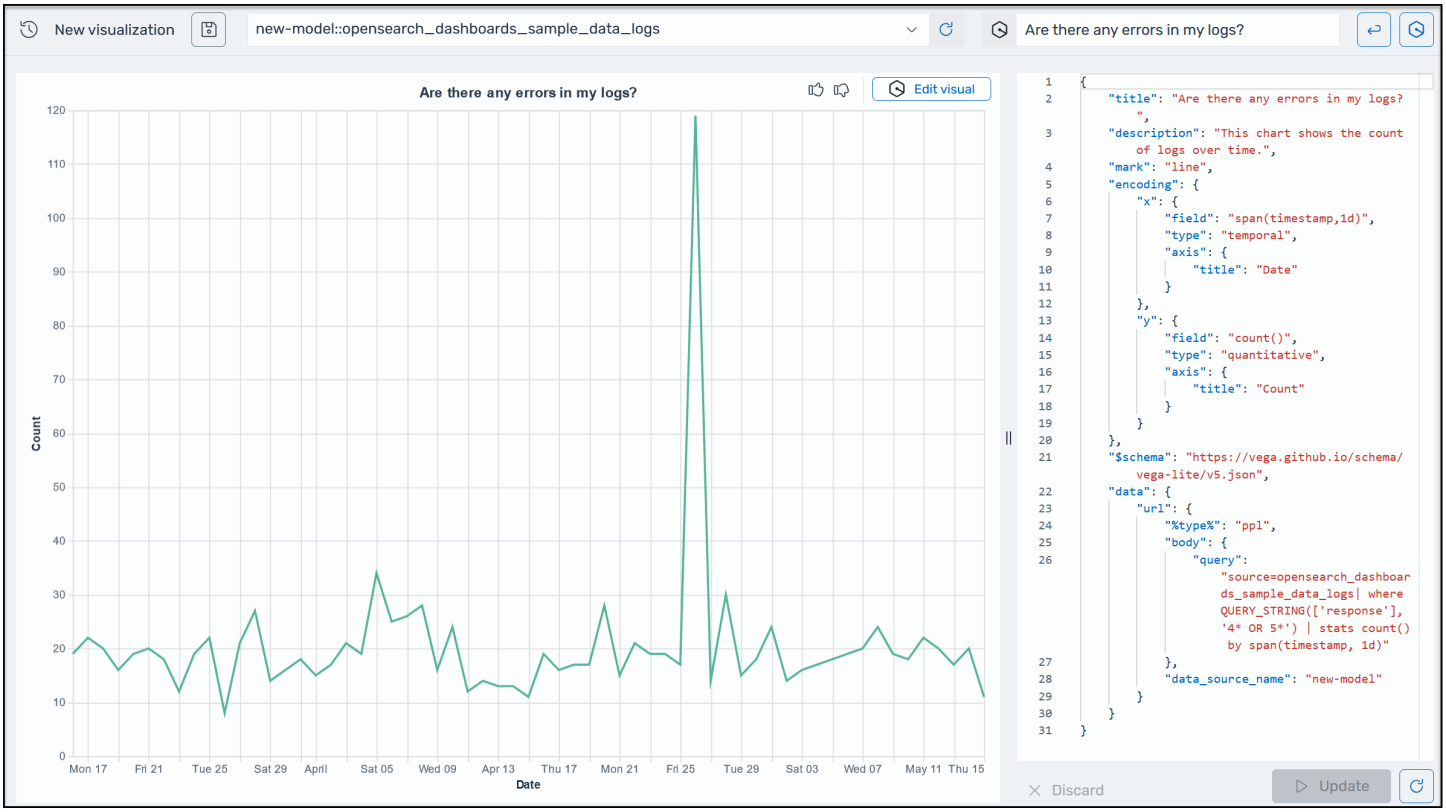
完成下列步驟，以設定 OpenSearch Service 的 AI 助理。

1. 確認您已設定 OpenSearch Service 精細存取控制。如需詳細資訊，請參閱Amazon OpenSearch Service 中的精細存取控制。
2. 確認您的資料來源位於 OpenSearch 2.17 版或更新版本。
3. 在建立網域或編輯叢集組態期間，請確認您已在人工智慧 (AI) 和Machine Learning (ML) 區段中選取啟用自然語言查詢產生核取方塊。

> **Note**
>
> Amazon Q for Developer OpenSearch 功能適用於 Q 免費方案。如需詳細資訊，請參閱《Amazon Q 使用者指南》中的了解服務層。

使用自然語言產生視覺化效果

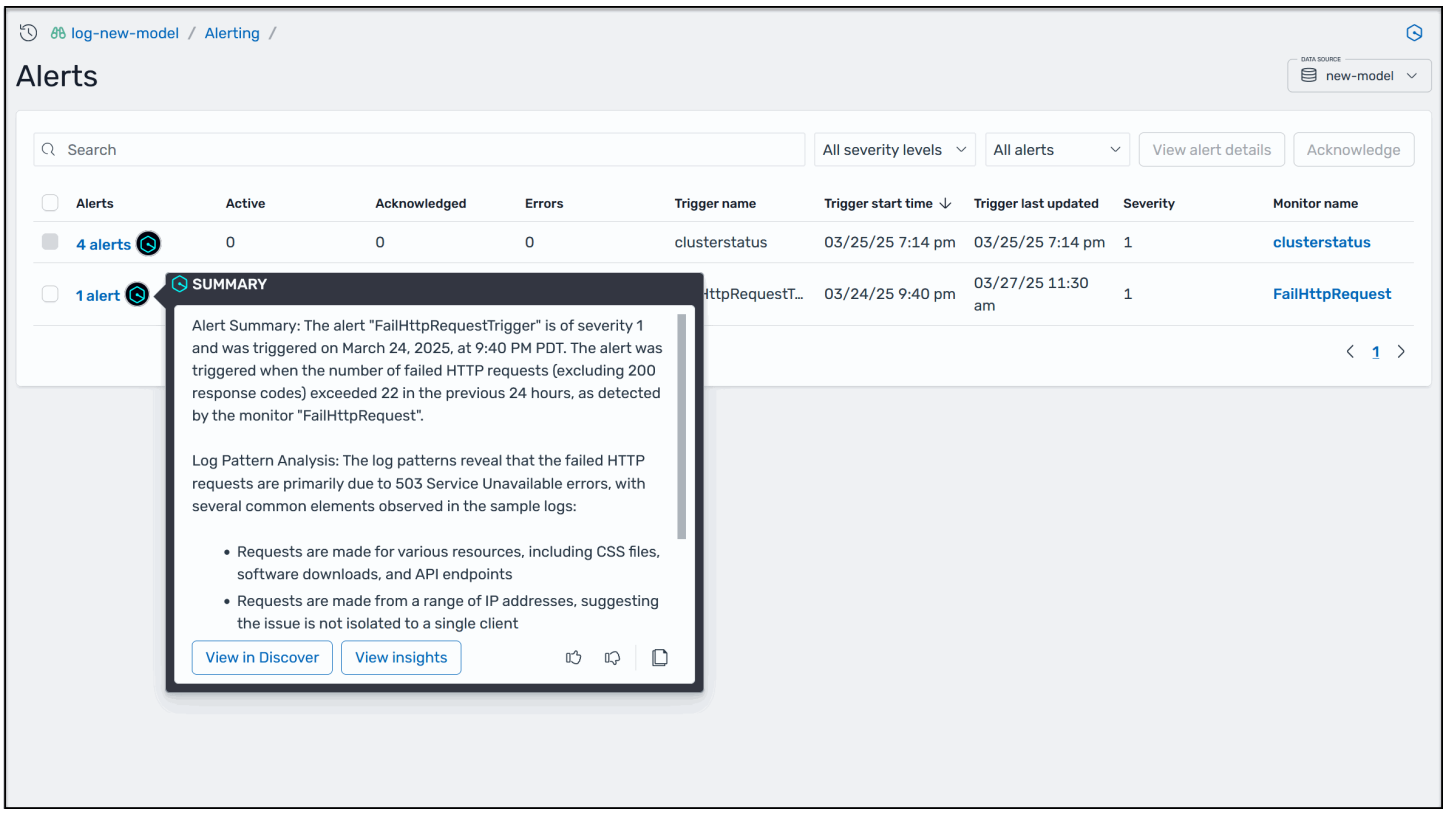
為了協助您深入了解營運資料，適用於 OpenSearch Service 的 AI Assistant 支援使用自然語言提示來建置視覺化效果。開始之前，請確認您已為 OpenSearch Service 設定 AI 助理。確認後，啟動 OpenSearch Service UI 應用程式，導覽至視覺化頁面，選取 + 建立視覺化按鈕，然後選擇自然語言。您現在可以指定資料來源，並以自然語言輸入產生視覺效果的需求。請見此處範例。



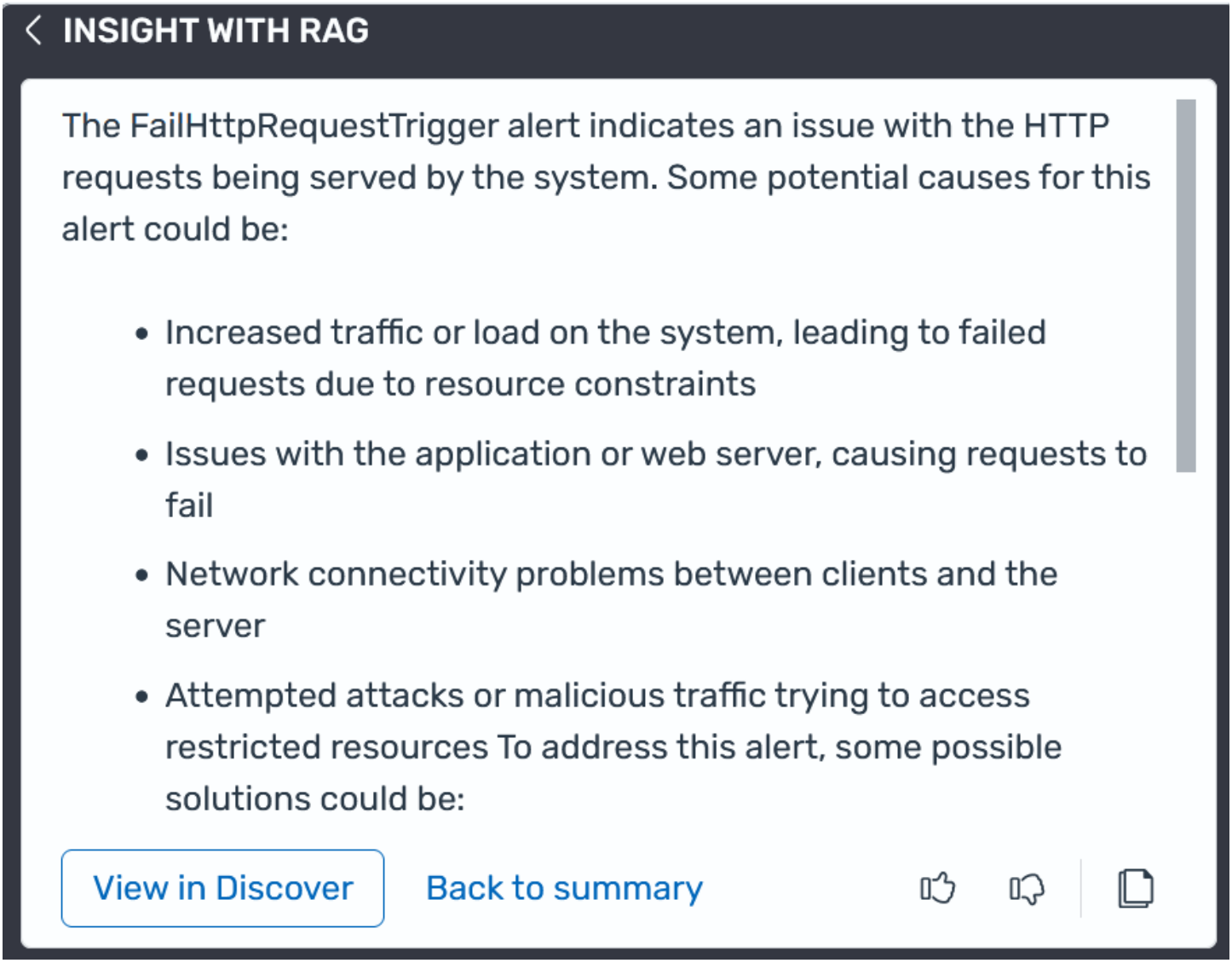
視覺化可以根據不同的維度細分錯誤分析，以加速故障診斷。視覺化也可協助您識別模式和趨勢、加速決策、揭露關係，以及簡化複雜的資料。或者，您也可以在探索頁面中建立視覺化，做為查詢結果的一部分。

檢視提醒摘要和洞見

您可以設定 OpenSearch Service 在一或多個索引的資料符合特定條件時建立警示監控。為了協助您快速了解警示並進行疑難排解，您可以按一下警示旁的 AI 助理圖示來檢視警示摘要。摘要提供有關觸發警示的基礎問題的詳細資訊，並在可用時提供額外的分析，以協助您找出問題的根本原因。下列螢幕擷取畫面顯示 AI Assistant 建立的提醒摘要範例。



如果您連接知識庫以提供環境的其他內容，如本主題稍後所述，AI Assistant 會建立提醒的洞察。Insights 提供詳細資訊和疑難排解選項，協助您修正警示的根本原因。針對先前顯示的提醒，AI Assistant 也會產生下列洞見。



> **Note**
>
> 根據提醒的性質和可用資訊，AI Assistant 可讓您選擇在 OpenSearch Dashboards 的探索頁面上檢視提醒資料。如果您在 AI 助理提醒摘要底部看到在探索中檢視按鈕，請按一下按鈕，在探索中開啟對應的資料集，其中包含提醒資料的作用中篩選條件。

主題

- 開始之前
- 檢視提醒摘要和洞見

開始之前

請完成下列步驟來設定 Amazon Bedrock 知識庫，以便 AI Assistant 可以為 OpenSearch Service 警示建立洞見。

步驟 1：建立 LambdaInvokeOpenSearchMLCommonsRole IAM 角色

在 AWS Identity and Access Management (IAM) `LambdaInvokeOpenSearchMLCommonsRole` 中建立新的名為 的角色。OpenSearch Service 使用此角色在 OpenSearch 中建立 AI 連接器，協助根據設定的知識庫文章產生洞見。您必須將此角色映射至 OpenSearch Service `ml_full_access`角色，如步驟 2 所述。

當您建立新的角色時，對於信任的實體類型，請選擇AWS 帳戶。您不需要指定許可政策。在 Add permissions (新增許可) 頁面上，選擇 Next (下一步)。如需建立新角色的詳細資訊，請參閱為 AWS 服務建立角色（主控台）。

步驟 2：將 LambdaInvokeOpenSearchMLCommonsRole 角色映射至 OpenSearch Service ml_full_access 角色

使用下列程序將`LambdaInvokeOpenSearchMLCommonsRole`角色映射至 OpenSearch Service `ml_full_access`角色。此映射也有助於 OpenSearch Service 建立 AI 連接器。

將所需的 IAM 角色映射至 OpenSearch Service ml_full_access 角色

1. 開啟 OpenSearch Service Dashboard 資料管理頁面。
2. 在資料存取和使用者下，選擇角色。
3. 使用搜尋方塊來尋找`ml_full_access`角色。
4. 在 ml_full_access 頁面上，選擇映射的使用者索引標籤。
5. 選擇映射使用者。
6. 在後端角色欄位中，貼上`LambdaInvokeOpenSearchMLCommonsRole`角色的 Amazon Resource Name (ARN)，然後選擇映射。

步驟 3：使用 設定 OpenSearch Service 知識庫 CloudFormation

請使用下列程序，使用 設定 OpenSearch Service 知識庫， AWS CloudFormation 以便 AI Assistant 產生洞見。

設定知識庫以取得洞見

1. 在支援的 中登入 Amazon OpenSearch Service 主控台 https://console.aws.amazon.com/aos/home AWS 區域。如需詳細資訊，請參閱支援的 AWS 區域。
2. 在導覽窗格中選擇整合。
3. 在整合範本區段中，選擇透過 Amazon Bedrock 範本與知識庫整合。如果您沒有看到此範本，請確認您位於支援的區域。
4. 在透過 Amazon Bedrock 圖磚與知識庫整合中，選擇設定網域，然後選擇其中一個可用的選項。OpenSearch Service 會開啟 CloudFormation 堆疊範本，並預先填入必要欄位。CloudFormation 堆疊支援公有和 VPC 網域的整合。
5. 選擇建立堆疊。 CloudFormation 建立 資源後，服務會顯示 Amazon Bedrock 代理程式 AgentId、ConnectorId 和 ModelId。

適用時，AI Assistant 現在會建立 OpenSearch Service 提醒的洞見。

檢視提醒摘要和洞見

使用下列程序在 OpenSearch Service 中檢視提醒摘要和洞見。

檢視提醒摘要和洞見

1. 確認您已為 OpenSearch Service 設定 AI 助理。
2. 確認您已設定 OpenSearch Service 的提醒。
3. 在 OpenSearch Dashboards 主功能表中，選擇提醒，然後選擇提醒。
4. 選擇提醒旁的 AI 助理圖示。AI Assistant 最多可能需要 10 秒才能產生摘要。
5. 如果警示摘要中存在，請選擇檢視洞見，根據您設定的知識庫檢視警示的詳細資訊。
6. 如果警示摘要中存在，請選擇在探索中檢視，以在 OpenSearch Dashboards 的探索頁面上檢視警示資料。

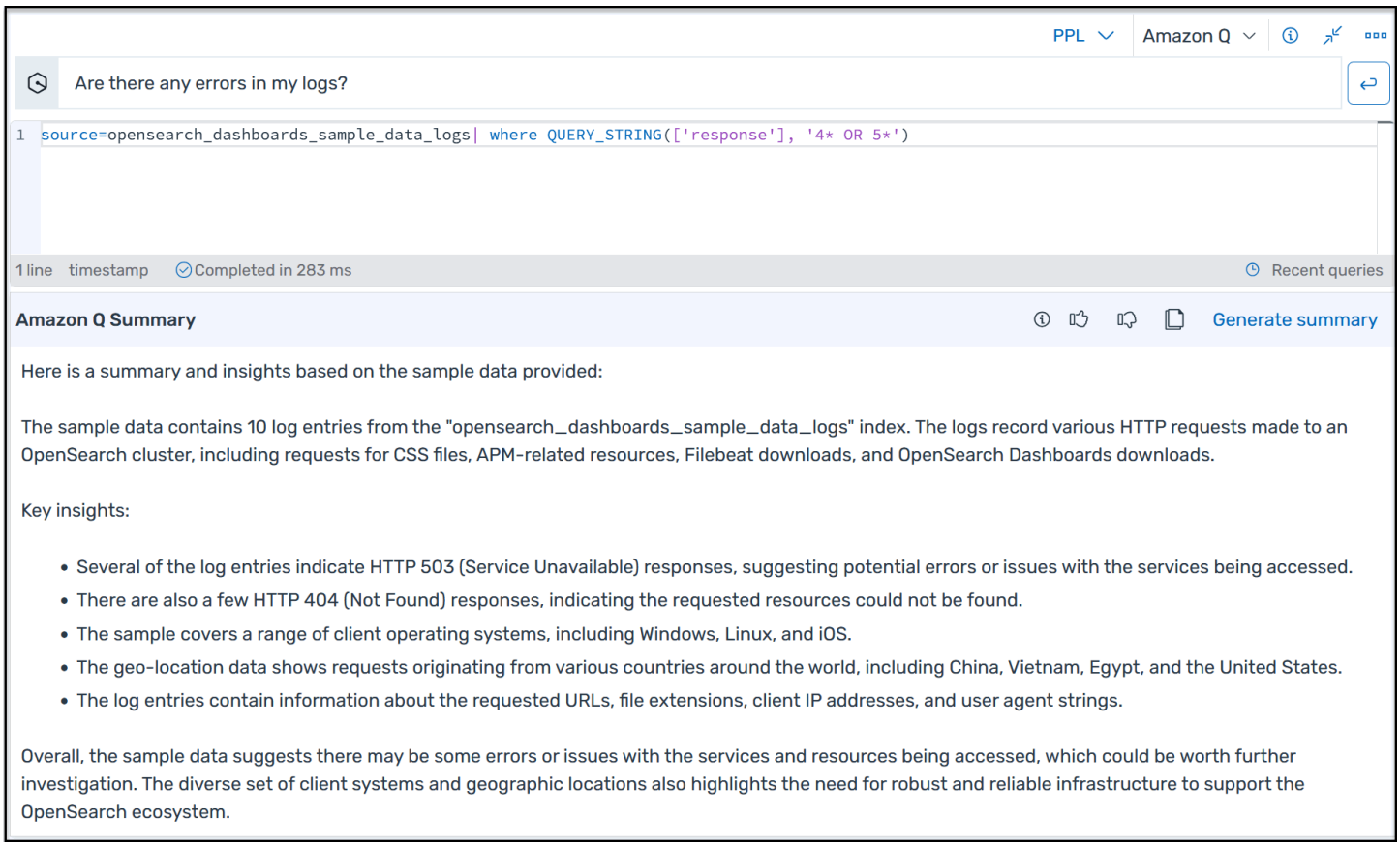
在探索頁面上檢視 AI 助理產生的查詢結果摘要

OpenSearch Service 可讓您使用探索頁面上的 Piped Processing Language (PPL) 查詢語言，使用自然語言提示來查詢資料。例如，您可以撰寫如下的查詢：

- 我的錯誤日誌中是否有任何錯誤？

- 每個星期的平均請求大小是多少？
- 上週有多少請求依回應碼分組？

為了回應，AI Assistant 會根據前十筆記錄產生查詢結果的自然語言摘要，如下所示：



對警示進行故障診斷時，結合自然語言查詢產生和查詢摘要可以新增額外層級的查詢，或使用易於理解資料的方式，而無需撰寫複雜的查詢。

在探索頁面上檢視 AI 助理產生的查詢結果摘要

1. 確認您已為 OpenSearch Service 設定 AI 助理。
2. 在 OpenSearch Dashboards 主功能表中，選擇探索。
3. 在查詢語言下拉式清單中，選擇 PPL。
4. 在 AI 助理文字方塊中，輸入提示，然後按一下文字方塊旁的按鈕來執行查詢。AI Assistant 最多可能需要 10 秒才能傳回摘要。初始查詢之後，您必須為後續摘要選擇產生摘要。

> **Note**
>
> 您可以從探索中的 AI 助理下拉式清單關閉摘要產生。

檢視建議的異常偵測器

Amazon OpenSearch Service 中的異常偵測功能會使用 Random Cut Forest (隨機分割森林，簡稱 RCF) 演算法，以近乎即時的速度自動偵測 OpenSearch 資料中的異常。RCF 是一種非監督式的機器學習演算法，它會為傳入資料串流的草圖建立模型。該演算法會為每個傳入資料點計算 `anomaly grade` 和 `confidence score` 值。異常偵測使用這些值來區分資料中的異常與正常變化。

為了簡化建立異常偵測器的程序，AI Assistant 可以根據您在探索頁面上選取的資料來源產生建議的偵測器。AI Assistant 支援任何語言的建議異常偵測器。

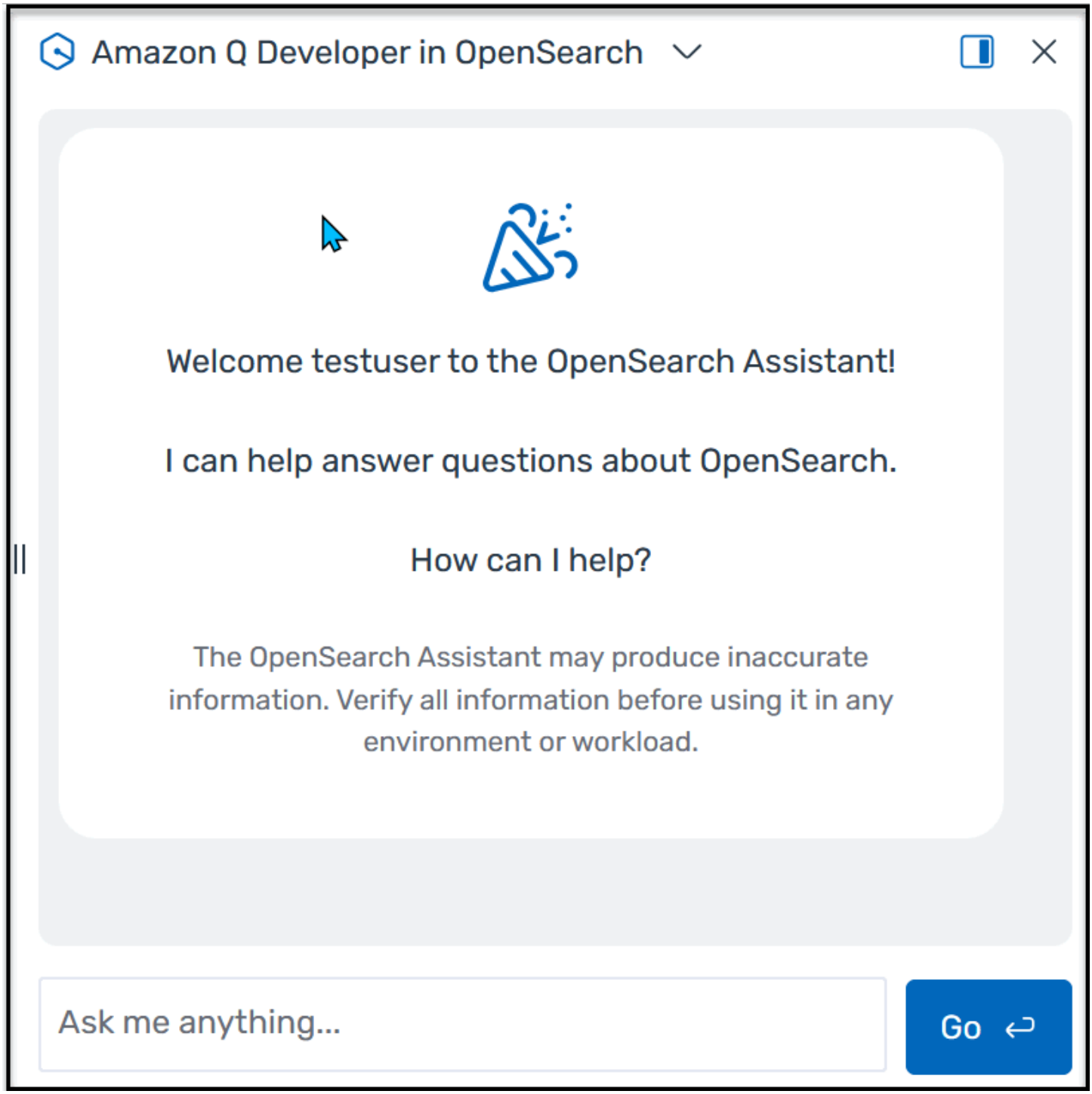
檢視 AI Assistant 建議的異常偵測器

1. 確認您已為 OpenSearch Service 設定 AI 助理。
2. 在 OpenSearch UI 主功能表中，選擇探索頁面，然後選擇資料來源。
3. 從 AI 助理選單中，選擇建議異常偵測器，如下列螢幕擷取畫面所示。

   AI Assistant 可能需要幾秒鐘的時間來產生偵測器的功能。
4. 選擇建立偵測器。

針對 OpenSearch 相關問題存取 AI 助理聊天

如果您對 Amazon OpenSearch Service 有任何疑問，包括與 OpenSearch Service 功能相關的概念性或程序性問題，您可以向 AI Assistant 詢問相關資訊。聊天機器人支援對話式聊天，保留您討論的內容。它也會儲存對話的歷史記錄，以供日後參考。若要存取聊天機器人，請按一下任何 OpenSearch Service 頁面右上角的 AI 助理圖示。在文字方塊中輸入問題，然後按一下 Go。AI Assistant 最多可能需要 30 秒才能產生回應。



以下是 AI Assistant 的範例回應。

Can you suggest some node management best practices?

Based on your context of being in the OpenSearch Dashboard in the us-west-2 region, here are some node management best practices for Amazon OpenSearch Service: [1]

1. Implement Dedicated Manager Nodes:
   - Use dedicated manager nodes to improve cluster stability. [2]
   - Configure three dedicated manager nodes for optimal domain stability across three Availability Zones.
   - This offloads cluster management tasks from data nodes, enhancing overall performance.
2. Multi-AZ Deployment:
   - Deploy your domain across multiple Availability Zones for increased availability and resilience.
   - Consider using the Multi-AZ with Standby option for 99.99% availability and consistent performance.
3. Proper Sizing:

適用於 Amazon OpenSearch Service 的機器學習

ML Commons 是一種 OpenSearch 外掛程式，可透過傳輸和 REST API 呼叫提供一組常見的機器學習 (ML) 演算法。這些呼叫會為每個 ML 請求選擇適當的節點和資源，並監控 ML 任務以確保正常執行時間。這可讓您利用現有的開放原始碼 ML 演算法，並減少開發新的 ML 功能所需的工作量。如需外掛程式的詳細資訊，請參閱 OpenSearch 文件中的機器學習。本章說明如何搭配 Amazon OpenSearch Service 使用 外掛程式。

主題

- 適用於 的 Amazon OpenSearch Service ML 連接器 AWS 服務
- 適用於第三方平台的 Amazon OpenSearch Service ML 連接器
- 使用 CloudFormation 設定語意搜尋的遠端推論
- 不支援的 ML Commons 設定
- OpenSearch Service 流程架構範本

適用於 的 Amazon OpenSearch Service ML 連接器 AWS 服務

當您將 Amazon OpenSearch Service Machine Learning (ML) 連接器與另一個連接器搭配使用時 AWS 服務，您需要設定 IAM 角色，以安全地將 OpenSearch Service 連線至該服務。 AWS 服務 您可以設定連接器來包含 Amazon SageMaker AI 和 Amazon Bedrock。在本教學課程中，我們將介紹如何建立從 OpenSearch Service 到 SageMaker 執行期的連接器。如需連接器的詳細資訊，請參閱支援的連接器。

主題

- 先決條件
- 建立 OpenSearch Service 連接器

先決條件

若要建立連接器，您必須擁有 Amazon SageMaker AI 網域端點和授予 OpenSearch Service 存取權的 IAM 角色。

設定 Amazon SageMaker AI 網域

請參閱《Amazon SageMaker AI 開發人員指南》中的在 Amazon SageMaker AI 中部署模型，以部署您的機器學習模型。請注意模型的端點 URL，您需要此 URL 才能建立 AI 連接器。

建立 IAM 角色

設定 IAM 角色將 SageMaker 執行期許可委派給 OpenSearch Service。若要建立新的角色，請參閱《IAM 使用者指南》中的建立 IAM 角色（主控台）。 或者，只要現有角色具有相同的權限集，您就可以使用該角色。如果您確實建立新的角色，而不是使用 AWS 受管角色，請在本教學`opensearch-sagemaker-role`課程中將 取代為您自己的角色名稱。

1. 將下列受管 IAM 政策連接至您的新角色，以允許 OpenSearch Service 存取您的 SageMaker AI 端點。若要將政策連接至角色，請參閱新增 IAM 身分許可。

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Action": [
                   "sagemaker:InvokeEndpointAsync",
                   "sagemaker:InvokeEndpoint"
               ],
               "Effect": "Allow",
               "Resource": "*"
           }
       ]
   }
   ```

2. 遵循修改角色信任政策中的指示來編輯角色的信任關係。在下列政策中，將 *`service-principal`* 取代為下列其中一個適用於 OpenSearch Service 或 OpenSearch Serverless 的服務主體：

   對於 OpenSearch Service

   ```
   opensearchservice.amazonaws.com
   ```

   對於 OpenSearch Serverless

   ```
   ml.opensearchservice.amazonaws.com
   ```

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ml.opensearchservice.amazonaws.com"
                ]
            }
        }
    ]
}
```

建議您使用 aws:SourceAccount和 aws:SourceArn條件金鑰來限制對特定網域的存取。SourceAccount 是屬於網域擁有者的 AWS 帳戶 ID，而 SourceArn是網域的 ARN。例如，您可以將下列條件區塊新增至信任政策：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

設定 許可

若要建立連接器，您需要將 IAM 角色傳遞至 OpenSearch Service 的許可。您也需要存取 es:ESHttpPost 動作。若要授予這兩個許可，請將下列政策連接至其憑證用於簽署請求的 IAM 角色：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/opensearch-sagemaker-
role"
        },
        {
            "Effect": "Allow",
            "Action": "es:ESHttpPost",
            "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
        }
    ]
}
```

如果您的使用者或角色沒有傳遞角色的`iam:PassRole`許可，當您在下一個步驟嘗試註冊儲存庫時，可能會遇到授權錯誤。

在 OpenSearch Dashboards 中映射 ML 角色（如果使用精細存取控制）

精細存取控制會在設定連接器時引入額外的步驟。即使您將 HTTP 基本身分驗證用於所有其他目的，您也需要將 `ml_full_access` 角色映射至擁有 `iam:PassRole` 許可能夠傳遞 `opensearch-sagemaker-role` 的 IAM 角色。

1. 導覽至 OpenSearch Service 網域的 OpenSearch Dashboards 外掛程式。您可以在 OpenSearch Service 主控台的網域儀表板上找到 Dashboards 端點。
2. 從主選單選擇安全性、角色，然後選取 ml_full_access 角色。
3. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
4. 在後端角色下，新增具有傳遞 許可之角色的 ARN`opensearch-sagemaker-role`。

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. 選擇 Map (映射)，並確認使用者或角色顯示在 Mapped users (已映射的使用者) 中。

建立 OpenSearch Service 連接器

若要建立連接器，請將POST請求傳送至 OpenSearch Service 網域端點。您可以使用 curl、範例 Python 用戶端、Postman 或其他方法來傳送已簽署的請求。請注意，您無法在 Kibana 主控台中使用POST請求。請求採用下列格式：

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
         "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
```

如果您的網域位於虛擬私有雲端 (VPC) 中，您的電腦必須連接到 VPC，請求才能成功建立 AI 連接器。存取 VPC 因網路組態而異，但通常涉及連線到 VPN 或公司網路。若要檢查您是否可以連線到 OpenSearch Service 網域，請在 Web 瀏覽器https://*your-vpc-domain*.*region*.es.amazonaws.com中導覽至 ，並確認您收到預設的 JSON 回應。

Python 用戶端範例

Python 用戶端比 HTTP 請求更易於自動化，且具有更佳的可重複使用性。若要使用 Python 用戶端建立 AI 連接器，請將下列範例程式碼儲存至 Python 檔案。用戶端需要 適用於 Python (Boto3) 的 AWS SDK、 requests和 requests-aws4auth套件。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Register repository
path = '_plugins/_ml/connectors/_create'
url = host + path

payload = {
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
```

```
        "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
headers = {"Content-Type": "application/json"}

r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

適用於第三方平台的 Amazon OpenSearch Service ML 連接器

在本教學課程中，我們將介紹如何從 OpenSearch Service 建立連接器到 Cohere。如需連接器的詳細資訊，請參閱支援的連接器。

當您搭配外部遠端模型使用 Amazon OpenSearch Service 機器學習 (ML) 連接器時，您需要將特定授權憑證存放在其中 AWS Secrets Manager。這可以是 API 金鑰或使用者名稱和密碼組合。這表示您也需要建立 IAM 角色，允許 OpenSearch Service 存取從 Secrets Manager 讀取。

主題

- 先決條件
- 建立 OpenSearch Service 連接器

先決條件

若要使用 OpenSearch Service 為 Cohere 或任何外部供應商建立連接器，您必須擁有授予 OpenSearch Service 存取權的 IAM 角色 AWS Secrets Manager，您可以在其中存放登入資料。您也必須將登入資料存放在 Secrets Manager 中。

建立 IAM 角色

設定 IAM 角色將 Secrets Manager 許可委派給 OpenSearch Service。您也可以使用現有的 `SecretManagerReadWrite` 角色。若要建立新角色，請參閱《IAM 使用者指南》中的建立 IAM 角色（主控台）。 如果您確實建立新的角色，而不是使用 AWS 受管角色，請在本教學`opensearch-secretmanager-role`課程中將 取代為您自己的角色名稱。

1. 將下列受管 IAM 政策連接至您的新角色，以允許 OpenSearch Service 存取您的 Secrets Manager 值。若要將政策連接至角色，請參閱新增 IAM Identity Permissions。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

2. 遵循修改角色信任政策中的指示來編輯角色的信任關係。在下列政策中，將 *service-principal* 取代為下列其中一個適用於 OpenSearch Service 或 OpenSearch Serverless 的服務主體：

   對於 OpenSearch Service

   ```
   opensearchservice.amazonaws.com
   ```

   對於 OpenSearch Serverless

   ```
   ml.opensearchservice.amazonaws.com
   ```

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Action": [
                   "sts:AssumeRole"
               ],
               "Effect": "Allow",
               "Principal": {
                   "Service": [
                       "service-principle"
                   ]
               }
           }
       ]
   ```

```
}
```

我們建議您使用 aws:SourceAccount和 aws:SourceArn條件金鑰來限制對特定網域的存取。SourceAccount 是屬於網域擁有者的 AWS 帳戶 ID，而 SourceArn是網域的 ARN。例如，您可以將下列條件區塊新增至信任政策：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

設定 許可

若要建立連接器，您需要將 IAM 角色傳遞至 OpenSearch Service 的許可。您也需要存取 es:ESHttpPost 動作。若要授予這兩個許可，請將下列政策連接至其憑證用於簽署請求的 IAM 角色：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/opensearch-secretmanager-role"
    },
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPost",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
    }
  ]
}
```

如果您的使用者或角色沒有傳遞角色的iam:PassRole許可，當您在下一個步驟嘗試註冊儲存庫時，可能會遇到授權錯誤。

設定 AWS Secrets Manager

若要在 Secrets Manager 中存放授權憑證，請參閱AWS Secrets Manager 《 使用者指南》中的建立 AWS Secrets Manager 秘密。

在 Secrets Manager 接受金鑰/值對做為秘密後，您會收到格式為 的 ARNarn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3。保留此 ARN 的記錄，當您在下一個步驟中建立連接器時使用該 ARN 和金鑰。

在 OpenSearch Dashboards 中映射 ML 角色 （ 如果使用精細存取控制 ）

精細存取控制會在設定連接器時引入額外的步驟。即使您將 HTTP 基本身分驗證用於所有其他目的，您也需要將 ml_full_access 角色映射至擁有 iam:PassRole 許可能夠傳遞 opensearch-sagemaker-role 的 IAM 角色。

1. 導覽至 OpenSearch Service 網域的 OpenSearch Dashboards 外掛程式。您可以在 OpenSearch Service 主控台的網域儀表板上找到 Dashboards 端點。
2. 從主選單選擇安全性、角色，然後選取 ml_full_access 角色。
3. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
4. 在後端角色下，新增具有傳遞 許可之角色的 ARNopensearch-sagemaker-role。

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. 選擇 Map (映射)，並確認使用者或角色顯示在 Mapped users (已映射的使用者) 中。

建立 OpenSearch Service 連接器

若要建立連接器，請將POST請求傳送至 OpenSearch Service 網域端點。您可以使用 curl、範例 Python 用戶端、Postman 或其他方法來傳送已簽署的請求。請注意，您無法在 Kibana 主控台中使用POST請求。請求採用下列格式：

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
    "name": "Cohere Connector: embedding",
    "description": "The connector to cohere embedding model",
    "version": 1,
```

```
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}
```

此請求的請求內文與開放原始碼連接器請求的請求內文有兩種不同。在 `credential`欄位中，您會傳遞 IAM 角色的 ARN，該角色允許 OpenSearch Service 從 Secrets Manager 讀取，以及哪些秘密的 ARN。在 `headers`欄位中，您會使用私密金鑰及其來自 ARN 的事實來參考秘密。

如果您的網域位於虛擬私有雲端 (VPC) 中，您的電腦必須連線至 VPC，才能請求成功建立 AI 執行個體。存取 VPC 因網路組態而異，但通常涉及連線到 VPN 或公司網路。若要檢查您是否可以連線到 OpenSearch Service 網域，請在 Web 瀏覽器`https://`*your-vpc-domain*`.`*region*`.es.amazonaws.com`中導覽至 ，並確認您收到預設的 JSON 回應。

Python 用戶端範例

Python 用戶端比 HTTP 請求更易於自動化，且具有更佳的可重複使用性。若要使用 Python 用戶端建立 AI 連接器，請將下列範例程式碼儲存至 Python 檔案。用戶端需要 適用於 Python (Boto3) 的 AWS SDK、 `requests`和 `requests-aws4auth`套件。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
credentials = boto3.Session().get_credentials()
```

```
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

path = '_plugins/_ml/connectors/_create'
url = host + path

payload = {
    "name": "Cohere Connector: embedding",
    "description": "The connector to cohere embedding model",
    "version": 1,
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}

headers = {"Content-Type": "application/json"}

r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

使用 CloudFormation 設定語意搜尋的遠端推論

從 OpenSearch 2.9 版開始，您可以使用遠端推論搭配語意搜尋來託管您自己的機器學習 (ML) 模型。遠端推論使用 ML Commons 外掛程式。

透過遠端推論，您可以在 Amazon SageMaker AI 和 Amazon Bedrock 等 ML 服務上遠端託管模型推論，然後使用 ML 連接器將其連接到 Amazon OpenSearch Service。

為了簡化遠端推論的設定，Amazon OpenSearch Service 在 主控台中提供 AWS CloudFormation 範本。CloudFormation 是 AWS 服務 ，您可以將基礎設施視為程式碼，藉此進行、佈建和管理 AWS 和第三方資源。

OpenSearch CloudFormation 範本會為您自動化模型佈建程序，讓您可以輕鬆地在 OpenSearch Service 網域中建立模型，然後使用模型 ID 擷取資料並執行神經搜尋查詢。

當您搭配 OpenSearch Service 2.12 版及更新版本使用神經稀疏編碼器時，建議您在本機使用字符器模型，而不是從遠端部署。如需詳細資訊，請參閱 OpenSearch 文件中的稀疏編碼模型。

主題

- 可用的 CloudFormation 範本
- 先決條件
- Amazon Bedrock 範本
- 使用 Bedrock Claude 設定代理程式搜尋
- MCP 伺服器整合範本
- Amazon SageMaker 範本
- 語意反白範本的遠端推論

可用的 CloudFormation 範本

下列 AWS CloudFormation 機器學習 (ML) 範本可供使用：

Amazon Bedrock 範本

Amazon Titan 文字內嵌整合

連接至 Amazon Bedrock 的託管 ML 模型，無需個別模型部署，並使用預定的 Amazon Bedrock 端點。如需詳細資訊，請參閱《Amazon Bedrock 使用者指南》中的 Amazon Titan 文字內嵌。

Cohere 內嵌整合

提供 Cohere Embed 模型的存取權，並針對特定文字處理工作流程進行最佳化。如需詳細資訊，請參閱 Cohere 文件網站上的內嵌。

Amazon Titan 多模態內嵌

同時支援文字和影像內嵌，並啟用多模式搜尋功能。如需詳細資訊，請參閱《Amazon Bedrock 使用者指南》中的 Amazon Titan 多模態內嵌。

MCP 伺服器整合範本

MCP 伺服器整合

部署 Amazon Bedrock AgentCore 執行期、提供代理程式端點、處理傳入和傳出身分驗證，並支援企業身分驗證的 OAuth。

Amazon SageMaker 範本

透過 Amazon SageMaker 與文字內嵌模型整合

在 Amazon SageMaker 執行期中部署文字內嵌模型、為模型成品存取建立 IAM 角色，以及為語意搜尋建立 ML 連接器。

透過 SageMaker 與稀疏編碼器整合

設定用於神經搜尋的稀疏編碼模型、建立 AWS Lambda 連接器管理的函數，以及傳回模型 IDs以供立即使用。

先決條件

若要搭配 OpenSearch Service 使用 CloudFormation 範本，請完成下列先決條件。

設定 OpenSearch Service 網域

在使用 CloudFormation 範本之前，您必須設定 Amazon OpenSearch Service 網域，並啟用 2.9 版或更新版本和精細存取控制。建立 OpenSearch Service 後端角色，以授予 ML Commons 外掛程式為您建立連接器的許可。

CloudFormation 範本會使用預設名稱 為您建立 Lambda IAM 角色`LambdaInvokeOpenSearchMLCommonsRole`，如果您想要選擇不同的名稱，可以覆寫此角色。範本建立此 IAM 角色後，您需要授予 Lambda 函數呼叫 OpenSearch Service 網域的許可。若要這樣做，請使用下列步驟將名為 的角色映射`ml_full_access`到您的 OpenSearch Service 後端角色：

1. 導覽至 OpenSearch Service 網域的 OpenSearch Dashboards 外掛程式。您可以在 OpenSearch Service 主控台的網域儀表板上找到 Dashboards 端點。
2. 從主選單選擇安全性、角色，然後選取 ml_full_access 角色。
3. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
4. 在後端角色下，新增需要呼叫網域許可的 Lambda 角色 ARN。

```
arn:aws:iam::account-id:role/role-name
```

5. 選擇 Map (映射)，並確認使用者或角色顯示在 Mapped users (已映射的使用者) 中。

映射角色之後，導覽至網域的安全組態，並將 Lambda IAM 角色新增至 OpenSearch Service 存取政策。

在您的 上啟用許可 AWS 帳戶

您的 AWS 帳戶 必須具有存取 CloudFormation 和 Lambda 的許可，以及您為範本 AWS 服務 選擇的任何項目 – SageMaker 執行期或 Amazon Bedrock。

如果您使用的是 Amazon Bedrock，您還必須註冊模型。請參閱《Amazon Bedrock 使用者指南》中的模型存取以註冊您的模型。

如果您使用自己的 Amazon S3 儲存貯體來提供模型成品，則必須將 CloudFormation IAM 角色新增至 S3 存取政策。如需詳細資訊，請參閱《 IAM 使用者指南》中的新增和移除 IAM 身分許可。

Amazon Bedrock 範本

Amazon Bedrock CloudFormation 範本會佈建在 OpenSearch Service 和 Amazon Bedrock 之間建立連接器所需的 AWS 資源。

首先，範本會建立 IAM 角色，允許未來的 Lambda 函數存取您的 OpenSearch Service 網域。範本接著會建立 Lambda 函數，讓網域使用 ML Commons 外掛程式建立連接器。OpenSearch Service 建立連接器後，遠端推論設定即完成，您可以使用 Amazon Bedrock API 操作執行語意搜尋。

> **Note**
>
> 由於 Amazon Bedrock 託管自己的 ML 模型，因此您不需要將模型部署到 SageMaker 執行期。反之，範本會針對 Amazon Bedrock 使用預先決定的端點，並略過端點佈建步驟。

使用 Amazon Bedrock CloudFormation 範本

1. 開啟 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇整合。
3. 在透過 Amazon Bedrock 與 Amazon Titan Text Embeddings 模型整合下，選擇設定網域、設定公有網域。

4. 依照提示設定您的模型。

> **Note**
>
> OpenSearch Service 也提供單獨的範本來設定 Amazon VPC 網域。如果您使用此範本，則需要提供 Lambda 函數的 Amazon VPC ID。

此外，OpenSearch Service 提供下列 Amazon Bedrock 範本，以連線至 Cohere 模型和 Amazon Titan 多模式內嵌模型：

- `Integration with Cohere Embed through Amazon Bedrock`
- `Integrate with Amazon Bedrock Titan Multi-modal`

使用 Bedrock Claude 設定代理程式搜尋

代理程式搜尋透過了解使用者意圖、協調正確的工具、產生最佳化的查詢，以及透過自然語言界面提供其決策的透明摘要，來利用自動代理程式代表您執行複雜的搜尋。這些代理程式採用推理模型，例如 Bedrock Claude。

請依照下列步驟開啟並執行 CloudFormation 範本，自動設定 Bedrock Claude 模型以進行代理程式搜尋，以及如何在 OpenSearch Dashboards 的 AI 搜尋流程外掛程式中設定和建立您的代理程式。

啟用 Bedrock Claude Access

1. 先決條件：如果您的網域使用精細存取控制，請在執行範本之前將 `arn:aws:iam::`*`your-account-id`*`:role/`*`LambdaInvokeOpenSearchMLCommonsRole`*對應為 `ml_full_access`角色的後端角色。如果此 IAM 角色尚未存在，CloudFormation 會自動建立此角色。如需如何設定映射的詳細資訊，請參閱 在 OpenSearch Dashboards 中映射 ML 角色（如果使用精細存取控制）。
2. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
3. 在左側導覽中，選擇整合。
4. 在與代理程式搜尋的 Bedrock Claude 整合下，選擇設定網域。確保您的網域位於 3.3 版或更新版本。

5. 在 CloudFormation 範本中，輸入您的 OpenSearch Service 網域端點，然後選取模型。其餘欄位為選用或預先填入。按一下建立堆疊並等待佈建完成。
6. 從 Amazon OpenSearch Service 主控台，選取網域，然後選取您的網域。按一下 OpenSearch Dashboards URL 以存取 OpenSearch Dashboards。

建置代理程式並執行代理程式搜尋

1. 從 OpenSearch Dashboards 開啟左側的選單。選取 OpenSearch 外掛程式 > AI 搜尋流程以存取外掛程式。
2. 在工作流程頁面上，選取新增工作流程索引標籤，然後在客服人員搜尋卡片下，按一下建立。
3. 為您的搜尋組態提供唯一的名稱，然後按一下建立。
4. 在設定代理程式下，按一下建立新代理程式。選取您新建立的 Bedrock Claude 模型，然後按一下建立代理程式。如果按鈕已停用，請檢查進階設定 > LLM 介面，並確認已選取有效的介面。CloudFormation 的所有模型都會是 Bedrock Claude 模型，因此您可以選取 Bedrock Claude，如果尚未選取，請按一下建立代理程式。
5. 在測試流程下，嘗試執行代理搜尋。提供自然語言搜尋查詢，然後按一下搜尋。

如需 AI 搜尋流程外掛程式的完整文件，請參閱 OpenSearch 文件中的設定客服人員搜尋。

如需客服人員搜尋運作方式的詳細資訊，請參閱 OpenSearch 文件中的客服人員搜尋。

MCP 伺服器整合範本

使用模型內容通訊協定 (MCP) 伺服器範本，您可以在 Amazon Bedrock AgentCore 上部署 OpenSearch 託管 MCP 伺服器，從而降低 AI 代理器和 OpenSearch 工具之間的整合複雜性。如需詳細資訊，請參閱什麼是 Amazon Bedrock AgentCore？。

範本功能

此範本包含下列用於部署和管理 MCP 伺服器的重要功能。

受管 MCP 伺服器部署

使用 Amazon Bedrock AgentCore 執行期部署 opensearch-mcp-server-py，並提供代理程式端點，將請求代理到基礎 MCP 伺服器。如需詳細資訊，請參閱 GitHub 上的 opensearch-mcp-server-py。

身分驗證和安全性

同時處理傳入身分驗證（從使用者到 MCP 伺服器）和傳出身分驗證（從 MCP 伺服器到 OpenSearch），並支援企業身分驗證的 OAuth。

> **Note**
>
> MCP 伺服器範本僅適用於下列 AWS 區域：
>
> - 美國東部 (維吉尼亞北部)
> - 美國西部 (奧勒岡)
> - 歐洲 (法蘭克福)
> - 亞太地區 (悉尼)

使用 MCP 伺服器範本

請依照下列步驟部署 MCP 伺服器範本，並將其連接到您的 OpenSearch 網域。

1. 開啟 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇整合。
3. 找到 MCP 伺服器整合範本。
4. 選擇設定網域。然後，輸入您的 OpenSearch 網域端點。

如果未指定對應的選用參數，範本會建立 AgentCore 執行期和下列元件：

- Amazon ECR 儲存庫
- 做為 OAuth 授權方的 Amazon Cognito 使用者集區
- AgentCore 執行期使用的執行角色

完成此程序後，您應該遵循這些建立後步驟：

1. 對於 Amazon OpenSearch Service：將您的執行角色 ARN 映射至 OpenSearch 後端角色，以控制對網域的存取。

   對於 Amazon OpenSearch Serverless：建立允許執行角色存取集合的資料存取政策。

2. 從您的授權方取得 OAuth 存取權杖。然後，使用此字符在 CloudFormation 堆疊輸出中列出的 URL 存取 MCP 伺服器。

如需詳細資訊，請參閱OpenSearch Serverless 的政策動作。

與 AI 代理器整合

部署之後，您可以將 MCP 伺服器與任何 MCP 相容代理程式整合。如需詳細資訊，請參閱《Amazon Bedrock 開發人員指南》中的叫用部署的 MCP 伺服器。

開發人員整合

您可以將 MCP 伺服器端點新增至您的代理程式組態。您也可以將其與 Amazon Q Developer CLI、自訂代理程式或其他 MCP 相容代理程式搭配使用。

企業部署

中央託管代理程式可以使用 OpenSearch 作為一個元件連接到多個 服務。此代理程式支援 OAuth 和企業身分驗證系統，並擴展以支援多個使用者和使用案例。

使用 Strands Agents 架構的範例

```
import os
import requests
from strands import Agent
from strands.tools.mcp import MCPClient
from mcp.client.streamable_http import streamablehttp_client

def get_bearer_token(discovery_url: str, client_id: str, client_secret: str):
    response = requests.get(discovery_url)
    discovery_data = response.json()
    token_endpoint = discovery_data['token_endpoint']

    data = {
        'grant_type': 'client_credentials',
        'client_id': client_id,
        'client_secret': client_secret
    }
    headers = {
        'Content-Type': 'application/x-www-form-urlencoded'
    }
```

```
    response = requests.post(token_endpoint, data=data, headers=headers)
    token_data = response.json()
    return token_data['access_token']

if __name__ == "__main__":
    discovery_url = os.environ["DISCOVERY_URL"]
    client_id = os.environ["CLIENT_ID"]
    client_secret = os.environ["CLIENT_SECRET"]
    mcp_url = os.environ["MCP_URL"]

    bearer_token = get_bearer_token(discovery_url, client_id, client_secret)

    opensearch_mcp_client = MCPClient(lambda: streamablehttp_client(mcp_url, {
        "authorization": f"Bearer {bearer_token}",
        "Content-Type": "application/json"
    }))

    with opensearch_mcp_client:
        tools = opensearch_mcp_client.list_tools_sync()
        agent = Agent(tools=tools)
        agent("list indices")
```

如需詳細資訊，請參閱 OpenSearch 網站上的使用 Amazon Bedrock AgentCore 託管 OpenSearch MCP 伺服器。 OpenSearch

Amazon SageMaker 範本

Amazon SageMaker CloudFormation 範本會定義多個 AWS 資源，以便為您設定神經外掛程式和語意搜尋。

首先透過 Amazon SageMaker 範本使用與文字內嵌模型的整合，在 SageMaker 執行期中部署文字內嵌模型做為伺服器。如果您不提供模型端點，CloudFormation 會建立 IAM 角色，允許 SageMaker 執行期從 Amazon S3 下載模型成品並將其部署到伺服器。如果您提供端點，CloudFormation 會建立 IAM 角色，允許 Lambda 函數存取 OpenSearch Service 網域，或者，如果角色已存在，則更新和重複使用該角色。端點提供遠端模型，用於具有 ML Commons 外掛程式的 ML 連接器。

然後，透過 Amazon SageMaker 範本使用與稀疏編碼器整合來建立 Lambda 函數，讓您的網域設定遠端推論連接器。在 OpenSearch Service 中建立連接器後，遠端推論可以使用 SageMaker 執行期中的遠端模型執行語意搜尋。範本會將網域中的模型 ID 傳回給您，以便開始搜尋。

使用 Amazon SageMaker AI CloudFormation 範本

1. 開啟 Amazon OpenSearch Service 主控台。
2. 在左側導覽窗格中，選擇整合。
3. 在每個 Amazon SageMaker AI 範本下，選擇設定網域、設定公有網域。
4. 遵循 CloudFormation 主控台中的提示來佈建堆疊並設定模型。

> **Note**
>
> OpenSearch Service 也提供單獨的範本來設定 VPC 網域。如果您使用此範本，則需要提供 Lambda 函數的 VPC ID。

語意反白範本的遠端推論

語意反白是一種進階搜尋功能，透過分析查詢的意義和內容來增強結果相關性，而不是僅依賴確切的關鍵字比對。此功能使用機器學習模型來評估搜尋查詢和文件內容之間的語意相似性，識別和強調文件中最相關的句子或段落。與專注於確切詞彙相符的傳統反白方法不同，語意反白利用 AI 模型，使用查詢和周圍文字的內容資訊來評估每個句子，即使確切搜尋詞彙不存在於反白段落中，也能夠呈現相關資訊。此方法對於 AI 驅動的搜尋實作特別有用，其中使用者將語意意義優先於文字字詞比對，讓搜尋管理員能夠提供更智慧且具有上下文意識的搜尋體驗，強調有意義的內容，而不只是關鍵字出現。如需詳細資訊，請參閱使用語意反白。

使用下列程序開啟並執行 CloudFormation 範本，自動設定 Amazon SageMaker 模型以進行語意反白。

使用語意反白 CloudFormation 範本

1. 開啟位於 https：//https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 主控台。
2. 在左側導覽中，選擇整合。
3. 在透過 Amazon SageMaker 整合啟用語意反白下，選擇設定網域、設定公有網域。
4. 依照提示設定您的模型。

> **Note**
>
> OpenSearch Service 也提供單獨的範本來設定 VPC 網域。如果您使用此範本，則需要提供 Lambda 函數的 VPC ID。

不支援的 ML Commons 設定

Amazon OpenSearch Service 不支援使用下列 ML Commons 設定：

- `plugins.ml_commons.allow_registering_model_via_url`
- `plugins.ml_commons.allow_registering_model_via_local_file`

> **Important**
>
> 在生產叢集上，請勿停用叢集設定 `plugins.ml_commons.only_run_on_ml_node`（請勿將其設定為 `false`）。停用此保護的選項是為了促進開發，但生產叢集應使用連接器。如需詳細資訊，請參閱the section called “的連接器 AWS 服務”。

如需 ML Commons 設定的詳細資訊，請參閱 ML Commons 叢集設定。

OpenSearch Service 流程架構範本

Amazon OpenSearch Service 流程架構範本可讓您為常見使用案例提供範本，以自動化複雜的 OpenSearch Service 設定和預先處理任務。例如，您可以使用流程架構範本來自動化機器學習設定任務。Amazon OpenSearch Service 流程架構範本提供 JSON 或 YAML 文件中設定程序的精簡描述。這些範本描述用於對話聊天或查詢產生、AI 連接器、工具、代理程式和其他元件的自動化工作流程組態，這些元件準備 OpenSearch Service 供生成模型後端使用。

您可以自訂 Amazon OpenSearch Service 流程架構範本，以符合您的特定需求。若要查看自訂流程架構範本的範例，請參閱流程架構。如需 OpenSearch Service 提供的範本，請參閱工作流程範本。如需完整文件，包括詳細步驟、API 參考和所有可用設定的參考，請參閱開放原始碼 OpenSearch 文件中的自動化組態。

> **Note**
>
> 流程架構不支援 OpenSearch Service 2.17 的後端角色篩選。

在 OpenSearch Service 中建立 ML 連接器

Amazon OpenSearch Service 流程架構範本可讓您使用 ml-commons 中提供的建立連接器 API 來設定和安裝 ML 連接器。您可以使用 ML 連接器將 OpenSearch Service 連接到 AWS 其他服務或第三方平台。如需詳細資訊，請參閱建立第三方 ML 平台的連接器。Amazon OpenSearch Service 流程架構 API 可讓您自動化 OpenSearch Service 設定和預先處理任務，並可用於建立 ML 連接器。

您必須先執行下列動作，才能在 OpenSearch Service 中建立連接器：

- 建立一個 Amazon SageMaker AI 網域。
- 建立 IAM 角色。
- 設定傳遞角色許可。
- 在 OpenSearch Dashboards 中映射流程架構和 ml-commons 角色。

如需如何設定 AWS 服務的 ML 連接器的詳細資訊，請參閱 AWS 服務的 Amazon OpenSearch Service ML 連接器。若要進一步了解如何搭配第三方平台使用 OpenSearch Service ML 連接器，請參閱適用於第三方平台的 Amazon OpenSearch Service ML 連接器。

透過流程架構服務建立連接器

若要使用連接器建立流程架構範本，您需要將POST請求傳送至 OpenSearch Service 網域端點。您可以使用 cURL、範例 Python 用戶端、Postman 或其他方法來傳送已簽署的請求。POST 請求採用下列格式：

```
POST /_plugins/_flow_framework/workflow
{
  "name": "Deploy Claude Model",
  "description": "Deploy a model using a connector to Claude",
  "use_case": "PROVISION",
  "version": {
    "template": "1.0.0",
    "compatibility": [
      "2.12.0",
      "3.0.0"
```

```
    ]
  },
  "workflows": {
    "provision": {
      "nodes": [
        {
          "id": "create_claude_connector",
          "type": "create_connector",
          "user_inputs": {
            "name": "Claude Instant Runtime Connector",
            "version": "1",
            "protocol": "aws_sigv4",
            "description": "The connector to Bedrock service for Claude model",
            "actions": [
              {
                "headers": {
                  "x-amz-content-sha256": "required",
                  "content-type": "application/json"
                },
                "method": "POST",
                "request_body": "{ \"prompt\":\"${parameters.prompt}\",
 \"max_tokens_to_sample\":${parameters.max_tokens_to_sample},
 \"temperature\":${parameters.temperature},  \"anthropic_version\":
\"${parameters.anthropic_version}\" }",
                "action_type": "predict",
                "url": "https://bedrock-runtime.us-west-2.amazonaws.com/model/
anthropic.claude-instant-v1/invoke"
              }
            ],
            "credential": {
                "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-
role"
             },
            "parameters": {
              "endpoint": "bedrock-runtime.us-west-2.amazonaws.com",
              "content_type": "application/json",
              "auth": "Sig_V4",
              "max_tokens_to_sample": "8000",
              "service_name": "bedrock",
              "temperature": "0.0001",
              "response_filter": "$.completion",
              "region": "us-west-2",
              "anthropic_version": "bedrock-2023-05-31"
            }
```

```
          }
        }
      ]
    }
  }
}
```

如果您的網域位於虛擬私有雲端 (Amazon VPC) 中，您必須連線到 Amazon VPC，請求才能成功建立 AI 連接器。存取 Amazon VPC 因網路組態而異，但通常涉及連線到 VPN 或公司網路。若要檢查您是否可以連線到 OpenSearch Service 網域，請在 Web 瀏覽器`https://`*`your-vpc-domain`*`.`*`region`*`.es.amazonaws.com`中導覽至 ，並確認您收到預設的 JSON 回應。（將*`#####`*取代為您自己的值。

Python 用戶端範例

Python 用戶端比HTTP請求更易於自動化，且具有更佳的可重複使用性。若要使用 Python 用戶端建立 AI 連接器，請將下列範例程式碼儲存至 Python 檔案。用戶端需要AWS 適用於 Python 的 SDK (Boto3)、Requests：HTTP for Humans 和 requests-aws4auth 1.2.3 套件。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

path = '_plugins/_flow_framework/workflow'
url = host + path

payload = {
  "name": "Deploy Claude Model",
  "description": "Deploy a model using a connector to Claude",
  "use_case": "PROVISION",
  "version": {
    "template": "1.0.0",
    "compatibility": [
      "2.12.0",
      "3.0.0"
```

```
    ]
  },
  "workflows": {
    "provision": {
      "nodes": [
        {
          "id": "create_claude_connector",
          "type": "create_connector",
          "user_inputs": {
            "name": "Claude Instant Runtime Connector",
            "version": "1",
            "protocol": "aws_sigv4",
            "description": "The connector to Bedrock service for Claude model",
            "actions": [
              {
                "headers": {
                  "x-amz-content-sha256": "required",
                  "content-type": "application/json"
                },
                "method": "POST",
                "request_body": "{ \"prompt\":\"${parameters.prompt}\",
 \"max_tokens_to_sample\":${parameters.max_tokens_to_sample},
 \"temperature\":${parameters.temperature},  \"anthropic_version\":
\"${parameters.anthropic_version}\" }",
                "action_type": "predict",
                "url": "https://bedrock-runtime.us-west-2.amazonaws.com/model/
anthropic.claude-instant-v1/invoke"
              }
            ],
            "credential": {
                "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-
role"
             },
            "parameters": {
              "endpoint": "bedrock-runtime.us-west-2.amazonaws.com",
              "content_type": "application/json",
              "auth": "Sig_V4",
              "max_tokens_to_sample": "8000",
              "service_name": "bedrock",
              "temperature": "0.0001",
              "response_filter": "$.completion",
              "region": "us-west-2",
              "anthropic_version": "bedrock-2023-05-31"
            }
```

```
          }
        }
      ]
    }
  }
}

headers = {"Content-Type": "application/json"}

r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

預先定義的工作流程範本

Amazon OpenSearch Service 為一些常見的機器學習 (ML) 使用案例提供多個工作流程範本。使用範本可簡化複雜的設定，並為語意或對話式搜尋等使用案例提供許多預設值。您可以在呼叫建立工作流程 API 時指定工作流程範本。

- 若要使用 OpenSearch Service 提供的工作流程範本，請將範本使用案例指定為`use_case`查詢參數。
- 若要使用自訂工作流程範本，請在請求內文中提供完整範本。如需自訂範本的範例，請參閱範例 JSON 範本或範例 YAML 範本。

範本使用案例

此資料表提供不同可用範本的概觀、範本的描述，以及必要參數。

範本使用案例	Description	必要參數
`bedrock_titan_embedding_model_deploy`	建立和部署 Amazon Bedrock 內嵌模型（預設為 `titan-embed-text-v1`	`create_connector.credential.roleArn`
`bedrock_titan_embedding_model_deploy`	建立和部署 Amazon Bedrock 多模態內嵌模型（預設為 `titan-embed-text-v1`	`create_connector.credential.roleArn`

範本使用案例	Description	必要參數
cohere_embedding_model_deploy	建立和部署 Cohere 內嵌模型（預設為 embed-english-v3.0)。	create_connector.credential.roleArn , create_connector.credential.secretArn
cohere_chat_model_deploy	建立和部署 Cohere 聊天模型（預設為 Cohere Command)。	create_connector.credential.roleArn , create_connector.credential.secretArn
open_ai_embedding_model_deploy	建立和部署 OpenAI 內嵌模型（預設為 text-embedding-ada-002)。	create_connector.credential.roleArn , create_connector.credential.secretArn
openai_chat_model_deploy	建立和部署 OpenAI 聊天模型（預設為 gpt-3.5-turbo)。	create_connector.credential.roleArn , create_connector.credential.secretArn
semantic_search_with_cohere_embedding	設定語意搜尋並部署 Cohere 內嵌模型。您必須提供 Cohere 模型的 API 金鑰。	create_connector.credential.roleArn , create_connector.credential.secretArn
semantic_search_with_cohere_embedding_query_enricher	設定語意搜尋並部署 Cohere 內嵌模型。新增 query_enricher 搜尋處理器，可設定神經查詢的預設模型 ID。您必須提供 Cohere 模型的 API 金鑰。	create_connector.credential.roleArn , create_connector.credential.secretArn

範本使用案例	Description	必要參數
multimodal_search_with_bedrock_titan	部署 Amazon Bedrock 多模態模型，並使用 text_image_embedding 處理器和用於多模態搜尋的 k-NN 索引來設定擷取管道。您必須提供您的 AWS 登入資料。	create_connector.credential.roleArn

> **Note**
>
> 對於需要秘密 ARN 的所有範本，預設值是將金鑰名稱為 "key" 的秘密存放在 AWS Secrets Manager 中。

具有預先訓練模型的預設範本

Amazon OpenSearch Service 提供開放原始碼 OpenSearch Service 中未提供的兩個額外預設工作流程範本。

範本使用案例	Description
semantic_search_with_local_model	設定語意搜尋並部署預先訓練的模型 (msmarco-distilbert-base-tas-b)。新增neural_query_enricher 搜尋處理器，設定神經查詢的預設模型 ID，並建立稱為「my-nlp-index」的連結 k-NN 索引。
hybrid_search_with_local_model	設定混合式搜尋並部署預先訓練的模型 (msmarco-distilbert-base-tas-b)。新增neural_query_enricher 搜尋處理器，設定神經查詢的預設模型 ID，並建立稱為「my-nlp-index」的連結 k-NN 索引。

設定 許可

如果您使用 2.13 版或更新版本建立新的網域，則已具備 許可。如果您在預先存在的 OpenSearch Service 網域上啟用流程架構，且其版本為 2.11 或更早版本，然後升級至 2.13 或更新版本，則必須定義`flow_framework_manager`角色。非系統管理員使用者必須映射至此角色，以便在使用精細存取控制的網域上管理暖索引。若要手動建立 `flow_framework_manager` 角色，請執行以下步驟：

1. 在 OpenSearch Dashboards 中，移至 Security (安全性)，然後選擇 Permissions (許可)。
2. 選擇 Create action group (建立動作群組) 並設定下列群組：

Group name (群組名稱)	許可
`flow_framework_full_access`	• `cluster:admin/opensearch/flow_framework/*` • `cluster_monitor`
`flow_framework_read_accesss`	• `cluster:admin/opensearch/flow_framework/workflow/get` • `cluster:admin/opensearch/flow_framework/workflow/search` • `cluster:admin/opensearch/flow_framework/workflow_state/get` • `cluster:admin/opensearch/flow_framework/workflow_state/search`

3. 選擇 Roles (角色)，然後選擇 Create role (建立角色)。
4. 命名角色 flow_framework_manager。
5. 對於 Cluster permissions (叢集許可)，選取 `flow_framework_full_access` 和 `flow_framework_read_access`。
6. 對於 Index (索引)，輸入 *。
7. 對於 Index permissions (索引許可)，選取 `indices:admin/aliases/get`、`indices:admin/mappings/get` 以及 `indices_monitor`。
8. 選擇建立。
9. 建立角色之後，請將其映射至將管理流程架構索引的任何使用者或後端角色。

Amazon OpenSearch Service 的安全分析

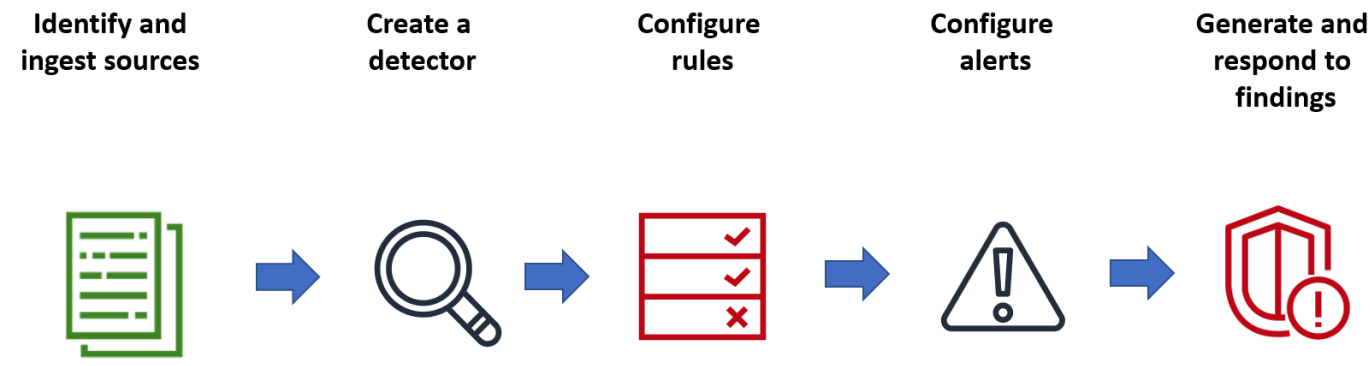
Security Analytics 是一種 OpenSearch 解決方案，可讓您了解組織的基礎設施、監控異常活動、即時偵測潛在的安全威脅，以及觸發對預先設定目的地的提醒。您可以持續評估安全規則並檢閱自動產生的安全調查結果，以監控安全事件日誌中的惡意活動。此外，Security Analytics 可以產生自動提醒，並將其傳送到指定的通知管道，例如 Slack 或電子郵件。

您可以使用 Security Analytics out-of-the-box 外掛程式，立即偵測常見威脅，並從現有的安全事件日誌產生重要的安全洞見，例如防火牆日誌、Windows 日誌和身分驗證稽核日誌。若要使用 Security Analytics，您的網域必須執行 OpenSearch 2.5 版或更新版本。

> **Note**
>
> 本文件提供 Security Analytics for Amazon OpenSearch Service 的簡短概觀。它定義了關鍵概念，並提供設定許可的步驟。如需完整的文件，包括設定指南、API 參考和所有可用設定的參考，請參閱 OpenSearch 文件中的 Security Analytics。

安全分析元件和概念

許多工具和功能為 Security Analytics 的操作提供了基礎。構成外掛程式的主要元件包括偵測器、日誌類型、規則、問題清單和提醒。



日誌類型

OpenSearch 支援多種類型的日誌，並為每個類型提供out-of-the-box映射。您可以在建立偵測器時指定日誌類型並設定時間間隔，然後 Security Analytics 會自動啟用在該間隔執行的相關規則集。

偵測器

偵測器會識別資料索引中日誌類型的各種網路安全威脅。您可以將偵測器設定為使用自訂規則和預先封裝的 Sigma 規則，以評估系統中發生的事件。然後，偵測器會從這些事件產生安全調查結果。如需偵測器的詳細資訊，請參閱 OpenSearch 文件中的建立偵測器。

Rules

威脅偵測規則定義偵測器套用到擷取日誌資料以識別安全事件的條件。Security Analytics 支援匯入、建立和自訂規則以符合您的需求，並提供預先封裝的開放原始碼 Sigma 規則，以偵測日誌中的常見威脅。Security Analytics 將許多規則映射到由 MITRE ATT&CK 組織維護的對手策略和技術不斷增長的知識庫。您可以使用 OpenSearch Dashboards 或 APIs來建立和使用規則。如需規則的詳細資訊，請參閱 OpenSearch 文件中的使用規則。

調查結果

當偵測器比對規則與日誌事件時，會產生問題清單。每個調查結果都包含選取規則、日誌類型和規則嚴重性的唯一組合。問題清單不一定指向系統內的緊迫威脅，但一律會隔離感興趣的事件。如需問題清單的詳細資訊，請參閱 OpenSearch 文件中的使用問題清單。

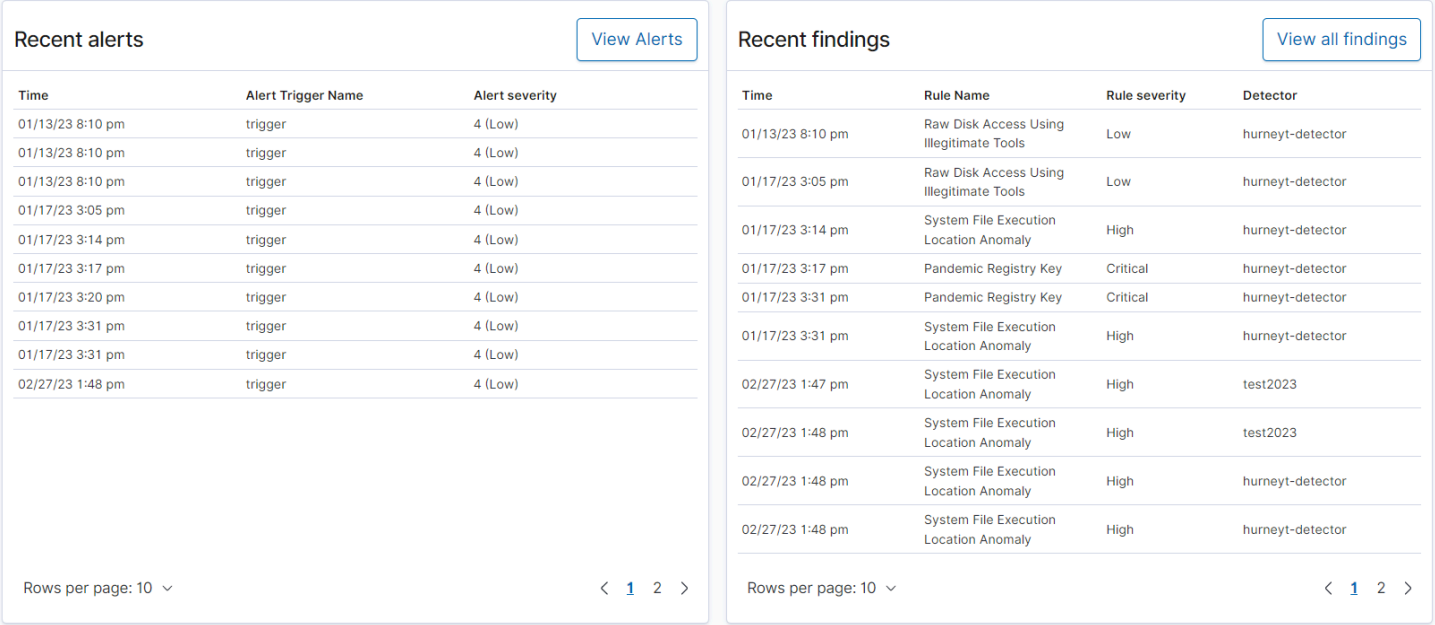
Alerts (提醒)

建立偵測器時，您可以指定一或多個觸發警示的條件。提醒是傳送到偏好管道的通知，例如 Slack 或電子郵件。您可以將警示設定為在偵測器符合一或多個規則時觸發，並且可以自訂通知訊息。如需警示的詳細資訊，請參閱 OpenSearch 文件中的使用警示。

探索安全分析

您可以使用 OpenSearch Dashboards 視覺化並深入了解 Security Analytics 外掛程式。概觀檢視提供的資訊包括問題清單和提醒計數、最近的問題清單和提醒、頻繁的偵測規則，以及偵測器清單。您可以看到包含多個視覺化效果的摘要檢視。例如，以下圖表顯示特定期間內各種日誌類型的調查結果和提醒趨勢。

接下來，您可以檢閱最新的問題清單和提醒。



Recent alerts

View Alerts

Time	Alert Trigger Name	Alert severity
01/13/23 8:10 pm	trigger	4 (Low)
01/13/23 8:10 pm	trigger	4 (Low)
01/13/23 8:10 pm	trigger	4 (Low)
01/17/23 3:05 pm	trigger	4 (Low)
01/17/23 3:14 pm	trigger	4 (Low)
01/17/23 3:17 pm	trigger	4 (Low)
01/17/23 3:20 pm	trigger	4 (Low)
01/17/23 3:31 pm	trigger	4 (Low)
01/17/23 3:31 pm	trigger	4 (Low)
02/27/23 1:48 pm	trigger	4 (Low)

Rows per page: 10 1 2

Recent findings

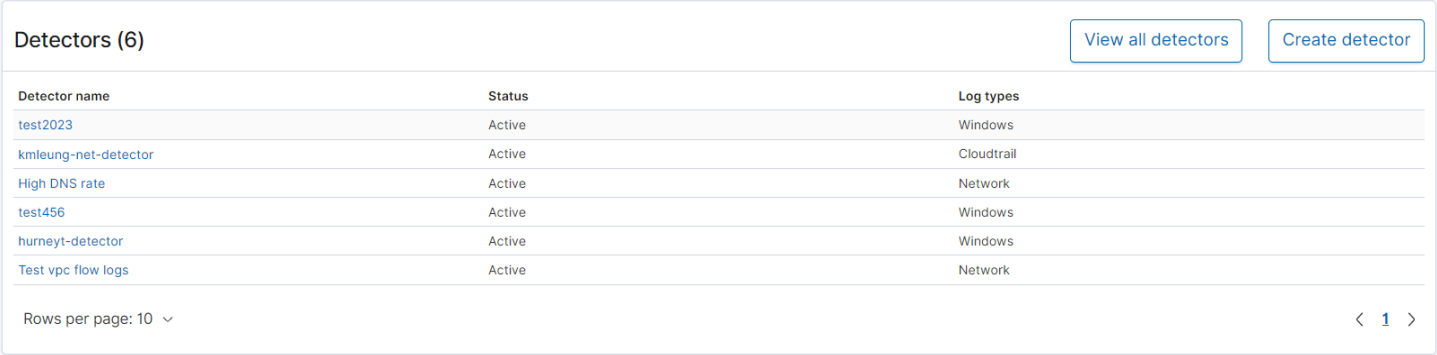
View all findings

Time	Rule Name	Rule severity	Detector
01/13/23 8:10 pm	Raw Disk Access Using Illegitimate Tools	Low	hurneyt-detector
01/17/23 3:05 pm	Raw Disk Access Using Illegitimate Tools	Low	hurneyt-detector
01/17/23 3:14 pm	System File Execution Location Anomaly	High	hurneyt-detector
01/17/23 3:17 pm	Pandemic Registry Key	Critical	hurneyt-detector
01/17/23 3:31 pm	Pandemic Registry Key	Critical	hurneyt-detector
01/17/23 3:31 pm	System File Execution Location Anomaly	High	hurneyt-detector
02/27/23 1:47 pm	System File Execution Location Anomaly	High	test2023
02/27/23 1:48 pm	System File Execution Location Anomaly	High	test2023
02/27/23 1:48 pm	System File Execution Location Anomaly	High	hurneyt-detector
02/27/23 1:48 pm	System File Execution Location Anomaly	High	hurneyt-detector

Rows per page: 10 1 2

此外，您可以看到所有作用中偵測器中最常觸發規則的分佈。這可協助您偵測和調查各種日誌類型的不同類型的惡意活動。

最後，您可以檢視已設定偵測器的狀態。在此面板中，您也可以導覽至建立偵測器工作流程。



若要設定 Security Analytics 設定，請使用規則頁面建立規則，並使用這些規則在偵測器頁面中寫入偵測器。如需更集中檢視 Security Analytics 結果，您可以使用問題清單和提醒頁面。

設定 許可

如果您在預先存在的 OpenSearch Service 網域上啟用 Security Analytics，則可能不會在網域上定義`security_analytics_manager`角色。非系統管理員使用者必須映射至此角色，以便在使用精細存取控制的網域上管理暖索引。若要手動建立 `security_analytics_manager` 角色，請執行以下步驟：

1. 在 OpenSearch Dashboards 中，移至 Security (安全性)，然後選擇 Permissions (許可)。
2. 選擇 Create action group (建立動作群組) 並設定下列群組：

Group name (群組名稱)	許可
security_analytics_full_access	• cluster:admin/opensearch/securityanalytics/alerts/* • cluster:admin/opensearch/securityanalytics/detector/* • cluster:admin/opensearch/securityanalytics/findings/* • cluster:admin/opensearch/securityanalytics/mapping/* • cluster:admin/opensearch/securityanalytics/rule/*
security_analytics_read_access	• cluster:admin/opensearch/securityanalytics/alerts/get • cluster:admin/opensearch/securityanalytics/detector/get • cluster:admin/opensearch/securityanalytics/detector/search • cluster:admin/opensearch/securityanalytics/findings/get • cluster:admin/opensearch/securityanalytics/mapping/get • cluster:admin/opensearch/securityanalytics/mapping/view/get • cluster:admin/opensearch/securityanalytics/rule/get • cluster:admin/opensearch/securityanalytics/rule/search

3. 選擇 Roles (角色)，然後選擇 Create role (建立角色)。
4. 將角色命名為 security_analytics_manager。

5. 對於 Cluster permissions (叢集許可)，選取 `security_analytics_full_access` 和 `security_analytics_read_access`。
6. 對於 Index (索引)，輸入 `*`。
7. 針對索引許可，選取 `indices:admin/mapping/put`和 `indices:admin/mappings/get`。
8. 選擇建立。
9. 建立角色之後，請將其對應至將管理 Security Analytics 索引的任何使用者或後端角色。

疑難排解

沒有此類索引錯誤

如果您沒有偵測器且開啟 Security Analytics 儀表板，您可能會在右下角看到通知，指出 `[index_not_found_exception] no such index [.opensearch-sap-detectors-config]`。您可以忽略此通知，這會在幾秒鐘內消失，而且一旦建立偵測器就不會再次出現。

Amazon OpenSearch Service 中的可觀測性

可觀測性是藉由檢查輸出，深入了解複雜系統內部狀態和效能的做法。傳統監控可以告訴您系統已停機，可觀測性可協助您了解為何停機，方法是允許您詢問有關資料的新問題。

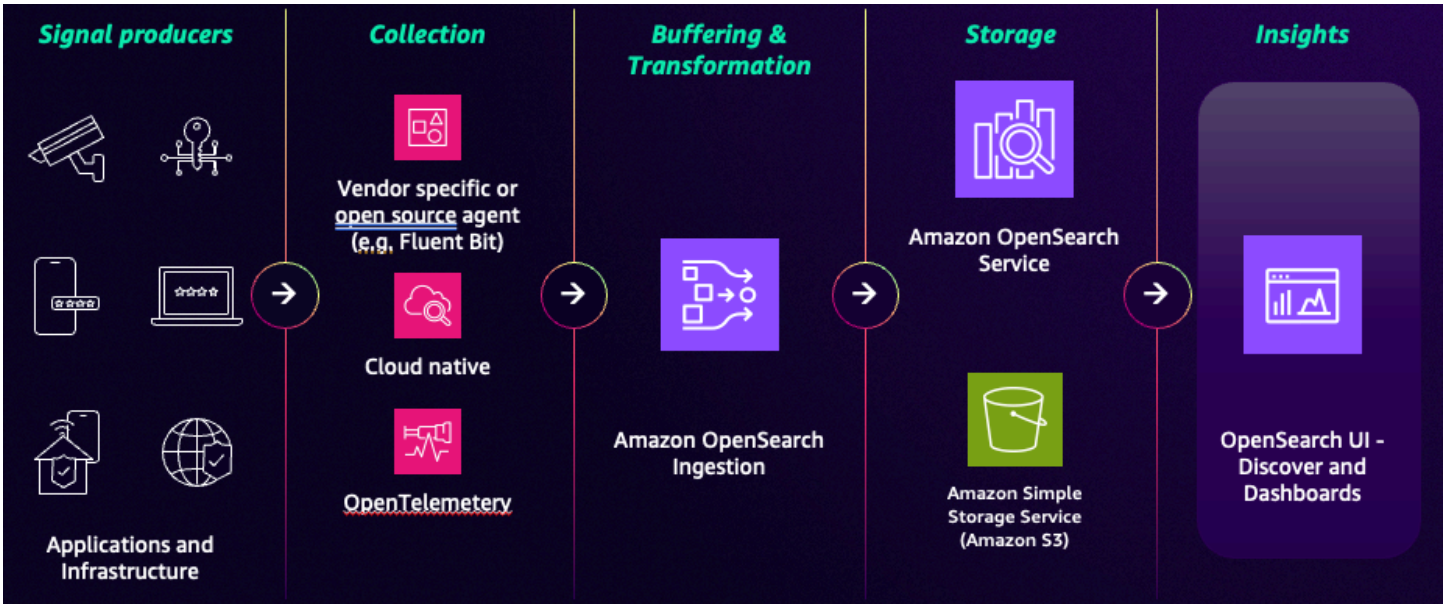
Amazon OpenSearch Service 透過收集和關聯遙測資料的金鑰類型，提供統一的解決方案。

- 日誌提供事件的時間戳記記錄，例如應用程式錯誤、使用者請求或系統狀態訊息。
- 追蹤代表請求在分散式系統中通過不同服務時的end-to-end旅程。

透過將這些資料類型結合在一起，Amazon OpenSearch Service 可協助營運團隊、站點可靠性工程師和開發人員更快地偵測、診斷和解決營運問題。

OpenSearch Service 中的可觀測性工作流程

從應用程式取得資料到 OpenSearch Service 進行分析會使用多階段管道：



1. 集合

   程序從來源開始，使用輕量型代理程式或收集器，從您的訊號生產者收集遙測資料，例如應用程式和基礎設施。兩個常見的開放原始碼代理程式為：

   - OpenTelemetry (OTel) 收集器 – 收集日誌和追蹤的業界標準和偏好方法。
   - Fluent Bit – 輕量、高效能的日誌處理器和轉送器，支援日誌和追蹤的 OTel 結構描述。

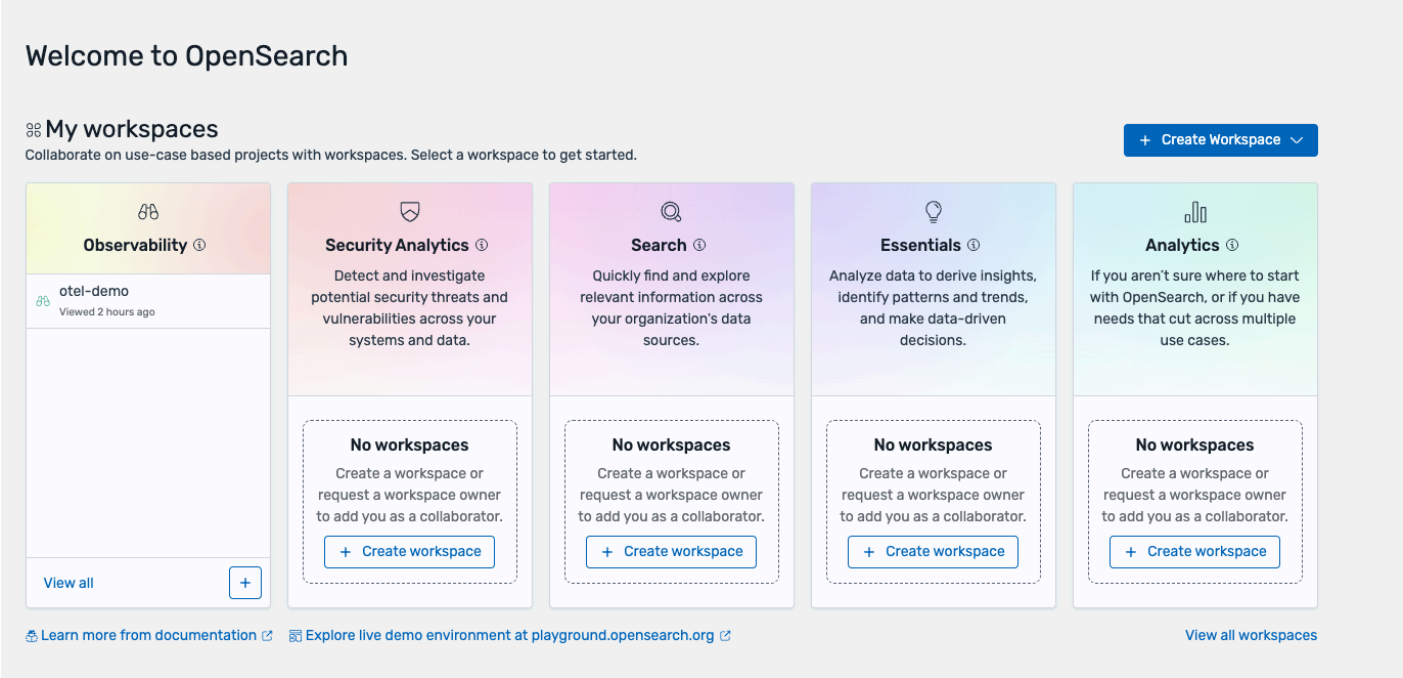
2. 擷取 (Amazon OpenSearch 擷取)

   收集遙測資料後，資料會傳送至 OpenSearch Ingestion，這是全受管、無伺服器的資料管道。您可以建立自訂管道以：

   - 篩選條件 – 移除不必要的資料以降低儲存成本。
   - 豐富 – 新增寶貴的中繼資料，例如來自 IP 地址的地理資訊。
   - 轉換和標準化 – 將非結構化日誌建構為一致的格式。
   - Route – 將不同類型的資料傳送至不同的 OpenSearch Service 索引或 Amazon S3。

3. 分析和視覺化

   處理之後，資料會載入 OpenSearch Service 網域或集合。您可以近乎即時地儲存、編製索引和分析大量資料。您可以透過視覺化界面與此資料互動，例如 OpenSearch UI 的可觀測性工作區，以執行查詢、建置儀表板和設定提醒。



OpenSearch UI 和 OpenSearch Dashboards

OpenSearch Service 為可觀測性提供了兩個不同的使用者介面。我們建議您使用 OpenSearch UI，並為新安裝設定可觀測性工作區，並從現有的 OpenSearch Dashboards 遷移。下表概述 OpenSearch UI v. 傳統 OpenSearch Dashboards 的優點。

功能	OpenSearch UI	OpenSearch Dashboards
資料來源	多來源 – 可以連線至多個 OpenSearch Service 網域、OpenSearch Serverless 集合和其他資料來源。	單一來源 – 與單一 OpenSearch Service 網域共置。
更新	新功能會先送達這裡，因為它未繫結至特定 OpenSearch 版本。	新功能繫結至 OpenSearch 版本。未來可能會棄用更新。
可用性	託管在 中， AWS 雲端 可確保叢集升級期間零停機時間。	可在網域維護和升級期間暫時無法使用
協作	具有在特定工作流程上策劃團隊協作的工作區。	協同合作的基礎是共用單一網域中儲存的物件。

注意：為了方便開始使用，我們已在 Amazon OpenSearch Service 主控台中為日誌建立新的入門工作流程，該工作流程將設定新的 OTel 自訂擷取管道，允許您選取現有的 OpenSearch 叢集，並使用建立的可觀測性工作區建立新的 OpenSearch UI 應用程式。您只需要將 OTel 代理器指向新的擷取端點，您就可以解鎖對 OTel 格式資料的洞察。

由於在日誌和追蹤中擷取和分析的處理方式不同，因此我們建立了要深入探討的個別區段。

日誌

OpenSearch Ingestion 可以在擷取期間將非結構化日誌資料轉換為結構化格式。OpenSearch Ingestion 提供處理器，在編製索引之前標準化和豐富您的資料。實用處理器的範例如下：

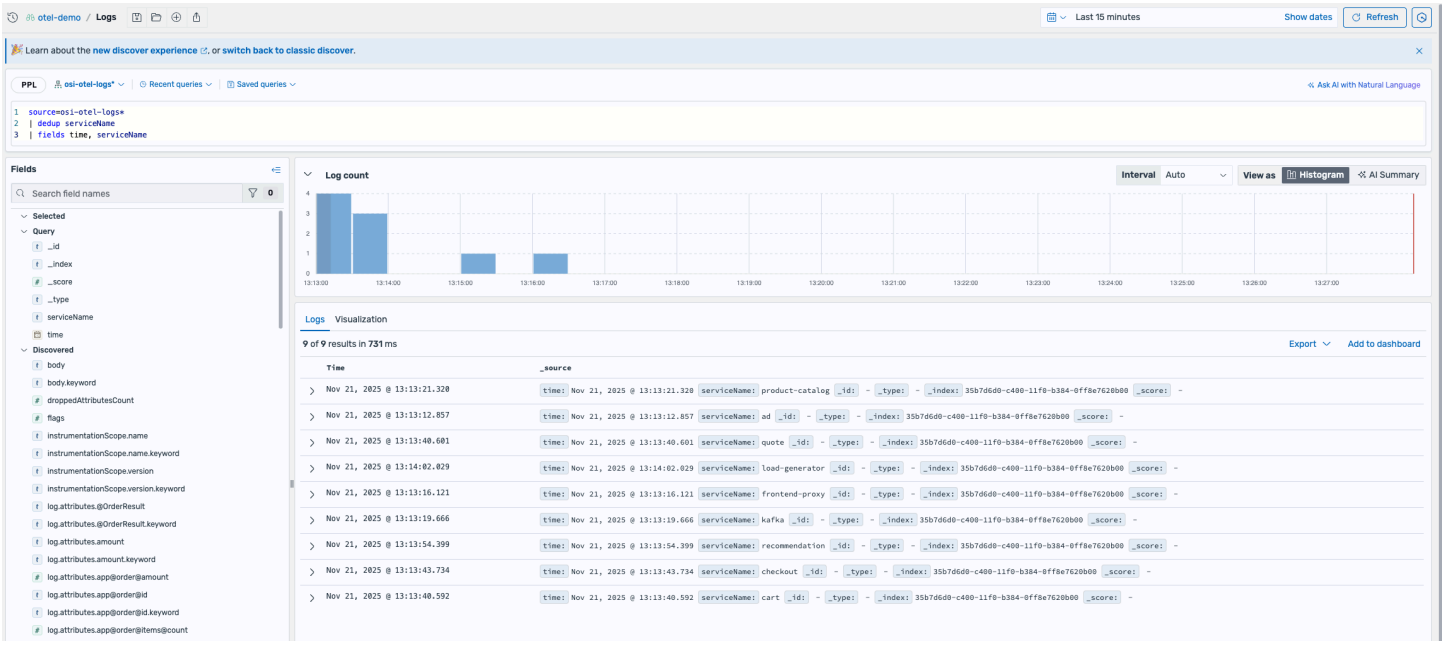
- `grok` – 將 Web 伺服器存取日誌等非結構化文字資料剖析並建構成不同的欄位。
- `date` – 從日誌欄位剖析日期，並將其設定為事件的時間戳記。
- `parse_json` – 剖析包含 JSON 物件的字串欄位。

注意：為了方便開始使用，我們已在 Amazon OpenSearch Service 主控台中為日誌建立新的入門工作流程，該工作流程將設定新的 Otel 自訂擷取管道，將其指向現有的 OpenSearch 叢集，並使用建立的可觀測性工作區建立新的 OpenSearch UI 應用程式。您只需將 Otel 代理器指向新的擷取端點。

OpenSearch UI 和可觀測性工作區

將日誌資料擷取到 Amazon OpenSearch Service 後，您可以使用 OpenSearch UI 中 Amazon OpenSearch Service 可觀測性工作區提供的工具來分析它。 OpenSearch 可觀測性工作區提供專門的工具，旨在擷取 Discover 和 Dashboards 中有意義的洞見。

可觀測性工作區隨附新的探索體驗，使用管道處理語言 (PPL) 搭配 Amazon Q Developer for Business 提供的自然語言助理。語言協助可讓任何人輕鬆開始使用管道語言。精簡查詢後，直接從新的探索建立視覺化和儀表板，而不跳到工具的其他部分。若要使用 DQL 或 SQL 查詢資料，請切換到舊的探索體驗。



使用 PPL 查詢日誌

您有多種選項可查詢日誌，以收集對應用程式或服務操作的洞見。

管道處理語言 (PPL) 是一種查詢語言，具有用於鏈結命令的管道型 (|) 語法。您可以使用它來建置強大的表達式來分析您的日誌。

注意：若要在 OpenSearch 2.19 中解鎖較新的 PPL 命令/函數，您需要使用下列查詢變更 OpenSearch 開發人員工具中的功能旗標 (OpenSearch 3.3 不需要）：

```
PUT /_plugins/_query/settings { "transient" : { "plugins.calcite.enabled" : true } }
```

尋找錯誤最多的主機

此範例會分析您的日誌，以判斷錯誤總數最多的服務主機。

```
source = my-index |
    where level = "ERROR" |
    stats count() as error_count by host |
    sort -error_count |
    head 5
```

計算平均請求時間

此範例會分析您的日誌，以計算日誌中每個狀態碼的平均請求時間。

```
source = my-index |
    stats avg(request_time) by status_code
```

如需 PPL 的詳細資訊，請參閱 https：//opensearch.org 上的 PPL 參考手冊。

使用 AI 查詢您的日誌

此範例會分析您的日誌，以顯示過去 5 分鐘內記錄的錯誤。

```
Show me all of the error logs from the last 5 minutes
```

使用 SQL 查詢您的日誌

SQL 提供熟悉的查詢日誌資料的方式。

此範例會分析您的日誌，以依時間戳記顯示錯誤。

```
SELECT timestamp, severity_text, body, service_name
```

```
FROM opentelemetry_logs
WHERE severity_text = 'ERROR' AND service_name = 'my-service'
ORDER BY timestamp DESC;
```

如需 SQL 的詳細資訊，請參閱 GitHub 上的 SQL 參考手冊。

使用 DQL 查詢您的日誌

DQL 適用於快速搜尋和篩選。

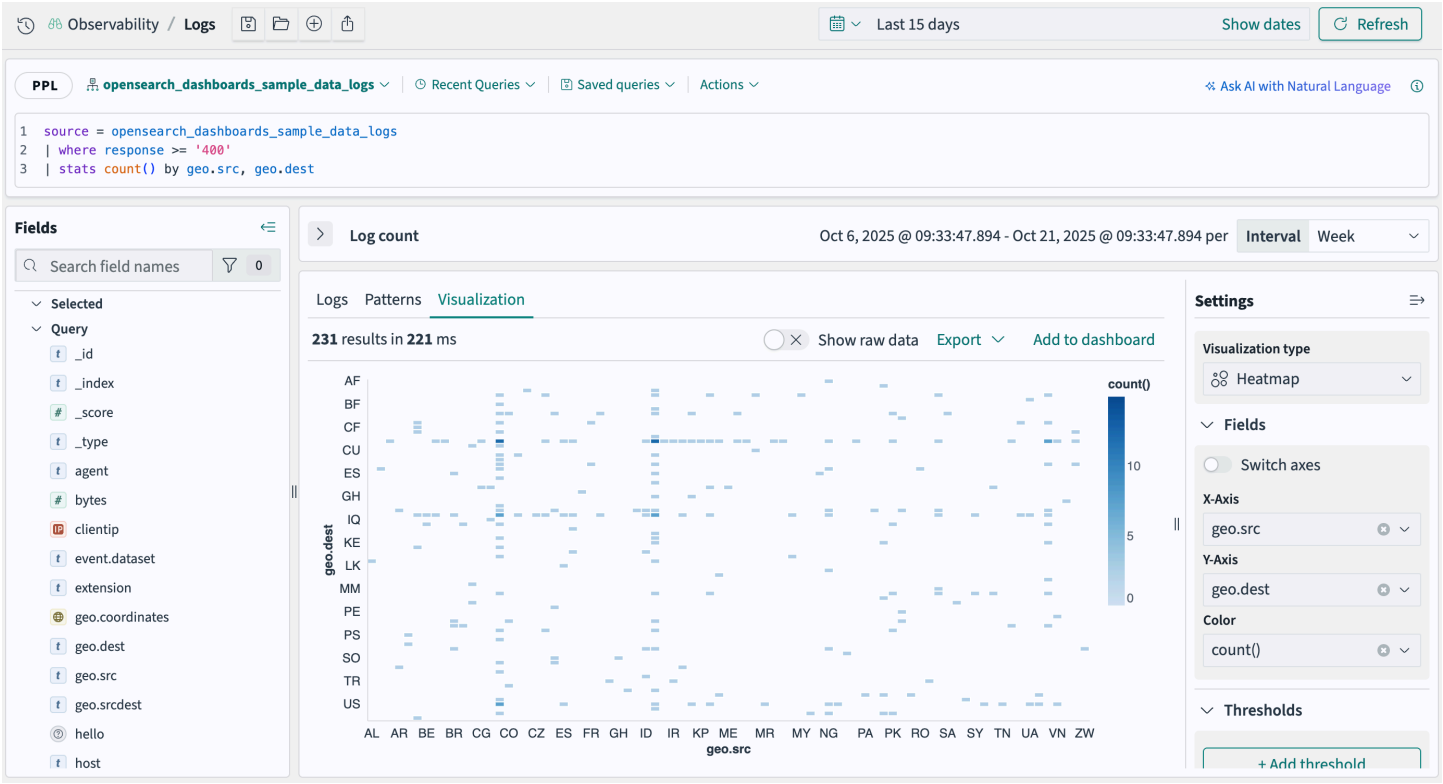
此範例會分析您的日誌，並傳回錯誤和例外狀況。

```
error OR exception
```

如需 DQL 的詳細資訊，請參閱 https：//opensearch.org 上的 DQL 參考手冊。

日誌的儀表板和提醒

在 PPL 的新探索體驗中，您可以從探索中的視覺化索引標籤建立視覺化。從 12 種視覺化類型中選擇並即時編輯，然後再將其新增至儀表板。在舊的探索體驗中，您將瀏覽至左側導覽中的視覺化以建立新的視覺效果，並瀏覽至儀表板以將視覺效果新增至儀表板。



您可以使用 PPL 或 OpenSearch Service 查詢 DSL 定義警示監控，以執行排定的查詢。觸發條件，例如特定數量的錯誤日誌，會觸發提醒。您可以透過 Amazon Simple Notification Service 或 Webhook 等管道傳送通知。

如需提醒的詳細資訊，請參閱 https：//opensearch.org 上的提醒文件。

追蹤

Amazon OpenSearch Service 提供全方位的分散式追蹤功能，可協助您了解應用程式效能，並診斷微服務架構的問題。透過使用 OpenSearch Ingestion 擷取 OpenTelemetry (OTel) 追蹤資料，OpenSearch Service 會自動處理和建構您的遙測資訊，讓您端對端 end-to-end 掌握透過分散式系統的請求流程。

追蹤資料處理和擷取

OpenSearch Ingestion 提供專門的處理器，可在擷取期間標準化和豐富追蹤資料，確保您的遙測遵循一致的模式並準備好進行分析。追蹤資料的關鍵處理器包括：

- `service_map` – 從跨關係自動建置服務相依性圖表，顯示請求如何在服務之間流動。
- `trace_group` – 根據 HTTP 方法和路徑等項目範圍屬性，將相關範圍彙總為邏輯追蹤群組。
- `otel_trace_raw` – 處理原始 OpenTelemetry 追蹤資料，並將跨屬性、資源屬性和檢測範圍資訊擷取至可搜尋的欄位。

OpenSearch UI 和可觀測性工作區

將追蹤資料擷取到 Amazon OpenSearch Service 後，您可以使用 OpenSearch UI 中 Amazon OpenSearch Service 可觀測性工作區提供的工具來分析它。可觀測性工作區提供專門的視覺化和分析工具，旨在協助您了解服務效能、識別瓶頸，以及對分散式架構的問題進行故障診斷。

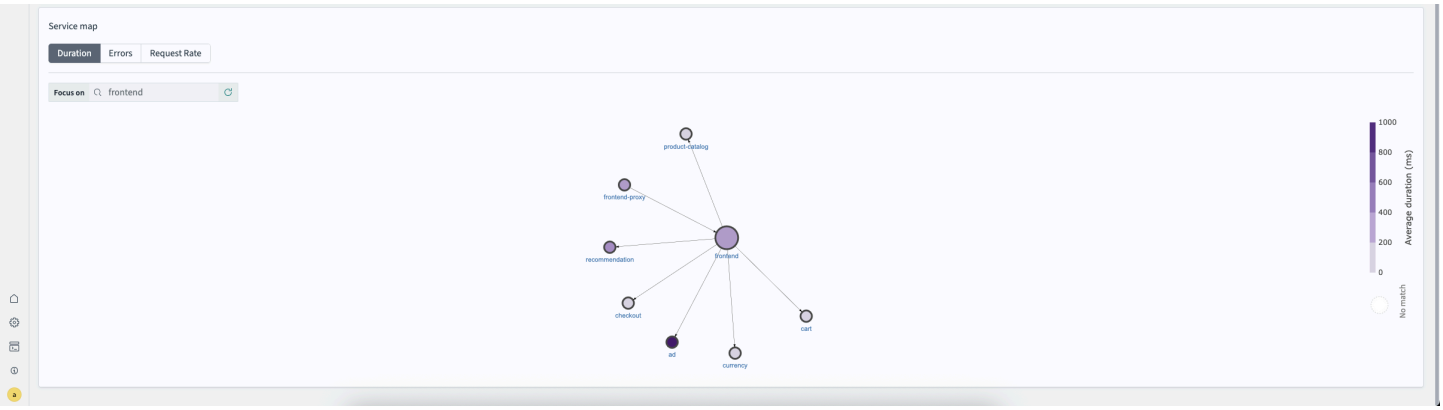
可觀測性工作區包含顯示所有檢測服務的 RED 指標（速率、錯誤率、持續時間）的服務檢視，以及顯示相依性和通訊模式的互動式服務映射。追蹤檢視可讓您使用追蹤 IDs 或跨度 IDs 搜尋特定追蹤，然後深入了解詳細的瀑布圖和跨度分析，以了解整個系統的完整請求歷程。

主要功能

服務檢視

服務檢視透過下列方式提供應用程式運作狀態和效能的完整概觀：

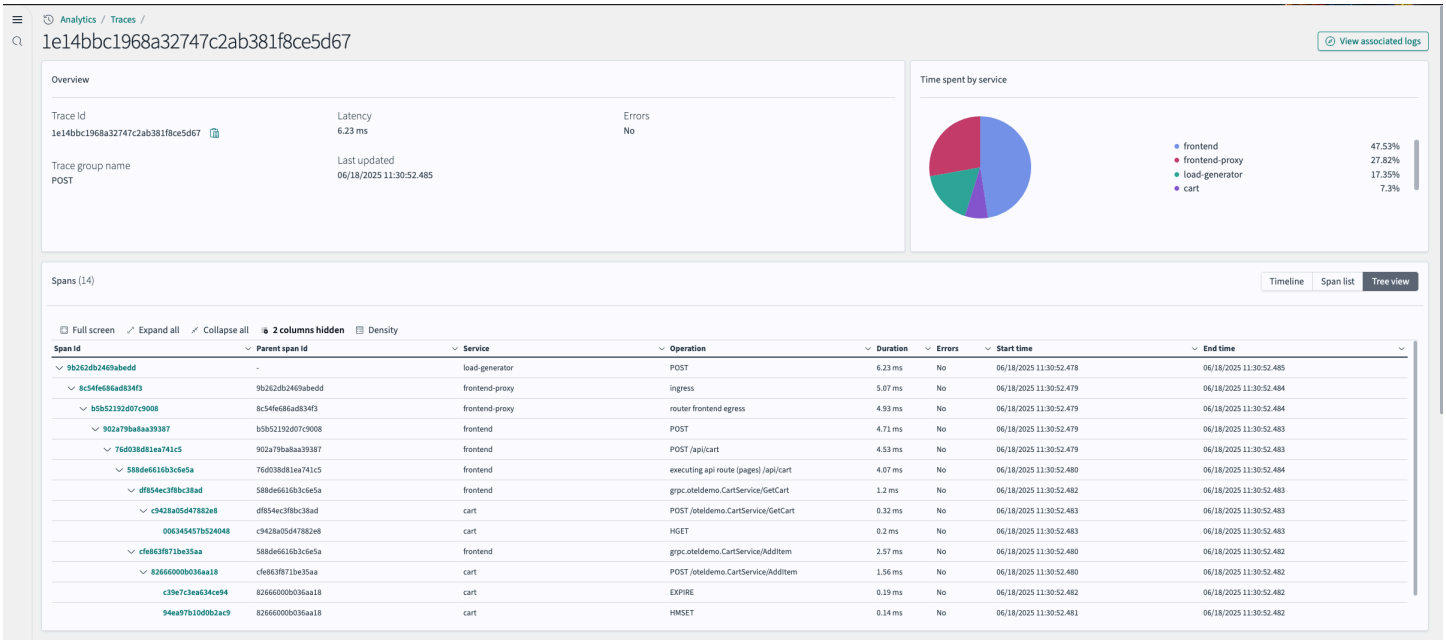
- RED 指標儀表板 – 監控分散式系統中每個服務的速率（每秒請求數）、錯誤率（失敗請求的百分比）和持續時間（延遲百分位數）。這些指標可讓您立即深入了解服務運作狀態，並協助您快速識別效能降低。
- 互動式服務地圖 – 透過自動產生的相依性圖表，視覺化您的服務如何彼此通訊。服務映射會顯示服務之間的請求流程，協助您了解系統架構並識別瓶頸或串聯失敗。
- 服務運作狀態指標 – 根據錯誤率和延遲閾值快速識別有問題的服務。服務以顏色編碼，以突顯需要立即關注的服務，讓您輕鬆排定故障診斷工作的優先順序。
- 服務相互關聯對話方塊 – 從任何服務向下切入，以分析相關日誌和追蹤。此整合檢視會將服務層級指標與詳細的追蹤資料和相關聯的日誌項目連線，在不在不同工具之間切換的情況下實現更快的根本原因分析。



追蹤檢視

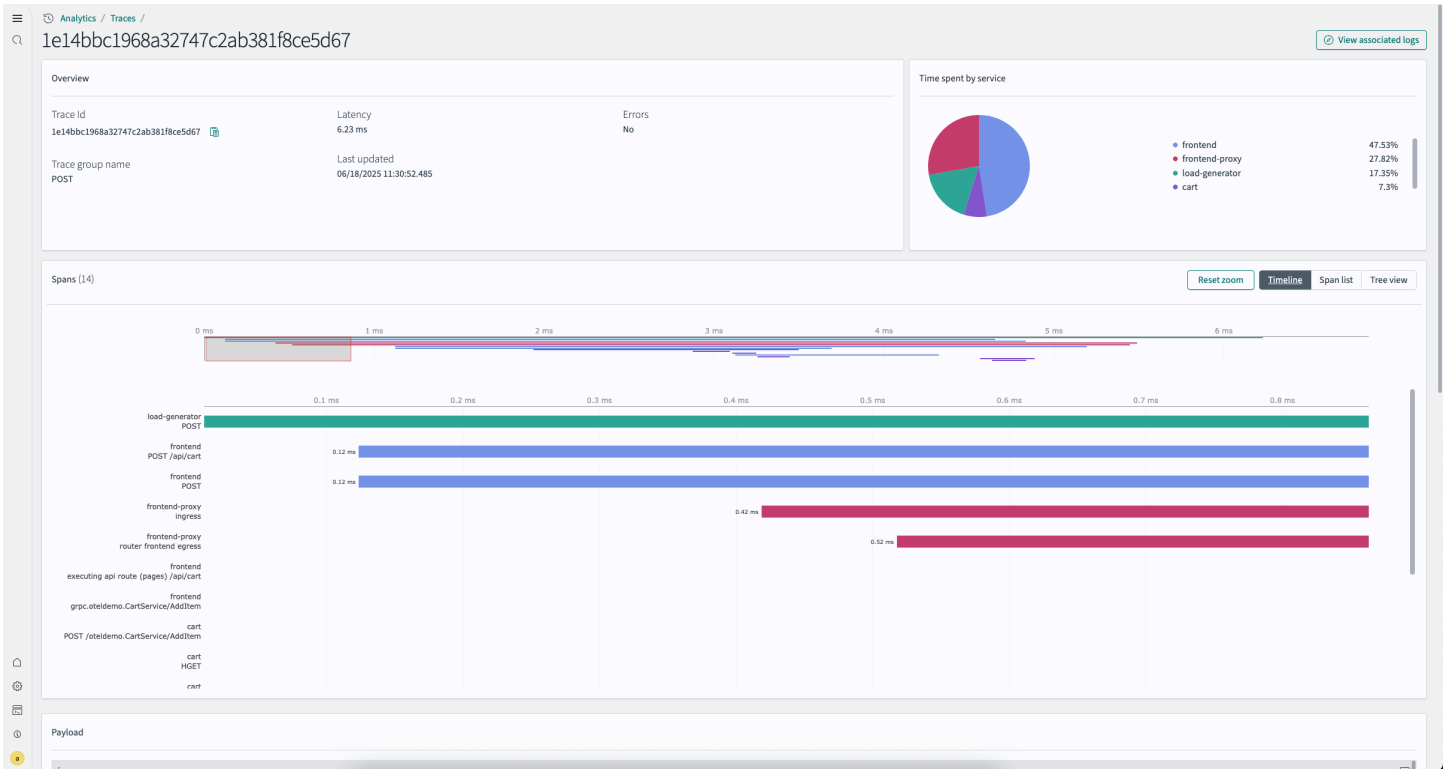
追蹤檢視可讓您透過分散式系統深入調查個別請求：

- 依 HTTP 方法和路徑分組的追蹤 – 根據 API 端點自動將追蹤組織到邏輯群組，顯示彙總指標，例如平均延遲、錯誤率和隨時間推移的效能趨勢。這可協助您識別哪些端點遇到問題，並追蹤效能改善。
- 追蹤 ID 和跨度 ID 搜尋 – 使用追蹤識別符或跨度識別符快速尋找特定追蹤。這在調查使用者報告的問題或與包含追蹤內容的錯誤日誌相關聯時特別有用。
- 瀑布圖 – 在請求流經您的服務時，將請求的完整時間軸視覺化。瀑布檢視會顯示跨時間和持續時間，讓您輕鬆識別慢速操作、循序與平行處理，以及分散式系統中花費的時間。
- 具有階層範圍明細的樹狀檢視 – 導覽跨度之間的父子關係，以了解追蹤中的呼叫階層。此檢視可協助您了解請求如何跨服務分支，並識別哪些服務呼叫造成了整體延遲。
- 關聯的日誌面板 – 檢視與追蹤在相同時間範圍內發生的日誌，依相關服務和追蹤內容篩選。透過在單一界面中提供請求流程和詳細的應用程式日誌，追蹤和日誌之間的這種相互關聯可大幅改善故障診斷。



進階功能

- 關聯分析 – 將追蹤、範圍和服務與對應的日誌無縫連結。可觀測性工作區會使用追蹤內容自動關聯遙測資料，讓您可以在相同請求的不同檢視之間進行樞紐分析，而不會遺失內容。
- 自訂索引名稱和跨叢集支援 – 設定 OpenSearch Service 從自訂索引模式或跨多個 OpenSearch 叢集讀取追蹤資料。此彈性支援複雜的部署案例，並可讓您根據營運需求整理遙測資料。
- 可設定的服務映射限制 – 調整服務映射中顯示的服務和連線數量，以處理大規模拓撲。對於具有數百個服務的系統，您可以篩選映射以專注於特定服務子集，或調整轉譯限制以維持效能。
- Gantt 圖表的迷你地圖導覽 – 使用迷你地圖概觀有效率地導覽大型追蹤瀑布圖。此功能在分析具有許多範圍的追蹤時特別有用，可讓您快速跳到時間軸的不同區段。



追蹤可根據 OpenTelemetry (OTel) 通訊協定資料提供應用程式效能的at-a-glance可見性。它可協助您了解請求如何透過分散式系統，追蹤其跨 服務的end-to-end旅程。

Amazon OpenSearch Service 的操作最佳實務

本章提供了關於操作 Amazon OpenSearch Service 網域的最佳實務，並包括了適用於許多使用案例的一般指導方針。每個工作負載都是獨一無二的，具有獨特的特性，因此沒有任何一個通用建議適合每個使用案例。最重要的最佳實務是在連續週期內部署、測試和調整網域，以找出適合您工作負載的最佳組態、穩定性和成本。

主題

- 監控和提醒
- 碎片策略
- 穩定性
- 效能
- 安全
- 成本最佳化
- Amazon OpenSearch Service 的建議 CloudWatch 警示
- 調整 Amazon OpenSearch Service 網域的大小
- Amazon OpenSearch Service 的 PB 規模
- Amazon OpenSearch Service 中的專用協調器節點
- Amazon OpenSearch Service 中的專用主節點

監控和提醒

下列最佳實務適用於監控您的 OpenSearch Service 網域。

設定 CloudWatch 警示

OpenSearch Service 會向 Amazon CloudWatch 發出效能指標。定期檢閱叢集和執行個體指標，並根據您的工作負載效能來配置建議的 CloudWatch 警示。

啟用日誌發佈

OpenSearch Service 會在 Amazon CloudWatch Logs 中公開 OpenSearch 錯誤日誌、搜尋慢速日誌、索引慢速日誌和稽核日誌。搜尋慢速日誌、索引慢速日誌和錯誤日誌對於疑難排解效能和穩定性問題非常有用。稽核日誌會追蹤使用者活動，只有在啟用精細存取控制時才能使用。如需詳細資訊，請參閱 OpenSearch 文件中的日誌。

搜尋慢速日誌和索引慢速日誌是了解和疑難排解搜尋和索引操作效能的重要工具。對所有生產網域啟用搜尋和索引慢速日誌傳送。您還必須設定記錄閾值，否則 CloudWatch 將不會擷取日誌。

碎片策略

碎片會將您的工作負載分配到 OpenSearch Service 網域中的資料節點。正確設定的索引有助於提升整體網域效能。

當您傳送資料至 OpenSearch Service 時，您會將該資料傳送至索引。索引類似於資料庫的資料表，以文件作為列，欄位作為行。當您建立索引時，會告知 OpenSearch 您要建立多少個主要碎片。主碎片是完整資料集的獨立分區。OpenSearch Service 會自動將您的資料分配到索引中的主要碎片。您也可以設定索引複本。每個複本都包含該索引之主要碎片的完整副本集。

OpenSearch Service 會將每個索引的碎片對應至叢集中的資料節點。它可確保索引的主要碎片和複本碎片駐留在不同的資料節點上。第一個複本可確保索引中有兩個資料副本。您應該始終使用至少一個複本。額外的複本提供額外的備援和讀取容量。

OpenSearch 會將索引請求傳送至所有資料節點，這些節點包含屬於索引的碎片。它首先將索引請求傳送至包含主要碎片的資料節點，然後傳送至包含複本碎片的資料節點。搜尋請求會由協調器節點路由至屬於索引之所有碎片的主要或複本碎片。

例如，對於具有五個主要碎片和一個複本的索引，每個索引編製請求接觸 10 個碎片。相反，搜尋請求會傳送至 n 個碎片，其中 n 是主要碎片的數量。對於具有五個主要碎片和一個複本的索引，每個搜尋查詢接觸該索引中的五個碎片 (主要碎片或複本碎片)。

確定碎片和資料節點計數

請使用下列最佳實務來確定網域的碎片計數和資料節點計數。

碎片大小 - 磁碟上的資料大小是來源資料大小的直接結果，並且會隨著您編製更多資料的索引而變更。來源與索引的比例可能會有很大的不同，從 1:10 到 10:1 或更大，但通常大約在 1:1.10 左右。您可以使用該比率來預測磁碟上的索引大小。您也可以為某些資料建立索引，並擷取實際索引大小，以確定工作負載的比率。掌握預測的索引大小後，設定碎片計數，以便每個碎片介於 10–30 GiB 之間 (對於搜尋工作負載)，或介於 30-50 GiB 之間 (對於日誌工作負載)。50 GiB 應該是最大值 – 確保為增長做好計劃。

碎片計數 - 資料節點的碎片分佈對網域的效能有很大影響。當您的索引具有多個碎片時，請嘗試將碎片計數設為資料節點計數的倍數。這有助於確保碎片在資料節點之間均勻分佈，並防止熱節點。例如，如果您有 12 個主要碎片，則資料節點計數應為 2、3、4、6 或 12。但是，碎片計數不若碎片大小重要 – 如果您有 5 GiB 的資料，則仍應使用單一碎片。

每個資料節點的碎片 - 節點可容納的碎片總數與節點的 Java 虛擬機器 (JVM) 堆積記憶體成正比。目標是每 GiB 堆積記憶體有 25 個或更少的碎片。例如，具有 32 GiB 堆積記憶體的節點應保留不超過 800 個碎片。雖然碎片分佈可能會根據您的工作負載模式而有所不同，但 Elasticsearch 和 OpenSearch 1.1 到 2.15 的每個節點限制為 1，000 個碎片，OpenSearch 2.17 及更高版本限制為 4，000 個碎片。cat/allocation API 可快速了解資料節點的碎片數量和碎片總體儲存。

碎片與 CPU 比率 - 當碎片涉及索引或搜尋請求時，它會使用 vCPU 來處理請求。最佳實務是使用每個碎片 1.5 vCPU 的初始縮放點。如果您的執行個體類型有 8 個 vCPU，請設定資料節點計數，讓每個節點的碎片不超過六個。請注意，這是一個近似值。請務必測試您的工作負載並相應調整叢集的規模。

如需有關儲存磁碟區、碎片大小和執行個體類型的建議，請參閱下列資源：

- the section called “調整網域大小”
- the section called “PB 規模”

避免儲存扭曲

儲存扭曲是指叢集中的一個或多個節點對一個或更多索引的儲存比例高於其他節點。儲存扭曲表示包含 CPU 使用率不均、間歇性和不均勻的延遲，以及跨資料節點的佇列不均勻。若要判斷是否有扭曲問題，請參閱下列疑難排解部分：

- the section called “節點碎片和儲存扭曲”
- the section called “索引碎片和儲存扭曲”

穩定性

下列最佳實務適用於維護穩定且健康的 OpenSearch Service 網域。

與 OpenSearch 保持同步

服務軟體更新

OpenSearch Service 會定期發行軟體更新以新增功能或強化您的網域。更新不會變更 OpenSearch 或 Elasticsearch 引擎版本。建議您安排一段重複時間來執行 DescribeDomain API 操作，如果 `UpdateStatus` 為 `ELIGIBLE`，則觸發服務軟體更新。如果您未在特定的時間範圍內 (通常為兩週) 更新網域，OpenSearch Service 會自動執行更新。

OpenSearch 版本升級

OpenSearch Service 會定期新增對社群維護的 OpenSearch 版本的支援。請務必在最新 OpenSearch 版本可用時進行升級。

OpenSearch Service 會同時升級 OpenSearch 和 OpenSearch Dashboards (如果您的網域正在執行舊版引擎，則同時升級 Elasticsearch 和 Kibana)。如果叢集有專用主節點，無須停機即可完成升級。否則叢集在選擇主節點時，可能會在升級後數秒沒有回應。OpenSearch Dashboards 在升級的部分或所有期間可能會無法使用。

有兩種方式可以升級網域：

- 就地升級 – 此選項更容易，因為您保留相同的叢集。
- 快照/還原升級 - 此選項適用於在新叢集上測試新版本或在叢集之間遷移。

無論您使用哪種升級程序，我們都建議您維護一個僅用於開發和測試的網域，並在升級您的生產網域之前先將其升級到新版本。建立測試網域時，對於部署類型，選擇 Development and testing (開發與測試)。請務必在網域升級後立即將所有用戶端升級至相容版本。

改善快照效能

為避免快照卡在處理中，專用主節點的執行個體類型應與碎片計數相符。如需詳細資訊，請參閱the section called “選擇專用主節點的執行個體類型”。此外，每個節點的 Java 堆積記憶體每個 GiB 不應超過建議的 25 個碎片。如需詳細資訊，請參閱the section called “選擇碎片數”。

啟用專用主節點

專用主節點可改善叢集穩定性。專用主節點會執行叢集管理任務，但不會保留索引資料或回應用戶端請求。此叢集管理任務的卸載可增加網域的穩定性，並可在不停機的情況下進行一些組態變更。

啟用並使用三個專用主節點，以在三個可用區域中獲得最佳的網域穩定性。使用異地同步備份與待命進行部署會為您設定三個專用主節點。如需執行個體類型建議，請參閱 the section called “選擇專用主節點的執行個體類型”。

在多個可用區域中進行部署

為避免因服務中斷造成的資料遺失並盡量減少叢集停機時間，您可以在相同 AWS 區域中將節點發佈至兩個或三個可用區域。最佳實務是使用異地同步備份搭配待命進行部署，這會設定三個可用區域，其中兩個區域作用中，一個做為待命，每個索引有兩個複本碎片。此組態可讓 OpenSearch Service 將複本碎片分配到與其對應的主碎片不同的可用區域。可用區域之間的叢集通訊無需支付跨可用區域資料傳輸費用。

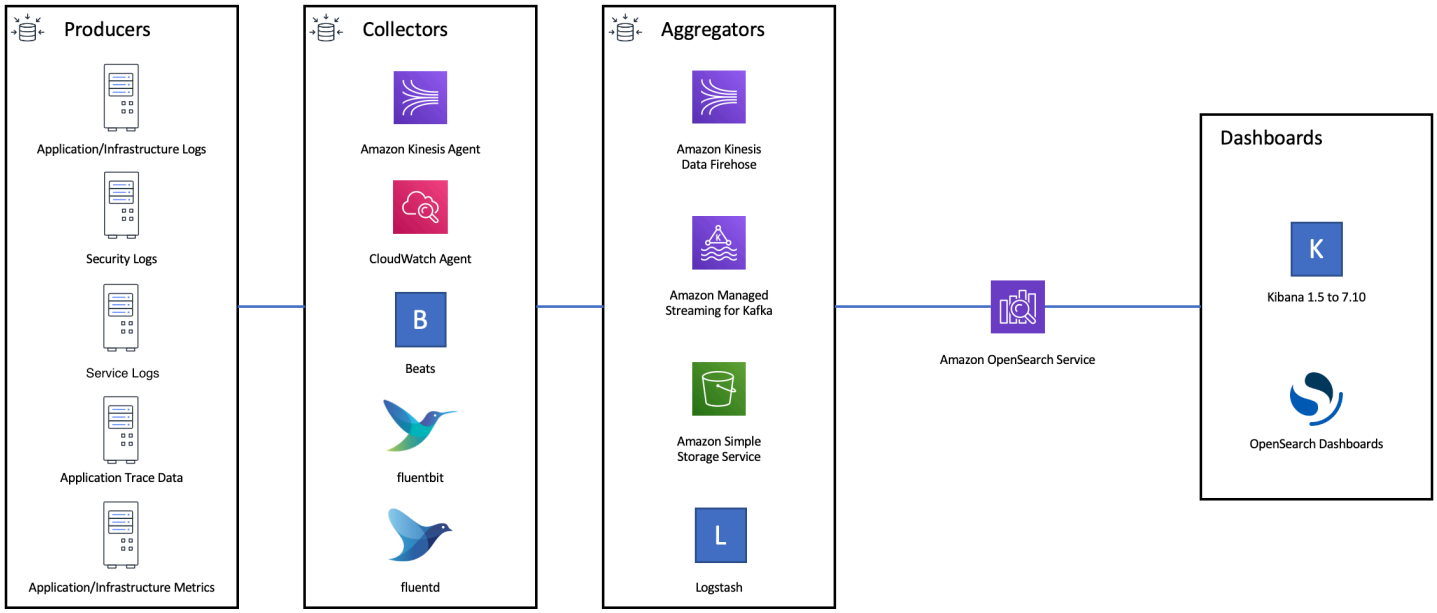
可用區域是每個 區域內的隔離位置。如果使用雙可用區域組態，遺失一個可用區域表示喪失了一半的網域容量。移至三個可用區域可進一步減少遺失單個可用區域的影響。

控制擷取流量和緩衝

我們建議您使用 _bulk API 操作來限制全部請求計數。傳送一個包含 5,000 個文件的 _bulk 請求比傳送包含單個文件的 5,000 個請求更有效率。

為了獲得最佳操作穩定性，有時需要限制甚至暫停索引請求的上游流程。限制索引請求的速率是處理請求中意外或偶爾出現的峰值的一種重要機制，否則可能會拖垮叢集。考慮在您的上游架構中建置流量控制機制。

下圖顯示日誌擷取架構的多個元件選項。設定彙整層，以允許有足夠的空間來緩衝傳入資料，從而應對突然的流量峰值和短暫的網域維護。



建立搜尋工作負載的映射

對於搜尋工作負載，請建立映射，定義 OpenSearch 如何儲存和索引文件及其欄位。將 dynamic 設定為 strict，防止意外新增新欄位。

```
PUT my-index
{
  "mappings": {
    "dynamic": "strict",
    "properties": {
      "title": { "type" : "text" },
```

```
      "author": { "type" : "integer" },
      "year": { "type" : "text" }
    }
  }
}
```

使用索引範本

您可以使用索引範本作為告知 OpenSearch 如何在建立索引時進行設定的方法。在建立索引之前先設定索引範本。然後，當您建立索引時，它會繼承範本中的設定和映射。您可以將多個範本套用至單個索引，以便您可以在一個範本中指定設定，並在另一個範本中指定映射。此策略允許多個索引中的一個通用設定版本，以及針對更具體設定和映射的單獨範本。

下列設定對於在範本中進行設定很有用：

- 主要碎片和複本碎片的數量
- 重新整理間隔 (重新整理和對索引進行最新變更以供搜尋的頻率)
- 動態映射控制
- 明確欄位映射

下列範例範本包含每個設定：

```
{
   "index_patterns":[
      "index-*"
   ],
   "order": 0,
   "settings": {
      "index": {
         "number_of_shards": 3,
         "number_of_replicas": 1,
         "refresh_interval": "60s"
      }
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "field_name1": {
            "type": "keyword"
         }
```

```
        }
    }
}
```

即使它們很少變更，在 OpenSearch 中集中定義設定和映射，相比更新多個上游用戶端更容易管理。

使用索引狀態管理功能來管理索引

如果您要管理日誌或時間序列資料，建議您使用索引狀態管理 (ISM)。ISM 可讓您自動化定期索引生命週期管理任務。使用 ISM，您可以建立政策來叫用索引別名轉換、建立索引快照、在儲存層之間移動索引，以及刪除舊索引。您甚至可以使用 ISM 轉換操作替代資料生命週期管理策略，以避免碎片扭曲。

首先，設定 ISM 政策。如需範例，請參閱 the section called “範例政策”。接著，將政策連接至一個或多個索引。如果在政策中包含一個 ISM 範本欄位，OpenSearch Service 會將該政策自動套用至符合指定模式的任何索引。

移除未使用的索引

定期檢閱叢集中的索引，並識別任何未使用的索引。擷取這些索引的快照，以便它們儲存在 S3 中，然後刪除它們。移除未使用的索引時，您可以減少碎片計數，並可使節點之間的儲存分佈和資源使用率更平衡。即使處於閒置狀態，索引仍會在內部索引維護活動期間耗用部分資源。

您可以使用 ISM 在一段時間後自動建立快照並刪除索引，而不是手動刪除未使用的索引。

使用多個網域實現高可用性

要在多個區域實現 99.9% 執行時間以上的高可用性，請考慮使用兩個網域。對於小型或緩慢變更的資料集，您可以設定跨叢集複寫以維護主動-被動模型。在此模型中，只能寫入到領導網域，但是可以從任一網域中讀取。對於較大的資料集和快速變更的資料，請在擷取管道中設定雙重傳遞，以便將所有資料獨立寫入主動-主動模型中的兩個網域。

建構您的上游和下游應用程式，同時考量容錯移轉。請務必測試容錯移轉程序以及其他災難復原程序。

效能

以下最佳實務適用於調整您的網域以獲得最佳效能。

最佳化批量請求大小和壓縮

批量大小取決於您的資料、分析和叢集組態，但是每個批量請求的最佳起點是 3 到 5 MiB。

使用 gzip 壓縮來傳送請求並接收來自 OpenSearch 網域的回應，以減少請求和回應的承載大小。您可以搭配使用 gzip 壓縮與 OpenSearch Python 用戶端，也可以從用戶端包括以下標頭：

- `'Accept-Encoding': 'gzip'`
- `'Content-Encoding': 'gzip'`

若要優化大量請求大小，請先從 3 MiB 的大量請求大小開始。然後，慢慢增加請求大小，直到索引效能停止改善為止。

> **Note**
>
> 若要在執行 Elasticsearch 6.x 版的網域上啟用 gzip 壓縮，您必須在叢集層級設定 `http_compression.enabled`。此設定在 Elasticsearch 7.x 版和所有版本的 OpenSearch 中預設為 true。

減少批量請求回應的大小

若要減少 OpenSearch 回應的大小，請使用 `filter_path` 參數排除不必要的欄位。請確保不要排除用於識別或重試失敗請求所需的任何欄位。如需詳細資訊和範例，請參閱 the section called “縮減回應大小”。

調校重新整理間隔

OpenSearch 索引具有最終讀取一致性。重新整理操作使得索引上執行的所有更新可用於搜尋。預設的重新整理間隔為一秒鐘，這表示 OpenSearch 會在寫入索引時每秒執行一次重新整理。

重新整理索引的頻率越低 (重新整理間隔越高)，整體索引效能就越好。增加重新整理間隔的折中方案是索引更新與新資料可供搜尋之間有較長的延遲時間。在您能忍受的範圍內，將重新整理間隔設定得盡可能高，以改善整體效能。

我們建議將所有索引的 `refresh_interval` 參數設為 30 秒或更長時間。

啟用自動調整

自動調整會使用 OpenSearch 叢集中的效能和使用量指標，以建議對節點上的佇列大小、快取大小以及 Java 虛擬機器 (JVM) 設定進行變更。這些選擇性變更可提高叢集速度與穩定性。您可以隨時還原為預設的 OpenSearch Service 設定。在新網域上預設啟用自動調整，除非您明確停用它。

我們建議您在所有網域上啟用自動調整，並設定週期性維護時段或定期檢閱其建議。

安全

以下最佳實務適用於保護您的網域。

啟用精細存取控制

精細存取控制可讓您控制哪些人可以存取 OpenSearch Service 網域內的特定資料。與一般存取控制相比，精細存取控制可為每個叢集、索引、文件和欄位提供其自己指定的存取政策。存取條件可以基於許多因素，包括請求存取之人員的角色，以及他們打算對資料執行的動作。例如，您可能會授予一位使用者寫入索引的存取權，而授予另一位使用者僅讀取索引上資料的存取權，而不能進行任何變更。

精細存取控制可讓具有不同存取需求的資料存在於相同的儲存空間中，而不會遇到安全性或合規性問題。

我們建議在您的網域中啟用精細存取控制。

在 VPC 內部署網域

在虛擬私有雲端 (VPC) 中放置 OpenSearch Service 網域可在 VPC 中的 OpenSearch Service 和其他服務之間進行安全通訊，而不需要網際網路閘道、NAT 裝置或 VPN 連接。所有流量都會安全地保留在 AWS 雲端內。因為其邏輯隔離，相較於使用公有端點的網域，位於 VPC 內的網域具有額外的安全層。

我們建議您在 VPC 內建立您的網域。

套用限制性存取政策

即使您的網域部署在 VPC 中，最佳實務是分層實作安全性。確保針對您目前的存取政策檢查組態。

將限制性以資源為基礎的存取政策套用至網域，並在授予組態 API 和 OpenSearch API 操作的存取權時遵循最低權限準則。一般而言，請避免在存取政策中使用匿名使用者主體 `"Principal": {"AWS": "*" }`。

不過，在某些情況下，您可以接受使用開放存取政策，例如當您啟用精細存取控制時。開放存取政策可讓您在請求簽署困難或不可能的情況下 (例如來自特定用戶端和工具) 存取網域。

啟用靜態加密

OpenSearch Service 網域提供靜態資料加密，可協助防止未經授權存取您的資料。靜態加密會使用 AWS Key Management Service (AWS KMS) 來存放和管理加密金鑰，並使用 256 位元金鑰 (AES-256) 的進階加密標準演算法來執行加密。

如果您的網域存放了敏感資料，請啟用靜態資料加密。

啟用節點對節點加密

節點對節點加密在 OpenSearch Service 的預設安全功能之上提供額外安全層級。會為佈建在 OpenSearch 中的節點之間的所有通訊實施 Transport Layer Security (TLS)。節點對節點加密，在節點之間分配和複寫資料時，透過 HTTPS 傳送至 OpenSearch Service 網域的任何資料在傳輸過程中保持加密狀態。

如果您的網域存放了敏感資料，請啟用節點對節點加密。

使用 監控 AWS Security Hub CSPM

使用 監控 OpenSearch Service 的使用情況，因為它與安全最佳實務相關AWS Security Hub CSPM。Security Hub CSPM 使用安全控制來評估資源組態和安全標準，以協助您遵守各種合規架構。如需使用 Security Hub CSPM 評估 OpenSearch Service 資源的詳細資訊，請參閱AWS Security Hub 《 使用者指南》中的Amazon OpenSearch Service 控制項。

成本最佳化

下列最佳實務適用於最佳化及節省您的 OpenSearch Service 成本。

使用最新一代執行個體類型

OpenSearch Service 一直採用新的 Amazon EC2 執行個體類型以較低的成本提供更好的效能。我們建議始終使用最新一代的執行個體。

對於生產網域，避免使用 T2 或 `t3.small` 執行個體，因為在持續的高負載下，它們可能會變得不穩定。`r6g.large` 執行個體是小型生產工作負載 (資料節點和專用主節點) 的一種選擇。

使用最新的 Amazon EBS gp3 磁碟區

OpenSearch 資料節點需要低延遲和高輸送量儲存，才能提供快速的索引和查詢。透過使用 Amazon EBS gp3 磁碟區，您能夠以比之前提供的 Amazon EBS gp2 磁碟區類型低 9.6% 的成本，獲得更高的基準效能 (IOPS 和輸送量)。您可以使用 gp3 佈建額外的 IOPS 和輸送量，而不受磁碟區大小影響。這些磁碟區也比上一代磁碟區更穩定，因為它們不會使用爆量額度。gp3 磁碟區類型也會將 gp2 磁碟區類型的每個資料節點磁碟區大小限制加倍。您可以使用這些更大的磁碟區，透過提高每個資料節點的儲存量來降低被動資料的成本。

使用 UltraWarm 和冷儲存來處理時間序列日誌資料

如果您使用 OpenSearch 進行日誌分析，請將您的資料移至 UltraWarm 或冷儲存以降低成本。使用索引狀態管理 (ISM) 在儲存層之間遷移資料並管理資料保留。

UltraWarm 提供經濟有效的方法，可在 OpenSearch Service 中儲存大量唯讀資料。UltraWarm 使用 Amazon S3 進行儲存，這意味著資料是不可變的，而且只需要一個副本。您只需支付相當於索引中主要碎片大小的儲存。UltraWarm 查詢的延遲時間會隨著服務該查詢所需的 S3 資料量而增加。在節點上快取資料之後，對 UltraWarm 索引執行查詢會類似於對熱索引的查詢。

冷儲存也由 S3 支持。當您需要查詢冷資料時，可以選擇性地將其連接到現有的 UltraWarm 節點。冷資料會產生與 UltraWarm 相同的受管儲存成本，但冷儲存中的物件不會使用 UltraWarm 節點資源。因此，冷儲存可提供大量儲存容量，而不會影響 UltraWarm 節點大小或計數。

當您有大約 2.5 TiB 的資料要從熱儲存遷移時，UltraWarm 會變得符合成本效益。監控您的填充率，並計劃在達到該資料量之前將索引移至 UltraWarm。

檢閱預留執行個體的建議

在對效能和運算耗用量有了良好的基準之後，考慮購買預留執行個體 (RI)。對於無預付款的 1 年預訂，折扣從 30% 左右開始；對於所有預付款的 3 年承諾，折扣可以增加到 50%。

在您觀察到至少 14 天的穩定操作後，請檢閱AWS Cost Management 《 使用者指南》中的存取保留建議。Amazon OpenSearch Service 標題會顯示特定 RI 購買建議和預估節省成本。

Amazon OpenSearch Service 的建議 CloudWatch 警示

當 CloudWatch 指標在經過一些時間超過指定的值時，CloudWatch 警示會執行動作。例如，如果您 AWS 的叢集運作狀態超過一分鐘`red`，建議您傳送電子郵件給您。本節包含 Amazon OpenSearch Service 的一些建議警示以及回應方式。

您可以使用 自動部署這些警示 CloudFormation。如需範例堆疊，請參閱相關的 GitHub 儲存庫。

> **Note**
>
> 如果您部署 CloudFormation 堆疊，`KMSKeyError` 和 `KMSKeyInaccessible` 警示將存在於 `Insufficient Data` 狀態，因為只有在網域遇到其加密金鑰問題時才會顯示這些指標。

如需有關設定警示的詳細資訊，請參閱 《Amazon CloudWatch 使用者指南》中的建立 Amazon CloudWatch 警示。

警示	問題
ClusterStatus.red 上限為 >= 1 達 1 分鐘，連續 1 次	至少一個主要碎片及其複本不會分配到節點。請參閱 the section called “紅色叢集狀態”。
ClusterStatus.yellow 上限為 >= 1 持續 1 分鐘，連續 5 次	至少一個複本碎片不會分配到節點。請參閱 the section called “黃色叢集狀態”。
FreeStorageSpace 下限為 <= 20480 達 1 分鐘，連續 1 次	您叢集內的節點縮減至 20 GiB 的可用儲存空間。請參閱 the section called “缺少可用儲存空間”。此值的單位為 MiB，所以建議您將其設為每個節點的 25% 儲存空間，而不是 20480。
ClusterIndexWritesBlocked 為 >= 1 達 5 分鐘，連續 1 次	您的叢集正在封鎖寫入請求。請參閱 the section called “ClusterBlockException”。
Nodes下限為 < x 達 1 天，連續 1 次	x 是您叢集中的節點數。此警示表示您叢集中至少有一個節點已無法連線達 1 天時間。請參閱 the section called “叢集節點失敗”。

AutomatedSnapshotFailure 上限為 >= 1 達 1 分鐘，連續 1 次	自動快照失敗。此故障通常是紅色叢集運作狀態的結果。請參閱 the section called “紅色叢集狀態”。
	
	如需所有自動快照的摘要和一些有關故障的資訊，請嘗試以下其中一個請求：
	
	`GET domain_endpoint /_snapshot/cs-automated/_all`
	`GET domain_endpoint /_snapshot/cs-automated-enc/_all`

CPUUtilization 或 WarmCPUUtilization 上限為	有時可能會出現 100% CPU 使用率，但持續高用量會有問題。可考慮使用較大的執行個體類型或新增執行個體。

警示	問題
>= 80%，15 分鐘，連續 3 次	
JVMMemory Pressure 上限為 >= 95% 達 1 分鐘，連續 3 次	如果使用量增加，叢集可能遇到記憶體不足錯誤。可考慮垂直擴展。OpenSearch Service 針對 Java 堆積使用執行個體 RAM 的一半，堆積大小最多可達 32 GiB。您可以垂直擴展執行個體高達 64 GiB 的 RAM，屆時便能透過新增執行個體進行水平擴展。
OldGenJVM MemoryPressure 上限為 >= 80% 達 1 分鐘，連續 3 次	
MasterCPU Utilization 上限為 >= 50% 達 15 分鐘，連續 3 次	可考慮使用較大的執行個體類型為您的專用主節點。因為其在叢集穩定性中的角色和藍/綠部署，專用主節點應該具有比資料節點較低的 CPU 使用量。
MasterJVM MemoryPressure 上限為 >= 95% 達 1 分鐘，連續 3 次	
MasterOld GenJVMMem oryPressure 上限為 >= 80% 達 1 分鐘，連續 3 次	
KMSKeyError 為 >= 1 達 1 分鐘，連續 1 次	用於加密網域中靜態資料的 AWS KMS 加密金鑰已停用。重新啟用它來恢復正常操作。如需詳細資訊，請參閱the section called “靜態加密”。
KMSKeyIna ccessible 為 >= 1 達 1 分鐘，連續 1 次	用於加密網域中靜態資料的 AWS KMS 加密金鑰已刪除或已撤銷其對 OpenSearch Service 的授予。您無法復原此狀態的網域。但是，如果您有手動快照，您可以使用它來遷移至新網域。如需詳細資訊，請參閱 the section called “靜態加密”。

警示	問題
shards.active 為 >= 30000 達 1 分鐘，連續 1 次	作用中主要碎片和複本碎片的總數大於 30,000。您可能太頻繁地輪換索引。考慮使用 ISM 在索引達到特定使用期限後將其移除。
5*xx* 警示 >= OpenSearchRequests 的 10%	一或多個資料節點可能會過載，或是請求無法在閒置逾時期間內完成。請考慮切換到較大型執行個體類型或在叢集中新增更多節點。確認您遵循碎片和叢集架構的最佳實務。
MasterReachableFromNode 最長 5 分鐘 < 1，連續 1 次	此警示表示主節點已停止或無法存取。這些失敗通常是網路連線問題或 AWS 相依性問題的結果。
ThreadpoolWriteQueue 平均為 >= 100 達 1 分鐘，連續 1 次	叢集正在經歷高索引並行狀況。檢閱和控制索引請求，或增加叢集資源。
ThreadpoolSearchQueue 平均為 >= 500 達 1 分鐘，連續 1 次	叢集正在經歷高搜尋並行狀況。考慮擴展您的叢集。您也可以增加搜尋佇列大小，但過度增加可能會導致記憶體不足錯誤。
ThreadpoolSearchQueue 上限為 >= 5000 達 1 分鐘，連續 1 次	
ThreadpoolSearchRejected SUM 增加 >=1{ 數學表達式 DIFF ( )}，持續 1 分鐘，連續 1 次	這些警示會通知您可能會影響效能和穩定性的網域問題。

警示	問題
ThreadpoolWriteRejected SUM 增加 >=1{ 數學表達式 DIFF ( )}，持續 1 分鐘，連續 1 次	

> **Note**
>
> 如果您只是想檢視指標，請參閱 the section called “監控叢集指標”。

您可能會考慮的其他警示

根據您經常使用的 OpenSearch Service 功能，請考慮設定下列警示。

警示	問題
WarmFreeStorageSpace 為 >= 10%	您已達到可用暖儲存總量的 10%。 會WarmFreeStorageSpace 測量 MiB 中可用暖儲存空間的總和。UltraWarm 使用 Amazon S3 而非連接的磁碟。
HotToWarmMigrationQueueSize 為 >= 20 達 1 分鐘，連續 3 次	大量的索引正在同時從熱儲存移至 UltraWarm 儲存。考慮擴展您的叢集。
HotToWarmMigrationSuccessLatency 為 >= 1 天，連續 1 次	設定此警示，以便在您嘗試滾動每日索引，HotToWarmMigrationSuccessCount x 延遲大於 24 小時時收到通知。
WarmJVMMemoryPressure 上	如果使用量增加，叢集可能遇到記憶體不足錯誤。可考慮垂直擴展。OpenSearch Service 針對 Java 堆積使用執行個體 RAM 的一半，

警示	問題
限為 >= 95% 達 1 分鐘，連續 3 次	堆積大小最多可達 32 GiB。您可以垂直擴展執行個體高達 64 GiB 的 RAM，屆時便能透過新增執行個體進行水平擴展。
WarmOldGenJVMMemoryPressure 上限為 >= 80% 達 1 分鐘，連續 3 次	
WarmToColdMigrationQueueSize 為 >= 20 達 1 分鐘，連續 3 次	大量的索引正在同時從 UltraWarm 移至冷儲存。考慮擴展您的叢集。
HotToWarmMigrationFailureCount 為 >= 1 達 1 分鐘，連續 1 次	遷移可能在快照、碎片重新配置或強制合併期間失敗。快照或碎片重新配置期間的失敗通常是因為節點故障或 S3 連線問題。磁碟空間不足通常是強制合併失敗的根本原因。
WarmToColdMigrationFailureCount 為 >= 1 達 1 分鐘，連續 1 次	嘗試將索引中繼資料遷移至冷儲存裝置失敗時，遷移通常會失敗。移除熱索引叢集狀態時也可能發生故障。
WarmToColdMigrationLatency 為 >= 1 天，連續 1 次	設定此警示，以便在您嘗試滾動每日索引，WarmToColdMigrationSuccessCount x 延遲大於 24 小時時收到通知。
AlertingDegraded 為 >= 1 達 1 分鐘，連續 1 次	提醒索引為紅色，或是有一或多個節點不在排程上。

警示	問題
ADPluginUnhealthy 為 >= 1 達 1 分鐘，連續 1 次	異常偵測外掛程式無法正常運作，原因是高故障率或使用的其中一個索引是紅色。
AsynchronousSearchFailureRate 為 >= 1 達 1 分鐘，連續 1 次	最後一分鐘內至少有一個非同步搜尋失敗，這可能表示協調器節點失敗。非同步搜尋請求的生命週期僅在協調器節點上受管，因此如果協調器停機，請求即會失敗。
AsynchronousSearchStoreHealth 為 >= 1 達 1 分鐘，連續 1 次	持續性索引中非同步搜尋回應存放區的運作狀態為紅色。您可能正在儲存大型非同步回應，這可能會破壞叢集的穩定性。請嘗試將您的非同步搜尋回應限制在 10 MB 以下。
SQLUnhealthy 為 >= 1 達 1 分鐘，連續 3 次	SQL 外掛程式正在傳回 5xx 回應碼或將無效的查詢 DSL 傳遞至 OpenSearch。針對用戶端向外掛程式提出的請求進行疑難排解。
LTRStatus.red 為 >= 1 達 1 分鐘，連續 1 次	至少有一個執行 Learning to Rank 外掛程式所需的索引缺少主要碎片，並且無法運作。

調整 Amazon OpenSearch Service 網域的大小

沒有完美的方法來調整 Amazon OpenSearch Service 網域大小。但是，透過開始了解您的儲存需求、服務和 OpenSearch 本身，您仍可以初步預估您的硬體需求。此預估可以做為調整網域大小最關鍵環節的實用起點：使用代表性工作負載進行測試和監控其效能。

主題

- 計算儲存需求
- 選擇碎片數
- 選擇執行個體類型並進行測試

計算儲存需求

大多數的 OpenSearch 工作負載為兩大類別之一：

- 長效索引：您編寫程式碼將資料處理到一個或多個 OpenSearch 索引，然後在來源資料變更時定期更新這些索引。一些常見的範例是網站、文件和電子商務搜尋。
- 動態索引：資料持續流入一組臨時索引，具有建立索引期和保留時段 (例如一組保留兩週的每日索引)。一些常見的範例是日誌分析、時間序列處理和點擊流分析。

對於長效索引的工作負載，您可以在磁碟上檢查來源資料並輕鬆判斷它會消耗多少儲存空間。如果資料來自多個來源，只要一起新增這些來源。

對於動態索引，您可以保留期乘上在代表時段產生的資料量。例如，若您每小時產生 200 MiB 的日誌資料，也就是每天 4.7 GiB，假設您的保留期間為兩週，則在任何指定時間的資料為 66 GiB。

您的來源資料的大小，不過是您的儲存需求的一個面向。您還必須有考量：

- 複本數量：每個複本都是主要碎片的完整複本，索引的存放大小會顯示主要碎片和複本碎片所採用的大小。在預設情況下，每個 OpenSearch 索引都有一個複本。建議您至少有一個以防資料遺失。複本也可提升搜尋效能，因此如果您的讀取工作負載繁重，您也許想要更多複本。使用 `PUT /my-index/_settings` 以更新索引的 `number_of_replicas` 設定。
- OpenSearch 索引負荷：索引在磁碟上的大小各不相同。來源資料加上索引的總大小通常是來源的 110%，索引最多可達來源資料的 10%。在您對資料編製索引之後，您可以使用 `_cat/indices?v` API 和 `pri.store.size` 值計算確切的負荷。`_cat/allocation?v` 也提供實用的摘要。
- 作業系統預留空間：在預設情況下，Linux 會保留 5% 的檔案系統供 `root` 使用者用於關鍵程序、系統復原，以及防止磁碟分段問題。
- OpenSearch Service 負荷：OpenSearch Service 會保留每個執行個體 20% 的儲存空間 (最多 20 GiB) 以用於區段合併、日誌和其他內部操作。

  由於此 20 GiB 上限，預留空間的總量可能依您網域內的執行個體數目而大不相同。例如，網域可能有三個 `m6g.xlarge.search` 執行個體，各有 500 GiB 的儲存空間，總計 1.46 TiB。在這種情況下，總預留空間僅 60 GiB。另一網域可能有 10 個 `m3.medium.search` 執行個體，各有 100 GiB 的儲存空間，總計 0.98 TiB。在這種情況下，總預留空間是 200 GiB，即使第一個網域大 50%。

  在以下公式中，我們對開銷套用「最壞情況」預估值。此預估值包含額外的可用空間，有助於將節點故障和可用區域中斷的影響降到最低。

總之，如果您在任何指定的時間有 66 GiB 的資料，並且想要一個複本，您的最低儲存需求為接近 66 * 2 * 1.1 / 0.95 / 0.8 = 191 GiB。您可以將此計算一般化，如下所示：

來源資料 * (1 + 複本的數量) * (1 + 索引負荷) / (1 – Linux 預留空間) / (1 – OpenSearch Service 負荷) = 最低儲存需求

或者，您可以使用此簡化版本：

來源資料 * (1 + 複本的數量) * 1.45 = 最低儲存需求

儲存空間不足是造成叢集不穩定的最常見原因之一。因此，您應在選擇執行個體類型、執行個體計數和儲存磁碟區時反復檢查數字。

也存在其他儲存考量事項：

- 如果您的最低儲存需求超過 1 PB，請參閱 the section called “PB 規模”。
- 如果您有輪流更新的索引，而且想要使用熱暖架構，請參閱the section called “UltraWarm 儲存”。

選擇碎片數

在您了解您的儲存需求之後，您可以調查您的索引策略。在預設情況下，在 OpenSearch Service 中，每個索引會分成五個主要碎片和一個複本 (共 10 個碎片)。這種行為不同於開放原始碼 OpenSearch，它預設為一個主要碎片和一個複本碎片。由於您無法輕易變更現有索引的主要碎片數，所以您應該在建立第一個文件的索引之前，決定好碎片計數。

選擇一些碎片的整體目標是要將索引平均分佈到叢集中的所有資料節點。不過，這些碎片不應該是太大或太多。一般指導方針是，針對搜尋延遲是關鍵效能目標的工作負載，嘗試將碎片大小保持在 10-30 GiB 之間，對於寫入操作繁重的工作負載 (例如日誌分析)，保持在 30-50 GiB。

大型碎片會造成 OpenSearch 難以從故障恢復，但因為每個碎片使用 CPU 和記憶體的一些量，擁有太多碎片可能導致效能問題和記憶體不足的錯誤。換言之，碎片應該夠小到基底 OpenSearch Service 執行個體能夠處理的程度，但又不會小到對硬體形成不必要的負擔。

例如，假設您有 66 GiB 的資料。您不希望該數量隨著時間增加，並且您想要保持每個碎片大約在 30 GiB 的大小。因此，您的碎片數應該大約為 66 * 1.1 / 30 = 3 個。您可以將此計算一般化，如下所示：

(來源資料 + 成長空間) * (1 + 索引負荷) / 所需碎片大小 = 主要碎片的大約數量

這個方程式有助於補償資料隨時間的成長。如果您預期同樣皆為 66 GiB 的資料到明年成為四倍，大約碎片數則會是 (66 + 198) * 1.1 / 30 = 10 個。不過請記住，您尚未擁有那些額外的 198 GiB 資料。請進

行檢查以確保這個對未來的準備不會產生不必要的微小碎片，大量消耗眼前的 CPU 和記憶體。在這種情況下，66 * 1.1 / 10 個碎片 = 每個碎片 7.26 GiB，這會消耗額外的資源且低於建議的大小範圍。您可以考慮中庸之道的六個碎片，讓您目前擁有 12 GiB 的碎片且爾後擁有 48 GiB 的碎片。接著，您可能希望再次以三個碎片開始並在碎片超過 50 GiB 時重新建立資料的索引。

有個較不常見的問題包含限制每個節點的碎片數量。如果您適當地調整碎片大小，通常要過很久一段時間才會用完磁碟空間，然後達到此限制。例如，一個 `m6g.large.search` 執行個體具有最大 512 GiB 的磁碟大小。如果您保持在低於 80% 的磁碟用量並將碎片大小調整為 20 GiB，即可容納大約 20 個碎片。Elasticsearch 7.x 和更新版本，以及截至 2.15 的所有 OpenSearch 版本，每個節點有 1，000 個碎片的限制。若要調整每個節點的最大碎片，請設定 `cluster.max_shards_per_node` 設定。對於 OpenSearch 2.17 及更新版本，OpenSearch Service 支援每 16GB JVM 堆積記憶體 1000 個碎片，每個節點最多 4000 個碎片。如需範例，請參閱叢集設定。如需碎片計數的詳細資訊，請參閱 the section called “碎片計數配額”。

只要適當地調整碎片大小，幾乎一律可保持低於此限制，但您也可以考慮 Java 堆積的每個 GiB 的碎片數量。在指定的節點上，Java 堆積的每 GiB 擁有不超過 25 個碎片。例如，一個 `m5.large.search` 執行個體具有 4 GiB 堆積，因此每個節點都應擁有不超過 100 個碎片。在該碎片計數，每個碎片的大小約為 5 GiB，這遠低於我們的建議。

選擇執行個體類型並進行測試

在您計算您的儲存需求並選擇所需的碎片數之後，您可以開始做出硬體的決策。硬體需求會依工作負載而大大不同，但是我們仍然可以提供一些基本建議。

一般而言，每個執行個體類型的儲存限制都會對應到您對於輕量型工作負載可能需要的 CPU 和記憶體量。例如，某個 `m6g.large.search` 執行個體具有最大 512 GiB 的 EBS 磁碟區大小、2 個 vCPU 核心和 8 GiB 的記憶體。如果您的叢集有許多碎片，執行課稅彙總、頻繁更新文件或處理大量的查詢，這些資源可能不足以滿足您的需求。如果您的叢集是這些類別之一，請嘗試針對您的每個 100 GiB 的儲存需求從組態較接近 2 個 vCPU 核心和 8 GiB 的記憶體開始。

> **Tip**
>
> 如需分配給每個執行個體類型的硬體資源的摘要，請參閱 Amazon OpenSearch Service 定價。

然而，即使這些資源可能會也不夠。有些 OpenSearch 使用者回報告他們很多次需要這些資源來滿足他們的需求。若要尋找適用於您的工作負載的合適硬體，您必須有明智的初始預估、測試代表的工作負載、進行調整並再次執行測試。

步驟 1：進行初步預估

若要開始，建議您使用至少三個節點，以避免潛在的 OpenSearch 問題，例如大腦分割狀態 (當通訊時間流逝導致具有兩個主節點的叢集時)。如果您有三個專用主節點，我們仍然建議您至少有兩個資料節點用於複寫。

步驟 2：計算每個節點的儲存需求

如果您有 184 GiB 的儲存需求和建議的至少三個節點，請使用方程式 184 / 3 = 61 GiB 算出每個節點所需的儲存量。在此範例中，您可以選取三個 `m6g.large.search` 執行個體，其中每個執行個體都分別使用 90 GiB 的 EBS 儲存磁碟區，讓您對於隨時間的成長擁有安全網路以及一些空間可供因應。這種組態提供 6 個 vCPU 核心和 24 GiB 的記憶體，所以適合較輕量型的工作負載。

針對更高額度的範例，則假設是儲存需求為 14 TiB (14,336 GiB) 的繁重工作負載。在這種情況下，您可以選擇開始測試 2 * 144 = 288 個 vCPU 核心和 8 * 144 = 1152 GiB 的記憶體。這些數字運作大約 18 個 `i3.4xlarge.search` 執行個體。如果您不需要快速的本機儲存，也可以測試 18 個 `r6g.4xlarge.search` 執行個體，各使用 1 TiB 的 EBS 儲存磁碟區。

如果您的叢集包含數百 TB 的資料，請參閱 the section called “PB 規模”。

步驟 3：執行代表性測試

設定叢集後，您可以使用您稍早計算的碎片數來新增您的索引、使用真實資料集來執行一些代表性的用戶端測試，以及監控 CloudWatch 指標來查看叢集處理工作負載的方式。

步驟 4：成功或反覆

如果效能滿足您的需求，表示測試成功且 CloudWatch 指標正常，叢集就已就緒可供使用。請記得設定 CloudWatch 警示，以偵測狀況不良的資源使用。

如果無法接受該效能，表示測試失敗或者 `CPUUtilization` 或 `JVMMemoryPressure` 偏高，您可能需要選擇不同的執行個體類型 (或新增執行個體)，並繼續進行測試。在您新增執行個體時，OpenSearch 會自動重新均衡整個叢集中碎片的分佈。

由於這在動力過強的叢集中測量過剩容量比在動力不足的叢集中測量欠缺容量來得容易，我們建議您從比您所需較大的叢集開始。接著，進行測試並縮減到有額外資源可確保在增加活動期間有穩定操作的有效率叢集。

生產叢集或狀態複雜的叢集受益於專用主節點，可提升效能和叢集可靠性。

Amazon OpenSearch Service 的 PB 規模

Amazon OpenSearch Service 網域提供高達 10 PB 的連接儲存空間。您可以使用 1000 個OR1.16xlarge.search執行個體類型來設定網域，每個執行個體類型都有 36 TB 的儲存空間。由於規模上的巨大差異，所以對此大小的網域的建議與我們的一般建議不同。本節討論建立網域、成本、儲存和碎片大小。

雖然本節經常參考i3.16xlarge.search執行個體類型，但您可以使用其他多種執行個體類型來達到總網域儲存體的 10 PB。

建立網域

此大小的網域超過每個網域 80 個執行個體的預設限制。若要請求增加每個網域最多 1000 個執行個體的服務限制，請在AWS 支援中心開立案例。

定價

建立此大小的網域之前，請檢查 Amazon OpenSearch Service 定價頁面，以確保相關的成本符合您的期望。檢查 the section called “UltraWarm 儲存”，確認熱暖架構是否適合您的使用案例。

儲存

此 i3 執行個體類型經過設計，可提供快速、本機非揮發性記憶體儲存裝置 (NVMe) 的儲存空間。由於這種本機儲存體相較於 Amazon Elastic Block Store EBS 磁碟區更能提供效能優勢，所以當您在 OpenSearch Service 中選取這些執行個體類型時，不會選擇 EBS 磁碟區。如果您偏好使用 EBS 儲存，請使用另一個執行個體類型 (例如 r6.12xlarge.search。

碎片大小和計數

一般的 OpenSearch 準則是每個碎片不超過 50 GB。鑒於大型網域所需的碎片數量，以及可用於 i3.16xlarge.search 執行個體的可用資源，我們建議碎片大小為 100 GB。

例如，如果您有 450 TB 的來源資料，並且想要一個複本，您的最低儲空間要求比較接近 450 TB * 2 * 1.1 / 0.95 = 1.04 PB。如需此計算詳細說明，請參閱the section called “計算儲存需求”。雖然有 1.04 PB / 15 TB = 70 個執行個體，但是您可以選擇 90 個或更多的 i3.16xlarge.search 執行個體，讓自己擁有儲存安全網並，處理節點故障並將隨時間增加資料量的一些變異納入考量。每個執行個體會新增另一組 20 GiB 到您的最低儲存需求，但對於此大小的磁碟，這組 20 GiB 幾乎可以忽略不計。

控制碎片數量不是那麼容易。OpenSearch 使用者經常會每天輪換索引，並保留資料一週或兩週。在這種情況下，您可能會發現區分「作用中」和「非作用中」的碎片數量很管用。作用中碎片會被

主動寫入或讀取。非作用中碎片可能服務一些讀取請求，但大部分都在閒置狀態。一般而言，您應該保留有效碎片數量在數千以下。隨著非作用中碎片數量達到 10,000 個，可觀的效能和穩定性風險也隨之出現。

若要計算主要碎片的數量，請使用下列公式：每個碎片 450,000 GB * 1.1 / 100 GB = 4,950 個碎片。複本數量的兩倍是 9,900 分片，其表示當所有碎片都在作用中時的主要考量。但是，如果您輪換索引，而且只有 1/7 或 1/14 的作用中碎片數量或碎片在任何指定一天 (1,414 或 707 碎片)，叢集可能有良好的運作狀態。如往常一樣，調整大小和設定您的網域的最重要步驟是使用實際的資料集執行代表性的用戶端測試。

Amazon OpenSearch Service 中的專用協調器節點

Amazon OpenSearch Service 中的專用協調器節點是特殊節點，可從資料節點卸載協調任務。這些任務包括管理搜尋請求和託管 OpenSearch Dashboards。透過分隔這些函數，專用協調器節點可減少資料節點的負載，這可讓它們專注於資料儲存、索引和搜尋操作。這可改善整體叢集效能和資源使用率。

此外，專用協調器節點有助於減少 VPC 組態所需的私有 IP 地址數量，進而實現更有效率的網路管理。此設定最多可改善 15% 的索引輸送量和 20% 更佳的查詢效能，視工作負載特性而定。

何時使用專用協調器節點

專用協調器節點在下列案例中最有用。

- 大型叢集 – 在具有大量資料或複雜查詢的環境中，將協調任務卸載至專用節點可以改善叢集效能。
- 頻繁查詢 – 涉及頻繁搜尋查詢或彙總的工作負載，特別是具有複雜日期長條圖或多個彙總的工作負載，受益於更快速的查詢處理。
- 重度儀表板使用 – OpenSearch Dashboards 可能會耗費大量資源。將此責任卸載至專用協調器節點可減少資料節點上的負擔。

架構和行為

在 OpenSearch 叢集中，專用協調器節點會處理兩個關鍵責任。

- 請求處理 – 這些節點會收到傳入的搜尋請求，並將其轉送至存放相關資料的適當資料節點。然後，它們會將來自多個資料節點的結果合併為單一全域結果集，並傳回給用戶端。
- 儀表板託管 – 協調器節點管理 OpenSearch Dashboards，可減輕資料節點託管 OpenSearch Dashboards 和處理相關流量的額外負擔。

在 VPC 網域中，將指派專用協調器節點彈性網路界面 (ENIs)，而非資料節點。此安排有助於減少 VPCs 所需的私有 IP 地址數量，進而改善網路效率。一般而言，專用協調器節點約佔總資料節點的 10%。

要求與限制

專用協調器節點有下列需求和限制。

- 所有 OpenSearch 版本和 Elasticsearch 6.8 至 7.10 版都支援專用協調器節點。
- 若要啟用專用協調器節點，您的網域必須啟用專用主節點。如需詳細資訊，請參閱the section called “專用主節點”。
- 佈建專用協調器節點可能會產生額外費用。不過，改善的資源效率和增強的效能可證明投資的合理性，尤其是大型或複雜的叢集。

佈建專用協調器節點

執行下列步驟，在現有網域中佈建專用協調器節點。在佈建協調器節點之前，請確定您的網域已啟用專用主節點。

主控台

在 中佈建專用協調器節點 AWS 管理主控台

1. 登入 Amazon OpenSearch Service 主控台，網址為 https：//https://console.aws.amazon.com/aos/home。
2. 選擇網域，然後選取您要修改的網域。
3. 在叢集組態區段中，選擇編輯。
4. 選擇啟用專用協調器節點。
5. 選取要佈建的執行個體類型和協調器節點數量。
6. 選擇儲存變更。網域更新可能需要幾分鐘的時間。

AWS CLI

若要使用 佈建專用協調器節點 AWS CLI，請使用 update-domain-config 命令。下列範例會在網域中佈建三個`r6g.large.search`協調器節點。

```
aws opensearch update-domain-config \
```

```
  --domain-name my-opensearch-domain \
  --cluster-config
InstanceCount=3,InstanceType=r6g.large.search,DedicatedCoordinatorCount=3,ZoneAwarenessEnabled
```

此命令可啟用專用協調器節點、設定協調器節點的執行個體類型和計數，以及啟用區域意識以提高可用性。

最佳實務

當您使用專用協調器節點時，請考慮下列最佳實務。

- 針對大多數使用案例使用一般用途執行個體。它們提供成本和效能之間的平衡方法。記憶體最佳化執行個體非常適合需要大量記憶體資源的工作負載，例如涉及複雜彙總或大規模搜尋的工作負載。
- 一個很好的起點是將 5% 到 10% 的資料節點佈建為專用協調器節點。例如，如果您的網域有 90 個特定執行個體類型的資料節點，請考慮佈建 5 到 9 個相同執行個體類型的協調器節點。

  > **Note**
  >
  > 執行個體類型可用性因區域而異。選取協調器節點的執行個體類型時，請確認您選擇的執行個體類型在目標區域中可用。您可以在建立或修改網域時，在 OpenSearch Service 主控台中檢查執行個體類型可用性。

- 若要將單一故障點的風險降至最低，請佈建至少兩個專用協調器節點。這可確保即使一個節點失敗，叢集仍能正常運作。
- 如果您使用跨區域搜尋，請在目的地網域中佈建專用協調器節點。來源網域通常不會處理協調任務。
- 對於索引繁重的環境，請考慮符合資料節點執行個體大小的 CPU 最佳化執行個體，以獲得最佳效能。
- 對於記憶體密集型工作負載，請針對專用協調器節點使用稍大的執行個體類型，以協助管理增加的記憶體需求。
- 追蹤 `CoordinatorCPUUtilization` Amazon CloudWatch 指標。如果持續超過 80%，可能表示您需要較大或額外的協調器節點來處理負載。
- 調整專用協調器節點的大小以符合您的資料節點。例如，使用 4xlarge 資料節點時，從 4xlarge 一般用途協調器節點開始。
- 對協調器節點使用多個較小的執行個體，而不是較少較大的執行個體，除非您的個別請求或回應需要極高的記憶體（以 GBs 為單位）。例如，選擇 12 個 4xl 執行個體，而不是 6 個 8xlarge 一般用途執行個體。

依叢集大小的節點建議

使用以下準則作為根據您的叢集大小佈建專用協調器節點的起點。根據工作負載特性和效能指標調整節點的數量和類型。

叢集大小	建議的協調器節點	執行個體類型
小型 （最多 50 個節點）	3-5 個節點	一般用途
中型 (50-100 個節點）	5-9 個節點	記憶體最佳化
大型 (100 個以上的節點）	10-15 個節點	記憶體最佳化

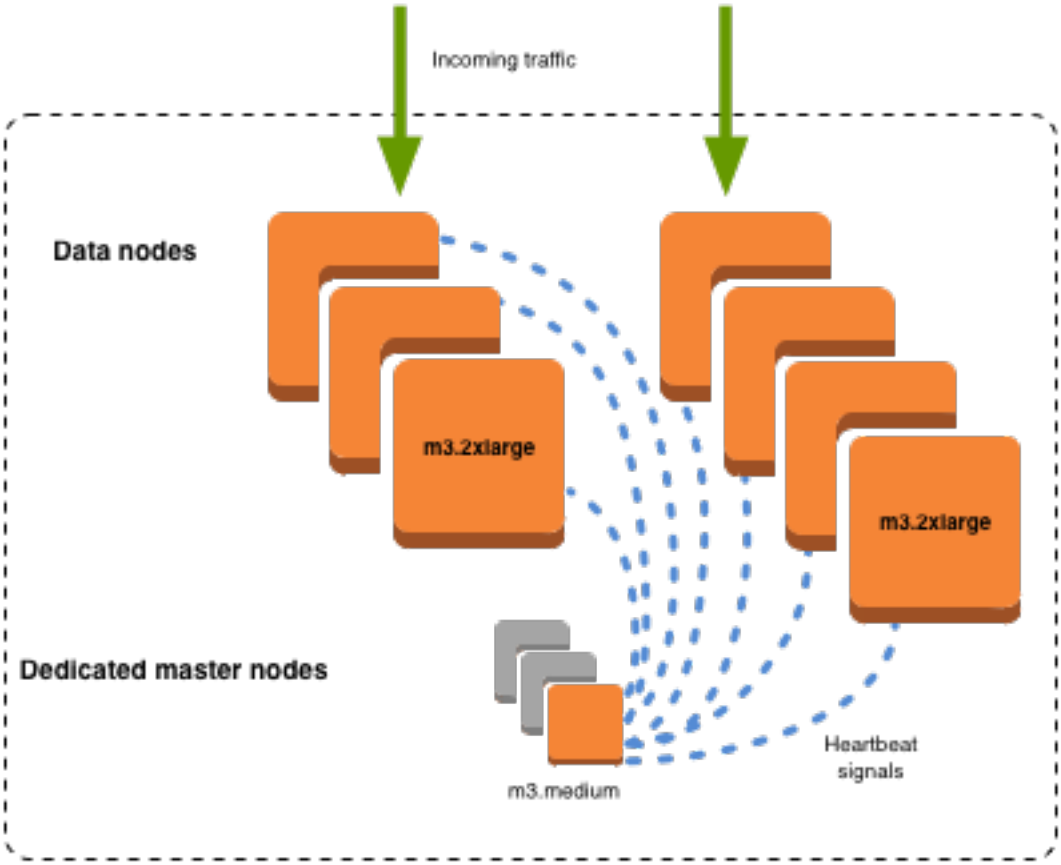
Amazon OpenSearch Service 中的專用主節點

Amazon OpenSearch Service 使用專用主節點來增加叢集的穩定性。專用主節點會執行叢集管理任務，但不會保留資料或回應資料上傳請求。此叢集管理任務的卸載可增加您網域的穩定性。就像所有其他節點類型一樣，您需要按照小時費率為每個專用主節點支付費用。

專用主節點會執行以下叢集管理任務：

- 追蹤叢集中的所有節點。
- 追蹤叢集中的索引數量。
- 追蹤屬於每個索引的碎片數。
- 維持叢集中節點的路由資訊。
- 狀態變更後更新叢集狀態，例如建立索引，以及在叢集中新增或移除節點。
- 跨叢集中的所有節點複製叢集狀態的變更。
- 透過傳送監控叢集中資料節點的可用性的活動訊號、定期信號，來監控所有叢集節點的運作狀態。

下圖顯示具有 10 個執行個體的 OpenSearch Service 網域。其中七個執行個體是資料節點，三個是專用主節點。只有其中一個專用主節點處於作用中狀態。這兩個灰色的專用主節點會等待作為備份，以防作用中的專用主節點發生故障。所有資料上傳請求是由七個資料節點提供服務，而所有叢集管理任務均卸載到作用中的專用主節點。



選擇專用主節點數目

我們建議您將多可用區與待命搭配使用，這會為每個生產 OpenSearch Service 網域新增三個專用主節點。如果您使用沒有待命或單一可用區的異地同步備份進行部署，我們仍建議三個專用主節點。切勿選擇偶數數量的專用主節點。選擇專用主節點的數目時，請考量下列事項：

- OpenSearch Service 明確禁止一個專用主節點，因為萬一故障時您沒有備份。如果您嘗試建立只有一個專用主節點的網域，則會收到驗證例外狀況。
- 如果您具有兩個專用主節點，表示您的叢集沒有節點的必要仲裁，供萬一發生故障時可選擇新的主節點。

  仲裁是專用主節點的數量 / 2 + 1 (四捨五入為最接近的整數)。在此案例中，為 2 / 2 + 1 = 2。因為一個專用主節點已故障且只存在一個備份，所以叢集沒有仲裁並且無法選擇新的主節點。
- 三個專用主節點 (建議的數量) 在萬一主節點故障時提供兩個備份節點，以及必要的仲裁 (2) 以選擇新的主節點。
- 四個專用主節點並不會比三個好，而且如果您使用多個可用區域，則可能導致問題。

- 如果一個主節點發生故障，您有仲裁 (3) 以選擇新的主節點。如果兩個節點故障，您會遺失該仲裁，就如同您使用三個專用主節點。
- 在三個可用區域組態中，兩個可用區域具有一個專用主節點，而一個可用區域具有兩個主節點。如果該可用區發生中斷，剩餘兩個不具備必要的仲裁 (3) 以選擇新的主節點。

- 擁有五個專用主節點的運作與三個的相同，並可讓您在維持仲裁的同時遺失兩個節點。但是因為在任何指定的時間只有一個專用主節點作用中，此組態表示支付四個閒置的節點。許多使用者發現這個層級的容錯移轉保護過於極端。

如果叢集具有偶數個合格的主節點，則 OpenSearch 和 Elasticsearch versions 7.x 及更新版本會忽略一個節點，使投票組態一律為奇數。在這種情況下，四個專用主節點基本上等同於三個專用主節點 (兩個則相當於一個)。

> **Note**
>
> 如果您的叢集不具備必要的仲裁以選擇新的主節點，將請求寫入到叢集「以及」讀取叢集請求則均會失敗。這種行為不同於 OpenSearch 預設。

選擇專用主節點的執行個體類型

OpenSearch Service 網域和執行個體配額

雖然專用主節點不會處理搜尋和查詢請求，但其大小與他們可以管理的執行個體大小和執行個體數量、索引和碎片高度相關。對於生產叢集，我們建議至少針對專用主節點使用下列執行個體類型。

這些建議是根據一般工作負載，並可能依您的需求而異。具有許多碎片或欄位映射的叢集可受益於更大的執行個體類型。如需詳細資訊，請參閱 Amazon OpenSearch Service 的建議 CloudWatch 警示，以判斷您是否需要使用較大的執行個體類型。

RAM	Elasticsearch 和 OpenSearch Service 1.x 到 2.15 的最大節點支援	Elasticsearch 和 OpenSearch Service 1.x 到 2.15 的最大碎片支援	OpenSearch Service 2.17 及更高版本的節點支援上限	OpenSearch Service 2.17 及更新版本的最大碎片支援
2 GB	不適用	不適用	10	1K

RAM	Elasticsearch 和 OpenSearch Service 1.x 到 2.15 的最大節點支援	Elasticsearch 和 OpenSearch Service 1.x 到 2.15 的最大碎片支援	OpenSearch Service 2.17 及更高版本的節點支援上限	OpenSearch Service 2.17 及更新版本的最大碎片支援
4 GB	不適用	不適用	10	5K
8 GB	10	10K	30	15K
16 GB	30	30K	60	30K
32 GB	75	40K	120	60K
64 GB	125	75K	240	120K
128 GB	200	75K	480	240K
256 GB	不適用	不適用	1002	500K

Amazon OpenSearch Service 的一般參考

Amazon OpenSearch Service 支援許多執行個體、操作、外掛程式和其他資源。

主題

- Amazon OpenSearch Service 中支援的執行個體類型
- 依 Amazon OpenSearch Service 中引擎版本區分的功能
- 依 Amazon OpenSearch Service 中引擎版本區分的外掛程式
- Amazon OpenSearch Service 中支援的操作
- Amazon OpenSearch Service 配額
- Amazon OpenSearch Service 中的預留執行個體
- Amazon OpenSearch Service 的成本最佳化技術
- Amazon OpenSearch Service 中的其他支援資源

Amazon OpenSearch Service 中支援的執行個體類型

Amazon OpenSearch Service 支援以下執行個體類型。並非所有區域皆支援所有執行個體類型。如需可用性詳細資訊，請參閱 Amazon OpenSearch Service 定價。

如需哪些執行個體適合用於您的使用案例之相關資訊，請參閱 the section called “調整網域大小”、the section called “EBS 磁碟區大小配額” 與 the section called “網路配額”。

請注意，下列執行個體系列不支援 VPC 加密控制：T2, T3, R3, I3, C4, M4, R4, C5, M5, R5, C6g, M6g, R6g和 R6gd。VPC 加密控制是一項安全與合規功能，可讓您集中進行授權控制，以監控流量的加密狀態、協助您識別允許純文字通訊的資源，並最終為您提供機制，以強制執行區域內跨 VPCs傳輸中的加密。若要進一步了解詳細資訊，請參閱 VPC 加密控制文件。

最新一代執行個體類型

為了獲得最佳效能，我們建議您在建立新 OpenSearch Service 網域時使用下列執行個體類型。

執行個體類型	執行個體	限制
i4i	i4i.large .search	i4i 執行個體類型需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch，且不支援 EBS 磁碟區儲存。

執行個體類型	執行個體	限制
	i4i.xlarge.search i4i.2xlarge.search i4i.4xlarge.search i4i.8xlarge.search i4i.12xlarge.search i4i.16xlarge.search i4i.24xlarge.search i4i.32xlarge.search	

執行個體類型	執行個體	限制
i7i	i7i.large.search i7i.xlarge.search i7i.2xlarge.search i7i.4xlarge.search i7i.8xlarge.search i7i.12xlarge.search i7i.16xlarge.search	i7i 執行個體類型需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch，且不支援 EBS 磁碟區儲存。

執行個體類型	執行個體	限制
i8g	i8g.large.search i8g.xlarge.search i8g.2xlarge.search i8g.4xlarge.search i8g.8xlarge.search i8g.12xlarge.search i8g.16xlarge.search	i8g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且不支援 EBS 儲存磁碟區。

執行個體類型	執行個體	限制
Graviton3	c7g.large.search c7g.xlarge.search c7g.2xlarge.search c7g.4xlarge.search c7g.8xlarge.search c7g.12xlarge.search c7g.16xlarge.search m7g.medium.search m7g.large.search m7g.xlarge.search m7g.2xlarge.search m7g.4xlarge.search	Graviton3 僅支援 gp3 儲存。Graviton3 執行個體 (c7g、m7g、r7g) 支援所有 OpenSearch 版本和 Elasticsearch 7.9 和 7.10 版。

執行個體類型	執行個體	限制
	m7g.8xlarge.search m7g.12xlarge.search m7g.16xlarge.search r7g.medium.search r7g.large.search r7g.xlarge.search r7g.2xlarge.search r7g.4xlarge.search r7g.8xlarge.search r7g.12xlarge.search r7g.16xlarge.search r7gd.large.search r7gd.xlarge.search	

執行個體類型	執行個體	限制
	r7gd.2xlarge.search r7gd.4xlarge.search r7gd.8xlarge.search r7gd.12xlarge.search r7gd.16xlarge.search	

執行個體類型	執行個體	限制
OR1	or1.medium.search or1.large.search or1.xlarge.search or1.2xlarge.search or1.4xlarge.search or1.8xlarge.search or1.12xlarge.search or1.16xlarge.search	• OR1 執行個體類型需要 OpenSearch 2.11 或更新版本。 • OR1 執行個體僅與其他 Graviton 執行個體類型主節點 (C6g、M6g, R6g) 相容。

執行個體類型	執行個體	限制
OR2	or2.medium.search or2.large.search or2.xlarge.search or2.2xlarge.search or2.4xlarge.search or2.8xlarge.search or2.12xlarge.search or2.16xlarge.search	• OR2 和 OM2 執行個體類型需要 OpenSearch 2.11 或更新版本，且僅與 Graviton 型主節點執行個體類型 (C6g、M6g, R6g) 相容。

執行個體類型	執行個體	限制
OM2	om2.large.search om2.xlarge.search om2.2xlarge.search om2.4xlarge.search om2.8xlarge.search om2.12xlarge.search om2.16xlarge.search	• OR2 和 OM2 執行個體類型需要 OpenSearch 2.11 或更新版本，且僅與 Graviton 型主節點執行個體類型 (C6g、M6g, R6g) 相容。

執行個體類型	執行個體	限制
OI2	oi2.large.search oi2.xlarge.search oi2.2xlarge.search oi2.4xlarge.search oi2.8xlarge.search oi2.12xlarge.search oi2.16xlarge.search oi2.24xlarge.search oi2.32xlarge.search	• OI2 執行個體類型需要 OpenSearch 2.11 或更新版本。 • 這些執行個體是以 NVMe 儲存體佈建，不需要個別的 EBS 儲存體。 • 這些執行個體（最多 8xlarge) 也可以做為 OpenSearch 3.3 或更新版本上的暖節點。

執行個體類型	執行個體	限制

Im4gn	`im4gn.large.search`
	
	`im4gn.xlarge.search`
	
	`im4gn.2xlarge.search`
	
	`im4gn.4xlarge.search`
	
	`im4gn.8xlarge.search`
	
	`im4gn.16xlarge.search`
		• Im4gn 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且不支援 EBS 儲存磁碟區。
		• Im4gn 執行個體僅與其他 Graviton 執行個體類型 (C6g、M6g, R6g, R6gd) 相容。您無法在同一個叢集中組合 Graviton 和非 Graviton 執行個體。

T3	`t3.small.search`
	
	`t3.medium.search`
		• T3 執行個體類型需要 Elasticsearch 5.6 或更新版本或任何版本的 OpenSearch。
		• 只有在您的網域佈建為無待命時，才能使用 T3 執行個體類型。如需詳細資訊，請參閱the section called “無待命的異地同步備份”。
		• 只有在您網域的執行個體計數為 10 或更少時，才能使用 T3 執行個體類型。
		• T3 執行個體類型不支援 UltraWarm 儲存、冷儲存或自動調整。

執行個體類型	執行個體	限制
c7i	c7i.large.search c7i.xlarge.search c7i.2xlarge.search c7i.4xlarge.search c7i.8xlarge.search c7i.12xlarge.search c7i.16xlarge.search	• c7i 執行個體需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch，且僅支援 GP3 儲存磁碟區。

執行個體類型	執行個體	限制
m7i	m7i.large.search m7i.xlarge.search m7i.2xlarge.search m7i.4xlarge.search m7i.8xlarge.search m7i.12xlarge.search m7i.16xlarge.search	• m7i 執行個體需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch，且僅支援 GP3 儲存磁碟區。

執行個體類型	執行個體	限制
r7i	r7i.large.search r7i.xlarge.search r7i.2xlarge.search r7i.4xlarge.search r7i.8xlarge.search r7i.12xlarge.search r7i.16xlarge.search	• r7i 執行個體需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch，且僅支援 GP3 儲存磁碟區。

執行個體類型	執行個體	限制
c8g	c8g.large.search c8g.xlarge.search c8g.2xlarge.search c8g.4xlarge.search c8g.8xlarge.search c8g.12xlarge.search c8g.16xlarge.search	• c8g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且僅支援 GP3 儲存磁碟區。

執行個體類型	執行個體	限制
m8g	m8g.medium.search m8g.large.search m8g.xlarge.search m8g.2xlarge.search m8g.4xlarge.search m8g.8xlarge.search m8g.12xlarge.search m8g.16xlarge.search	• m8g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且僅支援 GP3 儲存磁碟區。

執行個體類型	執行個體	限制
r8g	r8g.medium.search r8g.large.search r8g.xlarge.search r8g.2xlarge.search r8g.4xlarge.search r8g.8xlarge.search r8g.12xlarge.search r8g.16xlarge.search	• r8g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且僅支援 GP3 儲存磁碟區。

執行個體類型	執行個體	限制
r8gd	r8gd.medium.search r8gd.large.search r8gd.2xlarge.search r8gd.4xlarge.search r8gd.8xlarge.search r8gd.12xlarge.search r8gd.16xlarge.search	• r8gd 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且不支援 EBS 儲存磁碟區。

上一代執行個體類型

OpenSearch Service 提供上一代的執行個體類型，適合已據此將應用程式最佳化且尚未升級的使用者。建議您使用最新一代的執行個體類型以獲得最佳效能，但我們仍會繼續支援下列上一代執行個體類型。

執行個體類型	執行個體	限制
C4	c4.large.search c4.xlarge.search c4.2xlarge.search c4.4xlarge.search c4.8xlarge.search	
I2	i2.xlarge.search i2.2xlarge.search	
M3	m3.medium.search m3.large.search m3.xlarge.search m3.2xlarge.search	• M3 執行個體類型不支援靜態資料加密、精細存取控制或跨叢集搜尋。 • M3 執行個體類型在 OpenSearch 版本有額外的限制。如需詳細資訊，請參閱 the section called “無效的 M3 執行個體類型”。
M4	m4.large.search	

執行個體類型	執行個體	限制
	m4.xlarge.search m4.2xlarge.search m4.4xlarge.search m4.10xlarge.search	
R3	r3.large.search r3.xlarge.search r3.2xlarge.search r3.4xlarge.search r3.8xlarge.search	R3 執行個體類型不支援靜態資料加密或精細存取控制。

執行個體類型	執行個體	限制
R4	r4.large.search r4.xlarge.search r4.2xlarge.search r4.4xlarge.search r4.8xlarge.search r4.16xlarge.search	
i4g	i4g.large.search i4g.xlarge.search i4g.2xlarge.search i4g.4xlarge.search i4g.8xlarge.search i4g.16xlarge.search	i4g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且不支援 EBS 儲存磁碟區。

執行個體類型	執行個體	限制
C5	c5.large.search c5.xlarge.search c5.2xlarge.search c5.4xlarge.search c5.9xlarge.search c5.18xlarge.search	C5 執行個體類型需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch。

執行個體類型	執行個體	限制
C6g	c6g.large.search c6g.xlarge.search c6g.2xlarge.search c6g.4xlarge.search c6g.8xlarge.search c6g.12xlarge.search	• C6g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch。 • C6g 執行個體僅與其他 Graviton 執行個體類型 (Im4gn、M6g, R6g, R6gd) 相容。您無法在同一個叢集中組合 Graviton 和非 Graviton 執行個體。

執行個體類型	執行個體	限制
I3	i3.large.search i3.xlarge.search i3.2xlarge.search i3.4xlarge.search i3.8xlarge.search i3.16xlarge.search	
M5	m5.large.search m5.xlarge.search m5.2xlarge.search m5.4xlarge.search m5.12xlarge.search	M5 執行個體類型需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch。

執行個體類型	執行個體	限制
M6g	m6g.large.search m6g.xlarge.search m6g.2xlarge.search m6g.4xlarge.search m6g.8xlarge.search m6g.12xlarge.search	• M6g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch。 • M6g 執行個體僅與其他 Graviton 執行個體類型 (Im4gn、C6g, R6g, R6gd) 相容。您無法在同一個叢集中組合 Graviton 和非 Graviton 執行個體。

執行個體類型	執行個體	限制
R5	r5.large.search r5.xlarge.search r5.2xlarge.search r5.4xlarge.search r5.12xlarge.search	R5 執行個體類型需要 Elasticsearch 5.1 或更新版本或任何版本的 OpenSearch。

執行個體類型	執行個體	限制
R6g	r6g.large.search r6g.xlarge.search r6g.2xlarge.search r6g.4xlarge.search r6g.8xlarge.search r6g.12xlarge.search	• R6g 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch。 • R6g 執行個體僅與其他 Graviton 執行個體類型 (Im4gn、C6g, M6g, R6gd) 相容。您無法在同一個叢集中組合 Graviton 和非 Graviton 執行個體。

執行個體類型	執行個體	限制
R6gd	r6gd.large.search r6gd.xlarge.search r6gd.2xlarge.search r6gd.4xlarge.search r6gd.8xlarge.search r6gd.12xlarge.search r6gd.16xlarge.search	• R6gd 執行個體類型需要 Elasticsearch 7.9 或更新版本或任何版本的 OpenSearch，且不支援 EBS 儲存磁碟區。 • R6gd 執行個體僅與其他 Graviton 執行個體類型 (Im4gn、C6g, M6g, R6g) 相容。您無法在同一個叢集中組合 Graviton 和非 Graviton 執行個體。
T2	t2.micro.search t2.small.search t2.medium.search	• 如果您網域的執行個體計數為 10 個或更少，您僅可使用 T2 執行個體類型。 • t2.micro.search 執行個體類型只支援 Elasticsearch 1.5 和 2.3。 • T2 執行個體類型不支援靜態資料加密、精細存取控制、UltraWarm 儲存、冷儲存、跨叢集搜尋或自動調整。

> **Tip**
>
> 對於專用主節點及資料節點，我們經常建議使用不同的執行個體類型。

依 Amazon OpenSearch Service 中引擎版本區分的功能

許多 OpenSearch Service 功能都有最低的 OpenSearch 版本要求或舊版 Elasticsearch OSS 版本要求。如果您符合某項功能的最低版本，但您的網域無法使用該功能，請更新網域的服務軟體。

功能	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
多層架構	3.3	不包含
衍生來源	3.1	不包含
Star-tree 索引	2.17	不包含
Amazon Q 支援	2.17	不包含
自訂外掛程式	2.15	不包含
並行區段搜尋	2.13	不包含
產生自然語言查詢	2.13	不包含
多模式語意搜尋	2.11	不包含
直接查詢資料來源	2.11	不包含
機器學習連接器	2.9	不包含

功能	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
語意搜尋	2.9	不包含
時間點搜尋	2.5	不包含
通知	2.3	不包含
ML Commons	1.3	不包含
跨叢集複寫	1.1	7.10
索引轉換	1.0	不包含
專用協調器節點	1.0	6.8
VPC 支援	1.0	1.0
網路的所有流量都需使用 HTTPS		
異地同步備份支援		
專用主節點		
自訂套件		
自訂端點		
慢速日誌發佈		
錯誤日誌發佈	1.0	5.1
靜態資料加密		

功能	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
OpenSearch Dashboards 的 Cognito 身分驗證		
就地升級		
每小時自動快照	1.0	5.3
節點對節點加密	1.0	6.0
Java 高階 REST 用戶端支援		
HTTP 請求和回應壓縮		
提醒	1.0	6.2
SQL	1.0	6.5
跨叢集搜尋	1.0	6.7
精細定義存取控制		
OpenSearch Dashboards 的 SAML 身分驗證。		
自動調整		

功能	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
遠端重新索引		
UltraWarm	1.0	6.8
索引狀態管理		
依歐幾里德距離分類的 k-NN	1.0	7.1
異常偵測	1.0	7.4
依餘弦相似度分類的 k-NN	1.0	7.7
Learning to Rank		
Piped Processing Language	1.0	7.9
OpenSearch Dashboards 報告		
OpenSearch Dashboards 追蹤分析		

功能	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
以 ARM 為基礎的 Graviton 執行個體		
冷儲存		
漢明距離、L1 Norm 距離和 K-NN 的 Painless 指令碼	1.0	7.10
非同步搜尋		
Curator 支援	不包含	5.1

如需啟用這些功能中一部分功能及其他功能的外掛程式資訊，請參閱 the section called “依引擎版本分類的外掛程式”。如需各版本 OpenSearch API 的資訊，請參閱the section called “受支援的操作”。

依 Amazon OpenSearch Service 中引擎版本區分的外掛程式

Amazon OpenSearch Service 網域預先封裝有來自 OpenSearch 社群的外掛程式。該服務會自動為您部署和管理外掛程式，但是會根據您為網域選擇的 OpenSearch 或舊版 Elasticsearch OSS 的版本而部署不同的外掛程式。

下表列出 OpenSearch 版本的外掛程式，以及舊版 Elasticsearch OSS 的相容版本。它只包含您可能與之互動的外掛程式，它並不完整。OpenSearch Service 會使用其他外掛程式來啟用核心服務功能，例如將 S3 Repository 外掛程式用於快照，將 OpenSearch Performance Analyzer 外掛程式用於最佳化和監控。如需在您的網域上執行的所有外掛程式的完整清單，請進行以下請求：

```
GET _cat/plugins?v
```

外掛程式	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
HanLP	2.11	不支援
希伯來文分析	2.11	不支援
Amazon Personalize 搜尋排名	2.9	不支援
神經搜尋	2.9	不支援
安全分析	2.5	不支援
OpenSearch 通知	2.3	不支援
ML Commons	1.3	不支援
Sudachi 分析（建議使用日文）	1.3	不支援
STConvert	1.3	不支援
拼音分析	1.3	不支援
Nori 分析	1.3	不支援
OpenSearch 可觀測性	1.2	不支援
OpenSearch 跨叢集複寫	1.1	7.10

外掛程式	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
OpenSearch 非同步搜尋	1.0	7.10
IK (中文) 分析	1.0	7.7
越南文分析		
泰文分析		
Learning to Rank		
OpenSearch 異常偵測	1.0	7.4
OpenSearch k-NN	1.0	7.1
OpenSearch 索引狀態管理	1.0	6.8
OpenSearch 安全性	1.0	6.7
OpenSearch SQL	1.0	6.5
OpenSearch 提醒	1.0	6.2
烏克蘭文分析	1.0	5.3
映射器大小	1.0	5.3

外掛程式	需要的 OpenSearch 最低版本	需要的 Elasticsearch 最低版本
映射器 Murmur3	1.0	5.1
擷取使用者代理程式處理器	1.0	5.1
擷取附件處理器	1.0	5.1
Stempel 波蘭文分析	1.0	5.1
智慧型中文分析	1.0	5.1
Seunjeon 韓文分析	1.0	5.1
語音分析	1.0	2.3
日文 (kuromoji) 分析	1.0	包含在所有網域中
ICU 分析	1.0	包含在所有網域中

選用的外掛程式

除了預先安裝的預設外掛程式之外，Amazon OpenSearch Service 還支援數個選用的語言分析器外掛程式。您可以使用 AWS 管理主控台 和 AWS CLI 將外掛程式與網域建立關聯、取消外掛程式與網域的關聯，以及列出所有外掛程式。選用的外掛程式套件與特定 OpenSearch 版本相容，而且只能與該版本的網域建立關聯。

請注意，對於 Sudachi 外掛程式，當您重新關聯字典檔案時，它不會立即反映在網域上。當下一個藍/綠部署在網域上執行時，字典會重新整理，作為組態變更或其他更新的一部分。或者，您可以使用更新

的資料建立新套件、使用此新套件建立新索引、將現有索引重新索引至新索引，然後刪除舊索引。如果您偏好使用重新索引方法，請使用索引別名，以免流量中斷。

選用外掛程式使用 ZIP-PLUGIN套件類型。如需選用外掛程式的詳細資訊，請參閱 the section called "套件"。

Amazon OpenSearch Service 中支援的操作

OpenSearch Service 支援多種版本的 OpenSearch 和舊版 Elasticsearch OSS。下列章節顯示 OpenSearch Service 針對每個版本支援的操作。

主題

- 值得注意的 API 差異

值得注意的 API 差異

新清單 APIs

為了支援具有大量索引和碎片的大型叢集，我們推出了具有分頁支援的新清單 APIs，例如 _list/indices 和 _list/shards。List API 會以分頁格式擷取索引和碎片的統計資料。這可簡化處理包含許多索引之回應的任務。

- `_list/indices`：_list/indices
- `_list/shards`：_list/shards

現有 APIs的變更

為了支援大型叢集，我們在 `_cluster/stats` API 中新增支援以新增其他指標篩選條件，以支援僅擷取相關的統計資料回應，例如 `_cluster/stats/<metric>/nodes/<node-filters>`和 `_cluster/stats/<metric>/<index_metric>/nodes/<node-filters>`。如需詳細資訊，請參閱 _cluster/stats。

我們已透過指定`cancel_after_time_interval`請求參數，在 `_cat/shards` API 中新增取消任務的支援。如需詳細資訊，請參閱 _cat/shards。

限制 _cat API 的回應大小

為了支援跨資料和暖節點執行個體總數超過 200 的大型叢集，我們對 傳回的索引數量有 10K 的限制_cat/segments API。如果回應中的索引數目超過此限制，API 會傳回 429 錯誤。若要避免這種情況，您可以在查詢中指定索引模式篩選條件，例如 _cat/segments/<index-pattern>。

設定和統計數字

OpenSearch Service 只接受針對 _cluster/settings API 的 PUT 請求，這些請求使用 "flat" 設定形式。並拒絕使用展開設定表單的請求。

```
// Accepted
PUT _cluster/settings
{
  "persistent" : {
    "action.auto_create_index" : false
  }
}

// Rejected
PUT _cluster/settings
{
  "persistent": {
    "action": {
      "auto_create_index": false
    }
  }
}
```

高階 Java REST 用戶端會使用展開的表單，因此如果您需要傳送設定請求，請使用低階用戶端。

在 Elasticsearch 5.3 之前，OpenSearch Service 網域上的 _cluster/settings API 僅支援 HTTP PUT 方法，而不支援 GET 方法。OpenSearch 和更新版本的 Elasticsearch 支援 GET 方法，如以下範例所示：

```
GET https://domain-name.region.es.amazonaws.com/_cluster/settings?pretty
```

此為傳回範例：

```
{
  "persistent": {
    "cluster": {
      "routing": {
        "allocation": {
```

```
          "cluster_concurrent_rebalance": "2",
          "node_concurrent_recoveries": "2",
          "disk": {
            "watermark": {
              "low": "1.35gb",
              "flood_stage": "0.45gb",
              "high": "0.9gb"
            }
          },
          "node_initial_primarirecoveries": "4"
        }
      }
    },
    "indices": {
      "recovery": {
        "max_bytper_sec": "40mb"
      }
    }
  }
}
```

如果您針對特定設定和統計數字 API 來比較開源 OpenSearch 叢集和 OpenSearch Service 的回應，您可能會注意到缺少一些欄位。OpenSearch Service 會編輯某些公開服務內部細節的特定資訊，例如 `_nodes/stats` 的檔案系統資料路徑或作業系統名稱以及 `_nodes` 的版本。

縮小

`_shrink` API 可造成升級、組態變更與網域刪除失敗。我們不建議在執行 Elasticsearch 版本 5.3 或 5.1 的網域上使用它。這些版本皆含有可導致已縮小的索引之快照還原失敗。

若您在其他 Elasticsearch 或 OpenSearch 版本上使用 `_shrink` API，請在開始縮小操作前提出下列請求：

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": "name-of-the-node-to-shrink-to",
    "index.blocks.read_only": true
  }
}
```

然後在完成縮小操作後再提出下列請求：

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}

PUT https://domain-name.region.es.amazonaws.com/shrunken-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}
```

新清單 APIs

為了支援具有大量索引和碎片的大型叢集，我們推出了具有分頁支援的新清單 APIs，即 _list/indices和 。 _list/shardsList API 會以分頁格式擷取索引和碎片的統計資料。這可簡化處理包含許多索引之回應的任務。如需 的詳細資訊_list/indices，請參閱列出索引。如需 的詳細資訊_list/shards，請參閱列出碎片。

現有 APIs的變更

為了支援大型叢集，我們在 _cluster/stats/<metric>/nodes/<node-filters>和 中新增了支援_cluster/stats/<metric>/<index_metric>/nodes/<node-filters>。如需 的詳細資訊_cluster/stats，請參閱叢集統計資料。

限制 _cat APIs回應大小

為了支援執行個體總數超過 200 個資料和暖節點的大型叢集，我們對 _cat/segments API 傳回的索引數量有 10，000 個限制。如果回應中的索引數量超過此限制，API 會傳回429錯誤。若要避免這種情況，您可以在查詢中指定索引模式篩選條件（例如 _cat/segments/<index-pattern>）。

此外，透過指定cancel_after_time_interval請求參數，任務取消的支援現在可用於任務取消的_cat/shards API。如需詳細資訊，請參閱 CAT 碎片。

選擇專用主節點的執行個體類型

下表提供為專用主節點選擇適當執行個體類型的建議：

RAM	支援的節點上限	支援的碎片上限
2 GB	10	1,000
4 GB	10	5,000
8 GB	30	15,000
16 GB	60	30,000
32 GB	120	60,000
64 GB	240	120,000
128 GB	480	240，000
256 GB	1002	500,000

OpenSearch 2.19 版

如需 OpenSearch 2.19 操作的相關資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。如需此版本變更的詳細資訊，請參閱 2.19 版本備註。

OpenSearch 2.17 版

對於 OpenSearch 2.17，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

> **Note**
>
> 從 OpenSearch 2.17 開始，無法修改`cluster.max_shards_per_node`設定。對於 OpenSearch 2.17 及更新版本，OpenSearch Service 支援每 16GB JVM 堆積記憶體 1000 個碎片，每個節點最多 4000 個碎片。

- 索引路徑中的所有操作 (例如 */index-name* /_forceme
- /_delete_by_query [1]
- /_explain
- /_refresh
- /_reindex [1]

rge 、/*index-name* /update/*id* 以及 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.search.request.slowlog.level
  - cluster.search.request.slowlog.threshold.warn
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_list
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - cluster.search.request.slowlog.threshold.info
  - cluster.search.request.slowlog.threshold.debug
  - cluster.search.request.slowlog.threshold.trace
  - search.phase_took_enabled
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 請參閱 PUT方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅參考 OpenSearch Service 支援的一般 OpenSearch 操作，不包含用於異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

> **Note**
>
> 目前，未為具有異地同步備份（可用區域）且待命的客戶啟用cluster.max_shards_per_node設定功能。

OpenSearch 2.15 版

對於 OpenSearch 2.15，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - cluster.max_shards_per_node
  - cluster.search.request.slowlog.level
  - cluster.search.request.slowlog.threshold.warn
  - cluster.search.request.slowlog.threshold.info
  - cluster.search.request.slowlog.threshold.debug
  - cluster.search.request.slowlog.threshold.trace
  - search.phase_took_enabled
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 2.13 版

對於 OpenSearch 2.13，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - cluster.max_shards_per_node
  - cluster.search.request.slowlog.level
  - cluster.search.request.slowlog.threshold.warn
  - cluster.search.request.slowlog.threshold.info
  - cluster.search.request.slowlog.threshold.debug
  - cluster.search.request.slowlog.threshold.trace
  - search.phase_took_enabled
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 2.11 版

對於 OpenSearch 2.11，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 2.9 版

對於 OpenSearch 2.9，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline

- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。

4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 2.7 版

對於 OpenSearch 2.7，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 2.5 版

對於 OpenSearch 2.5，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

update/*id* 以及 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 2.3 版

若為 OpenSearch 2.3，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template

- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- _plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 1.3 版

若為 OpenSearch 1.3，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 1.2 版

若為 OpenSearch 1.2，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores

- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。

4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 1.1 版

若為 OpenSearch 1.1，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_plugins/_security
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_sql
- /_plugins/_transforms
- /_percolate
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

OpenSearch 1.0 版

對於 OpenSearch 1.0，OpenSearch Service 支援下列操作。如需大部分操作的詳細資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

update/*id* 以及 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_plugins/_transforms
- /_percolate
- /_rank_eval
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 7.10 版

對於 Elasticsearch 7.10，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_index_template
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_opendistro/_asynchronous_search
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_plugins/_replication
- /_rank_eval
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。
6. 舊版索引範本 (_template) 被可組合的範本 (_index_template) 所取代，從 Elasticsearch 7.8 開始。可組合的範本優先於舊版範本。如果可組合的範本與指定索引不匹配，則舊版範本仍然匹配並套用。_template 操作仍然適用於 OpenSearch 和更新版本的 Elasticsearch OSS，但對兩個範本類型的 GET 呼叫會傳回不同結果。

Elasticsearch 7.9 版

對於 Elasticsearch 7.9，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 OpenSearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。
6. 舊版索引範本 (_template) 被可組合的範本 (_index_template) 所取代，從 Elasticsearch 7.8 開始。可組合的範本優先於舊版範本。如果可組合的範本與指定索引不匹配，則舊版範本仍然匹配並套用。_template 操作仍然適用於 OpenSearch 和更新版本的 Elasticsearch OSS，但對兩個範本類型的 GET 呼叫會傳回不同結果。

Elasticsearch 7.8 版

對於 Elasticsearch 7.8，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile

- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。

5. 請參閱 the section called “縮小”。
6. 舊版索引範本 (_template) 被可組合的範本 (_index_template) 所取代，從 Elasticsearch 7.8 開始。可組合的範本優先於舊版範本。如果可組合的範本與指定索引不匹配，則舊版範本仍然匹配並套用。_template 操作仍然適用於 OpenSearch 和更新版本的 Elasticsearch OSS，但對兩個範本類型的 GET 呼叫會傳回不同結果。

Elasticsearch 7.7 版

對於 Elasticsearch 7.7，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node

- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 7.4 版

對於 Elasticsearch 7.4，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 、*/index-name* /update/*id* 以及 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores

- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 7.1 版

對於 Elasticsearch 7.1，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- cluster.max_shards_per_node

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 6.8 版

對於 Elasticsearch 6.8，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status

- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
  - cluster.blocks.read_only
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 6.7 版

對於 Elasticsearch 6.7，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*) , */index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 6.5 版

對於 Elasticsearch 6.5，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_sql
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

- /_percolate
- /_plugin/kibana
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 6.4 版

對於 Elasticsearch 6.4，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases

- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps

- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]

- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 6.3 版

對於 Elasticsearch 6.3，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 6.2 版

對於 Elasticsearch 6.2，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

- /_plugin/kibana
- /_rank_eval

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 6.0 版

對於 Elasticsearch 6.0，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores

- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 5.6 版

對於 Elasticsearch 5.6，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 5.5 版

對於 Elasticsearch 5.5，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[4]：
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

- /_reindex [1]

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 有關使用指令碼的考量事項，請參閱the section called “其他受支援的資源”。
4. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
5. 請參閱 the section called “縮小”。

Elasticsearch 5.3 版

對於 Elasticsearch 5.3，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_render
- /_rollover
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[4]

- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性[3]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 參考 PUT 方法。如需 GET 方法的詳細資訊，請參閱the section called “值得注意的 API 差異”。此清單僅涉及 OpenSearch Service 支援的一般 Elasticsearch 操作，不包括針對異常偵測、ISM 等的外掛程式特定支援操作。
4. 請參閱 the section called “縮小”。

Elasticsearch 5.1 版

對於 Elasticsearch 5.1，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /update/*id*)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings 適用於數個屬性 (僅適用於 PUT)：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [3]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 叢集組態變更完成之前，這些操作可能中斷。我們建議您隨這些操作一起使用 /_tasks 作業，以確認請求已成功完成。
2. 對 /_search/scroll 的 DELETE 請求及訊息本文，都必須在 HTTP 標頭指定 "Content-Length"。根據預設，大多數的用戶端新增此標頭。為避免發生 scroll_id 值中有 = 字元的問題，請使用要求主體而不是查詢字串，將 scroll_id 值傳遞至 OpenSearch Service。
3. 請參閱 the section called “縮小”。

Elasticsearch 2.3 版

對於 Elasticsearch 2.3，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_forcemerge 和 */index-name* /_recovery)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cache/clear (僅適用於索引)
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/health
- /_cluster/settings 適用於數個屬性 (僅適用於 PUT)：
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - threadpool.get.queue_size
  - threadpool.bulk.queue_size
  - threadpool.index.queue_size
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_render
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

- threadpool.percolate.queue_size
- threadpool.search.queue_size
- threadpool.suggest.queue_size

Elasticsearch 1.5 版

對於 Elasticsearch 1.5，OpenSearch Service 支援下列操作。

- 索引路徑中的所有操作 (例如 */index-name* /_optimize 和 */index-name* /_warmer)，*/index-name* /_close 除外
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat
- /_cluster/health
- /_cluster/settings 適用於數個屬性 (僅適用於 PUT)：
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - threadpool.get.queue_size
  - threadpool.bulk.queue_size
  - threadpool.index.queue_size
  - threadpool.percolate.queue_size
  - threadpool.search.queue_size
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_plugin/kibana3
- /_plugin/migration
- /_refresh
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

- `threadpool.suggest.queue_size`

Amazon OpenSearch Service 配額

AWS 您的帳戶具有每個 AWS 服務的預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定的。

若要檢視 OpenSearch Service 網域和執行個體、Amazon OpenSearch Serverless 和 Amazon OpenSearch Ingestion 的配額，請參閱 中的 Amazon OpenSearch Service 配額AWS 一般參考。

若要在 中檢視 OpenSearch Service 的配額 AWS 管理主控台，請開啟 Service Quotas 主控台。在導覽窗格中，選擇 AWS 服務，然後選取 Amazon OpenSearch Service。若要請求提高配額，請參閱《Service Quotas 使用者指南》中的請求提高配額。

暖節點儲存配額

下表列出暖節點執行個體類型，以及每種類型可以使用的儲存容量上限。對於 OpenSearch Optimized OI2 執行個體，80% 的本機儲存體可做為快取使用，且最大可定址暖儲存體是快取儲存體大小的 5 倍。

例如，如果 OI2 執行個體具有 468 GB 的本機儲存體：

- 快取儲存 = 375 GB (80% 的本機儲存）
- 最大可定址暖儲存 = 1875 GB (5 x 375 GB 快取）

執行個體類型	執行個體儲存 (GB)	快取大小 (GB)	最大可定址暖儲存 (GB)
oi2.large.search	468	375	1875
oi2.xlarge.search	937	750	3750
oi2.2xlarge.search	1875	1500	7500
oi2.4xlarge.search	3750	3000	15000
oi2.8xlarge.search	7500	6000	30000

UltraWarm 儲存配額

下表列出 UltraWarm 執行個體類型，以及每個類型可使用的儲存量上限。如需 UltraWarm 的詳細資訊，請參閱the section called “UltraWarm 儲存”。

執行個體類型	最大儲存空間
ultrawarm1.medium.search	1.5 TiB
ultrawarm1.large.search	20 TiB

每個 AZ 的資料節點數量

下表列出 AZ 部署的資料節點總數低於 ，整體限制表示每個限制的資料節點數量，包括熱節點和暖節點計數。每種類型都可以使用的儲存體。

AZ 組態	熱節點計數限制	暖節點計數限制	整體限制 （熱 + 暖）
1 - AZ	334	250	334
2 - AZ	668	500	668
3 - AZ	1002	750	1002

執行個體系列的總節點限制

下表列出執行個體系列的總節點限制。

執行個體系列	ElasticSearch OpenSearch 至 2.15	OpenSearch 2.17 及更高版本	預設值限制
T2	10	10	10
T3	10	10	10

執行個體系列	ElasticSearch OpenSearch 至 2.15	OpenSearch 2.17 及更高版本	預設值限制
M3, C4, M4, R4, C5, M5, R5, I2, I3	10	200	80
Graviton 2、Graviton 3、Graviton 4	200	400	80
C7i, R7i, M7i、i4i、i4g、i8g、i7i	200	400	80
OR1.medium.search OR1.large.search OR2.medium.search OR2.large.search OM2.large.search OI2.large.search	200	400	80
OR1.xlarge.search 及更高版本 OR2.xlarge.search 及更高版本 OM2.xlarge.search 及更高版本 OI2.xlarge.search 及更高版本	200	1002	80
Ultrawarm1	150	750	150

EBS 磁碟區大小配額

下表顯示 OpenSearch Service 支援的每個執行個體類型的 EBS 磁碟區大小下限和上限。如需執行個體類型及包含執行個體儲存體和額外硬體詳細資訊的資訊，請參閱 Amazon OpenSearch Service 定價。

- 如果您在建立網域時在 EBS 磁碟區類型下選擇磁性儲存，除了 t2.small 和 t2.medium 以外的所有執行個體類型，以及不支援磁性儲存的所有 Graviton 執行個體 (M6g、C6g, R6g 和 R6gd) 的磁碟區大小上限為 100 GiB。對於下表中列出的最大大小，請選擇其中一個 SSD 選項。
- 有些更早產生的執行個體類型包括執行個體儲存體，也支援 EBS 儲存。如果您選擇其中一種 EBS 執行個體類型、磁碟區儲存空間 不會 累加。您可以使用 EBS 磁碟區或執行個體儲存體，但不能兩者共用。
- 對於 EBS gp3 磁碟區，支援的最大限制為 80，000 IOPS 和 2，000 MB/s 輸送量。請注意，視您的執行個體類型而定，網域的有效限制可能會較低。如需特定執行個體的最大限制，請參閱執行個體類型限制。

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
t2.micro.search	10 GiB	35 GiB	N/A
t2.small.search	10 GiB	35 GiB	N/A
t2.medium.search	10 GiB	35 GiB	N/A
t3.small.search	10 GiB	100 GiB	100 GiB
t3.medium.search	10 GiB	200 GiB	200 GiB
m3.medium.search	10 GiB	100 GiB	N/A
m3.large.search	10 GiB	512 GiB	N/A
m3.xlarge.search	10 GiB	512 GiB	N/A
m3.2xlarge.search	10 GiB	512 GiB	N/A
m4.large.search	10 GiB	512 GiB	N/A
m4.xlarge.search	10 GiB	1 TiB	N/A
m4.2xlarge.search	10 GiB	1.5 TiB	N/A
m4.4xlarge.search	10 GiB	1.5 TiB	N/A

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
m4.10xlarge.search	10 GiB	1.5 TiB	N/A
m5.large.search	10 GiB	512 GiB	1 TiB
m5.xlarge.search	10 GiB	1 TiB	2 TiB
m5.2xlarge.search	10 GiB	1.5 TiB	3 TiB
m5.4xlarge.search	10 GiB	3 TiB	6 TiB
m5.12xlarge.search	10 GiB	9 TiB	18 TiB
m6g.large.search	10 GiB	512 GiB	1 TiB
m6g.xlarge.search	10 GiB	1 TiB	2 TiB
m6g.2xlarge.search	10 GiB	1.5 TiB	3 TiB
m6g.4xlarge.search	10 GiB	3 TiB	6 TiB
m6g.8xlarge.search	10 GiB	6 TiB	12 TiB
m6g.12xlarge.search	10 GiB	9 TiB	18 TiB
c4.large.search	10 GiB	100 GiB	N/A
c4.xlarge.search	10 GiB	512 GiB	N/A
c4.2xlarge.search	10 GiB	1 TiB	N/A
c4.4xlarge.search	10 GiB	1.5 TiB	N/A
c4.8xlarge.search	10 GiB	1.5 TiB	N/A
c5.large.search	10 GiB	256 GiB	256 GiB
c5.xlarge.search	10 GiB	512 GiB	512 GiB
c5.2xlarge.search	10 GiB	1 TiB	1 TiB

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
c5.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c5.9xlarge.search	10 GiB	3.5 TiB	3.5 TiB
c5.18xlarge.search	10 GiB	7 TiB	7 TiB
c6g.large.search	10 GiB	256 GiB	256 GiB
c6g.xlarge.search	10 GiB	512 GiB	512 GiB
c6g.2xlarge.search	10 GiB	1 TiB	1 TiB
c6g.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c6g.8xlarge.search	10 GiB	3 TiB	3 TiB
c6g.12xlarge.search	10 GiB	4.5 TiB	4.5 TiB
r3.large.search	10 GiB	512 GiB	N/A
r3.xlarge.search	10 GiB	512 GiB	N/A
r3.2xlarge.search	10 GiB	512 GiB	N/A
r3.4xlarge.search	10 GiB	512 GiB	N/A
r3.8xlarge.search	10 GiB	512 GiB	N/A
r4.large.search	10 GiB	1 TiB	N/A
r4.xlarge.search	10 GiB	1.5 TiB	N/A
r4.2xlarge.search	10 GiB	1.5 TiB	N/A
r4.4xlarge.search	10 GiB	1.5 TiB	N/A
r4.8xlarge.search	10 GiB	1.5 TiB	N/A
r4.16xlarge.search	10 GiB	1.5 TiB	N/A

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
r5.large.search	10 GiB	1 TiB	2 TiB
r5.xlarge.search	10 GiB	1.5 TiB	3 TiB
r5.2xlarge.search	10 GiB	3 TiB	6 TiB
r5.4xlarge.search	10 GiB	6 TiB	12 TiB
r5.12xlarge.search	10 GiB	12 TiB	24 TiB
r6g.large.search	10 GiB	1 TiB	2 TiB
r6g.xlarge.search	10 GiB	1.5 TiB	3 TiB
r6g.2xlarge.search	10 GiB	3 TiB	6 TiB
r6g.4xlarge.search	10 GiB	6 TiB	12 TiB
r6g.8xlarge.search	10 GiB	8 TiB	16 TiB
r6g.12xlarge.search	10 GiB	12 TiB	24 TiB
r6gd.large.search	N/A	N/A	N/A
r6gd.xlarge.search	N/A	N/A	N/A
r6gd.2xlarge.search	N/A	N/A	N/A
r6gd.4xlarge.search	N/A	N/A	N/A
r6gd.8xlarge.search	N/A	N/A	N/A
r6gd.12xlarge.search	N/A	N/A	N/A
r6gd.16xlarge.search	N/A	N/A	N/A
i2.xlarge.search	10 GiB	512 GiB	N/A
i2.2xlarge.search	10 GiB	512 GiB	N/A

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
i3.large.search	N/A	N/A	N/A
i3.xlarge.search	N/A	N/A	N/A
i3.2xlarge.search	N/A	N/A	N/A
i3.4xlarge.search	N/A	N/A	N/A
i3.8xlarge.search	N/A	N/A	N/A
i3.16xlarge.search	N/A	N/A	N/A
or1.medium.search	20 GiB	N/A	768 GiB
or1.large.search	20 GiB	N/A	1532 GiB
or1.xlarge.search	20 GiB	N/A	3 TiB
or1.2xlarge.search	20 GiB	N/A	6 TiB
or1.4xlarge.search	20 GiB	N/A	12 TiB
or1.8xlarge.search	20 GiB	N/A	16 TiB
or1.12xlarge.search	20 GiB	N/A	24 TiB
or1.16xlarge.search	20 GiB	N/A	36 TiB
or2.medium.search	20 GiB	N/A	768 GiB
or2.large.search	20 GiB	N/A	1532 GiB
or2.xlarge.search	20 GiB	N/A	3 TiB
or2.2xlarge.search	20 GiB	N/A	6 TiB
or2.4xlarge.search	20 GiB	N/A	12 TiB
or2.8xlarge.search	20 GiB	N/A	16 TiB

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
or2.12xlarge.search	20 GiB	N/A	24 TiB
or2.16xlarge.search	20 GiB	N/A	36 TiB
om2.large.search	20 GiB	N/A	768 GiB
om2.xlarge.search	20 GiB	N/A	2 Tib
om2.2xlarge.search	20 GiB	N/A	3 Tib
om2.4xlarge.search	20 GiB	N/A	6 Tib
om2.8xlarge.search	20 GiB	N/A	12 Tib
om2.12xlarge.search	20 GiB	N/A	18 Tib
om2.16xlarge.search	20 GiB	N/A	24 Tib
im4gn.large.search	N/A	N/A	N/A
im4gn.xlarge.search	N/A	N/A	N/A
im4gn.2xlarge.search	N/A	N/A	N/A
im4gn.4xlarge.search	N/A	N/A	N/A
im4gn.8xlarge.search	N/A	N/A	N/A
im4gn.16xlarge.search	N/A	N/A	N/A
C7g.large.search	10 GiB	N/A	256 GiB
C7g.xlarge.search	10 GiB	N/A	512 GiB
C7g.2xlarge.search	10 GiB	N/A	1 TiB
C7g.4xlarge.search	10 GiB	N/A	1.5 TiB
C7g.8xlarge.search	10 GiB	N/A	3 TiB

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
C7g.12xlarge.search	10 GiB	N/A	4.5 TiB
C7g.16xlarge.search	10 GiB	N/A	6 TiB
M7g.medium.search	10 GiB	N/A	512 GiB
M7g.large.search	10 GiB	N/A	768 GiB
M7g.xlarge.search	10 GiB	N/A	2 TiB
M7g.2xlarge.search	10 GiB	N/A	3 TiB
M7g.4xlarge.search	10 GiB	N/A	6 TiB
M7g.8xlarge.search	10 GiB	N/A	12 TiB
M7g.12xlarge.search	10 GiB	N/A	18 TiB
M7g.16xlarge.search	10 GiB	N/A	24 TiB
R7g.medium.search	10 GiB	N/A	768 GiB
R7g.large.search	10 GiB	N/A	1.5 TiB
R7g.xlarge.search	10 GiB	N/A	3 TiB
R7g.2xlarge.search	10 GiB	N/A	6 TiB
R7g.4xlarge.search	10 GiB	N/A	12 TiB
R7g.8xlarge.search	10 GiB	N/A	16 TiB
R7g.12xlarge.search	10 GiB	N/A	24 TiB
R7g.16xlarge.search	10 GiB	N/A	36 TiB
R7gd.large.search	N/A	N/A	N/A
R7gd.xlarge.search	N/A	N/A	N/A

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
R7gd.2xlarge.search	N/A	N/A	N/A
R7gd.4xlarge.search	N/A	N/A	N/A
R7gd.8xlarge.search	N/A	N/A	N/A
R7gd.12xlarge.search	N/A	N/A	N/A
R7gd.16xlarge.search	N/A	N/A	N/A
i4i.large.search	N/A	N/A	N/A
i4i.xlarge.search	N/A	N/A	N/A
i4i.2xlarge.search	N/A	N/A	N/A
i4i.4xlarge.search	N/A	N/A	N/A
i4i.8xlarge.search	N/A	N/A	N/A
i4i.12xlarge.search	N/A	N/A	N/A
i4i.16xlarge.search	N/A	N/A	N/A
i4i.24xlarge.search	N/A	N/A	N/A
i4i.32xlarge.search	N/A	N/A	N/A
i4g.large.search	N/A	N/A	N/A
i4g.xlarge.search	N/A	N/A	N/A
i4g.2xlarge.search	N/A	N/A	N/A
i4g.4xlarge.search	N/A	N/A	N/A
i4g.8xlarge.search	N/A	N/A	N/A
i4g.16xlarge.search	N/A	N/A	N/A

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
i8g.large.search	N/A	N/A	N/A
i8g.xlarge.search	N/A	N/A	N/A
i8g.2xlarge.search	N/A	N/A	N/A
i8g.4xlarge.search	N/A	N/A	N/A
i8g.8xlarge.search	N/A	N/A	N/A
i8g.12xlarge.search	N/A	N/A	N/A
i8g.16xlarge.search	N/A	N/A	N/A
i7i.large.search	N/A	N/A	N/A
i7i.xlarge.search	N/A	N/A	N/A
i7i.2xlarge.search	N/A	N/A	N/A
i7i.4xlarge.search	N/A	N/A	N/A
i7i.8xlarge.search	N/A	N/A	N/A
i7i.12xlarge.search	N/A	N/A	N/A
i7i.16xlarge.search	N/A	N/A	N/A
c7i.large.search	10 GiB	N/A	256 GiB
c7i.xlarge.search	10 GiB	N/A	512 GiB
c7i.2xlarge.search	10 GiB	N/A	1 TiB
c7i.4xlarge.search	10 GiB	N/A	1.5 TiB
c7i.8xlarge.search	10 GiB	N/A	3 TiB
c7i.12xlarge.search	10 GiB	N/A	4.5 TiB

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
c7i.16xlarge.search	10 GiB	N/A	6 TiB
m7i.large.search	10 GiB	N/A	768 GiB
m7i.xlarge.search	10 GiB	N/A	2 TiB
m7i.2xlarge.search	10 GiB	N/A	3 TiB
m7i.4xlarge.search	10 GiB	N/A	6 TiB
m7i.8xlarge.search	10 GiB	N/A	12 TiB
m7i.12xlarge.search	10 GiB	N/A	18 TiB
m7i.16xlarge.search	10 GiB	N/A	24 TiB
r7i.large.search	10 GiB	N/A	1.5 TiB
r7i.xlarge.search	10 GiB	N/A	3 TiB
r7i.2xlarge.search	10 GiB	N/A	6 TiB
r7i.4xlarge.search	10 GiB	N/A	12 TiB
r7i.8xlarge.search	10 GiB	N/A	16 TiB
r7i.12xlarge.search	10 GiB	N/A	24 TiB
r7i.12xlarge.search	10 GiB	N/A	36 TiB
c8g.large	10 GiB	N/A	256 GiB
c8g.xlarge	10 GiB	N/A	512 GiB
c8g.2xlarge	10 GiB	N/A	1 TiB
c8g.4xlarge	10 GiB	N/A	1.5 TiB
c8g.8xlarge	10 GiB	N/A	3 TiB

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
c8g.12xlarge	10 GiB	N/A	4.5 TiB
c8g.16xlarge	10 GiB	N/A	6 TiB
m8g.medium	10 GiB	N/A	512 GiB
m8g.large	10 GiB	N/A	768 GiB
m8g.xlarge	10 GiB	N/A	2 TiB
m8g.2xlarge	10 GiB	N/A	3 TiB
m8g.4xlarge	10 GiB	N/A	6 TiB
m8g.8xlarge	10 GiB	N/A	12 TiB
m8g.12xlarge	10 GiB	N/A	18 TiB
m8g.16xlarge	10 GiB	N/A	24 TiB
r8g.medium	10 GiB	N/A	768 GiB
r8g.large	10 GiB	N/A	1532 GiB
r8g.xlarge	10 GiB	N/A	3 TiB
r8g.2xlarge	10 GiB	N/A	6 TiB
r8g.4xlarge	10 GiB	N/A	12 TiB
r8g.8xlarge	10 GiB	N/A	16 TiB
r8g.12xlarge	10 GiB	N/A	24 TiB
r8g.16xlarge	10 GiB	N/A	36 TiB
r8gd.medium	N/A	N/A	N/A
r8gd.large	N/A	N/A	N/A

執行個體類型	最小 EBS 大小	最大 EBS 大小 (gp2)	最大 EBS 大小 (gp3)
r8gd.xlarge	N/A	N/A	N/A
r8gd.2xlarge	N/A	N/A	N/A
r8gd.4xlarge	N/A	N/A	N/A
r8gd.8xlarge	N/A	N/A	N/A
r8gd.12xlarge	N/A	N/A	N/A
r8gd.16xlarge	N/A	N/A	N/A
oi2.large.search	N/A	N/A	N/A
oi2.xlarge.search	N/A	N/A	N/A
oi2.2xlarge.search	N/A	N/A	N/A
oi2.4xlarge.search	N/A	N/A	N/A
oi2.8xlarge.search	N/A	N/A	N/A
oi2.12xlarge.search	N/A	N/A	N/A
oi2.16xlarge.search	N/A	N/A	N/A

網路配額

下表顯示 HTTP 請求承載的大小上限。

執行個體類型	HTTP 請求承載的大小上限
t2.micro.search	10 MiB
t2.small.search	10 MiB
t2.medium.search	10 MiB

執行個體類型	HTTP 請求承載的大小上限
t3.small.search	10 MiB
t3.medium.search	10 MiB
m3.medium.search	10 MiB
m3.large.search	10 MiB
m3.xlarge.search	100 MiB
m3.2xlarge.search	100 MiB
m4.large.search	10 MiB
m4.xlarge.search	100 MiB
m4.2xlarge.search	100 MiB
m4.4xlarge.search	100 MiB
m4.10xlarge.search	100 MiB
m5.large.search	10 MiB
m5.xlarge.search	100 MiB
m5.2xlarge.search	100 MiB
m5.4xlarge.search	100 MiB
m5.12xlarge.search	100 MiB
m6g.large.search	10 MiB
m6g.xlarge.search	100 MiB
m6g.2xlarge.search	100 MiB
m6g.4xlarge.search	100 MiB

執行個體類型	HTTP 請求承載的大小上限
m6g.8xlarge.search	100 MiB
m6g.12xlarge.search	100 MiB
c4.large.search	10 MiB
c4.xlarge.search	100 MiB
c4.2xlarge.search	100 MiB
c4.4xlarge.search	100 MiB
c4.8xlarge.search	100 MiB
c5.large.search	10 MiB
c5.xlarge.search	100 MiB
c5.2xlarge.search	100 MiB
c5.4xlarge.search	100 MiB
c5.9xlarge.search	100 MiB
c5.18xlarge.search	100 MiB
c6g.large.search	10 MiB
c6g.xlarge.search	100 MiB
c6g.2xlarge.search	100 MiB
c6g.4xlarge.search	100 MiB
c6g.8xlarge.search	100 MiB
c6g.12xlarge.search	100 MiB
r3.large.search	10 MiB

執行個體類型	HTTP 請求承載的大小上限
r3.xlarge.search	100 MiB
r3.2xlarge.search	100 MiB
r3.4xlarge.search	100 MiB
r3.8xlarge.search	100 MiB
r4.large.search	100 MiB
r4.xlarge.search	100 MiB
r4.2xlarge.search	100 MiB
r4.4xlarge.search	100 MiB
r4.8xlarge.search	100 MiB
r4.16xlarge.search	100 MiB
r5.large.search	100 MiB
r5.xlarge.search	100 MiB
r5.2xlarge.search	100 MiB
r5.4xlarge.search	100 MiB
r5.12xlarge.search	100 MiB
r6g.large.search	100 MiB
r6g.xlarge.search	100 MiB
r6g.2xlarge.search	100 MiB
r6g.4xlarge.search	100 MiB
r6g.8xlarge.search	100 MiB

執行個體類型	HTTP 請求承載的大小上限
r6g.12xlarge.search	100 MiB
r6gd.large.search	100 MiB
r6gd.xlarge.search	100 MiB
r6gd.2xlarge.search	100 MiB
r6gd.4xlarge.search	100 MiB
r6gd.8xlarge.search	100 MiB
r6gd.12xlarge.search	100 MiB
r6gd.16xlarge.search	100 MiB
i2.xlarge.search	100 MiB
i2.2xlarge.search	100 MiB
i3.large.search	100 MiB
i3.xlarge.search	100 MiB
i3.2xlarge.search	100 MiB
i3.4xlarge.search	100 MiB
i3.8xlarge.search	100 MiB
i3.16xlarge.search	100 MiB
or1.medium.search	10 MiB
or1.large.search	100 MiB
or1.xlarge.search	100 MiB
or1.2xlarge.search	100 MiB

執行個體類型	HTTP 請求承載的大小上限
or1.4xlarge.search	100 MiB
or1.8xlarge.search	100 MiB
or1.12xlarge.search	100 MiB
or1.16xlarge.search	100 MiB
or2.medium.search	100 MiB
or2.large.search	100 MiB
or2.xlarge.search	100 MiB
or2.2xlarge.search	100 MiB
or2.4xlarge.search	100 MiB
or2.8xlarge.search	100 MiB
or2.12xlarge.search	100 MiB
or2.16xlarge.search	100 MiB
om2.large.search	10 MiB
om2.xlarge.search	100 MiB
om2.2xlarge.search	100 MiB
om2.4xlarge.search	100 MiB
om2.8xlarge.search	100 MiB
om2.12xlarge.search	100 MiB
om2.16xlarge.search	100 MiB
im4gn.large.search	100 MiB

執行個體類型	HTTP 請求承載的大小上限
im4gn.xlarge.search	100 MiB
im4gn.2xlarge.search	100 MiB
im4gn.4xlarge.search	100 MiB
im4gn.8xlarge.search	100 MiB
im4gn.16xlarge.search	100 MiB
i4i.large.search	N/A
i4i.xlarge.search	N/A
i4i.2xlarge.search	N/A
i4i.4xlarge.search	N/A
i4i.8xlarge.search	N/A
i4i.12xlarge.search	N/A
i4i.16xlarge.search	N/A
i4i.24xlarge.search	N/A
i4i.32xlarge.search	N/A
i4g.large.search	N/A
i4g.xlarge.search	N/A
i4g.2xlarge.search	N/A
i4g.4xlarge.search	N/A
i4g.8xlarge.search	N/A
i4g.16xlarge.search	N/A

執行個體類型	HTTP 請求承載的大小上限
i8g.large.search	N/A
i8g.xlarge.search	N/A
i8g.2xlarge.search	N/A
i8g.4xlarge.search	N/A
i8g.8xlarge.search	N/A
i8g.12xlarge.search	N/A
i8g.16xlarge.search	N/A
i7i.large.search	100 MiB
i7i.xlarge.search	100 MiB
i7i.2xlarge.search	100 MiB
i7i.4xlarge.search	100 MiB
i7i.8xlarge.search	100 MiB
i7i.12xlarge.search	100 MiB
i7i.16xlarge.search	100 MiB
c7i.large.search	100 MiB
c7i.xlarge.search	100 MiB
c7i.2xlarge.search	100 MiB
c7i.4xlarge.search	100 MiB
c7i.8xlarge.search	100 MiB
c7i.12xlarge.search	100 MiB

執行個體類型	HTTP 請求承載的大小上限
c7i.16xlarge.search	100 MiB
m7i.large.search	100 MiB
m7i.xlarge.search	100 MiB
m7i.2xlarge.search	100 MiB
m7i.4xlarge.search	100 MiB
m7i.8xlarge.search	100 MiB
m7i.12xlarge.search	100 MiB
m7i.16xlarge.search	100 MiB
r7i.large.search	100 MiB
r7i.xlarge.search	100 MiB
r7i.2xlarge.search	100 MiB
r7i.4xlarge.search	100 MiB
r7i.8xlarge.search	100 MiB
r7i.12xlarge.search	100 MiB
r7i.16xlarge.search	100 MiB
c8g.large	100 MiB
c8g.xlarge	100 MiB
c8g.2xlarge	100 MiB
c8g.4xlarge	100 MiB
c8g.8xlarge	100 MiB

執行個體類型	HTTP 請求承載的大小上限
c8g.12xlarge	100 MiB
c8g.16xlarge	100 MiB
m8g.medium	100 MiB
m8g.large	100 MiB
m8g.xlarge	100 MiB
m8g.2xlarge	100 MiB
m8g.4xlarge	100 MiB
m8g.8xlarge	100 MiB
m8g.12xlarge	100 MiB
m8g.16xlarge	100 MiB
r8g.medium	100 MiB
r8g.large	100 MiB
r8g.xlarge	100 MiB
r8g.2xlarge	100 MiB
r8g.4xlarge	100 MiB
r8g.8xlarge	100 MiB
r8g.12xlarge	100 MiB
r8g.16xlarge	100 MiB
r8gd.medium	100 MiB
r8gd.large	100 MiB

執行個體類型	HTTP 請求承載的大小上限
r8gd.xlarge	100 MiB
r8gd.2xlarge	100 MiB
r8gd.4xlarge	100 MiB
r8gd.8xlarge	100 MiB
r8gd.12xlarge	100 MiB
r8gd.16xlarge	100 MiB
oi2.large.search	100 MiB
oi2.xlarge.search	100 MiB
oi2.2xlarge.search	100 MiB
oi2.4xlarge.search	100 MiB
oi2.8xlarge.search	100 MiB
oi2.12xlarge.search	100 MiB
oi2.16xlarge.search	100 MiB

碎片大小配額

下一節列出各種執行個體系列的碎片大小上限。

執行個體類型	無待命的異地同步備份	具有待命的異地同步備份
R5, C5, M5, C7i, M7i, R7i	N/A	65 GiB
I3、i4i、i4g、i8g、i7i	N/A	65 GiB

執行個體類型	無待命的異地同步備份	具有待命的異地同步備份
R6g, C6g, M6g, R6gd, C7g,M7g, R7g, R7gd, C8g, M8g, R8g, R8gd	N/A	65 GiB
OR1, OR2, OM2, OI2	N/A	N/A
Im4gn	N/A	65 GiB

若要申請增加配額，請聯絡AWS 支援人員。

碎片計數配額

下一節列出 OpenSearch 版本的碎片計數上限。

引擎版本	限制	備註
Elasticsearch 1.5 到 6.x	無預設限制	
Elasticsearch 7.x	1000	可以透過叢集變更預設限制。max_shards_per_node 設定。
OpenSearch 1.x 到 2.15	1000	可以透過叢集變更預設限制。max_shards_per_node 設定。
OpenSearch 2.17 及更高版本	每 16 GB 堆積 1000 個，最多 4000 個	預設限制無法變更。

Java 處理序配額

OpenSearch Service 將 Java 處理記憶體的限制為可用記憶體總數的 50%（上限為 32 GiB)。32 GB 的上限不適用於 r7g 和 OpenSearch 最佳化執行個體。

若要請求提高限制，請聯絡 AWS Support。

進階使用者可以指定用於欄位資料的堆積百分比。如需詳細資訊，請參閱進階叢集設定和 JVM OutOfMemoryError。

網域政策配額

OpenSearch Service 將網域上的存取政策限制為 100 KiB。

Amazon OpenSearch Service 中的預留執行個體

相較於標準隨需執行個體，Amazon OpenSearch Service 中的預留執行個體 (RI) 提供大幅折扣。執行個體本身是相同的；RIs 只是套用至您帳戶中隨需執行個體的帳單折扣。對於可預測使用量的長效應用程式，RI 可隨時間的累積產生可觀的節省成果。

OpenSearch Service RI 需要一年或三年期預留執行個體，並且提供三種對折扣費率有相當影響的付款選項：

- 不預付 - 無須任何預付款。在期限內，按折扣後的每小時費率支付每個小時的費用。
- 部分預付 - 您需要預付部分費用，並且在期限內，按折扣後的每小時費率支付每個小時的費用。
- 全部預付 - 您支付整個預付款的費用。您不是按照小時費率支付費用。

一般來說，較大的預付款表示較大的折扣。您無法取消預留執行個體 - 當您預留時，表示您承諾支付整個期限 - 且預付款項不可退還。

RI 並不靈活；它們只適用於您保留的確切執行個體類型。例如，八個 `c5.2xlarge.search` 執行個體保留不適用於十六個 `c5.xlarge.search` 執行個體或四個 `c5.4xlarge.search` 執行個體。不過，只要執行個體類型、區域、系列和大小相符，屬於 中組織一部分的連結帳戶 AWS Organizations 可以從 RI 擁有的帳戶的任何未使用的折扣應用程式受益。如需詳細資訊，請參閱 Amazon OpenSearch Service 定價和常見問答集。

購買預留執行個體 (主控台)

主控台可讓您檢視現有的預留執行個體和購買新的執行個體。

購買預訂

1. 前往 https://aws.amazon.com，然後選擇 Sign In to the Console (登入主控台)。
2. 在 Analytics (分析) 下，選擇 Amazon OpenSearch Service。
3. 在導覽窗格中，選擇 Reserved Instance Leases (預留執行個體租用)。

在此頁面上，您可以檢視現有的預訂。如果您有許多預訂，您可以篩選以更輕鬆地識別和檢視特定預訂。

> **Tip**
>
> 若您沒有看到 Reserved Instance Leases (預留執行個體租用) 連結，請在 AWS 區域中建立網域。

4. 選擇 Order Reserved Instance (預訂預留執行個體)。
5. 提供唯一且描述性名稱。
6. 選擇執行個體類型和執行個體數量。如需準則，請參閱the section called “調整網域大小”。
7. 選擇期限長度和付款選項。仔細檢閱付款詳細資訊。
8. 選擇下一步。
9. 仔細檢閱購買摘要。購買預留執行個體無法退款。
10. 選擇 Order (預訂)。

購買預留執行個體 (AWS CLI)

AWS CLI 具有檢視方案、購買保留和檢視保留的命令。下列命令和範例回應會顯示指定 的方案 AWS 區域：

```
aws opensearch describe-reserved-instance-offerings --region us-east-1
{
  "ReservedInstanceOfferings": [
    {
      "FixedPrice": x,
      "ReservedInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "UsagePrice": 0.0,
      "PaymentOption": "PARTIAL_UPFRONT",
      "Duration": 31536000,
      "InstanceType": "m4.2xlarge.search",
      "CurrencyCode": "USD"
```

```
    }
  ]
}
```

如需每個傳回值的詳細說明，請參閱下表。

欄位	Description
FixedPrice	預訂的預付費用。
ReservedInstanceOfferingId	方案 ID。如果您想要預訂產品，請備註此值。
RecurringCharges	預訂的小時費率。
UsagePrice	傳統欄位。對於 OpenSearch Service，此值始終為 0。
PaymentOption	無預付、部分預付或全部預付。
Duration	期限長度，以秒為單位： • 31536000 秒是一年。 • 94608000 秒是三年。
InstanceType	預訂的執行個體類型。如需分配給每個執行個體類型的硬體資源的資訊，請參閱 Amazon OpenSearch Service 定價。
CurrencyCode	用於 FixedPrice 和 RecurringChargeAmount 的貨幣。

這個下一個範例購買預訂：

```
aws opensearch purchase-reserved-instance-offering --reserved-instance-offering-
id 1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a --reservation-name my-reservation --instance-
count 3 --region us-east-1
{
  "ReservationName": "my-reservation",
  "ReservedInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a"
```

```
}
```

最後，您可以使用下列範例，列出指定區域的預訂：

```
aws opensearch describe-reserved-instances --region us-east-1
{
  "ReservedInstances": [
    {
      "FixedPrice": x,
      "ReservedInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "ReservationName": "my-reservation",
      "PaymentOption": "PARTIAL_UPFRONT",
      "UsagePrice": 0.0,
      "ReservedInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "State": "payment-pending",
      "StartTime": 1522872571.229,
      "InstanceCount": 3,
      "Duration": 31536000,
      "InstanceType": "m4.2xlarge.search",
      "CurrencyCode": "USD"
    }
  ]
}
```

> **Note**
>
> `StartTime` 是 Unix epoch 時間，以秒數計算，從 1970 年 1 月 1 日 UTC 午夜起已經過的秒數。例如，1522872571 epoch 時間是 2018 年 4 月 4 日 20:09:31 UTC。您可以使用線上轉換器。

若要進一步了解前述範例中使用的命令，請參閱 AWS CLI 命令參考。

購買預留執行個體 (AWS SDKs)

AWS SDKs(Android 和 iOS SDKs除外） 支援 Amazon OpenSearch Service API 參考中定義的所有操作，包括下列項目：

- `DescribeReservedInstanceOfferings`
- `PurchaseReservedInstanceOffering`
- `DescribeReservedInstances`

此範例指令碼使用來自 的 OpenSearchService 低階 Python 用戶端 適用於 Python (Boto3) 的 AWS SDK 來購買預留執行個體。您必須提供 `instance_type` 的值。

```
import boto3
from botocore.config import Config

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-east-1'
)

client = boto3.client('opensearch', config=my_config)

instance_type = '' # e.g. m4.2xlarge.search


def describe_RI_offerings(client):
    """Gets the Reserved Instance offerings for this account"""

    response = client.describe_reserved_instance_offerings()
    offerings = (response['ReservedInstanceOfferings'])
    return offerings


def check_instance(offering):
    """Returns True if instance type is the one you specified above"""

    if offering['InstanceType'] == instance_type:
```

```
        return True

    return False


def get_instance_id():
    """Iterates through the available offerings to find the ID of the one you
 specified"""

    instance_type_iterator = filter(
        check_instance, describe_RI_offerings(client))
    offering = list(instance_type_iterator)
    id = offering[0]['ReservedInstanceOfferingId']
    return id


def purchase_RI_offering(client):
    """Purchase Reserved Instances"""

    response = client.purchase_reserved_instance_offering(
        ReservedInstanceOfferingId = get_instance_id(),
        ReservationName = 'my-reservation',
        InstanceCount = 1
    )
    print('Purchased reserved instance offering of type ' + instance_type)
    print(response)


def main():
    """Purchase Reserved Instances"""
    purchase_RI_offering(client)
```

如需安裝和使用 AWS SDKs 的詳細資訊，請參閱AWS 軟體開發套件。

檢查成本

Cost Explorer 是一個免費的工具，可讓您用於檢視過去 13 個月的支出費用資料。分析這項資料可協助您識別趨勢，並了解 RI 是否符合您的使用案例。如果您已經有 RI，可以透過https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/groupdata.html購買選項和顯示分攤成本分類，進而針對隨需執行個體比較該項支出和您的支出。您也可以設定用量預算，確保充分利用您的預留額度。如需詳細資訊，請參閱 AWS Billing 使用者指南中的使用 Cost Explorer 分析您的成本。

Amazon OpenSearch Service 的成本最佳化技術

以下是在使用 Amazon OpenSearch Service 時最佳化成本的最常見技術：受管叢集和無伺服器。由於每個工作負載都是唯一的，請根據您的特定使用模式評估這些策略，並在測試環境中驗證它們，然後再套用至生產環境。

主題

- Amazon OpenSearch Service 受管叢集的成本最佳化
- Amazon OpenSearch Service Serverless 的成本最佳化

Amazon OpenSearch Service 受管叢集的成本最佳化

衍生來源 — 略過儲存 _source 欄位

衍生來源是一種儲存最佳化功能，可消除儲存`_source`欄位的額外負荷：

- OpenSearch 會儲存每個擷取的文件兩次：一次在 `_source` 欄位中（原始文件），一次作為索引欄位進行搜尋。
- 光是 `_source` 欄位就可能會耗用大量儲存空間，通常是總索引儲存空間的 30–50%。
- 使用衍生來源時，您可以略過儲存`_source`，並在需要時從索引欄位動態重建它（搜尋、取得、mget、重新索引或更新操作期間）。
- 這是選擇加入，在建立索引時使用複合索引設定啟用。
- 適用於支援 OpenSearch 3.1 或更新版本的所有區域。

最適合：分析和日誌工作負載，您不需要經常擷取原始文件，但仍需要搜尋和彙總欄位。

如需詳細資訊，請參閱開放原始碼文件和最新消息公告。

OR1 / OR2 / OM2 執行個體 — OpenSearch 最佳化執行個體系列

OR1 和更新的 OR2 和 OM2 執行個體透過區段複寫使用 Amazon S3 進行複本儲存：

- OR2：比 OR1 高出高達 26% 的索引輸送量，比 R7g 執行個體多出 70%。
- OM2：索引輸送量比 OR1 高出 15%，比 M7g 執行個體多出 66%。
- 兩者都使用相同的架構：用於主要儲存的本機 EBS 和用於耐用性的 S3。
- 消除複本儲存成本 — 存放在 S3 中的複本 (11 九秒耐久性），而不是昂貴的 EBS 磁碟區。

- 比上一代執行個體改善高達 30% 的價格效能。
- 支援淺快照 v2 — 近乎即時的快照，沒有 I/O 額外負荷。

最適合：索引密集型操作分析和日誌分析工作負載。

如需詳細資訊，請參閱 OR2 和 OM2 最新消息公告和the section called “OpenSearch 最佳化執行個體”開發人員指南主題。

索引彙總 — 彙總歷史時間序列資料

索引彙總摘要並將較舊的時間序列資料壓縮為更粗的時間間隔，大幅減少儲存磁碟區：

- IoT/感應器資料：將每秒資料保存在最近期間的熱儲存中；針對較舊的資料彙總為每小時或每日摘要。
- 系統指標：保留過去 30 天的詳細指標；將較舊的資料彙總為每小時或每日摘要。
- 日誌資料：保留作用中故障診斷時段的完整詳細資訊（例如 1 週）；維護較舊期間的摘要錯誤模式。
- 結合 ISM 政策，在單一生命週期政策中自動化彙總和層遷移。
- 從秒到小時的彙總與從秒到分鐘的彙總時節省更多。

如需詳細資訊，請參閱索引彙總部落格文章。

索引狀態管理 — 自動化完整資料生命週期

ISM 政策會透過儲存層和生命週期動作自動移動索引：

- 自動遷移索引：根據存留期、大小或文件計數，從熱到 UltraWarm 到冷再刪除。
- 在層轉換之前觸發彙總，以減少資料量。
- 設定轉換政策（例如，當索引達到 50 GB 或 30 天）以控制索引成長。
- 自動化唯讀索引上的強制合併，以從已刪除的文件回收儲存。
- 結合彙總，可大幅節省大量時間序列資料集的成本。

預留執行個體 — 承諾可預測折扣

對於穩定、可預測的分析工作負載，預留執行個體提供高於隨需定價的大幅折扣：

- 無預付、部分預付或所有預付付款選項的 1 年或 3 年承諾條款。

- 最適合持續執行的熱層資料節點和專用主節點。
- 使用 AWS 定價計算器在遞交之前預估節省成本。
- 預留執行個體是套用至隨需執行個體的帳單折扣，不需要變更基礎設施。

適當大小的執行個體類型和計數

Well-Architected OpenSearch Lens 和適當調整大小最佳實務的重要指導：

- 一律使用最新一代的執行個體 （例如，Graviton3 Graviton2-based 執行個體的效能提升高達 25%)。
- 使用 gp3 EBS 磁碟區而非 gp2 — 以較低的成本提供更好的效能，無需額外付費。
- 將執行個體類型與工作負載配對：針對大量搜尋進行記憶體最佳化、針對大量索引進行運算最佳化。
- 評估專用叢集管理員節點：只需要 3 個以上的資料節點；避免過度佈建主節點大小。
- 監控 CloudWatch 指標以偵測過度佈建：持續 CPU 低於 40%、JVM 低於 50%，以及儲存低於 50% 是浪費的跡象。
- 最佳範圍：持續工作負載的 CPU 60–80%、JVM 65–85%、儲存 70–85%。

如需詳細資訊，請參閱正確調整大小最佳實務部落格文章。

碎片最佳化 — 避免過度分片

過度分片是隱藏的成本驅動因素 — 太多小碎片浪費 CPU、記憶體和 JVM 堆積：

- 建議的碎片大小：根據工作負載，每個碎片 10–50 GiB。
- 每個 GiB 的 Java 堆積不超過 25 個碎片，每個資料節點不超過 1，000 個碎片。
- 使用 ISM 輪換政策來控制索引成長，並避免不受限制的碎片擴散。
- 在耐久性允許的情況下減少複本計數 (OR1/OR2 執行個體完全不需要複本）。
- 在唯讀索引上使用強制合併，以減少區段計數並回收儲存。

如需詳細資訊，請參閱我需要多少碎片？

使用 Amazon S3 進行零 ETL/直接查詢

對於極少被查詢但必須保持可存取的資料，直接查詢 （含 S3 的零 ETL) 允許直接從 OpenSearch 查詢 S3 資料，而不擷取資料：

- 沒有擷取成本 — 資料會保留在 S3 中。
- 封存資料沒有熱層儲存成本。
- Pay-per-query的運算模型。
- 支援 OpenSearch Dashboards 進行視覺化。
- 可接受秒或分鐘的延遲，不適用於即時使用案例。

在擷取時取樣和壓縮

在資料到達 OpenSearch 之前降低成本：

- 取樣：僅擷取大量日誌串流的代表性子集（例如 10% 的偵錯日誌）。
- 索引壓縮：啟用最佳壓縮轉碼器以減少儲存體使用量。
- 欄位篩選：在編製索引之前捨棄高基數、低值欄位（例如，舊日誌的原始堆疊追蹤）。
- 保留政策：定義符合合規要求的最大保留時段 — 永遠不會無限期存放資料。

避免延長支援成本 — 在引擎版本上保持最新狀態

Amazon OpenSearch Service 會針對延伸支援中的引擎版本，按標準化執行個體小時收取固定費用：

- 除了執行個體成本之外，繼續使用較舊、不支援的版本會產生額外費用。
- 升級至目前支援的版本，以避免延長支援費用。

成本分配標籤和 CloudWatch 監控

主動成本控管可防止浪費：

- 將成本分配標籤套用至 OpenSearch 網域，以追蹤每個團隊或工作負載的詳細成本。
- 設定儲存使用率、JVM 壓力和 CPU 的 CloudWatch 警示，以提早擷取過度佈建。
- 使用 AWS Cost Explorer 來識別持續低使用率的網域。
- 評估自動調整 — 自動調整 JVM 堆積大小和其他設定，以提高效能並減少資源浪費。

Amazon OpenSearch Service Serverless 的成本最佳化

磁碟最佳化向量搜尋 （向量集合）

磁碟最佳化向量搜尋是向量工作負載最強大的降低成本技術之一。它只會將壓縮向量保留在 RAM 中，並在磁碟上儲存完整精確向量，以記憶體內模式成本的一小部分執行向量搜尋。

運作方式：

- 在標準 (`in_memory`) 模式中，完整的 HNSW 圖形會載入 RAM 中，這會大規模變得非常昂貴。
- 在 `on_disk` 模式中，只有壓縮 （量化） 向量會保留在記憶體中，以產生候選項目；完整精確度向量只會從磁碟擷取到最終重新評分階段 （兩階段搜尋）。
- 這可大幅降低 RAM 需求，同時維持高搜尋品質。
- 預設`on_disk`模式使用 32 倍二進位量化，相較於記憶體內模式，記憶體需求減少 97%。
- 支援壓縮層級：2x (FP16)、4x （位元組）、8x、16x、32x （二進位）。
- 100–200 毫秒範圍內的 P90 延遲 – 適用於不需要單一位數毫秒回應時間的工作負載。

節省成本基準：

資料集	Recall@100	P90 延遲	降低成本
Cohere TREC-RAG	0.94	104 毫秒	83%
Tasb-1M	0.96	7 毫秒	67%
Marco-1M	0.99	7 毫秒	67%

最適合：RAG 管道、語意搜尋、文件擷取，以及任何接受 P90 延遲 100–200 毫秒且降低成本的向量工作負載。

> **Note**
>
> 若要將此變更套用至現有的索引資料，您需要重新索引。您可以使用外部管道工具，例如 將資料重新索引到新的索引。

如需詳細資訊，請參閱磁碟最佳化向量搜尋部落格文章、量化技術部落格文章和 the section called “使用向量搜尋集合”。

向量自動最佳化（向量集合）

自動最佳化會自動評估向量索引組態，並建議搜尋品質、延遲和記憶體成本之間的最佳權衡，而不需要向量專業知識：

- 在一小時內在 中提供最佳化建議。
- 與向量擷取管道整合。
- 可與 GPU 加速索引結合使用，適用於 10 億規模的向量資料庫。

如需詳細資訊，請參閱自動最佳化部落格文章。

GPU 加速向量索引（向量集合）

GPU 加速會將 HNSW 向量索引卸載至無伺服器 GPUs，大幅降低建置大型向量索引的時間和 OCU 成本：

- 相較於僅使用 CPU 的索引，索引建置時間快 6.4 倍到 13.8 倍。
- 高達 75% 的寫入密集型向量工作負載索引 OCU 成本。
- GPUs 是動態連接，您只需在 GPU 加速作用中時支付 OCUs 的費用。
- 允許在一小時內內建數十億的向量資料庫。
- 以向量加速 OCUs 單獨收費。

最適合：大規模初始向量擷取或頻繁的模型重新訓練案例，其中重建索引的成本很高。

如需詳細資訊，請參閱 GPU Acceleration 部落格文章。

設定最大 OCU 限制（所有集合類型）

Amazon OpenSearch Service Serverless 會根據需求自動擴展 OCUs。如果沒有上限，成本可能會意外激增。在帳戶層級或每個集合群組設定最大 OCU 限制，以防止失控擴展。在尖峰負載期間，系統會擴展至此限制，但不會超過。

允許的值：

- 帳戶層級：任何高達 1，700 個 OCUs 的值（不限於 16 的倍數）。

- 集合群組：1、2、4、8、16 和 16 到 1，696 個 OCUs。

監控 CloudWatch 指標 (`OCUUtilization`) 以隨時間調整最大 OCU 設定的大小。

> **Note**
>
> 如果使用率達到 OCU 上限，即使包含成本，效能也可能會大幅降低。嵌入上限無法解決 OCU 峰值的根本原因。對於向量集合，根本原因通常是記憶體內向量，應透過最佳化向量索引、減少索引大小或調校取回和準確性權衡來直接解決此問題。

> **Tip**
>
> 從保守最大 OCU 開始，並只在 CloudWatch 在上限附近顯示持續使用率時增加。如果您持續達到上限，請調查根本原因，特別是向量集合的記憶體內向量使用量，而不只是提高限制。

如需詳細資訊，請參閱the section called “管理容量限制”。

最佳化保留期間 （時間序列集合）

資料生命週期政策會在指定的保留期間後自動刪除時間序列集合中的資料，防止儲存成長不受限制。只有時間序列集合支援資料生命週期政策 — 搜尋和向量集合不支援。

時間序列集合的 OCU 計數直接取決於必須保留在本機儲存中的最近資料量。時間序列集合只會保留本機 OCU 儲存中最近一部分的資料；較舊的資料會卸載至 S3，而 OCUs 的數量也會隨之擴展：

所需的 OCUs = max（最小 OCUs、在保留時段內保留資料所需的 OCUs)

設定資料生命週期政策：

- 將每個索引或索引模式的保留期間從 24 小時設定為 3，650 天。
- Amazon OpenSearch Service Serverless 會盡力自動刪除資料 （通常在保留期間的 48 小時或 10% 內）。
- 規則可以在集合層級、索引模式層級或個別索引層級套用 — 以更具體的規則為優先。

調整大小範例：

- 1 TiB/天擷取，保留 30 天 = 大約 1 TiB 的熱資料 = 20 個用於索引OCUs + 20 個用於搜尋OCUs。

- 減少 7 天的保留期 = 大約 233 GiB 的熱資料 = 大約 4 個用於索引OCUs + 4 個用於搜尋OCUs。

較短的保留意味著本機儲存體中的熱資料較少、所需的 OCUs較少，以及較低的運算帳單。根據實際業務和合規要求調整保留期 — 預設不會無限期保留資料。

如需詳細資訊，請參閱the section called “使用資料生命週期政策”。

避免儲存不必要的資料（所有集合類型）

減少直接擷取的資料量可同時降低運算 (OCUs和儲存成本：

- 在擷取時篩選欄位：使用管道在欄位到達集合之前捨棄低值欄位。
- 避免擷取重複或備援的資料：在管道層級重複刪除。
- 使用適當的索引映射：在存放但從未搜尋的欄位上停用索引 (`"index": false`)。
- 對於搜尋集合：避免儲存大型二進位 Blob 或無需搜尋值即可膨脹儲存體的原始文字。

多租戶工作負載的集合群組（所有集合類型）

集合群組允許具有不同 KMS 金鑰的多個集合在相同的安全界限內共用 OCU 資源，大幅降低多租戶架構的成本。適用於每個租用戶或每個集合使用多個 KMS 金鑰的客戶：

- 先前，每個唯一的 KMS 金鑰都需要專用 OCUs，使得每個租戶的隔離成本過高。
- 使用集合群組時，具有個別加密金鑰的租用戶可以共用 OCU 容量。
- 為大量小型租戶工作負載節省高達 90% 的成本。
- 支援最小 OCUs（保證基準，無冷啟動）和最大 OCUs（成本上限）。
- 具有不同網路存取類型的集合（公有和 VPC) 可以共存在同一群組中。
- CloudWatch 指標提供每個群組對資源消耗和延遲的可見性。

最適合：SaaS 提供者、多租用戶平台或任何具有許多小型集合的工作負載，每個集合都需要自己的 KMS 金鑰。

如需詳細資訊，請參閱集合群組部落格文章、最新消息公告和 the section called “集合群組”。

Amazon OpenSearch Service 中的其他支援資源

本主題說明 Amazon OpenSearch Service 支援的其他資源。

bootstrap.memory_lock

OpenSearch Service 在 `opensearch.yml` 中啟用 `bootstrap.memory_lock`，將 JVM 記憶體鎖定並防止作業系統將其交換到磁碟。這適用於所有支援的執行個體類型，但下列項目除外：

- `t2.micro.search`
- `t2.small.search`
- `t2.medium.search`
- `t3.small.search`
- `t3.medium.search`

指令碼模組

OpenSearch Service 支援 Elasticsearch 5.x 和更新網域的指令碼。它不支援 1.5 或 2.3 的指令碼。

支援的指令碼選項包括下列項目：

- Painless
- Lucene 表達式
- Mustache

對於 Elasticsearch 5.5 和更新版本的網域，以及所有 OpenSearch 網域，OpenSearch Service 使用 `_scripts` 端點支援已儲存的指令碼。Elasticsearch 5.3 和 5.1 網域僅支援內嵌指令碼。

TLS 傳輸

OpenSearch Service 支援連接埠 80 的 HTTP 和連接埠 443 的 HTTPS，但不支援 TLS 傳輸。

Amazon OpenSearch Service 教學課程

本章包含幾個從開始到結束的教學課程，以便使用 Amazon OpenSearch Service，這包括如何遷移到服務、建置簡單的搜尋應用程式以及在 OpenSearch Dashboards 中建立視覺效果。

主題

- 教學課程：在 Amazon OpenSearch Service 中建立和搜尋文件
- 教學課程：遷移至 Amazon OpenSearch Service
- 教學課程：使用 Amazon OpenSearch Service 建立一個搜尋應用程式
- 教學課程：使用 OpenSearch Service 和 OpenSearch Dashboards，將客戶支援呼叫視覺化

教學課程：在 Amazon OpenSearch Service 中建立和搜尋文件

在本教學課程中，您可學習如何在 Amazon OpenSearch Service 中建立和搜尋文件。您可以 JSON 文件格式將資料新增至索引。OpenSearch Service 會圍繞您新增的第一個文件建立索引。

本教學課程說明如何發出 HTTP 請求以建立文件、自動產生文件 ID，以及如何對文件執行基本搜尋和進階搜尋。

> **Note**
>
> 本教程課程使用具有開放存取權的網域。為了實現最高等級的安全性，我們建議您將網域放入虛擬私有雲端 (VPC) 內。

先決條件

本教學課程具備下列先決條件：

- 您必須擁有 AWS 帳戶。
- 您必須擁有作用中的 OpenSearch Service 網域。

將文件新增至索引

若要將文件新增至索引，您可以使用任何 HTTP 工具，例如 Postman、cURL 或 OpenSearch Dashboards 主控台。這些範例假設您正在使用 OpenSearch Dashboards 中的開發人員主控台。如果您使用其他工具，請視需要提供完整的 URL 和憑證進行相應調整。

若要將文件新增至索引

1. 導覽至您網域的 OpenSearch Dashboards URL。您可以在 OpenSearch Service 主控台網域的儀表板上找到 URL。URL 遵循以下格式：

   ```
   domain-endpoint/_dashboards/
   ```

2. 使用您的主要使用者名稱和密碼登入。
3. 開啟左側導覽面板並選擇 Dev Tools (開發工具)。
4. 用於建立新資源的 HTTP 動詞為 PUT，可使用它來建立新文件和索引。在主控台中，輸入以下命令：

   ```
   PUT fruit/_doc/1
   {
     "name":"strawberry",
     "color":"red"
   }
   ```

   PUT 請求會建立一個名為 fruit 的索引，並將 ID 為 1 的單一文件新增至該索引。它產生如下回應：

   ```
   {
     "_index" : "fruit",
     "_type" : "_doc",
     "_id" : "1",
     "_version" : 1,
     "result" : "created",
     "_shards" : {
       "total" : 2,
       "successful" : 2,
       "failed" : 0
     },
     "_seq_no" : 0,
     "_primary_term" : 1
   }
   ```

建立自動產生的 ID

OpenSearch Service 可以自動為您的文件產生一個 ID。產生 ID 的命令使用 POST 請求而非 PUT 請求，並且不需要文件 ID (與之前的請求相比)。

在開發人員主控台中輸入以下請求：

```
POST veggies/_doc
{
  "name":"beet",
  "color":"red",
  "classification":"root"
}
```

此請求會建立一個名為 veggies 的索引並將文件新增至索引。它產生如下回應：

```
{
  "_index" : "veggies",
  "_type" : "_doc",
  "_id" : "3WgyS4IB5DLqbRIvLxtF",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

請注意，回應中的其他_id欄位，表示 ID 已自動建立。

> **Note**
>
> 您在 URL 的 _doc 後面沒有提供任何內容，ID 通常會出現在該位置。因為您要使用產生的 ID 建立文件，所以尚未提供 ID。這預留用於更新。

使用 POST 命令更新文件

若要更新文件，請使用帶有 ID 號的 HTTP POST 命令。

首先，建立 ID 為 42 的文件：

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow"
}
```

然後使用該 ID 更新文件：

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow",
  "classification":"berries"
}
```

此命令使用新欄位 `classification` 更新文件。它產生如下回應：

```
{
  "_index" : "fruits",
  "_type" : "_doc",
  "_id" : "42",
  "_version" : 2,
  "result" : "updated",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 1,
  "_primary_term" : 1
}
```

> **Note**
>
> 如果您嘗試更新不存在的文件，OpenSearch Service 會建立該文件。

執行大量動作

您可以使用 POST _bulk API 操作對一個請求中的一個或多個索引執行多個動作。大量動作命令採用下列格式：

```
POST /_bulk
<action_meta>\n
<action_data>\n
<action_meta>\n
<action_data>\n
```

每個動作需要兩行 JSON。首先，您要提供動作說明或中繼資料。在下一行中提供該資料。每部分由換行符號 (\n) 分隔。插入的動作描述可能如下所示：

```
{ "create" : { "_index" : "veggies", "_type" : "_doc", "_id" : "7" } }
```

包含資料的下一行可能如下所示：

```
{ "name":"kale", "color":"green", "classification":"leafy-green" }
```

中繼資料和資料結合在一起，表示大量操作中的單一動作。您可以在一個請求中執行許多操作，如下所示：

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "35" } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "36" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "37" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "38" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "39" } }
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
{ "delete" : { "_index" : "vegetables", "_id" : "1" } }
```

請注意，最後一個動作是 delete。delete 動作之後沒有資料。

搜尋文件

現在您的叢集中存在資料，您可以搜尋它。例如，您可能想要搜尋所有根莖類蔬菜，或取得所有綠葉蔬菜的計數，或者尋找每小時記錄的錯誤數量。

基本搜尋

基本搜尋如下所示：

```
GET veggies/_search?q=name:l*
```

該請求會產生一個包含萵苣文件的 JSON 回應。

進階搜尋

您可以在請求主體中以 JSON 格式提供查詢選項，以執行更進階的搜尋：

```
GET veggies/_search
{
  "query": {
    "term": {
      "name": "lettuce"
    }
  }
}
```

此範例還會產生一個帶有萵苣文件的 JSON 回應。

排序

您可以使用排序功能來執行更多此類查詢。首先，您需要重新建立索引，因為自動欄位映射選擇了預設情況下無法排序的類型。傳送下列請求，以刪除並重新建立索引：

```
DELETE /veggies

PUT /veggies
{
   "mappings":{
      "properties":{
         "name":{
            "type":"keyword"
         },
```

```
         "color":{
            "type":"keyword"
         },
         "classification":{
            "type":"keyword"
         }
      }
   }
}
```

然後用資料重新填入索引：

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "7"  } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "8" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "9" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "10" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "11" } }
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
```

現在，您可以使用排序進行搜尋。此請求會按照分類新增遞增排序：

```
GET veggies/_search
{
  "query" : {
    "term": { "color": "green" }
  },
  "sort" : [
      "classification"
  ]
}
```

相關資源

如需詳細資訊，請參閱下列資源：

- 開始使用
- 建立資料索引

- 搜尋資料

教學課程：遷移至 Amazon OpenSearch Service

索引快照是將自我管理的 OpenSearch 或舊版 Elasticsearch 叢集遷移至 Amazon OpenSearch Service 的常用方法。此程序大致包含下列步驟：

1. 建立現有叢集的快照，然後將快照上傳至 Amazon S3 儲存貯體。
2. 建立 OpenSearch Service 網域
3. 授與 OpenSearch Service 存取儲存貯體的許可，並確保您擁有使用快照的許可。
4. 在 OpenSearch Service 網域上還原快照。

此逐步解說提供更詳細的步驟和替代選項，如適用。

建立並上傳快照

雖然您可以使用儲存庫-s3 外掛程式直接將快照擷取到 S3，但您必須在每個節點上安裝外掛程式、調整 `opensearch.yml`(`elasticsearch.yml`如果使用 Elasticsearch 叢集）、重新啟動每個節點、新增您的 AWS 憑證，最後再拍攝快照。外掛程式是適合持續使用或遷移較大型叢集的絕佳選擇。

對於較小的叢集，一次性方法是拍攝共用檔案系統快照，然後使用 AWS CLI 將其上傳至 S3。如果您已經有快照，請跳至步驟 4。

若要建立快照並上傳至 Amazon S3

1. 將 `path.repo` 設定新增至所有節點上的 `opensearch.yml` (或 `Elasticsearch.yml`)，然後重新啟動每個節點。

   ```
   path.repo: ["/my/shared/directory/snapshots"]
   ```

2. 註冊快照儲存庫，此為建立快照前所需。儲存庫只是一個儲存位置：共用檔案系統、Amazon S3、Hadoop 分散式檔案系統 (HDFS) 等。在此情況下，我們將使用共用檔案系統 ("fs")：

   ```
   PUT _snapshot/my-snapshot-repo-name
   {
     "type": "fs",
     "settings": {
       "location": "/my/shared/directory/snapshots"
   ```

```
  }
}
```

3. 建立快照：

```
PUT _snapshot/my-snapshot-repo-name/my-snapshot-name
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

4. 安裝 AWS CLI，然後執行 `aws configure` 以新增您的登入資料。
5. 瀏覽至快照目錄。然後執行下列命令以建立新的 S3 儲存貯體，並將快照目錄的內容上傳至該儲存貯體：

```
aws s3 mb s3://amzn-s3-demo-bucket --region us-west-2
aws s3 sync . s3://amzn-s3-demo-bucket --sse AES256
```

視快照大小和網際網路連線速度而定，此操作可能需要一段時間。

建立網域

雖然主控台是建立網域最簡單的方式，但在這種情況下，您已經開啟終端機並 AWS CLI 安裝 。修改下列命令即可建立符合您需求的網域：

```
aws opensearch create-domain \
  --domain-name migration-domain \
  --engine-version OpenSearch_1.0 \
  --cluster-config InstanceType=c5.large.search,InstanceCount=2 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-user-password}' \
  --access-policies '{"Version": "2012-10-17",       "Statement":
[{"Effect":"Allow","Principal": {"AWS": "arn:aws:iam::aws-
region:user/UserName"},"Action":["es:ESHttp*"],"Resource":"arn:aws:es:aws-
region:111122223333:domain/migration-domain/*"}]}' \
```

```
  --region aws-region
```

若未經修改，此命令會建立具有兩個資料節點的網際網路存取網域，每個節點都有 100 GiB 的儲存空間。它也會啟用具有 HTTP 基本身分驗證和所有加密設定的精細存取控制。如果您需要更進階的安全組態 (例如 VPC)，請使用 OpenSearch Service 主控台。

發出命令之前，請先變更網域名稱、主要使用者登入資料和帳戶號碼。指定您用於 S3 儲存貯體 AWS 區域 的相同 ，以及與您的快照相容的 OpenSearch/Elasticsearch 版本。

> **Important**
>
> 快照只能正向相容，而且只能往前一個主要版本。例如，您無法從 Elasticsearch 7.x 叢集上的 OpenSearch 1.x 叢集還原快照，只能從 OpenSearch 1.x 或 2.x 叢集還原快照。次要版本也很重要。您無法在 5.3.2 OpenSearch Service 網域上從自我管理的 5.3.3 叢集中還原快照。我們建議您選擇您的快照支援的 OpenSearch 或 Elasticsearch 的最新版本。如需相容版本的表格，請參閱the section called “使用快照來遷移資料”。

提供許可以存取 S3 儲存貯體。

在 AWS Identity and Access Management (IAM) 主控台中，建立具有下列許可和信任關係的角色。https://docs.aws.amazon.com/IAM/latest/UserGuide/roles-managingrole-editing-console.html#roles-managingrole_edit-trust-policy在建立角色時，選擇 S3 作為 AWS Service。將角色命名為 `OpenSearchSnapshotRole` 以便輕鬆找到。

許可

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket"
      ]
```

```
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

信任關係

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

然後授予您的個人 IAM 角色許可，以擔任 OpenSearchSnapshotRole。建立下列政策，並將它連接到您的身分：

許可

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
    }
  ]
}
```

映射 OpenSearch Dashboards 中的快照角色 (如果使用精細存取控制)

如果您已啟用精細存取控制，即使您將 HTTP 基本身分驗證用於所有其他目的，您也需要將 manage_snapshots 角色映射至 IAM 角色，以便您可以使用快照。

若要為您的身分授予許可以便使用快照

1. 使用您在建立 OpenSearch Service 網域時指定的主要使用者憑證登入 Dashboards。您可以在 OpenSearch Service 主控台中找到 Dashboards URL。其格式為 `https://domain-endpoint/_dashboards/`。
2. 從主選單中選擇 Security (安全性)、Roles (角色)，然後選取 manage_snapshots 角色。
3. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
4. 在適當的欄位中新增您的個人 IAM 角色的網域 ARN。ARN 採用下列其中一種格式：

   ```
   arn:aws:iam::123456789123:user/user-name
   ```

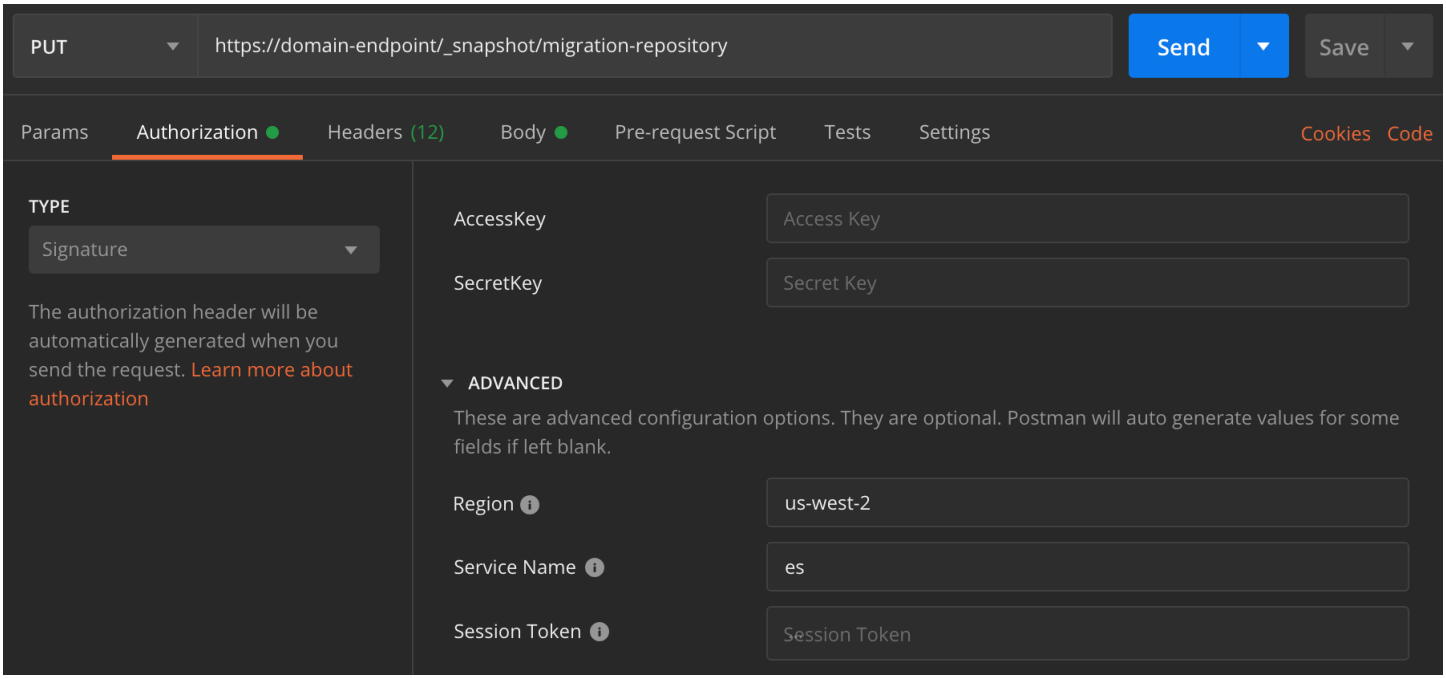
   ```
   arn:aws:iam::123456789123:role/role-name
   ```

5. 選擇 Map (映射)，並確認角色顯示在 Mapped users (已映射的使用者) 中。

還原快照

此時，您有兩種方式可存取 OpenSearch Service 網域：使用主要使用者憑證進行 HTTP 基本身分驗證，或使用 IAM 憑證進行 AWS 身分驗證。由於快照使用 Amazon S3，它沒有主要使用者的概念，因此您必須使用 IAM 憑證向您的 OpenSearch Service 網域註冊快照儲存庫。

大多數程式設計語言都有程式庫來協助簽署請求，但更簡單的方法是使用 Postman 之類的工具，並將您的 IAM 登入資料放入授權區段。



還原快照

1. 無論您選擇如何簽署請求，第一步都是註冊儲存庫：

```
PUT _snapshot/my-snapshot-repo-name
{
  "type": "s3",
  "settings": {
    "bucket": "amzn-s3-demo-bucket",
    "region": "us-west-2",
    "role_arn": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
  }
}
```

2. 然後列出儲存庫中的快照，找到您要還原的快照。此時，您可以繼續使用 Postman，或切換到 curl 等工具。

   速記

   ```
   GET _snapshot/my-snapshot-repo-name/_all
   ```

   curl

```
curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/
_snapshot/my-snapshot-repo-name/_all
```

3. 還原快照。

   速記

```
POST _snapshot/my-snapshot-repo-name/my-snapshot-name/_restore
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

   curl

```
curl -XPOST -u 'master-user:master-user-password' https://domain-endpoint/
_snapshot/my-snapshot-repo-name/my-snapshot-name/_restore \
  -H 'Content-Type: application/json' \
  -d '{"indices":"migration-index1,migration-index2,other-indices-
*","include_global_state":false}'
```

4. 最後，確認索引是否已依照預期還原。

   速記

```
GET _cat/indices?v
```

   curl

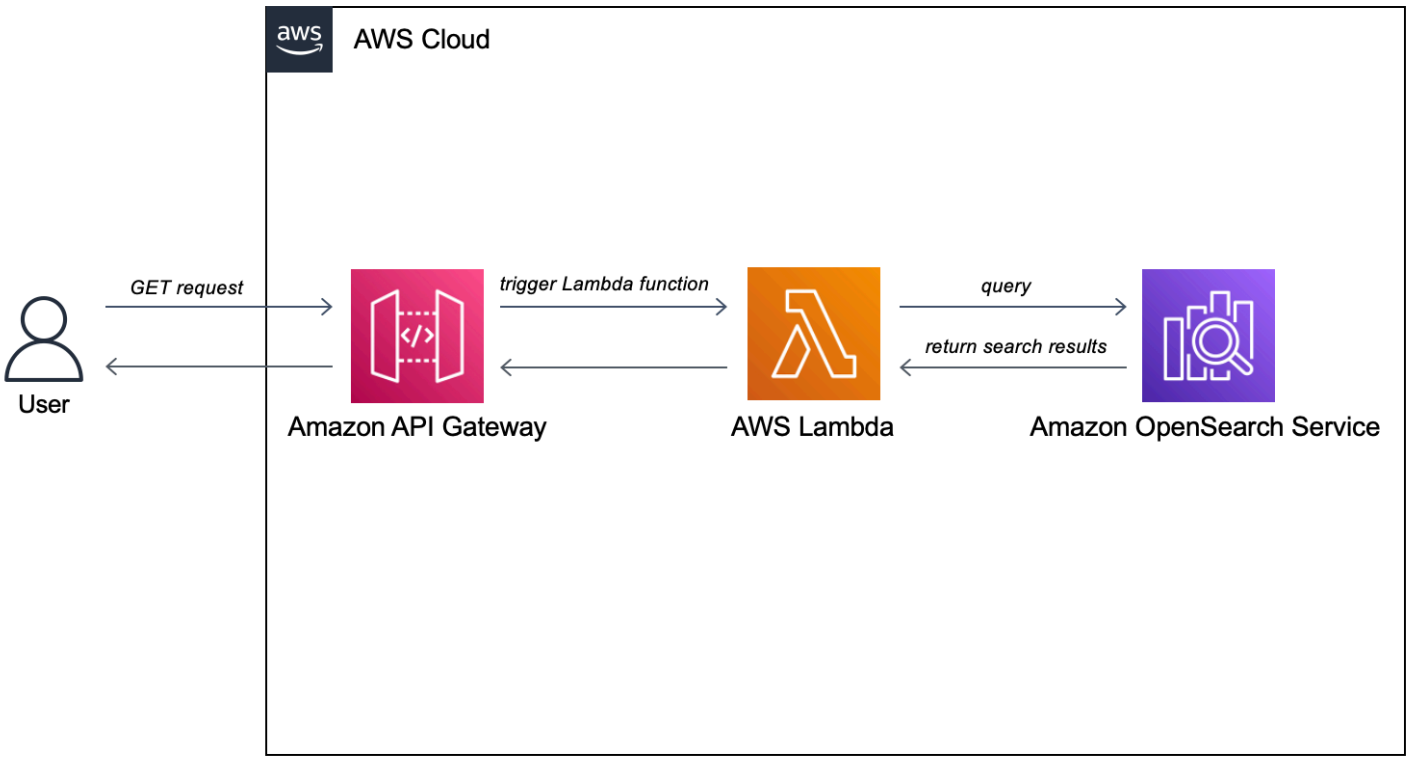
```
curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/_cat/
indices?v
```

此時，遷移即已完成。您可以設定用戶端以使用新的 OpenSearch Service 端點，調整網域大小以符合您的工作負載，檢查索引的碎片計數，切換到 IAM 主要使用者，或開始在 OpenSearch Dashboards 中建置視覺效果。

教學課程：使用 Amazon OpenSearch Service 建立一個搜尋應用程式

使用 Amazon OpenSearch Service 建立搜尋應用程式的常見方法是使用 web 表單將使用者查詢傳送到伺服器。接著，您可以授權伺服器直接呼叫 OpenSearch API 並讓伺服器將請求傳送至 OpenSearch Service。但是如果您想要編寫不依賴伺服器的用戶端程式碼，您應彌補安全性和效能風險。允許對 OpenSearch API 的未簽署、公開存取是不明智的做法。使用者可能會透過過於廣泛的查詢 (或太多查詢) 存取不安全的端點或影響叢集效能。

本章提供解決方案：使用 Amazon API Gateway 將使用者限制為 OpenSearch APIs的子集 AWS Lambda ，並簽署從 API Gateway 到 OpenSearch Service 的請求。



> **Note**
>
> 標準 API Gateway 和 Lambda 定價適用，但在此教學課程的限制使用量之內，成本應微乎其微。

先決條件

此教學的先決條件是 OpenSearch Service 網域。如果沒有，請遵循建立 OpenSearch Service 網域中的步驟建立一個。

步驟 1：索引範例資料

下載 sample-movies.zip、進行解壓縮，然後使用 _bulk API 操作來將 5,000 個文件新增到 movies 索引：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/_bulk
{ "index": { "_index": "movies", "_id": "tt1979320" } }
{"directors":["Ron
 Howard"],"release_date":"2013-09-02T00:00:00Z","rating":8.3,"genres":
["Action","Biography","Drama","Sport"],"image_url":"http://ia.media-imdb.com/images/
M/MV5BMTQyMDE0MTY0OV5BMl5BanBnXkFtZTcwMjI2OTI0OQ@@._V1_SX400_.jpg","plot":"A re-
creation of the merciless 1970s rivalry between Formula One rivals James Hunt and
 Niki Lauda.","title":"Rush","rank":2,"running_time_secs":7380,"actors":["Daniel
 Brühl","Chris Hemsworth","Olivia Wilde"],"year":2013,"id":"tt1979320","type":"add"}
{ "index": { "_index": "movies", "_id": "tt1951264" } }
{"directors":["Francis Lawrence"],"release_date":"2013-11-11T00:00:00Z","genres":
["Action","Adventure","Sci-Fi","Thriller"],"image_url":"http://ia.media-imdb.com/
images/M/
MV5BMTAyMjQ3OTAxMzNeQTJeQWpwZ15BbWU4MDU0NzA1MzAx._V1_SX400_.jpg","plot":"Katniss
 Everdeen and Peeta Mellark become targets of the Capitol after
 their victory in the 74th Hunger Games sparks a rebellion in
 the Districts of Panem.","title":"The Hunger Games: Catching
 Fire","rank":4,"running_time_secs":8760,"actors":["Jennifer Lawrence","Josh
 Hutcherson","Liam Hemsworth"],"year":2013,"id":"tt1951264","type":"add"}
...
```

請注意，上述是具有一小部分可用資料的範例命令。若要執行 _bulk操作，您需要複製並貼上sample-movies檔案的完整內容。如需進一步說明，請參閱 the section called “選項 2：上傳多個文件”。

您也可以使用下列 curl 命令來達成相同的結果：

```
curl -XPOST -u 'master-user:master-user-password' 'domain-endpoint/_bulk' --data-binary
 @bulk_movies.json -H 'Content-Type: application/json'
```

步驟 2：建立和部署 Lambda 函數

在 API Gateway 中建立 API 之前，請先建立傳遞請求的 Lambda 函數。

建立 Lambda 函式

在此解決方案中，API Gateway 會將請求傳遞至 Lambda 函數，該函數會查詢 OpenSearch Service 並傳回結果。由於此範例函數使用外部程式庫，您需要建立部署套件並將其上傳至 Lambda。

建立部署套件

1. 開啟命令提示字元並建立 `my-opensearch-function` 專案目錄。例如，在 macOS 上：

```
mkdir my-opensearch-function
```

2. 導覽至 `my-sourcecode-function` 專案目錄。

```
cd my-opensearch-function
```

3. 複製下列範例 Python 程式碼的內容，並將其儲存在名為 的新檔案中`opensearch-lambda.py`。將區域和主機端點新增至 檔案。

```
import boto3
import json
import requests
from requests_aws4auth import AWS4Auth

region = '' # For example, us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # The OpenSearch domain endpoint with https:// and without a trailing
 slash
index = 'movies'
url = host + '/' + index + '/_search'

# Lambda execution starts here
def lambda_handler(event, context):

    # Put the user query into the query DSL for more accurate search results.
```

```
    # Note that certain fields are boosted (^).
    query = {
        "size": 25,
        "query": {
            "multi_match": {
                "query": event['queryStringParameters']['q'],
                "fields": ["title^4", "plot^2", "actors", "directors"]
            }
        }
    }

    # Elasticsearch 6.x requires an explicit Content-Type header
    headers = { "Content-Type": "application/json" }

    # Make the signed HTTP request
    r = requests.get(url, auth=awsauth, headers=headers, data=json.dumps(query))

    # Create the response and add some extra content to support CORS
    response = {
        "statusCode": 200,
        "headers": {
            "Access-Control-Allow-Origin": '*'
        },
        "isBase64Encoded": False
    }

    # Add the search results to the response
    response['body'] = r.text
    return response
```

4. 將外部程式庫安裝到新的package目錄。

```
pip3 install --target ./package boto3
pip3 install --target ./package requests
pip3 install --target ./package requests_aws4auth
```

5. 在根目錄中使用已安裝的程式庫建立部署套件。下列命令會在您的專案目錄中產生my-deployment-package.zip檔案。

```
cd package
zip -r ../my-deployment-package.zip .
```

6. 將 opensearch-lambda.py 檔案新增至 zip 檔案的根目錄。

```
cd ..
zip my-deployment-package.zip opensearch-lambda.py
```

如需建立 Lambda 函數和部署套件的詳細資訊，請參閱 AWS Lambda 開發人員指南中的使用 .zip 封存檔部署 Python Lambda 函數以及本指南中的 the section called “建立 Lambda 部署套件”。

使用 Lambda 主控台建立函數

1. 導覽至位於 https：//https://console.aws.amazon.com/lambda/home 的 Lambda 主控台。在左側導覽窗格中，選擇函數。
2. 選取 Create function (建立函式)。
3. 設定下列欄位：

   - 函數名稱：openearch-function
   - 執行時間：Python 3.9
   - 架構：x86_64

   保留所有其他預設選項，然後選擇建立函數。
4. 在函數摘要頁面的程式碼來源區段中，選擇從下拉式清單上傳，然後選取 .zip 檔案。找到您建立`my-deployment-package.zip`的檔案，然後選擇儲存。
5. 處理常式是函數程式碼中處理事件的方法。在執行時間設定下，選擇編輯，並根據 Lambda 函數所在的部署套件中的檔案名稱變更處理常式名稱。由於您的檔案名為 `opensearch-lambda.py`，請將處理常式重新命名為 *`opensearch-lambda`*`.lambda_handler`。如需詳細資訊，請參閱 Python 中的 Lambda 函數處理常式。

步驟 3：在 API Gateway 中建立 API

使用 API Gateway 來建立更為有限的 API 並簡化與 OpenSearch `_search` API 互動的程序。API Gateway 可啟用安全性功能，例如 Amazon Cognito 身分驗證和請求調節。請執行下列步驟，來建立和部署 API：

建立和設定 API

若要使用 API Gateway 主控台建立 API

1. 導覽至 API Gateway 主控台，網址為 https：//https://console.aws.amazon.com/apigateway/home。在左側導覽窗格中，選擇 APIs。
2. 找到 REST API (非私有) 並選擇 Build (建置)。
3. 在下列頁面上，找到建立新 API 區段，並確認已選取新 API。
4. 設定下列欄位：
   - API 名稱：opensearch-api
   - 描述：搜尋 Amazon OpenSearch Service 網域的公有 API
   - 端點類型：區域
5. 選擇建立 API。
6. 選擇 Actions (動作) 和 Create Method (建立方法)。
7. 在下拉式選單中選擇 GET，然後按一下核取記號以確認。
8. 進行下列設定，然後選擇 Save (儲存)：

設定	Value
整合類型	Lambda 函式
使用 Lambda 代理整合	是
Lambda 區域	*us-west-1*
Lambda 函式	opensearch-lambda
使用預設逾時	是

設定方法請求

選擇 Method Request (方法請求)，然後進行下列設定：

設定	Value
Authorization	NONE
請求驗證程式	驗證查詢字串參數與標頭

設定	Value
需要 API 金鑰	false

在 URL 查詢字串參數下，選擇新增查詢字串並設定下列參數：

設定	Value
名稱	q
必要	是

部署 API 並設定階段

API Gateway 主控台可讓您透過建立部署並將它與全新或現有階段建立關聯來部署 API。

1. 選擇 Actions (動作) 和 Deploy API (部署 API)。
2. 對於 Deployment stage (部署階段)，選擇 New Stage (新增階段) 並將階段命名為 `opensearch-api-test`。
3. 選擇部署。
4. 在階段編輯器中進行下列設定，然後選擇 Save Changes (儲存變更)：

設定	Value
啟用調節	是
速率	1000
爆量	500

這些設定會設定僅有一個方法的 API：GET 對端點根進行要求 (`https://some-id.execute-api.us-west-1.amazonaws.com/search-es-api-test`)。要求需要單一參數 (q)，要搜尋的查詢字串。呼叫時，方法會將請求傳遞至 Lambda，它會執行 `opensearch-lambda` 函數。如需詳細資訊，請參閱在 Amazon API Gateway 中建立 API 和在 Amazon API Gateway 中部署 REST API。

步驟 4：(選用) 修改網域存取政策

OpenSearch Service 網域必須允許 Lambda 函數對 `movies` 索引進行 `GET` 請求。如果您的網域具有啟用了精細存取控制的開放存取政策，可以保持原樣：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

或者，您可以選擇使網域存取政策更精細。例如，下列最低政策提供對 `movies` 索引的 `opensearch-lambda-role` (透過 Lambda 建立) 讀取存取權：若要取得 Lambda 自動建立之角色的確切名稱，請前往 AWS Identity and Access Management (IAM) 主控台，選擇角色，然後搜尋「lambda」。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/service-role/opensearch-lambda-role-1abcdefg"
      },
      "Action": "es:ESHttpGet",
```

```
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/movies/
_search"
    }
  ]
}
```

> **Important**
>
> 如果您為網域啟用了精細存取控制，則還需要在 OpenSearch Dashboards 中將角色映射至使用者，否則您將看到許可錯誤。

設定 Lambda 執行角色許可

除了設定網域存取政策之外，您還必須確保 Lambda 執行角色具有存取 OpenSearch Service 網域所需的 IAM 許可。Lambda 函數需要特定許可，取決於您使用的是受管網域或 OpenSearch Service Serverless 集合。

對於受管 OpenSearch Service 網域：

將下列 IAM 政策連接至您的 Lambda 執行角色，以允許其向 OpenSearch Service 網域提出請求：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPost"
      ],
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

對於 OpenSearch Service Serverless 集合：

如果您使用的是 OpenSearch Service Serverless，請將下列 IAM 政策連接至 Lambda 執行角色：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aoss:*",
      "Resource": "arn:aws:aoss:us-west-1:123456789012:collection/collection-id"
    }
  ]
}
```

若要將這些政策連接至 Lambda 執行角色：

1. 導覽至 IAM 主控台，網址為 https://console.aws.amazon.com/iam/。
2. 選擇角色並搜尋您的 Lambda 執行角色（通常名為 `opensearch-lambda-role-xxxxxxxx`）。
3. 選擇新增許可，然後選擇建立內嵌政策。
4. 選擇 JSON 索引標籤，然後從上方貼上適當的政策，將預留位置值取代為實際的資源 ARNs。
5. 選擇檢閱政策，提供類似 的名稱`OpenSearchAccess`，然後選擇建立政策。

> **Note**
>
> 如果沒有這些 IAM 許可，即使網域存取政策允許請求，您的 Lambda 函數也會在嘗試查詢 OpenSearch Service 網域時收到「存取遭拒」錯誤。

如需存取政策的詳細資訊，請參閱the section called “設定存取政策”。

映射 Lambda 角色 (如果使用精細存取控制)

精細存取控制會在您可以測試應用程式之前引進額外的步驟。即使您將 HTTP 基本身分驗證用於所有其他目的，您也需要將 Lambda 角色映射至使用者，否則您會看到許可錯誤。

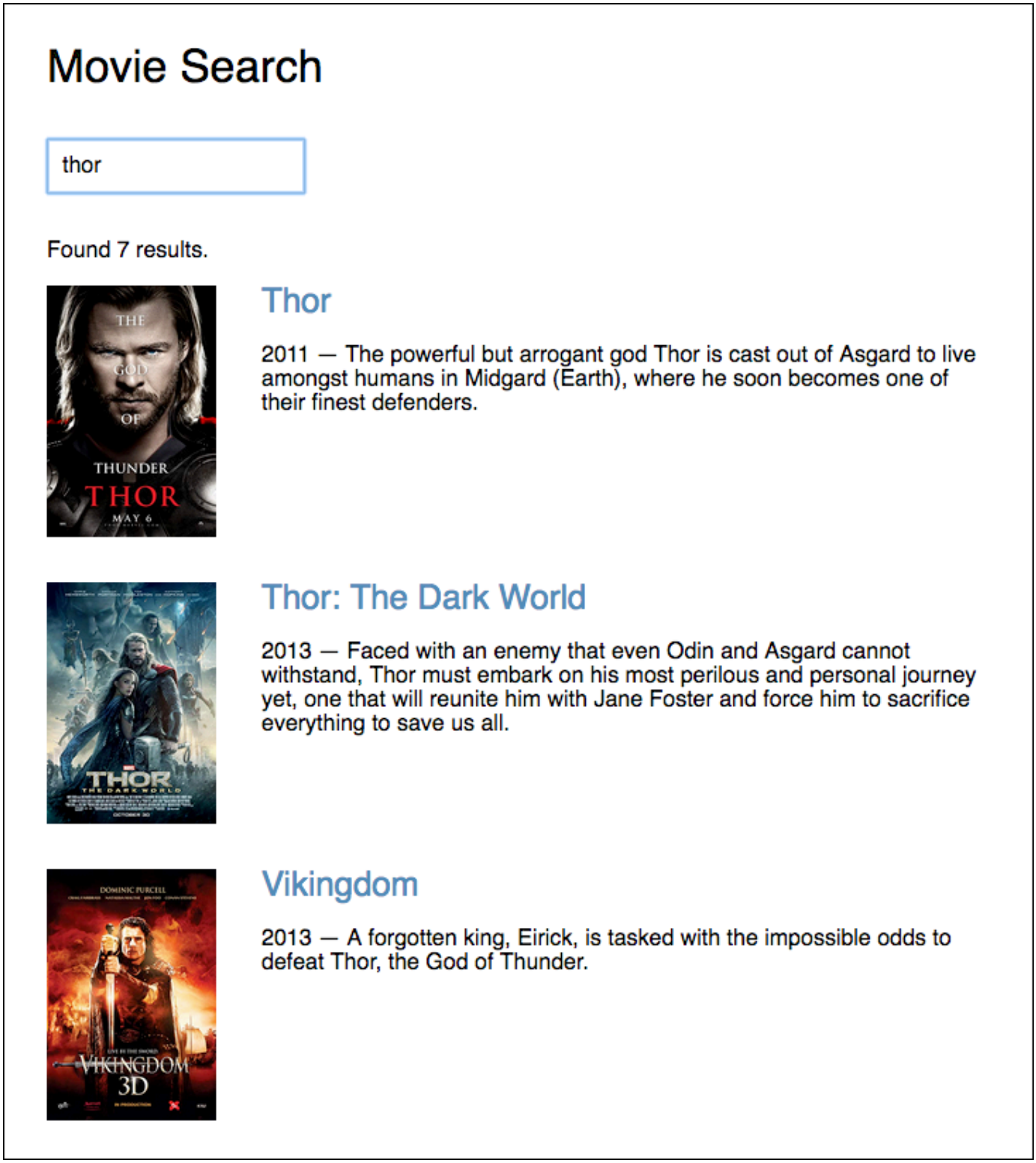
1. 導覽至網域的 OpenSearch Dashboards URL。

2. 從主功能表中，選擇安全性、角色，然後選取 的連結`all_access`，這是您需要映射 Lambda 角色的角色。
3. 選擇 Mapped users (已映射的使用者)、Manage mapping (管理映射)。
4. 在 Backend roles (後端角色) 下方，新增 Lambda 角色的 Amazon Resource Name (ARN)。ARN 應該採用 的形式`arn:aws:iam::`*`123456789123`*`:role/service-role/`*`opensearch-lambda-role-1abcdefg`*。
5. 選擇 Map (映射)，並確認使用者或角色顯示在 Mapped users (已映射的使用者) 中。

步驟 5：測試 Web 應用程式

若要測試 web 應用程式

1. 下載 sample-site.zip，將其解壓縮並使用您最愛的文字編輯器將 `scripts/search.js` 開啟。
2. 更新 `apigatewayendpoint`變數以指向您的 API Gateway 端點，並將反斜線新增至指定路徑的結尾。您可以透過選擇 Stages (階段)，然後選取 API 的名稱，在 API Gateway 中快速找到端點。`apigatewayendpoint` 變數應該採用 `https://`*`some-id`*`.execute-api.`*`us-west-1`*`.amazonaws.com/opensearch-api-test/` 的形式。
3. 開啟 `index.html` 並嘗試對 thor、house 和一些其他術語執行搜尋。



對 CORS 錯誤進行疑難排解

即使 Lambda 函數在回應中包含支援 CORS 的內容，仍可能會看到以下錯誤：

```
Access to XMLHttpRequest at '<api-gateway-endpoint>' from origin 'null' has been
 blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present in the
 requested resource.
```

如果發生此情況，請嘗試下列操作：

1. 在 GET 資源上啟用 CORS。在 Advanced (進階) 下，將 Access-Control-Allow-Credentials 設定為 `'true'`。
2. 在 API Gateway 中重新部署 API (Actions (動作)、Deploy API (部署 API))。
3. 刪除並重新新增您的 Lambda 函數觸發程序。新增重新新增，選擇新增觸發並建立叫用函數的 HTTP 端點。觸發必須具有以下組態：

觸發條件	API	部署階段	安全
API Gateway	opensearch-api	opensearch-api-test	開啟

後續步驟

本章只是展縣概念的起點。您可以考慮以下修改：

- 將自己的資料新增到 OpenSearch Service 網域。
- 將方法新增至 API
- 在 Lambda 函數中，修改搜尋查詢或提升不同的欄位。
- 樣式結果不同或修改 `search.js` 來向使用者顯示不同的欄位。

教學課程：使用 OpenSearch Service 和 OpenSearch Dashboards，將客戶支援呼叫視覺化

此章節是一個完整的逐步解說，情況如下：業務收到不少客戶支援呼叫，想要對其進行分析。每個呼叫的主題為何？ 有多少是正面的？ 有多少是負面的？ 主管人員如何搜尋或檢閱這些呼叫的文字記錄？

手動工作流程可能涉及員工聆聽錄音、記下每個呼叫的主題，以及判斷客戶互動是否正面。

這類的程序是非常費工的。假設每個呼叫平均時間為 10 分鐘，那麼每個員工每天只能聆聽 48 個呼叫。排除人的偏見後，所產生的資料使有極高的準確度，而資料量會降到最低：只需呼叫的主題，以及有關客戶是否滿意的布林值。若再加上更多資料如完整的文字記錄，便需要花大量時間。

使用 Amazon S3、Amazon Transcribe、Amazon Comprehend 和 Amazon OpenSearch Service，便能以極少的程式碼自動化類似的程序，所產生的資料量非常多。例如，您可以取得呼叫的完整文字記錄、文字記錄的關鍵字和呼叫的整體「情緒」(正面、負面、中立或混合)。然後，您可以使用 OpenSearch 和 OpenSearch Dashboards 來搜尋和視覺化資料。

您可以一字不漏地遵照此逐步解說來執行，用意只是為了讓您在 OpenSearch Service 中編製索引之前，激盪出有關如何使 JSON 文件更加豐富的想法。

評估成本

一般而言，執行這個逐步解說中的步驟的成本不到 2 美元。此逐步解說使用下列資源：

- S3 儲存貯體的傳輸和儲存量少於 100 MB

  如需進一步了解，請參閱 Amazon S3 定價。
- 具有一個 `t2.medium` 執行個體和 10 GiB 的 EBS 儲存空間，運作長達數小時的 OpenSearch Service 網域

  如需進一步了解，請參閱 Amazon OpenSearch Service 定價。
- 對 Amazon Transcribe 的數個呼叫

  如需進一步了解，請參閱 Amazon Transcribe 定價。
- 對 Amazon Comprehend 的數個自然語言處理呼叫

  如需進一步了解，請參閱 Amazon Comprehend 定價。

主題

- 步驟 1：設定先決條件
- 步驟 2：複製範本程式碼
- (選用) 步驟 3：索引範例資料
- 步驟 4：分析和視覺化您的資料
- 步驟 5：清除資源和後續步驟

步驟 1：設定先決條件

繼續之前，您必須準備好以下資源。

先決條件	Description
Amazon S3 儲存貯體	如需詳細資訊，請參閱 Amazon Simple Storage Service 主控台使用者指南中的建立儲存貯體。
OpenSearch Service 網域	資料的目的地。如需詳細資訊，請參閱建立 OpenSearch Service 網域。

如果尚未擁有這些資源，您可以使用下列 AWS CLI 命令建立資源：

```
aws s3 mb s3://my-transcribe-test --region us-west-2
```

```
aws opensearch create-domain --domain-name my-transcribe-test --engine-version
 OpenSearch_1.0 --cluster-config  InstanceType=t2.medium.search,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 --access-policies
 '{"Version": "2012-10-17",       "Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::123456789012:root"},"Action":"es:*","Resource":"arn:aws:es:us-
west-2:123456789012:domain/my-transcribe-test/*"}]}' --region us-west-2
```

> **Note**
>
> 這些命令使用 us-west-2 區域，但您可以使用 Amazon Comprehend 支援的任何區域。如需進一步了解，請參閱AWS 一般參考。

步驟 2：複製範本程式碼

1. 複製下列 Python 3 範本程式碼，並貼上到名為 call-center.py 的新檔案：

```
import boto3
import datetime
import json
import requests
from requests_aws4auth import AWS4Auth
import time
import urllib.request

# Variables to update
```

```
audio_file_name = '' # For example, 000001.mp3
bucket_name = '' # For example, my-transcribe-test
domain = '' # For example, https://search-my-transcribe-test-12345.us-
west-2.es.amazonaws.com
index = 'support-calls'
type = '_doc'
region = 'us-west-2'

# Upload audio file to S3.
s3_client = boto3.client('s3')

audio_file = open(audio_file_name, 'rb')

print('Uploading ' + audio_file_name + '...')
response = s3_client.put_object(
    Body=audio_file,
    Bucket=bucket_name,
    Key=audio_file_name
)

# # Build the URL to the audio file on S3.
# # Only for the us-east-1 region.
# mp3_uri = 'https://' + bucket_name + '.s3.amazonaws.com/' + audio_file_name

# Get the necessary details and build the URL to the audio file on S3.
# For all other regions.
response = s3_client.get_bucket_location(
    Bucket=bucket_name
)
bucket_region = response['LocationConstraint']
mp3_uri = 'https://' + bucket_name + '.s3-' + bucket_region + '.amazonaws.com/' +
 audio_file_name

# Start transcription job.
transcribe_client = boto3.client('transcribe')

print('Starting transcription job...')
response = transcribe_client.start_transcription_job(
    TranscriptionJobName=audio_file_name,
    LanguageCode='en-US',
    MediaFormat='mp3',
    Media={
        'MediaFileUri': mp3_uri
    },
```

```
    Settings={
        'ShowSpeakerLabels': True,
        'MaxSpeakerLabels': 2 # assumes two people on a phone call
    }
)

# Wait for the transcription job to finish.
print('Waiting for job to complete...')
while True:
    response =
 transcribe_client.get_transcription_job(TranscriptionJobName=audio_file_name)
    if response['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    else:
        print('Still waiting...')
    time.sleep(10)

transcript_uri = response['TranscriptionJob']['Transcript']['TranscriptFileUri']

# Open the JSON file, read it, and get the transcript.
response = urllib.request.urlopen(transcript_uri)
raw_json = response.read()
loaded_json = json.loads(raw_json)
transcript = loaded_json['results']['transcripts'][0]['transcript']

# Send transcript to Comprehend for key phrases and sentiment.
comprehend_client = boto3.client('comprehend')

# If necessary, trim the transcript.
# If the transcript is more than 5 KB, the Comprehend calls fail.
if len(transcript) > 5000:
    trimmed_transcript = transcript[:5000]
else:
    trimmed_transcript = transcript

print('Detecting key phrases...')
response = comprehend_client.detect_key_phrases(
    Text=trimmed_transcript,
    LanguageCode='en'
)

keywords = []
for keyword in response['KeyPhrases']:
```

```
    keywords.append(keyword['Text'])

print('Detecting sentiment...')
response = comprehend_client.detect_sentiment(
    Text=trimmed_transcript,
    LanguageCode='en'
)

sentiment = response['Sentiment']

# Build the Amazon OpenSearch Service URL.
id = audio_file_name.strip('.mp3')
url = domain + '/' + index + '/' + type + '/' + id

# Create the JSON document.
json_document = {'transcript': transcript, 'keywords': keywords, 'sentiment':
 sentiment, 'timestamp': datetime.datetime.now().isoformat()}

# Provide all details necessary to sign the indexing request.
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region,
 'opensearchservice', session_token=credentials.token)

# Index the document.
print('Indexing document...')
response = requests.put(url, auth=awsauth, json=json_document, headers=headers)

print(response)
print(response.json())
```

2. 更新初始六個變數。
3. 使用以下命令安裝所需套件：

```
pip install boto3
pip install requests
pip install requests_aws4auth
```

4. 將 MP3 放置在和 `call-center.py` 相同的目錄，並執行指令碼。範例輸出如下：

```
$ python call-center.py
Uploading 000001.mp3...
Starting transcription job...
Waiting for job to complete...
```

```
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Detecting key phrases...
Detecting sentiment...
Indexing document...
<Response [201]>
{u'_type': u'call', u'_seq_no': 0, u'_shards': {u'successful': 1, u'failed': 0,
 u'total': 2}, u'_index': u'support-calls4', u'_version': 1, u'_primary_term': 1,
 u'result': u'created', u'_id': u'000001'}
```

`call-center.py` 執行一些操作：

1. 指令碼上傳音訊檔案 (此案例為，MP3，但 Amazon Transcribe 支援數種格式) 到您的 S3 儲存貯體。
2. 它會將音訊檔案的 URL 傳送到 Amazon Transcribe，並等待文字記錄任務完成。

   完成文字記錄工作的時間取決於音訊檔案長度。採用分鐘數，而非秒數。

   > **Tip**
   >
   > 為了改善文字記錄的品質，您可以設定 Amazon Transcribe 適用的自訂詞彙。

3. 文字記錄任務完成後，指令碼會擷取文字，裁剪到 5,000 個字元，然後將其傳送至 Amazon Comprehend 進行關鍵字和情感分析。
4. 最後，指令碼會新增完整的文字記錄、關鍵字、情感及目前時間戳記至 JSON 文件，並在 OpenSearch Service 中編製索引。

> **Tip**
>
> LibriVox 已有公有網域有聲書可進行測試。

(選用) 步驟 3：索引範例資料

如果您手邊沒有大量呼叫記錄 (誰有呢)？您可以在 sample-calls.zip 中為範例文件編製索引，這與 `call-center.py` 所產生的結果相當。

1. 建立名為 `bulk-helper.py` 的檔案：

```
import boto3
from opensearchpy import OpenSearch, RequestsHttpConnection
import json
from requests_aws4auth import AWS4Auth

host = '' # For example, my-test-domain.us-west-2.es.amazonaws.com
region = '' # For example, us-west-2
service = 'es'

bulk_file = open('sample-calls.bulk', 'r').read()

credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

search = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    connection_class = RequestsHttpConnection
)

response = search.bulk(bulk_file)
print(json.dumps(response, indent=2, sort_keys=True))
```

2. 更新 `host` 和 `region` 兩個初始變數。
3. 使用以下命令安裝所需套件：

```
pip install opensearch-py
```

4. 下載並解壓縮 sample-calls.zip。
5. 將 `sample-calls.bulk` 放置在和 `bulk-helper.py` 相同的目錄，並執行協助程式。範例輸出如下：

```
$ python bulk-helper.py
{
  "errors": false,
  "items": [
    {
      "index": {
        "_id": "1",
        "_index": "support-calls",
        "_primary_term": 1,
        "_seq_no": 42,
        "_shards": {
          "failed": 0,
          "successful": 1,
          "total": 2
        },
        "_type": "_doc",
        "_version": 9,
        "result": "updated",
        "status": 200
      }
    },
    ...
  ],
  "took": 27
}
```

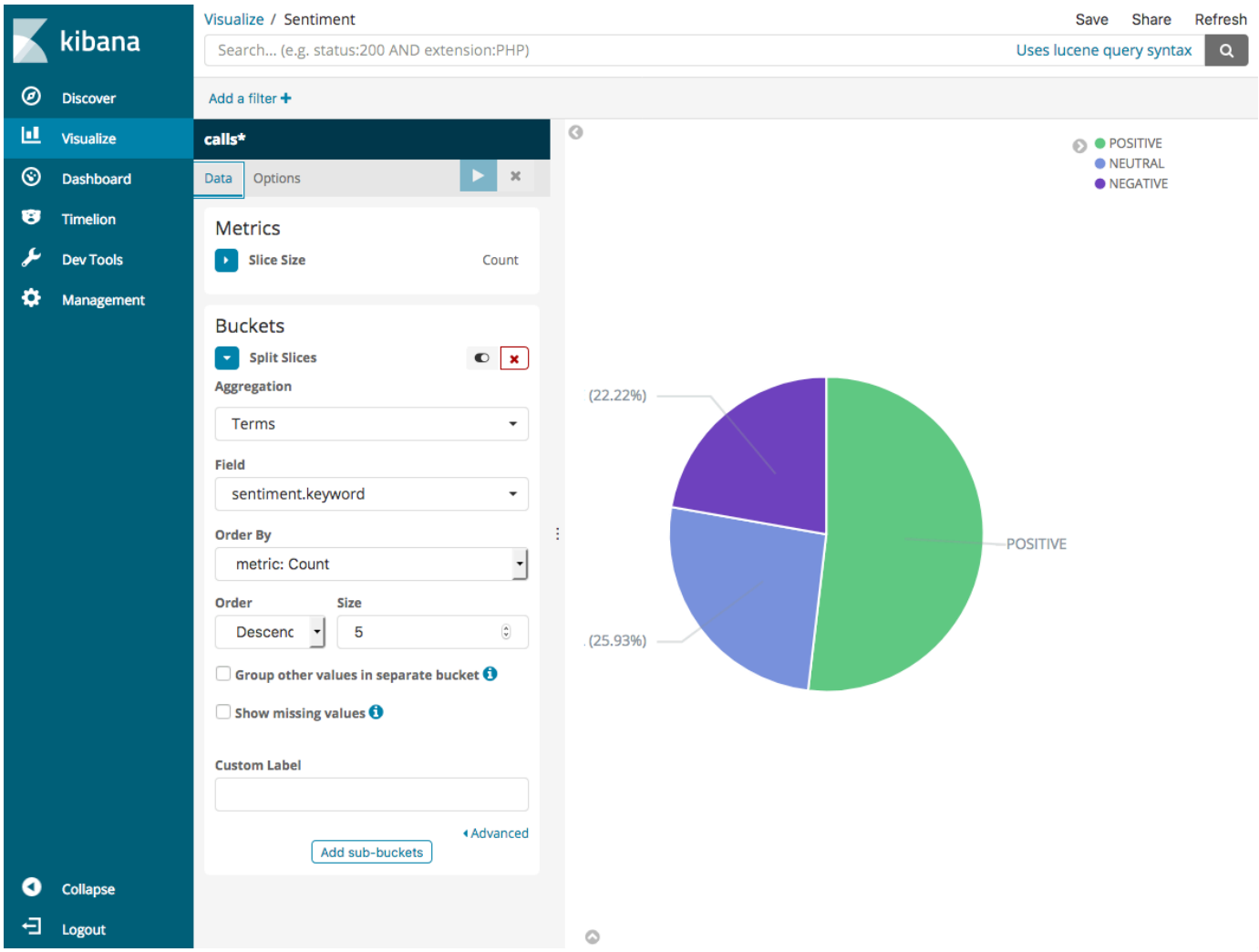
步驟 4：分析和視覺化您的資料

既然您在 OpenSearch Service 中有一些資料，您可以使用 OpenSearch Dashboards 將資料視覺化。

1. 導覽至 `https://search-domain.region.es.amazonaws.com/_dashboards`。
2. 在使用 OpenSearch Dashboards 之前，您需要索引模式。Dashboards 使用索引模式將分析縮小至一個或多個索引。若要匹配 `call-center.py` 建立的 `support-calls` 索引，請前往 Stack Management (堆疊管理)、Index Patterns (索引模式)，並定義 `support*` 的索引模式，然後選擇 Next step (下一個步驟)。
3. 對於 Time Filter field name (時間篩選條件欄位名稱)，請選擇 timestamp (時間戳記)。
4. 您現在可以開始建立視覺化效果。選擇 Visualize (視覺化)，然後新增視覺化。
5. 選擇圓餅圖和 `support*` 索引模式。

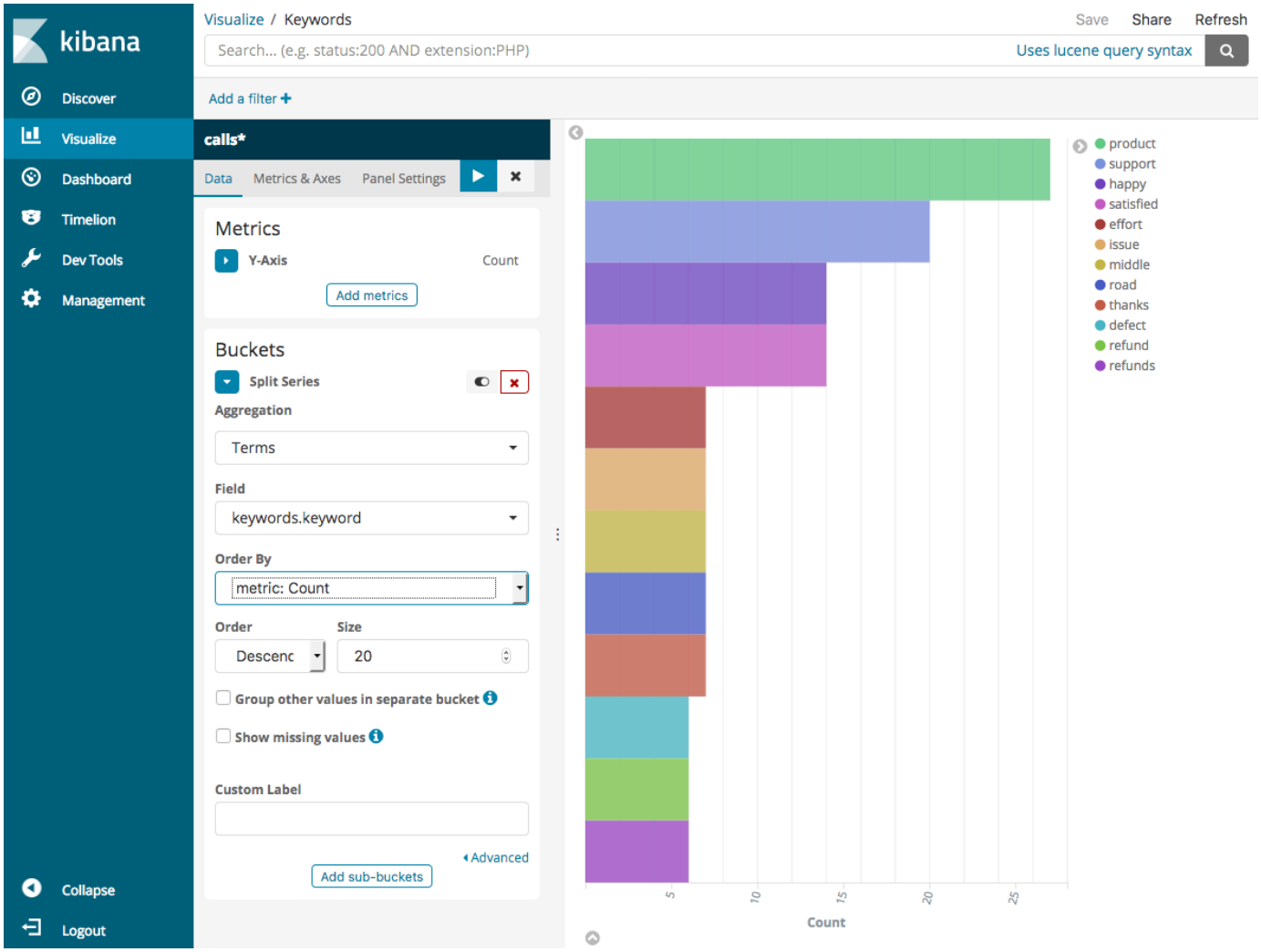
6. 預設視覺化是基本的，因此選擇 Split Slices (分割切片) 來建立更有趣的視覺化效果。

   對於 Aggregation (集合) 選項，請選擇 Terms (條款)。對於 Field (欄位)，選擇 sentiment.keyword。然後選擇 Apply changes (套用變更) 和 Save (儲存)。

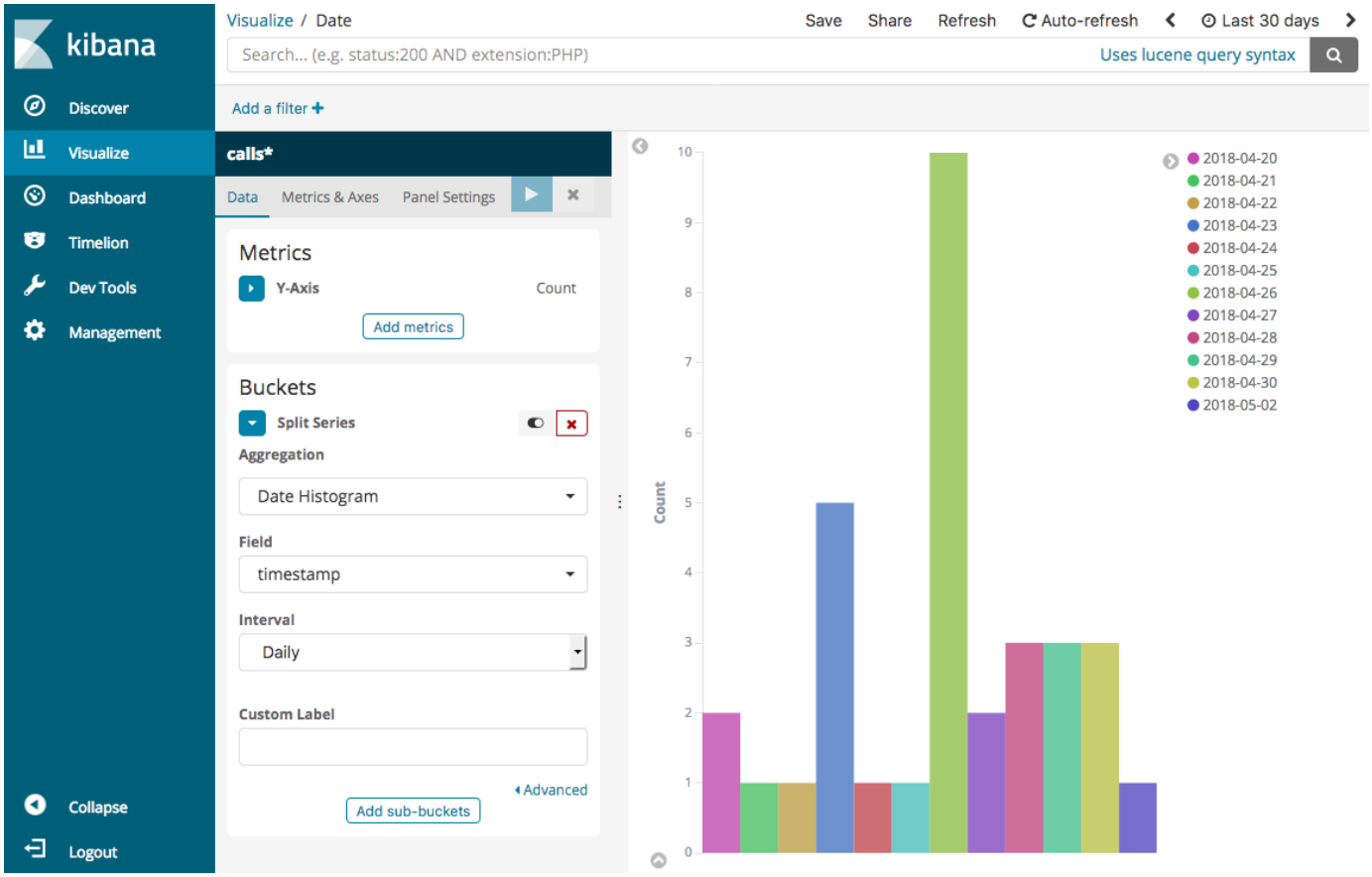


7. 返回 Visualize (視覺化) 頁面，並新增另一個視覺化。這時，選擇水平長條圖。
8. 選擇 Split Series (分割系列)。

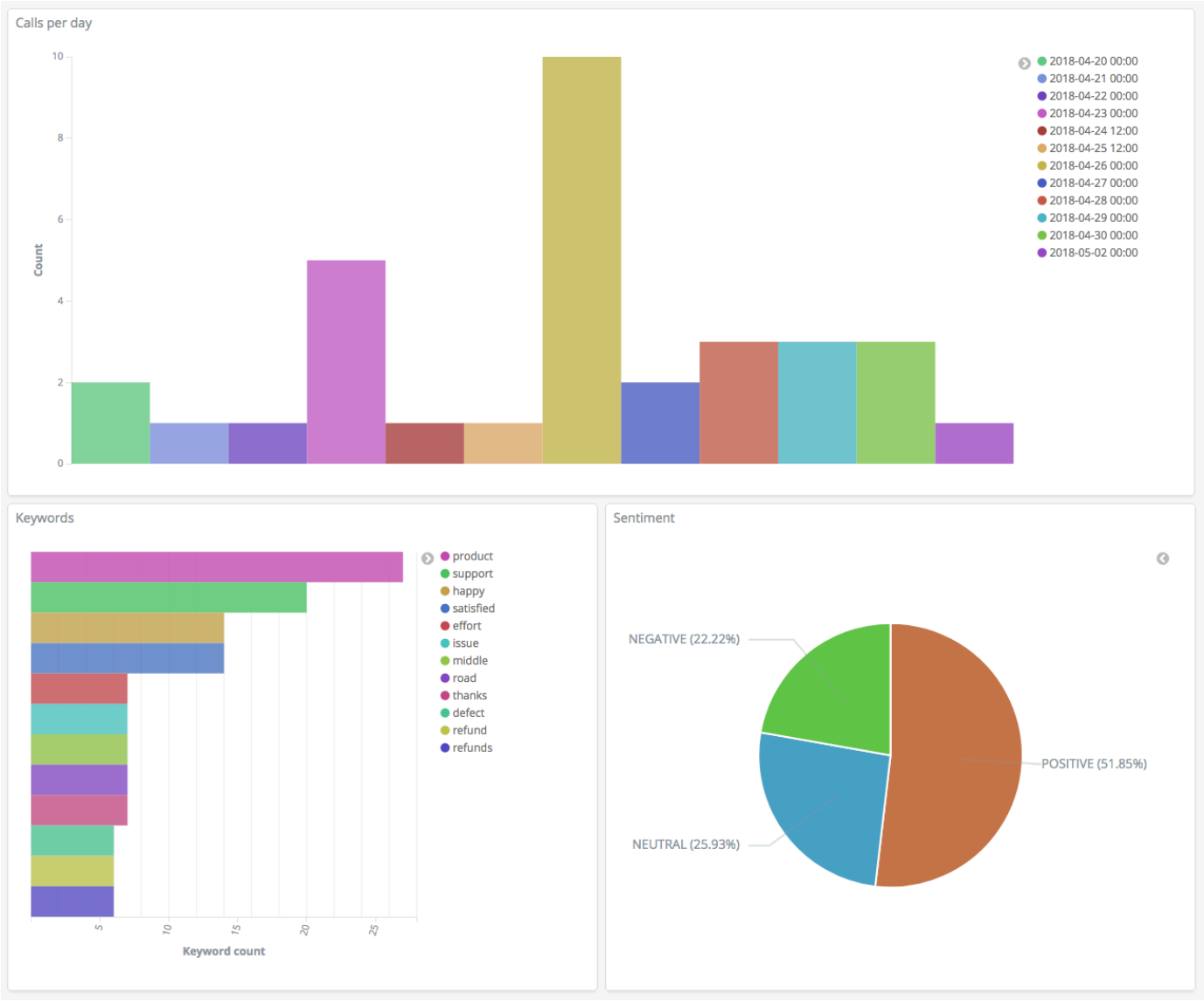
   對於 Aggregation (集合) 選項，請選擇 Terms (條款)。對於 Field (欄位)，選擇 keywords.keyword，並將 Size (大小) 變更為 20。然後選擇 Apply Changes (套用變更) 和 Save (儲存)。



9. 返回 Visualize (視覺化) 頁面，並新增一個最終視覺化垂直長條圖。
10. 選擇 Split Series (分割系列)。對於 Aggregation (彙總)，選擇 Date Histogram (日期分佈圖)。對於 Field (欄位)，選擇 timestamp (時間戳記)，將 Interval (間隔) 變更為 Daily (每日)。
11. 選擇 Metrics & Axes (指標和軸)，並將變更 Mode (模式) 變更為 normal (正常)。
12. 選擇 Apply Changes (套用變更) 和 Save (儲存)。



13. 既然您有三個視覺效果，您可以將它們新增到 Dashboards 視覺效果。選擇 Dashboard (儀表板)、建立儀表板，並新增視覺化效果。



步驟 5：清除資源和後續步驟

為避免不必要的費用，請刪除 S3 儲存貯體和 OpenSearch Service 網域。如需進一步了解，請參閱 Amazon Simple Storage Service 使用者指南中的刪除儲存貯體以及本指南中的刪除 OpenSearch Service 網域。

文字記錄需要比 MP3 檔更少的磁碟空間。您也許可以縮短您的 MP3 保留時段 (例如，從三個月的呼叫記錄到一個月)，保留數年文字記錄，並且仍能節省儲存成本。

您也可以使用 AWS Step Functions 和 Lambda 自動化轉錄程序、在編製索引之前新增其他中繼資料，或製作更複雜的視覺化效果，以符合您的確切使用案例。

Amazon OpenSearch Service 重新命名：變更摘要

2021 年 9 月 8 日，我們的搜尋和分析套件更名為 Amazon OpenSearch Service。OpenSearch Service 支援 OpenSearch 以及舊版 Elasticsearch OSS。下列各節說明由於重新命名而變更的服務的不同部分，以及您需要採取哪些動作，從而確保您的網域可繼續正常運作。

當您將網域從 Elasticsearch 升級至 OpenSearch 時，一些變更才適用。在其他情況下，例如，在「帳單和成本管理」主控台中，體驗會立即變化。

注意，此清單並不詳盡。雖然產品的其他部分也有所變更，但這些更新是最相關的。

新的 API 版本

新版 OpenSearch Service 組態 API (2021-01-01) 可與 OpenSearch 及舊版 Elasticsearch OSS 搭配使用。21 個 API 操作被替換為更簡潔且與引擎無關的名稱 (例如，`CreateElasticsearchDomain` 變更為 `CreateDomain`)，但 OpenSearch Service 會繼續支援這兩個 API 版本。

建議您使用新的 API 操作來建立和管理未來的網域。請注意，當您使用新的 API 操作來建立網域時，您需要指定 `EngineVersion` 參數，格式為 `Elasticsearch_X.Y` 或 `OpenSearch_X.Y`，而不僅僅是版本號。如果您不指定版本，它會預設為 OpenSearch 的最新版本。

將您的 升級至 1.20.40 AWS CLI 版或更新版本，以使用 `aws opensearch ...` 來建立和管理網域。如需新的 CLI 格式，請參閱 OpenSearch CLI 參考。

已重新命名的執行個體類型

Amazon OpenSearch Service 中的執行個體類型現在的格式為 `<type>.<size>.search`， 例如為 `m6g.large.search` 而非 `m6g.large.elasticsearch`。您無需執行任何動作。現有網域會開始自動參考 API 內以及 Billing and Cost Management 主控台中的新執行個體類型。

如果您有預留執行個體 (RI)，您的合約不會受到變更的影響。舊的組態 API 版本仍然與舊的命名格式相容，但如果您想要使用新的 API 版本，則需要使用新的格式。

存取政策變更

下列各節說明您需要採取哪些動作來更新存取政策。

IAM 政策

我們建議您更新 IAM 政策以使用已重新命名的 API 操作。不過，OpenSearch Service 透過內部複寫舊的 API 許可，繼續遵守現有的政策。例如，如果您目前擁有執行 `CreateElasticsearchDomain` 操作的許可，您現在可以呼叫 `CreateElasticsearchDomain` (舊的 API 操作) 和 `CreateDomain` (新的 API 操作)。這同樣適用於明確拒絕。如需已更新的 API 操作清單，請參閱政策元素參考。

SCP 政策

服務控制政策 (SCP)與標準 IAM 相比，增添另一層複雜性。若要防止您的 SCP 政策中斷，您需要將舊的和新的 API 操作新增至每個 SCP 政策。例如，如果使用者目前擁有 `CreateElasticsearchDomain` 的允許許可，您還需要授予他們 `CreateDomain` 的允許許可，以便他們能夠保留建立網域的能力。這同樣適用於明確拒絕。

例如：

```
"Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:CreateElasticsearchDomain",
        "es:CreateDomain"
         ...
      ],
    },
      "Effect": "Deny",
      "Action:" [
        "es:DeleteElasticsearchDomain",
        "es:DeleteDomain"
         ...
```

新資源類型

OpenSearch Service 引入下列新資源類型：

資源	Description
`AWS::OpenSearchService::Domain`	表示 Amazon OpenSearch Service 網域。此資源存在於服務層級，並非特定於網域上執行的軟體。它適用於諸如 CloudFormation 和 AWS

資源	Description
	Resource Groups 等服務，您可以在其中建立和管理整個服務的資源。 如需將 CloudFormation 中定義的網域從 Elasticsearch 升級到 OpenSearch 的說明，請參閱《CloudFormation 使用者指南》中的備註。
`AWS::OpenSearch::Domain`	表示在網域上執行的 OpenSearch/Elasticsearch 軟體。此資源適用於 AWS CloudTrail 和 AWS Config 等服務，其參考在網域上執行的軟體而非整個 OpenSearch Service。這些服務現在包含執行 Elasticsearch 的網域 (`AWS::Elasticsearch::Domain`) 與執行 OpenSearch 的網域 (`AWS::OpenSearch::Domain`) 的個別資源類型。

> **Note**
>
> 在 AWS Config 中，在數周內在現有 `AWS::Elasticsearch::Domain` 資源類型下可繼續看到您的資料，即使您將一個或多個網域升級至 OpenSearch。

Kibana 已更名為 OpenSearch Dashboards

OpenSearch Dashboards 是 Kibana 的 AWS 替代方案，是一種開放原始碼視覺化工具，旨在與 OpenSearch 搭配使用。將網域從 Elasticsearch 升級至 OpenSearch 之後，`/_plugin/kibana` 端點變更為 `/_dashboards`。OpenSearch Service 會將所有請求重新導向至新端點，但如果您在任何 IAM 政策中使用 Kibana 端點，請更新這些政策以包含新的 `/_dashboards` 端點。

如果您使用的是 the section called “OpenSearch Dashboards 的 SAML 身分驗證”，在將網域升級至 OpenSearch 之前，您必須將身分提供者 (IdP) 中設定的所有 Kibana URL 從 `/_plugin/kibana` 變更為 `/_dashboards`。最常見的 URL 是聲明消費者服務 (ACS) URL 和收件人 URL。

預設的 `kibana_read_only` 角色已重新命名為 `opensearch_dashboards_read_only`，`kibana_user` 角色已重新命名為

opensearch_dashboards_user。此變更適用於所有新建立的 OpenSearch 1.x 網域，其執行服務軟體 R20211203 或更高版本。如果將現有網域升級為服務軟體 R20211203，角色名稱將保持不變。

已重新命名的 CloudWatch 指標

針對執行 OpenSearch 的網域，數個 CloudWatch 指標會有所變更。當您將網域升級為 OpenSearch 時，指標會自動變更，而您目前的 CloudWatch 警示將中斷。在將叢集從 Elasticsearch 版本升級至 OpenSearch 版本之前，請務必更新您的 CloudWatch 警示以使用新的指標。

下列指標已變更：

原始指標名稱	新名稱
KibanaHealthyNodes	OpenSearchDashboardsHealthyNodes
KibanaConcurrentConnections	OpenSearchDashboardsConcurrentConnections
KibanaHeapTotal	OpenSearchDashboardsHeapTotal
KibanaHeapUsed	OpenSearchDashboardsHeapUsed
KibanaHeapUtilization	OpenSearchDashboardsHeapUtilization
KibanaOS1MinuteLoad	OpenSearchDashboardsOS1MinuteLoad
KibanaRequestTotal	OpenSearchDashboardsRequestTotal
KibanaResponseTimesMaxInMillis	OpenSearchDashboardsResponseTimesMaxInMillis
ESReportingFailedRequestSysErrCount	KibanaReportingFailedRequestSysErrCount
ESReportingRequestCount	KibanaReportingRequestCount
ESReportingFailedRequestUserErrCount	KibanaReportingFailedRequestUserErrCount

原始指標名稱	新名稱
ESReportingSuccessCount	KibanaReportingSuccessCount
ElasticsearchRequests	OpenSearchRequests

如需 OpenSearch Service 傳送至 Amazon CloudWatch 的完整指標清單，請參閱the section called “監控叢集指標”。

「帳單和成本管理」主控台的變更

帳單和成本管理主控台及成本與用量報告中的歷史資料將繼續使用舊的服務名稱，因此在搜尋資料時，需要開始針對 Amazon OpenSearch Service 和舊版 Elasticsearch 名稱使用篩選條件。如果您有已儲存的報告，請更新篩選條件，以確定它們也包含 OpenSearch Service。當 Elasticsearch 的使用量減少而 OpenSearch 的使用量增加時，您一開始可能會收到提醒，但它會在幾天內消失。

除了服務名稱以外，對於所有報告、帳單及價格清單 API 操作，下列欄位將變更：

欄位	舊格式	新格式
執行個體類型	m5.large.elasticsearch	m5.large.search
產品系列	Elasticsearch 執行個體 Elasticsearch 磁碟區	Amazon OpenSearch Service 執行個體 Amazon OpenSearch Service 磁碟區
定價說明	每個 c5.18xlarge.elasticsearch 執行個體小時 (或不足一小時) 5.098 美元 - 歐盟	每個 c5.18xlarge.search 執行個體小時 (或不足一小時) 5.098 美元 - 歐盟
執行個體系列	ultrawarm.elasticsearch	ultrawarm.search

新事件格式

OpenSearch Service 傳送到 Amazon EventBridge 和 Amazon CloudWatch 的事件格式已變更，尤其是 `detail-type` 欄位。來源欄位 (`aws.es`) 保持不變。如需每個事件類型的完整格式，請參閱the section called “監控事件”。如果您有依賴於舊格式的現有事件規則，請務必更新它們以符合新格式。

什麼保持不變？

下列特性和功能以及未列出項將保持不變：

- 服務委託人 (`es.amazonaws.com`)
- 廠商程式碼
- 網域 ARN
- 網域端點

開始使用：將您的網域升級至 OpenSearch 1.x

OpenSearch 1.x 支援從 Elasticsearch 版本 6.8 升級至 7.x。如需升級網域的指示，請參閱the section called “升級網域（主控台）”。如果您使用 AWS CLI 或 組態 API 來升級網域，則需要將 指定`TargetVersion`為 `OpenSearch_1.x`。

OpenSearch 1.x 引入名為啟用相容性模式的其他網域設定。由於某些 Elasticsearch OSS 用戶端和外掛程式會在連線前檢查叢集版本，因此相容性模式會設定 OpenSearch 以將其版本報告為 7.10，以便這些用戶端可以繼續運作。

在首次建立 OpenSearch 網域時，或是從 Elasticsearch 版本升級至 OpenSearch 時，您可啟用相容性模式。如果沒有設定，當您建立網域時，參數預設為 `false`，而當您升級網域時，預設為 `true`。

若要使用組態 API 啟用相容性模式，將 `override_main_response_version` 設定為 `true`：

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/upgradeDomain
{
  "DomainName": "domain-name",
  "TargetVersion": "OpenSearch_1.0",
  "AdvancedOptions": {
    "override_main_response_version": "true"
   }
}
```

若要啟用或停用現有 OpenSearch 網域中的相容性模式，您需要使用 OpenSearch _cluster/settings API 操作：

```
PUT /_cluster/settings
{
  "persistent" : {
    "compatibility.override_main_response_version" : true
  }
}
```

Amazon OpenSearch Service 疑難排解

本主題說明如何辨識和解決常見的 Amazon OpenSearch Service 問題。聯絡 AWS 支援之前，先參閱本節資訊。

無法存取 OpenSearch Dashboards

OpenSearch Dashboards 端點不支援已簽署的請求。如果網域的存取控制政策只對特定 IAM 角色授予存取權，而且您尚未設定 Amazon Cognito 身分驗證，當您嘗試存取 Dashboards 時可能會收到以下錯誤：

```
"User: anonymous is not authorized to perform: es:ESHttpGet"
```

如果您的 OpenSearch Service 網域使用 VPC 存取，您可能不會收到此錯誤，但請求可能逾時。
如要進一步了解有關修正此問題和各種可用組態選項的詳細資訊，請參閱the section called “控制對 Dashboards 的存取”、the section called “關於 VPC 網域上的存取政策”以及 the section called “身分和存取權管理”。

無法存取 VPC 網域

請參閱 the section called “關於 VPC 網域上的存取政策” 和 the section called “測試 VPC 網域”。

叢集處於唯讀狀態

與較早的 Elasticsearch 版本相比，OpenSearch 和彈性搜索 7.x 使用不同的系統進行叢集協調。在這個新系統中，當叢集遺失仲裁，在您採取動作之前，叢集將無法使用。遺失仲裁有兩種形式：

- 如果您的叢集使用專用主節點，當一半或更多個節點無法使用時，會發生仲裁遺失。
- 如果您的叢集使用專用主節點，當一半或更多個資料節點無法使用時，會發生仲裁遺失。

如果發生仲裁遺失，而且您的叢集有多個節點，則 OpenSearch Service 會還原仲裁並將叢集置於唯讀狀態。您有兩種選擇：

- 移除唯讀狀態，並依原狀使用叢集。
- 從快照還原叢集或個別索引。

如果您偏好依原樣使用叢集，請使用下列請求來驗證叢集運作狀態為綠色：

```
GET _cat/health?v
```

如果叢集運作狀態為紅色，建議從快照還原叢集。您也可以參閱 the section called “紅色叢集狀態” 以取得疑難排解步驟。如果叢集運作狀態為綠色，請使用下列請求來檢查所有預期索引是否存在：

```
GET _cat/indices?v
```

然後執行一些搜尋來驗證預期的資料存在。如果存在，您可以使用下列請求來移除唯讀狀態：

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.blocks.read_only": false
  }
}
```

如果發生仲裁遺失，而且您的叢集只有一個節點，則 OpenSearch Service 會取代節點，並且不會使叢集進入唯讀狀態。否則，您的選項會是相同的：依原樣使用叢集或從快照還原。

在這兩種情況下，OpenSearch Service 都會傳送兩個事件到您的 Health 儀板表。第一個會通知您遺失仲裁。第二個會在 OpenSearch Service 成功還原仲裁後發生。如需使用 的詳細資訊 Health 儀板表，請參閱AWS Health 《 使用者指南》。

紅色叢集狀態

紅色叢集狀態表示至少有一個主要碎片及其複本不會分配到節點。OpenSearch Service 會一直嘗試拍攝所有索引的自動快照，不論其狀態為何，但在紅色叢集狀態存在時，快照會失敗。

紅色叢集狀態的最常見原因是叢集節點失敗，而 OpenSearch 程序會因為持續的繁重處理負載而造成當機。

> **Note**
>
> 無論叢集狀態為何，OpenSearch Service 都會將自動快照存放 14 天。因此，如果紅色叢集狀態持續超過兩週，將會刪除最後一個狀況良好的自動快照，而您可能永遠遺失叢集資料。如果您的 OpenSearch Service 網域進入紅色叢集狀態， 支援 可能會與您聯絡，詢問您是否要自行

解決問題，還是希望支援團隊提供協助。您可以設定 CloudWatch 警示，以便在發生紅色叢集狀態時通知您。

最後，紅色碎片會導致紅色叢集，而紅色索引會造成紅色碎片。OpenSearch 有一些實用的 API，用來識別造成紅色叢集狀態的索引。

- `GET /_cluster/allocation/explain` 選擇第一個發現的未指派碎片，並說明為何無法將它分配到節點：

```
{
    "index": "test4",
    "shard": 0,
    "primary": true,
    "current_state": "unassigned",
    "can_allocate": "no",
    "allocate_explanation": "cannot allocate because allocation is not permitted to
 any of the nodes"
}
```

- `GET /_cat/indices?v` 顯示每個索引的良好運作狀態、文件數量和磁碟使用方式：

```
health status index            uuid                   pri rep docs.count docs.deleted
 store.size pri.store.size
green  open   test1            30h1EiMvS5uAFr2t5CEVoQ   5   0        820            0
       14mb           14mb
green  open   test2            sdIxs_WDT56afFGu5KPbFQ   1   0          0            0
       233b           233b
green  open   test3            GGRZp_TBRZuSaZpAGk2pmw   1   1          2            0
     14.7kb          7.3kb
red    open   test4            BJxfAErbTtu5HBjIXJV_7A   1   0
green  open   test5            _8C6MIXOSxCqVYicH3jsEA   1   0          7            0
     24.3kb         24.3kb
```

刪除紅色索引能以最快速度修正紅色叢集狀態。根據紅色叢集狀態的原因不同，您可能會擴展 OpenSearch Service 網域以使用較大的執行個體類型、更多執行個體或更多 EBS 型儲存，並嘗試重新建立有問題的索引。

若無法刪除有問題的索引，您可以還原快照、從索引刪除文件、變更索引設定、減少複本數量，或刪除其他索引以釋出更多可用磁碟空間。重要步驟是在重新設定 OpenSearch Service 網域之前先解決紅色

叢集狀態問題。重新設定具紅色叢集狀態的網域可能使問題複雜化，進而導致網域停滯在 Processing (處理中) 的設定狀態，直到您解決狀態問題為止。

紅色叢集的自動修復

如果叢集的狀態一直為紅色且持續超過一小時，OpenSearch Service 會嘗試藉由重新路由未分配的碎片或從過去的快照中復原來自動修復該問題。

如果無法修復一個或多個紅色索引，且叢集狀態保持紅色總共 14 天，則僅在叢集滿足至少一個以下準則時，OpenSearch Service 才會採取進一步的動作：

- 只有一個可用區域
- 沒有專用主節點
- 包含爆量執行個體類型 (T2 或 T3)

此時，如果您的叢集滿足這些條件之一，OpenSearch Service 會在接下來的 7 天每天向您發送通知，說明如果您不修復這些索引，則將會刪除所有未指派的碎片。如果您的叢集狀態在 21 天後仍為紅色，則 OpenSearch Service 會刪除所有紅色索引上的未指派碎片 (儲存和運算)。您會在 OpenSearch Service 主控台的 Notifications (通知) 面板中收到每個這些事件的通知。如需詳細資訊，請參閱the section called “叢集運作狀態事件”。

從持續繁重的處理負載中復原

若要判斷紅色叢集狀態是否是資料節點上的持續繁重處理負載所造成，可監控以下叢集指標。

相關指標	Description	復原
JVMMemoryPressure	指定用於叢集中所有資料節點的 Java heap 的百分比。檢視此指標的最大統計資料來獲知這項指標，並且檢視當 Java 垃圾回收器無法回收足夠記憶體，進而造成記憶體越來越小的壓力。此模式可能是因為複雜的查詢或大型資料欄位所造成。 x86 執行個體類型使用 Conconnect Mark Sweep (CMS) 廢棄項目收集器，與應用程式執行緒一起執行，將	設定 JVM 的記憶體斷路器。如需詳細資訊，請參閱the section called “JVM OutOfMemoryError”。 如果問題仍存在，請刪除不需要的索引、減少對網域的請求數或複雜度、新增執行個體，或使用較大的執行個體類型。

相關指標	Description	復原
	暫停可能情況降到短暫。如果 CMS 在一般收集期間無法回收足夠的記憶體，會觸發完整的廢棄項目收集，導致長時間應用程式暫停和影響叢集穩定性。 ARM 型 Graviton 執行個體類型使用 Garbage-First (G1) 廢棄項目收集器，它類似於 CMS，但使用額外的短暫暫停和堆積磁碟重組，進一步減少對完整廢棄項目收集的需求。 在任何一種情況下，如果記憶體使用量持續增長，超出廢棄項目收集器在完整廢棄項目收集期間的回收量，OpenSearch 會因記憶體不足錯誤而損毀。在所有執行個體類型上，最好的經驗法則是將使用量保持低於 80%。 `_nodes/stats/jvm` API 提供有用的 JVM 統計資料摘要、記憶體集區使用量，以及垃圾回收資訊： `GET domain-endpoint /_nodes/stats/jvm?pretty`	
CPUUtilization	指定用於叢集中的資料節點的 CPU 資源百分比。檢視此指標的最大統計資料，尋找持續的高用量模式。	新增資料節點，或增加現有資料節點之執行個體類型的大小。
節點	指定叢集中的節點數。檢視此指標的最小統計資料。當服務為叢集部署新的執行個體時，這個值會有所變動。	新增資料節點。

黃色叢集狀態

黃色叢集狀態表示已將所有索引的主要碎片分配給叢集中的節點，但至少有一個索引的複本碎片未獲分配。單一節點叢集在初始化時一定會呈現黃色叢集狀態，因為這時沒有其他節點可供 OpenSearch Service 指派複本。為了達到綠色叢集狀態，請增加節點數。如需詳細資訊，請參閱the section called “調整網域大小”。

建立新索引或發送節點故障後，多節點叢集可能會短暫呈現黃色叢集狀態。當 OpenSearch 在整個叢集中複寫資料時，此狀態會自我解析。磁碟空間不足也可能導致黃色叢集狀態；如果節點擁有容納它們的磁盤空間，叢集只能分配複本分片。

ClusterBlockException

您可能會接收到 `ClusterBlockException` 錯誤，原因如下。

缺少可用儲存空間

如果叢集中的一或多個節點的儲存空間小於 1) 20% 的可用儲存空間，或 2) 20 GiB 的儲存空間，則新增文件和建立索引等基本寫入操作可能會開始失敗。 the section called “計算儲存需求”會提供 OpenSearch Service 如何使用磁碟空間的摘要。

為了避免問題，請在 OpenSearch Service 主控台中監控 `FreeStorageSpace` 指標，並建立 CloudWatch 警示以便在 `FreeStorageSpace` 下降到特定閾值時觸發。`GET /_cat/allocation?v` 也提供有用的碎片分配和磁碟用量的摘要。若要解決缺乏儲存空間的相關問題，可擴展 OpenSearch Service 網域以使用較大的執行個體類型、更多執行個體或更多 EBS 型儲存體。

高 JVM 記憶體壓力

當 JVMMemoryPressure 指標超過 30 分鐘的 92% 時，OpenSearch Service 會觸發保護機制，封鎖所有寫入操作，以預防叢集達到紅色狀態。若保護已啟動，則當寫入操作失敗且出現 `ClusterBlockException` 錯誤時，新的索引將無法建立，並會擲出 `IndexCreateBlockException` 錯誤。

當 JVMMemoryPressure 指標傳回 88% 或更低達五分鐘之久，會關閉保護機制，並取消封鎖叢集的寫入操作。

高 JVM 記憶體壓力可能是因為叢集的請求數量猛增、節點之間的碎片配置不均衡、叢集中的碎片太多、欄位資料或索引映射激增，或無法處理傳入負載的執行個體類型所致。它也可能是由於在查詢中使用彙總、萬用字元或寬泛的時間範圍造成的。

若要減少叢集的流量並解決高 JVM 記憶體壓力問題，請嘗試執行下列其中一個或多個動作：

- 擴展網域，使每個節點的堆積大小上限為 32 GB。
- 透過刪除舊的或未使用的索引來減少碎片數量。
- 使用 `POST index-name/_cache/clear?fielddata=true` API 操作清除資料快取。請注意，清除快取可能會中斷進行中的查詢。

一般而言，為了避免將來出現高 JVM 記憶體壓力，請遵循下列最佳實務：

- 避免彙總文字欄位，或將您的索引的映射類型變更為 `keyword`。
- 透過選擇正確的碎片數量來優化搜尋和索引請求。
- 設定索引狀態管理 (ISM) 政策以定期移除未使用的索引。

使用待命遷移至異地同步備份時發生錯誤

當您將現有網域遷移至具有待命的異地同步備份時，可能會發生下列問題。

從沒有待命的網域遷移到有待命的網域期間建立索引、索引範本或 ISM 政策

如果您在使用待命從異地同步備份遷移網域至異地同步備份時建立索引，且索引範本或 ISM 政策未遵循建議的資料複製準則，這可能會導致資料不一致，且遷移可能會失敗。若要避免這種情況，請使用資料複製計數（包括主節點和複本）為三的倍數建立新的索引。您可以使用 `DescribeDomainChangeProgress` API 檢查遷移進度。如果您遇到複本計數錯誤，請修正錯誤，然後聯絡 AWS Support 重試遷移。

不正確的資料副本數量

如果您的網域中沒有正確數量的資料副本，遷移到具有待命的異地同步備份將會失敗。

JVM OutOfMemoryError

JVM `OutOfMemoryError` 通常表示達到以下其中一個 JVM 斷路器。

斷路器	Description	叢集設定屬性
上層中斷器	可用於所有斷路器的 JVM 堆積記憶體總百分比。預設值為 95%。	`indices.breaker.total.limit`
欄位資料中斷器	可讓 JVM 堆積記憶體將單一資料欄位載入到記憶體的百分比。預設值為 40%。如果您上傳具大型欄位的資料，您可能需要提高此限制。	`indices.breaker.fielddata.limit`
請求中斷器	允許資料結構用於回應的服務請求的 JVM 堆積記憶體百分比。預設值為 60%。如果您的服務請求涉及計算彙總，您可能需要提高此限制。	`indices.breaker.request.limit`

叢集節點失敗

Amazon EC2 執行個體可能遇到意外終止和重新啟動。一般來說，OpenSearch Service 會為您重新啟動節點。不過，OpenSearch 叢集中的一個或多個節點可能依然處於失敗狀況。

若要查看此情況，請在 OpenSearch Service 主控台上開啟您的網域儀表板。前往 Cluster health (叢集運作狀態) 標籤，然後找到 Total nodes (總節點) 指標。查看回報節點數是否少於您為叢集設定的數量。如果指標顯示一個或多個節點未運作一天以上，請聯絡 AWS 支援。

您也可以設定 CloudWatch 警示，以便在發生此問題時收到通知。

> **Note**
>
> 叢集組態變更以及服務的例行維護期間，總節點指標不準確。此行為在預期當中。該指標很快就會報告正確的叢集節點數量。如需詳細資訊，請參閱 the section called “組態變更”。

為了預防您的叢集發生意外的節點終止和重新啟動情況，在 OpenSearch Service 網域中為每個索引至少建立一個複本。

超出最大碎片限制

OpenSearch 以及 7.x 版本的 Elasticsearch 具有的預設設定為每個節點不超過 1,000 個碎片。如果請求 (例如建立新索引) 會導致您超出此限制，則 OpenSearch/Elasticearch 會擲出錯誤。如果您遇到此錯誤，您有幾個選項：

- 將更多資料節點新增至叢集。
- 增加 `_cluster/settings/cluster.max_shards_per_node` 設定。
- 使用 _shrink API 來減少節點上的碎片數目。

網域卡在處理狀態

當您的 OpenSearch Service 網域在進行組態變更時，將會進入「處理中」狀態。當您啟動組態變更時，網域狀態會變更成「處理中」，而 OpenSearch Service 會建立一個新環境。在新的環境中，OpenSearch Service 將啟動一組新的適用節點 (如資料、主要節點或 UltraWarm)。遷移完成後，舊節點將會終止。

如果發生下列其中一種情況，叢集可能會卡在「處理中」狀態：

- 一組新的資料節點無法啟動。

- 將碎片遷移至新的資料節點集不成功。
- 驗證檢查失敗並顯示錯誤。

如需有關上述每種情況下的詳細解決方案步驟，請參閱為什麼我的 Amazon OpenSearch Service 網域卡在「處理中」狀態？。

低 EBS 爆量餘額

當您其中一個一般用途 (SSD) 磁碟區上的 EBS 爆量餘額低於 70% 時，OpenSearch Service 會向您傳送主控台通知；如果餘額低於 20%，則會傳送後續通知。若要解決此問題，您可以縱向擴展叢集，或減少讀取和寫入 IOPS，以便記入爆量餘額。對於具有 gp3 磁碟區類型的網域，以及具有磁碟區大小高

於 1000 GiB 之 gp2 磁碟區的網域，爆量餘額會保持在 0。如需詳細資訊，請參閱一般用途 SSD 磁碟區 (gp2)。您可以使用 BurstBalance CloudWatch 指標監控 EBS 爆量餘額。

在磁碟區調整大小期間增加 EBS 指標

修改 Amazon Elastic Block Store 磁碟區大小時，您可能會發現各種 EBS 指標暫時增加，例如 Write Throughput、Write Throughput Micro bursting、 Disk Queue Depth和 IOPS。這是調整大小操作期間的預期行為，通常持續幾秒鐘。不過，持續時間可能會根據修改的磁碟區大小而有所不同。

為了避免延遲問題和請求拒絕，只有在叢集運作狀態良好且網路流量低時，才能執行 EBS 磁碟區調整大小。

無法啟用稽核日誌

當您嘗試使用 OpenSearch Service 主控台啟用稽核日誌發佈時，可能會遇到下列錯誤：

為 CloudWatch Logs 日誌群組指定的資源存取政策沒有為 Amazon OpenSearch Service 授予足夠的許可來建立日誌串流。請檢查資源存取政策。

如果您遇到此錯誤，請確認政策的 resource 元素包含正確的日誌群組 ARN。如果有，請遵循以下步驟：

1. 等候幾分鐘。
2. 在 Web 瀏覽器中重新整理該頁面。
3. 選擇 Select existing group (選擇現有群組)。
4. 對於 Existing log group (現有日誌群組)，選擇您在收到錯誤訊息之前建立的日誌群組。
5. 在存取政策區段中，選擇 Select existing policy (選取現有政策)。
6. 對於 Existing policy (現有政策)，選擇您在收到錯誤訊息之前建立的政策。
7. 選擇啟用。

如果在重複此程序數次後仍然存在錯誤，請連絡 AWS Support。

無法關閉索引

OpenSearch Service 僅支援適用於 OpenSearch 和 Elasticsearch 版本 7.4 及更新版本的 _close API。如果正在使用較舊版本並從快照中恢復索引，您可以刪除現有索引 (在重新編製索引前後)。

用戶端授權檢查

Logstash 和 Beats 的預設發行版包括專有授權檢查，並且無法連接到 OpenSearch 的開源版本。確保您搭配使用這些用戶端的 Apache 2.0 (OSS) 發行版與 OpenSearch Service。

請求調節

如果您收到持續的 `403 Request throttled due to too many requests` 或 `429 Too Many Requests` 錯誤，請考慮垂直擴展。如果承載會導致記憶體使用量超過 Java 堆積的大小上限，則 Amazon OpenSearch Service 會調節請求。

無法對節點執行 SSH

您無法使用 SSH 在 OpenSearch 叢集中存取任何節點，並且無法直接修改 `opensearch.yml`。反之，請使用 主控台 AWS CLI或 SDKs 來設定您的網域。您也可以使用 OpenSearch REST API 指定幾個叢集層級的設定。如需進一步了解，請參閱 Amazon OpenSearch Service API 參考和 the section called “受支援的操作”。

如果您需要深入了解叢集的效能，您可以將錯誤日誌和慢速日誌發佈至 CloudWatch。

「無效的物件存放區類別」快照錯誤

OpenSearch Service 快照不支援 Amazon Glacier 儲存類別。如果您的 S3 儲存貯體包含將物件轉換為 Amazon Glacier 儲存類別的生命週期規則，當您嘗試列出快照時，可能會遇到此錯誤。

如果您需要從儲存貯體還原快照，請從 Amazon Glacier 還原物件、將物件複製到新儲存貯體，並將新儲存貯體註冊為快照儲存庫。

無效主機標頭

OpenSearch Service 要求用戶端在請求標頭中指定 `Host`。有效的 `Host` 值是不含 `https://` 的網域端點，例如：

```
Host: search-my-sample-domain-ih2lhn2ew2scurji.us-west-2.es.amazonaws.com
```

如果您在提出請求時收到 `Invalid Host Header` 錯誤，請檢查您的用戶端或代理在 `Host` 標頭中包含 OpenSearch Service 網域端點 (而不是其 IP 地址 (舉例來說))。

無效的 M3 執行個體類型

OpenSearch Service 不支援對執行 OpenSearch 或 Elasticsearch 6.7 版和更新版本的現有網域新增或修改 M3 執行個體。您可以繼續搭配 Elasticsearch 6.5 或較舊版本使用 M3 執行個體。

建議您選擇較新的執行個體類型。對於執行 OpenSearch 或 Elasticsearch 6.7 和更新版本的網域，下列限制適用：

- 如果現有網域不使用 M3 執行個體，則無法再變更為它們。
- 如果您將現有網域從 M3 執行個體類型變更為其他執行個體類型，則無法切換回原本的執行個體類型。

熱查詢在啟用 UltraWarm 後停止運作

當您在網域上啟用 UltraWarm 時，如果 `search.max_buckets` 設定沒有預先存在的複寫，則 OpenSearch Service 會自動將值設定為 `10000`，以防止記憶體大量查詢充滿暖節點。如果您的熱查詢使用超過 10,000 個儲存貯體，當您啟用 UltraWarm 時，它們可能會停止運作。

由於 Amazon OpenSearch Service 的受管特性而無法修改此設定，因此您需要開啟支援案例來增加限制。增加限制不需要高級支援訂閱。

升級後無法降級

就地升級是無法復原的作業，但您仍可連絡 AWS 支援；支援人員可協助您在新網域上還原自動、預先升級的快照。例如，如果您將網域從 Elasticsearch 5.6 升級到 6.4， AWS Support 可協助您在新的 Elasticsearch 5.6 網域上還原預先升級快照。如果已經製作原有網域的手動快照，則您可以自行執行該步驟。

需要所有 的網域摘要 AWS 區域

下列指令碼使用 Amazon EC2 describe-regions AWS CLI 命令來建立可使用 OpenSearch Service 的所有區域的清單。接著，它會為個別區域呼叫 list-domain-names：

```
for region in `aws ec2 describe-regions --output text | cut -f4`
do
    echo "\nListing domains in region '$region':"
    aws opensearch list-domain-names --region $region --query 'DomainNames'
done
```

接著，您會收到下列個別區域的輸出：

```
Listing domains in region:'us-west-2'...
[
  {
    "DomainName": "sample-domain"
  }
]
```

不提供 OpenSearch Service 的區域會傳回「Could not connect to the endpoint URL (無法連接到端點 URL)」。

使用 OpenSearch Dashboards 時發生瀏覽器錯誤

當您使用 Dashboards 來檢視 OpenSearch Service 網域中的資料時，您的瀏覽器會在 HTTP 回應物件中包裝服務錯誤訊息。您可以使用 Web 瀏覽器中的常用開發人員工具，例如 Chrome 中的開發人員模式，來檢視基本服務錯誤和協助您的偵錯工作。

若要檢視 Chrome 中的服務錯誤

1. 從 Chrome 頂部選單列中，選擇 View (檢視)、Developer (開發人員)、Developer Tools (開發人員工具)。
2. 選擇網路標籤。
3. 在 Status (狀態) 欄中，選擇任何狀態為 500 的 HTTP 工作階段。

若要檢視 Firefox 中的服務錯誤

1. 從選單中，選擇 Tools (工具)、Web Developer (Web 開發人員)、Network (網路)。
2. 選擇狀態為 500 的任何 HTTP 工作階段。
3. 選擇 Response (回應) 索引標籤，以檢視服務回應。

節點碎片和儲存扭曲

節點碎片扭曲是指叢集中的一個或多個節點具有比其他節點更多的碎片。節點儲存扭曲是指叢集中的一個或多個節點具有比其他節點更多的儲存空間 (`disk.indices`)。雖然這兩種情況都可能暫時發生，例如當網域已替換節點並仍對其分配碎片，如果它們持續存在，則應進行解決。

若要識別這兩種類型的扭曲，請執行 _cat/allocation API 操作並比較回應中的 shards 和 disk.indices 條目：

```
shards    | disk.indices  | disk.used  | disk.avail   | disk.total   | disk.percent |
host     | ip        | node
   264    |      465.3mb  |   229.9mb  |      1.4tb   |      1.5tb   |            0 |
 x.x.x.x  | x.x.x.x   | node1
   115    |        7.9mb  |    83.7mb  |     49.1gb   |     49.2gb   |            0 |
x.x.x.x  | x.x.x.x   | node2
   264    |      465.3mb  |   235.3mb  |      1.4tb   |      1.5tb   |            0 |
 x.x.x.x  | x.x.x.x   | node3
   116    |        7.9mb  |    82.8mb  |     49.1gb   |     49.2gb   |            0 |
x.x.x.x  | x.x.x.x   | node4
   115    |        8.4mb  |      85mb  |     49.1gb   |     49.2gb   |            0 |
x.x.x.x  | x.x.x.x   | node5
```

雖然一些儲存扭曲是正常的，但超過平均值 10% 需引起注意。當碎片分配扭曲時，CPU、網絡和磁碟頻寬使用率也可能會變得扭曲。由於更多的資料通常意味著更多的索引和搜尋操作，負載最重的節點也往往是資源最緊張的節點，而負載較輕的節點表示未充分利用的容量。

修補：使用數倍於資料節點計數的碎片計數，以確保每個索引均勻分佈在資料節點間。

索引碎片和儲存扭曲

索引碎片扭曲是指一個或多個節點比其他節點持有更多的索引碎片。索引儲存扭曲是指一個或多個節點持有不成比例的大量索引總儲存空間。

索引扭曲比節點扭曲更難識別，因為它需要對 _cat/shards API 輸出結果進行一些處理。如果叢集或節點指標中出現一些扭曲跡象，請調查索引扭曲。以下是索引扭曲的常見跡象：

- 在資料節點的子集上發生 HTTP 429 錯誤
- 跨資料節點的索引或搜尋操作佇列不均勻
- 資料節點間的 JVM 堆積和/或 CPU 利用率不均勻

修補：使用數倍於資料節點計數的碎片計數，以確保每個索引均勻分佈在資料節點間。如果仍然會看到索引儲存或碎片扭曲，則可能需要強制重新分配碎片，OpenSearch Service 網域的每次藍/綠部署都會發生這種情況。

在選取 VPC 存取後未經授權的操作

當您使用 OpenSearch Service 主控台建立新網域時，您有選項可選擇 VPC 或公有存取。如果您選取 VPC access (VPC 存取)，OpenSearch Service 會查詢 VPC 資訊，且如果您沒有適當的許可則會失敗。

```
You are not authorized to perform this operation. (Service: AmazonEC2; Status Code:
 403; Error Code: UnauthorizedOperation
```

若要啟用此查詢，您必須有權存取 `ec2:DescribeVpcs`、`ec2:DescribeSubnets` 及 `ec2:DescribeSecurityGroups` 操作。此需求僅適用於主控台。如果您使用 AWS CLI 建立和設定具有 VPC 端點的網域，則不需要存取這些操作。

在建立 VPC 網域後載入停滯

建立使用 VPC 存取的新網域後，網域的 Configuration state (組態狀態) 的進度可能永不會超出 Loading (載入)。如果發生此問題，您的區域可能會停用 AWS Security Token Service (AWS STS)。

若要將 VPC 端點新增到您的 VPC，OpenSearch Service 需要擔任 `AWSServiceRoleForAmazonOpenSearchService` 角色。因此， AWS STS 必須啟用 ，才能在指定區域中建立使用 VPC 存取的新網域。若要進一步了解啟用和停用 AWS STS，請參閱 IAM 使用者指南。

對 OpenSearch API 的請求遭拒

引入 OpenSearch API 的標籤型存取控制，您可能會開始看到以前沒看過的存取遭拒錯誤。這可能是因為您的一或多個存取政策包含使用 `ResourceTag` 條件的 `Deny`，且目前遵守這些條件。

例如，以下政策僅用於網域具有標籤 `environment=production` 時，拒絕從組態 API 存取 `CreateDomain` 動作。即使動作清單同時包含 `ESHttpPut`，拒絕陳述式也不會套用於該動作或任何其他 `ESHttp*` 動作。

JSON

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

```
      "es:CreateDomain",
      "es:ESHttpPut"
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

為 OpenSearch HTTP 方法新支援標籤，如上所示的 IAM 身分型政策將導致連接的使用者遭拒絕存取 `ESHttpPut` 動作。之前在沒有標籤驗證的情況下，連接的使用者仍可傳送 PUT 請求。

如果您在將網域更新至服務軟體 R20220323 或更新版本後，開始看到存取遭拒錯誤，請查看身分型存取政策，了解是否是這種情況，並在必要時加以更新以允許存取。

無法從 Alpine Linux 連線

Alpine Linux 會將 DNS 回應大小限制在 512 位元組。如果您嘗試從 Alpine Linux 3.18.0 版或更低版本連線到 OpenSearch Service 網域，如果網域位於 VPC 中且具有超過 20 個節點，DNS 解析可能會失敗。如果您使用的 Alpine Linux 版本高於 3.18.0，您應該能夠解析超過 20 個主機。如需詳細資訊，請參閱 Alpine Linux 3.18.0 版本備註。

如果您的網域位於 VPC，我們建議使用其他 Linux 發行版本 (如 Debian、Ubuntu、CentOS、Red Hat Enterprise Linux 或 Amazon Linux 2) 來連線到該網域。

搜尋背壓的請求過多

CPU 型許可控制是一種閘道機制，可根據節點目前的容量主動限制對節點的請求數量，包括流量的有機增加和尖峰。過多的請求會在拒絕時傳回 HTTP 429「太多請求」狀態碼。此錯誤表示叢集資源不足、資源密集型搜尋請求，或工作負載中意外尖峰。

搜尋背壓提供拒絕的原因，這有助於微調資源密集型搜尋請求。對於流量尖峰，我們建議使用指數退避和抖動進行用戶端重試。

使用開發套件時發生憑證錯誤

由於 AWS SDKs使用來自您電腦的 CA 憑證，因此當您嘗試使用 SDK 時，伺服器上憑證的變更 AWS 可能會導致連線失敗。錯誤訊息會不同，但通常包含以下文字：

```
Failed to query OpenSearch
...
SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

您可以將電腦的 CA 憑證和作業系統保持在最新版本以避免此類作業失敗。如果您在企業環境中遇到此問題，且並無管理使用專屬的電腦，則您可能需要尋求管理員協助更新程序。

以下清單列出作業系統及 Java 版本的最低版本需求：

- 具備自 2005 年 1 月起之更新的 Microsoft Windows 版本，或更新版本 (其信任清單中需包含至少其中一項所需的 CA)。
- Mac OS X 10.4 搭配適用於 Mac OS X 10.4 發行版本 5 (2007 年 2 月) 的 Java，Mac OS X 10.5 (2007 年 10 月) 和更新版本 (其信任清單中需包含至少其中一項所需的 CA)。
- Red Hat Enterprise Linux 5 (2007 年 3 月)、6 和 7 以及 CentOS 5、6 和 7，全部均需在其預設信任 CA 清單中包含至少其中一項所需的 CA。
- Java 1.4.2_12 (2006 年 5 月)、5 更新版本 2 (2005 年 3 月) 及所有更新版本，包括 Java 6 (2006 年 12 月)、7 和 8 (其預設信任 CA 清單中需包含至少其中一項所需的 CA)。

三個憑證授權機構如下：

- Amazon 根 CA 1
- Starfield Services 根憑證認證機構：G2
- Starfield 類別 2 憑證認證機構

前兩個授權機構的根憑證可以從 Amazon 信任服務取得，但是將電腦保持在最新狀態是更直接的解決方案。若要進一步了解 ACM 提供的憑證，請參閱 AWS Certificate Manager 常見問答集。

> **Note**
>
> 目前，us-east-1 區域中的 OpenSearch Service 網域使用不同授權機構的憑證。我們計劃更新區域，於近期使用這些新的憑證授權機構。

由於版本相容性，自訂外掛程式安裝失敗

問題：外掛程式安裝失敗，因為外掛程式與執行中的 OpenSearch 執行個體之間的版本不相符。系統會傳回下列錯誤：

```
PluginValidationFailureReason : The provided plugin could not be loaded.
```

原因：已針對 OpenSearch ${*MAJOR*}.$ 編譯外掛程式*{MINOR}.{PATCH}*，但您的環境正在執行 OpenSearch ${*MAJOR*}.${*MINOR*}.0。基於穩定性和安全性考量，OpenSearch 需要外掛程式與核心 OpenSearch 安裝之間的確切版本相符。

可能修正：使用 OpenSearch 版本 ${*MAJOR*}.$ 建置外掛程式*{MINOR}*.0 以符合您的叢集版本。

驗證和更新 OpenSearch 的版本

1. 使用叢集的 API 或儀表板來執行下列命令。將*#######*取代為您自己的資訊。

   API 請求：

   ```
   curl -X GET your-opensearch-endpoint/
   ```

   儀表板中的開發工具主控台：

   ```
   GET /
   ```

   命令會以下列格式傳回資訊。

   ```
   {
     "name": "node-id",
     "cluster_name": "account-id:domain-name",
     "cluster_uuid": "cluster-uuid",
     "version": {
       "distribution": "opensearch",
       "number": "2.17.0",
       "build_type": "tar",
       "build_hash": "unknown",
       "build_date": "2024-12-17T11:00:09.799828091Z",
       "build_snapshot": false,
       "lucene_version": "9.11.1",
       "minimum_wire_compatibility_version": "7.10.0",
   ```

```
    "minimum_index_compatibility_version": "7.0.0"
  },
  "tagline": "The OpenSearch Project: https://opensearch.org/"
}
```

2. 如果版本號碼不是 ${*MAJOR*}.${*MINOR*}.0，執行下列動作來重建外掛程式：

   a. 更新外掛程式的 `descriptor.properties`以指定版本 ${*MAJOR*}.${*MINOR*}.0。

   b. 為您的專案類型使用 命令重建外掛程式。

   c. 使用新建置`.zip`的檔案執行 update-package 命令。

   d. 執行 associate-package 命令，將您在上一個步驟中執行`update-package`命令時建立的最新外掛程式版本建立關聯。

Amazon OpenSearch Service 的文件歷史記錄

此主題說明 Amazon OpenSearch Service 的重要變更。服務軟體更新會新增對新功能、安全修補程式和其他改進項目的支援。若要使用新功能，您可能需要更新網域上的服務軟體。如需詳細資訊，請參閱the section called “服務軟體更新”。

服務功能會逐步推展到提供服務 AWS 區域 的 。我們僅針對第一個版本更新此文件。我們不會提供有關區域可用性的資訊，也不會宣布後續區域的推展情況。如需服務功能的區域可用性以及訂閱更新通知的相關資訊，請參閱最新消息 AWS？

如需獲取更新通知，您可以訂閱 RSS 摘要。

> **Note**
>
> 修補程式版本：以 "-P" 和數字結尾的服務軟體版本，例如 R20211203-P4，是修補程式版本。修補程式可能包括效能改進、小錯誤修復、安全修復或狀態改善。由於修補程式不包括新功能或重大變更，因此通常不會對使用者或文件產生直接影響，這就是為什麼在此文件歷史記錄中不包含每個修補程式的詳細資訊。

變更	描述	日期
支援 OpenSearch 3.5	Amazon OpenSearch Service 支援 OpenSearch 3.4 和 3.5 版。如需詳細資訊，請參閱 OpenSearch 3.4 版本備註和 OpenSearch 3.5 版本備註。如需 OpenSearch 3.4 和 3.5 操作的相關資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。如需重大變更的完整清單，請參閱此處的 OpenSearch 文件。	2026 年 3 月 17 日
新功能：向量索引的 GPU 加速、自動最佳化和向量擷取	Amazon OpenSearch Service 現在支援向量索引的 GPU 加速，從而實現更快的向量搜尋	2025 年 12 月 2 日

	操作。Amazon OpenSearch Service 也引進自動最佳化功能，以自動調整集合效能和向量擷取，簡化資料載入。如需詳細資訊，請參閱向量索引的 GPU 加速、自動最佳化和向量擷取。	
支援 OpenSearch 3.3	Amazon OpenSearch Service 支援 OpenSearch 3.2 和 3.3 版。如需詳細資訊，請參閱 OpenSearch 3.2 版本備註和 OpenSearch 3.3 版本備註。如需有關 OpenSearch 3.2 和 3.3 操作的資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。	2025 年 11 月 19 日
支援 Graviton4-based執行個體系列 (c8g、m8g、r8g r8gd)	Amazon OpenSearch Service 現在支援Graviton4-based 執行個體系列 (c8g、m8g、r8g、r8gd)。如需詳細資訊，請參閱 Amazon OpenSearch Service 中支援的執行個體類型。	2025 年 10 月 17 日

新功能：OpenSearch Ingestion Machine Learning 離線批次推論	Amazon OpenSearch Ingestion (OSI) 管道支援機器學習 (ML) 離線批次推論處理，以低成本有效率地充實大量資料。每當您有可非同步處理的大型資料集時，請使用離線批次推論。離線批次推論適用於 Amazon Bedrock 和 SageMaker 模型。此功能適用於所有 AWS 區域 支援 OpenSearch Service 2.17+ 網域的 OpenSearch Ingestion。如需詳細資訊，請參閱使用 OpenSearch 擷取管道搭配機器學習離線批次推論。	2025 年 10 月 2 日
新功能：設定遠端推論以進行語意反白的 CloudFormation 範本	語意反白是一種進階搜尋功能，透過分析查詢的意義和內容來增強結果相關性，而不是僅依賴確切的關鍵字比對。此功能使用機器學習模型來評估搜尋查詢和文件內容之間的語意相似性，識別和強調文件中最相關的句子或段落。Amazon OpenSearch Service 現在包含一個 CloudFormation 範本，可自動設定 Amazon SageMaker 模型以進行語意反白。如需詳細資訊，請參閱語意反白範本的遠端推論。	2025 年 9 月 26 日
AmazonOpenSearchIngestionServiceRolePolicy 受管政策已更新	更新提供 OpenSearch Ingestion 許可，以修改 OpenSearch 建立的 VPC 端點，以便在 VPCs 之間共用管道。	2025 年 9 月 19 日

新功能：設定跨帳戶擷取的 OpenSearch 擷取管道	對於推送型來源，例如 HTTP 和 OTel，Amazon OpenSearch Ingestion 可讓您將管道 AWS 帳戶 從虛擬私有雲端 (VPC) 跨共用到個別 VPC 中的管道端點。如需詳細資訊，請參閱設定跨帳戶擷取的 OpenSearch 擷取管道	2025 年 9 月 19 日
Amazon OpenSearch Serverless 支援磁碟型向量搜尋	磁碟型向量搜尋可大幅降低低記憶體環境中向量工作負載的操作成本。如需詳細資訊，請參閱磁碟型向量搜尋。	2025 年 9 月 18 日
支援 OpenSearch 3.1	Amazon OpenSearch Service 支援 OpenSearch 3.1 版。如需升級相容性的詳細資訊，請參閱支援的升級路徑。如需新版本變更的詳細資訊，請參閱 OpenSearch 3.0 和 3.1 文件。如需 OpenSearch 3.1 操作的相關資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。	2025 年 9 月 9 日
Amazon OpenSearch Serverless 支援聯邦資訊處理標準 (FIPS) 140-2	FIPS 140-2 是美國和加拿大政府標準，指定保護敏感資訊之密碼編譯模組的安全要求。支援 FIPS AWS 區域 的提供 Amazon OpenSearch Serverless FIPS 端點。這些端點使用 TLS 1.2 或更新版本，以及 FIPS 驗證的密碼編譯演算法進行所有通訊。	2025 年 9 月 5 日

新功能：以 SAML 屬性為基礎的精細存取控制	Amazon OpenSearch Service 支援搭配 SAML 的精細存取控制，可將身分提供者的使用者和群組動態映射至 OpenSearch 精細存取控制使用者和角色。您可以將這些角色範圍限定在特定 OpenSearch 網域和無伺服器集合，並定義索引層級許可和文件層級安全性。如需詳細資訊，請參閱設定 SAML 屬性型精細存取控制。	2025 年 8 月 8 日
新主題：OpenSearch UI 端點和配額	Amazon OpenSearch UI 將其現代化營運分析體驗擴展至新的 AWS 區域，包括亞太區域（海德拉巴）、亞太區域（大阪）、亞太區域（首爾）、歐洲（米蘭）、歐洲（蘇黎世）、歐洲（西班牙）和美國西部（加利佛尼亞北部），讓您能夠從單一端點取得跨受管網域和無伺服器集合的資料洞見。OpenSearch UI 可以連接到 1.3 版以上的 OpenSearch 網域和 OpenSearch 無伺服器集合。現已在 22 中提供 AWS 區域。	2025 年 8 月 7 日

新功能：自動語意擴充	自動語意擴充可簡化 Amazon OpenSearch Service 中的語意搜尋實作和功能。語意搜尋會傳回查詢結果，不僅包含關鍵字比對，還包含使用者搜尋的意圖和內容意義。如需詳細資訊，請參閱關於自動語意擴充。 自動語意擴充提供下列優點： 簡化實作 您不需要機器學習 (ML) 專業知識或複雜的整合。 自動化程序 語意擴充會在資料擷取期間自動發生。 改善搜尋相關性 語意擴充可增強搜尋結果的品質和內容準確性。 可擴展性 語意擴充功能會將語意搜尋套用至大型資料集，無需手動介入。	2025 年 8 月 6 日

新功能：Amazon OpenSearch Serverless 上的Machine Learning	OpenSearch Serverless 現在支援適用於 AI 工作流程的 Machine Learning功能，例如 RAG（擷取增強生成）和語意搜尋。使用連接器來存取託管在第三方平台或 上的 ML 演算法和遠端模型 AWS 服務。如需詳細資訊，請參閱在 Amazon OpenSearch Serverless 上設定Machine Learning。 OpenSearch Serverless 上的 Machine Learning包含下列功能： 神經搜尋和混合搜尋 執行語意搜尋操作，了解查詢內容和更相關結果的意義。結合關鍵字和語意搜尋與混合式搜尋，以提高相關性。如需詳細資訊，請參閱在 OpenSearch Serverless 上設定神經搜尋和混合搜尋。 連接器和模型 連線至第三方 ML 平台和上託管的模型 AWS 服務。使用模型設定連接器並部署，以執行預測和 AI 工作流程。 工作流程 透過鏈結多個 API 呼叫來自動化複雜的 ML 設定和預	2025 年 8 月 6 日

	先處理任務。工作流程可簡化在 OpenSearch 上建置 AI 應用程式。如需詳細資訊，請參閱在 OpenSearch Serverless 上設定工作流程。	
新功能：OpenSearch Serverless 快照	快照是 Amazon OpenSearch Serverless 集合的point-in-time備份，可提供災難復原功能。OpenSearch Serverless 會自動建立和管理集合的快照，以確保業務連續性和資料保護。 主要優點 • 無需手動組態的自動每小時備份 • 零維護開銷 • 無需額外儲存成本 • 從意外資料遺失中快速復原 • 從快照還原特定索引的能力 如需詳細資訊，請參閱使用快照備份集合。	2025 年 7 月 30 日
OpenSearch Ingestion 支援兩個新來源	OpenSearch Ingestion 現在支援下列來源： • Amazon Aurora • Amazon RDS	2025 年 7 月 17 日

新功能：OpenSearch Service 與 Amazon S3 Vectors 整合	此整合提供下列功能，可大幅增強 OpenSearch Service 中的向量搜尋。 • Amazon S3 Vectors 提供第一個具有原生支援的雲端物件存放區，以存放和查詢向量。S3 向量提供經濟實惠、彈性且耐用的向量儲存，可根據語意意義和相似性進行查詢。它可提供一秒的查詢回應時間，並降低上傳、儲存和查詢向量的成本高達 90%。 • Amazon OpenSearch Service 提供使用 Amazon S3 作為向量索引向量引擎的功能。此功能可讓您將向量資料卸載至 Amazon S3，同時以低成本維持低於一秒的向量搜尋功能。 如需詳細資訊，請參閱（預覽）OpenSearch Service 與 Amazon S3 Vectors 的整合。	2025 年 7 月 15 日

新主題：Atlassian 與 OpenSearch Ingestion 整合	您現在可以使用 Amazon OpenSearch Ingestion 管道來處理來自 Atlassian Jira 和 Confluence 服務的資料。此整合可讓您將完整的 Jira 專案和 Confluence 空間同步到 OpenSearch，以建立統一的可搜尋知識庫。整合透過持續監控和自動同步更新來維持即時相關性，同時為特定專案、問題類型和內容提供靈活的篩選選項。如需詳細資訊，請參閱搭配 Atlassian Services 使用 OpenSearch 擷取管道。	2025 年 6 月 5 日
使用 進行自訂外掛程式管理 AWS CLI	您現在可以使用 管理 Amazon OpenSearch Service 的自訂外掛程式 AWS CLI。此功能可讓您透過簡化的命令列工作流程安裝、更新、保護和解除安裝自訂外掛程式。您也可以使用 AWS KMS 金鑰保護您的外掛程式，並在更新至較新版本的 OpenSearch 時管理外掛程式升級。如需詳細資訊，請參閱使用 管理自訂外掛程式 AWS CLI和 Amazon OpenSearch Service 自訂套件 AWS KMS 整合	2025 年 5 月 27 日

支援 OpenSearch 2.19	Amazon OpenSearch Service 支援 OpenSearch 2.19 版。如需詳細資訊，請參閱 OpenSearch 2.19 版本備註。如需 OpenSearch 2.19 操作的相關資訊，請參閱 OpenSearch REST API 參考或特定外掛程式的 API 參考。	2025 年 4 月 30 日
在 OpenSearch Ingestion 中指定管道角色的新支援	當您在 Amazon OpenSearch Ingestion 中建立或更新管道時，您不再在 YAML 格式管道組態中指定管道角色。相反地，在主控台中，您可以在新的管道角色欄位中指定角色。如果您使用的是 AWS CLI，則會使用新的 `--pipeline-role-arn` 參數。如需詳細資訊，請參閱下列主題： • 在 Amazon OpenSearch 擷取中設定角色和使用者 • 建立 Amazon OpenSearch Ingestion 管道 • 教學課程：使用 Amazon OpenSearch Ingestion 將資料擷取至網域 • 教學課程：使用 Amazon OpenSearch Ingestion 將資料擷取至集合 • CreatePipeline • UpdatePipeline	2025 年 4 月 22 日

新主題	疑難排解主題 自訂外掛程式安裝失敗，因為版本相容性解決了遇到錯誤 時要採取的步驟`PluginValidationFailureReason : The provided plugin could not be loaded` 。此錯誤通常是外掛程式與執行中 OpenSearch 執行個體之間版本不相符的結果。	2025 年 4 月 21 日
新功能：OpenSearch UI 與跨叢集搜尋的相容性	OpenSearch UI 現在與跨叢集搜尋相容。這可讓您在一個中使用 OpenSearch UI AWS 區域 來存取不同區域中的叢集。方法是將其設定為與相同區域中的叢集連線的遠端叢集。如需詳細資訊，請參閱跨區域和跨帳戶資料存取與跨叢集搜尋。	2025 年 4 月 16 日
修訂內容：「在 Amazon OpenSearch Service 中使用 OpenSearch UI」	我們已在 Amazon OpenSearch Service 中使用 OpenSearch UI 大修和擴展章節，包括擴展程序、其他設定詳細資訊，以及 Amazon OpenSearch Service 對 OpenSearch UI 支援的新版本歷史記錄。	2025 年 4 月 15 日

Amazon Q Developer for Amazon OpenSearch Service	Amazon Q 與 Amazon OpenSearch Service 整合提供下列生成功能： • 使用自然語言產生視覺化效果 • 檢視提醒摘要和洞見 • 在探索頁面上檢視 Amazon Q 產生的查詢結果摘要 • 檢視建議的異常偵測器 • 針對 OpenSearch 相關問題存取 Amazon Q 聊天	2025 年 3 月 31 日
AmazonOpenSearchServiceRolePolicy 受管政策已更新	此政策可讓 OpenSearch 更新僅由 OpenSearch 管理之任何 AWS IAM Identity Center 應用程式的存取範圍。	2025 年 3 月 28 日
Amazon OpenSearch Service 的 OR2 和 OM2 執行個體	Amazon OpenSearch Service 現在支援 OR2 和 OM2 執行個體。在內部基準中，OR2 的索引輸送量比 OR1 高出 26%，比 R7 高出 70%。OM2 比 OR1 高出 15%，比 M7g 高出 66%。	2025 年 3 月 25 日
CloudWatch Logs 和 Security Lake 的直接查詢	Amazon OpenSearch Service 現在支援直接查詢，以查詢 CloudWatch Logs 和 Security Lake 中的資料。	2024 年 12 月 1 日
k-NN 索引	從 OpenSearch 2.17 版開始，您可以將 k-NN 索引移至 UltraWarm 和冷儲存層。	2024 年 11 月 13 日

OpenSearch 2.17 支援	Amazon OpenSearch Service 現在支援 OpenSearch 2.17 版。此版本包含屬於 2.16 和 2.17 版的所有功能。如需詳細資訊，請參閱 2.16 和 2.17 版本備註。	2024 年 11 月 13 日
OpenSearch 2.15 支援	Amazon OpenSearch Service 現在支援 OpenSearch 2.15 版。此版本包含屬於 2.14 和 2.15 版的所有功能。如需詳細資訊，請參閱 2.14 和 2.15 版本備註。	2024 年 10 月 11 日
新的服務連結角色	Amazon OpenSearch Service 新增名為 的服務連結角色`AWSServiceRoleForOpensearchIngestionSelfManagedVpce`，允許 Amazon OpenSearch Ingestion Amazon CloudWatch 針對具有自我管理 VPC 端點的管道傳送指標資料至 。	2024 年 6 月 12 日
直接查詢	Amazon OpenSearch Service 現在支援直接在存放在 Amazon S3 的資料上執行查詢，而無需將資料擷取到 OpenSearch 索引。	2024 年 5 月 22 日
OpenSearch 2.13 支援	Amazon OpenSearch Service 現在支援 OpenSearch 2.13 版。此版本包含屬於 2.12 和 2.13 版的所有功能。如需詳細資訊，請參閱 2.12 和 2.13 版本備註。	2024 年 5 月 21 日

Data Prepper 2.7 版的 Amazon OpenSearch Ingestion 支援	Amazon OpenSearch Ingestion 新增對 Data Prepper 2.7 版的支援。如需詳細資訊，請參閱 2.7 版本備註。	2024 年 4 月 4 日
AWS 服務 OpenSearch Serverless 集合的私有存取	您現在可以在網路存取政策中授予 OpenSearch Serverless 集合的特定存取權， AWS 服務例如 Amazon Bedrock。	2024 年 3 月 28 日
就地 EBS 更新	您現在可以對網域進行一些 EBS 變更，而無需在 Amazon OpenSearch Service 中進行藍/綠部署。	2024 年 2 月 14 日
組態變更可見性	您現在可以在 Amazon OpenSearch Service 主控台和使用組態 API 追蹤網域組態變更。	2024 年 2 月 6 日
向量搜尋集合一般可用性	Amazon OpenSearch Serverless 向量搜尋集合現已正式推出。在預覽階段進行以下顯著的改進： • 向量搜尋集合現在支援具有數十億個向量的工作負載，每個向量最多可有 128 個維度。 • OpenSearch Dashboards 現在支援向量搜尋集合。	2023 年 11 月 29 日
OR1 執行個體	Amazon OpenSearch Service 現在支援 OR1 執行個體類型。	2023 年 11 月 29 日

使用 Amazon S3 的直接查詢（預覽）	直接查詢提供全受管解決方案，可在 Amazon OpenSearch Service 寫入 Amazon S3 儲存貯體後的幾秒內提供交易資料。	2023 年 11 月 29 日
時間序列集合的 10 TiB 容量	Amazon OpenSearch Serverless 新增對時間序列集合最多 10 TiB 索引資料的支援。此版本也支援所有集合類型的最大允許容量為 200 個 OCUs，以及在建立集合時停用待命複本的能力。	2023 年 11 月 29 日
OpenSearch 2.11 支援	Amazon OpenSearch Service 現在支援 OpenSearch 2.11 版。此版本包含屬於 2.10 和 2.11 版的所有功能。如需詳細資訊，請參閱 2.10 和 2.11 版本備註。	2023 年 11 月 17 日
Data Prepper 2.6 版的 Amazon OpenSearch Ingestion 支援	Amazon OpenSearch Ingestion 新增對 Data Prepper 2.6 版的支援。如需詳細資訊，請參閱 2.6 版本備註。此外，您可以將 Amazon DynamoDB 指定為管道來源。如需詳細資訊，請參閱搭配 Amazon DynamoDB 使用 OpenSearch 擷取管道。	2023 年 11 月 17 日

Amazon OpenSearch Ingestion 支援 Data Prepper 2.5 版	Amazon OpenSearch Ingestion 新增對 Data Prepper 2.5 版的支援。如需詳細資訊，請參閱 2.5 版本備註。此外，您現在可以將 OpenSearch Service 網域或 OpenSearch Serverless 集合指定為管道來源。如需詳細資訊，請參閱 Data Prepper 文件中的 OpenSearch 來源外掛程式。	2023 年 11 月 17 日
遠端推論的 CloudFormation 範本	為了簡化語意搜尋的遠端推論設定，Amazon OpenSearch Service 在主控台中提供 AWS CloudFormation 範本，可為您自動化模型佈建程序。	2023 年 11 月 7 日
更新至服務連結角色政策	新增服務連結角色政策所需的許可，`AmazonOpenSearchServiceRolePolicy` 以指派和取消指派 IPv6 地址。已取代的 Elasticsearch 政策`AmazonElasticsearchServiceRolePolicy` 也已更新，以確保回溯相容性。	2023 年 10 月 26 日
Amazon OpenSearch Serverless 生命週期政策	Amazon OpenSearch Serverless 推出索引生命週期政策，以簡化資料保留和刪除的管理。您現在可以在主控台中使用 APIs或組態界面來設定時間序列集合的資料保留政策，無需建立每日索引或指令碼來刪除舊資料。	2023 年 10 月 25 日

Im4gn 執行個體支援	Amazon OpenSearch Service 現在支援 Im4gn 執行個體類型。Im4gn 執行個體已針對管理大型資料集且每個 vCPU 需要高儲存密度的工作負載進行最佳化。	2023 年 10 月 20 日
管理選項	Amazon OpenSearch Service 現在提供多種管理選項，如果您需要對網域的問題進行疑難排解，可提供精細的控制。這些選項包括在資料節點上重新啟動 OpenSearch 程序的功能，以及重新啟動資料節點的功能。	2023 年 10 月 17 日
選用的外掛程式	Amazon OpenSearch Service 新增對四種新語言分析器外掛程式的支援：Nori（韓文）、Sudachi（日文）、Pinyin（中文）和 STConvert Analysis（中文），以及 Amazon Personalize Search Ranking 外掛程式。	2023 年 10 月 16 日
OpenSearch 2.9 支援	Amazon OpenSearch Service 現在支援 OpenSearch 2.9 版。此版本包含屬於 2.8 和 2.9 版的所有功能。如需詳細資訊，請參閱 2.8 和 2.9 版本備註。	2023 年 10 月 2 日

ML 連接器	Amazon OpenSearch Service 新增對機器學習 (ML) 連接器的支援。連接器有助於存取託管在其他 AWS 服務或第三方機器學習 (ML) 平台上的 ML 模型。	2023 年 9 月 6 日
Amazon OpenSearch Ingestion 新增對 Data Prepper 2.4 版的支援	Amazon OpenSearch Ingestion 新增對 Data Prepper 2.4 版的支援。如需詳細資訊，請參閱 2.4 版本備註。此外，您現在可以將 Amazon Managed Streaming for Apache Kafka (Amazon MSK) 指定為管道來源。	2023 年 8 月 31 日
時間序列集合的 6 TiB 容量	Amazon OpenSearch Serverless 新增對時間序列集合最多 6 TiB 索引資料的支援。此版本也支援搜尋和時間序列集合的允許容量上限為 100 個 OCUs。	2023 年 8 月 15 日
向量搜尋集合	Amazon OpenSearch Serverless 新增了建立向量搜尋集合的選項，您可以使用它來存放向量內嵌，以支援同步性和語意搜尋。	2023 年 7 月 26 日
OpenSearch 2.7 支援	Amazon OpenSearch Service 現在支援 OpenSearch 2.7 版。此版本包含屬於 2.6 和 2.7 版的所有功能。如需詳細資訊，請參閱 2.6 和 2.7 版本備註。	2023 年 7 月 10 日

Data Prepper 2.3 支援	Amazon OpenSearch Ingestion 新增支援 Data Prepper 2.3 版。如需詳細資訊，請參閱 2.3 版本備註。此外，您現在可以將 Amazon Security Lake 指定為管道來源。	2023 年 6 月 26 日
具有待命的異地同步備份	Amazon OpenSearch Service 新增了將網域部署到三個可用區域 (AZs) 的選項，每個 AZ 都包含完整的資料副本，以及其中一個 AZs 中的節點做為待命。具有待命部署的異地同步備份選項可在基礎設施故障時提供 99.99% 的可用性和一致的效能。	2023 年 5 月 3 日
新的服務連結角色	Amazon OpenSearch Service 新增名為 的服務連結角色`AWSServiceRoleForAmazonOpenSearchIngestionService` ，允許 Amazon OpenSearch Ingestion 將指標資料傳送至其中 Amazon CloudWatch。	2023 年 4 月 26 日
Amazon OpenSearch Ingestion	Amazon OpenSearch Ingestion 是全受管的資料收集器，可將即時日誌和追蹤資料交付至 OpenSearch Service 網域和 OpenSearch Serverless 集合。OpenSearch Ingestion 讓您不再需要使用 Logstash 或 Jaeger 等第三方解決方案，將資料擷取到您的網域和集合。	2023 年 4 月 26 日

OpenSearch 2.5 支援	Amazon OpenSearch Service 現在支援 OpenSearch 2.5 版。此版本包含屬於 2.4 和 2.5 版的所有功能。如需詳細資訊，請參閱 2.4 和 2.5 版本備註。	2023 年 3 月 13 日
離峰維護時段	Amazon OpenSearch Service 新增了離峰時段，這是每天 10 小時的低流量時段，在此期間可以排程需要藍/綠部署的服務軟體更新和自動調整最佳化。離峰更新有助於在較高的流量期間，將叢集專用主節點上的壓力降至最低。 對於 2 月 16 日之後建立的新網域，離峰時段會在當地時間下午 10：00 到上午 8：00 之間自動設定。對於現有的網域，您需要明確啟用視窗。	2023 年 2 月 16 日
在網域建立期間設定 SAML 身分驗證	Amazon OpenSearch Service 現在支援在網域建立期間設定 SAML 身分驗證。先前，您必須在建立網域之後設定 SAML 選項。	2023 年 2 月 1 日
VPC 網域的遠端重建索引	Amazon OpenSearch Service 新增兩個網域之間 VPC 端點連線的選項。您現在可以使用遠端重建索引，在不使用反向代理的情況下，將索引從某個 VPC 網域複製到另一個網域。VPC 網域必須執行服務軟體 R20221114 或更新版本，才能使用此功能。	2023 年 1 月 31 日

Amazon OpenSearch Serverless 正式版本	Amazon OpenSearch Serverless 現已推出正式版本。在預覽階段進行以下顯著的改進： • 當集合端點上的流量減少時，容量現在可以縮減規模到最小設定的 OCU。 • 索引和搜尋所允許的最大 OCU 已從 20 增加到 50。每個 OCU 都包含足夠的暫時性熱儲存，可容納 120 GiB 的索引資料。 • 您現在可以在建立集合時設定資料存取設定，而不必在單獨的工作流程中進行設定。	2023 年 1 月 25 日
非同步試轉	Amazon OpenSearch Service 現在支援非同步試轉，可讓您在變更組態之前執行驗證檢查，並通知您變更是否會導致藍/綠部署。	2023 年 1 月 19 日
新的服務連結角色	Amazon OpenSearch Service 新增了一個名為 `AWSServiceRoleForAmazonOpenSearchServerless` 的服務連結角色，可讓 OpenSearch Serverless 將指標資料傳送到 Amazon CloudWatch。	2022 年 11 月 29 日

Amazon OpenSearch Serverless 預覽版	Amazon OpenSearch Serverless 是 Amazon OpenSearch Service 的隨需、自動調整規模的無伺服器組態。Serverless 可消除佈建、設定及調整 OpenSearch 叢集的操作複雜性。	2022 年 11 月 29 日
支援 OpenSearch 2.3	Amazon OpenSearch Service 現在支援 OpenSearch 2.3 版。此版本包含屬於 2.0、2.1 和 2.2 版的所有功能。如需詳細資訊，請參閱 2.0、2.1、2.2 和 2.3 版本備註。2.3 版包含突破性變更。如需詳細資訊，請參閱支援的升級路徑。	2022 年 11 月 15 日
支援通知外掛程式	Amazon OpenSearch Service 現在支援通知外掛程式，該外掛程式提供來自 OpenSearch 外掛程式之所有通知的集中位置。從 2.0 版開始，提醒目的地已棄用，並由通知頻道取代。	2022 年 11 月 15 日
支援 Kibana 7.1.1	執行 Elasticsearch 7.1 的 Amazon OpenSearch Service 網域現在支援 Kibana 7.1.1 的最新修補程式版本，它可新增錯誤修正並改善安全性。當您將 7.1 網域更新至服務軟體 R20221114 時，OpenSearch Service 會自動將其升級至此修補程式版本。	2022 年 11 月 15 日

支援 Kibana 6.8.13	執行 Elasticsearch 6.8 的 Amazon OpenSearch Service 網域現在支援 Kibana 6.8.13 的最新修補程式版本，它可新增錯誤修正並改善安全性。當您將 6.8 網域更新至服務軟體 R20221114 時，OpenSearch Service 會自動將其升級至此修補程式版本。	2022 年 11 月 15 日
支援 Kibana 6.3.2	執行 Elasticsearch 6.3 的 Amazon OpenSearch Service 網域現在支援 Kibana 6.3.2 的最新修補程式版本，它可新增錯誤修正並改善安全性。當您將 6.3 網域更新至服務軟體 R20221114 時，OpenSearch Service 會自動將其升級至此修補程式版本。	2022 年 11 月 15 日
AWS PrivateLink	透過 Amazon OpenSearch Service 受管 VPC 端點，您可以使用介面 VPC 端點直接連線至 OpenSearch Service VPC 網域，而不用透過網際網路進行連線。在路由表和安全群組許可的情況下，OpenSearch Service 受管 VPC 端點只能在佈建端點的 VPC 內存取，或從與佈建端點之 VPC 對等的任何 VPC 存取。您的 VPC 網域必須執行服務軟體 R20220928 或更新版本，才能連線至介面 VPC 端點。	2022 年 11 月 7 日

錯誤修正與效能改進	服務軟體 R20220928 包含錯誤修正和效能強化功能，包括改良的 SAML 日誌記錄功能。此更新也會將預設租用戶變更為 `Global` 而非 `Private`。	2022 年 10 月 3 日
改善的 API 參考	Amazon OpenSearch Service 提供改善的全方位組態 API 參考。新參考包含所有可用的動作和資料類型、範例請求和回應語法，以及所有支援語言的對應 SDK 參考連結。	2022 年 9 月 13 日
藍/綠驗證	Amazon OpenSearch Service 現在會在藍/綠部署之前執行驗證檢查，如果您的網域不符合更新資格，則會顯示驗證錯誤。	2022 年 8 月 16 日
支援 OpenSearch 1.3	Amazon OpenSearch Service 現在支援 OpenSearch 1.3 版。如需詳細資訊，請參閱 1.3 版本備註。	2022 年 7 月 27 日
支援 ML Commons 外掛程式	Amazon OpenSearch Service 新增了對 ML Commons 外掛程式的支援，該外掛程式透過傳輸和 REST API 呼叫提供了一組常見的機器學習演算法。您還可以透過 PPL 命令與 ML Commons 外掛程式進行互動。	2022 年 7 月 27 日

支援 gp3 磁碟區	Amazon OpenSearch Service 新增了對 gp3 EBS 一般用途 SSD 磁碟區類型的支援。您可以在建立或修改網域時，指定其他佈建 IOPS 和輸送量。	2022 年 7 月 26 日
增強型最佳實務文件	Amazon OpenSearch Service 文件提供改進的可操作最佳實務和一般建議，以建立和操作 OpenSearch Service 網域。	2022 年 7 月 6 日
與 Service Quotas 整合	您現在可以從 Service Quotas 主控台，檢視 Amazon OpenSearch Service 的配額，並請求增加配額。	2022 年 6 月 29 日
OpenSearch API 的標籤型存取控制	您現在可以使用標籤來控制對 OpenSearch API 的存取。您之前只能使用標籤來控制對組態 API 的存取。	2022 年 6 月 16 日
區域間跨叢集搜尋	現在支援跨 跨叢集搜尋 AWS 區域 ，只要兩個網域都執行 Elasticsearch 7.10 版或更新版本，或任何版本的 OpenSearch。	2022 年 6 月 14 日
支援單一 Kibana 5.6	Amazon OpenSearch Service 新增對單一 Kibana 5.6.16 的支援。藉助單一 Kibana 5.6.16，您可以將 Kibana 5.6 做為您的前端，同時連接至 Elasticsearch 版本 5.1、5.3、5.5 和 5.6。您必須使用服務軟體 R20220323 或更新版本，才能使用單一 Kibana 5.6。	2022 年 4 月 4 日

R20220323-P1	Amazon OpenSearch Service 最近發佈了服務軟體更新 R20220323，但更新隨後因為一個問題而被還原。我們建議您將網域更新至修補程式版本 R20220323-P1 或更新版本，以解決此問題。	2022 年 4 月 4 日
支援 OpenSearch 1.2	Amazon OpenSearch Service 現在支援 OpenSearch 1.2 版。如需詳細資訊，請參閱 1.2 版本備註。	2022 年 4 月 4 日
可觀測性	適用於 Amazon OpenSearch Service 的 OpenSearch 儀表板預設安裝包括可觀測性外掛程式，您可以用來將使用 Piped Processing Language (PPL) 的資料驅動型事件視覺化，以便探索和查詢您的資料。該外掛程式需要 OpenSearch 1.2 或更新版本以及服務軟體 R20220323 或更新版本。	2022 年 4 月 4 日
支援 Kibana 7.7.1	執行 Elasticsearch 7.7 的 Amazon OpenSearch Service 網域現在支援 Kibana 7.7 的最新修補程式版本，新增錯誤修正並改善安全性。當您將 7.7 網域更新至軟體 R20220323 或更新版本時，OpenSearch Service 會自動將其升級至此修補程式版本。	2022 年 4 月 4 日

JVM 記憶體壓力指標變更	Amazon OpenSearch Service 變更了 `JVMMemoryPressure` CloudWatch 指標的邏輯，可更準確地反映記憶體利用率。過去，這些指標只考慮 JVM 堆的舊世代記憶體集區。經由此變更，該指標亦會考慮新世代記憶體集區。將網域更新為服務軟體 R20220323 後，您可能會看到 `JVMMemoryPressure` 、`MasterJVMMemoryPressure` 及/或 `WarmJVMMemoryPressure` 指標有所增加。	2022 年 4 月 4 日
自訂字典搭配 IK (中文) 分析外掛程式	Amazon OpenSearch Service 現在支援將自訂字典與 IK (中文) 分析外掛程式搭配使用。	2022 年 4 月 4 日
現有網域上的跨叢集複寫	Amazon OpenSearch Service 移除了您只能在 2020 年 6 月 3 日或之後建立的網域中實施跨叢集搜尋和跨叢集複寫的限制。現在，不論網域是在何時建立，您都能在所有網域上啟用這些功能。兩個網域都必須位於服務軟體 R20220323 或更新版本上。	2022 年 4 月 4 日
藍/綠部署可見性	Amazon OpenSearch Service 現可提供有關藍/綠部署進度的更高可見性。您可以在主控台中或使用組態 API 來監控這些詳細資訊。	2022 年 1 月 27 日

現有網域上的精細存取控制	您現可在現有網域上啟用精細存取控制。您可以啟用開放/IP型存取政策的臨時遷移期，以確保使用者在您建立和映射角色時可以繼續存取您的網域。若要在現有網域上啟用精細存取控制，則需使用服務軟體 R20211203 或更新版本。	2022 年 1 月 6 日
已重新命名的 OpenSearch 儀表板角色	使用服務軟體 R20211203，`kibana_user` 角色已重新命名為 `opensearch_dashboards_user`，`kibana_read_only` 已重新命名為 `opensearch_dashboards_read_only`。此變更適用於所有新建立的 OpenSearch 1.x 網域。對於升級至服務軟體 R20211203 的現有 OpenSearch 網域，角色保持不變。	2022 年 1 月 4 日
支援 OpenSearch 1.1	Amazon OpenSearch Service 現已支援 OpenSearch 1.1 版本。如需詳細資訊，請參閱 1.1 版本備註。	2022 年 1 月 4 日
ISM 視覺化編輯器	預設安裝的適用於 Amazon OpenSearch Service 的 OpenSearch 儀表板現在支援 ISM 政策的視覺化編輯器。此功能需要 OpenSearch 1.1 或更新版本。	2022 年 1 月 4 日

跨服務混淆代理人預防更新	Amazon OpenSearch Service 可支援使用 IAM 資源政策中的 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容金鑰，以避免混淆代理人問題。您必須位於服務軟體 R20211203 或更新版本上，才能使用這些條件索引鍵。	2022 年 1 月 4 日
Log4j 修補程式	服務軟體 R20211203-P2 會依照 CVE-2021-44228 和 CVE-2021-45046 中的建議，更新 OpenSearch Service 中使用的 Log4j 版本。此修補程式適用於執行所有版本的 OpenSearch 和 Elasticsearch。OpenSearch Service 將繼續在內部更新各種 Log4j 版本，並且它們不一定局限於 Log4j 的最新版本。您網域上的 Log4j 版本取決於網域執行的軟體版本。不過，無論 Log4j 版本為何，只要您執行的是 R20211203-P2 或更新版本，您的網域都包含處理 CVE-2021-44228 和 CVE-2021-45046 所需的 Log4j 更新。	2021 年 12 月 15 日
跨叢集複寫	跨叢集複寫可讓您將索引、映射和中繼資料從一個 OpenSearch Service 網域複寫到另一個網域。跨叢集複寫需要執行 Elasticsearch 7.10 或 OpenSearch 1.1 或更新版本的網域。	2021 年 10 月 5 日

新 AWS受管政策	Amazon OpenSearch Service 的推出包括新的 AWS受管政策和舊政策的棄用。	2021 年 9 月 8 日
支援 Kibana 6.4.3	執行舊版 Elasticsearch 6.4 版的 Amazon OpenSearch Service 網域現在支援 Kibana 6.4 的最新修補程式版本，它可新增錯誤修正並改善安全性。OpenSearch Service 會自動將網域升級至此修補程式版本。	2021 年 9 月 8 日
資料串流	Amazon OpenSearch Service 新增了對資料串流的支援，可簡化管理時間序列資料的程序。您的網域必須執行 OpenSearch 1.0 或更新版本才能使用資料串流。	2021 年 9 月 8 日
Amazon OpenSearch Service	AWS 重新命名 Amazon OpenSearch Service 以移除舊版「Elasticsearch」品牌。Amazon OpenSearch Service 支援 OpenSearch 和舊版 Elasticsearch OSS。在您建立叢集時，您可選擇要使用的搜尋引擎。OpenSearch Service 提供與 Elasticsearch OSS 7.10 的廣泛相容性，它是該軟體的最終開源版本。	2021 年 9 月 8 日

冷儲存	冷儲存是用於不常存取的資料或歷史資料的新儲存層。冷索引僅佔用 S3 儲存體，並且沒有連接任何運算。冷存儲需要一個執行 Elasticsearch 7.9 或更新版本的網域以及服務軟體 R20210426 或更新版本。	2021 年 5 月 13 日
以 ARM 為基礎的 Graviton 執行個體	Amazon OpenSearch Service 現在支援以 ARM 為基礎的 Graviton 執行個體類型 (M6G、C6G、R6G 和 R6GD)。Graviton 執行個體類型適用於執行 Elasticsearch 7.9 或更新版本的新網域和現有網域，以及服務軟件 R20210331 或更新版本。	2021 年 5 月 4 日
ISM 範本	Amazon OpenSearch Service 新增了對 ISM 範本的支援，可讓您在索引符合政策中定義的模式時，自動將 ISM 政策連接至索引。ISM 範本需要服務軟體 R20210426 或更新版本。此更新也會取代 `policy_id` 設定，這表示您無法再使用索引範本將 ISM 政策套用至新建立的索引。該更新對使用此設定的現有 CloudFormation 範本引入了重大變更。	2021 年 4 月 27 日
支援 Elasticsearch 7.10	Amazon OpenSearch Service 現在支援 Elasticsearch 7.10 版。如需詳細資訊，請參閱 7.10 版本備註。	2021 年 4 月 21 日

非同步搜尋	Amazon OpenSearch Service 現在支援非同步搜尋，讓您可以在後台執行搜尋請求。非同步搜尋需要一個執行 Elasticsearch 7.10 或更新版本的網域以及服務軟體 R20210331 或更新版本。	2021 年 4 月 21 日
組態 API 的標籤型存取控制	您現在可以使用 AWS 標籤來控制對 Amazon ES 組態 API 的存取。	2021 年 3 月 2 日
自動調校	Amazon OpenSearch Service 新增了自動調整，它使用叢集中的效能和使用指標來建議節點上 JVM 設定的變更。自動調整需要一個執行 Elasticsearch 6.7 或更新版本的網域以及服務軟體 R20201117 或更新版本。	2021 年 2 月 24 日
Trace Analytics	Amazon OpenSearch Service 的 Kibana 預設安裝現在包含追蹤分析外掛程式，可讓您從分散式應用程式監控追蹤資料。外掛程式需要一個執行 Elasticsearch 7.9 或更新版本的網域以及服務軟體 R20210201 或更新版本。	2021 年 2 月 17 日

碎片指標	Amazon OpenSearch Service 新增以下 CloudWatch 指標，用於追蹤碎片狀態：Shards.active 、Shards.unassigned 、Shards.delayedUnassigned Shards.activePrimary 、Shards.initializing 、Shards.relocating 。可在執行服務軟體 R20210201 或更新版本的網域上取得這些指標。	2021 年 2 月 17 日
Kibana 報告	Amazon OpenSearch Service 的 Kibana 預設安裝現在支援 Discover (探索)、Visualize (視覺化) 和 Dashboard (儀表板) 頁面的隨需報告。此功能需要 Elasticsearch 7.9 或更新版本以及服務軟體 R20210201 或更新版本。	2021 年 2 月 17 日
支援 Kibana 5.6.16	執行 Elasticsearch 5.6 的 Amazon OpenSearch Service 網域現在支援 Kibana 5.6 的最新修補程式版本，它可新增錯誤修正並改善安全性。Amazon ES 會自動將網域升級至此修補程式版本。	2021 年 2 月 17 日

現有網域的加密	Amazon OpenSearch Service 現在支援在執行 Elasticsearch 6.7 或更新版本的現有網域上啟用靜態資料加密和節點對節點加密。啟用這些設定後，您便無法加以停用。	2021 年 1 月 27 日
遠端重新索引	Amazon OpenSearch Service 現在支援遠端重新索引，可讓您從遠端網域遷移索引。此功能需要服務軟體 R20201117 或更新版本。	2020 年 11 月 24 日
Piped Processing Language	Amazon OpenSearch Service 現在支援 Piped Processing Language (PPL)，它是一種查詢語言，可讓您使用管道 (|) 語法來查詢存放在 Elasticsearch 中的資料。此功能需要服務軟體 R20201117 或更新版本。如需詳細資訊，請參閱 。	2020 年 11 月 24 日
Kibana 筆記本	Amazon OpenSearch Service 新增了對 Kibana 筆記本的支援，可讓您在單一介面中結合即時視覺效果和敘述性文字。此功能需要服務軟體 R20201117 或更新版本。	2020 年 11 月 24 日
甘特圖	Amazon OpenSearch Service 的 Kibana 預設安裝現在支援新的視覺化類型：甘特圖。此功能需要服務軟體 R20201117 或更新版本。	2020 年 11 月 24 日

支援 Elasticsearch 7.9	Amazon OpenSearch Service 現在支援 Elasticsearch 7.9 版。如需詳細資訊，請參閱 7.9 版本備註。	2020 年 11 月 24 日
異常偵測更新	Amazon OpenSearch Service 的異常偵測新增了對高基數的支援，可讓您使用 IP 地址、產品 ID、國家/地區代碼等維度來分類異常。此功能需要服務軟體 R20201117 或更新版本。	2020 年 11 月 24 日
動態字典更新	Amazon OpenSearch Service 現在可讓您在不重新索引的情況下更新搜尋分析器。您可以更新部分或全部網域上的字典檔案，Amazon ES 會隨著時間追蹤套件版本，以便您擁有變更內容和變更時間的歷史記錄。此功能需要服務軟體 R20201019 或更新版本。	2020 年 11 月 17 日
自訂端點	Amazon OpenSearch Service 現在支援自訂端點，它可讓您為您的 Amazon ES 網域提供一個新的 URL。如果您曾經交換網域，您可以維護相同的 URL。此功能需要服務軟體 R20201019 或更新版本。	2020 年 11 月 5 日
新的語言外掛程式	Amazon OpenSearch Service 現在支援使用服務軟體 R20201019 或更新版本執行 Elasticsearch 7.7 或更新版本的網域上的 IK (中文) 分析、越南文分析和泰文分析外掛程式。	2020 年 10 月 28 日

支援 Elasticsearch 7.8	Amazon OpenSearch Service 現在支援 Elasticsearch 7.8 版。如需詳細資訊，請參閱 7.8 版本備註。	2020 年 10 月 28 日
Kibana 的 SAML 身分驗證	Amazon OpenSearch Service 現在支援 Kibana 的 SAML 身分驗證，可讓您使用第三方身分提供者登入 Kibana、管理精細存取控制、搜尋資料以及建立視覺效果。此功能需要服務軟體 R20201019 或更新版本。	2020 年 10 月 27 日
T3 執行個體	Amazon OpenSearch Service 現在支援 `t3.small` 和 `t3.medium` 執行個體類型。	2020 年 9 月 23 日
稽核日誌	Amazon OpenSearch Service 現在支援資料的稽核日誌，可讓您追蹤失敗的登入嘗試，以及使用者對索引、文件、欄位等的存取。此功能需要服務軟體 R20200910 或更新版本。	2020 年 9 月 16 日
UltraWarm 更新	Amazon OpenSearch Service 的 UltraWarm 新增了新指標、新設定、更大的遷移佇列和取消 API。這些更新需要服務軟體 R20200910 或更新版本。如需詳細資訊，請參閱。	2020 年 9 月 14 日
Learning to Rank	Amazon OpenSearch Service 現在支援開放原始碼 Learning to Rank 外掛程式，可讓您使用機器學習技術來改善搜尋相關性。此功能需要服務軟體 R20200721 或更新版本。	2020 年 7 月 27 日

k-NN 餘弦相似度	除了歐幾里德距離外，K 近鄰 (k-NN) 現在允許您按照餘弦相似度來搜尋「近鄰」。此功能需要服務軟體 R20200721 或更新版本。	2020 年 7 月 23 日
gzip 壓縮	Amazon OpenSearch Service 現在支援大多數 HTTP 請求和回應的 gzip 壓縮，這可以減少延遲並節省頻寬。此功能需要服務軟體 R20200721 或更新版本。	2020 年 7 月 23 日
支援 Elasticsearch 7.7	Amazon OpenSearch Service 現在支援 Elasticsearch 7.7 版。如需詳細資訊，請參閱 7.7 版本備註。	2020 年 7 月 23 日
Kibana 地圖服務	Amazon OpenSearch Service 的 Kibana 預設安裝現在包含 WMS 地圖伺服器，但印度和中國區域的網域除外。	2020 年 6 月 18 日
SQL 改進	Amazon OpenSearch Service 的 SQL 支援現在可支援許多新操作、用於資料探索的專用 Kibana 使用者介面以及互動式 CLI。	2020 年 6 月 3 日
跨叢集搜尋	Amazon OpenSearch Service 可讓您跨多個連線的網域執行跨叢集查詢和匯總。	2020 年 6 月 3 日
異常偵測	Amazon OpenSearch Service 可讓您以近乎即時的速度自動偵測異常。	2020 年 6 月 3 日

UltraWarm	Amazon OpenSearch Service 的 UltraWarm 儲存已開放預覽，現已正式推出。此功能現在支援更廣泛的版本 和 AWS 區域。如需詳細資訊，請參閱。	2020 年 5 月 5 日
自訂字典	Amazon OpenSearch Service 可讓您上傳自訂字典檔案，以便與叢集搭配使用。字典檔案透過讓 Elasticsearch 忽略某些高頻詞或將術語視為相同字詞，來改善您的搜尋結果。	2020 年 4 月 21 日
支援 Elasticsearch 7.4	Amazon OpenSearch Service 現在支援 Elasticsearch 7.4 版。如需詳細資訊，請參閱支援的版本。	2020 年 3 月 12 日
k-NN	Amazon OpenSearch Service 新增了對 K 近鄰 (k-NN) 搜尋的支援。K-NN 需要服務軟體 R20200302 或更新版本。	2020 年 3 月 3 日
索引狀態管理	Amazon OpenSearch Service 新增了索引狀態管理 (ISM)，可讓您自動化例行任務，例如在達到特定使用期限時刪除索引。此功能需要服務軟體 R20200302 或更新版本。	2020 年 3 月 3 日

支援 Elasticsearch 5.6.16	Amazon OpenSearch Service 現在支援 5.6 版的最新修補程式版本，它可新增錯誤修正並改善安全性。Amazon ES 會自動將現有 5.6 網域升級至此版本。請注意，此 Elasticsearch 版本會錯誤地將其版本報告為 5.6.17。	2020 年 3 月 2 日
精細定義存取控制	Amazon OpenSearch Service 現在支援精細存取控制，這種存取控制可為您的叢集提供索引、文件和欄位層級、Kibana 多租用以及選用的 HTTP 基本身分驗證的安全性。	2020 年 2 月 11 日
UltraWarm 儲存 (預覽版)	Amazon OpenSearch Service 新增了 UltraWarm，這是一個全新暖儲存層級，它使用 Amazon S3 和精密的快取解決方案來改善效能。對於您未積極寫入且較不常查詢的索引，UltraWarm 儲存大幅降低了每 GiB 的費用。	2019 年 12 月 3 日
適用於中國區域的加密功能	靜態資料加密和節點對節點加密現在適用於 `cn-north-1` 中國 (北京) 和 `cn-northwest-1` 中國 (寧夏) 區域。	2019 年 11 月 20 日
需要 HTTPS	您現在可以要求所有流入您 Amazon ES 網域的流量都透過 HTTPS 到達。設定網域時，請勾選 Require HTTPS (需要 HTTPS) 方塊。此功能需要服務軟體 R20190808 或更新版本。	2019 年 10 月 3 日

支援 Elasticsearch 7.1 和 6.8	Amazon OpenSearch Service 現在支援 Elasticsearch 7.1 和 6.8 版。如需詳細資訊，請參閱支援的版本。	2019 年 8 月 13 日
每小時快照	Amazon OpenSearch Service 會為執行 Elasticsearch 5.3 和更新版本的網域每小時拍攝一次快照，而不是每日拍攝一次快照，以便您能更頻繁地備份資料，以從中還原資料。	2019 年 7 月 8 日
支援 Elasticsearch 6.7	Amazon OpenSearch Service 現在支援 Elasticsearch 6.7 版。如需詳細資訊，請參閱支援的版本。	2019 年 5 月 29 日
SQL 支援	Amazon OpenSearch Service 現在可讓您使用 SQL 查詢資料。SQL 支援需要服務軟體 R20190418 或更新版本。	2019 年 5 月 15 日
5 系列執行個體類型	Amazon OpenSearch Service 現在支援 M5、C5 和 R5 執行個體類型。相較於上一代執行個體類型，這些新類型價格較低，效能更優異。如需詳細資訊，請參閱限制。	2019 年 4 月 24 日
支援 Elasticsearch 6.5	Amazon OpenSearch Service 現在支援 Elasticsearch 6.5 版。	2019 年 4 月 8 日

提醒	當一個或多個 Amazon ES 索引中的資料符合特定條件時，Amazon OpenSearch Service 的提醒功能就會向您發出通知。提醒需要服務軟體 R20190221 或更新版本。	2019 年 3 月 25 日
支援三個可用區域	Amazon OpenSearch Service 現於許多區域支援三種可用區域。此版本還包含簡化的主控台體驗。此多可用區域需要服務軟體 R20181023 或更新版本。	2019 年 2 月 7 日
支援 Elasticsearch 6.4	Amazon OpenSearch Service 現在支援 Elasticsearch 6 .4 版。	2019 年 1 月 23 日
200 個節點叢集	Amazon ES 現在可讓您建立擁有高達 200 個資料節點的叢集，總共 3 PB 的儲存空間。	2019 年 1 月 22 日
服務軟體更新	Amazon ES 如今讓您能夠手動更新網域的服務軟體，從而更快地受益於新功能或選擇在低流量的時段進行更新。如需詳細資訊，請參閱 。	2018 年 11 月 20 日
新的 CloudWatch 指標	在 Amazon ES 主控台中，Amazon ES 現在提供節點層級指標和新的叢集運作狀態和執行個體運作狀態索引標籤。	2018 年 11 月 20 日
支援中國 (北京)	Amazon OpenSearch Service 在 cn-north-1 區域現已可供使用，且支援 M4、C4 和 R4 執行個體類型。	2018 年 10 月 17 日

節點對節點加密	Amazon OpenSearch Service 現已支援節點對節點加密，當 Amazon ES 將資料分發至叢集時，它可維持資料加密。	2018 年 9 月 18 日
就地版本升級	Amazon OpenSearch Service 現在支援就地版本升級。	2018 年 8 月 14 日
支援 Elasticsearch 6.3 和 5.6	Amazon OpenSearch Service 現在支援 Elasticsearch 6.3 和 5.6 版。	2018 年 8 月 14 日
錯誤日誌	Amazon ES 現在可讓您發布 Elasticsearch 錯誤日誌到 Amazon CloudWatch。	2018 年 7 月 31 日
中國 (寧夏) 預留執行個體	Amazon ES 目前在中國 (寧夏) 區域提供預留執行個體。	2018 年 5 月 29 日
預留執行個體	Amazon ES 現在提供對預留執行個體的支援。	2018 年 5 月 7 日

舊版更新

下表說明 2018 年 5 月前 Amazon ES 的重要變更。

變更	描述	Date
Kibana 的 Amazon Cognito 身分驗證	Amazon ES 現在提供 Kibana 的登入頁面保護。如需詳細資訊，請參閱 the section called “OpenSearch Dashboards 的 Amazon Cognito 身分驗證”。	2018 年 4 月 2 日
支援 Elasticsearch 6.2	Amazon OpenSearch Service 現在支援 Elasticsearch 6.2 版。	2018 年 3 月 14 日
韓文分析外掛程式	Amazon ES 現在支援 Seunjeon 韓文分析外掛程式的記憶體最佳化版本。	2018 年 3 月 13 日

變更	描述	Date
立即存取控制更新	Amazon ES 網域的存取控制政策變更現在會立即生效。	2018 年 3 月 7 日
PB 規模	Amazon ES 現在支援 I3 執行個體類型且網域總儲存量最多為 1.5 PB。如需詳細資訊，請參閱 the section called “PB 規模”。	2017 年 12 月 19 日
靜態資料加密	Amazon ES 現在支援靜態資料加密。如需詳細資訊，請參閱 the section called “靜態加密”。	2017 年 12 月 7 日
支援 Elasticsearch 6.0	Amazon ES 現在支援 Elasticsearch 6.0 版。如需遷移的考量和說明，請參閱the section called “升級網域”。	2017 年 12 月 6 日
VPC 支援	Amazon ES 現在可讓您在 Amazon Virtual Private Cloud 中啟動網域。VPC 支援提供多一層的安全性並可簡化 Amazon ES 和 VPC 內其他服務之間的通訊。如需詳細資訊，請參閱 the section called “VPC 支援”。	2017 年 10 月 17 日
慢速日誌發佈	Amazon ES 現在支援將慢速日誌發佈至 CloudWatch Logs。如需詳細資訊，請參閱 the section called “監控日誌”。	2017 年 10 月 16 日
支援 Elasticsearch 5.5	Amazon ES 現在支援 Elasticsearch 5.5 版。 您現在可以還原自動快照而無需聯絡 支援 ，以及使用 `_scripts` API 儲存指令碼。	2017 年 9 月 7 日
支援 Elasticsearch 5.3	Amazon ES 已新增對 Elasticsearch 5.3 版的支援。	2017 年 6 月 1 日
每個叢集更多執行個體和 EBS 容量	Amazon ES 現在支援高達 100 個節點以及每個叢集 150 TB 的 EBS 容量。	2017 年 4 月 5 日
加拿大 (中部) 和歐洲 (倫敦) 支援	Amazon ES 已新增對下列區域的支援：加拿大 (中部)、ca-central-1 和歐洲 (倫敦)、eu-west-2。	2017 年 3 月 20 日

變更	描述	Date
更多執行個體和更大的 EBS 磁碟區	Amazon ES 已新增支援更多執行個體和更大的 EBS 磁碟區。	2017 年 2 月 21 日
支援 Elasticsearch 5.1	Amazon ES 已新增對 Elasticsearch 5.1 版的支援。	2017 年 1 月 30 日
支援語音分析外掛程式	Amazon ES 現在提供內建的語音分析外掛程式整合，其可讓您對您的資料執行「類似聲音」的查詢。	2016 年 12 月 22 日
美國東部 (俄亥俄) 支援	Amazon ES 已新增支援以下區域：美國東部 (俄亥俄)、us-east-2。	2016 年 10 月 17 日
新的效能指標	Amazon ES 新增效能指標 `ClusterUsedSpace` 。	2016 年 7 月 29 日
支援 Elasticsearch 2.3	Amazon ES 已新增對 Elasticsearch 2.3 版的支援。	2016 年 7 月 27 日
亞太區域 (孟買) 支援	Amazon ES 已新增對以下區域的支援：亞太區域 (孟買)、ap-south-1。	2016 年 6 月 27 日
每個叢集更多執行個體	Amazon ES 已將每個叢集的執行個體數目上限 (執行個體計數) 從 10 提高到 20。	2016 年 5 月 18 日
亞太區域 (首爾) 支援	Amazon ES 已新增支援以下區域：亞太區域 (首爾)、ap-northeast-2。	2016 年 1 月 28 日
Amazon ES	初始版本。	2015 年 10 月 1 日

AWS 詞彙表

如需最新的 AWS 術語，請參閱 AWS 詞彙表 參考中的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。