aws

개발자 가이드

Amazon OpenSearch Service

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon OpenSearch Service: 개발자 가이드

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께 사용되어서는 안되며, 고객에게 혼동을 일으키거나 Amazon 브랜드 이미지를 떨어뜨리고 폄하하는 방식으로 이용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Amazon OpenSearch Service란 무엇입니까? 1
Amazon OpenSearch Service의 기능 1
사용해야 하는 경우 3
지원되는 Elasticsearch 및 OpenSearch 버전 3
표준 및 확장 지원 4
표준 지원 및 확장 버전 6
추가 지원 요금 계산 6
요금 7
관련 서비스 8
설정 10
권한 부여 10
프로그래밍 방식 액세스 권한 부여 10
설정 AWS CLI 12
콘솔을 엽니다. 12
시작 13
도메인 생성 13
인덱싱을 위한 데이터 업로드 15
옵션 1: 단일 문서 업로드 15
옵션 2: 여러 문서 업로드 16
문서 검색 17
명령줄에서 문서 검색 17
OpenSearch Dashboards를 사용하여 문서 검색 18
도메인 삭제 19
Amazon OpenSearch Ingestion 20
주요 개념 20
이점 21
제한 사항 22
지원되는 Data Prepper 버전 22
파이프라인 크기 조정 23
요금 24
지원됨 AWS 리전 25
역할 및 사용자 설정 25
파이프라인 역할 26
수집 역할 29

도메인에 대한 파이프라인 액세스 권한 부여 30
컬렉션에 대한 액세스 권한을 파이프라인에 부여 34
OpenSearch Ingestion 시작하기 38
튜토리얼: 도메인에 데이터 수집 39
튜토리얼: 컬렉션에 데이터 수집 46
Pipeline features 53
영구 버퍼링 53
분할 55
Chaining 56
배달 못한 편지 대기열 57
인덱스 관리 59
엔드 투 엔드 승인 62
소스 역압 63
파이프라인 생성 64
사전 조건 및 필요한 IAM 역할 64
필수 IAM 권한 65
파이프라인 버전 지정 66
수집 경로 지정 67
파이프라인 생성 67
파이프라인 생성 상태 추적 72
블루프린트 작업 73
파이프라인 보기 75
파이프라인 업데이트 78
고려 사항 78
필요한 권한 79
파이프라인 업데이트 80
파이프라인 업데이트를 위한 블루/그린 배포 80
파이프라인 비용 관리 81
파이프라인 중지 82
파이프라인 시작 83
파이프라인 삭제 83
지원되는 작업 및 플러그인 84
지원되는 플러그인 85
상태 비저장 프로세서와 상태 저장 프로세서 비교 87
구성 요구 사항 및 제약 조건 88
파이프라인 통합 93

수집 엔드포인트 구성 93
수집 역할 생성 93
Atlassian 서비스 95
Amazon Aurora 112
Amazon DynamoDB 140
Amazon DocumentDB 154
Confluent Cloud Kafka 170
Amazon MSK 180
Amazon RDS 187
Amazon S3 215
Amazon Security Lake 225
Fluent Bit 229
Fluentd 231
OpenTelemetry Collector 233
자체 관리형 Kafka 235
자체 관리형 OpenSearch 클러스터 242
Amazon Kinesis Data Streams 249
다음 단계 261
AWS Lambda 262
도메인과 컬렉션 간 데이터 마이그레이션 265
제한 사항 266
OpenSearch Service를 소스로 사용 266
여러 OpenSearch Service 도메인 싱크 지정 268
OpenSearch Serverless VPC 컬렉션으로 데이터 마이그레이션 269
AWS SDKs를 사용하여 파이프라인 관리 270
Python 270
OpenSearch Ingestion의 보안 274
파이프라인에 대한 VPC 액세스 구성 275
ID 및 액세스 관리 279
CloudTrail을 사용한 모니터링 287
파이프라인 태그 지정 291
필요한 권한 292
태그 작업(콘솔) 292
태그 작업(AWS CLI) 292
로깅 및 모니터링 293
파이프라인 모니터링 293

파이프라인 지표 모니터링 295
모범 사례 324
일반 모범 사례 325
권장되는 CloudWatch 경보 325
Amazon OpenSearch Serverless 331
이점 331
Amazon OpenSearch Serverless란 무엇인가요? 331
OpenSearch Serverless 사용 사례 332
작동 방법 333
컬렉션 유형 선택 334
요금 334
지원됨 AWS 리전 335
제한 사항 336
OpenSearch Service와 OpenSearch Serverless 비교 336
자습서: OpenSearch Serverless 시작하기 340
1단계: 권한 구성 340
2단계: 컬렉션 생성 341
3단계: 데이터 업로드 및 검색 342
4단계: 컬렉션 삭제 343
다음 단계 343
컬렉션 생성 및 관리 344
컬렉션 관리 344
벡터 검색 컬렉션 작업 362
데이터 수명 주기 정책 사용 369
AWS SDKs를 사용하여 컬렉션 관리 378
CloudFormation을 사용하여 컬렉션 생성 389
스냅샷을 사용하여 컬렉션 백업 391
용량 제한 관리 397
용량 설정 구성 399
최대 용량 제한 399
용량 사용량 모니터링 400
컬렉션으로 데이터 수집 400
최소 필수 권한 400
OpenSearch Ingestion 401
Fluent Bit 402
Amazon Data Firehose 402

Go ... 403
Java ... 405
JavaScript ... 407
Logstash ... 409
Python ... 412
Ruby ... 413
기타 클라이언트 ... 414
Machine Learning 구성 ... 415
기계 학습 ... 415
커넥터 ... 415
모델 ... 416
Machine Learning에 대한 권한 구성 ... 416
지원되지 않는 APIs 및 기능 ... 417
신경 및 하이브리드 검색 구성 ... 418
신경망 검색 ... 418
하이브리드 검색 ... 418
신경 및 하이브리드 쿼리 ... 419
권한 구성 ... 420
워크플로 구성 ... 423
워크플로 ... 423
권한 구성 ... 423
OpenSearch Serverless의 보안 ... 424
암호화 정책 ... 426
네트워크 정책 ... 426
데이터 액세스 정책 ... 426
IAM 및 SAML 인증 ... 427
인프라 보안 ... 428
보안 시작하기 ... 428
ID 및 액세스 관리 ... 442
암호화 ... 459
네트워크 액세스 ... 468
데이터 액세스 제어 ... 479
VPC 엔드포인트 ... 490
SAML 인증 ... 501
규정 준수 확인 ... 509
컬렉션 태그 지정 ... 510

필요한 권한 511
컬렉션 태그 지정(콘솔) 511
컬렉션 태그 지정(AWS CLI) 511
지원되는 작업 및 플러그인 512
지원되는 OpenSearch API 작업 및 권한 512
지원되는 OpenSearch 플러그인 522
OpenSearch Serverless 모니터링 523
CloudWatch를 사용하여 모니터링 524
CloudTrail을 사용한 모니터링 529
EventBridge로 모니터링 531
도메인 생성 및 관리 535
OpenSearch Service 도메인 생성 535
OpenSearch Service 도메인(콘솔) 생성 535
OpenSearch Service 도메인 생성(AWS CLI) 541
OpenSearch Service 도메인(AWS SDKs) 생성 543
OpenSearch Service 도메인 생성(AWS CloudFormation) 543
액세스 정책 구성 543
고급 클러스터 설정 544
구성 변경 545
블루/그린 배포의 원인이 되는 변경 사항 545
블루/그린 배포가 발생하지 않는 변경 사항 546
변경 사항으로 인해 블루/그린 배포가 발생하는지 판단 547
구성 변경 추적 551
구성 변경 단계 553
블루/그린 배포의 성능 영향 555
구성 변경 비용 556
Troubleshooting validation errors(검증 오류 문제 해결 중) 556
서비스 소프트웨어 업데이트 561
선택적 업데이트와 필수 업데이트 비교 561
패치 업데이트 562
고려 사항 562
업그레이드 시작 563
사용량이 적은 기간 566
업데이트 모니터링 567
도메인이 업데이트에 적합하지 않은 경우 568
사용량이 적은 기간 568

사용량이 적은 기간 서비스 소프트웨어 업데이트 ... 569
사용량이 적은 자동 조정 최적화 ... 569
사용량이 적은 시간 활성화하기 ... 570
사용량이 적은 사용자 지정 기간 구성 ... 571
예약된 작업 보기 ... 571
작업 일정 조정 ... 573
자동 조정 유지 관리 기간에서 마이그레이션하기 ... 575
알림 ... 576
알림 시작하기 ... 576
알림 심각도 ... 577
샘플 EventBridge 이벤트 ... 578
다중 AZ 도메인 구성 ... 578
Multi-AZ with Standby ... 579
Multi-AZ without Standby ... 580
가용 영역 중단 ... 582
VPC 지원 ... 583
VPC 대 퍼블릭 도메인 ... 583
제한 사항 ... 584
아키텍처 ... 584
인덱스 스냅샷 생성 ... 590
사전 조건 ... 591
수동 스냅샷 리포지토리 등록 ... 595
수동 스냅샷 생성 ... 599
스냅샷 복원 ... 601
수동 스냅샷 삭제 ... 603
Snapshot Management를 사용한 스냅샷 자동화 ... 603
인덱스 상태 관리를 사용한 스냅샷 자동화 ... 605
스냅샷에 Curator 사용 ... 606
도메인 업그레이드 ... 606
지원되는 업그레이드 경로 ... 607
도메인 업그레이드(콘솔) ... 609
도메인 업그레이드(CLI) ... 610
도메인 업그레이드(SDK) ... 611
검증 장애 문제 해결 ... 612
업그레이드 문제 해결 ... 613
스냅샷을 사용하여 데이터 마이그레이션 ... 615

사용자 지정 엔드포인트 만들기 ... 622
새 도메인에 대한 사용자 지정 엔드포인트 ... 622
기존 도메인에 대한 사용자 지정 엔드포인트 ... 623
CNAME 매핑 ... 624
자동 조정 ... 624
변경 유형 ... 625
자동 조정 활성화 또는 비활성화 ... 626
자동 조정 강화 예약 ... 627
자동 조정 변경 사항 모니터링 ... 628
도메인 태그 지정 ... 628
태그 예제 ... 628
도메인 태그 지정(콘솔) ... 629
도메인 태그 지정(AWS CLI) ... 630
도메인 태그 지정(AWS SDKs) ... 631
관리 작업 수행 ... 632
데이터 노드에서 OpenSearch 프로세스 다시 시작 ... 633
데이터 노드 재부팅 ... 633
대시보드 프로세스 다시 시작 ... 634
제한 사항 ... 635
직접 쿼리에 대한 작업 ... 636
요금 ... 636
제한 사항 ... 637
일반 제한 사항 ... 637
Amazon S3에 대한 제한 사항 ... 637
Amazon CloudWatch Logs에 대한 제한 사항 ... 638
Amazon Security Lake에 대한 제한 사항 ... 638
추천 ... 638
일반 권장 사항 ... 639
Amazon S3 권장 사항 ... 639
CloudWatch Logs 권장 사항 ... 640
Security Lake 권장 사항 ... 640
할당량 ... 641
Amazon S3 할당량 ... 641
CloudWatch Logs 할당량 ... 642
Security Lake 할당량 ... 643
지원됨 AWS 리전 ... 644

Amazon S3 AWS 리전 에서 사용 가능 ... 644
CloudWatch Logs에 사용 가능 AWS 리전 ... 645
Security Lake에 사용 가능 AWS 리전 ... 645
S3의 직접 쿼리 ... 646
S3 데이터 소스 생성 ... 646
S3 데이터 소스 구성 ... 654
CloudWatch Logs의 직접 쿼리 ... 656
CloudWatch Logs 데이터 소스 생성 ... 656
CloudWatch Logs 데이터 소스 구성 ... 662
Security Lake의 직접 쿼리 ... 663
Security Lake 데이터 소스 생성 ... 664
Security Lake 데이터 소스 구성 ... 670
데이터 소스 관리 ... 672
CloudWatch 지표 데이터 소스를 사용한 모니터링 ... 673
데이터 소스 활성화 및 비활성화 ... 675
AWS Budget을 사용한 모니터링 ... 676
데이터 소스 삭제 ... 676
쿼리 성능 최적화 ... 677
건너뛰기 인덱스 ... 678
구체화된 뷰 ... 678
커버링 인덱스 ... 679
지원되는 SQL 및 PPL 명령 ... 679
지원되는 SQL 명령 ... 680
지원되는 PPL 명령 ... 875
도메인 모니터링 ... 1051
클러스터 지표 모니터링 ... 1052
CloudWatch에서 지표 보기 ... 1053
OpenSearch Service의 상태 차트 해석 ... 1053
클러스터 지표 ... 1054
전용 프라이머리 노드 지표입니다. ... 1061
전용 조정자 노드 지표 ... 1062
EBS 볼륨 지표입니다. ... 1063
인스턴스 지표 ... 1065
UltraWarm 지표 ... 1077
콜드 스토리지 지표 ... 1082
OR1 지표 ... 1083

알림 지표 1083
이상 탐지 지표 1085
비동기 검색 지표 1086
지표 자동 조정 1088
Multi-AZ with Standby 지표 1089
특정 시점 지표 1091
SQL 지표 1092
k-NN 지표 1093
클러스터 간 검색 지표 1096
클러스터 간 복제 지표 1096
순위 학습 지표 1098
파이프 처리 언어 지표 1099
로그 모니터링 1099
로그 게시 활성화(콘솔) 1101
로그 게시 활성화(AWS CLI) 1103
로그 게시 활성화(AWS SDK) 1105
로그 게시 활성화(CloudFormation) 1105
느린 검색 요청 로그 임계치 설정 1107
느린 샤드 로그 임계치 설정 1108
느린 로그 테스트 1108
로그 보기 1109
감사 로그 모니터링 1109
제한 사항 1110
감사 로그 활성화 1110
를 사용하여 감사 로깅 활성화 AWS CLI 1112
구성 API를 사용하여 감사 로깅 활성화 1113
감사 로그 계층 및 범주 1113
감사 로그 설정 1115
감사 로그 예제 1118
REST API를 사용하여 감사 로그 구성 1121
이벤트 모니터링 1122
서비스 소프트웨어 업데이트 이벤트 1123
이벤트 자동 조정 1130
클러스터 상태 이벤트 1135
VPC 엔드포인트 이벤트 1148
노드 만료 이벤트 1150

성능 저하된 노드 사용 중지 이벤트 1152
도메인 오류 이벤트 1154
자습서: OpenSearch Service 이벤트 수신 1156
자습서: 사용 가능한 업데이트에 대한 SNS 알림 보내기 1158
CloudTrail을 사용한 모니터링 1160
CloudTrail 의 Amazon OpenSearch Service 정보 529
Amazon OpenSearch Service 로그 파일 항목 이해 530
보안 1164
데이터 보호 1165
저장 시 암호화 1166
노드 간 암호화 1169
ID 및 액세스 관리 1170
정책 유형 1170
OpenSearch Service 요청 작성 및 서명 1180
정책 충돌 시 1181
정책 요소 참조 1182
고급 옵션 및 API 고려 사항 1187
액세스 정책 구성 1190
추가 샘플 정책 1190
API 권한 참조 1191
AWS 관리형 정책 1191
교차 서비스 혼동된 대리자 방지 1200
세분화된 액세스 제어 1201
개요: 세분화된 액세스 제어와 OpenSearch Service 보안 1202
주요 개념 1206
마스터 사용자 정보 1206
세분화된 액세스 제어 활성화 1207
마스터 사용자로 OpenSearch 대시보드에 액세스 1210
권한 관리 1211
권장 구성 1215
제한 사항 1218
마스터 사용자 수정 1219
추가 마스터 사용자 1220
수동 스냅샷 수 1221
통합 1221
REST API 차이점 1222

자습서: Cognito 인증을 사용한 세분화된 액세스 제어 1224
자습서: 기본 인증을 사용하는 내부 사용자 데이터베이스 1228
규정 준수 확인 1231
복원성 1232
JSON 웹 토큰 1233
고려 사항 1233
도메인 액세스 정책 수정 1233
JWT 인증 및 권한 부여 구성 1234
JWT를 사용하여 테스트 요청 전송 1235
인프라 보안 1236
OpenSearch Service 관리형 VPC 엔드포인트 작업 1237
OpenSearch 대시보드에 대한 SAML 인증 1241
SAML 구성 개요 1241
고려 사항 1242
VPC 도메인에 대한 SAML 인증 1242
도메인 액세스 정책 수정 1242
SP 및 IdP 시작 인증 구성 1244
SP 및 IdP 시작 인증 모두 구성 1250
SAML 인증 구성(AWS CLI) 1250
SAML 인증 구성(구성 API) 1251
SAML 문제 해결 1251
SAML 인증 비활성화 1254
OpenSearch에 대한 IAM Identity Center 지원 1255
고려 사항 1255
도메인 액세스 정책 수정 1256
IAM Identity Center 인증 및 권한 부여 구성(콘솔) 1256
세분화된 액세스 제어 구성 1257
IAM Identity Center 인증 및 권한 부여 구성(CLI) 1257
도메인에서 IAM Identity Center 인증 비활성화 1257
OpenSearch Dashboards에 대한 Amazon Cognito 인증 1258
사전 조건 1259
Amazon Cognito 인증을 사용하도록 도메인 구성 1262
인증된 역할 허용 1266
자격 증명 공급자 구성 1267
(선택 사항) 세분화된 액세스 구성 1267
(선택 사항) 로그인 페이지 사용자 지정 1268

(선택 사항) 고급 보안 구성 1268
테스트 1269
할당량 1269
일반적인 구성 문제 1269
OpenSearch Dashboards에 대한 Amazon Cognito 인증 비활성화 1273
OpenSearch Dashboards에 대한 Amazon Cognito 인증을 사용하는 도메인 삭제 1274
서비스 연결 역할 사용 1274
VPC 도메인 및 데이터 소스 생성 역할 1275
컬렉션 생성 역할 1278
파이프라인 생성 역할 1280
샘플 코드 1284
Elasticsearch 클라이언트 호환성 1284
HTTP 요청 압축 1285
gzip 압축 활성화 1285
필수 헤더 1286
샘플 코드(Python 3) 1286
AWS SDKs 사용 1287
Java 1288
Python 1299
노드 1302
데이터 인덱싱 1305
인덱스에 대한 이름 지정 제약 조건 1305
응답 크기 감소 1306
인덱스 코덱 1308
OpenSearch Service로 스트리밍 데이터 로드 1308
OpenSearch Ingestion에서 스트리밍 데이터 로드 1309
Amazon S3에서 스트리밍 데이터 로드 1309
Amazon Kinesis Data Streams에서 스트리밍 데이터 로드 1313
Amazon DynamoDB에서 스트리밍 데이터 로드 1318
Amazon Kinesis Data Firehose에서 스트리밍 데이터 로드 1323
Amazon CloudWatch에서 스트리밍 데이터 로드 1323
에서 스트리밍 데이터 로드 AWS IoT 1324
Logstash를 사용하여 데이터 로딩 1324
구성 1324
데이터 검색 1327
URI 검색 1327

요청 본문 검색 1329
필드 부스팅 1331
검색 결과 강조 표시 1331
Count API 1333
검색 결과 페이지 매김 1334
특정 시점 1334
from 파라미터를 size 추가합니다. 1334
Dashboards Query Language 1334
패키지 1336
필수 권한 1337
Amazon S3에 패키지 업로드 1338
패키지 가져오기 및 연결 1338
OpenSearch에서 사용자 정의 패키지 사용 1339
사용자 지정 패키지 업데이트 1343
새 사전으로 인덱스 수동 업데이트 1347
패키지 분리 및 제거 1349
사용자 지정 플러그인 1350
타사 플러그인 1365
SQL 지원 1369
샘플 호출 1371
참고 사항 및 차이점 1371
SQL Workbench 1372
SQL CLI 1235
JDBC 드라이버 1372
ODBC 드라이버 1372
클러스터 간 검색 1373
제한 사항 1374
클러스터 간 검색 전제 조건 1374
클러스터 간 검색 요금 1374
연결 설정 1374
연결 제거 1375
보안 설정 및 샘플 시연 1376
OpenSearch Dashboards 1382
순위 학습 1382
순위 학습 시작하기 1383
순위 학습 API 1404

비동기 검색 1410
샘플 검색 호출 1410
비동기 검색 권한 1412
비동기 검색 설정 1412
클러스터 간 검색 1413
UltraWarm 1414
특정 시점 1415
고려 사항 1415
PIT 생성 1416
특정 시점 권한 1417
PIT 설정 1418
클러스터 간 검색 1418
UltraWarm 1419
시맨틱 검색 1419
동시 세그먼트 검색 1419
자연어 쿼리 생성 1420
사전 조건 1420
시작 1420
권한 구성 1421
구성 자동화 1421
벡터 검색 1422
(미리 보기) Amazon S3 벡터와 OpenSearch Service 통합 1423
(미리 보기) Amazon S3 벡터에서 OpenSearch Serverless로 가져오기 1423
(미리 보기) Amazon S3 벡터 엔진을 사용한 고급 검색 기능 1430
k-NN 검색 1438
k-NN 시작하기 1438
k-NN의 차이점, 조정, 제한 사항 1441
OpenSearch 대시보드 1442
대시보드에 대한 액세스 제어 1442
프록시를 사용하여 Dashboards에서 OpenSearch Service에 액세스 1443
WMS 맵 서버를 사용하도록 대시보드 구성 1446
로컬 Dashboards 서버를 OpenSearch Service에 연결 1447
대시보드에서 인덱스 관리 1448
기타 기능 1449
OpenSearch UI 1450
릴리스 이력 1451

시작 1452
Amazon OpenSearch Service 애플리케이션을 생성하는 데 필요한 권한 1453
애플리케이션 생성 1456
애플리케이션 관리자 관리 1462
IAM을 사용하여 SAML 페더레이션 활성화 1466
1단계: 자격 증명 공급자 애플리케이션 설정(Okta) 1466
2단계: Okta에 대한 AWS 구성 설정 1469
3단계: IAM에서 Amazon OpenSearch Service 액세스 정책 생성 1471
4단계: SAML을 사용하여 자격 증명 공급자가 시작한 Single Sign-On 환경 확인 1473
5단계: SAML 속성 기반 세분화된 액세스 제어 구성 1476
데이터 소스 연결 및 VPC 액세스 권한 관리 1479
OpenSearch UI 애플리케이션과 데이터 소스 연결 1480
VPC의 도메인에 대한 액세스 관리 1481
VPC에서 OpenSearch Serverless 컬렉션에 대한 액세스 구성 1482
Amazon OpenSearch Service 워크스페이스 사용 1485
OpenSearch UI 애플리케이션 워크스페이스 생성 1485
Workspace 개인 정보 보호 및 공동 작업자 1486
Workspace 유형 1486
클러스터 간 검색을 통한 리전 간 및 계정 간 데이터 액세스 1487
클러스터 간 검색을 사용하여 리전 간 및 계정 간 데이터 액세스에 대한 액세스 권한 설정 .. 1489
도메인 간 연결 생성 1492
교차 클러스터 검색을 사용하여 교차 리전 및 교차 계정 데이터 액세스를 위한 보안 설정 테스트 1493
연결 삭제 1496
VPC 엔드포인트에서 OpenSearch UI에 대한 액세스 관리 1496
VPC와 OpenSearch UI 간에 프라이빗 연결 생성 1497
OpenSearch UI 애플리케이션에 대한 액세스를 허용하도록 VPC 엔드포인트 정책 업데이트 1498
VPC 엔드포인트 정책에서 OpenSearch UI에 대한 액세스 취소 1499
엔드포인트 및 할당량 1499
OpenSearch UI 엔드포인트 1499
OpenSearch UI 서비스 할당량 1502
인덱스 관리 1504
UltraWarm 스토리지 1504
사전 조건 1505
UltraWarm 스토리지 요구 사항 및 성능 고려 사항 1507

UltraWarm 가격 1507
UltraWarm 활성화 1508
인덱스를 UltraWarm 스토리지로 마이그레이션 1510
마이그레이션 자동화 1513
마이그레이션 조정 1513
마이그레이션 취소 1514
핫 인덱스 및 웜 인덱스 나열 1514
핫 스토리지로 웜 인덱스 되돌리기 1514
스냅샷에서 웜 인덱스 복원 1515
웜 인덱스의 수동 스냅샷 1516
콜드 스토리지로 웜 인덱스 마이그레이션 1517
KNN 인덱스 모범 사례 1517
UltraWarm 비활성화 1518
콜드 스토리지 1518
사전 조건 1519
콜드 스토리지 요구 사항 및 성능 고려 사항 1521
콜드 스토리지 요금 1521
콜드 스토리지 활성화 1521
OpenSearch Dashboards의 콜드 인덱스 관리 1523
콜드 스토리지로 인덱스 마이그레이션 1523
콜드 스토리지로 마이그레이션 자동화 1525
콜드 스토리지로의 마이그레이션 취소 1525
콜드 인덱스 목록 표시 1526
웜 스토리지로 콜드 인덱스 마이그레이션 1529
스냅샷에서 콜드 인덱스 복원 1531
콜드 스토리지에서 웜 스토리지로의 마이그레이션 취소 1531
콜드 인덱스 메타데이터 업데이트 1532
콜드 인덱스 삭제 1532
콜드 스토리지 비활성화 1532
OpenSearch 최적화 스토리지 1533
제한 사항 1533
더 나은 수집 처리량을 위한 조정 1534
OpenSearch 최적화 인스턴스와 다른 인스턴스의 차이점 1534
OR1과 UltraWarm 스토리지의 차이 1534
OR1 인스턴스를 사용하여 도메인 프로비저닝 1535
인덱스 상태 관리 1536

ISM 정책 생성 1537
샘플 정책 1538
ISM 템플릿 1542
차이 1542
자습서: ISM 프로세스 자동화 1544
인덱스 롤업 1548
인덱스 롤업 작업 생성 1549
인덱스 변환 1550
인덱스 변환 작업 만들기 1551
클러스터 간 복제 1552
제한 사항 1553
사전 조건 1553
권한 요구 사항 1554
클러스터 간 연결 설정 1555
복제 시작 1556
복제 확인 1557
복제 일시 중지 및 다시 시작 1558
복제 중지 1559
자동 팔로우 1559
연결된 도메인 업그레이드 1560
원격 재인덱스 1561
사전 조건 1561
OpenSearch Service 인터넷 도메인 간의 데이터 재인덱스 1562
원격 도메인이 VPC에 있을 때 데이터를 재인덱싱합니다. 1563
비 OpenSearch Service 도메인 간 데이터 재인덱스 1568
대용량 데이터 집합 재인덱스 1568
원격 재인덱스 설정 1570
데이터 스트림 1570
데이터 스트림 시작하기 1571
데이터 모니터링 1574
알림 1574
알림 권한 1575
알림 시작하기 1575
알림 1576
차이 1576
이상 탐지 1578

..... 1578
자습서: 이상 탐지로 높은 CPU 사용량 탐지 1580
Amazon Q 지원 1584
지원됨 AWS 리전 1584
OpenSearch Service용 Amazon Q 설정 1585
자연어를 사용하여 시각화 생성 1585
알림 요약 및 인사이트 보기 1586
시작하기 전 준비 사항 1587
알림 요약 및 인사이트 보기 1588
검색 페이지에서 Amazon Q 생성 쿼리 결과 요약 보기 1589
권장 이상 탐지기 보기 1589
OpenSearch Service 질문을 위한 Amazon Q 채팅 액세스 1590
기계 학습 1591
용 커넥터 AWS 서비스 1591
사전 조건 1591
OpenSearch Service 커넥터 생성 1595
외부 플랫폼용 커넥터 1597
사전 조건 1597
OpenSearch Service 커넥터 생성 1601
CloudFormation 템플릿 통합 1603
사전 조건 1603
Amazon SageMaker AI 템플릿 1605
Amazon Bedrock 템플릿 1605
지원되지 않는 ML Commons 설정 1606
흐름 프레임워크 플러그인 1607
OpenSearch Service에서 ML 커넥터 생성 1607
권한 구성 1615
보안 분석 1616
보안 분석 구성 요소 및 개념 1616
로그 유형 1616
탐지기 1616
규칙 1617
조사 결과 1617
알림 1617
보안 분석 살펴보기 1617
권한 구성 1618

문제 해결 1619
해당 인덱스 오류가 없습니다. 1619
관찰성 1620
이벤트 분석으로 데이터 탐색 1620
시각화 생성 1621
Trace Analytics 자세히 살펴보기 1622
Trace Analytics 1622
사전 조건 1622
OpenTelemetry Collector 샘플 구성 1623
OpenSearch Ingestion 샘플 구성 1623
데이터 추적 탐색 1625
파이프 처리 언어 1625
..... 1625
모범 사례 1627
모니터링 및 알림 1627
CloudWatch 경보 구성 1627
로그 게시 사용 설정 1627
샤드 전략 1628
샤드 및 데이터 노드 수 결정 1628
스토리지 스큐 방지 1629
안정성 1630
OpenSearch로 최신 정보 유지 1630
스냅샷 성능 개선 1630
전용 프라이머리 노드 사용 설정 1631
여러 가용 영역에 걸쳐 배포 1631
수집 흐름 및 버퍼링 제어 1631
검색 워크로드에 대한 매핑 생성 1632
인덱스 템플릿 사용 1632
인덱스 상태 관리를 사용한 인덱스 관리 1633
사용되지 않는 인덱스 삭제 1633
고가용성을 위한 여러 도메인을 사용 1634
성능 1634
대량 요청 크기 및 압축 최적화 1634
대량 요청 응답의 크기를 줄입니다. 1635
새로 고침 주기 조정 1635
자동 조정 사용 설정 1635

보안 1635
세분화된 액세스 제어 사용 설정 1635
VPC 내에 도메인 배포 1636
제한적 액세스 정책 적용 1636
저장 시 암호화 사용 설정 1636
노드 간 암호화 사용 설정 1637
를 사용하여 모니터링 AWS Security Hub 1637
비용 최적화 1637
최신 세대 인스턴스 유형 사용 1637
최신 Amazon EBS gp3 볼륨 사용 1637
시계열 로그 데이터에 UltraWarm 및 콜드 스토리지 사용 1638
예약 인스턴스 권장 사항 검토 1638
도메인 크기 조정 1638
스토리지 요구 사항 계산 1639
샤드 수 선택 1640
인스턴스 유형 선택 및 테스트 1642
페타바이트 규모 1643
전용 조정자 노드 1645
전용 조정자 노드를 사용해야 하는 경우 1645
아키텍처 및 동작 1645
요구 사항 및 제한 사항 1646
전용 조정자 노드 프로비저닝 1646
모범 사례 1647
전용 프라이머리 노드 1648
전용 프라이머리 노드 수 선택 1649
전용 프라이머리 노드에 대한 인스턴스 유형 선택 1650
권장되는 CloudWatch 경보 1651
일반 참조 1658
지원되는 인스턴스 유형 1658
현재 세대 인스턴스 유형 1658
이전 세대 인스턴스 유형 1677
엔진 버전별 기능 1681
엔진 버전별 플러그인 1685
옵션 플러그인 1689
지원되는 연산자 1689
주요 API 차이점 1689

할당량 1742
UltraWarm 스토리지 할당량 1742
AZ당 데이터 노드 수 1650
인스턴스 패밀리별 총 노드 제한 1743
EBS 볼륨 크기 할당량 1744
네트워크 할당량 1754
샤드 크기 할당량 1761
샤드 수 할당량 1762
Java 프로세스 할당량 1762
도메인 정책 할당량 1762
예약 인스턴스 1762
예약 인스턴스 구입(콘솔) 1763
예약 인스턴스 구매(AWS CLI) 1764
예약 인스턴스(AWS SDKs) 구매 1766
비용 검사 1768
지원되는 기타 리소스 1768
자습서 1770
문서 생성 및 검색 1770
사전 조건 1770
인덱스에 문서 추가 1771
자동으로 생성되는 ID 만들기 1772
POST 명령으로 문서 업데이트 1773
대량 작업 수행 1774
문서 검색 1775
관련 리소스 1777
OpenSearch Service로 마이그레이션 1777
스냅샷 생성 및 업로드 1777
도메인 생성 1778
S3 버킷에 권한 부여 1779
스냅샷을 복원합니다. 1782
검색 애플리케이션 생성 1784
사전 조건 1784
1단계: 샘플 데이터 인덱싱 1784
2단계: Lambda 함수 생성 및 배포 1785
3단계: API Gateway에서 API 생성 1788
4단계: (선택 사항) 도메인 액세스 정책 수정 1790

Lambda 역할 매핑(세분화된 액세스 제어를 사용하는 경우) 1793
5단계: 웹 애플리케이션 테스트 1794
다음 단계 1795
지원 통화 시각화 1795
1단계: 사전 조건 구성 1796
2단계: 샘플 코드 복사 1797
(선택 사항) 3단계: 샘플 데이터 인덱싱 1801
4단계: 데이터 분석 및 시각화 1803
5단계: 리소스 정리 및 다음 단계 1804
Amazon OpenSearch Service 이름 변경 1805
새로운 API 버전 1805
인스턴스 유형의 이름 변경 1805
액세스 정책 변경 사항 1806
IAM 정책 1806
SCP 정책 1806
새로운 리소스 유형 1807
Kibana의 이름이 OpenSearch Dashboards로 변경 1808
CloudWatch 지표의 이름 변경 1808
Billing and Cost Management 콘솔 변경 사항 1809
새로운 이벤트 형식 1810
변경되지 않는 것은 무엇입니까? 1810
시작하기: 도메인을 OpenSearch 1.x로 업그레이드 1811
문제 해결 1812
OpenSearch Dashboards에 액세스할 수 없습니다. 1812
VPC 도메인에 액세스할 수 없습니다. 1812
읽기 전용 상태의 클러스터 1812
빨간색 클러스터 상태 1813
빨간색 클러스터의 자동 수정 1815
지속해서 과도한 처리 로드에서 복구 1815
노란색 클러스터 상태 1817
ClusterBlockException 1818
사용 가능한 스토리지 공간 부족 1818
높은 JVM 메모리 압력 1818
Multi-AZ with Standby로의 마이그레이션 오류 1819
대기 모드가 없는 도메인에서 대기 모드가 있는 도메인으로 마이그레이션하는 동안 인덱스, 인덱스 템플릿 또는 ISM 정책 생성 1619

잘못된 데이터 복사본 수 1819
JVM OutOfMemoryError 1819
실패한 클러스터 노드 1820
최대 샤드 제한 초과 1821
도메인이 처리 상태에 멈춤 1821
낮은 EBS 버스트 밸런스 1821
볼륨 크기 조정 중 EBS 지표 증가 1822
감사 로그를 활성화할 수 없음 1822
인덱스를 닫을 수 없음 1823
클라이언트 라이선스 확인 1823
요청 제한 1823
노드에 SSH할 수 없음 1823
"객체 스토리지 클래스의 경우 유효하지 않음" 스냅샷 오류 1823
잘못된 호스트 헤더 1824
잘못된 M3 인스턴스 유형 1824
UltraWarm 활성화 후 핫 쿼리의 작동이 중지됨 1824
업그레이드 후 다운그레이드할 수 없음 1824
모든 AWS 리전에 대한 도메인의 요약 필요 1825
OpenSearch Dashboards를 사용할 때 브라우저 오류 1825
노드 샤드 및 스토리지 스큐 1826
인덱스 샤드 및 스토리지 스큐 1827
VPC 액세스를 선택한 후 허용되지 않은 작업 1827
VPC 도메인 생성 후 로딩 단계에서 멈춤 1827
OpenSearch API에 대한 요청 거부됨 1828
Alpine Linux에서 연결할 수 없음 1829
Search Backpressure에 대한 요청이 너무 많음 1829
SDK를 사용할 때 인증서 오류 1829
버전 호환성으로 인해 사용자 지정 플러그인 설치 실패 1830
문서 기록 1833
이전 업데이트 1883
AWS 용어집 1886
..... mdccclxxxvii

Amazon OpenSearch Service란 무엇입니까?

Amazon OpenSearch Service는 AWS 클라우드에서 OpenSearch 클러스터를 쉽게 배포, 운영 및 확장할 수 있는 관리형 서비스입니다. OpenSearch Service 도메인은 OpenSearch 클러스터와 동의어입니다. 도메인은 지정된 설정, 인스턴스 유형, 인스턴스 수, 스토리지 리소스를 갖고 있는 클러스터입니다. Amazon OpenSearch Service는 OpenSearch 및 레거시 Elasticsearch OSS(소프트웨어의 최종 오픈 소스 버전인 7.10까지)를 지원합니다. 도메인을 생성할 때 어떤 검색 엔진을 사용할지 선택할 수 있습니다.

OpenSearch는 로그 분석, 실시간 애플리케이션 모니터링, 클릭 스트림 분석 같은 사용 사례를 위한 완전한 오픈 소스 검색 및 분석 엔진입니다. 자세한 내용은 OpenSearch 설명서를 참조하세요.

Amazon OpenSearch Service는 OpenSearch 클러스터에 대한 모든 리소스를 프로비저닝하고 시작합니다. 또한 실패한 OpenSearch Service 노드를 자동으로 감지한 다음 교체해 자체 관리형 인프라와 관련된 오버헤드를 줄입니다. API를 한 번만 호출하거나 콘솔에서 몇 번만 클릭하여 클러스터를 조정할 수 있습니다.

OpenSearch Service 사용을 시작하려면 OpenSearch 클러스터와 동일한 OpenSearch Service 도메인을 생성합니다. 클러스터의 각 EC2 인스턴스는 하나의 OpenSearch Service 노드 역할을 합니다.

OpenSearch Service 콘솔을 사용하여 몇 분 만에 도메인을 설정 및 구성할 수 있습니다. 프로그래밍 방식 액세스를 선호하는 경우 AWS CLI 또는 AWS SDK 또는 Terraform을 사용할 수 있습니다.

Amazon OpenSearch Service의 기능

OpenSearch Service에는 다음 기능이 포함되어 있습니다.

크기 조정

- 비용 효율적인 Graviton 인스턴스를 포함한 다양한 CPU, 메모리 및 스토리지 용량 구성(인스턴스 유형이라고 함)
- 최대 1002개의 데이터 노드 지원
- 연결된 스토리지의 최대 25PB
- 읽기 전용 데이터를 위한 비용 효율적인 UltraWarm 및 콜드 스토리지

보안

- AWS Identity and Access Management (IAM) 액세스 제어
- Amazon VPC 및 VPC 보안 그룹을 사용하는 쉬운 통합
- 저장된 데이터의 암호화 및 노드 간 암호화
- OpenSearch 대시보드에 대한 Amazon Cognito, HTTP 기본 또는 SAML 인증
- 인덱스 수준, 문서 수준 및 필드 수준 보안
- 감사 로그
- Dashboards 멀티테넌시

안정성

- 리소스를 위한 여러 지리적 위치(리전 및 가용 영역이라고 함)입니다.
- 동일한 AWS 리전의 가용 영역 두 개 또는 세 개에 노드 할당(다중 AZ)
- 클러스터 관리 작업 부담을 줄여주는 전용 프라이머리 노드
- 자동 스냅샷으로 OpenSearch Service 도메인 백업 및 복원

유연성

- 비즈니스 인텔리전스(BI) 애플리케이션과의 통합을 위한 SQL 지원
- 검색 결과 개선을 위한 사용자 지정 패키지

유명 서비스와의 통합

- OpenSearch 대시보드를 사용한 데이터 시각화
- OpenSearch Service 도메인 지표 및 설정 경보 모니터링을 위한 Amazon CloudWatch와의 통합
- OpenSearch Service 도메인에 AWS CloudTrail 대한 구성 API 호출 감사를 위해와 통합
- 스트리밍 데이터를 OpenSearch Service로 로드하기 위해 Amazon S3, Amazon Kinesis 및 Amazon DynamoDB와의 통합
- 데이터가 특정 임계값을 초과하는 경우 Amazon SNS의 알림

OpenSearch를 사용하는 경우와 Amazon OpenSearch Service를 사용하는 경우

다음 표를 사용하면 프로비저닝된 Amazon OpenSearch Service 또는 자체 관리형 OpenSearch 중 적합한 옵션을 결정하는 데 도움이 됩니다.

OpenSearch	Amazon OpenSearch Service
• 조직은 자체 프로비저닝된 클러스터를 수동으로 모니터링하고 유지 관리할 의지가 있으며 이를 위한 올바른 기술을 갖춘 인력을 보유하고 있습니다. • 코드를 컴파일 수준에서 완전히 제어하려고 합니다. • 조직에서 오픈 소스 소프트웨어를 선호하거나 고유하게 사용합니다. • 다중 클라우드 전략이 있으므로 이때 공급업체에 특정하지 않은 기술이 필요합니다. • 팀에서 중요한 프로덕션 문제를 해결할 수 있습니다. • 원하는 대로 제품을 사용, 수정 및 확장할 수 있는 유연성을 원합니다. • 새 기능이 출시되는 즉시 액세스하고 싶습니다.	• 인프라를 직접 관리, 모니터링 및 유지 관리하고 싶지 않습니다. • Amazon S3의 내구성과 저렴한 비용을 활용하여 여러 스토리지 계층에서 데이터를 계층화하여 증가하는 분석 비용을 관리하는 간단한 방법이 필요합니다. • DynamoDB, Amazon DocumentDB(MongoDB 호환), IAM, CloudWatch 및 CloudFormation과 같은 다른 AWS 서비스 와의 통합을 활용하고자 합니다. • 예방적 유지 관리를 지원 위해 및 프로덕션 문제 발생 시에서 지원에 쉽게 액세스할 수 있어야 합니다. • 자체 복구, 선제적 유지 관리, 복원력 및 백업과 같은 기능을 활용하고자 합니다.

지원되는 Elasticsearch 및 OpenSearch 버전

OpenSearch Service는 다음 버전의 OpenSearch를 지원합니다.

- 2.19, 2.17, 2.15, 2.13, 2.11, 2.9, 2.7, 2.5, 2.3, 1.3, 1.2, 1.1 및 1.0

OpenSearch Service는 다음과 같은 레거시 Elasticsearch 버전을 지원합니다.

- 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 및 1.5

가격 대비 성능, 기능 풍부성 및 보안 개선 측면에서 OpenSearch Service를 최대한 활용하려면 사용 가능한 최신 OpenSearch 버전으로 업그레이드하는 것이 좋습니다.

표준 및 확장 지원

AWS 는 표준 지원 버전에 대한 버그 수정 및 보안 업데이트를 제공합니다. 확장 지원 버전의 경우는 표준 지원이 종료된 후 최소 12개월 동안 정규화된 인스턴스 시간(NIH)당 고정 요금으로 중요한 보안 수정 사항을 AWS 제공합니다. NIH는 인스턴스 크기 및 사용 시간을 기반으로 합니다.

도메인이 더 이상 표준 지원을 받지 않는 버전을 실행하면 추가 지원 요금이 자동으로 적용됩니다. 이러한 요금을 방지하려면 지원되는 버전으로 업그레이드하세요.

다음 표에는 OpenSearch 및 레거시 Elasticsearch 버전에 대한 지원 종료 일정이 나와 있습니다.

OpenSearch Service는 여러 버전의 OpenSearch 및 레거시 오픈 소스 Elasticsearch 버전을 지원합니다. 일부 버전의 경우 표준 지원 종료 및 연장 지원 날짜를 이미 게시했습니다. 가격 대비 성능, 기능 풍부성 및 보안 개선 측면에서 OpenSearch Service를 최대한 활용하려면 사용 가능한 최신 OpenSearch 버전으로 업그레이드하는 것이 좋습니다. 다음 표에는 Elasticsearch 및 OpenSearch 버전 목록과 지원 일정이 나와 있습니다.

Elasticsearch 버전에 대한 지원 종료 일정은 다음과 같습니다.

소프트웨어 버전	표준 지원 종료	추가 지원 종료
Elasticsearch 버전 1.5 및 2.3	2025년 11월 7일	2026년 11월 7일
Elasticsearch 버전 5.1~5.5	2025년 11월 7일	2026년 11월 7일
Elasticsearch 버전 5.6	2025년 11월 7일	2028년 11월 7일

소프트웨어 버전	표준 지원 종료	추가 지원 종료
Elasticsearch 버전 6.0~6.7	2025년 11월 7일	2026년 11월 7일
Elasticsearch 버전 6.8	발표되지 않음	발표되지 않음
Elasticsearch 버전 7.1~7.8	2025년 11월 7일	2026년 11월 7일
Elasticsearch 버전 7.9	발표되지 않음	발표되지 않음
Elasticsearch 버전 7.10	발표되지 않음	발표되지 않음

OpenSearch 버전에 대한 지원 종료 일정은 다음과 같습니다.

소프트웨어 버전	표준 지원 종료	추가 지원 종료
OpenSearch 버전 1.0~1.2	2025년 11월 7일	2026년 11월 7일
OpenSearch 버전 1.3	발표되지 않음	발표되지 않음

소프트웨어 버전	표준 지원 종료	추가 지원 종료
OpenSearch 버전 2.3~2.9	2025년 11월 7일	2026년 11월 7일
OpenSearch 버전 2.11 이상	발표되지 않음	발표되지 않음

OpenSearch 및 Elasticsearch에 대한 표준 지원 및 확장 지원

AWS 는 표준 지원에서 다루는 버전에 대한 정기적인 버그 수정 및 보안 업데이트를 제공합니다. 추가 지원에 따른 버전의 경우는 표준 지원 종료 후 최소 12개월 동안 정규화된 인스턴스 시간(NIH)마다 추가 고정 요금을 지불하고 중요한 보안 수정 사항을 AWS 제공합니다. NIH는 인스턴스 크기(예: 중형, 대형) 및 인스턴스 시간 수의 인자로 계산됩니다(예는 아래 추가 지원 요금 계산 섹션 참조). 추가 지원 요금은 도메인이 표준 지원이 종료된 버전을 실행할 때 자동으로 적용됩니다. 추가 지원 요금이 발생하지 않도록 표준 지원이 적용되는 최신 버전으로 업그레이드할 수 있습니다. 추가 지원 요금에 대한 자세한 내용은 요금 페이지를 참조하세요. 추가 지원에 대한 일반적인 내용은 추가 지원 FAQ를 참조하세요.

추가 지원 요금 계산

추가 지원 중인 버전을 실행하는 도메인에는 미국 동부(버지니아 북부) 리전에서 0.0065 USD와 같은 고정 추가 요금/정규화된 인스턴스 시간(NIH)이 부과됩니다. NIH는 인스턴스 크기(예: 중간, 큰) 및 인스턴스 시간 수의 인자로 계산됩니다. 예를 들어 미국 동부(버지니아 북부) 리전에서 24시간 동안 m7g.medium.search 인스턴스를 실행하는 경우 인스턴스 시간당(온디맨드) 가격이 0.068 USD인 경우 일반적으로 1.632 USD(0.068x24 USD)를 지불합니다. 확장 지원 버전을 실행하는 경우 추가 0.0065 USD/NIH를 지불합니다.이 금액은 0.0065 USD x 24(인스턴스 시간 수) x 2(크기 정규화 인수, 중간 규모 인스턴스의 경우 2)로 계산되며, 24시간 동안 추가 지원을 받으려면 0.312 USD입니다. 24시간 동안 지불할 총 금액은 표준 인스턴스 사용 비용과 추가 지원 비용, 즉 1.944 USD(1.632 USD+0.312 USD)의 합계입니다. 아래 표에는 OpenSearch Service의 다양한 인스턴스 크기에 대한 정규화 인자가 나와 있습니다.

인스턴스 크기	정규화 인자	
nano	0.25	
micro	0.5	
small	1	
medium	2	
large	4	
xlarge	8	
2xlarge	16	
4xlarge	32	
8xlarge	64	
9xlarge	72	
10xlarge	80	
12xlarge	96	
16xlarge	128	
18xlarge	144	
24xlarge	192	
32xlarge	256	

Amazon OpenSearch Service 요금

OpenSearch Service에서는 EC2 인스턴스의 사용 시간과 인스턴스에 연결된 EBS 스토리지 볼륨의 누적 크기에 대해 요금을 지불합니다. 표준 AWS 데이터 전송 요금도 적용됩니다.

하지만 알아둘 만한 데이터 전송 예외가 몇 가지 존재합니다. 도메인에서 다중 가용 영역을 사용하는 경우, OpenSearch Service는 도메인이 속하는 두 가용 영역 간의 트래픽에 대해 요금을 청구하지 않습니다. 또한 샤드 할당 및 재분배 과정에서는 도메인 내부에서 상당한 데이터 전송이 발생합니다. OpenSearch Service는 이 트래픽에 대해서도 측정하거나 요금을 청구하지 않습니다. 마찬가지로, OpenSearch Service는 UltraWarm/콜드 노드와 Amazon S3 간의 데이터 전송에 대해 요금을 청구하지 않습니다

자세한 내용은 Amazon OpenSearch Service 요금을 참조하세요. 구성 변경 도중 발생하는 변경 사항에 대한 자세한 내용은 the section called “구성 변경 비용” 섹션을 참조하세요.

관련 서비스

OpenSearch Service는 일반적으로 다음 서비스와 함께 사용됩니다.

Amazon CloudWatch

OpenSearch Service 도메인은 CloudWatch에 지표를 자동으로 보내므로 도메인 상태 및 성능을 모니터링할 수 있습니다. 자세한 내용은 Amazon CloudWatch로 OpenSearch 클러스터 지표 모니터링 섹션을 참조하세요.

CloudWatch Logs가 다른 방향으로 진행될 수도 있습니다. 분석을 위해 OpenSearch Service로 데이터를 스트리밍하도록 CloudWatch Logs를 구성할 수 있습니다. 자세한 내용은 the section called “Amazon CloudWatch에서 스트리밍 데이터 로드” 섹션을 참조하세요.

AWS CloudTrail

AWS CloudTrail 를 사용하여 계정의 OpenSearch Service 구성 API 호출 및 관련 이벤트 기록을 가져옵니다. 자세한 내용은 AWS CloudTrail을 사용한 Amazon OpenSearch Service API 호출 모니터링 단원을 참조하십시오.

Amazon Kinesis

Kinesis는 방대한 규모의 스트리밍 데이터를 실시간으로 처리하는 관리형 서비스입니다. 자세한 내용은 the section called “Amazon Kinesis Data Streams에서 스트리밍 데이터 로드” 및 the section called “Amazon Kinesis Data Firehose에서 스트리밍 데이터 로드” 섹션을 참조하세요.

Amazon S3

Amazon Simple Storage Service(Amazon S3)는 인터넷 스토리지 를 제공합니다. 이 가이드에서는 Amazon S3와의 통합을 위한 Lambda 샘플 코드를 제공합니다. 자세한 내용은 the section called “Amazon S3에서 스트리밍 데이터 로드” 섹션을 참조하세요.

AWS IAM

AWS Identity and Access Management (IAM)는 OpenSearch Service 도메인에 대한 액세스를 관리하는 데 사용할 수 있는 웹 서비스입니다. 자세한 내용은 the section called “ID 및 액세스 관리” 단원을 참조하십시오.

AWS Lambda

AWS Lambda 는 서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있는 컴퓨팅 서비스입니다. 이 가이드는 DynamoDB, Amazon S3 및 Kinesis의 데이터를 스트리밍하기 위한 Lambda 샘플 코드를 제공합니다. 자세한 내용은 the section called “OpenSearch Service로 스트리밍 데이터 로드” 섹션을 참조하세요.

Amazon DynamoDB

Amazon DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다. OpenSearch Service에 데이터를 스트리밍하는 데 대한 자세한 내용은 the section called “Amazon DynamoDB에서 스트리밍 데이터 로드” 섹션을 참조하세요.

Amazon QuickSight

QuickSight 대시보드를 사용하여 OpenSearch Service의 데이터를 시각화할 수 있습니다. 자세한 내용은 QuickSight 사용 설명서의 QuickSight에서 Amazon OpenSearch Service 사용을 참조하세요. QuickSight

> **Note**
>
> OpenSearch에는 Elasticsearch B.V.의 특정 Apache 라이선스 Elasticsearch 코드 및 기타 소스 코드가 포함되어 있습니다. Elasticsearch B.V.는 이러한 기타 소스 코드의 소스가 아닙니다. ELASTICSEARCH는 Elasticsearch B.V.의 등록 상표입니다.

Amazon OpenSearch Service 설정

권한 부여

프로덕션 환경에서는 더 세밀한 정책을 사용하는 것이 좋습니다. 액세스 관리에 대한 자세한 내용은 IAM 사용 설명서의 AWS 리소스에 대한 액세스 관리를 참조하세요.

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.
- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.
- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

프로그래밍 방식 액세스 권한 부여

사용자는 AWS 외부에서와 상호 작용하려는 경우 프로그래밍 방식의 액세스가 필요합니다 AWS Management Console. 프로그래밍 방식 액세스를 부여하는 방법은에 액세스하는 사용자 유형에 따라 다릅니다 AWS.

사용자에게 프로그래밍 방식 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자는 누구인가요?	To	액세스 권한을 부여하는 사용자
작업 인력 ID	임시 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다.

프로그래밍 방식 액세스가 필요한 사용자는 누구인가요?	To	액세스 권한을 부여하는 사용자
(IAM Identity Center가 관리하는 사용자)		• 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 를 사용하도록 구성을 AWS IAM Identity Center 참조하세요. • AWS SDKs, 도구 및 AWS APIs의 경우 SDK 및 도구 참조 안내서의 IAM Identity Center 인증을 참조하세요. AWS SDKs
IAM	임시 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	IAM 사용 설명서의 AWS 리소스에서 임시 자격 증명 사용의 지침을 따릅니다.
IAM	(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 IAM 사용자 자격 증명을 사용하여 인증을 참조하세요. • AWS SDKs 및 도구의 경우 SDK 및 도구 참조 안내서의 장기 자격 증명을 사용하여 인증을 참조하세요. AWS SDKs • AWS APIs 경우 IAM 사용 설명서의 IAM 사용자의 액세스 키 관리를 참조하세요.

설치 및 구성 AWS CLI

OpenSearch Service API를 사용하려면 최신 버전의 AWS Command Line Interface (AWS CLI)를 설치해야 합니다. 콘솔에서 OpenSearch Service를 사용하는 AWS CLI 데가 필요하지 않으며의 단계에 따라 CLI 없이 시작할 수 있습니다Amazon OpenSearch Service 시작하기.

를 설정하려면 AWS CLI

1. macOS, Linux 또는 Windows AWS CLI 용의 최신 버전을 설치하려면 의 최신 버전 설치 또는 업데이트를 참조하세요 AWS CLI.
2. OpenSearch Service를 AWS 서비스포함하여에 대한 액세스 AWS CLI 및 보안 설정을 구성하려면 를 사용한 빠른 구성을 `aws configure`참조하세요.
3. 명령 프롬프트에 다음 명령을 입력하여 설정을 확인합니다.

   ```
   aws opensearch help
   ```

   AWS CLI 명령은 파라미터 또는 프로파일로 설정하지 않는 한 구성 AWS 리전 의 기본값을 사용합니다. 파라미터를 AWS 리전 사용하여를 설정하려면 각 명령에 `--region` 파라미터를 추가할 수 있습니다.

   프로파일 AWS 리전 로를 설정하려면 먼저 `~/.aws/config` 파일 또는 `%UserProfile%/.aws/config` 파일(Microsoft Windows용)에 명명된 프로파일을 추가합니다. AWS CLI에 이름이 지정된 프로필의 단계를 따르세요. 그런 다음 다음 예제의 명령과 유사한 명령을 사용하여 AWS 리전 및 기타 설정을 설정합니다.

   ```
   [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
   ```

콘솔을 엽니다.

이 섹션의 콘솔 관련 주제의 대부분은 콘솔에서 시작됩니다. 아직에 로그인하지 않은 경우 AWS 계정 로그인한 다음 OpenSearch Service 콘솔을 열고 다음 섹션으로 계속 진행하여 OpenSearch Service를 계속 시작합니다.

Amazon OpenSearch Service 시작하기

아직 계정이 없는 경우 AWS 계정에 가입하여 시작합니다. 계정을 설정한 후 Amazon OpenSearch Service 시작하기 자습서를 완료합니다. 이 서비스에 대해 알아보는 중 추가 정보가 필요한 경우 다음 소개 주제를 참조하세요.

- 도메인 생성.
- 워크로드에 맞게 도메인 크기를 조정합니다.
- 도메인 액세스 정책 또는 세분화된 액세스 제어를 사용하여 도메인에 대한 액세스를 제어합니다.
- 수동으로 또는 다른 AWS 서비스에서 데이터를 인덱싱합니다.
- OpenSearch 대시보드를 사용하여 데이터를 검색하고 시각화를 생성합니다.
- 도메인 생성을 위한 고급 옵션에 대해 알아봅니다. 자세한 내용은 도메인 생성 및 관리 섹션을 참조하세요.
- 도메인에서 인덱스를 관리하는 방법을 알아보세요. 자세한 내용은 인덱스 관리 섹션을 참조하세요.
- Amazon OpenSearch Service 작업에 대한 자습서 중 하나를 시도해보세요. 자세한 내용은 자습서 단원을 참조하십시오.

자체 관리형 OpenSearch 클러스터에서 OpenSearch Service로의 마이그레이션에 대한 자세한 내용은 the section called “OpenSearch Service로 마이그레이션” 섹션을 참조하세요.

자세한 내용은 이 설명서의 도메인 생성 및 관리 및 기타 주제 섹션을 참조하세요. 자체 관리형 OpenSearch 클러스터에서 OpenSearch Service로의 마이그레이션에 대한 자세한 내용은 the section called “OpenSearch Service로 마이그레이션” 섹션을 참조하세요.

OpenSearch Service 콘솔, AWS CLI또는 SDK를 사용하여 다음 단계를 완료할 수 있습니다 AWS . 설치 및 설정에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 AWS CLI참조하세요.

Amazon OpenSearch Service 서비스 도메인 생성

> **Important**
>
> 이것은 테스트 Amazon OpenSearch Service 도메인을 구성하기 위한 약식 자습서입니다. 프로덕션 도메인을 생성하기 위해 이 프로세스를 사용하지 않습니다. 동일한 프로세스의 전체 버전은 도메인 생성 및 관리 섹션을 참조하세요.

OpenSearch Service 도메인은 OpenSearch 클러스터와 동의어입니다. 도메인은 지정된 설정, 인스턴스 유형, 인스턴스 수, 스토리지 리소스를 갖고 있는 클러스터입니다. 콘솔, AWS CLI또는 AWS SDK를 사용하여 OpenSearch Service 도메인을 생성할 수 있습니다. SDKs

콘솔을 사용하여 OpenSearch Service 도메인을 만들려면

1. https://aws.amazon.com으로 이동하여 콘솔에 로그인(Sign In to the Console)을 선택합니다.
2. Analytics(분석)에서 Amazon OpenSearch Service를 선택합니다.
3. [도메인 생성(Create domain)]을 선택합니다.
4. 도메인의 이름을 입력합니다. 이 자습서의 예제에서는 movies라는 이름을 사용합니다.
5. 도메인 생성 방법으로 [표준 생성]을 선택합니다.

   > **Note**
   >
   > 모범 사례에 따라 프로덕션 도메인을 빠르게 구성하려면 간편 생성을 선택할 수 있습니다. 이 튜토리얼에서는 개발 및 테스트 목적으로 표준 생성을 사용하겠습니다.

6. 템플릿의 경우 개발/테스트를 선택합니다.
7. 배포 옵션으로는 대기 모드가 있는 도메인을 선택합니다.
8. 버전(Version)에서 최신 버전을 선택합니다.
9. 지금은 데이터 노드, 웜 및 콜드 데이터 스토리지, 전용 마스터 노드, 스냅샷 구성, 사용자 지정 엔드포인트 섹션을 무시하세요.
10. 이 자습서에서는 간단한 설명을 위해 퍼블릭 액세스 도메인을 사용합니다. [네트워크(Network)]에서 [퍼블릭 액세스(Public access)]를 선택합니다.
11. 세분화된 액세스 제어 설정에서 세분화된 액세스 제어 활성화 확인란을 선택한 상태로 유지합니다. 마스터 사용자 생성을 선택하고 사용자 이름과 암호를 입력합니다.
12. 지금은SAML 인증 및 Amazon Cognito 인증 섹션을 무시합니다.
13. [액세스 정책(Access policy)]에서 [세분화된 액세스 제어만 사용(Only use fine-grained access control)]을 선택합니다. 이 자습서에서는 세분화된 액세스 제어를 통해 도메인 액세스 정책이 아닌 인증을 처리합니다.
14. 나머지 설정은 무시하고 [생성(Create)]을 선택합니다. 새 도메인은 일반적으로 초기화하는 데 15~30분 정도 걸리지만 구성에 따라 시간이 더 오래 걸릴 수 있습니다. 도메인을 초기화한 후 도메인을 선택하여 구성 창을 엽니다. 다음 단계에서 사용할 일반 정보(General information)에서의 도메인 엔드포인트(예: `https://search-my-domain.us-east-1.es.amazonaws.com`)를 기록합니다.

다음: 인덱싱을 위해 OpenSearch Service 도메인에 데이터 업로드

인덱싱을 위해 Amazon OpenSearch Service 도메인에 데이터 업로드

> **Important**
>
> 이것은 Amazon OpenSearch Service에 소량의 테스트 데이터를 업로드하기 위한 약식 자습서입니다. 프로덕션 도메인에서 데이터를 업로드하는 방법에 대한 자세한 내용은 데이터 인덱싱 섹션을 참조하세요.

명령줄이나 대부분의 프로그래밍 언어를 사용하여 OpenSearch Service 도메인에 데이터를 업로드할 수 있습니다.

다음 예제의 요청에서는 편의상 간단히 일반적인 HTTP 클라이언트인 curl을 사용합니다. 액세스 정책에서 IAM 사용자 또는 역할을 지정한 경우 curl 같은 클라이언트에서는 필요한 요청 서명을 실행할 수 없습니다. 이 프로세스를 성공적으로 수행하려면 1단계에서 구성한 것처럼 기본 사용자 이름 및 암호로 세분화된 액세스 제어를 사용해야 합니다.

Windows에 curl을 설치하고 명령 프롬프트에서 이를 사용할 수 있지만, Cygwin 같은 도구나 Linux용 Windows 하위 시스템을 권장합니다. macOS 및 대부분의 Linux 배포판은 curl이 사전 설치된 상태로 제공됩니다.

옵션 1: 단일 문서 업로드

다음 명령을 실행하여 movies 도메인에 문서 하나를 추가합니다.

```
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'
```

명령에서 1단계에서 생성한 사용자 이름과 암호를 입력합니다.

이 명령에 대한 자세한 설명과 OpenSearch Service에 대한 서명된 요청을 작성하는 방법은 데이터 인덱싱 섹션을 참조하세요.

옵션 2: 여러 문서 업로드

문서 여러 개가 포함된 JSON 파일을 OpenSearch Service 도메인에 업로드하려면

1. bulk_movies.json이라는 로컬 파일을 생성합니다. 다음 내용을 파일에 복사하여 붙여넣고, 후행 줄바꿈을 추가합니다.

```
{ "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. 파일이 저장되는 로컬 디렉터리에서 다음 명령을 실행하여 movies 도메인에 파일을 업로드합니다.

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

벌크 파일 형식에 대한 자세한 내용은 데이터 인덱싱 섹션을 참조하세요.

다음: 문서 검색

Amazon OpenSearch Service에서 문서 검색

Amazon OpenSearch Service 도메인에서 문서를 검색하려면 OpenSearch 검색 API를 사용합니다. 그 밖에 OpenSearch Dashboards를 사용하여 도메인의 문서를 검색할 수도 있습니다.

명령줄에서 문서 검색

다음 명령을 실행하여 movies 도메인에서 mars를 검색합니다.

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

이전 페이지에서 대량 데이터를 사용한 경우, 대신에 rebel을 검색해 보세요.

다음과 유사한 응답이 나타납니다.

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

OpenSearch Dashboards를 사용하여 문서 검색

OpenSearch Dashboards는 OpenSearch와 함께 작동하도록 제작된 인기 있는 오픈 소스 시각화 도구입니다. 인덱스를 검색하고 모니터링할 수 있는 유용한 사용자 인터페이스를 제공합니다.

Dashboards를 사용하여 OpenSearch Service 도메인에서 문서를 검색하려면

1. 도메인에 대한 OpenSearch Dashboards URL으로 이동합니다. OpenSearch Service 콘솔의 도메인 대시보드에서 URL을 찾을 수 있습니다. URL은 다음 형식을 따릅니다.

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

2. 기본 사용자 이름 및 암호를 사용하여 로그인합니다.
3. Dashboards를 사용하려면 인덱스 패턴을 1개 이상 생성해야 합니다. Dashboards는 이러한 패턴을 사용하여 분석할 인덱스를 식별하기 때문입니다. 왼쪽 탐색 패널을 열고 스택 관리(Stack Management)를 선택하고 인덱스 패턴(Index Patterns)을 선택한 다음 인덱스 패턴 생성(Create index pattern)을 선택합니다. 본 자습서에서는 movies를 입력합니다.
4. 다음 단계(Next step)를 선택한 후 인덱스 패턴 생성(Create index pattern)을 선택합니다. 패턴이 생성되면 `actor`, `director`와 같은 다양한 문서 필드를 볼 수 있습니다.
5. 인덱스 패턴(Index Patterns) 페이지로 돌아가 `movies`가 기본값으로 설정되어 있는지 확인합니다. 그렇지 않은 경우 패턴을 선택하고 별 아이콘을 선택하여 기본값으로 설정합니다.
6. 데이터 검색을 시작하려면 왼쪽 탐색 패널을 다시 열고 발견(Discover)을 선택합니다.
7. 단일 문서를 업로드한 경우 검색 창에 mars를 입력하고 여러 문서를 업로드한 경우 rebel을 입력한 다음, Enter를 누릅니다. 배우나 감독 이름과 같은 다른 단어를 검색해 볼 수 있습니다.

다음: 도메인 삭제

Amazon OpenSearch Service 도메인 삭제

자습서의 movies 도메인은 테스트용이므로, 시험 사용을 완료하면 비용 발생을 방지하기 위해 도메인을 삭제해야 합니다.

콘솔에서 OpenSearch Service 도메인을 삭제하려면

1. Amazon OpenSearch Service 콘솔에 로그인합니다.
2. [도메인(Domains)]에서 movies 도메인을 선택합니다.
3. [삭제(Delete)]를 선택하고 삭제 의사를 확인합니다.

Amazon OpenSearch Ingestion 개요

Amazon OpenSearch Ingestion은 실시간 로그, 지표 및 추적 데이터를 Amazon OpenSearch Service 도메인 및 OpenSearch Serverless 컬렉션으로 스트리밍하는 완전 관리형 서버리스 데이터 수집기입니다. OpenSearch

OpenSearch Ingestion을 사용하면 데이터를 수집하는 데 Logstash 또는 Jaeger와 같은 타사 도구가 더 이상 필요하지 않습니다. OpenSearch Ingestion으로 데이터를 전송하도록 데이터 생산자를 구성하면 지정된 도메인 또는 컬렉션으로 데이터가 자동으로 전송됩니다. 전송 전에 데이터를 변환할 수도 있습니다.

OpenSearch Ingestion은 서버리스이므로 인프라를 관리하거나 소프트웨어를 패치하거나 클러스터를 수동으로 확장할 필요가 없습니다. 에서 직접 수집 파이프라인을 프로비저닝할 수 있으며 AWS Management Console OpenSearch Ingestion이 나머지를 처리합니다.

Amazon OpenSearch Service의 구성 요소인 OpenSearch Ingestion은 다운스트림 분석 및 시각화를 위해 데이터를 필터링, 보강, 변환, 정규화 및 집계하는 오픈 소스 데이터 수집기인 Data Prepper로 구동됩니다.

Amazon OpenSearch Ingestion의 주요 개념

OpenSearch Ingestion 사용을 시작하기 전에 이러한 주요 개념을 이해하는 것이 좋습니다.

파이프라인

OpenSearch Ingestion 관점에서 볼 때 파이프라인은 OpenSearch Service 내에서 사용자가 생성하는 프로비저닝된 단일 데이터 수집기를 말합니다. 이는 하나 이상의 하위 파이프라인이 포함된 전체 YAML 구성 파일이라고 생각하면 됩니다. 수집 파이프라인을 만드는 단계는 the section called “파이프라인 생성”을 참조하세요.

하위 파이프라인

YAML 구성 파일 내에서 하위 파이프라인을 정의합니다. 각 하위 파이프라인은 소스, 버퍼, 0개 이상의 프로세서, 1개 이상의 싱크의 조합입니다. YAML 파일 하나에 각각 고유한 소스, 프로세서, 싱크가 있는 여러 개의 하위 파이프라인을 정의할 수 있습니다. CloudWatch 및 기타 서비스를 통한 모니터링을 지원하려면 모든 하위 파이프라인과 구별되는 파이프라인 이름을 지정하는 것이 좋습니다.

단일 YAML 파일 내에 여러 하위 파이프라인을 함께 문자화하여 한 하위 파이프라인의 소스가 다른 하위 파이프라인이고 해당 싱크가 세 번째 하위 파이프라인이 되도록 할 수 있습니다. 예제는 the section called “OpenTelemetry Collector” 섹션을 참조하세요.

소스

하위 파이프라인의 입력 구성 요소입니다. 파이프라인이 레코드를 소비하는 메커니즘을 정의합니다. 소스는 HTTPS를 통해 이벤트를 수신하거나 Amazon S3와 같은 외부 엔드포인트에서 읽어 이벤트를 소비할 수 있습니다. 소스에는 푸시 기반과 풀 기반의 두 가지 유형이 있습니다. HTTP 및 OTel 로그와 같은 푸시 기반 소스는 레코드를 수집 엔드포인트로 스트리밍합니다. OTel 추적 및 S3와 같은 풀 기반 소스는 소스에서 데이터를 가져옵니다.

Processors

레코드를 싱크에 게시하기 전에 원하는 형식으로 필터링, 변환 및 보강할 수 있는 중간 처리 장치입니다. 프로세서는 파이프라인의 선택적 구성 요소입니다. 프로세서를 정의하지 않으면 소스에 정의된 형식으로 레코드가 게시됩니다. 프로세서가 하나 이상 있을 수 있습니다. 파이프라인은 사용자가 정의한 순서대로 프로세서를 실행합니다.

Sink

하위 파이프라인의 출력 구성 요소입니다. 하위 파이프라인이 레코드를 게시하는 하나 이상의 대상을 정의합니다. OpenSearch Ingestion은 OpenSearch Service 도메인을 싱크로 지원합니다. 또한 하위 파이프라인을 싱크로 지원합니다. 즉, 단일 OpenSearch Ingestion 파이프라인(YAML 파일) 내에서 여러 하위 파이프라인을 하나로 묶을 수 있습니다. 자체 관리형 OpenSearch 클러스터는 싱크로 지원되지 않습니다.

Buffer

소스와 싱크 사이의 계층 역할을 하는 프로세서의 일부입니다. 파이프라인 내에서 수동으로 버퍼를 구성할 수 없습니다. OpenSearch Ingestion은 기본 버퍼 구성을 사용합니다.

경로

파이프라인 작성자가 특정 조건에 맞는 이벤트만 다른 싱크로 전송할 수 있도록 하는 프로세서의 일부입니다.

유효한 하위 파이프라인 정의에는 소스와 싱크가 포함되어야 합니다. 각 파이프라인 요소에 대한 자세한 내용은 구성 참조를 참조하세요.

Amazon OpenSearch Ingestion의 이점

OpenSearch Ingestion에는 다음과 같은 주요 이점이 있습니다.

- 자체 프로비저닝된 파이프라인을 수동으로 관리할 필요가 없습니다.
- 정의한 용량 한도에 따라 파이프라인을 자동으로 확장합니다.
- 보안 및 버그 패치를 통해 파이프라인을 최신 상태로 유지합니다.
- 추가 보안 계층을 위해 파이프라인을 Virtual Private Cloud(VPC)에 연결하는 옵션을 제공합니다.
- 파이프라인을 중지하고 시작하여 비용을 제어할 수 있습니다.
- 자주 사용되는 사용 사례에 대한 파이프라인 구성 청사진을 제공하여 더 빠르게 시작하고 실행할 수 있도록 지원합니다.
- 다양한 AWS SDKs 및 OpenSearch Ingestion API를 통해 프로그래밍 방식으로 파이프라인과 상호 작용할 수 있습니다.
- Amazon CloudWatch의 성능 모니터링과 CloudWatch Logs의 오류 로깅을 지원합니다.

Amazon OpenSearch Ingestion의 제한 사항

OpenSearch Ingestion에는 다음과 같은 제한 사항이 있습니다.

- OpenSearch 1.0 이상 또는 Elasticsearch 6.8 이상을 실행하는 도메인으로만 데이터를 수집할 수 있습니다. OTel 추적 소스를 사용하는 경우 OpenSearch 대시보드 플러그인 을 사용할 수 있도록 Elasticsearch 7.9 이상을 사용하는 것이 좋습니다.
- 파이프라인이 VPC 내에 있는 OpenSearch Service 도메인에 쓰는 경우 파이프라인은 도메인 AWS 리전 과 동일한에서 생성되어야 합니다.
- 파이프라인 정의 내에서 단일 데이터 소스만 구성할 수 있습니다.
- 자체 관리형 OpenSearch 클러스터를 싱크로 지정할 수 없습니다.
- 사용자 지정 엔드포인트를 싱크로 지정할 수 없습니다. 사용자 지정 엔드포인트가 활성화된 도메인에도 계속해서 쓸 수 있지만 표준 엔드포인트를 지정해야 합니다.
- 옵트인 리전 내의 리소스를 소스 또는 싱크로 지정할 수 없습니다.
- 파이프라인 구성에 포함할 수 있는 파라미터에는 몇 가지 제약이 있습니다. 자세한 내용은 the section called “구성 요구 사항 및 제약 조건” 단원을 참조하십시오.

지원되는 Data Prepper 버전

OpenSearch Ingestion은 현재 다음과 같은 메이저 버전의 Data Prepper를 지원합니다.

- 2.x

코드 편집기를 사용하여 파이프라인을 생성할 때 필수 `version` 옵션을 사용하여 사용할 Data Prepper의 메이저 버전을 지정합니다. 예를 들어 `version: "2"`입니다. OpenSearch Ingestion은 해당 메이저 버전의 지원되는 최신 마이너 버전을 검색하고 해당 버전으로 파이프라인을 프로비저닝합니다.

코드 편집기를 사용하여 파이프라인을 생성하지 않으면 OpenSearch Ingestion은 지원되는 최신 버전으로 파이프라인을 자동으로 프로비저닝합니다.

현재 OpenSearch Ingestion은 Data Prepper 버전 2.7로 파이프라인을 프로비저닝합니다. 자세한 내용은 2.7 release notes를 참조하세요. OpenSearch Ingestion이 특정 메이저 버전의 모든 마이너 버전을 지원하는 것은 아닙니다.

파이프라인 구성을 업데이트할 때 Data Prepper의 새 마이너 버전을 지원하는 경우 OpenSearch Ingestion은 파이프라인 구성에 지정된 메이저 버전의 지원되는 최신 마이너 버전으로 파이프라인을 자동으로 업그레이드합니다. 예를 들어, 파이프라인 구성에 `version: "2"`이 있고 OpenSearch Ingestion이 처음에 파이프라인을 버전 2.6.0으로 프로비저닝했을 수 있습니다. 버전 2.7.0에 대한 지원이 추가되고 파이프라인 구성을 변경하면 OpenSearch Ingestion은 파이프라인을 버전 2.7.0으로 업그레이드합니다. 이 프로세스를 통해 최신 버그를 수정하고 성능을 개선하여 파이프라인을 최신 상태로 유지할 수 있습니다. OpenSearch Ingestion은 파이프라인 구성 내에서 `version` 옵션을 수동으로 변경하지 않는 한 파이프라인의 메이저 버전을 업데이트할 수 없습니다. 자세한 내용은 the section called “파이프라인 업데이트” 단원을 참조하십시오.

Amazon OpenSearch Ingestion에서 파이프라인 조정

OpenSearch Ingestion은 지정된 최소 및 최대 Ingestion OpenSearch 컴퓨팅 단위(Ingestion OCUs를 기반으로 파이프라인 용량을 자동으로 조정합니다. 따라서 수동 프로비저닝 및 관리가 필요하지 않습니다.

각 Ingestion OCU는 약 15GiB의 메모리와 2개의 vCPUs. 파이프라인의 최소 및 최대 OCU 값을 지정할 수 있으며, OpenSearch Ingestion은 이러한 제한에 따라 파이프라인 용량을 자동으로 조정합니다.

파이프라인을 생성할 때 다음 값을 지정합니다.

- 최소 용량 - 파이프라인은 이 Ingestion OCU 수만큼 용량을 줄일 수 있습니다. 지정된 최소 용량은 파이프라인의 시작 용량이기도 합니다.
- 최대 용량 - 파이프라인은 이 Ingestion OCU 수만큼 용량을 늘릴 수 있습니다.

파이프라인의 최대 용량이 워크로드의 급증을 처리할 수 있을 만큼 충분히 높고, 최소 용량이 파이프라인의 사용량이 많지 않을 때 비용을 최소화할 수 있을 만큼 낮추세요. 설정에 따라 OpenSearch Ingestion은 파이프라인이 수집 워크로드를 처리하기 위해 Ingestion OCU 수를 자동으로 조정합니다. 특정 시간에 파이프라인에서 활발하게 사용 중인 Ingestion OCU에 대해서만 비용이 청구됩니다.

OpenSearch Ingestion 파이프라인에 할당된 용량은 파이프라인의 처리 요구 사항 및 클라이언트 애플리케이션에서 생성된 로드에 따라 확장 및 축소됩니다. 용량이 제한된 경우 OpenSearch Ingestion은 더 많은 컴퓨팅 유닛(GiB 메모리)을 할당하여 규모를 확장합니다. 파이프라인이 더 작은 워크로드를 처리하거나 데이터를 전혀 처리하지 않는 경우에는 구성된 최소 Ingestion OCU로 스케일 다운할 수있습니다.

상태 비저장 파이프라인에는 최소 1개의 Ingestion OCU와 최대 96개의 Ingestion OCU를, 상태 저장 파이프라인에는 최대 48개의 Ingestion OCU를 지정할 수 있습니다. 푸시 기반 소스의 경우 최소 2개 이상의 Ingestion OCU를 사용하는 것이 좋습니다. 영구 버퍼링이 활성화된 경우 최소 2개에서 최대 384개의 Ingestion OCU를 지정할 수 있습니다.

단일 소스, 간단한 Grok 패턴, 싱크가 있는 표준 로그 파이프라인의 경우 각 컴퓨팅 유닛은 초당 최대 2MiB를 지원할 수 있습니다. 프로세서가 여러 개 있는 더 복잡한 로그 파이프라인의 경우 각 컴퓨팅 유닛이 더 적은 수집 부하를 지원할 수 있습니다. 파이프라인 용량과 리소스 사용률을 기반으로 OpenSearch Ingestion 규모 조정 프로세스가 시작됩니다.

고가용성을 보장하기 위해 Ingestion OCU는 가용 영역(AZ)에 분산됩니다. AZ 수는 지정한 최소 용량에 따라 달라집니다.

예를 들어 최소 2개의 컴퓨팅 유닛을 지정하는 경우 특정 시점에 사용 중인 Ingestion OCU가 2개의 AZ에 균등하게 분배됩니다. 최소 3개 이상의 컴퓨팅 유닛을 지정하는 경우 Ingestion OCU는 3개의 AZ에 균등하게 분배됩니다. 수집 파이프라인의 99.9% 가용성을 보장하려면 최소 2개의 이상의 Ingestion OCU를 프로비저닝하는 것이 좋습니다.

파이프라인이 `Create failed`, `Creating`, `Deleting`, 및 `Stopped` 상태일 때는 Ingestion OCU에 대한 요금이 청구되지 않습니다.

파이프라인의 용량 설정을 구성하고 검색하는 방법에 대한 지침은 the section called “파이프라인 생성”을 참조하세요.

OpenSearch Ingestion 가격

특정 시기에는 파이프라인에 할당된 Ingestion OCU 수에 대해서만 비용을 지불합니다. 단, 파이프라인을 통한 데이터 흐름 여부와 상관없습니다. OpenSearch Ingestion은 사용량에 따라 파이프라인 용량을 늘리거나 줄여 워크로드를 즉시 수용합니다.

자세한 내용은 Amazon OpenSearch Service 요금을 참조하세요.

지원됨 AWS 리전

OpenSearch Ingestion은 OpenSearch Service를 사용할 수 AWS 리전 있는의 하위 집합에서 사용할 수 있습니다. 지원되는 리전 목록은 AWS 일반 참조의 Amazon OpenSearch Service 엔드포인트 및 할당량을 참조하세요.

Amazon OpenSearch Ingestion 내 역할 및 사용자 설정

Amazon OpenSearch Ingestion은 소스 애플리케이션이 파이프라인에 쓸 수 있도록 허용하고 파이프라인이 싱크에 쓸 수 있도록 허용하기 위해 다양한 권한 모델과 IAM 역할을 사용합니다. 데이터 수집을 시작하려면 먼저 사용 사례에 따라 특정 권한을 가진 하나 이상의 IAM 역할을 생성해야 합니다.

성공적인 파이프라인을 설정하려면 최소한 다음과 같은 역할이 필요합니다.

명칭	설명
파이프라인 역할	파이프라인 역할은 파이프라인이 소스에서 읽고 도메인 또는 컬렉션 싱크에 쓰는 데 필요한 권한을 제공합니다. 파이프라인 역할을 수동으로 생성하거나 OpenSearch Ingestion에서 자동으로 생성하도록 할 수 있습니다.
수집 역할	수집 역할에는 파이프라인 리소스에 대한 `osis:Ingest` 권한이 포함됩니다. 이 권한을 사용하면 푸시 기반 소스가 데이터를 파이프라인으로 수집할 수 있습니다.

다음 이미지는 Amazon S3 또는 Fluent Bit와 같은 데이터 소스가 다른 계정의 파이프라인에 쓰는 일반적인 파이프라인 설정을 보여줍니다. 이 경우 클라이언트가 수집 역할을 맡아야 파이프라인에 액세스할 수 있습니다. 자세한 내용은 the section called “계정 간 수집” 단원을 참조하십시오.

간단한 설정 안내서는 the section called “튜토리얼: 도메인에 데이터 수집”을 참조하세요.

주제

- the section called “파이프라인 역할”
- the section called “수집 역할”

- the section called “계정 간 수집”

파이프라인 역할

파이프라인은 소스에서 읽고 싱크에 쓸 수 있는 특정 권한이 필요합니다. 이러한 권한 AWS 서비스은 클라이언트 애플리케이션 또는 파이프라인에 쓰는 , 싱크가 OpenSearch Service 도메인인지, OpenSearch Serverless 컬렉션인지 또는 Amazon S3인지에 따라 달라집니다. 또한 파이프라인에는 소스 애플리케이션에서 데이터를 물리적으로 가져올 수 있는 권한(소스가 풀 기반 플러그인인 경우)과 활성화된 경우 S3 배달 못한 편지 대기열에 쓸 수 있는 권한이 필요할 수 있습니다.

파이프라인을 생성할 때 수동으로 생성한 기존 IAM 역할을 지정하거나 선택한 소스 및 싱크를 기반으로 OpenSearch Ingestion이 파이프라인 역할을 자동으로 생성하도록 할 수 있습니다. 다음 이미지는에서 파이프라인 역할을 지정하는 방법을 보여줍니다 AWS Management Console.

주제

- 파이프라인 역할 생성 자동화
- 파이프라인 역할 수동 생성

파이프라인 역할 생성 자동화

OpenSearch Ingestion이 파이프라인 역할을 생성하도록 선택할 수 있습니다. 구성된 소스 및 싱크를 기반으로 역할에 필요한 권한을 자동으로 식별합니다. 접두사 `OpenSearchIngestion-`와 입력한 접미사를 사용하여 IAM 역할을 생성합니다. 예를 들어 접미사`PipelineRole`로를 입력하면 OpenSearch Ingestion은 라는 역할을 생성합니다`OpenSearchIngestion-PipelineRole`.

파이프라인 역할을 자동으로 생성하면 설정 프로세스가 간소화되고 구성 오류가 발생할 가능성이 줄어듭니다. 역할 생성을 자동화하면 권한을 수동으로 할당하지 않고도 보안 구성 오류 위험 없이 올바른 정책이 적용되도록 할 수 있습니다. 또한 여러 파이프라인 배포에서 일관성을 유지하면서 모범 사례를 적용하여 시간을 절약하고 보안 규정 준수를 강화합니다.

OpenSearch Ingestion만에서 파이프라인 역할을 자동으로 생성하도록 할 수 있습니다 AWS Management Console. AWS CLI, OpenSearch Ingestion API 또는 SDKs 중 하나를 사용하는 경우 수동으로 생성된 파이프라인 역할을 지정해야 합니다.

OpenSearch Ingestion이 역할을 자동으로 생성하도록 하려면 새 서비스 역할 생성 및 사용을 선택합니다.

Important

파이프라인 역할에 대한 액세스 권한을 부여하려면 도메인 또는 컬렉션 액세스 정책을 수동으로 수정해야 합니다. 세분화된 액세스 제어를 사용하는 도메인의 경우 파이프라인 역할도 백엔드 역할에 매핑해야 합니다. 파이프라인을 생성하기 전이나 후에이 단계를 수행할 수 있습니다.
지침은 다음 주제를 참조하세요.

- 도메인에 대한 데이터 액세스 구성
- 컬렉션에 대한 데이터 및 네트워크 액세스 구성

파이프라인 역할 수동 생성

특정 보안 또는 규정 준수 요구 사항을 충족하기 위해 권한을 더 잘 제어해야 하는 경우 파이프라인 역할을 수동으로 생성하는 것이 좋습니다. 수동 생성을 통해 기존 인프라 또는 액세스 관리 전략에 맞게 역할을 조정할 수 있습니다. 수동 설정을 선택하여 역할을 다른와 통합 AWS 서비스 하거나 고유한 운영 요구 사항에 맞게 조정할 수도 있습니다.

수동으로 생성된 파이프라인 역할을 선택하려면 기존 IAM 역할 사용을 선택하고 기존 역할을 선택합니다. 역할에는 선택한 소스에서 데이터를 수신하고 선택한 싱크에 쓰는 데 필요한 모든 권한이 있어야 합니다. 다음 섹션에서는 파이프라인 역할을 수동으로 생성하는 방법을 간략하게 설명합니다.

주제

- 소스에서 읽을 수 있는 권한
- 도메인 싱크에 쓸 수 있는 권한
- 컬렉션 싱크에 쓸 수 있는 권한
- Amazon S3 또는 배달 못한 편지 대기열에 쓸 수 있는 권한

소스에서 읽을 수 있는 권한

OpenSearch Ingestion 파이프라인에는 지정된 소스에서 데이터를 읽고 수신할 수 있는 권한이 필요합니다. 예를 들어 Amazon DynamoDB 소스의 경우 `dynamodb:DescribeTable` 및와 같은 권한이 필요합니다`dynamodb:DescribeStream`. Amazon S3, Fluent Bit 및 OpenTelemetry Collector와 같은 일반적인 소스에 대한 샘플 파이프라인 역할 액세스 정책은 섹션을 참조하세요the section called “파이프라인 통합”.

도메인 싱크에 쓸 수 있는 권한

OpenSearch Ingestion 파이프라인에는 싱크로 구성된 OpenSearch Service 도메인에 쓸 수 있는 권한이 필요합니다. 이러한 권한에는 도메인을 설명하고 도메인에 HTTP 요청을 보내는 기능이 포함됩니다. 이러한 권한은 퍼블릭 도메인과 VPC 도메인에서 동일합니다. 파이프라인 역할을 생성하고 도메인 액세스 정책에서 지정하는 지침은 파이프라인이 도메인에 액세스하도록 허용을 참조하세요.

컬렉션 싱크에 쓸 수 있는 권한

OpenSearch Ingestion 파이프라인에는 싱크로 구성된 OpenSearch Serverless 컬렉션에 쓸 수 있는 권한이 필요합니다. 이러한 권한에는 컬렉션을 설명하고 컬렉션에 HTTP 요청을 보내는 기능이 포함됩니다.

먼저 파이프라인 역할 액세스 정책이 필요한 권한을 부여하는지 확인합니다. 그런 다음 이 역할을 데이터 액세스 정책에 포함시키고 컬렉션 내에서 인덱스 생성, 인덱스 업데이트, 인덱스 설명, 문서 작성 등의 권한을 부여합니다. 각 단계를 완료하기 위한 지침은 파이프라인의 컬렉션 액세스 허용을 참조하세요.

Amazon S3 또는 배달 못한 편지 대기열에 쓸 수 있는 권한

Amazon S3를 파이프라인의 싱크 대상으로 지정하거나 DLQ(Dead Letter Queue)를 활성화하는 경우 파이프라인 역할은 대상으로 지정한 S3 버킷에 액세스할 수 있도록 허용해야 합니다.

DLQ 액세스를 제공하는 파이프라인 역할에 별도의 권한 정책을 연결합니다. 최소한 버킷 리소스에 대한 `S3:PutObject` 작업을 역할에 부여해야 합니다.

JSON

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

수집 역할

수집 역할은 외부 서비스가 OpenSearch Ingestion 파이프라인과 안전하게 상호 작용하고 데이터를 전송할 수 있도록 하는 IAM 역할입니다. Amazon Security Lake와 같은 푸시 기반 소스의 경우이 역할은를 포함하여 파이프라인으로 데이터를 푸시할 수 있는 권한을 부여해야 합니다`osis:Ingest`. Amazon S3와 같은 풀 기반 소스의 경우 역할은 OpenSearch Ingestion을 활성화하여 이를 수임하고 필요한 권한으로 데이터에 액세스해야 합니다.

주제

- 푸시 기반 소스의 수집 역할
- 풀 기반 소스의 수집 역할
- 계정 간 수집

푸시 기반 소스의 수집 역할

푸시 기반 소스의 경우 데이터는 Amazon Security Lake 또는 Amazon DynamoDB와 같은 다른 서비스에서 수집 파이프라인으로 전송되거나 푸시됩니다. 이 시나리오에서 수집 역할에는 최소한 파이프라인과 상호 작용할 수 있는 `osis:Ingest` 권한이 필요합니다.

다음 IAM 액세스 정책은 수집 역할에이 권한을 부여하는 방법을 보여줍니다.

JSON

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

풀 기반 소스의 수집 역할

풀 기반 소스의 경우 OpenSearch Ingestion 파이프라인은 Amazon S3와 같은 외부 소스에서 데이터를 적극적으로 가져오거나 가져옵니다. 이 경우 파이프라인은 데이터 소스에 액세스하는 데 필요한 권한을 부여하는 IAM 파이프라인 역할을 맡아야 합니다. 이러한 시나리오에서 수집 역할은 파이프라인 역할과 동의어입니다.

역할에는 OpenSearch Ingestion이 이를 수임할 수 있도록 허용하는 신뢰 관계와 데이터 소스와 관련된 권한이 포함되어야 합니다. 자세한 내용은 the section called “소스에서 읽을 수 있는 권한” 단원을 참조하십시오.

계정 간 수집

애플리케이션 계정 AWS 계정과 같은 다른에서 파이프라인으로 데이터를 수집해야 할 수 있습니다. 계정 간 수집을 구성하려면 파이프라인과 동일한 계정 내에 수집 역할을 정의하고 수집 역할과 애플리케이션 계정 간에 신뢰 관계를 설정하세요.

JSON

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

그런 다음, 수집 역할을 맡도록 애플리케이션을 구성하세요. 애플리케이션 계정은 파이프라인 계정의 수집 역할에 대한 AssumeRole 권한을 애플리케이션 역할에 부여해야 합니다.

자세한 단계 및 예제 IAM 정책은 the section called “교차 계정 수집 액세스 제공”을 참조하세요.

도메인에 대한 Amazon OpenSearch Ingestion 파이프라인 액세스 권한 부여

Amazon OpenSearch Ingestion 파이프라인에는 싱크로 구성된 OpenSearch Service 도메인에 쓸 수 있는 권한이 필요합니다. 액세스를 제공하려면 파이프라인이 데이터를 전송하는 도메인에 대한 액세스를 제한하는 제한적인 권한 정책을 사용하여 AWS Identity and Access Management (IAM) 역할을

구성합니다. 예를 들어 수집 파이프라인을 사용 사례를 지원하는 데 필요한 도메인과 인덱스로만 제한할 수 있습니다.

> **Important**
>
> 파이프라인 역할을 수동으로 생성하도록 선택하거나 파이프라인 생성 중에 OpenSearch Ingestion에서 자동으로 생성하도록 할 수 있습니다. 자동 역할 생성을 선택하면 OpenSearch Ingestion은 선택한 소스 및 싱크를 기반으로 파이프라인 역할 액세스 정책에 필요한 모든 권한을 추가합니다. 입력한 접두사`OpenSearchIngestion-`와 접미사를 사용하여 IAM에 파이프라인 역할을 생성합니다. 자세한 내용은 the section called “파이프라인 역할” 단원을 참조하십시오.
>
> OpenSearch Ingestion이 파이프라인 역할을 생성하더라도 파이프라인을 생성하기 전이나 후에 도메인 액세스 정책에 역할을 포함시키고 백엔드 역할(도메인이 세분화된 액세스 제어를 사용하는 경우)에 매핑해야 합니다. 지침은 2단계를 참조하세요.

주제

- 1단계: 파이프라인 역할 생성
- 2단계: 도메인에 대한 데이터 액세스 구성

1단계: 파이프라인 역할 생성

파이프라인 역할에는 도메인 싱크로 데이터를 전송하도록 허용하는 권한 정책이 연결되어 있어야 합니다. 또한 OpenSearch Ingestion이 그 역할을 수임할 수 있도록 하는 신뢰 관계도 있어야 합니다. 정책을 역할에 연결하는 지침은 IAM 사용 설명서의 IAM 자격 증명 권한 추가를 참조하세요.

다음 샘플 정책은 파이프라인 역할에서 단일 도메인에 쓸 수 있는 최소 권한을 보여줍니다.

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/*"
        }
    ]
}
```

역할을 재사용하여 여러 도메인에 쓸 계획이라면 도메인 이름을 와일드카드 문자(*)로 대체하여 정책을 더 광범위하게 적용할 수 있습니다.

역할에는 다음과 같은 신뢰 관계가 있어야 합니다. 그러면 OpenSearch Ingestion이 파이프라인 역할을 맡을 수 있습니다.

JSON

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

2단계: 도메인에 대한 데이터 액세스 구성

파이프라인이 도메인에 데이터를 쓰려면 도메인에 파이프라인 역할이 액세스할 수 있도록 허용하는 도메인 수준 액세스 정책이 있어야 합니다.

다음 샘플 도메인 액세스 정책은 라는 파이프라인 역할이 라는 도메인에 데이터를 `pipeline-role` 쓸 수 있도록 허용합니다`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/*"
        }
    ]
}
```

파이프라인 역할 매핑(세분화된 액세스 제어를 사용하는 도메인에만 해당)

도메인에서 인증에 세분화된 액세스 제어를 사용하는 경우 도메인에 대한 파이프라인 액세스 권한을 제공하기 위해 수행해야 하는 추가 단계가 있습니다. 단계는 도메인 구성에 따라 다릅니다.

- 시나리오 1: 다양한 마스터 역할 및 파이프라인 역할 - IAM Amazon 리소스 이름(ARN)을 마스터 사용자로 사용하고 파이프라인 역할과 다른 경우 파이프라인 역할을 OpenSearch `all_access` 백엔드 역할에 매핑해야 합니다. 그러면 파이프라인 역할이 추가 마스터 사용자로 추가됩니다. 자세한 내용은 추가 마스터 사용자를 참조하세요.
- 시나리오 2: 내부 사용자 데이터베이스의 마스터 사용자 - 도메인에서 내부 사용자 데이터베이스의 마스터 사용자와 OpenSearch Dashboard의 HTTP 기본 인증을 사용하는 경우 마스터 사용자 이름과 비밀번호를 파이프라인 구성으로 직접 전달할 수 없습니다. 대신 파이프라인 역할을 OpenSearch `all_access` 백엔드 역할에 매핑합니다. 그러면 파이프라인 역할이 추가 마스터 사용자로 추가됩니다. 자세한 내용은 추가 마스터 사용자를 참조하세요.
- 시나리오 3: 동일한 마스터 역할 및 파이프라인 역할(흔하지 않음) - IAM ARN을 마스터 사용자로 사용하고 파이프라인 역할로 사용하는 것과 동일한 ARN인 경우 추가 조치를 취할 필요가 없습니다. 파이프라인에는 도메인에 쓰는 데 필요한 권한이 있습니다. 대부분의 환경에서 관리자 역할 또는 다른 역할을 마스터 역할로 사용하기 때문에이 시나리오는 흔하지 않습니다.

다음 이미지는 파이프라인 역할을 백엔드 역할에 매핑하는 방법을 보여줍니다.

컬렉션에 대한 액세스 권한을 Amazon OpenSearch Ingestion 파이프라인에 부여

Amazon OpenSearch Ingestion 파이프라인은 OpenSearch Serverless 퍼블릭 컬렉션 또는 VPC 컬렉션에 쓸 수 있습니다. 컬렉션에 대한 액세스를 제공하려면 컬렉션에 대한 액세스 권한을 부여하는 권한 정책을 사용하여 AWS Identity and Access Management (IAM) 파이프라인 역할을 구성합니다. 파이프라인은 OpenSearch Serverless 컬렉션 싱크에 대한 요청에 서명하기 위해 이 역할을 맡습니다.

> **Important**
>
> 파이프라인 역할을 수동으로 생성하도록 선택하거나 파이프라인 생성 중에 OpenSearch Ingestion에서 자동으로 생성하도록 할 수 있습니다. 자동 역할 생성을 선택하면 OpenSearch Ingestion은 선택한 소스 및 싱크를 기반으로 파이프라인 역할 액세스 정책에 필요한 모든 권한을 추가합니다. 입력한 접두사`OpenSearchIngestion-`와 접미사를 사용하여 IAM에 파이프라인 역할을 생성합니다. 자세한 내용은 the section called “파이프라인 역할” 단원을 참조하십시오.
>
> OpenSearch Ingestion이 파이프라인 역할을 생성하더라도 파이프라인을 생성하기 전이나 후에 컬렉션의 데이터 액세스 정책에 역할을 포함해야 합니다. 지침은 2단계를 참조하세요.

파이프라인 생성 중에 OpenSearch Ingestion은 파이프라인과 OpenSearch Serverless 컬렉션 간에 AWS PrivateLink 연결을 생성합니다. 파이프라인의 모든 트래픽은 이 VPC 엔드포인트를 통과하여 컬렉션으로 라우팅됩니다. 컬렉션에 도달하려면 네트워크 액세스 정책을 통해 컬렉션에 대한 액세스 권한을 엔드포인트에 부여해야 합니다.

주제

- 1단계: 파이프라인 역할 생성
- 2단계: 컬렉션에 대한 데이터 및 네트워크 액세스 구성

1단계: 파이프라인 역할 생성

파이프라인 역할에는 컬렉션 싱크로 데이터를 전송하도록 허용하는 권한 정책이 연결되어 있어야 합니다. 또한 OpenSearch Ingestion이 그 역할을 수임할 수 있도록 하는 신뢰 관계도 있어야 합니다. 정책을 역할에 연결하는 지침은 IAM 사용 설명서의 IAM 자격 증명 권한 추가를 참조하세요.

다음 샘플 정책은 파이프라인 역할 액세스 정책에서 컬렉션에 쓸 수 있는 최소 권한을 보여줍니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll",
                "aoss:BatchGetCollection",
                "aoss:CreateSecurityPolicy",
                "aoss:GetSecurityPolicy",
                "aoss:UpdateSecurityPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

OpenSearch Ingestion이 역할을 수임할 수 있도록 하는 다음 신뢰 관계가 있어야 합니다.

JSON

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

```
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2단계: 컬렉션에 대한 데이터 및 네트워크 액세스 구성

다음 설정을 사용하여 OpenSearch Serverless 컬렉션을 생성합니다. 컬렉션 생성 지침은 the section called “컬렉션 생성” 섹션을 참조하세요.

데이터 액세스 정책

파이프라인 역할에 필요한 권한을 부여하는 컬렉션에 대한 데이터 액세스 정책을 생성합니다. 예:

```
[
  {
    "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**
>
> `Principal` 요소에서 파이프라인 역할의 Amazon 리소스 이름(ARN)을 지정합니다.

네트워크 액세스 정책

OpenSearch Serverless에서 생성하는 각 컬렉션에는 하나 이상의 네트워크 액세스 정책이 연결되어 있습니다. 네트워크 액세스 정책은 퍼블릭 네트워크에서 인터넷을 통해 컬렉션에 액세스할 수 있는지 아니면 프라이빗 액세스로 액세스해야 하는지 여부를 결정합니다. 네트워크 정책에 대한 자세한 내용은 the section called “네트워크 액세스” 섹션을 참조하세요.

네트워크 액세스 정책 내에서 OpenSearch Serverless 관리형 VPC 엔드포인트만 지정할 수 있습니다. 자세한 내용은 the section called “VPC 엔드포인트” 단원을 참조하십시오. 그러나 파이프라인에서 컬렉션에 쓰려면 OpenSearch Ingestion이 파이프라인과 컬렉션 간에 자동으로 생성하는 VPC 엔드포인트에 대한 액세스 권한도 정책에 부여해야 합니다. 따라서 파이프라인의 대상 싱크로 OpenSearch Serverless 컬렉션을 선택하는 경우 네트워크 정책 이름 필드에 연결된 네트워크 정책의 이름을 입력해야 합니다.

파이프라인 생성 중에 OpenSearch Ingestion은 지정된 네트워크 정책이 있는지 확인합니다. 존재하지 않는 경우 OpenSearch Ingestion에서 생성합니다. 존재하는 경우 OpenSearch Ingestion은 새 규칙을 추가하여 업데이트합니다. 이 규칙에서 파이프라인과 컬렉션을 연결하는 VPC 엔드포인트에 대한 액세스 권한을 부여합니다.

예:

```
{
   "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"
}
```

콘솔에서 OpenSearch Ingestion이 네트워크 정책에 추가하는 모든 규칙의 이름은 Data Prepper에서 생성됨으로 지정됩니다.

> **Note**
>
> 일반적으로 컬렉션에 대한 퍼블릭 액세스를 지정하는 규칙은 프라이빗 액세스를 지정하는 규칙을 재정의합니다. 따라서 정책에 이미 퍼블릭 액세스가 구성된 경우 OpenSearch Ingestion이 추가하는 이 새 규칙은 실제로 정책의 동작을 변경하지 않습니다. 자세한 내용은 the section called “정책 우선순위” 단원을 참조하십시오.

파이프라인을 중지하거나 삭제하면 OpenSearch Ingestion은 파이프라인과 컬렉션 사이의 VPC 엔드포인트를 삭제합니다. 또한 허용된 엔드포인트 목록에서 VPC 엔드포인트를 제거하도록 네트워크 정책을 수정합니다. 파이프라인을 다시 시작하면 VPC 엔드포인트가 다시 생성되고 네트워크 정책이 엔드포인트 ID로 다시 업데이트됩니다.

Amazon OpenSearch Ingestion 시작하기

Amazon OpenSearch Ingestion은 관리형 OpenSearch Service 도메인 및 OpenSearch Serverless 컬렉션으로의 데이터 수집을 지원합니다. 다음 자습서에서는 파이프라인을 시작하고 실행하기 위한 기본 단계를 안내합니다.

첫 번째 자습서에서는 Amazon OpenSearch Ingestion을 사용하여 간단한 파이프라인을 구성하고 Amazon OpenSearch Service 도메인으로 데이터를 수집하는 방법을 보여줍니다.

두 번째 자습서에서는 Amazon OpenSearch Ingestion을 사용하여 간단한 파이프라인을 구성하고 Amazon OpenSearch Serverless 컬렉션으로 데이터를 수집하는 방법을 보여줍니다.

> **Note**
>
> 올바른 권한을 설정하지 않으면 파이프라인 생성이 실패합니다. 파이프라인을 생성하기 전에 필요한 역할을 더 잘 이해하려면 the section called “역할 및 사용자 설정”(을)를 참조하세요.

주제

- 튜토리얼: Amazon OpenSearch Ingestion을 사용하여 도메인에 데이터 수집
- 튜토리얼: Amazon OpenSearch Ingestion을 사용하여 컬렉션에 데이터 수집

튜토리얼: Amazon OpenSearch Ingestion을 사용하여 도메인에 데이터 수집

이 튜토리얼에서는 Amazon OpenSearch Ingestion을 사용하여 간단한 파이프라인을 구성하고 Amazon OpenSearch Service 도메인에 데이터를 수집하는 방법을 보여줍니다. 파이프라인은 OpenSearch Ingestion이 프로비저닝하고 관리하는 리소스입니다. 파이프라인을 사용하여 OpenSearch Service의 다운스트림 분석 및 시각화를 위해 데이터를 필터링, 강화, 변환, 정규화 및 집계할 수 있습니다.

이 튜토리얼은 파이프라인을 준비하여 빠르게 실행하기 위한 기본 단계를 안내합니다. 자세한 지침은 섹션을 참조하세요the section called “파이프라인 생성”.

이 튜토리얼에서는 다음 단계를 완료합니다.

1. 도메인 생성.
2. 파이프라인 생성.
3. 일부 샘플 데이터 수집.

이 튜토리얼에서는 다음 리소스를 생성합니다.

- 파이프라인`ingestion-domain`이 쓰는 라는 도메인
- `ingestion-pipeline`이라는 파이프라인

필수 권한

이 자습서를 완료하려면 사용자 또는 역할에 다음과 같은 최소 권한이 있는 자격 증명 기반 정책이 연결되어 있어야 합니다. 이러한 권한을 통해 파이프라인 역할을 생성하고 정책(`iam:Create*` 및 `iam:Attach*`)을 연결하고, 도메인(`es:*`)을 생성 또는 수정하고, 파이프라인()으로 작업할 수 있습니다`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"
         ]
      }
   ]
}
```

1단계: 파이프라인 역할 생성

먼저 OpenSearch Serverless 도메인 싱크에 액세스하기 위해 파이프라인이 맡을 역할을 생성합니다. 이 튜토리얼의 뒷부분에서 파이프라인 구성에 이 역할을 포함시킬 것입니다.

파이프라인 역할을 생성하려면

1. https://console.aws.amazon.com/iamv2/ AWS Identity and Access Management 콘솔을 엽니다.
2. 정책을 선택한 후 정책 생성을 선택합니다.
3. 이 튜토리얼에서는 다음 단계에서 생성할 `ingestion-domain`이라는 도메인으로 데이터를 수집해 보겠습니다. JSON을 선택한 후 다음 정책을 편집기에 붙여넣습니다. *`your-account-id`*(을)를 계정 ID로 바꾸고 필요한 경우 리전을 수정하세요.

   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/*"
        }
    ]
}
```

기존 도메인에 데이터를 쓰려면 `ingestion-domain`(을)를 도메인 이름으로 바꾸세요.

> **Note**
>
> 이 자습서에서는 간소화를 위해 광범위한 액세스 정책을 사용합니다. 하지만 프로덕션 환경에서는 파이프라인 역할에 보다 제한적인 액세스 정책을 적용하는 것이 좋습니다. 필요한 최소 권한을 제공하는 예제 정책은 the section called “도메인에 대한 파이프라인 액세스 권한 부여”(을)를 참조하세요.

4. 다음을 선택하고 다음을 선택한 후, 정책 이름을 파이프라인-정책으로 지정합니다.
5. 정책 생성을 선택합니다.
6. 다음으로, 역할을 생성하여 역할에 정책을 연결합니다. 역할을 선택한 다음 역할 생성을 선택합니다.
7. 사용자 지정 신뢰 정책을 선택하고 다음 정책을 편집기에 붙여넣습니다.

   JSON

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

```
        }
    ]
}
```

8. 다음을 선택합니다. 그런 다음 방금 생성한 파이프라인-정책을 검색하여 선택합니다.
9. 다음을 선택하고 역할 이름을 PipelineRole로 지정합니다.
10. 역할 생성을 선택합니다.

역할의 Amazon 리소스 이름(ARN)을 기억하세요(예: `arn:aws:iam::your-account-id:role/PipelineRole`). 파이프라인을 생성할 때 사용합니다.

2단계: 도메인 생성

먼저 데이터를 수집할 `ingestion-domain` 라는 도메인을 생성합니다.

https://console.aws.amazon.com/aos/home 에서 Amazon OpenSearch Service 콘솔로 이동하여 다음 필요 조건을 충족하는 도메인을 생성합니다.

- OpenSearch 1.0 이상 또는 Elasticsearch 7.4 이상을 실행합니다
- 퍼블릭 액세스 사용
- 세분화된 액세스 제어 사용 금지

> **Note**
>
> 이러한 요구 사항은 이 튜토리얼에서 단순성을 보장하기 위한 것입니다. 프로덕션 환경에서는 VPC 액세스가 가능한 도메인을 구성하거나 세분화된 액세스 제어를 사용할 수 있습니다. 세분화된 액세스 제어를 사용하려면 파이프라인 역할 매핑을 참조하세요.

도메인에는 다음 단계에서 OpenSearch Service가 생성하는 `OpenSearchIngestion-PipelineRole` IAM 역할에 권한을 부여하는 액세스 정책이 있어야 합니다. 파이프라인은 도메인 싱크로 데이터를 전송하기 위해이 역할을 수임합니다.

도메인에 파이프라인 역할에 도메인에 대한 액세스 권한을 부여하는 다음과 같은 도메인 수준 액세스 정책이 있는지 확인합니다. 을 리전으로 바꾸고 를 계정 ID로 바꿉니다.

JSON

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

도메인 수준 액세스 정책 생성에 대한 자세한 내용은 섹션을 참조하세요the section called “리소스 기반 정책”.

이미 도메인을 생성한 경우 기존의 액세스 정책을 수정하여 위의 권한을 `OpenSearchIngestion-PipelineRole`에 제공하세요.

3단계: 파이프라인 생성

이제 도메인이 있으므로 파이프라인을 생성할 수 있습니다.

파이프라인을 만들려면

1. Amazon OpenSearch Service 콘솔 내 왼쪽 탐색 창에서 파이프라인을 선택합니다.
2. [파이프라인 생성]을 선택합니다.
3. 빈 파이프라인을 선택한 다음 블루프린트 선택을 선택합니다.
4. 이 자습서에서는 HTTP 소스 플러그인을 사용하는 간단한 파이프라인을 생성합니다. 플러그인은 JSON 배열 형식의 로그 데이터를 받아들입니다. 단일 OpenSearch Service 도메인을 싱크로 지정하고 모든 데이터를 `application_logs` 인덱스로 수집합니다.

   소스 메뉴에서 HTTP를 선택합니다. 경로에 /logs를 입력합니다.
5. 이 튜토리얼에서는 간소화를 위해 파이프라인에 대한 공개 액세스를 구성해 보겠습니다. 소스 네트워크 옵션에서 퍼블릭 액세스를 선택합니다. VPC에 대한 액세스를 구성하는 방법에 대한 자세한 정보는 the section called “파이프라인에 대한 VPC 액세스 구성” 섹션을 참조하세요.

6. 다음을 선택합니다.
7. 프로세서에 날짜를 입력하고 추가를 선택합니다.
8. 수신된 시작 시간을 활성화합니다. 다른 모든 설정은 기본값으로 둡니다.
9. 다음을 선택합니다.
10. 싱크 세부 정보를 구성합니다. OpenSearch 리소스 유형에서 관리형 클러스터를 선택합니다. 그런 다음 이전 섹션에서 생성한 OpenSearch Service 도메인을 선택합니다.

    인덱스 이름에 application_logs를 입력합니다. OpenSearch Ingestion은 도메인이 아직 없는 경우 도메인에이 인덱스를 자동으로 생성합니다.
11. 다음을 선택합니다.
12. 파이프라인 이름을 ingestion-pipeline으로 지정합니다. 용량 설정을 기본값으로 둡니다.
13. 파이프라인 역할에서 새 서비스 역할 생성 및 사용을 선택합니다. 파이프라인 역할은 파이프라인이 도메인 싱크에 쓰고 풀 기반 소스에서 읽는 데 필요한 권한을 제공합니다. 이 옵션을 선택하면 OpenSearch Ingestion이 IAM에서 역할을 수동으로 생성하는 대신 역할을 생성하도록 허용할 수 있습니다. 자세한 내용은 the section called “역할 및 사용자 설정” 단원을 참조하십시오.
14. 서비스 역할 이름 접미사에 PipelineRole을 입력합니다. IAM에서 역할은 형식을 갖습니다`arn:aws:iam::`*`your-account-id`*`:role/OpenSearchIngestion-`**`PipelineRole`**.
15. 다음을 선택합니다. 파이프라인 구성을 검토하고 파이프라인 생성을 선택합니다. 파이프라인이 활성화되려면 5~10분이 걸립니다.

4단계: 일부 샘플 데이터 수집

파이프라인이 `Active` 상태가 되면 데이터 수집을 시작할 수 있습니다. 서명 버전 4를 사용하여 파이프라인에 대한 모든 HTTP 요청에 서명해야 합니다. Postman 또는 awscurl과 같은 HTTP 도구를 사용하여 파이프라인에 일부 데이터를 전송하세요. 데이터를 컬렉션에 직접 인덱싱하는 것과 마찬가지로, 파이프라인으로 데이터를 수집하려면 항상 IAM 역할 또는 IAM 액세스 키와 암호 키가 필요합니다.

> **Note**
>
> 요청에 서명하는 주체에게는 `osis:Ingest` IAM 권한이 있어야 합니다.

먼저 파이프라인 설정 페이지에서 수집 URL을 가져옵니다.

그런 다음 일부 샘플 데이터를 수집하세요. 다음 요청은 awscurl을 사용하여 단일 로그 파일을 파이프라인으로 보냅니다.

```
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
```

200 OK 응답이 표시되어야 합니다. 인증 오류가 발생하는 경우 파이프라인이 있는 별도의 계정에서 데이터를 수집하고 있기 때문일 수 있습니다. the section called “권한 문제 해결”을(를) 참조하세요.

이제 application_logs 인덱스를 쿼리하여 로그 항목이 성공적으로 수집되었는지 확인하세요.

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

샘플 응답:

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


권한 문제 해결

자습서의 단계를 따랐지만 데이터를 수집하려고 할 때 여전히 인증 오류가 표시되는 경우 파이프라인에 쓰는 역할이 파이프라인 자체 AWS 계정 와 다른에 있기 때문일 수 있습니다. 이 경우 데이터를 수집할 수 있도록 특별히 지원하는 역할을 만들고 수임해야 합니다. 지침은 the section called “교차 계정 수집 액세스 제공” 섹션을 참조하세요.

관련 리소스

이 튜토리얼에서는 HTTP를 통해 단일 문서를 수집하는 간단한 사용 사례를 제시했습니다. 프로덕션 시나리오에서는 하나 이상의 파이프라인으로 데이터를 전송하도록 클라이언트 애플리케이션(예: Fluent Bit, Kubernetes 또는 OpenTelemetry Collector)을 구성합니다. 파이프라인은 이 튜토리얼의 간단한 예제보다 더 복잡할 수 있습니다.

클라이언트 구성 및 데이터 수집을 시작하려면 다음 리소스를 참조하세요.

- 파이프라인 생성 및 관리
- OpenSearch Ingestion으로 데이터를 전송하도록 클라이언트 구성
- Data Prepper 설명서

튜토리얼: Amazon OpenSearch Ingestion을 사용하여 컬렉션에 데이터 수집

이 튜토리얼에서는 Amazon OpenSearch Ingestion을 사용하여 간단한 파이프라인을 구성하고 Amazon OpenSearch Serverless 컬렉션에 데이터를 수집하는 방법을 보여줍니다. 파이프라

인은 OpenSearch Ingestion이 프로비저닝하고 관리하는 리소스입니다. 파이프라인을 사용하여 OpenSearch Service의 다운스트림 분석 및 시각화를 위해 데이터를 필터링, 강화, 변환, 정규화 및 집계할 수 있습니다.

프로비저닝된 OpenSearch Service 도메인으로 데이터를 수집하는 방법을 보여주는 튜토리얼은 the section called “튜토리얼: 도메인에 데이터 수집”을 참조하세요.

이 자습서에서는 다음 단계를 완료합니다.

1. 모음을 만듭니다.
2. 파이프라인 생성.
3. 일부 샘플 데이터 수집.

이 튜토리얼에서는 다음 리소스를 생성합니다.

- 파이프라인이 쓸 `ingestion-collection`이라는 컬렉션입니다.
- `ingestion-pipeline-serverless`이라는 파이프라인

필수 권한

이 자습서를 완료하려면 사용자 또는 역할에 다음과 같은 최소 권한이 있는 자격 증명 기반 정책이 연결되어 있어야 합니다. 이러한 권한을 통해 파이프라인 역할을 생성하고 정책(`iam:Create*` 및 `iam:Attach*`)을 연결하고, 컬렉션을 생성 또는 수정하고(`aoss:*`), 파이프라인()으로 작업할 수 있습니다`osis:*`.

또한 파이프라인 역할을 자동으로 생성하고 컬렉션에 데이터를 쓸 수 있도록 OpenSearch Ingestion에 전달하려면 몇 가지 IAM 권한이 필요합니다.

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"
         ]
      }
   ]
}
```

1단계: 컬렉션 생성

먼저 데이터를 수집할 컬렉션을 생성합니다. 컬렉션 이름을 `ingestion-collection`로 지정하겠습니다.

1. https://console.aws.amazon.com/aos/home에서 Amazon OpenSearch Service 콘솔로 이동합니다.
2. 왼쪽 탐색에서 컬렉션을 선택하고 컬렉션 생성을 선택합니다.
3. 컬렉션 이름을 ingestion-collection으로 지정하세요.
4. 보안에서 표준 생성을 선택합니다.
5. 네트워크 액세스 설정에서 액세스 유형을 공개로 변경합니다.
6. 다른 모든 설정을 기본값으로 유지하고 Next(다음)를 선택합니다.
7. 이제 컬렉션에 대한 데이터 액세스 정책을 구성합니다. 액세스 정책 설정 자동 일치를 선택 취소합니다.
8. 정의 방법에는 JSON을 선택하고 편집기에 다음 정책을 붙여 넣습니다. 이 정책은 두 가지 기능을 합니다.

   - 파이프라인 역할이 컬렉션에 쓸 수 있도록 허용합니다.

- 컬렉션에서 읽을 수 있도록 허용합니다. 나중에 일부 샘플 데이터를 파이프라인으로 수집한 후 컬렉션을 쿼리하여 데이터가 성공적으로 수집되고 인덱스에 기록되었는지 확인합니다.

```
[
  {
    "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. AWS 계정 ID를 포함하도록 `Principal` 요소를 수정합니다. 두 번째 보안 주체의 경우 나중에 컬렉션을 쿼리하는 데 사용할 수 있는 사용자 또는 역할을 지정합니다.
10. 다음을 선택합니다. 액세스 정책 이름을 pipeline-collection-access로 지정하고 다음을 다시 선택합니다.
11. 컬렉션 구성을 검토하고 Submit(제출)을 선택합니다.

2단계: 파이프라인 생성

이제 컬렉션이 있으므로 파이프라인을 생성할 수 있습니다.

파이프라인을 만들려면

1. Amazon OpenSearch Service 콘솔 내 왼쪽 탐색 창에서 파이프라인을 선택합니다.

2. [파이프라인 생성]을 선택합니다.
3. 빈 파이프라인을 선택한 다음 블루프린트 선택을 선택합니다.
4. 이 자습서에서는 HTTP 소스 플러그인을 사용하는 간단한 파이프라인을 생성합니다. 플러그인은 JSON 배열 형식의 로그 데이터를 받아들입니다. 단일 OpenSearch Serverless 컬렉션을 싱크로 지정하고 모든 데이터를 `my_logs` 인덱스로 수집하겠습니다.

   소스 메뉴에서 HTTP를 선택합니다. 경로에 /logs를 입력합니다.
5. 이 튜토리얼에서는 간소화를 위해 파이프라인에 대한 공개 액세스를 구성해 보겠습니다. 소스 네트워크 옵션에서 퍼블릭 액세스를 선택합니다. VPC에 대한 액세스를 구성하는 방법에 대한 자세한 정보는 the section called “파이프라인에 대한 VPC 액세스 구성” 섹션을 참조하세요.
6. 다음을 선택합니다.
7. 프로세서에 날짜를 입력하고 추가를 선택합니다.
8. 수신된 시작 시간을 활성화합니다. 다른 모든 설정은 기본값으로 둡니다.
9. 다음을 선택합니다.
10. 싱크 세부 정보를 구성합니다. OpenSearch 리소스 유형에서 컬렉션(서버리스)을 선택합니다. 그런 다음 이전 섹션에서 생성한 OpenSearch Service 컬렉션을 선택합니다.

    네트워크 정책 이름을 기본값으로 둡니다. 인덱스 이름에 my_logs를 입력합니다. OpenSearch Ingestion은 아직 없는 경우 컬렉션에이 인덱스를 자동으로 생성합니다.
11. 다음을 선택합니다.
12. 파이프라인 이름을 ingestion-pipeline-serverless로 지정합니다. 용량 설정을 기본값으로 둡니다.
13. 파이프라인 역할에서 새 서비스 역할 생성 및 사용을 선택합니다. 파이프라인 역할은 파이프라인이 컬렉션 싱크에 쓰고 풀 기반 소스에서 읽는 데 필요한 권한을 제공합니다. 이 옵션을 선택하면 OpenSearch Ingestion이 IAM에서 역할을 수동으로 생성하는 대신 역할을 생성하도록 허용할 수 있습니다. 자세한 내용은 the section called “역할 및 사용자 설정” 단원을 참조하십시오.
14. 서비스 역할 이름 접미사에 PipelineRole을 입력합니다. IAM에서 역할은 형식을 갖습니다`arn:aws:iam::`*`your-account-id`*`:role/OpenSearchIngestion-`**`PipelineRole`**.
15. 다음을 선택합니다. 파이프라인 구성을 검토하고 파이프라인 생성을 선택합니다. 파이프라인이 활성화되려면 5~10분이 걸립니다.

3단계: 일부 샘플 데이터 수집

파이프라인이 `Active` 상태가 되면 데이터 수집을 시작할 수 있습니다. 서명 버전 4를 사용하여 파이프라인에 대한 모든 HTTP 요청에 서명해야 합니다. Postman 또는 awscurl과 같은 HTTP 도구를 사용

하여 파이프라인에 일부 데이터를 전송하세요. 데이터를 컬렉션에 직접 인덱싱하는 것과 마찬가지로, 파이프라인으로 데이터를 수집하려면 항상 IAM 역할 또는 IAM 액세스 키와 암호 키가 필요합니다.

> **Note**
>
> 요청에 서명하는 주체에게는 `osis:Ingest` IAM 권한이 있어야 합니다.

먼저 파이프라인 설정 페이지에서 수집 URL을 가져옵니다.

그런 다음 일부 샘플 데이터를 수집 경로로 전송합니다. 다음 샘플 요청은 awscurl을 사용하여 단일 로그 파일을 파이프라인으로 보냅니다.

```
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
```

`200 OK` 응답이 표시되어야 합니다.

이제 `my_logs` 인덱스를 쿼리하여 로그 항목이 성공적으로 수집되었는지 확인하세요.

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

샘플 응답:

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


관련 리소스

이 튜토리얼에서는 HTTP를 통해 단일 문서를 수집하는 간단한 사용 사례를 제시했습니다. 프로덕션 시나리오에서는 하나 이상의 파이프라인으로 데이터를 전송하도록 클라이언트 애플리케이션(예: Fluent Bit, Kubernetes 또는 OpenTelemetry Collector)을 구성합니다. 파이프라인은 이 튜토리얼의 간단한 예제보다 더 복잡할 수 있습니다.

클라이언트 구성 및 데이터 수집을 시작하려면 다음 리소스를 참조하세요.

- 파이프라인 생성 및 관리
- OpenSearch Ingestion으로 데이터를 전송하도록 클라이언트 구성
- Data Prepper 설명서

Amazon OpenSearch Ingestion 파이프라인 기능 개요

Amazon OpenSearch Ingestion은 소스, 버퍼, 0개 이상의 프로세서, 하나 이상의 싱크로 구성된 파이프라인을 프로비저닝합니다. 수집 파이프라인은 데이터 엔진인 Data Prepper에 의해 구동됩니다. 파이프라인의 다양한 구성 요소에 대한 개요는 the section called “주요 개념”(을)를 참조하세요.

다음 섹션에는 Amazon OpenSearch Ingestion에서 가장 일반적으로 사용되는 기능 중 일부에 대한 개요가 나와 있습니다.

> **Note**
>
> 이 목록은 파이프라인에서 사용할 수 있는 기능 중 전체 목록이 아닙니다. 사용 가능한 모든 파이프라인 기능에 대한 포괄적인 설명서는 Data Prepper 설명서를 참조하세요. 참고로 OpenSearch Ingestion은 사용할 수 있는 플러그인과 옵션에 몇 가지 제약을 가합니다. 자세한 내용은 the section called “지원되는 작업 및 플러그인” 단원을 참조하십시오.

주제

- 영구 버퍼링
- 분할
- Chaining
- 배달 못한 편지 대기열
- 인덱스 관리
- 엔드 투 엔드 승인
- 소스 역압

영구 버퍼링

영구 버퍼는 여러 가용 영역에 걸쳐 디스크 기반 버퍼에 데이터를 저장하여 데이터 내구성을 향상시킵니다. 영구 버퍼링을 사용하면 독립 실행형 버퍼를 설정하지 않고도 지원되는 모든 푸시 기반 소스에서 데이터를 수집할 수 있습니다. 이러한 소스에는 로그, 추적 및 지표에 대한 HTTP 및 OpenTelemetry가 포함됩니다. 영구 버퍼링을 활성화하려면 파이프라인을 생성하거나 업데이트할 때 영구 버퍼 활성화를 선택합니다. 자세한 내용은 the section called “파이프라인 생성” 단원을 참조하십시오.

OpenSearch Ingestion은 영구 버퍼링, 데이터 소스 고려, 스트리밍 변환 및 싱크 대상에 사용할 OCUs 수를 동적으로 결정합니다. 버퍼링에 일부 OCUs 할당하므로 동일한 수집 처리량을 유지하기 위해 최

소 및 최대 OCUs를 늘려야 할 수 있습니다. 파이프라인은 최대 72시간 동안 버퍼에 데이터를 보관합니다.

파이프라인에 대해 영구 버퍼링을 활성화하면 기본 최대 요청 페이로드 크기는 다음과 같습니다.

- HTTP 소스 - 10MB
- OpenTelemetry 소스 - 4MB

HTTP 소스의 경우 최대 페이로드 크기를 20MB로 늘릴 수 있습니다. 요청 페이로드 크기에는 일반적으로 여러 이벤트가 포함된 전체 HTTP 요청이 포함됩니다. 각 이벤트는 3.5MB를 초과할 수 없습니다.

영구 버퍼링이 있는 파이프라인은 구성된 파이프라인 단위를 컴퓨팅 단위와 버퍼 단위로 분할합니다. 파이프라인이 grok, 키-값 또는 분할 문자열과 같은 CPU 집약적 프로세서를 사용하는 경우 단위를 1:1 buffer-to-compute 비율로 할당합니다. 그렇지 않으면 3:1 비율로 할당되므로 항상 컴퓨팅 단위를 선호합니다.

예:

- grok 및 최대 2개의 단위가 있는 파이프라인 - 컴퓨팅 단위 1개 및 버퍼 단위 1개
- grok 및 최대 5개의 단위가 있는 파이프라인 - 컴퓨팅 단위 3개 및 버퍼 단위 2개
- 프로세서가 없고 최대 단위가 2개인 파이프라인 - 컴퓨팅 단위 1개 및 버퍼 단위 1개
- 프로세서가 없고 최대 단위가 4개인 파이프라인 - 컴퓨팅 단위 1개 및 버퍼 단위 3개
- grok 및 최대 5개의 단위가 있는 파이프라인 - 컴퓨팅 단위 2개 및 버퍼 단위 3개

기본적으로 파이프라인은를 사용하여 버퍼 데이터를 암호화 AWS 소유 키 합니다. 이 파이프라인에는 파이프라인 역할에 대한 추가 권한이 필요하지 않습니다.

또는 고객 관리형 키를 지정하고 파이프라인 역할에 다음 IAM 권한을 추가할 수 있습니다.

JSON

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

```
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "arn:aws:kms:us-east-1:{aws-account-
id}:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키를 참조하세요.

> **Note**
>
> 영구 버퍼링을 비활성화하면 파이프라인이 인 메모리 버퍼링에서 완전히 실행되기 시작합니다.

분할

수신 이벤트를 하위 파이프라인으로 분할하도록 OpenSearch Ingestion 파이프라인을 구성하여 동일한 수신 이벤트에 대해 다양한 유형의 처리를 수행할 수 있습니다.

다음 예제 파이프라인은 수신 이벤트를 두 개의 하위 파이프라인으로 분할합니다. 각 하위 파이프라인은 자체 프로세서를 사용하여 데이터를 보강하고 조작한 다음 데이터를 다양한 OpenSearch 인덱스로 보냅니다.

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

logs_enriched_one_pipeline:
  source:
```

```
    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:
    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"
```

Chaining

데이터 처리 및 보강을 청크 단위로 수행하기 위해 여러 하위 파이프라인을 함께 연결할 수 있습니다. 즉, 수신 이벤트를 하나의 하위 파이프라인에서 특정 처리 기능으로 보강한 다음 다른 하위 파이프라인으로 전송하여 다른 프로세서로 추가 보강하고 마지막으로 해당 OpenSearch 싱크로 보낼 수 있습니다.

다음 예제에서 `log_pipeline` 하위 파이프라인은 수신 로그 이벤트를 프로세서 집합으로 보강한 `enriched_logs`이라는 OpenSearch 인덱스로 이벤트를 보냅니다. 파이프라인은 동일한 이벤트를 `log_advanced_pipeline` 하위 파이프라인으로 전송하며, 이는 해당 이벤트를 처리하여 `enriched_advanced_logs`이라는 다른 OpenSearch 인덱스로 보냅니다.

```
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:
    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"
```

배달 못한 편지 대기열

DLQ(Dead Letter Queue)는 파이프라인이 싱크에 기록하지 못하는 이벤트의 대상입니다. OpenSearch Ingestion에서는 DLQ로 사용할 적절한 쓰기 권한이 있는 Amazon S3 버킷을 지정해야 합니다. 파이프라인 내의 모든 싱크에 DLQ 구성을 추가할 수 있습니다. 파이프라인에서 쓰기 오류가 발생하면 구성된 S3 버킷에 DLQ 객체가 생성됩니다. DLQ 객체는 JSON 파일 내에 실패한 이벤트의 배열로 존재합니다.

다음 조건 중 하나가 충족되면 파이프라인이 DLQ에 이벤트를 기록합니다.

- OpenSearch 싱크의 최대 재시도 횟수가 소진되었습니다. OpenSearch Ingestion은이 설정에 최소 16개가 필요합니다.
- 싱크가 오류 조건으로 인해 이벤트를 거부하고 있습니다.

구성

하위 파이프라인에 대한 배달 못한 편지 대기열을 구성하려면 싱크 대상을 구성할 때 S3 DLQ 활성화를 선택합니다. 그런 다음 대기열에 필요한 설정을 지정합니다. 자세한 내용은 Data Prepper DLQ 설명서의 구성을 참조하세요.

이 S3 DLQ에 작성된 파일의 이름 지정 패턴은 다음과 같습니다.

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

DLQ가 쓰는 S3 버킷에 대한 액세스를 허용하도록 파이프라인 역할을 수동으로 구성하는 지침은 섹션을 참조하세요the section called “Amazon S3 또는 배달 못한 편지 대기열에 쓸 수 있는 권한”.

예시

다음 예제 DLQ 파일을 고려하세요.

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

다음은 싱크에 기록되지 않고 추가 분석을 위해 DLQ S3 버킷으로 전송되는 데이터의 예입니다.

```
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"
```

인덱스 관리

Amazon OpenSearch Ingestion에는 다음을 비롯한 다양한 인덱스 관리 기능이 있습니다.

인덱스 만들기

파이프라인 싱크에 인덱스 이름을 지정할 수 있으며, OpenSearch Ingestion은 파이프라인을 프로비저닝할 때 인덱스를 생성합니다. 인덱스가 이미 있는 경우 파이프라인은 해당 인덱스를 사용하여 수신하는 이벤트를 인덱싱합니다. 파이프라인을 중지했다가 다시 시작하거나 YAML 구성을 업데이트하면 파이프라인은 새 인덱스가 아직 없는 경우 새 인덱스를 만들려고 시도합니다. 파이프라인은 인덱스를 삭제할 수 없습니다.

다음 예제 싱크는 파이프라인이 프로비저닝될 때 두 개의 인덱스를 생성합니다.

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

인덱스 이름 및 패턴 생성

수신 이벤트 필드의 변수를 사용하여 동적 인덱스 이름을 생성할 수 있습니다. 싱크 구성에서는 형식 `string${}`을 사용하여 문자열 보간을 알리고 JSON 포인터를 사용하여 이벤트에서 필드를 추출합니다. `index_type`의 옵션은 `custom` 또는 `management_disabled`입니다. OpenSearch 도메인의 `custom`과 OpenSearch 서버리스 컬렉션의 `management_disabled`에는 `index_type` 기본값이 사용되므로 설정하지 않은 상태로 둘 수 있습니다.

예를 들어 다음 파이프라인은 수신 이벤트에서 `metadataType` 필드를 선택하여 인덱스 이름을 생성합니다.

```
pipeline:
  ...
```

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

다음 구성은 매일 또는 1시간마다 새 인덱스를 계속 생성합니다.

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

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

인덱스 이름은 `my-index-%{yyyy.MM.dd}`와 같은 접미사로 날짜-시간 패턴을 포함하는 일반 문자열일 수도 있습니다. 싱크가 OpenSearch로 데이터를 보내면 날짜-시간 패턴을 UTC 시간으로 바꾸고 각 날짜에 대해 `my-index-2022.01.25`와 같은 새 인덱스를 생성합니다. 자세한 내용은 DateTimeformatter 클래스를 참조하세요.

이 인덱스 이름은 `my-${index}-name`(와)과 같은 접미사로 날짜-시간 패턴을 포함 또는 미포함하는 문자열 형식일 수도 있습니다. 싱크가 OpenSearch로 데이터를 보내면 `"${index}"` 부분이 처리 중인 이벤트의 값으로 바뀝니다. 형식이 `"${index1/index2/index3}"`인 경우 필드 `index1/index2/index3`가 이벤트의 값으로 대체합니다.

문서 ID 생성

파이프라인은 문서를 OpenSearch에 인덱싱하는 동안 문서 ID를 생성할 수 있습니다. 수신 이벤트 내의 필드에서 이러한 문서 ID를 유추할 수 있습니다.

이 예제에서는 수신 이벤트의 `uuid` 필드를 사용하여 문서 ID를 생성합니다.

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

```
      "document_id": "uuid"
```

다음 예제에서 항목 추가 프로세서는 수신 이벤트의 uuid 및 other_field 필드를 병합하여 문서 ID를 생성합니다.

create 작업을 수행하면 ID가 동일한 문서를 덮어쓰지 않습니다. 파이프라인은 재시도 또는 DLQ 이벤트 없이 중복 문서를 삭제합니다. 기존 문서를 업데이트하지 않는 것이 목적이므로 이 작업을 사용하는 파이프라인 작성자에게는 이 작업을 예상하는 것이 합리적입니다.

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

이벤트의 문서 ID를 하위 객체의 필드로 설정하고 싶을 수도 있습니다. 다음 예제에서 OpenSearch 싱크 플러그인은 하위 개체 info/id를 사용하여 문서 ID를 생성합니다.

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

다음 이벤트가 발생하면 파이프라인은 json001로 설정된 _id 필드로 문서를 생성합니다.

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

라우팅 ID 생성

OpenSearch 싱크 플러그인 내 routing_field 옵션을 사용하여 문서 라우팅 속성(_routing)의 값을 수신 이벤트의 값으로 설정할 수 있습니다.

라우팅은 JSON 포인터 구문을 지원하므로 최상위 필드뿐 아니라 중첩된 필드도 사용할 수 있습니다.

```
sink:
  - opensearch:
        ...
        routing_field: metadata/id
        document_id: id
```

다음 이벤트가 발생하면 플러그인은 abcd로 설정된 _routing 필드로 문서를 생성합니다.

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

파이프라인이 인덱스 생성 중에 사용할 수 있는 인덱스 템플릿을 만드는 방법에 대한 지침은 인덱스 템플릿을 참조하세요.

엔드 투 엔드 승인

OpenSearch Ingestion은 엔드 투 엔드 승인을 사용하여 상태 비저장 파이프라인의 소스에서 싱크까지의 데이터 전달을 추적함으로써 데이터의 내구성과 신뢰성을 보장합니다.

> **Note**
>
> 현재는 S3 소스 플러그인만 엔드 투 엔드 승인을 지원합니다.

엔드 투 엔드 승인을 통해 파이프라인 소스 플러그인은 승인 세트를 생성하여 이벤트 배치를 모니터링합니다. 이벤트가 싱크로 성공적으로 전송되면 긍정적인 승인을 받고, 싱크로 전송할 수 없는 이벤트가 있을 때는 부정적인 승인을 받습니다.

파이프라인 구성 요소에 장애 또는 충돌이 발생하거나 소스가 승인을 받지 못하는 경우 소스는 제한 시간이 초과되어 실패 재시도 또는 로깅과 같은 필요한 조치를 취합니다. 파이프라인에 싱크가 여러 개 있거나 하위 파이프라인이 여러 개 구성된 경우 이벤트가 모든 하위 파이프라인의 모든 싱크에 전송된 후에만 이벤트 수준 승인이 전송됩니다. 싱크에 DLQ가 구성된 경우 엔드 투 엔드 승인은 DLQ에 기록된 이벤트도 추적합니다.

종단 간 승인을 활성화하려면 Amazon S3 소스 구성에서 추가 옵션을 확장하고 end-to-end 메시지 승인 활성화를 선택합니다.

소스 역압

파이프라인이 데이터를 처리하느라 바쁠 때, 싱크가 일시적으로 다운되거나 데이터 수집 속도가 느릴 경우 역압이 발생할 수 있습니다. OpenSearch Ingestion은 파이프라인이 사용하는 소스 플러그인에 따라 배압을 처리하는 방법이 다릅니다.

HTTP 소스

HTTP 소스 플러그인을 사용하는 파이프라인은 혼잡한 파이프라인 구성 요소에 따라 배압을 다르게 처리합니다.

- 버퍼 - 버퍼가 가득 차면 파이프라인이 오류 코드 408과 함께 HTTP 상태 REQUEST_TIMEOUT를 소스 엔드포인트로 반환하기 시작합니다. 버퍼가 비워지면 파이프라인은 HTTP 이벤트 처리를 다시 시작합니다.
- 소스 스레드 — 모든 HTTP 소스 스레드가 요청을 실행하는 중이고 처리되지 않은 요청 대기열 크기가 허용된 최대 요청 수를 초과하면 파이프라인은 오류 코드 429와 함께 HTTP 상태 TOO_MANY_REQUESTS를 소스 엔드포인트로 반환하기 시작합니다. 요청 대기열이 최대 허용 대기열 크기 아래로 떨어지면 파이프라인은 요청 처리를 다시 시작합니다.

OTel 소스

OpenTelemetry 소스(OTel 로그, oTel 지표, OTel 추적)를 사용하는 파이프라인의 버퍼가 가득 차면 파이프라인은 오류 코드 408과 함께 HTTP 상태 REQUEST_TIMEOUT를 소스 엔드포인트에 반환하기 시작합니다. 버퍼가 비워지면 파이프라인은 이벤트 처리를 다시 시작합니다.

S3 소스

S3 소스가 있는 파이프라인의 버퍼가 가득 차면 파이프라인의 SQS 알림 처리가 중지됩니다. 버퍼가 비워지면 파이프라인에서 알림 처리를 다시 시작합니다.

싱크가 중단되거나 데이터를 수집할 수 없고 소스에 대한 포괄적인 승인이 활성화된 경우 파이프라인은 모든 싱크로부터 성공적인 승인을 받을 때까지 SQS 알림 처리를 중단합니다.

Amazon OpenSearch Ingestion 파이프라인 생성

파이프라인은 Amazon OpenSearch Ingestion이 데이터를 소스(데이터의 출처)에서 싱크 (데이터가 이동하는 곳)로 이동하는 데 사용하는 메커니즘입니다. OpenSearch Ingestion에서 싱크는 항상 단일 Amazon OpenSearch Service 도메인이지만, 데이터 소스는 Amazon S3, Fluent Bit 또는 OpenTelemetry Collector와 같은 클라이언트일 수 있습니다.

자세한 내용은 OpenSearch 설명서의 파이프라인을 참조하세요.

주제

- 사전 조건 및 필요한 IAM 역할
- 필수 IAM 권한
- 파이프라인 버전 지정
- 수집 경로 지정
- 파이프라인 생성
- 파이프라인 생성 상태 추적
- 블루프린트 작업

사전 조건 및 필요한 IAM 역할

OpenSearch Ingestion 파이프라인을 생성하려면 다음과 같은 리소스가 있어야 합니다.

- OpenSearch Ingestion이 싱크에 쓰기 위해 수임하는 파이프라인 역할이라는 IAM 역할입니다. 이 역할을 미리 생성하거나 파이프라인을 생성하는 동안 OpenSearch Ingestion에서 자동으로 생성하도록 할 수 있습니다.
- 싱크 역할을 하는 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션입니다. 도메인에 쓰는 경우 OpenSearch 1.0 이상 또는 Elasticsearch 7.4 이상이 실행되고 있어야 합니다. 싱크에는 IAM 파이프라인 역할에 적절한 권한을 부여하는 액세스 정책이 있어야 합니다.

이러한 리소스를 만드는 방법 설명은 다음 주제를 참조하세요.

- the section called “도메인에 대한 파이프라인 액세스 권한 부여”
- the section called “컬렉션에 대한 액세스 권한을 파이프라인에 부여”

> **Note**
>
> 세분화된 액세스 제어를 사용하는 도메인에 쓰는 경우 완료해야 할 추가 단계가 있습니다. the section called “파이프라인 역할 매핑(세분화된 액세스 제어를 사용하는 도메인에만 해당)”을(를) 참조하세요.

필수 IAM 권한

OpenSearch Ingestion은 다음 IAM 권한을 사용하여 파이프라인을 생성합니다.

- `osis:CreatePipeline` - 파이프라인을 생성합니다.
- `osis:ValidatePipeline`— 파이프라인 구성이 유효한지 확인하세요.
- `iam:CreateRole` 및 `iam:AttachPolicy` - OpenSearch Ingestion이 자동으로 파이프라인 역할을 생성하도록 합니다.
- `iam:PassRole` - OpenSearch Ingestion에 파이프라인 역할을 전달하여 도메인에 데이터를 쓸 수 있도록 합니다. 이 권한은 파이프라인 역할 리소스에 대한 것이거나 각 파이프라인에서 서로 다른 역할을 사용할 계획인 * 경우에만 필요합니다.

예를 들어 다음 정책에서 파이프라인을 호출할 권한을 부여합니다.

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 Ingestion에는 서명 버전 4를 사용하여 파이프라인에 서명된 요청을 보내는 데 필요한 `osis:Ingest` 권한도 포함되어 있습니다. 자세한 내용은 the section called “수집 역할 생성” 단원을 참조하십시오.

> **Note**
>
> 또한 계정에서 파이프라인을 생성하는 첫 번째 사용자에게 `iam:CreateServiceLinkedRole` 작업에 대한 권한이 있어야 합니다. 자세한 내용은 파이프라인 역할 리소스를 참조하세요.

각 권한에 대한 자세한 내용은 서비스 권한 부여 참조에서 OpenSearch Ingestion에 대한 작업, 리소스 및 조건 키에 대한 액션, 리소스 및 조건 키를 참조하세요.

파이프라인 버전 지정

구성 편집기를 사용하여 파이프라인을 생성할 때는 파이프라인이 실행될 Data Prepper의 메이저 버전을 지정해야 합니다. 버전을 지정하려면 파이프라인 구성에 `version` 옵션을 포함하세요.

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

생성을 선택하면 OpenSearch Ingestion은 지정한 메이저 버전의 사용 가능한 최신 마이너 버전을 확인하고 해당 버전으로 파이프라인을 프로비저닝합니다. 예를 들어 `version: "2"`을 지정하고 Data Prepper의 최신 지원 버전이 2.1.1인 경우 OpenSearch Ingestion은 파이프라인을 버전 2.1.1로 프로비저닝합니다. 파이프라인이 실행 중인 마이너 버전은 공개적으로 표시하지 않습니다.

Data Prepper의 새 메이저 버전이 출시될 때 파이프라인을 업그레이드하려면 파이프라인 구성을 편집하고 새 버전을 지정하세요. 파이프라인을 이전 버전으로 다운그레이드할 수 없습니다.

> **Note**
>
> OpenSearch Ingestion은 새 버전의 Data Prepper가 출시되자마자 바로 지원되지는 않습니다. 새 버전이 공개되는 시점과 OpenSearch Ingestion에서 새 버전이 지원되는 시점 사이에는 약간의 지연이 있을 수 있습니다. 또한 OpenSearch Ingestion은 명시적으로 특정 메이저 또는 마이너 버전을 모두 지원하지 않을 수도 있습니다. 포괄적인 목록은 the section called “지원되는 Data Prepper 버전” 섹션을 참조하십시오.

블루/그린 배포를 시작하는 파이프라인을 변경할 때마다 OpenSearch Ingestion은 파이프라인에 대해 현재 구성된 메이저 버전의 최신 마이너 버전으로 업그레이드할 수 있습니다. 자세한 내용은 the section called “파이프라인 업데이트를 위한 블루/그린 배포” 단원을 참조하십시오. OpenSearch Ingestion은 파이프라인 구성 내에서 `version` 옵션을 명시적으로 업데이트하지 않는 한 파이프라인의 메이저 버전을 변경할 수 없습니다.

수집 경로 지정

OTel 추적 및 oTel 지표와 같은 풀 기반 소스의 경우 OpenSearch Ingestion을 사용하려면 소스 구성에 `path` 옵션이 추가로 필요합니다. 경로는 수집을 위한 URI 경로를 나타내는 `/log/ingest`와 같은 문자열입니다. 이 경로는 파이프라인으로 데이터를 전송하는 데 사용하는 URI를 정의합니다.

예를 들어 HTTP 소스가 있는 파이프라인에 대해 다음 경로를 지정한다고 가정해 보겠습니다.

파이프라인으로 데이터를 수집할 때는 클라이언트 구성에서 `https://`*`pipeline-name-abc123`*`.`*`us-west-2`*`.osis.amazonaws.com/`*`my`*`/`*`test_path`*와 같은 엔드포인트를 지정해야 합니다.

경로는 슬래시(/) 로 시작해야 하며 특수 문자 '-', '_', '.', '/'를 비롯해 `${pipelineName}` 자리 표시자를 포함할 수 있습니다. `${pipelineName}` (예: `/${pipelineName}/test_path`)를 사용하는 경우 OpenSearch Ingestion은 변수를 연결된 하위 파이프라인의 이름으로 바꿉니다.

파이프라인 생성

이 섹션에서는 OpenSearch Service 콘솔 및 AWS CLI을 사용하여 OpenSearch Ingestion 파이프라인을 생성하는 방법을 설명합니다.

콘솔

파이프라인을 생성하려면 https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인하고 파이프라인 생성을 선택합니다.

빈 파이프라인을 선택하거나 구성 블루프린트를 선택합니다. 블루프린트에는 다양한 일반 사용 사례를 위한 사전 구성된 파이프라인이 포함되어 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

블루프린트 선택을 선택합니다.

소스를 구성합니다.

1. 빈 파이프라인에서 시작하는 경우 드롭다운 메뉴에서 소스를 선택합니다. 사용 가능한 소스에는 기타 AWS 서비스, OpenTelemetry 또는 HTTP가 포함될 수 있습니다. 자세한 내용은 the section called “파이프라인 통합” 단원을 참조하십시오.
2. 선택한 소스에 따라 소스에 대한 추가 설정을 구성합니다. 예를 들어 Amazon S3를 소스로 사용하려면 파이프라인에서 메시지를 수신하는 Amazon SQS 대기열의 URL을 지정해야 합니다. 지원되는 소스 플러그인 및 설명서 링크 목록은 섹션을 참조하세요the section called “지원되는 작업 및 플러그인”.
3. 일부 소스의 경우 소스 네트워크 옵션을 지정해야 합니다. VPC 액세스 또는 퍼블릭 액세스를 선택합니다. 퍼블릭 액세스(Public access)를 선택한 경우, 다음 단계로 건너뜁니다. VPC 액세스를 선택하는 경우 다음 설정을 구성하세요.

설정	설명
엔드포인트 관리	Virtual Private Cloud(VPC) 엔드포인트를 직접 생성할지 아니면 OpenSearch Ingestion이 자동으로 생성할지 선택합니다. 엔드포인트 관리는 기본적으로 OpenSearch Ingestion에서 관리하는 엔드포인트로 설정됩니다.
VPC	사용할 VPC의 ID를 선택합니다. VPC와 파이프라인의 AWS 리전(은)는 동일해야 합니다.
서브넷	서브넷을 하나 이상 선택합니다. OpenSearch Service가 서브넷에 VPC 엔드포인트와 탄력적 네트워크 인터페이스를 배치합니다.

설정	설명
보안 그룹	필요한 애플리케이션이 파이프라인에 의해 노출된 포트(80 또는 443) 및 프로토콜(HTTP 또는 HTTPS)에서 OpenSearch Ingestion 파이프라인에 도달하도록 허용하는 VPC 보안 그룹을 하나 이상 선택합니다.
VPC 연결 옵션	소스가 자체 관리형 엔드포인트인 경우 파이프라인을 VPC에 연결합니다. 제공된 기본 CIDR 옵션 중 하나를 선택하거나 사용자 지정 CIDR을 사용합니다.

자세한 내용은 the section called “파이프라인에 대한 VPC 액세스 구성” 단원을 참조하십시오.

4. 다음을 선택합니다.

프로세서 구성

파이프라인에 프로세서를 하나 이상 추가합니다. 프로세서는 도메인 또는 컬렉션 싱크에 레코드를 게시하기 전에 이벤트를 필터링, 변환 및 보강할 수 있는 하위 파이프라인 내의 구성 요소입니다. 지원되는 프로세서 목록과 설명서 링크는 섹션을 참조하세요the section called “지원되는 작업 및 플러그인”.

작업을 선택하고 다음을 추가할 수 있습니다.

- 조건부 라우팅 - 특정 조건에 따라 이벤트를 다른 싱크로 라우팅합니다. 자세한 내용은 조건부 라우팅을 참조하세요.
- 하위 파이프라인 - 각 하위 파이프라인은 단일 소스, 0개 이상의 프로세서 및 단일 싱크의 조합입니다. 하나의 하위 파이프라인에만 외부 소스가 있을 수 있습니다. 다른 모든 소스에는 전체 파이프라인 구성 내의 다른 하위 파이프라인인 소스가 있어야 합니다. 단일 파이프라인 구성에는 1~10개의 하위 파이프라인이 포함될 수 있습니다.

다음을 선택합니다.

싱크 구성

파이프라인이 레코드를 게시하는 대상을 선택합니다. 모든 하위 파이프라인에는 하나 이상의 싱크가 포함되어야 합니다. 파이프라인에 최대 10개의 싱크를 추가할 수 있습니다.

OpenSearch 싱크의 경우 다음 필드를 구성합니다.

설정	설명
네트워크 정책 이름 (서버리스 싱크만 해당)	OpenSearch Serverless 컬렉션을 선택한 경우 네트워크 정책 이름을 입력합니다. OpenSearch Ingestion은 정책이 없는 경우 정책을 생성하거나 파이프라인과 컬렉션을 연결하는 VPC 엔드포인트에 대한 액세스 권한을 부여하는 규칙으로 정책을 업데이트합니다. 자세한 내용은 the section called “컬렉션에 대한 액세스 권한을 파이프라인에 부여” 단원을 참조하십시오.
인덱스 이름	파이프라인이 데이터를 전송하는 인덱스의 이름입니다. OpenSearch Ingestion은 인덱스가 아직 없는 경우이 인덱스를 생성합니다.
인덱스 매핑 옵션	파이프라인이 문서와 해당 필드를 OpenSearch 싱크에 저장하고 인덱싱하는 방법을 선택합니다. 동적 매핑을 선택하면 문서를 인덱싱할 때 OpenSearch가 자동으로 필드를 추가합니다. 매핑 사용자 지정을 선택한 경우 인덱스 매핑 템플릿을 입력합니다. 자세한 내용은 인덱스 템플릿을 참조하세요.
DLQ 활성화	파이프라인에 대한 Amazon S3 DLQ(Dead Letter Queue)를 구성합니다. 자세한 내용은 the section called “배달 못한 편지 대기열” 단원을 참조하십시오.
추가 설정	OpenSearch 싱크에 대한 고급 옵션을 구성합니다. 자세한 내용은 Data Prepper 설명서의 구성 옵션을 참조하세요.

Amazon S3 싱크를 추가하려면 싱크 및 Amazon S3 추가를 선택합니다. 자세한 내용은 the section called “대상으로서의 Amazon S3” 단원을 참조하십시오.

다음을 선택합니다.

파이프라인 구성

다음과 같은 추가 파이프라인 설정을 구성합니다.

설정	설명
파이프라인 이름	파이프라인의 고유한 이름입니다.
영구 버퍼	영구 버퍼는 여러 가용 영역에 걸쳐 디스크 기반 버퍼에 데이터를 저장합니다. 자세한 내용은 the section called “영구 버퍼링” 단원을 참조하십시오.

설정	설명
	영구 버퍼링을 활성화하는 경우 AWS Key Management Service 키를 선택하여 버퍼 데이터를 암호화합니다.
파이프라인 용량	Ingestion OpenSearch 컴퓨팅 단위(OCUs. 자세한 내용은 the section called “파이프라인 크기 조정” 단원을 참조하십시오.
파이프라인 역할	파이프라인이 싱크에 쓰고 풀 기반 소스에서 읽는 데 필요한 권한을 제공하는 IAM 역할입니다. 역할을 직접 생성하거나 선택한 사용 사례에 따라 OpenSearch Ingestion에서 역할을 생성하도록 할 수 있습니다. 자세한 내용은 the section called “역할 및 사용자 설정” 단원을 참조하십시오.
태그	파이프라인에 하나 이상의 태그를 추가합니다. 자세한 내용은 the section called “파이프라인 태그 지정” 단원을 참조하십시오.
로그 게시 옵션	Amazon CloudWatch Logs에 파이프라인 로그 게시를 활성화합니다. 파이프라인 문제를 보다 쉽게 해결할 수 있도록 로그 게시를 활성화하는 것이 좋습니다. 자세한 내용은 the section called “파이프라인 모니터링” 단원을 참조하십시오.

다음을 선택한 후 파이프라인 구성을 검토하고 파이프라인 생성을 선택합니다.

OpenSearch Ingestion은 비동기 프로세스를 실행하여 파이프라인을 구축합니다. 파이프라인이 `Active` 상태가 되면 데이터 수집을 시작할 수 있습니다.

AWS CLI

create-pipeline 명령은 파이프라인 구성을 문자열로 수락하거나 .yaml 또는 .json 파일 내에서 수락합니다. 구성을 문자열로 제공하는 경우 각 새 줄을 `\n`로 이스케이프해야 합니다. 예: `"log-pipeline:\n source:\n http:\n processor:\n - grok:\n ..."`

다음 샘플 명령은 다음과 같은 구성으로 파이프라인을 생성합니다.

- 최소 4개의 Ingestion OCU, 최대 10개의 Ingestion OCU
- Virtual Private Cloud(VPC) 내에서 프로비저닝됨
- 로그 게시 활성

```
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 Ingestion은 비동기 프로세스를 실행하여 파이프라인을 구축합니다. 파이프라인이 `Active` 상태가 되면 데이터 수집을 시작할 수 있습니다. 파이프라인 상태를 확인하려면 GetPipeline 명령을 사용하세요.

OpenSearch Ingestion API

OpenSearch Ingestion API를 사용하여 OpenSearch Ingestion 파이프라인을 생성하려면 CreatePipeline 작업을 호출하세요.

파이프라인이 성공적으로 생성되면 클라이언트를 구성하고 OpenSearch Service 도메인으로 데이터 수집을 시작할 수 있습니다. 자세한 내용은 the section called “파이프라인 통합” 단원을 참조하십시오.

파이프라인 생성 상태 추적

OpenSearch Ingestion이 파이프라인을 프로비저닝하고 데이터 수집을 준비할 때 파이프라인의 상태를 추적할 수 있습니다.

콘솔

파이프라인을 처음 생성한 후에는 OpenSearch Ingestion에서 데이터 수집을 준비하면서 여러 단계를 거칩니다. 파이프라인 생성의 다양한 단계를 보려면 파이프라인 이름을 선택하여 해당 파이프라인 설정 페이지를 확인하세요. 상태에서 세부 정보 보기를 선택합니다.

파이프라인은 다음 단계를 거친 후 데이터를 수집할 수 있게 됩니다.

- 검증 — 파이프라인 구성을 검증합니다. 이 단계가 완료되면 모든 검증이 성공한 것입니다.
- 환경 조성 — 리소스를 준비 및 프로비저닝합니다. 이 단계가 완료되면 새 파이프라인 환경이 만들어진 것입니다.
- 파이프라인 배포 - 파이프라인을 배포합니다. 이 단계가 완료되면 파이프라인이 성공적으로 배포된 것입니다.

- 파이프라인 상태 확인 - 파이프라인 상태를 확인합니다. 이 단계가 완료되면 모든 상태 확인이 통과된 것입니다.
- 트래픽 활성화 - 파이프라인이 데이터를 수집할 수 있도록 합니다. 이 단계가 완료되면 파이프라인으로 데이터 수집을 시작할 수 있습니다.

CLI

파이프라인 상태를 확인하려면 get-pipeline-change-progress 명령을 사용하세요. 다음 AWS CLI 요청은 라는 파이프라인의 상태를 확인합니다`my-pipeline`.

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

응답:

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

OpenSearch Ingestion API

OpenSearch Ingestion API를 사용하여 파이프라인 생성 상태를 추적하려면 GetPipelineChangeProgress 작업을 호출하세요.

블루프린트 작업

파이프라인 정의를 처음부터 생성하는 대신 Trace Analytics 또는 Apache 로그와 같은 일반적인 수집 시나리오를 위해 미리 구성된 템플릿인 구성 블루프린트를 사용할 수 있습니다. 구성 청사진을 사용하면 구성을 처음부터 작성하지 않고도 파이프라인을 쉽게 프로비저닝할 수 있습니다.

콘솔

파이프라인 청사진 사용

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 파이프라인을 선택한 후 파이프라인 생성을 선택합니다.
3. 사용 사례 목록에서 블루프린트를 선택한 다음, 블루프린트 선택을 선택합니다. 파이프라인 구성은 선택한 사용 사례의 하위 파이프라인으로 채워집니다.

   파이프라인 청사진은 현재 상태로는 유효하지 않습니다. 선택한 소스에 따라 추가 설정을 지정해야 합니다.

CLI

를 사용하여 사용 가능한 모든 블루프린트 목록을 가져오려면 list-pipeline-blueprints 요청을 AWS CLI 보냅니다.

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

이 요청은 사용 가능한 모든 청사진의 목록을 반환합니다.

특정 청사진에 대한 자세한 정보를 얻으려면 get-pipeline-blueprint 명령어를 사용하세요.

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

이 요청은 Apache 로그 파이프라인 청사진의 콘텐츠를 반환합니다.

```
{
   "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 Ingestion API

OpenSearch Ingestion API를 사용하여 파이프라인 청사진에 대한 정보를 가져오려면 ListPipelineBlueprints 및 GetPipelineBlueprint 작업을 사용하세요.

Amazon OpenSearch Ingestion 파이프라인 보기

AWS Management Console, AWS CLI또는 OpenSearch Ingestion API를 사용하여 Amazon OpenSearch Ingestion 파이프라인에 대한 세부 정보를 확인할 수 있습니다.

콘솔

파이프라인을 보려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 파이프라인을 선택합니다.
3. (선택 사항) 특정 상태의 파이프라인을 보려면 모든 상태를 선택하고 필터링 기준으로 사용할 상태를 선택합니다.

   파이프라인은 다음과 같은 상태일 수 있습니다.

   - `Active`— 파이프라인이 활성 상태이며 데이터를 수집할 준비가 되었습니다.
   - `Creating`— 파이프라인이 생성되고 있습니다.
   - `Updating`— 파이프라인이 업데이트되고 있습니다.

- `Deleting`— 파이프라인이 삭제되고 있습니다.
- `Create failed`— 파이프라인을 생성할 수 없습니다.
- `Update failed` - 파이프라인을 업데이트할 수 없습니다.
- `Stop failed` - 파이프라인을 중지할 수 없습니다.
- `Start failed` - 파이프라인을 시작할 수 없습니다.
- `Stopping`— 파이프라인이 중지되고 있습니다.
- `Stopped`— 파이프라인이 중지되었으며 언제든지 다시 시작할 수 있습니다.
- `Starting`— 파이프라인이 시작되고 있습니다.

파이프라인이 `Create failed`, `Creating`, `Deleting`, 및 `Stopped` 상태일 때는 Ingestion OCU에 대한 요금이 청구되지 않습니다.

CLI

를 사용하여 파이프라인을 보려면 list-pipelines 요청을 AWS CLI보냅니다.

```
aws osis list-pipelines
```

요청은 기존의 모든 파이프라인 목록을 반환합니다.

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

단일 파이프라인에 대한 정보를 가져오려면 get-pipeline 명령을 사용하세요.

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

요청은 지정된 파이프라인의 구성 정보를 반환합니다.

```
{
    "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 Ingestion API

OpenSearch Ingestion API를 사용하여 OpenSearch Ingestion 파이프라인을 보려면 ListPipelines 및 GetPipeline 작업을 호출하세요.

Amazon OpenSearch Ingestion 파이프라인 업데이트

AWS Management Console AWS CLI, 또는 OpenSearch Ingestion API를 사용하여 Amazon OpenSearch Ingestion 파이프라인을 업데이트할 수 있습니다. OpenSearch Ingestion은 파이프라인 구성을 업데이트할 때 블루/그린 배포를 시작합니다. 자세한 내용은 the section called “파이프라인 업데이트를 위한 블루/그린 배포” 단원을 참조하십시오.

주제

- 고려 사항
- 필요한 권한
- 파이프라인 업데이트
- 파이프라인 업데이트를 위한 블루/그린 배포

고려 사항

파이프라인을 업데이트할 때 다음 사항을 고려하세요.

- 파이프라인의 이름 또는 네트워크 설정은 업데이트할 수 없습니다.
- 파이프라인이 VPC 도메인 싱크에 쓰는 경우, 파이프라인이 생성된 후에는 되돌아가서 다른 VPC 도메인으로 싱크를 변경할 수 없습니다. 파이프라인을 삭제하고 새 싱크로 재생성해야 합니다. VPC 도메인에서 퍼블릭 도메인으로, 퍼블릭 도메인에서 VPC 도메인으로 또는 퍼블릭 도메인에서 다른 퍼블릭 도메인으로 싱크를 전환할 수 있습니다.
- 퍼블릭 OpenSearch Service 도메인과 OpenSearch Serverless 컬렉션 간에 언제든지 파이프라인 싱크를 전환할 수 있습니다.
- 파이프라인의 소스, 프로세서 또는 싱크 구성을 업데이트하면 OpenSearch Ingestion이 블루/그린 배포를 시작합니다. 자세한 내용은 the section called “파이프라인 업데이트를 위한 블루/그린 배포” 단원을 참조하십시오.
- 파이프라인의 소스, 프로세서 또는 싱크 구성을 업데이트하면 OpenSearch Ingestion은 파이프라인을 파이프라인이 실행 중인 Data Prepper 메이저 버전의 지원되는 최신 마이너 버전으로 자동으로 업그레이드합니다. 이 프로세스를 통해 최신 버그를 수정하고 성능을 개선하여 파이프라인을 최신 상태로 유지할 수 있습니다.

- 파이프라인이 중지된 후에도 여전히 파이프라인을 업데이트할 수 있습니다.

필요한 권한

OpenSearch Ingestion은 다음 IAM 권한을 사용하여 파이프라인을 업데이트합니다.

- `osis:UpdatePipeline` - 파이프라인을 업데이트합니다.
- `osis:ValidatePipeline`— 파이프라인 구성이 유효한지 확인하세요.
- `iam:PassRole` - OpenSearch Ingestion에 파이프라인 역할을 전달하여 도메인에 데이터를 쓸 수 있도록 합니다. 이 권한은 파이프라인 구성을 업데이트하는 경우에만 필요하며 로그 게시 또는 용량 제한과 같은 다른 설정을 수정하는 경우에는 필요하지 않습니다.

예를 들어 다음 정책에서 파이프라인을 업데이트할 권한을 부여합니다.

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"
            ]
        }
    ]
}
```

파이프라인 업데이트

AWS Management Console AWS CLI, 또는 OpenSearch Ingestion API를 사용하여 Amazon OpenSearch Ingestion 파이프라인을 업데이트할 수 있습니다.

콘솔

파이프라인을 업데이트하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 파이프라인을 선택합니다.
3. 파이프라인을 선택하여 해당 설정을 엽니다. 그런 다음 편집 옵션 중 하나를 선택합니다.
4. 변경 작업을 마치면 저장을 선택합니다.

CLI

를 사용하여 파이프라인을 업데이트하려면 업데이트 파이프라인 요청을 AWS CLI보냅니다. 다음 샘플 요청은 새 구성 파일을 업로드하고 최소 및 최대 용량 값을 업데이트합니다.

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

OpenSearch Ingestion API

OpenSearch Ingestion API를 사용하여 OpenSearch Ingestion 파이프라인을 업데이트하려면 UpdatePipeline 작업을 호출하세요.

파이프라인 업데이트를 위한 블루/그린 배포

OpenSearch Ingestion은 파이프라인 구성을 업데이트할 때 블루/그린 배포 프로세스를 시작합니다.

블루/그린은 파이프라인 업데이트용으로 새 환경을 만들고 업데이트가 완료되면 트래픽을 새 환경으로 라우팅하는 관행을 지칭합니다. 이렇게 하면 가동 중지가 최소화되고, 새로운 환경에 배포하는 데 실패하더라도 원래의 환경이 유지됩니다. 블루/그린 배포 자체는 성능에 영향을 주지 않지만, 파이프라인 구성이 성능을 변경하는 방식으로 변경되면 성능이 변경될 수 있습니다.

OpenSearch Ingestion은 블루/그린 배포 중에 Auto Scaling을 차단합니다. 새 파이프라인으로 리디렉션되기 전까지는 이전 파이프라인으로 향하는 트래픽에 대해서만 계속 요금이 부과됩니다. 트래픽이 리디렉션되면 새 파이프라인에 대한 비용만 청구됩니다. 두 파이프라인에 대해 동시에 요금이 청구되는 일은 없습니다.

파이프라인의 소스, 프로세서 또는 싱크 구성을 업데이트하면 OpenSearch Ingestion은 파이프라인을 파이프라인이 실행 중인 메이저 버전의 지원되는 최신 마이너 버전으로 자동으로 업그레이드할 수 있습니다. 예를 들어, 파이프라인 구성에 `version: "2"`이 있고 OpenSearch Ingestion이 처음에 파이프라인을 버전 2.1.0으로 프로비저닝했을 수 있습니다. 버전 2.1.1에 대한 지원이 추가되고 파이프라인 구성을 변경하면 OpenSearch Ingestion은 파이프라인을 버전 2.1.1로 업그레이드합니다.

이 프로세스를 통해 최신 버그를 수정하고 성능을 개선하여 파이프라인을 최신 상태로 유지할 수 있습니다. OpenSearch Ingestion은 파이프라인 구성 내에서 `version` 옵션을 수동으로 변경하지 않는 한 파이프라인의 메이저 버전을 업데이트할 수 없습니다.

Amazon OpenSearch Ingestion 파이프라인 비용 관리

Amazon OpenSearch Ingestion에서 수집 파이프라인을 시작하고 중지하여 필요에 따라 데이터 흐름을 제어할 수 있습니다. 파이프라인을 중지하면 구성을 유지하면서 데이터 처리가 중단되므로 다시 구성하지 않고도 다시 시작할 수 있습니다. 이를 통해 비용을 최적화하거나, 리소스 사용량을 관리하거나, 문제를 해결할 수 있습니다. 파이프라인을 중지하면 OpenSearch Ingestion은 수신 데이터를 처리하지 않지만 이전에 수집된 데이터는 OpenSearch에서 계속 사용할 수 있습니다.

시작 및 중지는 지속적인 가용성이 필요하지 않은 개발, 테스트 또는 유사한 활동에 사용하는 파이프라인의 설정 및 해체 프로세스를 간소화합니다. 파이프라인이 중지된 동안에는 Insting OCU 시간에 대해 요금이 부과되지 않습니다. 중지된 파이프라인은 계속 업데이트할 수 있으며, 자동 마이너 버전 업데이트와 보안 패치를 받게 됩니다. 파이프라인을 다시 시작하면 새 수신 데이터에서 처리가 재개됩니다.

> **Note**
>
> 파이프라인의 용량이 초과되었지만 계속 작동해야 하는 경우 최대 용량 제한을 중지했다가 다시 시작하는 대신 조정하는 것이 좋습니다. 이를 통해 비용을 관리하는 동시에 파이프라인이 데이터를 효율적으로 계속 처리할 수 있습니다. 자세한 내용은 the section called “파이프라인 크기 조정”을 참조하세요.

다음 주제에서는 AWS Management Console AWS CLI및 OpenSearch Ingestion API를 사용하여 파이프라인을 시작하고 중지하는 방법을 설명합니다.

주제

- Amazon OpenSearch Ingestion 파이프라인 중지
- Amazon OpenSearch Ingestion 파이프라인 시작

Amazon OpenSearch Ingestion 파이프라인 중지

OpenSearch Ingestion 파이프라인을 사용하거나 관리를 수행하려면 항상 활성 파이프라인으로 시작한 다음 파이프라인을 중지하고 파이프라인을 다시 시작해야 합니다. 파이프라인이 중지된 동안에는 Ingestion OCU 시간에 대해 요금이 부과되지 않습니다.

콘솔

파이프라인 중지

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 탐색 창에서 파이프라인을 선택한 후 파이프라인을 선택합니다. 이 페이지에서 중지 작업을 수행하거나 중지하려는 파이프라인의 세부 정보 페이지로 이동하세요.
3. 작업에서 파이프라인 중지를 선택합니다.

   파이프라인을 중지하거나 시작할 수 없는 경우 파이프라인 중지 작업을 사용할 수 없습니다.

AWS CLI

를 사용하여 파이프라인을 중지하려면 다음 파라미터와 함께 stop-pipeline 명령을 AWS CLI호출합니다.

- `--pipeline-name` - 파이프라인의 이름.

Example

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

OpenSearch Ingestion API

OpenSearch Ingestion API를 사용하여 파이프라인을 중지하려면 다음 파라미터와 함께 StopPipeline 작업을 호출하세요.

- `PipelineName` - 파이프라인의 이름.

Amazon OpenSearch Ingestion 파이프라인 시작

이미 중지 상태인 파이프라인으로 시작하는 OpenSearch Ingestion 파이프라인을 항상 시작합니다. 파이프라인은 용량 제한, 네트워크 설정 및 로그 게시 옵션과 같은 구성 설정을 유지합니다.

파이프라인을 다시 시작하려면 일반적으로 몇 분 정도 걸립니다.

콘솔

파이프라인 시작

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 탐색 창에서 파이프라인을 선택한 후 파이프라인을 선택합니다. 이 페이지에서 시작 작업을 수행하거나 시작하려는 파이프라인의 세부 정보 페이지로 이동하세요.
3. 작업에는 파이프라인 시작을 선택합니다.

AWS CLI

를 사용하여 파이프라인을 시작하려면 다음 파라미터와 함께 start-pipeline 명령을 AWS CLI호출합니다.

- `--pipeline-name` - 파이프라인의 이름.

Example

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

OpenSearch Ingestion API

OpenSearch Ingestion API를 사용하여 OpenSearch Ingestion 파이프라인을 시작하려면 다음 파라미터와 함께 StartPipeline 작업을 호출하세요.

- `PipelineName` - 파이프라인의 이름.

Amazon OpenSearch Ingestion 파이프라인 삭제

AWS Management Console AWS CLI, 또는 OpenSearch Ingestion API를 사용하여 Amazon OpenSearch Ingestion 파이프라인을 삭제할 수 있습니다. `Creating` 또는 `Updating` 상태인 경우 파이프라인을 삭제할 수 없습니다.

콘솔

파이프라인 삭제

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 파이프라인을 선택합니다.
3. 삭제할 파이프라인을 선택하고 작업, 삭제를 선택합니다.
4. 삭제를 확인하고 삭제(Delete)를 선택합니다.

CLI

를 사용하여 파이프라인을 삭제하려면 다음과 같이 파이프라인 삭제 요청을 AWS CLI보냅니다.

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

OpenSearch Ingestion API

OpenSearch Ingestion API를 사용하여 OpenSearch Ingestion 파이프라인을 시작하려면 다음 파라미터와 함께 DeletePipeline 작업을 호출하세요.

- `PipelineName` - 파이프라인의 이름.

Amazon OpenSearch Ingestion 파이프라인에 지원되는 플러그인 및 옵션

Amazon OpenSearch Ingestion은 오픈 소스 OpenSearch Data Prepper 내에서 소스, 프로세서 및 싱크의 하위 집합을 지원합니다. 또한 OpenSearch Ingestion은 지원되는 각 플러그인의 사용 가능한 옵션에 적용하는 몇 가지 제약이 있습니다. 다음 섹션에서는 OpenSearch Ingestion 기능이 지원하는 플러그인 및 관련 옵션에 대해 설명합니다.

> **Note**
>
> OpenSearch Ingestion은 기본 버퍼를 자동으로 구성하므로 어떤 버퍼 플러그인도 지원하지 않습니다. 파이프라인 구성에 버퍼를 포함하면 유효성 검사 오류가 발생합니다.

주제

- 지원되는 플러그인
- 상태 비저장 프로세서와 상태 저장 프로세서 비교
- 구성 요구 사항 및 제약 조건

지원되는 플러그인

OpenSearch Ingestion은 다음과 같은 Data Prepper 플러그인을 지원합니다.

소스:

- documentdb
- DynamoDB

- HTTP
- Kafka
- Kinesis
- OpenSearch
- OTel 로그
- OTel 지표
- OTel 추적
- S3

Processors:

- 항목 추가
- 집계
- 이상 탐지기
- AWS Lambda
- 항목 유형 변환
- 값 복사
- CSV
- 날짜

- 지연
- 압축 해제
- 항목 삭제
- 해체
- 이벤트 삭제
- 평면화
- 지리적 IP
- Grok
- 키 값
- 매핑할 목록
- 소문자 문자열
- 목록에 매핑
- 이벤트 변형(프로세서 시리즈)
- 문자열 변형(프로세서 시리즈)
- 난독화
- OTel 지표
- OTel 추적 그룹
- OTel 추적
- Ion 구문 분석
- JSON 구문 분석
- XML 구문 분석
- 키 이름 바꾸기
- 경로
- 항목 선택
- 서비스 맵
- 이벤트 분할
- 문자열 분할
- 문자열 변환기
- 대체 문자열

- 추적 피어 전달자
- [Translate]
- 문자열 잘라내기
- 자르기
- 대문자 문자열
- 사용자 에이전트
- JSON 쓰기

싱크:

- OpenSearch(OpenSearch Service, OpenSearch Serverless, Elasticsearch 6.8 이상 지원)
- S3

싱크 코덱:

- Avro
- NDJSON
- JSON
- PARQUET

상태 비저장 프로세서와 상태 저장 프로세서 비교

상태 비저장 프로세서는 변환 및 필터링과 같은 작업을 수행하는 반면, 상태 저장 프로세서는 이전 실행 결과를 기억하는 집계와 같은 작업을 수행합니다. OpenSearch Ingestion은 상태 저장 프로세서 집계 및 서비스-맵을 지원합니다. 지원되는 다른 모든 프로세서는 상태 비저장 프로세서입니다.

상태 비저장 프로세서만 포함하는 파이프라인의 경우 최대 용량 제한은 96개의 Ingestion OCU입니다. 파이프라인이 상태 비저장 프로세서를 포함하는 경우 최대 용량 제한은 48개의 Ingestion OCU입니다. 그러나 파이프라인에 영구 버퍼링이 활성화된 경우 상태 비저장 프로세서만 있는 최대 384개의 Ingestion OCU 또는 상태 저장 프로세서를 포함하는 경우 192개의 Ingestion OCU를 보유할 수 있습니다. 자세한 내용은 the section called “파이프라인 크기 조정” 단원을 참조하십시오.

엔드 투 엔드 승인은 상태 비저장 프로세서에서만 지원됩니다. 자세한 내용은 the section called “엔드 투 엔드 승인” 단원을 참조하십시오.

구성 요구 사항 및 제약 조건

아래에 달리 명시되지 않는 한, 위에 나열된 지원 플러그인의 Data Prepper 구성 참조에 설명된 모든 옵션은 OpenSearch Ingestion 파이프라인에서 허용됩니다. 다음 섹션에서는 OpenSearch Ingestion이 특정 플러그인 옵션에 적용하는 제약 조건에 대해 설명합니다.

> **Note**
>
> OpenSearch Ingestion은 기본 버퍼를 자동으로 구성하므로 어떤 버퍼 플러그인도 지원하지 않습니다. 파이프라인 구성에 버퍼를 포함하면 유효성 검사 오류가 발생합니다.

OpenSearch Ingestion은 다양한 옵션을 내부적으로 구성하고 관리합니다 (예: `authentication` 및 `acm_certificate_arn`). `thread_count` 및 `request_timeout`와 같은 다른 옵션은 수동으로 변경할 경우 성능에 영향을 미칩니다. 따라서 파이프라인의 성능을 최적화하기 위해 이러한 값이 내부적으로 설정됩니다.

마지막으로 `ism_policy_file` 및 `sink_template`와 같은 일부 옵션은 OpenSearch Ingestion으로 전달할 수 없습니다. 오픈 소스 Data Prepper에서 실행할 경우 로컬 파일이기 때문입니다. 이 값은 지원되지 않습니다.

주제

- 일반 파이프라인 옵션
- Grok 프로세서
- HTTP 소스
- OpenSearch sink
- OTel 지표 소스, OTel 추적 소스 및 OTel 로그 소스
- OTel 추적 그룹 프로세서
- OTel 추적 프로세서
- 서비스 맵 프로세서
- S3 소스

일반 파이프라인 옵션

다음 일반 파이프라인 옵션은 OpenSearch Ingestion에서 설정되며 파이프라인 구성에서는 지원되지 않습니다.

- workers
- delay

Grok 프로세서

다음 과 같은 공급자 옵션이 지원됩니다.

- patterns_directories
- patterns_files_glob

HTTP 소스

HTTP 소스 플러그인에는 다음과 같은 요구 사항 및 제약이 있습니다.

- 옵션은 path 필수입니다. 경로는 수집을 위한 URI 경로를 나타내는 /log/ingest와 같은 문자열입니다. 이 경로는 파이프라인으로 데이터를 전송하는 데 사용하는 URI를 정의합니다. 예를 들어 https://log-pipeline.us-west-2.osis.amazonaws.com*/log/ingest*입니다. 경로는 슬래시(/)로 시작해야 하며 특수 문자 '-', '_', '.', '/'를 비롯해 ${pipelineName} 자리 표시자를 포함할 수 있습니다.
- 다음 HTTP 소스 옵션은 OpenSearch Ingestion에서 설정되며 파이프라인 구성에서는 지원되지 않습니다.
  - 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 sink

OpenSearch 싱크 플러그인에는 다음과 같은 요구 사항 및 제한 사항이 있습니다.

- aws 옵션은 필수이며 다음 옵션을 포함해야 합니다.
  - sts_role_arn
  - region
  - hosts
  - serverless(싱크가 OpenSearch Serverless 컬렉션인 경우)
- sts_role_arn 옵션은 YAML 정의 파일 내 각 싱크에 대해 동일한 역할을 가리켜야 합니다.
- hosts 옵션은 OpenSearch Service 도메인 엔드포인트 또는 OpenSearch Serverless 컬렉션 엔드포인트를 지정해야 합니다. 도메인의 사용자 지정 엔드포인트는 지정할 수 없으며 표준 엔드포인트여야 합니다.
- hosts 옵션이 서버리스 컬렉션 엔드포인트인 경우 serverless 옵션을 true로 설정해야 합니다. 또한 YAML 정의 파일에 index_type 옵션이 포함된 경우 management_disabled로 설정해야 합니다. 그렇지 않으면 검증이 실패합니다.
- 다음 옵션은 JSON에서 지원되지 않습니다.
  - username
  - password
  - cert
  - proxy
  - dlq_file - 실패한 이벤트를 DLQ(Dead Letter Queue)로 오프로드하려면 dlq 옵션을 사용하고 S3 버킷을 지정해야 합니다.
  - ism_policy_file
  - socket_timeout
  - template_file
  - insecure

OTel 지표 소스, OTel 추적 소스 및 OTel 로그 소스

OTel 지표 소스, OTel 추적 소스 및 OTel 로그 소스 플러그인에는 다음과 같은 요구 사항 및 제한 사항이 있습니다.

- 옵션은 path 필수입니다. 경로는 수집을 위한 URI 경로를 나타내는 /log/ingest와 같은 문자열입니다. 이 경로는 파이프라인으로 데이터를 전송하는 데 사용하는 URI를 정의합니다. 예를 들어 https://log-pipeline.us-west-2.osis.amazonaws.com*/log/ingest*입니다. 경로는 슬래시(/)로 시작해야 하며 특수 문자 '-', '_', '.', '/'를 비롯해 ${pipelineName} 자리 표시자를 포함할 수 있습니다.
- 다음 옵션은 OpenSearch Ingestion에서 설정되며 파이프라인 구성에서는 지원되지 않습니다.
  - 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 추적 그룹 프로세서

OTel 추적 그룹 프로세서에는 다음과 같은 요구 사항 및 제한 사항이 있습니다.

- aws 옵션은 필수이며 다음 옵션을 포함해야 합니다.

  - sts_role_arn
  - region
  - hosts
- sts_role_arn 옵션은 OpenSearch 싱크 구성에서 지정하는 파이프라인 역할과 동일한 역할을 지정합니다.
- username, password, cert, insecure 옵션은 지원되지 않습니다.
- aws_sigv4 옵션은 필수이며 true로 설정되어야 합니다.
- OpenSearch 싱크 플러그인 내의 serverless 옵션은 지원되지 않습니다. OTel 추적 그룹 프로세서는 현재 OpenSearch Serverless 컬렉션과 함께 작동하지 않습니다.
- 파이프라인 구성 본문 내의 otel_trace_group 프로세서 수는 8개를 초과할 수 없습니다.

OTel 추적 프로세서

OTel 추적 프로세서에는 다음과 같은 요구 사항 및 제한 사항이 있습니다.

- trace_flush_interval 옵션 값은 300초를 초과할 수 없습니다.

서비스 맵 프로세서

서비스-맵 프로세서에는 다음과 같은 요구 사항 및 제한 사항이 있습니다.

- window_duration 옵션 값은 300초를 초과할 수 없습니다.

S3 소스

S3 소스 플러그인에는 다음과 같은 요구 사항 및 제한 사항이 있습니다.

- aws 옵션은 필수이며 region 및 sts_role_arn 옵션을 포함해야 합니다.
- records_to_accumulate 옵션 값은 200초를 초과할 수 없습니다.
- maximum_messages 옵션 값은 10초를 초과할 수 없습니다.
- 지정된 경우 disable_bucket_ownership_validation 옵션은 false로 설정되어야 합니다.
- 지정된 경우 input_serialization 옵션은 parquet(으)로 설정되어야 합니다.

Amazon OpenSearch Ingestion 파이프라인을 다른 서비스 및 애플리케이션과 통합

Amazon OpenSearch Ingestion 파이프라인으로 데이터를 성공적으로 수집하려면 파이프라인 엔드포인트로 데이터를 전송하도록 클라이언트 애플리케이션(소스)을 구성해야 합니다. 소스는 Fluent Bit 로그, OpenTelemetry Collector 또는 간단한 S3 버킷과 같은 클라이언트일 수 있습니다. 정확한 구성은 각 클라이언트마다 다릅니다.

소스 구성 중 중요한 차이점은 (OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션으로 직접 데이터를 전송하는 것과 비교하여) AWS 서비스 이름(`osis`)과 호스트 엔드포인트이며, 이는 파이프라인 엔드포인트여야 합니다.

수집 엔드포인트 구성

파이프라인으로 데이터를 수집하려면 데이터를 수집 엔드포인트로 전송합니다. 수집 URL을 찾으려면 파이프라인 설정 페이지로 이동하여 수집 URL을 복사합니다.

OTel 추적 및 OTel 메트릭과 같은 풀 기반 소스에 대한 전체 수집 엔드포인트를 구성하려면 파이프라인 구성의 수집 경로를 수집 URL에 추가하세요.

예를 들어 파이프라인 구성의 수집 경로가 다음과 같다고 가정해 보겠습니다.

클라이언트 구성에서 지정하는 전체 수집 엔드포인트는 `https://ingestion-pipeline-abcdefg.us-east-1.osis.amazonaws.com/my/test_path`와 같은 형식을 취합니다.

수집 역할 생성

OpenSearch Ingestion에 대한 모든 요청은 서명 버전 4로 서명되어야 합니다. 요청에 서명하는 역할에는 최소한 `osis:Ingest` 작업에 대한 권한이 부여되어야 하며, 해당 역할은 OpenSearch Ingestion 파이프라인으로 데이터를 보낼 수 있습니다.

예를 들어, 다음 AWS Identity and Access Management (IAM) 정책은 해당 역할이 단일 파이프라인으로 데이터를 전송하도록 허용합니다.

JSON

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

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

> **Note**
>
> 모든 파이프라인에 이 역할을 사용하려면 `Resource` 요소의 ARN을 와일드카드(*)로 바꾸세요.

교차 계정 수집 액세스 제공

> **Note**
>
> VPC 파이프라인이 아닌 퍼블릭 파이프라인에 대한 교차 계정 수집 액세스만 제공할 수 있습니다.

소스 애플리케이션을 포함하는 계정 AWS 계정과 같은 다른에서 파이프라인으로 데이터를 수집해야 할 수 있습니다. 파이프라인에 쓰는 보안 주체가 파이프라인 자체와 다른 계정에 있는 경우, 파이프라인으로 데이터를 수집하는 다른 IAM 역할을 신뢰할 수 있도록 보안 주체를 구성해야 합니다.

교차 계정 수집 권한 구성

1. 파이프라인과 동일한 내에 `osis:Ingest` 권한(이전 섹션에서 설명)을 AWS 계정 사용하여 수집 역할을 생성합니다. 자세한 내용은 IAM 역할 생성을 참조하세요.
2. 다른 계정의 보안 주체가 이를 수임할 수 있도록 수집 역할에 신뢰 정책을 연결하세요.

   JSON

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

```
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
       },
      "Action": "sts:AssumeRole"
  }]
}
```

3. 다른 계정에서는 수집 역할을 맡도록 클라이언트 애플리케이션(예: Fluent Bit)을 구성하세요. 이 기능을 사용하려면 애플리케이션 계정이 애플리케이션 사용자 또는 역할에 수집 역할을 맡을 수 있는 권한을 부여해야 합니다.

   다음 예제 ID 기반 정책은 연결된 보안 주체가 파이프라인 계정에서 `ingestion-role`을 수임하도록 허용합니다.

   JSON

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

그러면 클라이언트 애플리케이션이 AssumeRole 작업을 사용하여 `ingestion-role`을 수임하고 관련 파이프라인으로 데이터를 수집할 수 있습니다.

Atlassian Services에서 OpenSearch Ingestion 파이프라인 사용

Atlassian Jira 및 Confluence 소스 플러그인을 사용하여 Atlassian 서비스에서 OpenSearch Ingestion 파이프라인으로 데이터를 수집할 수 있습니다. 이러한 통합을 통해 전체 Jira 프로젝트와 Confluence 공간을 동기화하는 동시에 업데이트의 지속적인 모니터링 및 자동 동기화를 통해 실시간 관련성을 유지하여 검색 가능한 통합 지식 기반을 생성할 수 있습니다.

Integrating with Jira

Jira 콘텐츠를 OpenSearch에 통합하여 강력한 컨텍스트 검색 기능으로 Jira 경험을 혁신합니다. Data Prepper Atlassian Jira 소스 플러그인을 사용하면 업데이트의 지속적인 모니터링 및 자동 동기화를 통해 실시간 관련성을 유지하면서 전체 Jira 프로젝트를 동기화하여 검색 가능한 통합 지식 기반을 생성할 수 있습니다. 이 통합을 통해 특정 프로젝트, 문제 유형 및 상태에 대한 유연한 필터링 옵션을 통해 데이터를 동기화할 수 있으므로 필요한 정보만 가져올 수 있습니다.

안전하고 안정적인 연결을 보장하기 위해 플러그인은 기본 API 키 인증 및 OAuth2 인증을 비롯한 여러 인증 방법을 지원하며,에 저장된 보안 암호를 사용하여 자격 증명을 관리하는 보안 기능이 추가되었습니다 AWS Secrets Manager. 또한 중단 없는 액세스를 위한 자동 토큰 갱신 기능을 제공하여 지속적인 작업을 보장합니다. Atlassian의 API 버전 2를 기반으로 구축된이 통합을 통해 팀은 OpenSearch의 고급 검색 기능을 통해 Jira 데이터에서 중요한 인사이트를 얻을 수 있습니다.

Integrating with Confluence

Data Prepper의 Confluence 소스 플러그인을 통해 Atlassian Confluence 콘텐츠를 OpenSearch에 통합하여 팀의 지식 관리 및 협업 기능을 개선합니다. 이 통합을 통해 검색 가능한 중앙 집중식 집합 지식 리포지토리를 생성하여 정보 검색 및 팀 생산성을 개선할 수 있습니다. 플러그인은 Confluence 콘텐츠를 동기화하고 업데이트를 지속적으로 모니터링하여 OpenSearch 인덱스를 up-to-date 상태로 포괄적으로 유지합니다.

통합은 유연한 필터링 옵션을 제공하므로 특정 스페이스 또는 페이지 유형에서 콘텐츠를 선택적으로 가져와 조직의 필요에 맞게 동기화된 콘텐츠를 조정할 수 있습니다. 플러그인은 기본 API 키와 OAuth2 인증 방법을 모두 지원하며를 통해 자격 증명을 안전하게 관리할 수 있습니다 AWS Secrets Manager. 플러그인의 자동 토큰 갱신 기능은 중단 없는 액세스와 원활한 작동을 보장합니다. Atlassian의 Confluence API를 기반으로 구축된이 통합을 통해 팀은 Confluence 콘텐츠에서 OpenSearch의 고급 검색 기능을 활용하여 조직 내 정보 접근성과 사용률을 높일 수 있습니다.

주제

- 사전 조건
- 파이프라인 역할 구성
- Jira 커넥터 파이프라인 구성
- Confluence 커넥터 파이프라인 구성
- 데이터 일관성
- 제한 사항
- Atlassian 커넥터용 CloudWatch의 지표

- OAuth 2.0을 사용하여 Amazon OpenSearch Ingestion 파이프라인을 Atlassian Jira 또는 Confluence에 연결

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 완료합니다.

1. 다음 옵션 중 하나를 선택하여 Jira 사이트의 자격 증명을 준비합니다. OpenSearch Ingestion은 콘텐츠에 대한 `ReadOnly` 권한 부여만 필요합니다.

   a. 옵션 1: API 키 - Atlassian 계정에 로그인하고 다음 주제의 정보를 사용하여 API 키를 생성합니다.

   - Atlassian 계정의 API 토큰 관리

   b. 옵션 2: OAuth2 - Atlassian 계정에 로그인하고의 정보를 사용합니다the section called “OAuth 2.0을 사용하여 Amazon OpenSearch Ingestion 파이프라인을 Atlassian Jira 또는 Confluence에 연결”.

2. 에서 보안 암호를 생성 AWS Secrets Manager하여 이전 단계에서 생성한 자격 증명을 저장합니다. 절차에 따라 다음을 선택합니다.

   - 보안 암호 유형에서 다른 유형의 보안 암호를 선택합니다.
   - 키/값 페어의 경우 선택한 권한 부여 유형에 따라 다음 페어를 생성합니다.

   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
   }
   ```

   보안 암호를 생성한 후 보안 암호의 Amazon 리소스 이름(ARN)을 복사합니다. 파이프라인 역할 권한 정책에 포함합니다.

파이프라인 역할 구성

파이프라인에 전달된 역할에는 사전 조건 섹션에서 생성된 보안 암호를 읽고 쓰려면 다음 정책이 연결되어 있어야 합니다.

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::secret:secret-
name-random-6-characters"
        }
    ]
}
```

역할에는 선택한 싱크에 액세스하고 쓸 수 있는 정책도 연결되어 있어야 합니다. 예를 들어 싱크로 OpenSearch를 선택하면 정책은 다음과 비슷합니다.

JSON

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

```
        }
    ]
}
```

Jira 커넥터 파이프라인 구성

미리 구성된 Atlassian Jira 블루프린트를 사용하여이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

*## ###*를 자신의 정보로 바꿉니다.

```
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 소스의 속성에 대한 키:

1. 호스트: Jira 클라우드 또는 온프레미스 URL. 일반적으로 처럼 보입니다`https://your-domain-name.atlassian.net/`.
2. 승인: 싱크로 데이터가 완전히 전송되도록 합니다.
3. 인증: 파이프라인이 Jira 인스턴스에 액세스하는 방법을 설명합니다. `Basic` 또는 `OAuth2`를 선택하고 AWS 보안 암호의 키를 참조하는 해당 키 속성을 지정합니다.
4. 필터:이 섹션에서는 추출하고 동기화할 Jira 데이터 부분을 선택하는 데 도움이 됩니다.
   a. project: `include` 섹션에서 동기화하려는 프로젝트 키를 나열합니다. 그렇지 않으면 `exclude` 섹션에서 제외하려는 프로젝트를 나열합니다. 지정된 시간에 포함 또는 제외 옵션 중 하나만 제공합니다.

b. issue_type: 동기화하려는 특정 문제 유형입니다. 필요에 맞는 유사한 `include` 또는 `exclude` 패턴을 따릅니다. 첨부 파일은 원래 첨부 파일에 대한 앵커 링크로 표시되지만 첨부 파일 콘텐츠는 추출되지 않습니다.

c. status: 데이터 추출 쿼리에 적용할 특정 상태 필터입니다. 를 지정하면 해당 상태의 티켓`include`만 동기화됩니다. 를 지정하면 제외된 상태가 나열된 티켓을 제외한 `exclude`모든 티켓이 동기화됩니다.

Confluence 커넥터 파이프라인 구성

미리 구성된 Atlassian Confluence 블루프린트를 사용하여이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

```
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 소스의 주요 속성:

1. 호스트: Confluence 클라우드 또는 온프레미스 URL. 일반적으로 다음과 같습니다. `https://your-domain-name.atlassian.net/`
2. 승인: 싱크로 데이터가 완전히 전송되도록 합니다.
3. 인증: 파이프라인이 Confluence 인스턴스에 액세스하는 방법을 설명합니다. `Basic` 또는 `OAuth2`를 선택하고 AWS 보안 암호의 키를 참조하는 해당 키 속성을 지정합니다.
4. 필터:이 섹션에서는 추출하고 동기화할 Confluence 데이터 부분을 선택하는 데 도움이 됩니다.
   a. space: `include` 섹션에서 동기화하려는 스페이스 키를 나열합니다. 그렇지 않으면 `exclude` 섹션에서 제외할 공백을 나열합니다. 지정된 시간에 포함 또는 제외 옵션 중 하나만 제공합니다.
   b. page_type: 동기화하려는 특정 페이지 유형(예: 페이지, 블로그 게시물 또는 첨부 파일)입니다. 필요에 맞는 유사한 `include` 또는 `exclude` 패턴을 따릅니다. 첨부 파일은 원래 첨부 파일에 대한 앵커 링크로 표시되지만 첨부 파일 콘텐츠는 추출되지 않습니다.

데이터 일관성

파이프라인 YAML에 지정된 필터를 기반으로 선택한 프로젝트(또는 공백)가 한 번 추출되고 대상 싱크에 완전히 동기화됩니다. 그런 다음 지속적인 변경 모니터링을 통해 변경 사항이 발생할 때 이를 캡처하고 싱크의 데이터를 업데이트합니다. 한 가지 예외는 변경 모니터링이 및 `update` 작업만 동기화`create`하고 `delete` 작업은 동기화하지 않는다는 것입니다.

제한 사항

- 사용자 삭제 작업은 동기화되지 않습니다. 싱크에 기록된 데이터는 싱크에 남아 있습니다. 싱크 설정에서 ID 매핑을 지정하면 업데이트가 기존 콘텐츠를 새 변경 사항으로 덮어씁니다.
- 다음 APIs는이 소스와 호환되지 않습니다.
  - Jira Search API 버전 3
    - `rest/api/3/search`
    - `rest/api/3/issue`
  - Confluence
    - `wiki/rest/api/content/search`
    - `wiki/rest/api/content`
    - `wiki/rest/api/settings/systemInfo`

Atlassian 커넥터용 CloudWatch의 지표

유형: Jira 커넥터 지표

소스	지표	지표 유형
acknowledgementSetSuccesses.count	Counter	확인이 활성화된 경우이 지표는 성공적으로 동기화된 티켓 수를 제공합니다.
acknowledgementSetFailures.count	Counter	확인이 활성화된 경우이 지표는 동기화에 실패한 티켓 수를 제공합니다.

소스	지표	지표 유형
crawlingTime.avg	타이머	모든 새 변경 사항을 크롤링하는 데 걸린 시간입니다.
ticketFetchLatency.avg	타이머	티켓은 API 지연 시간 평균을 가져옵니다.
ticketFetchLatency.max	타이머	티켓은 API 지연 시간을 최대로 가져옵니다.
ticketsRequested.count	Counter	수행된 티켓 가져오기 요청 수입니다.
ticketRequestedFailed.count	Counter	티켓 가져오기 요청 수가 실패했습니다.
ticketRequestedSuccess.count	Counter	성공한 티켓 가져오기 요청 수입니다.
searchCallLatency.avg	타이머	API 호출 지연 시간 평균을 검색합니다.
searchCallLatency.max	타이머	API 호출 지연 시간 최대값을 검색합니다.
searchResultsFound.count	Counter	지정된 검색 호출에서 발견된 항목 수입니다.

소스	지표	지표 유형
searchRequestFailed.count	Counter	검색 API 호출 실패 수입니다.
authFailures.count	Counter	인증 실패 횟수입니다.

유형: Confluence 커넥터 지표

소스	지표	지표 유형
acknowledgementSetSuccesses.count	Counter	확인이 활성화된 경우이 지표는 성공적으로 동기화된 페이지 수를 제공합니다.
acknowledgementSetFailures.count	Counter	확인이 활성화된 경우이 지표는 동기화에 실패한 페이지 수를 제공합니다.
crawlingTime.avg	타이머	모든 새 변경 사항을 크롤링하는 데 걸린 시간입니다.
pageFetchLatency.avg	타이머	콘텐츠 가져오기 API 지연 시간(평균).
pageFetchLatency.max	타이머	콘텐츠 가져오기 API 지연 시간(최대).
pagesRequested.count	Counter	API를 가져오는 콘텐츠의 호출 수입니다.

소스	지표	지표 유형
pageRequestFailed.count	Counter	콘텐츠 가져오기 API의 실패한 요청 수입니다.
pageRequestedSuccess.count	Counter	API를 가져오는 콘텐츠의 성공적인 요청 수입니다.
searchCallLatency.avg	타이머	API 호출 지연 시간 평균을 검색합니다.
searchCallLatency.max	타이머	검색 API 호출 지연 시간 최대
searchResultsFound.count	Counter	지정된 검색 호출에서 발견된 항목 수입니다.
searchRequestsFailed.count	Counter	검색 API 호출 실패 수입니다.
authFailures.count	Counter	인증 실패 횟수입니다.

OAuth 2.0을 사용하여 Amazon OpenSearch Ingestion 파이프라인을 Atlassian Jira 또는 Confluence에 연결

이 주제의 정보를 사용하면 OAuth 2.0 인증을 사용하여 Amazon OpenSearch Ingestion 파이프라인을 구성하고 Jira 또는 Confluence 계정에 연결하는 데 도움이 됩니다. 이 Atlassian Services에서 OpenSearch Ingestion 파이프라인을 사용하기 the section called “사전 조건” 위해를 완료하는 경우이 작업을 수행하되 API 키 자격 증명을 사용하지 않도록 선택합니다.

주제

- OAuth 2.0 통합 앱 생성
- Atlassian Developer 액세스 토큰 생성 및 새로 고침

OAuth 2.0 통합 앱 생성

다음 절차에 따라 Atlassian Developer 웹 사이트에서 OAuth 2.0 통합 앱을 생성할 수 있습니다.

OAuth 2.0 통합 앱을 생성하려면

1. https://developer.atlassian.com/console/myapps/ Atlassian Developer 계정에 로그인합니다.
2. 생성, OAuth 2.0 통합을 선택합니다.
3. 이름에 앱의 목적을 식별할 이름을 입력합니다.
4. Atlassian의 개발자 약관에 구속된다는 데 동의합니다 확인란을 선택한 다음 생성을 선택합니다.
5. 왼쪽 탐색 창에서 권한 부여를 선택한 다음 추가를 선택합니다.
6. 콜백 URL에 **`https://www.amazon.com`** 또는 **`https://www.example.com`**와 같은 URL을 입력한 다음 변경 사항 저장을 선택합니다.
7. 왼쪽 탐색에서 권한 페이지를 선택한 다음 Jira API 행에서 추가를 선택한 다음 구성을 선택하고 모든 클래식 범위 읽기 권한(아래에 나와 있음)을 선택한 다음 저장을 선택합니다.
8. 세분화된 범위 탭을 선택한 다음 범위 편집을 선택하여 Jira API 편집 대화 상자를 엽니다.
9. 사용 중인 소스 플러그인에 대한 권한을 선택합니다.

   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. 저장을 선택합니다.

관련 정보는 Atlassian Developer 웹 사이트에서 OAuth 2.0(3LO) 구현 및 작업에 필요한 범위 결정을 참조하세요.

Atlassian Developer 액세스 토큰 생성 및 새로 고침

다음 절차를 사용하면 Atlassian Developer 웹 사이트에서 Atlassian Developer 액세스 토큰을 생성하고 새로 고치는 데 도움이 됩니다.

Jira 액세스 토큰을 생성하고 새로 고치려면

1. https://developer.atlassian.com/console/myapps/ Atlassian Developer 계정에 로그인합니다.
2. 에서 생성한 앱을 선택합니다the section called “OAuth 2.0 통합 앱 생성”.
3. 왼쪽 탐색 창에서 권한 부여를 선택합니다.
4. 페이지 하단에서 세분화된 Atlassian API 권한 부여 URL 값을 복사하여 원하는 텍스트 편집기에 붙여 넣습니다.

   URL의 형식은 다음과 같습니다.

   ```
   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. 의 경우 파라미터 값을 state="**sample_text**"와 같이 선택한 항목으로 `state=YOUR_USER_BOUND_VALUE`변경합니다.

   자세한 내용은 Atlassian Developer 웹 사이트의 상태 파라미터는 무엇입니까?를 참조하세요.

6. `scope` 섹션에는 이전 작업에서 선택한 세분화된 범위가 나열됩니다. 예: `scope=read%3Ajira-work%20read%3Ajira-user%20offline_access`

   `offline_access`는를 생성하고자 함을 나타냅니다`refresh_token`.

7. 웹 브라우저 창을 열고 복사한 권한 부여 URL을 브라우저 창의 주소 표시줄에 입력합니다.
8. 대상 페이지가 열리면 정보가 올바른지 확인한 다음 수락을 선택하여 Jira 또는 Confluence 홈페이지로 리디렉션합니다.
9. 홈페이지를 로드한 후이 페이지의 URL을 복사합니다. 여기에는 애플리케이션의 권한 부여 코드가 포함되어 있습니다. 이 코드를 사용하여 액세스 토큰을 생성합니다. 이후의 전체 섹션은 권한 부여 코드`code=`입니다.
10. 다음 cURL 명령을 사용하여 액세스 토큰을 생성합니다. `##  ###`를 자신의 정보로 바꿉니다.

> **Tip**
>
> Postman과 같은 타사 서비스를 사용할 수도 있습니다.

```
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"}'
```

이 명령에 대한 응답에는 `access_code` 및의 값이 포함됩니다`refresh_token`.

Amazon Aurora에서 OpenSearch Ingestion 파이프라인 사용

Amazon Aurora와 함께 OpenSearch Ingestion 파이프라인을 사용하여 기존 데이터를 내보내고 변경 사항(예: 생성, 업데이트 및 삭제)을 Amazon OpenSearch Service 도메인 및 컬렉션으로 스트리밍할 수 있습니다. OpenSearch Ingestion 파이프라인은 변경 데이터 캡처(CDC) 인프라를 통합하여 Amazon Aurora에서 데이터를 지속적으로 스트리밍할 수 있는 지연 시간이 짧은 대규모 방법을 제공합니다. Aurora MySQL 및 Aurora PostgreSQL이 지원됩니다.

Amazon Aurora를 소스로 사용하여 전체 초기 스냅샷을 사용하거나 사용하지 않고 데이터를 처리할 수 있는 두 가지 방법이 있습니다. 전체 초기 스냅샷은 지정된 테이블의 스냅샷이며이 스냅샷은 Amazon S3로 내보내집니다. 그러면 OpenSearch Ingestion 파이프라인이 이 스냅샷을 도메인의 한 인덱스로 보내거나 분할하여 도메인의 여러 인덱스로 보냅니다. Amazon Aurora 및 OpenSearch의 데이터를 일관되게 유지하기 위해 파이프라인은 Amazon Aurora 클러스터의 테이블에 있는 모든 생성, 업데이트 및 삭제 이벤트를 OpenSearch 인덱스 또는 인덱스에 저장된 문서와 동기화합니다.

전체 초기 스냅샷을 사용하면 OpenSearch Ingestion 파이프라인이 먼저 스냅샷을 수집한 다음 Amazon Aurora 변경 스트림에서 데이터 읽기를 시작합니다. 결국 Amazon Aurora와 OpenSearch 간에 거의 실시간에 가까운 데이터 일관성을 유지하고 따라잡습니다.

또한 Amazon Aurora와의 OpenSearch Ingestion 통합을 사용하여 변경 데이터 캡처를 추적하고 Aurora의 모든 업데이트를 OpenSearch로 수집할 수 있습니다. 다른 메커니즘의 전체 스냅샷이 이미 있거나 Amazon Aurora 클러스터의 데이터에 대한 모든 변경 사항을 캡처하려는 경우이 옵션을 선택합니다.

이 옵션을 선택하면 Aurora MySQL에 대한 바이너리 로깅을 구성하거나 클러스터에서 Aurora PostgreSQL에 대한 논리적 복제를 설정해야 합니다.

주제

- Aurora MySQL
- Aurora PostgreSQL

Aurora MySQL

Amazon Aurora for Aurora MySQL을 사용하여 OpenSearch Ingestion 파이프라인을 구성하려면 다음 단계를 완료하세요.

주제

- Aurora MySQL 사전 조건
- 1단계: 파이프라인 역할 구성
- 2단계: 파이프라인 생성
- 데이터 일관성
- 데이터 형식 매핑
- 제한 사항
- 권장되는 CloudWatch 경보

Aurora MySQL 사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. Amazon Aurora에서 사용자 지정 Aurora DB 클러스터 파라미터 그룹을 생성하여 바이너리 로깅을 구성합니다.

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

   또한 `binlog_transaction_compression` 파라미터가 로 설정되지 않았는지ON, `binlog_row_value_options` 파라미터가 로 설정되지 않았는지 확인합니다PARTIAL_JSON.

2. Aurora MySQL DB 클러스터를 선택하거나 생성하고 이전 단계에서 생성한 파라미터 그룹을 DB 클러스터와 연결합니다.
3. 이진 로그 보존을 24시간 이상으로 구성합니다.
4. Aurora 및의 암호 관리를 사용하여 Amazon Aurora 클러스터에서 사용자 이름과 암호 AWS Secrets Manager 인증을 설정합니다. Secrets Manager 보안 암호를 생성하여 사용자 이름/암호 조합을 생성할 수도 있습니다.
5. 전체 초기 스냅샷 기능을 사용하는 경우 Amazon Aurora에서 Amazon S3로 데이터를 내보내기 위한 AWS KMS key 및 IAM 역할을 생성합니다.

   IAM 역할에는 다음과 같은 권한 정책이 있어야 합니다.

   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/*"
            ]
        }
    ]
}
```

   역할에는 다음과 같은 신뢰 관계도 있어야 합니다.

   JSON

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

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "export.rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

6. OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 선택하거나 생성합니다. 자세한 내용은 OpenSearch Service 도메인 생성 및 컬렉션 생성을 참조하세요.
7. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 Amazon Aurora DB 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

1단계: 파이프라인 역할 구성

Amazon Aurora 파이프라인 사전 조건을 설정한 후 파이프라인 구성에 사용할 파이프라인 역할을 구성합니다. 또한 Amazon Aurora 소스에 대한 다음 권한을 역할에 추가합니다.

```
{
  "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:*:account-id:network-interface/*",
        "arn:aws:ec2:*:account-id:subnet/*",
        "arn:aws:ec2:*:account-id:security-group/*"
      ]
    },
    {
      "Sid": "allowDescribeEC2",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "allowTagCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:account-id:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/OSISManaged": "true"
        }
      }
    },
    {
      "Sid": "AllowDescribeInstances",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:db:*"
```

```
      ]
    },
    {
      "Sid": "AllowDescribeClusters",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBClusters"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:cluster:DB-id"
      ]
    },
    {
      "Sid": "AllowSnapshots",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBClusterSnapshots",
        "rds:CreateDBClusterSnapshot",
        "rds:AddTagsToResource"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:cluster:DB-id",
        "arn:aws:rds:region:account-id:cluster-snapshot:DB-id*"
      ]
    },
    {
      "Sid": "AllowExport",
      "Effect": "Allow",
      "Action": [
        "rds:StartExportTask"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:cluster:DB-id",
        "arn:aws:rds:region:account-id:cluster-snapshot:DB-id*"
      ]
    },
    {
      "Sid": "AllowDescribeExports",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeExportTasks"
      ],
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:RequestedRegion": "region",
          "aws:ResourceAccount": "account-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToKmsForExport",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:DescribeKey",
        "kms:RetireGrant",
        "kms:CreateGrant",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*"
      ],
      "Resource": [
        "arn:aws:kms:region:account-id:key/export-key-id"
      ]
      },
    {
      "Sid": "AllowPassingExportRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": [
        "arn:aws:iam::account-id:role/export-role"
      ]
    },
    {
      "Sid": "SecretsManagerReadAccess",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:*:account-id:secret:*"
      ]
    }
  ]
}
```

2단계: 파이프라인 생성

다음과 유사한 OpenSearch Ingestion 파이프라인을 구성합니다. 예제 파이프라인은 Amazon Aurora 클러스터를 소스로 지정합니다.

```
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
```

미리 구성된 Amazon Aurora 블루프린트를 사용하여이 파이프라인을 생성할 수 있습니다. 자세한 내용은 블루프린트 작업 단원을 참조하십시오.

Amazon Aurora를 소스로 사용하려면 파이프라인에 대한 VPC 액세스를 구성해야 합니다. 선택한 VPC는 Amazon Aurora 소스에서 사용하는 것과 동일한 VPC여야 합니다. 그런 다음 하나 이상의 서브넷과 하나 이상의 VPC 보안 그룹을 선택합니다. 파이프라인에는 Aurora MySQL 데이터베이스에 대한 네트워크 액세스가 필요하므로 Aurora 클러스터가 파이프라인의 VPC 보안 그룹에서 데이터베이스 포트로의 인바운드 트래픽을 허용하는 VPC 보안 그룹으로 구성되어 있는지도 확인해야 합니다. 자세한 내용은 보안 그룹을 사용한 액세스 제어를 참조하세요.

를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 Amazon Aurora를 소스로 사용하려면 VPC에 파이프라인도 연결해야 합니다. 이를 수행하려면 네트워크 구성 섹션을 찾아 VPC에 연결 확인란을 선택하고 제공된 기본 옵션 중 하나에서 CIDR을 선택하거나 직접 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.

사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 Amazon Aurora 간의 IP 주소 충돌을 방지하려면 Amazon Aurora VPC CIDR이 OpenSearch Ingestion의 CIDR과 다른지 확인합니다.

자세한 내용은 파이프라인에 대한 VPC 액세스 구성을 참조하세요.

데이터 일관성

파이프라인은 Amazon Aurora 클러스터에서 변경 사항을 지속적으로 폴링하거나 수신하고 OpenSearch 인덱스에서 해당 문서를 업데이트하여 데이터 일관성을 보장합니다.

OpenSearch Ingestion은 데이터 내구성을 보장하는 엔드 투 엔드 승인을 지원합니다. 파이프라인에서는 스냅샷이나 스트림을 읽을 때 병렬 처리를 위해 동적으로 분할을 생성합니다. 파이프라인에서는 OpenSearch 도메인이나 컬렉션에서 모든 레코드를 수집한 후 승인을 받으면 분할을 완료 상태로 표시합니다. OpenSearch Serverless 검색 컬렉션에 수집하려는 경우 파이프라인에서 문서 ID를 생성할 수 있습니다. OpenSearch Serverless 시계열 컬렉션에 수집하려면 파이프라인이 문서 ID를 생성하지 않으므로 파이프라인 싱크 구성document_id: "${getMetadata(\"primary_key\")}"에서를 생략해야 합니다.

또한 OpenSearch Ingestion 파이프라인에서는 수신 이벤트 작업을 해당하는 대량 인덱싱 작업에 매핑하여 문서를 쉽게 수집할 수 있게 합니다. 이렇게 하면 데이터가 일관되게 유지되므로 Amazon Aurora의 모든 데이터 변경 사항이 OpenSearch의 해당 문서 변경 사항과 조정됩니다.

데이터 형식 매핑

OpenSearch Ingestion 파이프라인은 MySQL 데이터 유형을 OpenSearch Service 도메인 또는 컬렉션이 사용할 수 있는 표현에 매핑합니다. OpenSearch에 매핑 템플릿이 정의되지 않은 경우 OpenSearch는 처음 전송된 문서를 기반으로 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 매핑 템플릿을 통해 OpenSearch에서 가장 적합한 필드 유형을 명시적으로 정의할 수도 있습니다.

아래 표에는 MySQL 데이터 유형과 해당 OpenSearch 필드 유형이 나열되어 있습니다. 명시적 매핑이 정의되지 않은 경우 기본 OpenSearch 필드 유형 열에 OpenSearch의 해당 필드 유형이 표시됩니다. 이 경우 OpenSearch는 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 권장 OpenSearch 필드 유형 열은 매핑 템플릿에서 명시적으로 지정하는 데 권장되는 해당 필드 유형입니다. 이러한 필드 유형은 MySQL의 데이터 유형과 더 밀접하게 정렬되며 일반적으로 OpenSearch에서 더 나은 검색 기능을 사용할 수 있습니다.

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
BIGINT	long	long
BIGINT UNSIGNED	long	서명되지 않은 길이
BIT	long	비트 수에 따라 바이트, 짧음, 정수 또는 깁니다.
DECIMAL	텍스트	double 또는 키워드
DOUBLE	float	double
FLOAT	float	float
INT	long	정수
INT UNSIGNED	long	long

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
MEDIUMIN	long	정수
MEDIUMIN UNSIGNED	long	정수
NUMERIC	텍스트	double 또는 키워드
SMALLINT	long	bigint
SMALLINT UNSIGNED	long	정수
TINYINT	long	바이트
TINYINT UNSIGNED	long	bigint
BINARY	텍스트	이진수
BLOB	텍스트	이진수
CHAR	텍스트	텍스트
ENUM	텍스트	키워드
LONGBLOE	텍스트	이진수
LONGTEXT	텍스트	텍스트
MEDIUMBL B	텍스트	이진수
MEDIUMTE T	텍스트	텍스트
SET	텍스트	키워드

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
TEXT	텍스트	텍스트
TINYBLOB	텍스트	이진수
TINYTEXT	텍스트	텍스트
VARBINAR	텍스트	이진수
VARCHAR	텍스트	텍스트
날짜	long(에포크 밀리초)	date
DATETIME	long(에포크 밀리초)	date
TIME	long(에포크 밀리초)	date
TIMESTAM	long(에포크 밀리초)	date
YEAR	long(에포크 밀리초)	date
GEOMETR	text(WKT 형식)	geo_shape
GEOMETR OLLECTIO	text(WKT 형식)	geo_shape
LINESTRIN G	text(WKT 형식)	geo_shape
MULTILINE STRING	text(WKT 형식)	geo_shape
MULTIPOI T	text(WKT 형식)	geo_shape
MULTIPOL GON	text(WKT 형식)	geo_shape

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
POINT	text(WKT 형식)	geo_point 또는 geo_shape
POLYGON	text(WKT 형식)	geo_shape
JSON	텍스트	객체

OpenSearch Ingestion 파이프라인에서 DLQ(Dead Letter Queue)를 구성하는 것이 좋습니다. 이 대기열을 구성하면 OpenSearch Service에서는 동적 매핑 실패로 인해 수집할 수 없는 모든 실패한 문서를 대기열로 전송합니다.

자동 매핑이 실패하면 파이프라인 구성`template_content`에서 `template_type` 및를 사용하여 명시적 매핑 규칙을 정의할 수 있습니다. 또는 파이프라인을 시작하기 전에 검색 도메인이나 컬렉션에서 직접 매핑 템플릿을 생성할 수도 있습니다.

제한 사항

Aurora MySQL용 OpenSearch Ingestion 파이프라인을 설정할 때 다음 제한 사항을 고려하세요.

- 통합은 파이프라인당 하나의 MySQL 데이터베이스만 지원합니다.
- 통합은 현재 리전 간 데이터 수집을 지원하지 않습니다. Amazon Aurora 클러스터와 OpenSearch 도메인은 동일한에 있어야 합니다 AWS 리전.
- 통합은 현재 교차 계정 데이터 수집을 지원하지 않습니다. Amazon Aurora 클러스터와 OpenSearch Ingestion 파이프라인은 동일한에 있어야 합니다 AWS 계정.
- 지원되는 유일한 인증 메커니즘인 Secrets Manager를 사용하여 Amazon Aurora 클러스터에 인증이 활성화되어 있는지 확인합니다.
- 다른 데이터베이스 및/또는 다른 테이블에서 데이터를 수집하도록 기존 파이프라인 구성을 업데이트할 수 없습니다. 파이프라인의 데이터베이스 및/또는 테이블 이름을 업데이트하려면 파이프라인을 중지하고 업데이트된 구성으로 다시 시작하거나 새 파이프라인을 생성해야 합니다.
- 데이터 정의 언어(DDL) 문은 일반적으로 지원되지 않습니다. 다음과 같은 경우에는 데이터 일관성이 유지되지 않습니다.
  - 기본 키가 변경됩니다(add/delete/rename).
  - 테이블이 삭제/절단됩니다.

  - 열 이름 또는 데이터 형식이 변경됩니다.
- 동기화할 MySQL 테이블에 기본 키가 정의되지 않은 경우 데이터 일관성이 보장되지 않습니다. 업데이트/삭제를 OpenSearch에 동기화하려면 OpenSearch 싱크 구성에서 사용자 지정 `document_id` 옵션을 올바르게 정의해야 합니다.
- 계단식 삭제 작업이 있는 외래 키 참조는 지원되지 않으며 Aurora MySQL과 OpenSearch 간에 데이터 불일치가 발생할 수 있습니다.
- 지원되는 버전: Aurora MySQL 버전 3.05.2 이상.

권장되는 CloudWatch 경보

수집 파이프라인의 성능을 모니터링하려면 다음 CloudWatch 지표를 사용하는 것이 좋습니다. 이러한 지표는 내보내기에서 처리된 데이터의 양, 스트림에서 처리된 이벤트 수, 내보내기 및 스트림 이벤트 처리 오류, 대상에 기록된 문서 수를 식별하는 데 도움이 될 수 있습니다. 이러한 지표 중 하나가 지정된 시간 동안 지정된 값을 초과하면 작업을 수행하도록 CloudWatch 경보를 설정할 수 있습니다.

지표	설명
pipeline-name .rds.credentialsChanged	이 지표는 AWS 보안 암호가 교체되는 빈도를 나타냅니다.
pipeline-name .rds.executorRefreshErrors	이 지표는 AWS 보안 암호 새로 고침 실패를 나타냅니다.
pipeline-name .rds.exportRecordsTotal	이 지표는 Amazon Aurora에서 내보낸 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordsProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordProcessingErrors	이 지표는 Amazon Aurora 클러스터에서 데이터를 읽는 동안 OpenSearch Ingestion 파이프라인의 처리 오류 수를 나타냅니다.

지표	설명
pipeline-name .rds.exportRecordsSuccessTotal	이 지표는 성공적으로 처리한 내보내기 레코드의 총 수를 나타냅니다.
pipeline-name .rds.exportRecordsFailedTotal	이 지표는 처리하지 못한 내보내기 레코드의 총 수를 나타냅니다.
pipeline-name .rds.bytesReceived	이 지표는 OpenSearch Ingestion 파이프라인에서 수신한 총 바이트 수를 나타냅니다.
pipeline-name .rds.bytesProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 총 바이트 수를 나타냅니다.
pipeline-name .rds.streamRecordsSuccessTotal	이 지표는 스트림에서 성공적으로 처리한 레코드 수를 나타냅니다.
pipeline-name .rds.streamRecordsFailedTotal	이 지표는 스트림에서 처리하지 못한 총 레코드 수를 나타냅니다.

Aurora PostgreSQL

Amazon Aurora for Aurora PostgreSQL을 사용하여 OpenSearch Ingestion 파이프라인을 구성하려면 다음 단계를 완료하세요.

주제

- Aurora PostgreSQL 사전 조건

- 1단계: 파이프라인 역할 구성
- 2단계: 파이프라인 생성
- 데이터 일관성
- 데이터 형식 매핑
- 제한 사항
- 권장되는 CloudWatch 경보

Aurora PostgreSQL 사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. Amazon Aurora에서 사용자 지정 DB 클러스터 파라미터 그룹을 생성하여 논리적 복제를 구성합니다.

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

2. Aurora PostgreSQL DB 클러스터를 선택하거나 생성하고 1단계에서 생성한 파라미터 그룹을 DB 클러스터와 연결합니다.
3. Aurora 및에서 암호 관리를 사용하여 Amazon Aurora 클러스터에서 사용자 이름과 암호 AWS Secrets Manager 인증을 설정합니다. Secrets Manager 보안 암호를 생성하여 사용자 이름/암호 조합을 생성할 수도 있습니다.
4. 전체 초기 스냅샷 기능을 사용하는 경우 Amazon Aurora에서 Amazon S3로 데이터를 내보내기 위한 AWS KMS key 및 IAM 역할을 생성합니다.

   IAM 역할에는 다음과 같은 권한 정책이 있어야 합니다.

   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/*"
            ]
        }
    ]
}
```

역할에는 다음과 같은 신뢰 관계도 있어야 합니다.

JSON

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

5. OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 선택하거나 생성합니다. 자세한 내용은 OpenSearch Service 도메인 생성 및 컬렉션 생성을 참조하세요.
6. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 Amazon Aurora DB 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

1단계: 파이프라인 역할 구성

Amazon Aurora 파이프라인 사전 조건을 설정한 후 파이프라인 구성에 사용할 파이프라인 역할을 구성합니다. 또한 Amazon Aurora 소스에 대한 다음 권한을 역할에 추가합니다.

```
{
  "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:*:account-id:network-interface/*",
        "arn:aws:ec2:*:account-id:subnet/*",
        "arn:aws:ec2:*:account-id:security-group/*"
      ]
    },
    {
      "Sid": "allowDescribeEC2",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*"
```

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

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

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

2단계: 파이프라인 생성

Aurora PostgreSQL 클러스터를 소스로 지정하는 다음과 같이 OpenSearch Ingestion 파이프라인을 구성합니다.

```
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**
>
> 미리 구성된 Amazon Aurora 블루프린트를 사용하여이 파이프라인을 생성할 수 있습니다. 자세한 내용은 블루프린트 작업 단원을 참조하십시오.

Amazon Aurora를 소스로 사용하려면 파이프라인에 대한 VPC 액세스를 구성해야 합니다. 선택한 VPC는 Amazon Aurora 소스에서 사용하는 것과 동일한 VPC여야 합니다. 그런 다음 하나 이상의 서브넷과 하나 이상의 VPC 보안 그룹을 선택합니다. 파이프라인에는 Aurora MySQL 데이터베이스에 대한 네트워크 액세스가 필요하므로 Aurora 클러스터가 파이프라인의 VPC 보안 그룹에서 데이터베이스 포트로의 인바운드 트래픽을 허용하는 VPC 보안 그룹으로 구성되어 있는지도 확인해야 합니다. 자세한 내용은 보안 그룹을 사용한 액세스 제어를 참조하세요.

를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 Amazon Aurora를 소스로 사용하려면 VPC에 파이프라인도 연결해야 합니다. 이렇게 하려면 네트워크 구성 섹션을 찾고 VPC에 연결을 선택한 다음 제공된 기본 옵션 중 하나에서 CIDR을 선택하거나 직접 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.

사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 Amazon Aurora 간의 IP 주소 충돌을 방지하려면 Amazon Aurora VPC CIDR이 OpenSearch Ingestion의 CIDR과 다른지 확인합니다.

자세한 내용은 파이프라인에 대한 VPC 액세스 구성을 참조하세요.

데이터 일관성

파이프라인은 Amazon Aurora 클러스터의 변경 사항을 지속적으로 폴링하거나 수신하고 OpenSearch 인덱스에서 해당 문서를 업데이트하여 데이터 일관성을 보장합니다.

OpenSearch Ingestion은 데이터 내구성을 보장하는 엔드 투 엔드 승인을 지원합니다. 파이프라인에서는 스냅샷이나 스트림을 읽을 때 병렬 처리를 위해 동적으로 분할을 생성합니다. 파이프라인에서는 OpenSearch 도메인이나 컬렉션에서 모든 레코드를 수집한 후 승인을 받으면 분할을 완료 상태로 표시합니다. OpenSearch Serverless 검색 컬렉션에 수집하려는 경우 파이프라인에서 문서 ID를 생성할 수 있습니다. OpenSearch Serverless 시계열 컬렉션에 수집하려면 파이프라인이 문서 ID를 생성하지 않으므로 파이프라인 싱크 구성`document_id: "${getMetadata(\"primary_key\")}"`에서를 생략해야 합니다.

또한 OpenSearch Ingestion 파이프라인에서는 수신 이벤트 작업을 해당하는 대량 인덱싱 작업에 매핑하여 문서를 쉽게 수집할 수 있게 합니다. 이렇게 하면 데이터가 일관되게 유지되므로 Amazon Aurora의 모든 데이터 변경 사항이 OpenSearch의 해당 문서 변경 사항과 조정됩니다.

데이터 형식 매핑

OpenSearch Ingestion 파이프라인은 Aurora PostgreSQL 데이터 형식을 OpenSearch Service 도메인 또는 컬렉션이 사용할 수 있는 표현에 매핑합니다. OpenSearch에 매핑 템플릿이 정의되지 않은 경우 OpenSearch는 처음 전송된 문서를 기반으로 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 매핑 템플릿을 통해 OpenSearch에서 가장 적합한 필드 유형을 명시적으로 정의할 수도 있습니다.

아래 표에는 Aurora PostgreSQL 데이터 형식과 해당 OpenSearch 필드 형식이 나열되어 있습니다. 명시적 매핑이 정의되지 않은 경우 기본 OpenSearch 필드 유형 열에 OpenSearch의 해당 필드 유형이 표시됩니다. 이 경우 OpenSearch는 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 권장 OpenSearch 필드 유형 열은 매핑 템플릿에서 명시적으로 지정하는 해당 권장 필드 유형입니다. 이러

한 필드 유형은 Aurora PostgreSQL의 데이터 유형과 더 밀접하게 정렬되며 일반적으로 OpenSearch에서 더 나은 검색 기능을 사용할 수 있습니다.

Aurora PostgreSQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
smallint	long	bigint
정수	long	정수
bigint	long	long
decimal	텍스트	double 또는 키워드
숫자[(p, s)]	텍스트	double 또는 키워드
real	float	float
double precision	float	double
smallserial	long	bigint
serial	long	정수
bigserial	long	long
money	객체	객체
character varying(n)	텍스트	텍스트
varchar(n)	텍스트	텍스트
character (n)	텍스트	텍스트

Aurora PostgreSQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
char(n)	텍스트	텍스트
bpchar(n)	텍스트	텍스트
bpchar	텍스트	텍스트
텍스트	텍스트	텍스트
enum	텍스트	텍스트
bytea	텍스트	이진수
타임스탬프 [(p)] [시간대 없음]	long(에포크 밀리초)	date
시간대가 있는 타임스탬프 [(p)]	long(에포크 밀리초)	date
date	long(에포크 밀리초)	date
time [(p)] (시간대 제외)	long(에포크 밀리초)	date
시간대가 있는 시간 [(p)]	long(에포크 밀리초)	date
간격 [필드] [(p)]	text(ISO8601 형식)	텍스트

Aurora PostgreSQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
boolean	boolean	boolean
point	text(WKT 형식)	geo_shape
line	text(WKT 형식)	geo_shape
lseg	text(WKT 형식)	geo_shape
상자	text(WKT 형식)	geo_shape
경로	text(WKT 형식)	geo_shape
다각형	text(WKT 형식)	geo_shape
원	객체	객체
cidr	텍스트	텍스트
inet	텍스트	텍스트
macaddr	텍스트	텍스트
macaddr8	텍스트	텍스트
bit(n)	long	byte, short, integer 또는 long(비트 수에 따라 다름)
bit varying(n)	long	byte, short, integer 또는 long(비트 수에 따라 다름)
json	객체	객체
jsonb	객체	객체
jsonpath	텍스트	텍스트

OpenSearch Ingestion 파이프라인에서 DLQ(Dead Letter Queue)를 구성하는 것이 좋습니다. 이 대기열을 구성하면 OpenSearch Service에서는 동적 매핑 실패로 인해 수집할 수 없는 모든 실패한 문서를 대기열로 전송합니다.

자동 매핑이 실패할 경우 파이프라인 구성에서 `template_type` 및 `template_content`를 사용하여 명시적 매핑 규칙을 정의할 수 있습니다. 또는 파이프라인을 시작하기 전에 검색 도메인이나 컬렉션에서 직접 매핑 템플릿을 생성할 수도 있습니다.

제한 사항

Aurora PostgreSQL용 OpenSearch Ingestion 파이프라인을 설정할 때 다음 제한 사항을 고려하세요.

- 통합은 파이프라인당 하나의 Aurora PostgreSQL 데이터베이스만 지원합니다.
- 통합은 현재 리전 간 데이터 수집을 지원하지 않습니다. Amazon Aurora 클러스터와 OpenSearch 도메인은 동일한에 있어야 합니다 AWS 리전.
- 통합은 현재 교차 계정 데이터 수집을 지원하지 않습니다. Amazon Aurora 클러스터와 OpenSearch Ingestion 파이프라인은 동일한에 있어야 합니다 AWS 계정.
- 지원되는 유일한 인증 메커니즘 AWS Secrets Manager인를 사용하여 Amazon Aurora 클러스터에 인증이 활성화되어 있는지 확인합니다.
- 기존 파이프라인 구성은 다른 데이터베이스 및/또는 다른 테이블에서 데이터를 수집하도록 업데이트할 수 없습니다. 파이프라인의 데이터베이스 및/또는 테이블 이름을 업데이트하려면 파이프라인을 중지하고 업데이트된 구성으로 다시 시작하거나 새 파이프라인을 생성해야 합니다.
- 데이터 정의 언어(DDL) 문은 일반적으로 지원되지 않습니다. 다음과 같은 경우에는 데이터 일관성이 유지되지 않습니다.
  - 기본 키가 변경됩니다(add/delete/rename).
  - 테이블이 삭제/절단됩니다.
  - 열 이름 또는 데이터 형식이 변경됩니다.
- 동기화할 Aurora PostgreSQL 테이블에 기본 키가 정의되지 않은 경우 데이터 일관성이 보장되지 않습니다. 업데이트/삭제를 OpenSearch에 동기화하려면 OpenSearch 및 싱크 구성에서 사용자 지정 `document_id` 옵션을 올바르게 정의해야 합니다 OpenSearch.
- 지원되는 버전: Aurora PostgreSQL 버전 16.4 이상.

권장되는 CloudWatch 경보

수집 파이프라인의 성능을 모니터링하려면 다음 CloudWatch 지표를 사용하는 것이 좋습니다. 이러한 지표는 내보내기에서 처리된 데이터의 양, 스트림에서 처리된 이벤트 수, 내보내기 및 스트림 이벤트

처리 오류, 대상에 기록된 문서 수를 식별하는 데 도움이 될 수 있습니다. 이러한 지표 중 하나가 지정된 시간 동안 지정된 값을 초과하면 작업을 수행하도록 CloudWatch 경보를 설정할 수 있습니다.

지표	설명
pipeline-name .rds.credentialsChanged	이 지표는 AWS 보안 암호 교체 빈도를 나타냅니다.
pipeline-name .rds.executorRefreshErrors	이 지표는 AWS 보안 암호 새로 고침 실패를 나타냅니다.
pipeline-name .rds.exportRecordsTotal	이 지표는 Amazon Aurora에서 내보낸 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordsProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordProcessingErrors	이 지표는 Amazon Aurora 클러스터에서 데이터를 읽는 동안 OpenSearch Ingestion 파이프라인의 처리 오류 수를 나타냅니다.
pipeline-name .rds.exportRecordsSuccessTotal	이 지표는 성공적으로 처리한 내보내기 레코드의 총 수를 나타냅니다.
pipeline-name .rds.exportRecordsFailedTotal	이 지표는 처리하지 못한 내보내기 레코드의 총 수를 나타냅니다.

지표	설명
pipeline-name .rds.bytesReceived	이 지표는 OpenSearch Ingestion 파이프라인에서 수신한 총 바이트 수를 나타냅니다.
pipeline-name .rds.bytesProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 총 바이트 수를 나타냅니다.
pipeline-name .rds.streamRecordsSuccessTotal	이 지표는 스트림에서 성공적으로 처리한 레코드 수를 나타냅니다.
pipeline-name .rds.streamRecordsFailedTotal	이 지표는 스트림에서 처리하지 못한 총 레코드 수를 나타냅니다.

Amazon DynamoDB와 함께 OpenSearch Ingestion 파이프라인 사용

DynamoDB 플러그인을 사용하여 생성, 업데이트 및 삭제와 같은 테이블 이벤트를 Amazon OpenSearch Service 도메인 및 Amazon OpenSearch Serverless 컬렉션으로 스트리밍할 수 있습니다. 파이프라인은 지연 시간이 짧은 대규모 스트리밍을 위해 변경 데이터 캡처(CDC)를 사용합니다.

전체 초기 스냅샷을 사용하거나 사용하지 않고 DynamoDB 데이터를 처리할 수 있습니다.

- 전체 스냅샷 사용 - DynamoDB는 point-in-time 복구(PITR)를 사용하여 백업을 생성하고 Amazon S3에 업로드합니다. 그런 다음 OpenSearch Ingestion은 스냅샷을 하나 이상의 OpenSearch 인덱스로 인덱싱합니다. 일관성을 유지하기 위해 파이프라인은 모든 DynamoDB 변경 사항을 OpenSearch와 동기화합니다. 이 옵션을 사용하려면 PITR 스트림과 DynamoDB 스트림을 모두 활성화해야 합니다.
- 스냅샷 없음 - OpenSearch Ingestion은 새 DynamoDB 이벤트만 스트리밍합니다. 이미 스냅샷이 있거나 기록 데이터 없이 실시간 스트리밍이 필요한 경우이 옵션을 선택합니다. 이 옵션을 사용하려면 DynamoDB 스트림만 활성화해야 합니다.

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB zero-ETL integration with Amazon OpenSearch Service를 참조하세요.

주제

- 사전 조건
- 1단계: 파이프라인 역할 구성
- 2단계: 파이프라인 생성
- 데이터 일관성
- 데이터 형식 매핑
- 제한 사항
- DynamoDB에 대해 권장되는 CloudWatch 경보

사전 조건

파이프라인을 설정하려면 DynamoDB 스트림이 활성화된 DynamoDB 테이블이 있어야 합니다. 스트림은 NEW_IMAGE 스트림 뷰 유형을 사용해야 합니다. 하지만 NEW_AND_OLD_IMAGES 유형이 사용 사례에 맞는 경우 OpenSearch Ingestion 파이프라인에서는 이 스트림 뷰 유형으로도 이벤트를 스트리밍할 수 있습니다.

스냅샷을 사용하는 경우 테이블에서 특정 시점 복구도 활성화해야 합니다. 자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 생성, 특정 시점으로 복구 활성화 및 스트림 활성화를 참조하세요.

1단계: 파이프라인 역할 구성

DynamoDB 테이블을 설정한 후 파이프라인 구성에서 사용하려는 파이프라인 역할을 설정하고 다음 DynamoDB 권한을 해당 역할에 추가합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowRunExportJob",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeContinuousBackups",
```

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

AWS KMS 고객 관리형 키를 사용하여 내보내기 데이터 파일을 암호화할 수도 있습니다. 내보낸 객체를 해독하려면 파이프라인의 내보내기 구성에서 키 ID에 `arn:aws:kms:region:account-id:key/my-key-id` 형식으로 `s3_sse_kms_key_id`를 지정합니다. 다음 정책에는 고객 관리형 키를 사용하는 데 필요한 권한이 포함되어 있습니다.

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

2단계: 파이프라인 생성

이제 DynamoDB를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다. 이 샘플 파이프라인은 PITR 스냅샷이 있는 `table-a`에서 데이터를 수집한 다음 DynamoDB 스트림에서 이벤트를 수집합니다. `LATEST` 시작 위치는 파이프라인이 DynamoDB 스트림에서 최신 데이터를 읽어야 함을 나타냅니다.

```
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"
```

사전 구성된 DynamoDB 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

데이터 일관성

OpenSearch Ingestion은 데이터 내구성을 보장하는 엔드 투 엔드 승인을 지원합니다. 파이프라인에서는 스냅샷이나 스트림을 읽을 때 병렬 처리를 위해 동적으로 분할을 생성합니다. 파이프라인에서는 OpenSearch 도메인이나 컬렉션에서 모든 레코드를 수집한 후 승인을 받으면 분할을 완료 상태로 표시합니다.

OpenSearch Serverless 검색 컬렉션에 수집하려는 경우 파이프라인에서 문서 ID를 생성할 수 있습니다. OpenSearch Serverless 시계열 컬렉션에 수집하려는 경우에는 파이프라인에서 문서 ID를 생성하지 않습니다.

또한 OpenSearch Ingestion 파이프라인에서는 수신 이벤트 작업을 해당하는 대량 인덱싱 작업에 매핑하여 문서를 쉽게 수집할 수 있게 합니다. 이렇게 하면 데이터 일관성이 유지되므로 DynamoDB의 모든 데이터 변경 사항이 OpenSearch에서 해당하는 문서 변경 사항으로 조정됩니다.

데이터 형식 매핑

OpenSearch Service에서는 각 수신 문서의 데이터 형식을 DynamoDB의 해당 데이터 형식에 동적으로 매핑합니다. 다음 표에서는 OpenSearch Service에서 다양한 데이터 형식을 자동으로 매핑하는 방법을 보여줍니다.

데이터 유형	OpenSearch	DynamoDB
숫자	OpenSearch에서는 숫자 데이터를 자동으로 매핑합니다. 숫자가 정수인 경우 OpenSearch에서는 숫자를 긴 정수 값으로 매핑합니다. 숫자가 소수인 경우 OpenSearch에서는 숫자를 부동 소수점 값으로 매핑합니다. OpenSearch에서는 처음 보낸 문서를 기반으로 다양한 속성을 동적으로 매핑합니	DynamoDB는 숫자를 지원합니다.

데이터 유형	OpenSearch	DynamoDB
	다. DynamoDB의 동일한 속성에 정수와 분수와 같은 여러 데이터 형식이 혼합되어 있는 경우 매핑이 실패할 수 있습니다. 예를 들어 첫 번째 문서에는 정수인 속성이 있고 이후 문서에는 동일한 속성이 소수로 되어 있는 경우 OpenSearch에서 두 번째 문서를 수집하지 못합니다. 이러한 경우에는 다음과 같은 명시적 매핑 템플릿을 제공해야 합니다. `{ "template": { "mappings": { "properties": { "MixedNumberAttribute": { "type": "float" } } } } }` 배정밀도가 필요한 경우 문자열 형식의 필드 매핑을 사용합니다. OpenSearch에는 38자리 정밀도를 지원하는 동등한 숫자 형식이 없습니다.	
숫자 집합	OpenSearch에서는 숫자 집합을 긴 정수 값이나 부동 소수점 값의 배열에 자동으로 매핑합니다. 스칼라 수와 마찬가지로, 매핑은 수집된 첫 번째 숫자가 정수인지 소수인지에 따라 달라집니다. 스칼라 문자열을 매핑하는 것과 같은 방식으로 숫자 집합에 대한 매핑을 제공할 수 있습니다.	DynamoDB는 숫자 집합을 나타내는 형식을 지원합니다.

데이터 유형	OpenSearch	DynamoDB
String	OpenSearch에서는 문자열 값을 자동으로 텍스트에 매핑합니다. 열거된 값과 같은 일부 상황에서는 키워드 형식에 매핑할 수 있습니다. 다음 예제에서는 이름이 `PartType`인 DynamoDB 속성을 OpenSearch 키워드에 매핑하는 방법을 보여줍니다. `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	DynamoDB는 문자열을 지원합니다.
문자열 집합	OpenSearch에서는 문자열 집합을 자동으로 문자열 배열에 매핑합니다. 스칼라 문자열을 매핑하는 것과 같은 방식으로 문자열 집합에 대한 매핑을 제공할 수 있습니다.	DynamoDB는 문자열 집합을 나타내는 형식을 지원합니다.

데이터 유형	OpenSearch	DynamoDB
바이너리	OpenSearch에서는 바이너리 데이터를 자동으로 텍스트에 매핑합니다. OpenSearch에서 이러한 데이터를 바이너리 필드로 작성하기 위한 매핑을 제공할 수 있습니다. 다음 예제에서는 이름이 `ImageData` 인 DynamoDB 속성을 OpenSearch 바이너리 필드에 매핑하는 방법을 보여줍니다. `{ "template": { "mappings": { "properties": { "ImageData": { "type": "binary" } } } } }`	DynamoDB는 이진수 형식 속성을 지원합니다.
이진수 집합	OpenSearch에서는 이진수 집합을 텍스트 형식인 이진수 데이터 배열에 자동으로 매핑합니다. 스칼라 이진수를 매핑하는 것과 같은 방식으로 숫자 집합에 대한 매핑을 제공할 수 있습니다.	DynamoDB에서는 이진수 값 집합을 나타내는 형식을 지원합니다.
불	OpenSearch에서는 DynamoDB 부울 형식을 OpenSearch 부울 형식으로 매핑합니다.	DynamoDB에서는 부울 형식 속성을 지원합니다.

데이터 유형	OpenSearch	DynamoDB
Null	OpenSearch에서는 DynamoDB null 형식의 문서를 수집할 수 있습니다. 값을 문서에 null 값으로 저장합니다. 이 형식에는 매핑이 없으며 이 필드는 인덱싱되거나 검색할 수 없습니다. 한 속성 이름을 null 형식에 사용한 후 나중에 문자열과 같은 다른 형식으로 변경하면 OpenSearch에서 null이 아닌 첫 번째 값에 대해 동적 매핑을 생성합니다. 후속 값은 여전히 DynamoDB null 값일 수 있습니다.	DynamoDB는 null 형식 속성을 지원합니다.
맵	OpenSearch에서는 DynamoDB 맵 속성을 중첩 필드에 매핑합니다. 중첩 필드 내에도 동일한 매핑이 적용됩니다. 다음 예제에서는 중첩 필드의 문자열을 OpenSearch의 키워드 형식에 매핑합니다. `{ "template": { "mappings": { "properties": { "AdditionalDescriptions": { "properties": { "PartType": { "type": "keyword" } } } } } } }`	DynamoDB는 맵 형식 속성을 지원합니다.

데이터 유형	OpenSearch	DynamoDB
나열	OpenSearch에서는 목록의 내용에 따라 DynamoDB 목록에 대해 다른 결과를 제공합니다. 목록에 모두 동일한 유형의 스칼라 형식(예: 모든 문자열 목록)이 포함된 경우 OpenSearch에서는 해당 목록을 해당 형식의 배열로 수집합니다. 이 방식은 문자열, 숫자, 부울 및 null 유형에서 작동합니다. 각 형식에 대한 제한은 해당 형식의 스칼라에 대한 제한과 동일합니다. 맵에 사용하는 것과 동일한 매핑을 사용하여 맵 목록에 대한 매핑을 제공할 수도 있습니다. 혼합 형식 목록은 제공할 수 없습니다.	DynamoDB는 목록 형식 속성을 지원합니다.
설정	OpenSearch에서는 집합의 내용에 따라 DynamoDB 집합에 대해 다른 결과를 제공합니다. 집합에 모두 동일한 유형의 스칼라 형식(예: 모든 문자열 집합)이 포함된 경우 OpenSearch에서는 해당 집합을 해당 형식의 배열로 수집합니다. 이 방식은 문자열, 숫자, 부울 및 null 유형에서 작동합니다. 각 형식에 대한 제한은 해당 형식의 스칼라에 대한 제한과 동일합니다. 맵에 사용하는 것과 동일한 매핑을 사용하여 맵 집합에 대한 매핑을 제공할 수도 있습니다. 혼합 형식 집합은 제공할 수 없습니다.	DynamoDB는 집합을 나타내는 형식을 지원합니다.

OpenSearch Ingestion 파이프라인에서 DLQ(Dead Letter Queue)를 구성하는 것이 좋습니다. 이 대기열을 구성하면 OpenSearch Service에서는 동적 매핑 실패로 인해 수집할 수 없는 모든 실패한 문서를 대기열로 전송합니다.

자동 매핑이 실패할 경우 파이프라인 구성에서 `template_type` 및 `template_content`를 사용하여 명시적 매핑 규칙을 정의할 수 있습니다. 또는 파이프라인을 시작하기 전에 검색 도메인이나 컬렉션에서 직접 매핑 템플릿을 생성할 수도 있습니다.

제한 사항

DynamoDB용 OpenSearch Ingestion 파이프라인을 설정하는 경우 다음과 같은 제한 사항을 고려하세요.

- DynamoDB와의 OpenSearch Ingestion 통합 기능에서는 현재 교차 리전 수집을 지원하지 않습니다. DynamoDB 테이블과 OpenSearch Ingestion 파이프라인은 동일한 AWS 리전에 속해야 합니다.
- DynamoDB 테이블과 OpenSearch Ingestion 파이프라인은 동일한 AWS 계정에 속해야 합니다.
- OpenSearch Ingestion 파이프라인은 단일 DynamoDB 테이블만 소스로 지원합니다.
- DynamoDB 스트림은 최대 24시간 동안만 데이터를 로그에 저장합니다. 대규모 테이블의 초기 스냅샷에서 수집하는 데 24시간 이상 걸리는 경우 일부 초기 데이터 손실이 발생합니다. 이러한 데이터 손실 문제를 완화하려면 테이블 크기를 예측하고 OpenSearch Ingestion 파이프라인의 적절한 컴퓨팅 유닛을 구성하십시오.

DynamoDB에 대해 권장되는 CloudWatch 경보

수집 파이프라인의 성능을 모니터링하려면 다음 CloudWatch 지표가 권장됩니다. 이러한 지표는 내보내기에서 처리된 데이터의 양, 스트림에서 처리된 이벤트의 양, 내보내기 및 스트림 이벤트 처리 오류, 대상에 쓴 문서 수를 식별하는 데 도움이 될 수 있습니다. 이러한 지표 중 하나가 지정된 시간 동안 지정된 값을 초과하면 작업을 수행하도록 CloudWatch 경보를 설정할 수 있습니다.

지표	설명
`dynamodb-pipeline.BlockingBuffer.bufferUsage.value`	사용 중인 버퍼의 양을 나타냅니다.
`dynamodb-pipeline.dynamodb.activeExportS3ObjectConsumers.value`	내보내기를 위해 Amazon S3 객체를 적극적으로 처리하는 총 OCU 수를 표시합니다.

지표	설명
dynamodb-pipeline.dynamodb.bytesProcessed.count	DynamoDB 소스에서 처리된 바이트 수.
dynamodb-pipeline.dynamodb.changeEventsProcessed.count	DynamoDB 스트림에서 처리된 변경 이벤트 수.
dynamodb-pipeline.dynamodb.changeEventsProcessingErrors.count	DynamoDB에서 처리된 변경 이벤트의 오류 수.
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	내보내기에서 처리된 총 레코드 수.
dynamodb-pipeline.dynamodb.exportRecordsTotal.count	데이터 내보내기 볼륨을 추적하는 데 필요한 DynamoDB에서 내보낸 레코드의 총 수.
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.

지표	설명
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.
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.

지표	설명
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.
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.

Amazon DocumentDB에서 OpenSearch Ingestion 파이프라인 사용

DocumentDB 플러그인을 사용하여 생성, 업데이트 및 삭제와 같은 문서 변경 사항을 Amazon OpenSearch Service로 스트리밍할 수 있습니다. 파이프라인은 가능한 경우 변경 데이터 캡처(CDC) 또는 지연 시간이 짧은 대규모 스트리밍을 위한 API 폴링을 지원합니다.

전체 초기 스냅샷을 사용하거나 사용하지 않고 데이터를 처리할 수 있습니다. 전체 스냅샷은 전체 Amazon DocumentDB 컬렉션을 캡처하여 Amazon S3에 업로드합니다. 그런 다음 파이프라인은 데이터를 하나 이상의 OpenSearch 인덱스로 전송합니다. 스냅샷을 수집한 후 파이프라인은 지속적인 변경 사항을 동기화하여 일관성을 유지하고 결국 거의 실시간에 가까운 업데이트를 포착합니다.

다른 소스의 전체 스냅샷이 이미 있거나 새 이벤트만 처리해야 하는 경우 스냅샷 없이 스트리밍할 수 있습니다. 이 경우 파이프라인은 초기 대량 로드 없이 Amazon DocumentDB 변경 스트림에서 직접 읽습니다.

스트리밍을 활성화하는 경우 Amazon DocumentDB 컬렉션에서 변경 스트림을 활성화해야 합니다. 그러나 전체 로드 또는 내보내기만 수행하는 경우 변경 스트림이 필요하지 않습니다.

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. Amazon DocumentDB 개발자 안내서의 Create an Amazon DocumentDB cluster에 나온 단계를 수행하여 데이터 읽기 권한을 가진 Amazon DocumentDB 클러스터를 생성합니다. CDC 인프라를 사용하는 경우 변경 스트림을 게시하도록 Amazon DocumentDB 클러스터를 구성합니다.
2. Amazon DocumentDB 클러스터에서 TLS를 활성화합니다.
3. OpenSearch Ingestion에서 사용할 프라이빗 주소 공간의 VPC CIDR을 설정합니다.
4. 를 사용하여 Amazon DocumentDB 클러스터에 대한 인증을 설정합니다 AWS Secrets Manager. Amazon DocumentDB의 암호 자동 교체 단계에 따라 보안 암호 교체를 활성화합니다. 자세한 내용은 Amazon DocumentDB의 역할 기반 액세스 제어 및 보안을 사용한 데이터베이스 액세스를 참조하세요.
5. 변경 스트림을 사용하여 Amazon DocumentDB 컬렉션의 데이터 변경 사항을 구독하는 경우 `change_stream_log_retention_duration` 파라미터를 사용하여 보존 기간을 최대 7일로 연장해 데이터 손실을 방지합니다. 변경 스트림 이벤트는 이벤트가 기록된 후 기본적으로 3시간 동안 저장됩니다.단, 대규모 수집에 충분한 시간이 아닙니다. 변경 스트림 보존 기간을 수정하려면 변경 스트림 로그 보존 기간 수정을 참조하세요.

6. OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성합니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 및 the section called “컬렉션 생성” 섹션을 참조하세요.
7. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 Amazon DocumentDB 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

   다음 샘플 도메인 액세스 정책은 다음 단계에서 생성하는 파이프라인 역할을 사용하여 도메인에 데이터를 쓰도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   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"
      ]
    }
  ]
}
```

   컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 섹션을 참조하세요the section called “역할 및 사용자 설정”.

1단계: 파이프라인 역할 구성

Amazon DocumentDB 파이프라인 사전 조건을 설정한 후 파이프라인 구성에서 사용하려는 파이프라인 역할을 구성하고 역할에서 다음 Amazon DocumentDB 권한을 추가합니다.

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"
                }
            }
        }
    ]
}
```

파이프라인은 이러한 권한을 사용하여 VPC에서 네트워크 인터페이스를 생성하고 삭제하므로 OpenSearch Ingestion 파이프라인을 생성하는 데 사용하는 IAM 역할에서 위의 Amazon EC2 권한을 제공해야 합니다. 파이프라인은 오직 이 네트워크 인터페이스를 통해 Amazon DocumentDB 클러스터에 액세스할 수 있습니다.

2단계: 파이프라인 생성

이제 Amazon DocumentDB를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다. 인덱스 이름을 채우기 위해 `getMetadata` 함수는 *`documentdb_collection`*을 메타데이터 키로 사용합니다. `getMetadata` 메서드 없이 다른 인덱스 이름을 사용하려면 `index: "`*`my_index_name`*`"` 구성을 사용할 수 있습니다.

```
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
```

사전 구성된 Amazon DocumentDB 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 Amazon DocumentDB를 소스로 사용하려면 VPC에 파이프라인도 연결해야 합니다. 이렇게 하려면 소스 네트워크 옵션 섹션을 찾아 VPC에 연결 확인란을 선택하고 제공된 기본 옵션 중 하나에서 CIDR을 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.

사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 자Amazon DocumentDB 간의 IP 주소 충돌을 방지하려면 Amazon DocumentDB VPC CIDR이 OpenSearch Ingestion의 CIDR과 달라야 합니다.

자세한 내용은 파이프라인에 대한 VPC 액세스 구성을 참조하세요.

데이터 일관성

파이프라인은 Amazon DocumentDB 클러스터의 변경 사항을 지속적으로 폴링하거나 수신하고 OpenSearch 인덱스에서 해당 문서를 업데이트하여 데이터 일관성을 보장합니다.

OpenSearch Ingestion은 데이터 내구성을 보장하는 엔드 투 엔드 승인을 지원합니다. 파이프라인에서는 스냅샷이나 스트림을 읽을 때 병렬 처리를 위해 동적으로 분할을 생성합니다. 파이프라인에서는 OpenSearch 도메인이나 컬렉션에서 모든 레코드를 수집한 후 승인을 받으면 분할을 완료 상태로 표시합니다.

OpenSearch Serverless 검색 컬렉션에 수집하려는 경우 파이프라인에서 문서 ID를 생성할 수 있습니다. OpenSearch Serverless 시계열 컬렉션으로 수집하려는 경우에는 파이프라인에서 문서 ID를 생성하지 않습니다. 따라서 파이프라인 싱크 구성에서 `document_id: "${getMetadata(\"primary_key\")}"`를 생략해야 합니다.

또한 OpenSearch Ingestion 파이프라인에서는 수신 이벤트 작업을 해당하는 대량 인덱싱 작업에 매핑하여 문서를 쉽게 수집할 수 있게 합니다. 이렇게 하면 데이터 일관성이 유지되므로 Amazon DocumentDB의 모든 데이터 변경 사항이 OpenSearch에서 해당하는 문서 변경 사항으로 조정됩니다.

데이터 형식 매핑

OpenSearch Service에서는 각 수신 문서의 데이터 형식을 Amazon DocumentDB의 해당 데이터 형식에 동적으로 매핑합니다. 다음 표에서는 OpenSearch Service에서 다양한 데이터 형식을 자동으로 매핑하는 방법을 보여줍니다.

데이터 유형	OpenSearch	Amazon DocumentDB
Integer	OpenSearch는 Amazon DocumentDB 정수 값을 OpenSearch 정수에 자동으로 매핑합니다. OpenSearch에서는 처음 보낸 문서를 기반으로 필드를 동적으로 매핑합니다. Amazon DocumentDB에서 동일한 속성에 대해 데이터 유형을 혼합한 경우 자동 매핑이 실패할 수 있습니다. 예를 들어 첫 번째 문서에는 long인 속성이 있고 이후 문서에는 동일한 속성이 정수인 경우 OpenSearch는 두 번째 문서를 수집하지 못합니다. 이러한 경우에는 다음과 같이 보다 유연한 숫자 유형을 선택하는 명시적 매핑 템플릿을 제공해야 합니다. `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB는 정수를 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
Long	OpenSearch는 Amazon DocumentDB long 값을 OpenSearch long에 자동으로 매핑합니다. OpenSearch에서는 처음 보낸 문서를 기반으로 필드를 동적으로 매핑합니다. Amazon DocumentDB에서 동일한 속성에 대해 데이터 유형을 혼합한 경우 자동 매핑이 실패할 수 있습니다. 예를 들어 첫 번째 문서에는 long인 속성이 있고 이후 문서에는 동일한 속성이 정수인 경우 OpenSearch는 두 번째 문서를 수집하지 못합니다. 이러한 경우에는 다음과 같이 보다 유연한 숫자 유형을 선택하는 명시적 매핑 템플릿을 제공해야 합니다. `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB는 longs를 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
String	OpenSearch에서는 문자열 값을 자동으로 텍스트에 매핑합니다. 열거된 값과 같은 일부 상황에서는 키워드 형식에 매핑할 수 있습니다. 다음 예제에서는 이름이 `PartType`인 Amazon DocumentDB 속성을 OpenSearch 키워드에 매핑하는 방법을 보여줍니다. `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	Amazon DocumentDB는 문자열을 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
배정밀도 실수	OpenSearch는 Amazon DocumentDB double 값을 OpenSearch double 값에 자동으로 매핑합니다. OpenSearch에서는 처음 보낸 문서를 기반으로 필드를 동적으로 매핑합니다. Amazon DocumentDB에서 동일한 속성에 대해 데이터 유형을 혼합한 경우 자동 매핑이 실패할 수 있습니다. 예를 들어 첫 번째 문서에는 long인 속성이 있고 이후 문서에는 동일한 속성이 정수인 경우 OpenSearch는 두 번째 문서를 수집하지 못합니다. 이러한 경우에는 다음과 같이 보다 유연한 숫자 유형을 선택하는 명시적 매핑 템플릿을 제공해야 합니다. `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB는 doubles를 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
날짜	기본적으로 날짜는 OpenSearch의 정수에 매핑됩니다. 사용자 지정 매핑 템플릿을 정의하여 날짜를 OpenSearch 날짜에 매핑할 수 있습니다. `{ "template": { "mappings": { "properties": { "myDateField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB는 날짜를 지원합니다.
Timestamp	기본적으로 타임스탬프는 OpenSearch의 정수에 매핑됩니다. 사용자 지정 매핑 템플릿을 정의하여 날짜를 OpenSearch 날짜에 매핑할 수 있습니다. `{ "template": { "mappings": { "properties": { "myTimestampField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB는 타임스탬프를 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
불	OpenSearch에서는 Amazon DocumentDB 부울 유형을 OpenSearch 부울 유형에 매핑합니다.	Amazon DocumentDB는 부울 유형 속성을 지원합니다.
10진수	OpenSearch는 Amazon DocumentDB 맵 속성을 중첩 필드에 매핑합니다. 중첩 필드 내에도 동일한 매핑이 적용됩니다. 다음 예제에서는 중첩 필드의 문자열을 OpenSearch의 키워드 형식에 매핑합니다. `{ "template": { "mappings": { "properties": { "myDecimalField": { "type": "double" } } } } }` 이 사용자 지정 매핑을 사용하면 double 수준의 정밀도로 필드를 쿼리하고 집계할 수 있습니다. 원래 값은 OpenSearch 문서의 `_source` 속성에서 전체 정밀도를 유지합니다. 이 매핑이 없으면 OpenSearch는 기본적으로 텍스트를 사용합니다.	Amazon DocumentDB는 소수를 지원합니다.
정규식	정규식 유형은 중첩 필드를 생성합니다. 여기에는 *<myFieldName>* `.pattern` 및 *<myFieldName>* `.options`가 포함됩니다.	Amazon DocumentDB는 정규식을 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
이진 데이터	OpenSearch는 Amazon DocumentDB 바이너리 데이터를 OpenSearch 텍스트에 자동으로 매핑합니다. OpenSearch에서 이러한 데이터를 바이너리 필드로 작성하기 위한 매핑을 제공할 수 있습니다. 다음 예제에서는 이름이 `imageData` 인 Amazon DocumentDB 필드를 OpenSearch 바이너리 필드에 매핑하는 방법을 보여줍니다. `{ "template": { "mappings": { "properties": { "imageData": { "type": "binary" } } } } }`	Amazon DocumentDB는 바이너리 데이터 필드를 지원합니다.
ObjectId	objectId 유형의 필드는 OpenSearch 텍스트 필드에 매핑됩니다. 값은 objectId의 문자열 표현입니다.	Amazon DocumentDB는 objectIds를 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
Null	OpenSearch는 Amazon DocumentDB null 유형의 문서를 수집할 수 있습니다. 값을 문서에 null 값으로 저장합니다. 이 형식에는 매핑이 없으며 이 필드는 인덱싱되거나 검색할 수 없습니다. 한 속성 이름을 null 형식에 사용한 후 나중에 문자열과 같은 다른 형식으로 변경하면 OpenSearch에서 null이 아닌 첫 번째 값에 대해 동적 매핑을 생성합니다. 후속 값은 여전히 Amazon DocumentDB null 값일 수 있습니다.	Amazon DocumentDB는 null 유형 필드를 지원합니다.
정의되지 않음	OpenSearch는 Amazon DocumentDB 정의되지 않은 유형의 문서를 수집할 수 있습니다. 값을 문서에 null 값으로 저장합니다. 이 형식에는 매핑이 없으며 이 필드는 인덱싱되거나 검색할 수 없습니다. 정의되지 않은 유형에 동일한 필드 이름을 사용한 후 문자열과 같이 다른 유형으로 변경하면 OpenSearch에서 정의되지 않은 값이 아닌 첫 번째 값에 대해 동적 매핑을 생성합니다. 후속 값은 여전히 Amazon DocumentDB 정의되지 않은 값일 수 있습니다.	Amazon DocumentDB는 정의되지 않은 유형 필드를 지원합니다.

데이터 유형	OpenSearch	Amazon DocumentDB
MinKey	OpenSearch는 Amazon DocumentDB minKey 유형의 문서를 수집할 수 있습니다. 값을 문서에 null 값으로 저장합니다. 이 형식에는 매핑이 없으며 이 필드는 인덱싱되거나 검색할 수 없습니다. minKey 유형에 동일한 필드 이름을 사용한 후 문자열과 같이 다른 유형으로 변경하면 OpenSearch에서 minKey가 아닌 첫 번째 값에 대해 동적 매핑을 생성합니다. 후속 값은 여전히 Amazon DocumentDB minKey 값일 수 있습니다.	Amazon DocumentDB는 minKey 유형 필드를 지원합니다.
MaxKey	OpenSearch는 Amazon DocumentDB maxKey 유형의 문서를 수집할 수 있습니다. 값을 문서에 null 값으로 저장합니다. 이 형식에는 매핑이 없으며 이 필드는 인덱싱되거나 검색할 수 없습니다. maxKey 유형에 동일한 필드 이름을 사용한 후 문자열과 같이 다른 유형으로 변경하면 OpenSearch에서 maxKey가 아닌 첫 번째 값에 대해 동적 매핑을 생성합니다. 후속 값은 여전히 Amazon DocumentDB maxKey 값일 수 있습니다.	Amazon DocumentDB는 maxKey 유형 필드를 지원합니다.

OpenSearch Ingestion 파이프라인에서 DLQ(Dead Letter Queue)를 구성하는 것이 좋습니다. 이 대기열을 구성하면 OpenSearch Service에서는 동적 매핑 실패로 인해 수집할 수 없는 모든 실패한 문서를 대기열로 전송합니다.

자동 매핑이 실패할 경우 파이프라인 구성에서 `template_type` 및 `template_content`를 사용하여 명시적 매핑 규칙을 정의할 수 있습니다. 또는 파이프라인을 시작하기 전에 검색 도메인이나 컬렉션에서 직접 매핑 템플릿을 생성할 수도 있습니다.

제한 사항

Amazon DocumentDB용 OpenSearch Ingestion 파이프라인을 설정하는 경우 다음과 같은 제한 사항을 고려합니다.

- Amazon DocumentDB와의 OpenSearch Ingestion 통합 기능에서는 현재 교차 리전 수집을 지원하지 않습니다. Amazon DocumentDB 클러스터와 OpenSearch Ingestion 파이프라인은 동일한 AWS 리전에 있어야 합니다.
- Amazon DocumentDB와의 OpenSearch Ingestion 통합 기능에서는 현재 교차 계정 수집을 지원하지 않습니다. Amazon DocumentDB 클러스터와 OpenSearch Ingestion 파이프라인은 동일한 AWS 계정에 있어야 합니다.
- OpenSearch Ingestion 파이프라인은 단일 Amazon DocumentDB 테이블만 소스로 지원합니다.
- Amazon DocumentDB와의 OpenSearch Ingestion 통합은 특별히 Amazon DocumentDB 인스턴스 기반 클러스터를 지원합니다. Amazon DocumentDB 탄력적 클러스터는 지원하지 않습니다.
- OpenSearch Ingestion 통합은 Amazon DocumentDB 클러스터의 인증 메커니즘 AWS Secrets Manager 으로 만 지원합니다.
- 다른 데이터베이스 또는 컬렉션에서 데이터를 수집하도록 기존 파이프라인 구성을 업데이트할 수 없습니다. 대신 새 파이프라인을 생성해야 합니다.

권장되는 CloudWatch 경보

최상의 성능을 위해 다음 Amazon DocumentDB 클러스터에 소스로 액세스하도록 OpenSearch Ingestion 파이프라인을 생성할 때 다음 CloudWatch 경보를 사용하는 것이 좋습니다.

CloudWatch 경보	설명
<pipeline-name> .doucmentdb.credentialsChanged	이 지표는 AWS 보안 암호 교체 빈도를 나타냅니다.
<pipeline-name> .doucmentdb.executorRefreshErrors	이 지표는 AWS 시크릿의 새로 고침 실패를 나타냅니다.
<pipeline-name> .doucmentdb.exportRecordsTotal	이 지표는 Amazon DocumentDB에서 내보낸 레코드 수를 나타냅니다.
<pipeline-name> .doucmentdb.exportRecordsProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 레코드 수를 나타냅니다.

CloudWatch 경보	설명
<pipeline-name> .doucmentdb.exportRecordProcessingErrors	이 지표는 Amazon DocumentDB 클러스터에서 데이터를 읽는 동안 OpenSearch Ingestion 파이프라인에서 처리한 오류 수를 나타냅니다.
<pipeline-name> .doucmentdb.exportRecordsSuccessTotal	이 지표는 성공적으로 처리한 내보내기 레코드의 총 수를 나타냅니다.
<pipeline-name> .doucmentdb.exportRecordsFailedTotal	이 지표는 처리하지 못한 내보내기 레코드의 총 수를 나타냅니다.
<pipeline-name> .doucmentdb.bytesReceived	이 지표는 OpenSearch Ingestion 파이프라인에서 수신한 총 바이트 수를 나타냅니다.
<pipeline-name> .doucmentdb.bytesProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 총 바이트 수를 나타냅니다.
<pipeline-name> .doucmentdb.exportPartitionQueryTotal	이 지표는 내보내기 파티션 합계를 나타냅니다.
<pipeline-name> .doucmentdb.streamRecordsSuccessTotal	이 지표는 스트림에서 성공적으로 처리한 레코드 수를 나타냅니다.
<pipeline-name> .doucmentdb.streamRecordsFailedTotal	이 지표는 스트림에서 처리하지 못한 총 레코드 수를 나타냅니다.

Confluent Cloud Kafka에서 OpenSearch Ingestion 파이프라인 사용

OpenSearch Ingestion 파이프라인을 사용하여 Confluent Cloud Kafka 클러스터에서 Amazon OpenSearch Service 도메인 및 OpenSearch Serverless 컬렉션으로 데이터를 스트리밍할 수 있습니다. OpenSearch Ingestion은 Confluent Cloud Kafka 클러스터에서 OpenSearch Service 또는 OpenSearch Serverless가 관리하는 도메인 또는 컬렉션으로 데이터를 스트리밍하기 위한 퍼블릭 및 프라이빗 네트워크 구성을 모두 지원합니다.

Confluent Cloud 퍼블릭 Kafka 클러스터에 대한 연결

OpenSearch Ingestion 파이프라인을 사용하여 퍼블릭 구성을 통해 자체 관리형 Confluent Cloud Kafka 클러스터에서 데이터를 마이그레이션할 수 있습니다. 즉, 도메인 DNS 이름을 공개적으로 확

인할 수 있습니다. 이렇게 하려면 Confluent Cloud 퍼블릭 Kafka 클러스터를 소스로, OpenSearch Service 또는 OpenSearch Serverless를 대상으로 하는 OpenSearch Ingestion 파이프라인을 설정합니다. 이렇게 하면 자체 관리형 소스 클러스터에서 AWS관리형 대상 도메인 또는 컬렉션으로 스트리밍 데이터가 처리됩니다.

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. 소스 역할을 하는 Confluent Cloud Kafka 클러스터를 생성합니다. 클러스터에는 OpenSearch Service로 수집할 데이터가 포함되어 있어야 합니다.
2. 데이터를 마이그레이션할 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성합니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 및 the section called “컬렉션 생성” 섹션을 참조하세요.
3. AWS Secrets Manager를 사용하여 Confluent Cloud Kafka 클러스터에서 인증을 설정합니다. AWS Secrets Manager 시크릿 교체의 단계를 수행하여 시크릿 교체를 활성화합니다.
4. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 자체 관리형 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

   다음 샘플 도메인 액세스 정책은 다음 단계에서 생성하는 파이프라인 역할을 사용하여 도메인에 데이터를 쓰도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   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"
      ]
    }
```

```
  ]
}
```

컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 섹션을 참조하세요the section called “역할 및 사용자 설정”.

1단계: 파이프라인 역할 구성

Confluent Cloud Kafka 클러스터 파이프라인 사전 조건을 설정한 후 파이프라인 구성에 사용할 파이프라인 역할을 구성하고 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션에 쓸 수 있는 권한과 Secrets Manager에서 시크릿을 읽을 수 있는 권한을 추가합니다.

네트워크 인터페이스를 관리하려면 다음 권한이 필요합니다.

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:*::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" }
            }
        }
    ]
}
```

다음은 AWS Secrets Manager 서비스에서 보안 암호를 읽는 데 필요한 권한입니다.

JSON

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

Amazon OpenSearch Service 도메인에 쓰려면 다음 권한이 필요합니다.

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

2단계: 파이프라인 생성

이제 Confluent Cloud Kafka를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

여러 OpenSearch Service 도메인을 데이터의 대상으로 지정할 수 있습니다. 이 기능을 사용하여 조건부로 라우팅하거나 수신 데이터를 여러 OpenSearch Service 도메인으로 복제할 수 있습니다.

소스 Confluent Kafka의 데이터를 OpenSearch Serverless VPC 컬렉션으로 마이그레이션할 수도 있습니다. 파이프라인 구성 내에서 네트워크 액세스 정책을 제공해야 합니다. Confluent 스키마 레지스트리를 사용하여 Confluent 스키마를 정의할 수 있습니다.

```
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"
```

사전 구성된 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

VPC에서 Confluent Cloud Kafka 클러스터에 대한 연결

OpenSearch Ingestion 파이프라인을 사용하여 VPC에서 실행되는 Confluent Cloud Kafka 클러스터에서 데이터를 마이그레이션할 수도 있습니다. 이렇게 하려면 Confluent Cloud Kafka 클러스터를 소스로, OpenSearch Service 또는 OpenSearch Serverless를 대상으로 하는 OpenSearch Ingestion 파이프라인을 설정합니다. 이렇게 하면 Confluent Cloud Kafka 소스 클러스터에서 AWS관리형 대상 도메인 또는 컬렉션으로 스트리밍 데이터를 처리합니다.

OpenSearch Ingestion은 Confluent에서 지원되는 모든 네트워크 모드에서 구성된 Confluent Cloud Kafka 클러스터를 지원합니다. OpenSearch Ingestion에서 다음 네트워크 구성 모드가 소스로 지원됩니다.

- AWS VPC 피어링
- AWS PrivateLink 전용 클러스터용
- AWS PrivateLink 엔터프라이즈 클러스터용
- AWS Transit Gateway

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. OpenSearch Service로 수집하려는 데이터가 포함된 VPC 네트워크 구성을 사용하여 Confluent Cloud Kafka 클러스터를 생성합니다.
2. 데이터를 마이그레이션할 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성합니다. 자세한 내용은 및 단원을 참조하십시오the section called “ OpenSearch Service 도메인 생성”the section called “컬렉션 생성”.
3. AWS Secrets Manager를 사용하여 Confluent Cloud Kafka 클러스터에서 인증을 설정합니다. AWS Secrets Manager 시크릿 교체의 단계를 수행하여 시크릿 교체를 활성화합니다.
4. Confluent Cloud Kafka 클러스터에 액세스할 수 있는 VPC의 ID를 가져옵니다. OpenSearch Ingestion에서 사용할 VPC CIDR을 선택합니다.

> **Note**
>
> 를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 Confluent Cloud Kafka 클러스터를 사용하려면 OpenSearch Ingestion 파이프라인도 VPC에 연결해야 합니다. 이를 수행하려면 네트워크 구성 섹션을 찾아 VPC에 연결 확인란을 선택하고 제공된 기본 옵션 중 하나에서 CIDR을 선택하거나 직접 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.
> 사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 자체 관리형 OpenSearch 간의 IP 주소 충돌을 방지하려면 자체 관리형 OpenSearch VPC CIDR이 OpenSearch Ingestion의 CIDR과 달라야 합니다.

5. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 자체 관리형 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

> **Note**
>
> AWS PrivateLink 를 사용하여 Confluent Cloud Kafka를 연결하는 경우 VPC DHCP 옵션을 구성해야 합니다. DNS 호스트 이름과 DNS 확인이 활성화되어 있어야 합니다.
> 특히 다음 옵션 세트 값을 사용합니다.
>
> ```
> domain-name: aws.private.confluent.cloud
> domain-name-servers: AmazonProvidedDNS
> ```
>
> 이 변경 사항을 통해 Confluent PrivateLink 엔드포인트의 DNS 확인이 VPC 내에서 올바르게 작동합니다.

다음 샘플 도메인 액세스 정책은 다음 단계에서 생성하는 파이프라인 역할을 사용하여 도메인에 데이터를 쓰도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

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"
      ]
    }
  ]
}
```

컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 섹션을 참조하세요the section called “역할 및 사용자 설정”.

1단계: 파이프라인 역할 구성

파이프라인 사전 조건을 설정한 후 파이프라인 구성에서 사용하려는 파이프라인 역할을 구성하고 역할에서 다음 권한을 추가합니다.

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"
                    }
                }
            }
        ]
}
```

파이프라인은 이러한 권한을 사용하여 VPC에서 네트워크 인터페이스를 생성하고 삭제하므로 OpenSearch Ingestion 파이프라인을 생성하는 데 사용하는 IAM 역할에서 위의 Amazon EC2 권한을 제공해야 합니다. 파이프라인은 오직 이 네트워크 인터페이스를 통해 Kafka 클러스터에 액세스할 수 있습니다.

2단계: 파이프라인 생성

이제 Kafka를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

여러 OpenSearch Service 도메인을 데이터의 대상으로 지정할 수 있습니다. 이 기능을 사용하여 조건부로 라우팅하거나 수신 데이터를 여러 OpenSearch Service 도메인으로 복제할 수 있습니다.

소스 Confluent Kafka의 데이터를 OpenSearch Serverless VPC 컬렉션으로 마이그레이션할 수도 있습니다. 파이프라인 구성 내에서 네트워크 액세스 정책을 제공해야 합니다. Confluent 스키마 레지스트리를 사용하여 Confluent 스키마를 정의할 수 있습니다.

```
 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"
```

Amazon Managed Streaming for Apache Kafka와 함께 OpenSearch Ingestion 파이프라인 사용

Kafka 플러그인을 사용하여 Amazon Managed Streaming for Apache Kafka(Amazon MSK)에서 OpenSearch Ingestion 파이프라인으로 데이터를 수집할 수 있습니다. Amazon MSK로 Apache Kafka를 사용하여 스트리밍 데이터를 처리하는 애플리케이션을 구축하고 실행할 수 있습니다. OpenSearch Ingestion은 AWS PrivateLink 를 사용하여 Amazon MSK에 연결합니다. Amazon MSK 및 Amazon MSK Serverless 클러스터 모두에서 데이터를 수집할 수 있습니다. 두 프로세스의 유일한 차이는 파이프라인을 설정하기 전에 수행해야 하는 사전 조건 단계입니다.

주제

- 프로비저닝된 Amazon MSK 사전 조건
- Amazon MSK Serverless 사전 조건
- 1단계: 파이프라인 역할 구성
- 2단계: 파이프라인 생성
- 3단계: (선택 사항) AWS Glue 스키마 레지스트리 사용
- 4단계: (선택 사항) Amazon MSK 파이프라인의 권장 컴퓨팅 유닛(OCU) 구성

프로비저닝된 Amazon MSK 사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. Amazon Managed Streaming for Apache Kafka 개발자 안내서의 클러스터 생성에 나온 단계를 수행하여 Amazon MSK 클러스터를 생성합니다. 브로커 유형의 경우 OpenSearch Ingestion에서 지원되지 않으므로 `t3` 유형을 제외한 옵션을 선택합니다.
2. 클러스터가 활성 상태가 되면 다중 VPC 연결 켜기 단계를 따르세요.
3. 클러스터와 파이프라인이 동일한 AWS 계정에 있는지 여부에 따라 MSK 클러스터에 클러스터 정책 연결의 단계에 따라 다음 정책 중 하나를 연결합니다. 이 정책은 OpenSearch Ingestion이 Amazon MSK 클러스터에 대한 AWS PrivateLink 연결을 생성하고 Kafka 주제에서 데이터를 읽을 수 있도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   클러스터와 파이프라인이 동일한 AWS 계정에 있는 경우 다음 정책이 적용됩니다.

   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"
    }
  ]
}
```

Amazon MSK 클러스터가 파이프라인 AWS 계정 과 다른에 있는 경우 대신 다음 정책을 연결합니다. 교차 계정 액세스는 프로비저닝된 Amazon MSK 클러스터에서만 가능하며 Amazon MSK Serverless 클러스터에서는 불가능합니다. 의 AWS `principal` ARN은 파이프라인 구성에 제공하는 것과 동일한 파이프라인 역할에 대한 ARN이어야 합니다.

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. 주제 생성의 단계에 따라 Kafka 주제를 생성하세요. *BootstrapServerString*이 프라이빗 엔드포인트(단일 VPC) 부트스트랩 URL 중 하나인지 확인하세요. `--replication-factor`의 값은 Amazon MSK 클러스터의 영역 수에 따라 2 또는 3이어야 합니다. `--partitions`의 값은 최소 `10` 이상이어야 합니다.
5. 데이터 생산 및 소비의 단계에 따라 데이터를 생산하고 소비하세요. 다시, *BootstrapServerString*이 프라이빗 엔드포인트(단일 VPC) 부트스트랩 URL 중 하나인지 확인하세요.

Amazon MSK Serverless 사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. Amazon Managed Streaming for Apache Kafka 개발자 안내서의 클러스터 생성에 있는 단계에 따라 Amazon MSK Serverless를 생성합니다.
2. 클러스터가 활성 상태이면 Attach a cluster policy to the MSK cluster에 나온 단계를 수행하여 다음 정책을 연결합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   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"
    }
  ]
}
```

이 정책은 OpenSearch Ingestion이 Amazon MSK Serverless 클러스터에 대한 AWS PrivateLink 연결을 생성하고 Kafka 주제에서 데이터를 읽을 수 있도록 허용합니다. 이 정책은 클러스터와 파이프라인이 동일한에 있는 경우 적용되며 AWS 계정, Amazon MSK Serverless는 교차 계정 액세스를 지원하지 않으므로 true여야 합니다.

3. 주제 생성의 단계에 따라 Kafka 주제를 생성하세요. *BootstrapServerString*이 Simple Authentication and Security Layer(SASL) IAM 부트스트랩 URL 중 하나인지 확인합니다. `--replication-factor`의 값은 Amazon MSK Serverless 클러스터의 영역 수에 따라 2 또는 3이어야 합니다. `--partitions`의 값은 최소 10 이상이어야 합니다.
4. 데이터 생산 및 소비의 단계에 따라 데이터를 생산하고 소비하세요. 다시 *BootstrapServerString*이 Simple Authentication and Security Layer(SASL) IAM 부트스트랩 URL 중 하나인지 확인합니다.

1단계: 파이프라인 역할 구성

Amazon MSK 프로비저닝 또는 서버리스 클러스터를 설정한 후 파이프라인 구성에서 사용하려는 파이프라인 역할에 다음 Kafka 권한을 추가합니다.

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

```
            ]
        }
    ]
}
```

2단계: 파이프라인 생성

그런 다음 Kafka를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

```
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
```

사전 구성된 Amazon MSK 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

3단계: (선택 사항) AWS Glue 스키마 레지스트리 사용

Amazon MSK에서 OpenSearch Ingestion을 사용하는 경우 AWS Glue 스키마 레지스트리에서 호스팅되는 스키마에 AVRO 데이터 형식을 사용할 수 있습니다. AWS Glue 스키마 레지스트리를 사용하면 데이터 스트림 스키마를 중앙에서 검색, 제어 및 발전시킬 수 있습니다.

이 옵션을 사용하려면 파이프라인 구성에서 스키마 `type`를 활성화하세요.

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

또한 파이프라인 역할에 AWS Glue 읽기 액세스 권한을 제공해야 합니다. AWSGlueSchemaRegistryReadonlyAccess라는 AWS 관리형 정책을 사용할 수 있습니다. 또한 레지스트리는 OpenSearch Ingestion 파이프라인과 동일한 AWS 계정 및 리전에 있어야 합니다.

4단계: (선택 사항) Amazon MSK 파이프라인의 권장 컴퓨팅 유닛(OCU) 구성

각 컴퓨팅 유닛에는 주제당 한 명의 소비자가 있습니다. 브로커는 특정 주제에 대해 이러한 소비자 간의 파티션을 조정합니다. 하지만 파티션 수가 소비자 수보다 많을 경우 Amazon MSK는 모든 소비자에게 여러 파티션을 호스팅합니다. OpenSearch Ingestion에는 CPU 사용량 또는 파이프라인에서 보류 중인 레코드 수에 따라 규모를 늘리거나 줄일 수 있는 auto-scaling 기능이 내장되어 있습니다.

성능을 최적화하려면 여러 컴퓨팅 유닛에 파티션을 분산하여 병렬 처리하세요. 주제에 많은 수의 파티션이 있는 경우(예: 파이프라인당 최대 OCU인 96개 초과), 1~96개의 OCU로 파이프라인을 구성하는 것이 좋습니다. 필요에 따라 자동으로 크기가 조정되기 때문입니다. 주제의 파티션 수가 적은 경우(예: 96개 미만), 최대 컴퓨팅 유닛을 파티션 수와 동일하게 유지하세요.

파이프라인에 주제가 한 개 이상 있는 경우 파티션 수가 가장 많은 주제를 참조로 선택하여 최대 컴퓨팅 유닛을 구성하세요. 새 OCU 세트가 포함된 다른 파이프라인을 동일한 주제 및 소비자 그룹에 추가하면 처리량을 거의 선형적으로 확장할 수 있습니다.

Amazon RDS에서 OpenSearch Ingestion 파이프라인 사용

Amazon RDS와 함께 OpenSearch Ingestion 파이프라인을 사용하여 기존 데이터를 내보내고 변경 사항(예: 생성, 업데이트 및 삭제)을 Amazon OpenSearch Service 도메인 및 컬렉션으로 스트리밍할 수 있습니다. OpenSearch Ingestion 파이프라인은 변경 데이터 캡처(CDC) 인프라를 통합하여 Amazon RDS에서 데이터를 지속적으로 스트리밍할 수 있는 지연 시간이 짧은 대규모 방법을 제공합니다. RDS for MySQL 및 RDS for PostgreSQL이 지원됩니다.

Amazon RDS를 소스로 사용하여 전체 초기 스냅샷을 사용하거나 사용하지 않고 데이터를 처리할 수 있는 두 가지 방법이 있습니다. 전체 초기 스냅샷은 지정된 테이블의 스냅샷이며이 스냅샷은 Amazon S3로 내보내집니다. 그러면 OpenSearch Ingestion 파이프라인이 이 스냅샷을 도메인의 한 인덱스로 보내거나 분할하여 도메인의 여러 인덱스로 보냅니다. Amazon RDS 및 OpenSearch의 데이터를 일관되게 유지하기 위해 파이프라인은 Amazon RDS 인스턴스의 테이블에 있는 모든 생성, 업데이트 및 삭제 이벤트를 OpenSearch 인덱스 또는 인덱스에 저장된 문서와 동기화합니다.

전체 초기 스냅샷을 사용하면 OpenSearch Ingestion 파이프라인이 먼저 스냅샷을 수집한 다음 Amazon RDS 변경 스트림에서 데이터 읽기를 시작합니다. 결국 Amazon RDS와 OpenSearch 간에 거의 실시간에 가까운 데이터 일관성을 유지하고 따라잡습니다.

또한 Amazon RDS와의 OpenSearch Ingestion 통합을 사용하여 변경 데이터 캡처를 추적하고 Aurora의 모든 업데이트를 OpenSearch로 수집할 수 있습니다. 다른 메커니즘의 전체 스냅샷이 이미 있거나 Amazon RDS 인스턴스의 데이터에 대한 모든 변경 사항을 캡처하려는 경우이 옵션을 선택합니다.

이 옵션을 선택하면 Amazon RDS for MySQL 바이너리 로깅을 구성하거나 Amazon RDS for PostgresSQL DB 인스턴스에 대한 논리적 복제를 설정해야 합니다.

주제

- RDS for MySQL
- RDS for PostgreSQL

RDS for MySQL

Amazon RDS for RDS for MySQL을 사용하여 OpenSearch Ingestion 파이프라인을 구성하려면 다음 단계를 완료하세요.

주제

- RDS for MySQL 사전 조건
- 1단계: 파이프라인 역할 구성
- 2단계: 파이프라인 생성
- 데이터 일관성
- 데이터 형식 매핑
- 제한 사항
- 권장되는 CloudWatch 경보

RDS for MySQL 사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. Amazon RDS에서 사용자 지정 DB 파라미터 그룹을 생성하여 바이너리 로깅을 구성하고 다음 파라미터를 설정합니다.

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

   또한 `binlog_row_value_options` 파라미터가 로 설정되지 않았는지 확인합니다PARTIAL_JSON.

   자세한 내용은 RDS for MySQL 바이너리 로깅 구성을 참조하세요.
2. RDS for MySQL DB 인스턴스를 선택하거나 생성하고 이전 단계에서 생성한 파라미터 그룹을 DB 인스턴스와 연결합니다.
3. 데이터베이스에서 자동 백업이 활성화되어 있는지 확인합니다. 자세한 내용은 자동 백업 활성화를 참조하세요.
4. 예를 들어 24시간과 같이 복제에 충분한 시간을 두고 이진 로그 보존을 구성합니다. 자세한 내용은 Amazon RDS 사용 설명서의 이진 로그 구성 설정 및 표시를 참조하세요.
5. Amazon RDS 및에서 암호 관리를 사용하여 Amazon RDS 인스턴스에 사용자 이름과 암호 AWS Secrets Manager 인증을 설정합니다. Secrets Manager 보안 암호를 생성하여 사용자 이름/암호 조합을 생성할 수도 있습니다.
6. 전체 초기 스냅샷 기능을 사용하는 경우 Amazon RDS에서 Amazon S3로 데이터를 내보내기 위한 AWS KMS key 및 IAM 역할을 생성합니다.

   IAM 역할에는 다음과 같은 권한 정책이 있어야 합니다.

   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/*"
            ]
        }
    ]
}
```

역할에는 다음과 같은 신뢰 관계도 있어야 합니다.

JSON

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

7. OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 선택하거나 생성합니다. 자세한 내용은 OpenSearch Service 도메인 생성 및 컬렉션 생성을 참조하세요.
8. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 Amazon RDS DB 인스턴스의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

1단계: 파이프라인 역할 구성

Amazon RDS 파이프라인 사전 조건을 설정한 후 파이프라인 구성에 사용할 파이프라인 역할을 구성합니다. 또한 Amazon RDS 소스에 대한 다음 권한을 역할에 추가합니다.

```
{
  "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:*:account-id:network-interface/*",
        "arn:aws:ec2:*:account-id:subnet/*",
        "arn:aws:ec2:*:account-id:security-group/*"
      ]
    },
    {
      "Sid": "allowDescribeEC2",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*"
```

```
    },
    {
      "Sid": "allowTagCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:account-id:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/OSISManaged": "true"
        }
      }
    },
    {
      "Sid": "AllowDescribeInstances",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:db:*"
      ]
    },
     {
      "Sid": "AllowSnapshots",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBSnapshots",
        "rds:CreateDBSnapshot",
        "rds:AddTagsToResource"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:db:DB-id",
        "arn:aws:rds:region:account-id:snapshot:DB-id*"
      ]
    },
    {
      "Sid": "AllowExport",
      "Effect": "Allow",
      "Action": [
        "rds:StartExportTask"
      ],
      "Resource": [
```

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

```
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:*:account-id:secret:*"
      ]
    }
  ]
}
```

2단계: 파이프라인 생성

다음과 유사한 OpenSearch Ingestion 파이프라인을 구성합니다. 예제 파이프라인은 Amazon RDS 인스턴스를 소스로 지정합니다.

```
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
```

미리 구성된 Amazon RDS 블루프린트를 사용하여이 파이프라인을 생성할 수 있습니다. 자세한 내용은 블루프린트 작업 단원을 참조하십시오.

Amazon Aurora를 소스로 사용하려면 파이프라인에 대한 VPC 액세스를 구성해야 합니다. 선택한 VPC는 Amazon Aurora 소스에서 사용하는 것과 동일한 VPC여야 합니다. 그런 다음 하나 이상의 서브넷과 하나 이상의 VPC 보안 그룹을 선택합니다. 파이프라인에는 Aurora MySQL 데이터베이스에 대한 네트워크 액세스가 필요하므로 Aurora 클러스터가 파이프라인의 VPC 보안 그룹에서 데이터베이스 포트로의 인바운드 트래픽을 허용하는 VPC 보안 그룹으로 구성되어 있는지도 확인해야 합니다. 자세한 내용은 보안 그룹을 사용한 액세스 제어를 참조하세요.

를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 Amazon Aurora를 소스로 사용하려면 VPC에 파이프라인도 연결해야 합니다. 이렇게 하려면 네트워크 구성 섹션을 찾아 VPC에 연결을 선택하고 제공된 기본 옵션 중 하나에서 CIDR을 선택하거나 직접 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.

사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 Amazon RDS 간의 IP 주소 충돌을 방지하려면 Amazon RDS VPC CIDR이 OpenSearch Ingestion용 CIDR과 다른지 확인합니다.

자세한 내용은 파이프라인에 대한 VPC 액세스 구성을 참조하세요.

데이터 일관성

파이프라인은 Amazon RDS 인스턴스의 변경 사항을 지속적으로 폴링하거나 수신하고 OpenSearch 인덱스에서 해당 문서를 업데이트하여 데이터 일관성을 보장합니다.

OpenSearch Ingestion은 데이터 내구성을 보장하는 엔드 투 엔드 승인을 지원합니다. 파이프라인에서는 스냅샷이나 스트림을 읽을 때 병렬 처리를 위해 동적으로 분할을 생성합니다. 파이프라인에서는 OpenSearch 도메인이나 컬렉션에서 모든 레코드를 수집한 후 승인을 받으면 분할을 완료 상태로 표시합니다. OpenSearch Serverless 검색 컬렉션에 수집하려는 경우 파이프라인에서 문서 ID를 생성할 수 있습니다. OpenSearch Serverless 시계열 컬렉션에 수집하려면 파이프라인이 문서 ID를 생성하지 않으므로 파이프라인 싱크 구성`document_id: "${getMetadata(\"primary_key\")}"`에서를 생략해야 합니다.

또한 OpenSearch Ingestion 파이프라인에서는 수신 이벤트 작업을 해당하는 대량 인덱싱 작업에 매핑하여 문서를 쉽게 수집할 수 있게 합니다. 이렇게 하면 데이터가 일관되게 유지되므로 Amazon RDS의 모든 데이터 변경 사항이 OpenSearch의 해당 문서 변경 사항과 조정됩니다.

데이터 형식 매핑

OpenSearch Ingestion 파이프라인은 MySQL 데이터 유형을 OpenSearch Service 도메인 또는 컬렉션이 사용할 수 있는 표현에 매핑합니다. OpenSearch에 매핑 템플릿이 정의되지 않은 경우 OpenSearch는 처음 전송된 문서를 기반으로 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 매핑 템플릿을 통해 OpenSearch에서 가장 적합한 필드 유형을 명시적으로 정의할 수도 있습니다.

아래 표에는 MySQL 데이터 유형과 해당 OpenSearch 필드 유형이 나열되어 있습니다. 명시적 매핑이 정의되지 않은 경우 기본 OpenSearch 필드 유형 열에 OpenSearch의 해당 필드 유형이 표시됩니다. 이 경우 OpenSearch는 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 권장 OpenSearch 필드 유형 열은 매핑 템플릿에서 명시적으로 지정하는 데 권장되는 해당 필드 유형입니다. 이러한 필드 유형은 MySQL의 데이터 유형과 더 밀접하게 정렬되며 일반적으로 OpenSearch에서 더 나은 검색 기능을 사용할 수 있습니다.

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
BIGINT	long	long
BIGINT UNSIGNED	long	서명되지 않은 길이
BIT	long	비트 수에 따라 바이트, 짧음, 정수 또는 깁니다.
DECIMAL	텍스트	double 또는 키워드

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
DOUBLE	float	double
FLOAT	float	float
INT	long	정수
INT UNSIGNED	long	long
MEDIUMIN	long	정수
MEDIUMIN UNSIGNED	long	정수
NUMERIC	텍스트	double 또는 키워드
SMALLINT	long	bigint
SMALLINT UNSIGNED	long	정수
TINYINT	long	바이트
TINYINT UNSIGNED	long	bigint
BINARY	텍스트	이진수
BLOB	텍스트	이진수
CHAR	텍스트	텍스트
ENUM	텍스트	키워드
LONGBLOE	텍스트	이진수
LONGTEXT	텍스트	텍스트

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
MEDIUMBL B	텍스트	이진수
MEDIUMTE T	텍스트	텍스트
SET	텍스트	키워드
TEXT	텍스트	텍스트
TINYBLOB	텍스트	이진수
TINYTEXT	텍스트	텍스트
VARBINAR	텍스트	이진수
VARCHAR	텍스트	텍스트
날짜	long(에포크 밀리초)	date
DATETIME	long(에포크 밀리초)	date
TIME	long(에포크 밀리초)	date
TIMESTAM	long(에포크 밀리초)	date
YEAR	long(에포크 밀리초)	date
GEOMETR	text(WKT 형식)	geo_shape
GEOMETR OLLECTIO	text(WKT 형식)	geo_shape
LINESTRIN G	text(WKT 형식)	geo_shape

MySQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
MULTILINESTRING	text(WKT 형식)	geo_shape
MULTIPOINT	text(WKT 형식)	geo_shape
MULTIPOLYGON	text(WKT 형식)	geo_shape
POINT	text(WKT 형식)	geo_point 또는 geo_shape
POLYGON	text(WKT 형식)	geo_shape
JSON	텍스트	객체

OpenSearch Ingestion 파이프라인에서 DLQ(Dead Letter Queue)를 구성하는 것이 좋습니다. 이 대기열을 구성하면 OpenSearch Service에서는 동적 매핑 실패로 인해 수집할 수 없는 모든 실패한 문서를 대기열로 전송합니다.

자동 매핑이 실패하면 파이프라인 구성`template_content`에서 `template_type` 및를 사용하여 명시적 매핑 규칙을 정의할 수 있습니다. 또는 파이프라인을 시작하기 전에 검색 도메인이나 컬렉션에서 직접 매핑 템플릿을 생성할 수도 있습니다.

제한 사항

RDS for MySQL용 OpenSearch Ingestion 파이프라인을 설정할 때 다음 제한 사항을 고려하세요.

- 통합은 파이프라인당 하나의 MySQL 데이터베이스만 지원합니다.
- 통합은 현재 리전 간 데이터 수집을 지원하지 않습니다. Amazon RDS 인스턴스와 OpenSearch 도메인은 동일한에 있어야 합니다 AWS 리전.
- 통합은 현재 교차 계정 데이터 수집을 지원하지 않습니다. Amazon RDS 인스턴스와 OpenSearch Ingestion 파이프라인은 동일한에 있어야 합니다 AWS 계정.
- 지원되는 유일한 인증 메커니즘인 Secrets Manager를 사용하여 Amazon RDS 인스턴스에 인증이 활성화되어 있는지 확인합니다.

- 다른 데이터베이스 및/또는 다른 테이블에서 데이터를 수집하도록 기존 파이프라인 구성을 업데이트할 수 없습니다. 파이프라인의 데이터베이스 및/또는 테이블 이름을 업데이트하려면 새 파이프라인을 생성해야 합니다.
- 데이터 정의 언어(DDL) 문은 일반적으로 지원되지 않습니다. 다음과 같은 경우에는 데이터 일관성이 유지되지 않습니다.
  - 기본 키가 변경됩니다(add/delete/rename).
  - 테이블이 삭제/절단됩니다.
  - 열 이름 또는 데이터 형식이 변경됩니다.
- 동기화할 MySQL 테이블에 기본 키가 정의되지 않은 경우 데이터 일관성이 보장되지 않습니다. 업데이트/삭제를 OpenSearch에 동기화하려면 OpenSearch 싱크 구성에서 사용자 지정 `document_id` 옵션을 올바르게 정의해야 합니다.
- 계단식 삭제 작업이 있는 외래 키 참조는 지원되지 않으며 RDS for MySQL과 OpenSearch 간에 데이터 불일치가 발생할 수 있습니다.
- Amazon RDS 다중 가용 영역 DB 클러스터는 지원되지 않습니다.
- 지원되는 버전: MySQL 버전 8.0 이상.

권장되는 CloudWatch 경보

수집 파이프라인의 성능을 모니터링하려면 다음 CloudWatch 지표를 사용하는 것이 좋습니다. 이러한 지표는 내보내기에서 처리된 데이터의 양, 스트림에서 처리된 이벤트 수, 내보내기 및 스트림 이벤트 처리 오류, 대상에 기록된 문서 수를 식별하는 데 도움이 될 수 있습니다. 이러한 지표 중 하나가 지정된 시간 동안 지정된 값을 초과하면 작업을 수행하도록 CloudWatch 경보를 설정할 수 있습니다.

지표	설명
pipeline-name .rds.credentialsChanged	이 지표는 AWS 보안 암호가 교체되는 빈도를 나타냅니다.
pipeline-name .rds.executorRefreshErrors	이 지표는 AWS 보안 암호 새로 고침 실패를 나타냅니다.

지표	설명
pipeline-name .rds.exportRecordsTotal	이 지표는 Amazon Aurora에서 내보낸 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordsProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordProcessingErrors	이 지표는 Amazon Aurora 클러스터에서 데이터를 읽는 동안 OpenSearch Ingestion 파이프라인의 처리 오류 수를 나타냅니다.
pipeline-name .rds.exportRecordsSuccessTotal	이 지표는 성공적으로 처리한 내보내기 레코드의 총 수를 나타냅니다.
pipeline-name .rds.exportRecordsFailedTotal	이 지표는 처리하지 못한 내보내기 레코드의 총 수를 나타냅니다.
pipeline-name .rds.bytesReceived	이 지표는 OpenSearch Ingestion 파이프라인에서 수신한 총 바이트 수를 나타냅니다.
pipeline-name .rds.bytesProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 총 바이트 수를 나타냅니다.
pipeline-name .rds.streamRecordsSuccessTotal	이 지표는 스트림에서 성공적으로 처리한 레코드 수를 나타냅니다.

지표	설명
pipeline-name .rds.streamRecordsFailedTotal	이 지표는 스트림에서 처리하지 못한 총 레코드 수를 나타냅니다.

RDS for PostgreSQL

Amazon RDS for RDS for PostgreSQL을 사용하여 OpenSearch Ingestion 파이프라인을 구성하려면 다음 단계를 완료하세요.

주제

- RDS for PostgreSQL 사전 조건
- 1단계: 파이프라인 역할 구성
- 2단계: 파이프라인 생성
- 데이터 일관성
- 데이터 형식 매핑
- 제한 사항
- 권장되는 CloudWatch 경보

RDS for PostgreSQL 사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. Amazon RDS에서 사용자 지정 DB 파라미터 그룹을 생성하여 논리적 복제를 구성합니다.

   ```
   rds.logical_replication=1
   ```

   자세한 내용은 Amazon RDS for PostgreSQL에 대한 논리적 복제 수행을 참조하세요.
2. RDS for PostgreSQL DB 인스턴스를 선택하거나 생성하고 1단계에서 생성한 파라미터 그룹을 DB 인스턴스와 연결합니다.
3. Aurora 및에서 암호 관리를 사용하여 Amazon RDS 인스턴스에 사용자 이름과 암호 AWS Secrets Manager 인증을 설정합니다. Secrets Manager 보안 암호를 생성하여 사용자 이름/암호 조합을 생성할 수도 있습니다.

4. 전체 초기 스냅샷 기능을 사용하는 경우 Amazon RDS에서 Amazon S3로 데이터를 내보내기 위한 AWS KMS key 및 IAM 역할을 생성합니다.

   IAM 역할에는 다음과 같은 권한 정책이 있어야 합니다.

   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/*"
            ]
        }
    ]
}
```

   역할에는 다음과 같은 신뢰 관계도 있어야 합니다.

   JSON

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

```
}
```

5. OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 선택하거나 생성합니다. 자세한 내용은 OpenSearch Service 도메인 생성 및 컬렉션 생성을 참조하세요.
6. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 Amazon RDS DB 인스턴스의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

1단계: 파이프라인 역할 구성

Amazon RDS 파이프라인 사전 조건을 설정한 후 파이프라인 구성에 사용할 파이프라인 역할을 구성합니다. 또한 Amazon RDS 소스에 대한 다음 권한을 역할에 추가합니다.

```
{
  "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:*:account-id:network-interface/*",
        "arn:aws:ec2:*:account-id:subnet/*",
        "arn:aws:ec2:*:account-id:security-group/*"
      ]
    },
    {
      "Sid": "allowDescribeEC2",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "allowTagCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:account-id:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/OSISManaged": "true"
        }
      }
    },
    {
      "Sid": "AllowDescribeInstances",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:db:*"
      ]
    },
     {
      "Sid": "AllowSnapshots",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBSnapshots",
        "rds:CreateDBSnapshot",
```

```
        "rds:AddTagsToResource"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:db:DB-id",
        "arn:aws:rds:region:account-id:snapshot:DB-id*"
      ]
    },
    {
      "Sid": "AllowExport",
      "Effect": "Allow",
      "Action": [
        "rds:StartExportTask"
      ],
      "Resource": [
        "arn:aws:rds:region:account-id:snapshot:DB-id*"
      ]
    },
    {
      "Sid": "AllowDescribeExports",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeExportTasks"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "region",
          "aws:ResourceAccount": "account-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToKmsForExport",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:DescribeKey",
        "kms:RetireGrant",
        "kms:CreateGrant",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*"
      ],
      "Resource": [
```

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

2단계: 파이프라인 생성

RDS for PostgreSQL 인스턴스를 소스로 지정하는 다음과 같이 OpenSearch Ingestion 파이프라인을 구성합니다.

```
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**
>
> 미리 구성된 Amazon RDS 블루프린트를 사용하여이 파이프라인을 생성할 수 있습니다. 자세한 내용은 블루프린트 작업 단원을 참조하십시오.

Amazon Aurora를 소스로 사용하려면 파이프라인에 대한 VPC 액세스를 구성해야 합니다. 선택한 VPC는 Amazon Aurora 소스에서 사용하는 것과 동일한 VPC여야 합니다. 그런 다음 하나 이상의 서브넷과 하나 이상의 VPC 보안 그룹을 선택합니다. 파이프라인에는 Aurora MySQL 데이터베이스에 대한 네트워크 액세스가 필요하므로 Aurora 클러스터가 파이프라인의 VPC 보안 그룹에서 데이터베이스 포

트로의 인바운드 트래픽을 허용하는 VPC 보안 그룹으로 구성되어 있는지도 확인해야 합니다. 자세한 내용은 보안 그룹을 사용한 액세스 제어를 참조하세요.

를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 Amazon Aurora를 소스로 사용하려면 VPC에 파이프라인도 연결해야 합니다. 이렇게 하려면 네트워크 구성 섹션을 찾고 VPC에 연결을 선택한 다음 제공된 기본 옵션 중 하나에서 CIDR을 선택하거나 직접 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.

사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 Amazon RDS 간의 IP 주소 충돌을 방지하려면 Amazon Aurora VPC CIDR이 OpenSearch Ingestion의 CIDR과 다른지 확인합니다.

자세한 내용은 파이프라인에 대한 VPC 액세스 구성을 참조하세요.

데이터 일관성

파이프라인은 Amazon RDS 인스턴스의 변경 사항을 지속적으로 폴링하거나 수신하고 OpenSearch 인덱스에서 해당 문서를 업데이트하여 데이터 일관성을 보장합니다.

OpenSearch Ingestion은 데이터 내구성을 보장하는 엔드 투 엔드 승인을 지원합니다. 파이프라인에서는 스냅샷이나 스트림을 읽을 때 병렬 처리를 위해 동적으로 분할을 생성합니다. 파이프라인에서는 OpenSearch 도메인이나 컬렉션에서 모든 레코드를 수집한 후 승인을 받으면 분할을 완료 상태로 표시합니다. OpenSearch Serverless 검색 컬렉션에 수집하려는 경우 파이프라인에서 문서 ID를 생성할 수 있습니다. OpenSearch Serverless 시계열 컬렉션에 수집하려면 파이프라인이 문서 ID를 생성하지 않으므로 파이프라인 싱크 구성`document_id: "${getMetadata(\"primary_key\")}"`에서를 생략해야 합니다.

또한 OpenSearch Ingestion 파이프라인에서는 수신 이벤트 작업을 해당하는 대량 인덱싱 작업에 매핑하여 문서를 쉽게 수집할 수 있게 합니다. 이렇게 하면 데이터가 일관되게 유지되므로 Amazon RDS의 모든 데이터 변경 사항이 OpenSearch의 해당 문서 변경 사항과 조정됩니다.

데이터 형식 매핑

OpenSearch Ingestion 파이프라인은 PostgreSQL 데이터 형식을 OpenSearch Service 도메인 또는 컬렉션이 사용하기에 적합한 표현에 매핑합니다. OpenSearch에 매핑 템플릿이 정의되지 않은 경우 OpenSearch는 처음 전송된 문서를 기반으로 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 매핑 템플릿을 통해 OpenSearch에서 가장 적합한 필드 유형을 명시적으로 정의할 수도 있습니다.

아래 표에는 RDS for PostgreSQL 데이터 형식과 해당 OpenSearch 필드 형식이 나와 있습니다. 명시적 매핑이 정의되지 않은 경우 기본 OpenSearch 필드 유형 열에 OpenSearch의 해당 필드 유형이

표시됩니다. 이 경우 OpenSearch는 동적 매핑을 사용하여 필드 유형을 자동으로 결정합니다. 권장 OpenSearch 필드 유형 열은 매핑 템플릿에서 명시적으로 지정하는 해당 권장 필드 유형입니다. 이러한 필드 유형은 RDS for PostgreSQL의 데이터 형식과 더 밀접하게 정렬되며 일반적으로 OpenSearch에서 더 나은 검색 기능을 사용할 수 있습니다.

RDS for PostgreSQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
smallint	long	bigint
정수	long	정수
bigint	long	long
decimal	텍스트	double 또는 키워드
숫자[(p, s)]	텍스트	double 또는 키워드
real	float	float
double precision	float	double
smallserial	long	bigint
serial	long	정수
bigserial	long	long
money	객체	객체
character varying(n)	텍스트	텍스트
varchar(n)	텍스트	텍스트

RDS for PostgreSQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
character (n)	텍스트	텍스트
char(n)	텍스트	텍스트
bpchar(n)	텍스트	텍스트
bpchar	텍스트	텍스트
텍스트	텍스트	텍스트
enum	텍스트	텍스트
bytea	텍스트	이진수
타임스탬프 [(p)] [시간대 없음]	long(에포크 밀리초)	date
시간대가 있는 타임스탬프 [(p)]	long(에포크 밀리초)	date
date	long(에포크 밀리초)	date
time [(p)] (시간대 제외)	long(에포크 밀리초)	date
시간대가 있는 시간 [(p)]	long(에포크 밀리초)	date

RDS for PostgreSQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
간격 [필드] [(p)]	text(ISO8601 형식)	텍스트
boolean	boolean	boolean
point	text(WKT 형식)	geo_shape
line	text(WKT 형식)	geo_shape
lseg	text(WKT 형식)	geo_shape
상자	text(WKT 형식)	geo_shape
경로	text(WKT 형식)	geo_shape
다각형	text(WKT 형식)	geo_shape
원	객체	객체
cidr	텍스트	텍스트
inet	텍스트	텍스트
macaddr	텍스트	텍스트
macaddr8	텍스트	텍스트
bit(n)	long	byte, short, integer 또는 long(비트 수에 따라 다름)
bit varying(n)	long	byte, short, integer 또는 long(비트 수에 따라 다름)
json	객체	객체
jsonb	객체	객체

RDS for PostgreSQL 데이터 형식	기본 OpenSearch 필드 유형	권장 OpenSearch 필드 유형
jsonpath	텍스트	텍스트

OpenSearch Ingestion 파이프라인에서 DLQ(Dead Letter Queue)를 구성하는 것이 좋습니다. 이 대기열을 구성하면 OpenSearch Service에서는 동적 매핑 실패로 인해 수집할 수 없는 모든 실패한 문서를 대기열로 전송합니다.

자동 매핑이 실패할 경우 파이프라인 구성에서 `template_type` 및 `template_content`를 사용하여 명시적 매핑 규칙을 정의할 수 있습니다. 또는 파이프라인을 시작하기 전에 검색 도메인이나 컬렉션에서 직접 매핑 템플릿을 생성할 수도 있습니다.

제한 사항

RDS for PostgreSQL용 OpenSearch Ingestion 파이프라인을 설정할 때 다음 제한 사항을 고려하세요.

- 통합은 파이프라인당 하나의 PostgreSQL 데이터베이스만 지원합니다.
- 통합은 현재 리전 간 데이터 수집을 지원하지 않습니다. Amazon RDS 인스턴스와 OpenSearch 도메인은 동일한에 있어야 합니다 AWS 리전.
- 통합은 현재 교차 계정 데이터 수집을 지원하지 않습니다. Amazon RDS 인스턴스와 OpenSearch Ingestion 파이프라인은 동일한에 있어야 합니다 AWS 계정.
- 지원되는 유일한 인증 메커니즘 AWS Secrets Manager인를 사용하여 Amazon RDS 인스턴스에 인증이 활성화되어 있는지 확인합니다.
- 다른 데이터베이스 및/또는 다른 테이블에서 데이터를 수집하도록 기존 파이프라인 구성을 업데이트할 수 없습니다. 파이프라인의 데이터베이스 및/또는 테이블 이름을 업데이트하려면 파이프라인을 중지하고 업데이트된 구성으로 다시 시작하거나 새 파이프라인을 생성해야 합니다.
- 데이터 정의 언어(DDL) 문은 일반적으로 지원되지 않습니다. 다음과 같은 경우에는 데이터 일관성이 유지되지 않습니다.
  - 기본 키가 변경됩니다(add/delete/rename).
  - 테이블이 삭제/절단됩니다.
  - 열 이름 또는 데이터 형식이 변경됩니다.

- 동기화할 PostgreSQL 테이블에 기본 키가 정의되지 않은 경우 데이터 일관성이 보장되지 않습니다. 업데이트/삭제를 OpenSearch에 동기화하려면 OpenSearch 및 싱크 구성에서 사용자 지정 `document_id` 옵션을 올바르게 정의해야 합니다 OpenSearch.
- RDS 다중 AZ DB 클러스터는 지원되지 않습니다.
- 지원되는 버전: PostgreSQL 16 이상.

권장되는 CloudWatch 경보

수집 파이프라인의 성능을 모니터링하려면 다음 CloudWatch 지표를 사용하는 것이 좋습니다. 이러한 지표는 내보내기에서 처리된 데이터의 양, 스트림에서 처리된 이벤트 수, 내보내기 및 스트림 이벤트 처리 오류, 대상에 기록된 문서 수를 식별하는 데 도움이 될 수 있습니다. 이러한 지표 중 하나가 지정된 시간 동안 지정된 값을 초과하면 작업을 수행하도록 CloudWatch 경보를 설정할 수 있습니다.

지표	설명
pipeline-name .rds.credentialsChanged	이 지표는 AWS 보안 암호가 교체되는 빈도를 나타냅니다.
pipeline-name .rds.executorRefreshErrors	이 지표는 AWS 보안 암호 새로 고침 실패를 나타냅니다.
pipeline-name .rds.exportRecordsTotal	이 지표는 Amazon Aurora에서 내보낸 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordsProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 레코드 수를 나타냅니다.
pipeline-name .rds.exportRecordProcessingErrors	이 지표는 Amazon Aurora 클러스터에서 데이터를 읽는 동안 OpenSearch Ingestion 파이프라인의 처리 오류 수를 나타냅니다.
pipeline-name .rds.expo	이 지표는 성공적으로 처리한 내보내기 레코드의 총 수를 나타냅니다.

지표	설명
rtRecords SuccessTotal	
pipeline-name .rds.expo rtRecordsFailedTot al	이 지표는 처리하지 못한 내보내기 레코드의 총 수를 나타냅니다.
pipeline-name .rds.byte sReceived	이 지표는 OpenSearch Ingestion 파이프라인에서 수신한 총 바이트 수를 나타냅니다.
pipeline-name .rds.byte sProcessed	이 지표는 OpenSearch Ingestion 파이프라인에서 처리한 총 바이트 수를 나타냅니다.
pipeline-name .rds.stre amRecords SuccessTotal	이 지표는 스트림에서 성공적으로 처리한 레코드 수를 나타냅니다.
pipeline-name .rds.stre amRecords FailedTotal	이 지표는 스트림에서 처리하지 못한 총 레코드 수를 나타냅니다.

Amazon S3와 함께 OpenSearch Ingestion 파이프라인 사용

OpenSearch Ingestion 기능을 사용하면 Amazon S3를 원본 또는 대상으로 사용할 수 있습니다. Amazon S3를 소스로 사용하는 경우 OpenSearch Ingestion 파이프라인으로 데이터를 전송합니다. Amazon S3를 대상으로 사용하는 경우 OpenSearch Ingestion 파이프라인의 데이터를 하나 이상의 S3 버킷에 기록합니다.

주제

- 소스로서의 Amazon S3
- 대상으로서의 Amazon S3

- 소스 역할을 하는 Amazon S3 교차 계정

소스로서의 Amazon S3

Amazon S3를 데이터 처리 원본으로 사용할 수 있는 두 가지 방법, 즉 S3-SQS 처리와 예약 스캔이 있습니다.

S3에 파일을 기록한 후 파일을 거의 실시간으로 스캔해야 하는 경우 S3-SQS 처리를 사용하세요. 객체가 버킷 내에 저장되거나 수정될 때마다 이벤트를 발생시키도록 Amazon S3 버킷을 구성할 수 있습니다. 일회성 또는 반복되는 예약 스캔을 사용하여 S3 버킷의 데이터를 일괄 처리하세요.

주제

- 사전 조건
- 1단계: 파이프라인 역할 구성
- 2단계: 파이프라인 생성

사전 조건

예약 스캔 또는 S3-SQS 처리 모두에 대해 Amazon S3를 OpenSearch Ingestion 파이프라인의 소스로 사용하려면 먼저 S3 버킷을 생성하세요.

> **Note**
>
> OpenSearch Ingestion 파이프라인에서 소스로 사용되는 S3 버킷이 다른에 AWS 계정있는 경우 버킷에 대한 교차 계정 읽기 권한도 활성화해야 합니다. 이렇게 하면 파이프라인이 데이터를 읽고 처리할 수 있습니다. 교차 계정 권한을 활성화하려면 Amazon S3 사용 설명서의 계정 간 버킷 권한 부여하는 버킷 소유자를 참조하세요.
> S3 버킷이 여러 계정에 있는 경우 `bucket_owners` 맵을 사용합니다. 예제는 OpenSearch 설명서의 Cross-account S3 access를 참조하세요.

S3-SQS 처리를 설정하려면 다음 단계도 수행해야 합니다.

1. Amazon SQS 대기열을 생성합니다.
2. SQS 대기열을 대상으로 하는 S3 버킷에서 이벤트 알림을 활성화합니다.

1단계: 파이프라인 역할 구성

데이터를 파이프라인으로 푸시하는 다른 소스 플러그인과 달리 S3 소스 플러그인은 파이프라인이 소스에서 데이터를 가져오는 읽기 기반 아키텍처를 사용합니다.

따라서 S3에서 파이프라인을 읽으려면 S3 버킷과 Amazon SQS 대기열 모두에 액세스할 수 있는 파이프라인의 S3 소스 구성 내에서 역할을 지정해야 합니다. 파이프라인은 대기열에서 데이터를 읽기 위해 이 역할을 맡습니다.

> **Note**
>
> S3 소스 구성 내에서 지정하는 역할은 파이프라인 역할이어야 합니다. 따라서 파이프라인 역할에는 두 개의 개별 권한 정책이 포함되어야 합니다. 하나는 싱크에 쓰는 정책이고 다른 하나는 S3 소스에서 가져오기 위한 것입니다. 모든 파이프라인 구성 요소에서 `sts_role_arn`을 동일하게 사용해야 합니다.

다음 샘플 정책은 S3를 소스로 사용하는 데 필요한 권한을 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action":[
          "s3:ListBucket",
          "s3:GetBucketLocation",
          "s3:GetObject"
       ],
      "Resource": "arn:aws:s3:::/*"
    },
    {
       "Effect":"Allow",
       "Action":"s3:ListAllMyBuckets",
       "Resource":"arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "sqs:DeleteMessage",
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility"
      ],
      "Resource": "arn:aws:sqs:us-west-2:111122223333:MyS3EventSqsQueue"
    }
  ]
}
```

S3 소스 플러그인 구성 내 `sts_role_arn` 옵션에 지정하는 IAM 역할에 다음 권한을 연결해야 합니다.

```
version: "2"
source:
  s3:
    ...
    aws:
      ...
processor:
  ...
sink:
  - opensearch:
      ...
```

2단계: 파이프라인 생성

권한을 설정한 후 Amazon S3 사용 사례에 따라 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

S3-SQS 처리

S3-SQS 처리를 설정하려면 S3를 소스로 지정하도록 파이프라인을 구성하고 Amazon SQS 알림을 설정하세요.

```
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"
```

Amazon S3에서 작은 파일을 처리하는 동안 CPU 사용률이 낮게 나타나면 `workers` 옵션 값을 수정하여 처리량을 늘리는 방법을 고려합니다. 자세한 내용은 S3 plugin configuration options를 참조하세요.

예약 스캔

예약 스캔을 설정하려면 모든 S3 버킷에 적용되는 스캔 수준 또는 버킷 수준의 일정으로 파이프라인을 구성하세요. 버킷 수준 일정 또는 스캔 간격 구성은 항상 스캔 수준 구성을 덮어씁니다.

예약 스캔은 데이터 마이그레이션에 적합한 1회성 스캔 또는 일괄 처리에 적합한 반복 스캔으로 구성할 수 있습니다.

Amazon S3에서 읽을 파이프라인을 구성하려면 사전 구성된 Amazon S3 블루프린트를 사용합니다. 일정 요구 사항에 맞게 파이프라인 구성의 `scan` 일부를 편집할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

1회성 스캔

1회성 예약 스캔은 한 번 실행됩니다. 파이프라인 구성에서 `start_time` 및를 사용하여 버킷의 객체를 스캔할 시기를 `end_time` 지정할 수 있습니다. 또는 버킷의 객체를 스캔하려는 현재 시간을 기준으로 시간 간격을 지정하는 데 `range`을 사용할 수 있습니다.

예를 들어 최근 4시간 동안 생성된 모든 파일을 `PT4H` 스캔하도록 범위를 설정합니다. 한 번 스캔을 두 번 실행하도록 구성하려면 파이프라인을 중지하고 다시 시작해야 합니다. 범위를 구성하지 않은 경우 시작 시간 및 종료 시간도 업데이트해야 합니다.

다음 구성은 모든 버킷과 해당 버킷의 모든 객체를 한 번 스캔하도록 설정합니다.

```
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"
```

다음 구성은 지정된 기간 동안 모든 버킷에 대한 1회성 스캔을 설정합니다. 즉, S3는 생성 시간이 이 기간에 해당하는 객체만 처리합니다.

```
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
```

다음 구성은 스캔 수준과 버킷 수준 모두에서 1회성 스캔을 설정합니다. 버킷 수준의 시작 및 종료 시간은 스캔 수준의 시작 및 종료 시간보다 우선합니다.

```
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
```

파이프라인을 중지하면 중지 전에 파이프라인에서 스캔한 객체에 대한 기존 참조가 제거됩니다. 단일 스캔 파이프라인이 중지되면 이미 스캔되었어도 시작된 이후 모든 객체를 다시 스캔합니다. 단일 스캔 파이프라인을 중지해야 하는 경우 파이프라인을 다시 시작하기 전에 기간을 변경하는 것이 좋습니다.

시작 시간 및 종료 시간별로 객체를 필터링해야 하는 경우 파이프라인을 중지하고 시작하는 것이 유일한 옵션입니다. 시작 시간 및 종료 시간을 기준으로 필터링할 필요가 없는 경우 이름을 기준으로 객체를 필터링할 수 있습니다. 이름을 기준으로 필터링하는 경우 파이프라인을 중지하고 시작할 필요가 없습니다. 이렇게 하려면 `include_prefix` 및 `exclude_suffix`를 사용합니다.

반복 스캔

반복 예약 스캔은 지정된 S3 버킷의 스캔을 정기적으로 예약된 간격으로 실행합니다. 개별 버킷 수준 구성은 지원되지 않으므로 스캔 수준에서만 이러한 간격을 구성할 수 있습니다.

파이프라인 구성에서는 반복 스캔 빈도를 `interval` 지정하며 30초에서 365일 사이일 수 있습니다. 파이프라인을 생성할 때 항상 첫 번째 스캔이 발생합니다. `count`는 스캔 인스턴스 총 수를 정의합니다.

다음 구성은 스캔 사이에 12시간의 지연을 두고 반복 스캔을 설정합니다.

```
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
```

대상으로서의 Amazon S3

OpenSearch Ingestion 파이프라인의 데이터를 S3 버킷에 쓰려면 사전 구성된 S3 블루프린트를 사용하여 S3 싱크가 있는 파이프라인을 생성합니다. 이 파이프라인은 선택적 데이터를 OpenSearch 싱크로 라우팅하고 동시에 S3에 보관할 모든 데이터를 전송합니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

S3 싱크를 생성할 때 다양한 싱크 코덱 중에서 원하는 형식을 지정할 수 있습니다. 예를 들어 데이터를 열 형식으로 쓰려면 Parquet 또는 Avro 코덱을 선택하세요. 행 기반 형식을 선호하는 경우 JSON 또는 NDJSON을 선택합니다. 지정된 스키마로 S3에 데이터를 쓰려면 Avro 형식을 사용하여 싱크 코덱 내에 인라인 스키마를 정의할 수도 있습니다.

다음 예제에서는 S3 싱크에 인라인 스키마를 정의합니다.

```
- 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"}
          ]
        }
```

이 스키마를 정의할 때는 파이프라인이 싱크에 전달하는 다양한 유형의 이벤트에 존재할 수 있는 모든 키의 상위 세트를 지정하세요.

예를 들어 이벤트에 키가 누락될 가능성이 있는 경우 스키마에 해당 키를 `null` 값과 함께 추가하세요. Null 값 선언을 사용하면 스키마가 비균일 데이터를 처리할 수 있습니다(일부 이벤트에는 이러한 키가 있고 다른 이벤트에는 이러한 키가 없는 경우). 수신 이벤트에 이러한 키가 있는 경우 해당 값이 싱크에 기록됩니다.

이 스키마 정의는 정의된 키만 싱크로 전송하도록 허용하고 수신 이벤트에서 정의되지 않은 키를 삭제하는 필터 역할을 합니다.

싱크에서 `include_keys` 및 `exclude_keys`를 사용하여 다른 싱크로 라우팅되는 데이터를 필터링할 수도 있습니다. 이 두 필터는 상호 배타적이므로 스키마에서 한 번에 하나만 사용할 수 있습니다. 또한 사용자 정의 스키마 내에서는 이러한 스키마를 사용할 수 없습니다.

이러한 필터를 사용하여 파이프라인을 생성하려면 사전 구성된 싱크 필터 블루프린트를 사용합니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

소스 역할을 하는 Amazon S3 교차 계정

OpenSearch Ingestion 파이프라인이 다른 계정의 S3 버킷에 소스로 액세스할 수 있도록 Amazon S3가 있는 여러 계정에서 액세스 권한을 부여할 수 있습니다. 교차 계정 액세스를 활성화하려면 Amazon S3 사용 설명서의 교차 계정 버킷 권한을 부여하는 버킷 소유자를 참조하세요. 액세스 권한을 부여한 후 파이프라인 역할에 필요한 권한이 있는지 확인합니다.

그런 다음를 사용하여 Amazon S3 버킷`bucket_owners`에 대한 교차 계정 액세스를 소스로 활성화하는 파이프라인을 생성할 수 있습니다.

```
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"
```

Amazon Security Lake와 함께 OpenSearch Ingestion 파이프라인 사용

S3 소스 플러그인을 사용하여 Amazon Security Lake의 데이터를 OpenSearch Ingestion 파이프라인으로 수집할 수 있습니다. Security Lake는 AWS 환경, 온프레미스 환경 및 SaaS 공급자의 보안 데이터를 특별히 구축된 데이터 레이크로 자동으로 중앙 집중화합니다. Security Lake의 데이터를 OpenSearch Ingestion 파이프라인으로 복제하는 구독을 생성하여 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션에 데이터를 쓸 수 있습니다.

Security Lake에서 읽을 파이프라인을 구성하려면 사전 구성된 Security Lake 블루프린트를 사용합니다. 청사진에는 Security Lake에서 Open Cybersecurity Schema Framewor(OCSF) 파켓 파일을 수집하기 위한 기본 구성이 포함되어 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

주제

- Amazon Security Lake를 소스로 사용하여 OpenSearch Ingestion 파이프라인 사용
- Amazon Security Lake를 싱크로 사용하여 OpenSearch Ingestion 파이프라인 사용

Amazon Security Lake를 소스로 사용하여 OpenSearch Ingestion 파이프라인 사용

OpenSearch Ingestion 파이프라인 내에서 Amazon S3 소스 플러그인을 사용하여 Amazon Security Lake에서 데이터를 수집할 수 있습니다. Security Lake는 환경, 온프레미스 시스템 및 SaaS 공급자의 AWS 보안 데이터를 특별히 구축된 데이터 레이크로 자동으로 중앙 집중화합니다.

Amazon Security Lake에는 파이프라인 내에 다음과 같은 메타데이터 속성이 있습니다.

- `bucket_name`: 보안 데이터를 저장하기 위해 Security Lake에서 생성한 Amazon S3 버킷의 이름입니다.
- `path_prefix`: Security Lake IAM 역할 정책에 정의된 사용자 지정 소스 이름입니다.
- `region`: Security Lake S3 버킷이 AWS 리전 위치한 입니다.

- `accountID`: Security Lake가 활성화된 AWS 계정 ID입니다.
- `sts_role_arn`: Security Lake와 함께 사용하기 위한 IAM 역할의 ARN입니다.

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

- Security Lake를 활성화합니다.
- Security Lake에서 구독자를 생성하세요.
  - 파이프라인에 수집하려는 소스를 선택하세요.
  - 구독자 자격 증명에 파이프라인을 생성하려는 AWS 계정 의 ID를 추가합니다. 외부 ID의 경우 `OpenSearchIngestion-{accountid}`을 지정하세요.
  - 데이터 액세스 메서드로는 S3를 선택합니다.
  - 알림 세부 정보를 보려면 SQS 대기열을 선택합니다.

구독자를 생성하면 Security Lake는 자동으로 두 개의 인라인 권한 정책을 생성합니다. 하나는 S3용이고 다른 하나는 SQS용입니다. 정책 형식은 `AmazonSecurityLake-{12345}-S3` 및 `AmazonSecurityLake-{12345}-SQS`입니다. 파이프라인이 구독자 소스에 액세스할 수 있게 하려면 필요한 권한을 파이프라인 역할에 연결해야 합니다.

파이프라인 역할 구성

Security Lake에서 자동으로 생성한 두 정책의 필수 권한만 결합하는 새 권한 정책을 IAM에 생성하세요. 다음 예제 정책은 OpenSearch Ingestion 파이프라인이 여러 Security Lake 소스의 데이터를 읽는 데 필요한 최소 권한을 보여줍니다.

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 Lake는 파이프라인 역할 정책을 대신 관리하지 않습니다. Security Lake 구독에서 소스를 추가하거나 제거하는 경우 정책을 수동으로 업데이트해야 합니다. Security Lake는 각 로그 소스에 대해 파티션을 생성하므로 파이프라인 역할에서 권한을 수동으로 추가하거나 제거해야 합니다.

sqs에서 S3 소스 플러그인 구성 내 `sts_role_arn` 옵션에 지정하는 IAM 역할에 다음 권한을 연결해야 합니다.

```
version: "2"
source:
  s3:
    ...
```

```
    sqs:
      queue_url: "https://sqs.us-east-1amazonaws.com/account-id/
AmazonSecurityLake-abcde-Main-Queue"
    aws:
      ...
processor:
  ...
sink:
  - opensearch:
      ...
```

파이프라인 생성

파이프라인 역할에 권한을 추가한 후 사전 구성된 Security Lake 블루프린트를 사용하여 파이프라인을 생성합니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

s3 소스 구성 내에서 읽을 Amazon SQS 대기열 URL인 queue_url 옵션을 지정해야 합니다. URL 형식을 지정하려면 구독자 구성에서 구독 엔드포인트를 찾아 arn:aws:를 https://로 변경하세요. 예를 들어 https://sqs.*us-east-1*amazonaws.com/*account-id*/AmazonSecurityLake-*abdcef*-Main-Queue입니다.

S3 소스 구성 내에서 지정하는 sts_role_arn은 파이프라인 역할의 ARN이어야 합니다.

Amazon Security Lake를 싱크로 사용하여 OpenSearch Ingestion 파이프라인 사용

OpenSearch Ingestion의 Amazon S3 싱크 플러그인을 사용하여 지원되는 소스에서 Amazon Security Lake로 데이터를 전송합니다. Security Lake는, AWS온프레미스 환경 및 SaaS 공급자로부터 보안 데이터를 수집하여 전용 데이터 레이크에 저장합니다.

Security Lake에 로그 데이터를 쓰도록 파이프라인을 구성하려면 미리 구성된 방화벽 트래픽 로그 블루프린트를 사용합니다. 블루프린트에는 Amazon S3 버킷에 저장된 원시 보안 로그 또는 기타 데이터를 검색하고, 레코드를 처리하고, 정규화하기 위한 기본 구성이 포함되어 있습니다. 그런 다음 데이터를 OCSF(Open Cybersecurity Schema Framework)에 매핑하고 변환된 OCSF 준수 데이터를 Security Lake로 전송합니다.

파이프라인에는 다음과 같은 메타데이터 속성이 있습니다.

- bucket_name: 보안 데이터를 저장하기 위해 Security Lake에서 생성한 Amazon S3 버킷의 이름입니다.
- path_prefix: Security Lake IAM 역할 정책에 정의된 사용자 지정 소스 이름입니다.
- region: Security Lake S3 버킷이 AWS 리전 위치한 입니다.

- `accountID`: Security Lake가 활성화된 AWS 계정 ID입니다.
- `sts_role_arn`: Security Lake와 함께 사용하기 위한 IAM 역할의 ARN입니다.

사전 조건

Security Lake로 데이터를 전송할 파이프라인을 생성하기 전에 다음 단계를 수행합니다.

- Amazon Security Lake 활성화 및 구성: 다양한 소스의 보안 데이터를 중앙 집중화하도록 Amazon Security Lake를 설정합니다. 지침은 콘솔을 사용하여 Security Lake 활성화를 참조하세요.

  소스를 선택할 때 특정 AWS 소스 수집을 선택하고 수집하려는 로그 및 이벤트 소스를 하나 이상 선택합니다.
- 권한 설정: Security Lake에 데이터를 쓰는 데 필요한 권한으로 파이프라인 역할을 구성합니다. 자세한 내용은 파이프라인 역할을 참조하세요.

파이프라인 생성

미리 구성된 Security Lake 블루프린트를 사용하여 파이프라인을 생성합니다. 자세한 내용은 블루프린트를 사용하여 파이프라인 생성을 참조하세요.

Fluent Bit와 함께 OpenSearch Ingestion 파이프라인 사용

이 샘플 Fluent Bit 구성 파일은 Fluent Bit의 로그 데이터를 OpenSearch Ingestion 파이프라인으로 보냅니다. 로그 데이터 수집에 대한 자세한 내용은 Data Prepper 설명서의 로그 분석을 참조하세요.

다음 사항에 유의하세요.

- `host` 값은 파이프라인 엔드포인트여야 합니다. 예를 들어 *`pipeline-endpoint`*`.`*`us-east-1`*`osis.amazonaws.com`입니다.
- `aws_service` 값은 `osis`여야 합니다.
- `aws_role_arn` 값은 클라이언트가 서명 버전 4 인증에 수임하고 사용할 AWS IAM 역할의 ARN입니다.

```
[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
```

그런 다음 HTTP를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

```
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"
```

Fluent Bit에서 OpenSearch Ingestion 파이프라인 사용

Fluentd는 Fluent Bit와 같은 다양한 언어 및 하위 프로젝트용 SDK를 제공하는 오픈 소스 데이터 컬렉션 에코시스템입니다. 이 샘플 Fluentd 구성 파일은 Fluentd의 로그 데이터를 OpenSearch Ingestion 파이프라인으로 전송합니다. 로그 데이터 수집에 대한 자세한 내용은 Data Prepper 설명서의 로그 분석을 참조하세요.

다음 사항에 유의하세요.

- endpoint 값은 파이프라인 엔드포인트여야 합니다. 예를 들어 *pipeline-endpoint*.*us-east-1*osis.amazonaws.com/apache-log-pipeline/logs입니다.
- aws_service 값은 osis여야 합니다.
- aws_role_arn 값은 클라이언트가 서명 버전 4 인증에 수임하고 사용할 AWS IAM 역할의 ARN입니다.

```
<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>
```

그런 다음 HTTP를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

```
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
```

OpenTelemetry Collector와 함께 OpenSearch Ingestion 파이프라인 사용

이 샘플 OpenTelemetry 구성 파일은 OpenTelemetry Collector에서 추적 데이터를 내보내 OpenSearch Ingestion 파이프라인으로 보냅니다. 수집 추적 데이터에 대한 자세한 내용은 Data Prepper 설명서의 로그 분석을 참조하세요.

다음 사항에 유의하세요.

- `endpoint` 값에는 파이프라인 엔드포인트가 포함되어야 합니다. 예를 들어 `https://pipeline-endpoint.us-east-1osis.amazonaws.com`입니다.
- `service` 값은 `osis`여야 합니다.
- OTLP/HTTP Exporter의 `compression` 옵션은 파이프라인의 OpenTelemetry 소스에서 `compression` 옵션과 일치해야 합니다.

```
extensions:
  sigv4auth:
    region: "region"
    service: "osis"

receivers:
  jaeger:
    protocols:
      grpc:

exporters:
  otlphttp:
    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]
```

그런 다음 OTel 추적 플러그인을 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

```
version: "2"
otel-trace-pipeline:
  source:
    otel_trace_source:
      path: "/v1/traces"
  processor:
    - trace_peer_forwarder:
  sink:
    - pipeline:
        name: "trace-pipeline"
    - pipeline:
        name: "service-map-pipeline"
trace-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - otel_traces:
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index_type: trace-analytics-raw
        aws:
          region: "region"
service-map-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - service_map:
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index_type: trace-analytics-service-map
        aws:
          region: "region"
```

다른 예제 파이프라인은 사전 구성된 추적 분석 파이프라인 블루프린트를 참조하세요. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

Kafka에서 OpenSearch Ingestion 파이프라인 사용

Kafka 플러그인을 사용하여 자체 관리형 Kafka 클러스터의 데이터를 Amazon OpenSearch Service 도메인 및 OpenSearch Serverless 컬렉션으로 스트리밍할 수 있습니다. OpenSearch Ingestion은 퍼블릭 또는 프라이빗(VPC) 네트워킹으로 구성된 Kafka 클러스터의 연결을 지원합니다. 이 주제에서는 상호 TLS(mTLS), SASL/SCRAM 또는 IAM과 같은 네트워크 설정 및 인증 방법 구성을 포함하여 수집 파이프라인을 설정하기 위한 사전 조건과 단계를 간략하게 설명합니다.

퍼블릭 Kafka 클러스터에서 데이터 마이그레이션

OpenSearch Ingestion 파이프라인을 사용하여 퍼블릭 자체 관리형 Kafka 클러스터에서 데이터를 마이그레이션할 수 있습니다. 즉, 도메인 DNS 이름을 공개적으로 확인할 수 있습니다. 이렇게 하려면 자체 관리형 Kafka를 소스로, OpenSearch Service 또는 OpenSearch Serverless를 대상으로 하는 OpenSearch Ingestion 파이프라인을 설정합니다. 이렇게 하면 자체 관리형 소스 클러스터에서 AWS관리형 대상 도메인 또는 컬렉션으로 스트리밍 데이터가 처리됩니다.

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. 퍼블릭 네트워크 구성을 사용하여 자체 관리형 Kafka 클러스터를 생성합니다. 클러스터에는 OpenSearch Service로 수집할 데이터가 포함되어 있어야 합니다.
2. 데이터를 마이그레이션할 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성합니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 및 the section called “컬렉션 생성” 섹션을 참조하세요.
3. 를 사용하여 자체 관리형 클러스터에 인증을 설정합니다 AWS Secrets Manager. AWS Secrets Manager 시크릿 교체의 단계를 수행하여 시크릿 교체를 활성화합니다.
4. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 자체 관리형 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

   다음 샘플 도메인 액세스 정책은 다음 단계에서 생성하는 파이프라인 역할을 사용하여 도메인에 데이터를 쓰도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   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"
      ]
    }
  ]
}
```

컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 섹션을 참조하세요the section called “역할 및 사용자 설정”.

1단계: 파이프라인 역할 rnfigure

Kafka 파이프라인 사전 조건을 설정한 후 파이프라인 구성에 사용할 파이프라인 역할을 구성하고 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션에 쓸 수 있는 권한과 Secrets Manager에서 시크릿을 읽을 수 있는 권한을 추가합니다.

2단계: 파이프라인 생성

이제 Kafka를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

여러 OpenSearch Service 도메인을 데이터의 대상으로 지정할 수 있습니다. 이 기능을 사용하여 조건부로 라우팅하거나 수신 데이터를 여러 OpenSearch Service 도메인으로 복제할 수 있습니다.

소스 Confluent Kafka의 데이터를 OpenSearch Serverless VPC 컬렉션으로 마이그레이션할 수도 있습니다. 파이프라인 구성 내에서 네트워크 액세스 정책을 제공해야 합니다. Confluent 스키마 레지스트리를 사용하여 Confluent 스키마를 정의할 수 있습니다.

```
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"
```

사전 구성된 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

VPC의 Kafka 클러스터에서 데이터 마이그레이션

OpenSearch Ingestion 파이프라인을 사용하여 VPC에서 실행되는 자체 관리형 Kafka 클러스터에서 데이터를 마이그레이션할 수도 있습니다. 이렇게 하려면 자체 관리형 Kafka를 소스로, OpenSearch Service 또는 OpenSearch Serverless를 대상으로 하는 OpenSearch Ingestion 파이프라인을 설정합니다. 이렇게 하면 자체 관리형 소스 클러스터에서 AWS관리형 대상 도메인 또는 컬렉션으로 스트리밍 데이터가 처리됩니다.

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. OpenSearch Service로 수집하려는 데이터가 포함된 VPC 네트워크 구성을 사용하여 자체 관리형 Kafka 클러스터를 생성합니다.
2. 데이터를 마이그레이션할 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성합니다. 자세한 내용은 OpenSearch Service 도메인 생성 및 컬렉션 생성을 참조하세요.
3. 를 사용하여 자체 관리형 클러스터에 인증을 설정합니다 AWS Secrets Manager. AWS Secrets Manager 시크릿 교체의 단계를 수행하여 시크릿 교체를 활성화합니다.
4. 자체 관리형 Kafka에 대한 액세스 권한을 보유한 VPC의 ID를 가져옵니다. OpenSearch Ingestion에서 사용할 VPC CIDR을 선택합니다.

   > **Note**
   >
   > 를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 자체 관리형 Kafka를 사용하려면 OpenSearch Ingestion 파이프라인도 VPC에 연결해야 합니다. 이를 수행하려면 네트워크 구성 섹션을 찾아 VPC에 연결 확인란을 선택하고 제공된 기본 옵션 중 하나에서 CIDR을 선택하거나 직접 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.
   > 사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 자체 관리형 OpenSearch 간의 IP 주소 충돌을 방지하려면 자체 관리형 OpenSearch VPC CIDR이 OpenSearch Ingestion의 CIDR과 달라야 합니다.

5. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 자체 관리형 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

   다음 샘플 도메인 액세스 정책은 다음 단계에서 생성하는 파이프라인 역할을 사용하여 도메인에 데이터를 쓰도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   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"
      ]
    }
  ]
}
```

컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 섹션을 참조하세요the section called “역할 및 사용자 설정”.

1단계: 파이프라인 역할 구성

파이프라인 사전 조건을 설정한 후 파이프라인 구성에서 사용하려는 파이프라인 역할을 구성하고 역할에서 다음 권한을 추가합니다.

JSON

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

파이프라인은 이러한 권한을 사용하여 VPC에서 네트워크 인터페이스를 생성하고 삭제하므로 OpenSearch Ingestion 파이프라인을 생성하는 데 사용하는 IAM 역할에서 위의 Amazon EC2 권한을 제공해야 합니다. 파이프라인은 오직 이 네트워크 인터페이스를 통해 Kafka 클러스터에 액세스할 수 있습니다.

2단계: 파이프라인 생성

이제 Kafka를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

여러 OpenSearch Service 도메인을 데이터의 대상으로 지정할 수 있습니다. 이 기능을 사용하여 조건부로 라우팅하거나 수신 데이터를 여러 OpenSearch Service 도메인으로 복제할 수 있습니다.

소스 Confluent Kafka의 데이터를 OpenSearch Serverless VPC 컬렉션으로 마이그레이션할 수도 있습니다. 파이프라인 구성 내에서 네트워크 액세스 정책을 제공해야 합니다. Confluent 스키마 레지스트리를 사용하여 Confluent 스키마를 정의할 수 있습니다.

```
 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"
```

사전 구성된 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

Amazon OpenSearch Ingestion을 사용하여 자체 관리형 OpenSearch 클러스터에서 데이터 마이그레이션

자체 관리형 OpenSearch 또는 Elasticsearch와 함께 Amazon OpenSearch Ingestion 파이프라인을 사용하여 데이터를 Amazon OpenSearch Service 도메인 및 OpenSearch Serverless 컬렉션으로 마이그레이션할 수 있습니다. OpenSearch Ingestion은 자체 관리형 OpenSearch 및 Elasticsearch에서 데이터를 마이그레이션하기 위해 퍼블릭 및 프라이빗 네트워크 구성을 모두 지원합니다.

퍼블릭 OpenSearch 클러스터에서 마이그레이션

OpenSearch Ingestion 파이프라인을 사용하여 퍼블릭 구성을 통해 자체 관리형 OpenSearch 또는 Elasticsearch 클러스터에서 데이터를 마이그레이션할 수 있습니다. 즉, 도메인 DNS 이름을 공개적으로 확인할 수 있습니다. 이렇게 하려면 자체 관리형 OpenSearch 또는 Elasticsearch를 소스로, OpenSearch Service 또는 OpenSearch Serverless를 대상으로 하는 OpenSearch Ingestion 파이프라인을 설정합니다. 이렇게 하면 자체 관리형 소스 클러스터에서 AWS관리형 대상 도메인 또는 컬렉션으로 데이터를 효과적으로 마이그레이션할 수 있습니다.

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. 퍼블릭 DNS 이름을 마이그레이션하고 구성하려는 데이터가 포함된 자체 관리형 OpenSearch 또는 Elastisearch 클러스터를 생성합니다. 지침은 OpenSearch 설명서의 클러스터 생성을 참조하세요.
2. 데이터를 마이그레이션할 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성합니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 및 the section called “컬렉션 생성” 섹션을 참조하세요.
3. 를 사용하여 자체 관리형 클러스터에 인증을 설정합니다 AWS Secrets Manager. AWS Secrets Manager 시크릿 교체의 단계를 수행하여 시크릿 교체를 활성화합니다.

4. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 자체 관리형 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

   다음 샘플 도메인 액세스 정책은 다음 단계에서 생성하는 파이프라인 역할을 사용하여 도메인에 데이터를 쓰도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   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"
         ]
       }
     ]
   }
   ```

   컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 섹션을 참조하세요the section called “역할 및 사용자 설정”.

1단계: 파이프라인 역할 구성

OpenSearch 파이프라인 사전 조건을 설정한 후 파이프라인 구성에 사용할 파이프라인 역할을 구성하고 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션에 쓸 수 있는 권한과 Secrets Manager에서 시크릿을 읽을 수 있는 권한을 추가합니다.

2단계: 파이프라인 생성

이제 OpenSearch를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

여러 OpenSearch Service 도메인을 데이터의 대상으로 지정할 수 있습니다. 이 기능을 사용하여 조건부로 라우팅하거나 수신 데이터를 여러 OpenSearch Service 도메인으로 복제할 수 있습니다.

소스 OpenSearch 또는 Elasticsearch 클러스터의 데이터를 OpenSearch Serverless VPC 컬렉션으로 마이그레이션할 수도 있습니다. 파이프라인 구성 내에서 네트워크 액세스 정책을 제공해야 합니다.

```
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
```

사전 구성된 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

VPC의 OpenSearch 클러스터에서 데이터 마이그레이션

OpenSearch Ingestion 파이프라인을 사용하여 VPC에서 실행되는 자체 관리형 OpenSearch 또는 Elasticsearch 클러스터에서 데이터를 마이그레이션할 수도 있습니다. 이렇게 하려면 자체 관리형 OpenSearch 또는 Elasticsearch를 소스로, OpenSearch Service 또는 OpenSearch Serverless를 대상으로 하는 OpenSearch Ingestion 파이프라인을 설정합니다. 이렇게 하면 자체 관리형 소스 클러스터에서 AWS관리형 대상 도메인 또는 컬렉션으로 데이터를 효과적으로 마이그레이션할 수 있습니다.

사전 조건

OpenSearch Ingestion 파이프라인을 생성하기 전에 다음 단계를 수행하세요.

1. 마이그레이션하려는 데이터가 포함된 VPC 네트워크 구성을 사용하여 자체 관리형 OpenSearch 또는 Elastisearch 클러스터를 생성합니다.
2. 데이터를 마이그레이션할 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성합니다. 자세한 내용은 OpenSearch Service 도메인 생성 및 컬렉션 생성을 참조하세요.
3. 를 사용하여 자체 관리형 클러스터에 인증을 설정합니다 AWS Secrets Manager. AWS Secrets Manager 시크릿 교체의 단계를 수행하여 시크릿 교체를 활성화합니다.
4. 자체 관리형 OpenSearch 또는 Elasticsearch에 대한 액세스 권한을 보유한 VPC의 ID를 가져옵니다. OpenSearch Ingestion에서 사용할 VPC CIDR을 선택합니다.

> **Note**
>
> 를 사용하여 파이프라인 AWS Management Console 을 생성하는 경우 자체 관리형 OpenSearch 또는 Elasticsearch를 사용하려면 OpenSearch Ingestion 파이프라인도 VPC에 연결해야 합니다. 이렇게 하려면 소스 네트워크 옵션 섹션을 찾아 VPC에 연결 확인란을 선택하고 제공된 기본 옵션 중 하나에서 CIDR을 선택합니다. RFC 1918 Best Current Practice에 정의된 대로 프라이빗 주소 공간에서 모든 CIDR을 사용할 수 있습니다.

사용자 지정 CIDR을 제공하려면 드롭다운 메뉴에서 기타를 선택합니다. OpenSearch Ingestion과 자체 관리형 OpenSearch 간의 IP 주소 충돌을 방지하려면 자체 관리형 OpenSearch VPC CIDR이 OpenSearch Ingestion의 CIDR과 달라야 합니다.

5. 리소스 기반 정책을 도메인에 연결하거나 데이터 액세스 정책을 컬렉션에 연결합니다. 이러한 액세스 정책을 통해 OpenSearch Ingestion은 자체 관리형 클러스터의 데이터를 도메인 또는 컬렉션에 쓸 수 있습니다.

   다음 샘플 도메인 액세스 정책은 다음 단계에서 생성하는 파이프라인 역할을 사용하여 도메인에 데이터를 쓰도록 허용합니다. 자체 ARN으로 `resource`을 업데이트해야 합니다.

   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:account-id:domain/domain-name"
         ]
       }
     ]
   }
   ```

   컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 섹션을 참조하세요the section called “역할 및 사용자 설정”.

1단계: 파이프라인 역할 구성

파이프라인 사전 조건을 설정한 후 파이프라인 구성에서 사용하려는 파이프라인 역할을 구성하고 역할에서 다음 권한을 추가합니다.

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"
                    }
            }
        }
    ]
}
```

파이프라인은 이러한 권한을 사용하여 VPC에서 네트워크 인터페이스를 생성하고 삭제하므로 OpenSearch Ingestion 파이프라인을 생성하는 데 사용하는 IAM 역할에서 위의 Amazon EC2 권한을 제공해야 합니다. 파이프라인은 오직 이 네트워크 인터페이스를 통해 OpenSearch 클러스터에 액세스할 수 있습니다.

2단계: 파이프라인 생성

이제 OpenSearch를 소스로 지정하는 다음과 같은 OpenSearch Ingestion 파이프라인을 구성할 수 있습니다.

여러 OpenSearch Service 도메인을 데이터의 대상으로 지정할 수 있습니다. 이 기능을 사용하여 조건부로 라우팅하거나 수신 데이터를 여러 OpenSearch Service 도메인으로 복제할 수 있습니다.

소스 OpenSearch 또는 Elasticsearch 클러스터의 데이터를 OpenSearch Serverless VPC 컬렉션으로 마이그레이션할 수도 있습니다. 파이프라인 구성 내에서 네트워크 액세스 정책을 제공해야 합니다.

```
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
```

사전 구성된 블루프린트를 사용하여 이 파이프라인을 생성할 수 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

Amazon Kinesis Data Streams에서 OpenSearch Ingestion 파이프라인 사용

Amazon Kinesis Data Streams와 함께 OpenSearch Ingestion 파이프라인을 사용하여 여러 스트림의 스트림 레코드 데이터를 Amazon OpenSearch Service 도메인 및 컬렉션으로 수집합니다. OpenSearch Ingestion 파이프라인은 스트리밍 수집 인프라를 통합하여 Kinesis에서 스트림 레코드를 지속적으로 수집할 수 있는 대기 시간이 짧은 대규모 방법을 제공합니다.

주제

- 소스로서의 Amazon Kinesis Data Streams
- Amazon Kinesis Data Streams 교차 계정을 소스로 사용

소스로서의 Amazon Kinesis Data Streams

다음 절차에서는 Amazon Kinesis Data Streams를 데이터 소스로 사용하는 OpenSearch Ingestion 파이프라인을 설정하는 방법을 알아봅니다. 이 섹션에서는 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 생성하고 파이프라인 역할을 구성하고 파이프라인을 생성하는 단계를 안내하는 등 필요한 사전 조건을 다룹니다.

사전 조건

파이프라인을 설정하려면 활성 Kinesis Data Streams가 하나 이상 필요합니다. 이러한 스트림은 레코드를 수신하거나 다른 소스에서 레코드를 수신할 준비가 되어 있어야 합니다. 자세한 내용은 OpenSearch Ingestion 개요를 참조하세요.

파이프라인을 설정하려면

1. OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션 생성

   도메인 또는 컬렉션을 생성하려면 OpenSearch Ingestion 시작하기를 참조하세요.

   컬렉션 또는 도메인에 쓰기 데이터에 액세스할 수 있는 올바른 권한이 있는 IAM 역할을 생성하려면 리소스 기반 정책을 참조하세요.
2. 권한을 사용하여 파이프라인 역할 구성

   파이프라인 구성에 사용할 파이프라인 역할을 설정하고 다음 권한을 추가합니다. ## ###를 자신의 정보로 바꿉니다.

   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:AWS ##:account-id:stream/stream-name"
            ]
        }
    ]
}
```

스트림에서 서버 측 암호화가 활성화된 경우 다음 AWS KMS 정책은가 레코드를 해독하도록 허용합니다. *## ###*를 자신의 정보로 바꿉니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowDecryptionOfCustomManagedKey",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kinesis:AWS ##:account-id:key/key-id"
        }
    ]
}
```

파이프라인이 도메인에 데이터를 쓰려면 도메인에 sts_role_arn 파이프라인 역할이 도메인에 액세스할 수 있도록 허용하는 도메인 수준 액세스 정책이 있어야 합니다.

다음 예제는 이전 단계(pipeline-role)에서 생성된 파이프라인 역할이 도메인에 데이터를 쓸 수 있도록 허용하는 ingestion-domain 도메인 액세스 정책입니다. *## ###*를 자신의 정보로 바꿉니다.

```
{
  "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. 파이프라인 생성

Kinesis-data-streams를 소스로 지정하여 OpenSearch Ingestion 파이프라인을 구성합니다. 이러한 파이프라인을 생성하기 위해 OpenSearch Ingestion 콘솔에서 사용할 수 있는 준비된 블루프린트를 찾을 수 있습니다. (선택 사항)를 사용하여 파이프라인을 생성하려면 "**AWS-KinesisDataStreamsPipeline**"라는 블루프린트를 사용할 AWS CLI수 있습니다. *## ###*를 자신의 정보로 바꿉니다.

```
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"
```

구성 옵션

Kinesis 구성 옵션은 OpenSearch 설명서의 구성 옵션을 참조하세요.

사용 가능한 메타데이터 속성

- stream_name - 레코드가 수집된 Kinesis Data Streams의 이름입니다.
- partition_key - 수집 중인 Kinesis Data Streams 레코드의 파티션 키
- sequence_number – 수집 중인 Kinesis Data Streams 레코드의 시퀀스 번호
- sub_sequence_number – 수집 중인 Kinesis Data Streams 레코드의 하위 시퀀스 번호

4. (선택 사항) Kinesis Data Streams 파이프라인에 대한 권장 컴퓨팅 유닛(OCUs) 구성

   OpenSearch Kinesis Data Streams 소스 파이프라인은 둘 이상의 스트림에서 스트림 레코드를 수집하도록 구성할 수도 있습니다. 수집 속도를 높이려면 추가된 새 스트림당 컴퓨팅 단위를 추가하는 것이 좋습니다.

데이터 일관성

OpenSearch Ingestion은 데이터 내구성을 보장하는 엔드 투 엔드 승인을 지원합니다. 파이프라인은 Kinesis에서 스트림 레코드를 읽을 때 스트림과 연결된 샤드를 기반으로 스트림 레코드를 읽는 작업을 동적으로 분산합니다. 파이프라인은 OpenSearch 도메인 또는 컬렉션의 모든 레코드를 수집한 후 승인을 받으면 스트림을 자동으로 체크포인트합니다. 이렇게 하면 스트림 레코드의 중복 처리가 방지됩니다.

스트림 이름을 기반으로 인덱스를 생성하려면 오픈서치 싱크 섹션의 인덱스를 "index_${getMetadata(\"stream_name\")}"로 정의합니다.

Amazon Kinesis Data Streams 교차 계정을 소스로 사용

OpenSearch Ingestion 파이프라인이 다른 계정의 Kinesis Data Streams에 소스로 액세스할 수 있도록 Amazon Kinesis Data Streams를 사용하여 계정 간에 액세스 권한을 부여할 수 있습니다. 교차 계정 액세스를 활성화하려면 다음 단계를 완료하세요.

교차 계정 액세스 구성

1. Kinesis 스트림이 있는 계정에서 리소스 정책 설정

   ## ###를 자신의 정보로 바꿉니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "StreamReadStatementID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::osi-pipeline-account-ID:role/Pipeline-
Role"
            },
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:GetRecords",
                "kinesis:GetShardIterator",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:AWS ##:stream-account-id:stream/
stream-name"
        },
        {
            "Sid": "StreamEFOReadStatementID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::osi-pipeline-account-ID:role/Pipeline-
Role"
            },
            "Action": [
                "kinesis:DescribeStreamSummary",
                "kinesis:ListShards"
            ],
            "Resource": "Consumer ARN"
        }
    ]
}
```

2. (선택 사항) 소비자 및 소비자 리소스 정책 설정

   이는 선택적 단계이며 스트림 레코드 읽기에 향상된 팬아웃 소비자 전략을 사용하려는 경우에만 필요합니다. 자세한 내용은 전용 처리량으로 향상된 팬아웃 소비자 개발을 참조하세요.

   a. 소비자 설정

      기존 소비자를 재사용하려면이 단계를 건너뛸 수 있습니다. 자세한 내용은 Amazon Kinesis Data Streams API 참조의 RegisterStreamConsumer를 참조하세요. Amazon Kinesis

      다음 CLI 명령 예제에서 *## ### ##* 자신의 정보로 바꿉니다.

      Example CLI 명령 예제:

```
aws kinesis register-stream-consumer \
--stream-arn "arn:aws:kinesis:AWS ##:account-id:stream/stream-name" \
--consumer-name consumer-name
```

   b. 소비자 리소스 정책 설정

      다음 문에서 *## ### ##* 자신의 정보로 바꿉니다.

      JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConsumerEFOReadStatementID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::osi-pipeline-account-ID:role/
Pipeline-Role"
            },
            "Action": [
                "kinesis:DescribeStreamConsumer",
                "kinesis:SubscribeToShard"
            ],
```

```
            "Resource": "arn:aws::kinesis:AWS ##:customer-kinesis-stream-
account-ID:stream/stream-1/consumer-name:<>"
        }
    ]
}
```

3. 파이프라인 구성

   교차 계정 수집의 경우 각 스트림에 kinesis_data_streams 대해 아래에 다음 속성을 추가합니다.

   - stream_arn - 스트림이 있는 계정에 속한 스트림의 ARN
   - consumer_arn - 이는 선택적 속성이며 기본 향상된 팬아웃 소비자 전략을 선택한 경우 지정해야 합니다. 이 필드의 실제 소비자 ARN을 지정합니다. ## ###를 자신의 정보로 바꿉니다.

```
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 파이프라인 역할 Kinesis Data Streams

   a. IAM 정책

      파이프라인 역할에 다음 정책을 추가합니다. *## ###*를 자신의 정보로 바꿉니다.

      JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStreamConsumer",
                "kinesis:SubscribeToShard"
            ],
            "Resource": [
                "consumer ARN",
                "consumer ARN:*"
            ]
        },
        {
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:DescribeStreamSummary",
                "kinesis:GetRecords",
```

```
                "kinesis:GetShardIterator",
                "kinesis:ListShards",
                "kinesis:ListStreams",
                "kinesis:ListStreamConsumers",
                "kinesis:RegisterStreamConsumer"
            ],
            "Resource": [
                "stream ARN"
            ]
        }
    ]
}
```

b. 신뢰 정책

스트림 계정에서 데이터를 수집하려면 파이프라인 수집 역할과 스트림 계정 간에 신뢰 관계를 설정해야 합니다. 파이프라인 역할에 다음을 추가합니다. ## ###를 자신의 정보로 바꿉니다.

JSON

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

다음 단계

데이터를 파이프라인으로 내보낸 후 파이프라인의 싱크로 구성된 OpenSearch Service 도메인에서 데이터를 쿼리할 수 있습니다. 다음 리소스는 시작하는 데 도움이 됩니다.

- 관찰성

- the section called “Trace Analytics”
- the section called “파이프 처리 언어”

AWS Lambda와 함께 OpenSearch Ingestion 파이프라인 사용

AWS Lambda 프로세서를 사용하여 사용자 지정 코드를 사용하여 OpenSearch Ingestion에서 지원하는 모든 소스 또는 대상의 데이터를 보강합니다. Lambda 프로세서를 사용하면 자체 데이터 변환 또는 보강을 적용한 다음 처리된 이벤트를 파이프라인에 반환하여 추가 처리를 수행할 수 있습니다. 이 프로세서를 사용하면 사용자 지정 데이터 처리를 활성화하고 파이프라인을 통과하기 전에 데이터를 조작하는 방법을 완벽하게 제어할 수 있습니다.

> **Note**
>
> Lambda 프로세서에서 처리하는 단일 이벤트의 페이로드 크기 제한은 5MB입니다. 또한 Lambda 프로세서는 JSON 배열 형식의 응답만 지원합니다.

사전 조건

Lambda 프로세서로 파이프라인을 생성하기 전에 다음 리소스를 생성합니다.

- 소스 데이터를 보강하고 변환하는 AWS Lambda 함수입니다. 지침은 첫 번째 Lambda 함수 생성을 참조하세요.
- 파이프라인 싱크가 될 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션입니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 및 the section called “컬렉션 생성” 섹션을 참조하세요.
- 도메인 또는 컬렉션 싱크에 쓸 수 있는 권한이 포함된 파이프라인 역할입니다. 자세한 내용은 the section called “파이프라인 역할” 단원을 참조하십시오.

  파이프라인 역할에는 파이프라인 구성에 지정된 Lambda 함수를 호출할 수 있는 연결된 권한 정책도 필요합니다. 예:

  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"

        }
    ]
}
```

파이프라인 생성

를 프로세서 AWS Lambda 로 사용하려면 OpenSearch Ingestion 파이프라인을 구성하고를 프로세서aws_lambda로 지정합니다. 사용자 AWS Lambda 지정 보강 블루프린트를 사용하여 파이프라인을 생성할 수도 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

다음 예제 파이프라인은 HTTP 소스에서 데이터를 수신하고, 날짜 프로세서와 AWS Lambda 프로세서를 사용하여 데이터를 보강하고, 처리된 데이터를 OpenSearch 도메인에 수집합니다.

```
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
```

다음 예제 AWS Lambda 함수는 제공된 이벤트 배열의 각 요소에 새 키-값 페어("transformed": "true")를 추가하여 수신 데이터를 변환한 다음 수정된 버전을 다시 보냅니다.

```
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
```

배칭

파이프라인은 배치 이벤트를 Lambda 프로세서로 전송하고 배치 크기를 동적으로 조정하여 5MB 제한 미만으로 유지되도록 합니다.

다음은 파이프라인 배치의 예입니다.

```
batch:
    key_name: "events"

input_arrary = event.get('events', [])
```

> **Note**
>
> 파이프라인을 생성할 때 Lambda 프로세서 구성의 key_name 옵션이 Lambda 핸들러의 이벤트 키와 일치하는지 확인합니다.

조건부 필터링

조건부 필터링을 사용하면 AWS Lambda 프로세서가 이벤트 데이터의 특정 조건을 기반으로 Lambda 함수를 호출하는 시기를 제어할 수 있습니다. 이는 다른 이벤트를 무시하면서 특정 유형의 이벤트를 선택적으로 처리하려는 경우에 특히 유용합니다.

다음 예제 구성에서는 조건부 필터링을 사용합니다.

```
processors:
  - aws_lambda:
      function_name: "my-lambda-function"
      aws:
        region: "region"
      lambda_when: "/sourceIp == 10.10.10.10"
```

Amazon OpenSearch Ingestion을 사용하여 도메인과 컬렉션 간 데이터 마이그레이션

OpenSearch Ingestion 파이프라인을 사용하여 Amazon OpenSearch Service 도메인 또는 OpenSearch Serverless VPC 컬렉션 간에 데이터를 마이그레이션할 수 있습니다. 이를 수행하려면 한 도메인 또는 컬렉션을 소스로 구성하고 다른 도메인 또는 컬렉션을 싱크로 구성하는 파이프라인을 설정합니다. 그러면 한 도메인 또는 컬렉션에서 다른 도메인이나 컬렉션으로 데이터가 효과적으로 마이그레이션됩니다.

데이터를 마이그레이션하려면 다음 리소스가 있어야 합니다.

- 소스 OpenSearch Service 도메인 또는 소스 OpenSearch Serverless VPC 컬렉션. 이 도메인 또는 컬렉션에 마이그레이션하려는 데이터가 포함되어 있습니다. 도메인을 사용하는 경우 OpenSearch 1.0 이상 또는 Elasticsearch 7.4 이상이 실행되고 있어야 합니다. 도메인에는 파이프라인 역할에 적절한 권한을 부여하는 액세스 정책도 있어야 합니다.
- 데이터를 마이그레이션하려는 별도의 도메인 또는 VPC 컬렉션. 이 도메인 또는 컬렉션은 파이프라인 싱크로 작동합니다.
- OpenSearch Ingestion이 컬렉션이나 도메인에서 읽고 쓸 때 사용하는 파이프라인 역할. 파이프라인 구성에 이 역할의 Amazon 리소스 이름(ARN)을 포함합니다. 자세한 정보는 다음 자료를 참조하세요.
  - the section called “도메인에 대한 파이프라인 액세스 권한 부여”
  - the section called “컬렉션에 대한 액세스 권한을 파이프라인에 부여”

주제

- 제한 사항
- OpenSearch Service를 소스로 사용
- 여러 OpenSearch Service 도메인 싱크 지정
- OpenSearch Serverless VPC 컬렉션으로 데이터 마이그레이션

제한 사항

OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 싱크로 지정할 때 다음과 같은 제한 사항이 적용됩니다.

- 파이프라인은 둘 이상의 VPC 도메인에 쓸 수 없습니다.
- VPC 액세스를 사용하는 OpenSearch Serverless 컬렉션 사이에서만 데이터를 마이그레이션할 수 있습니다. 퍼블릭 컬렉션은 지원되지 않습니다.
- 단일 파이프라인 구성에서 VPC와 퍼블릭 도메인의 조합을 지정할 수 없습니다.
- 단일 파이프라인 구성 내에서 최대 20개의 비파이프라인 싱크를 보유할 수 있습니다.
- 단일 파이프라인 구성에서 최대 3개의 싱크를 지정할 수 AWS 리전 있습니다.
- 싱크가 여러 개 있는 파이프라인에서 싱크가 너무 오랫동안 중단된 상태이거나 수신 데이터를 처리할 수 있는 충분한 용량이 프로비저닝되지 않은 경우 시간이 지남에 따라 처리 속도가 저하될 수 있습니다.

OpenSearch Service를 소스로 사용

소스로 지정하는 도메인 또는 컬렉션이 데이터가 마이그레이션되는 소스 위치입니다.

IAM에서 파이프라인 역할 생성

OpenSearch Ingestion 파이프라인을 생성하려면 먼저 도메인 또는 컬렉션 사이에서 읽기 및 쓰기 액세스 권한을 부여하도록 파이프라인 역할을 생성해야 합니다. 이렇게 하려면 다음 단계를 수행하세요.

1. IAM에서 새 권한 정책을 생성하여 파이프라인 역할에 연결합니다. 소스에서 읽고 싱크에 쓸 수 있는 권한을 허용해야 합니다. OpenSearch Service 도메인의 IAM 파이프라인 권한 설정에 대한 자세한 내용은 the section called “도메인에 대한 파이프라인 액세스 권한 부여” 및 the section called “컬렉션에 대한 액세스 권한을 파이프라인에 부여” 섹션을 참조하세요.
2. 소스에서 읽을 수 있도록 파이프라인 역할에 다음 권한을 지정합니다.

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"
         ]
      }
   ]
}
```

파이프라인 생성

정책을 파이프라인 역할에 연결한 후 AWSOpenSearchDataMigrationPipeline 마이그레이션 블루프린트를 사용하여 파이프라인을 생성합니다. 이 블루프린트에는 OpenSearch Service 도메인 또는 컬렉션 사이에서 데이터를 마이그레이션하기 위한 기본 구성이 포함되어 있습니다. 자세한 내용은 the section called “블루프린트 작업” 단원을 참조하십시오.

> **Note**
>
> OpenSearch Ingestion에서는 소스 도메인 버전과 배포를 사용하여 마이그레이션에 사용할 메커니즘을 결정합니다. 일부 버전은 `point_in_time` 옵션을 지원합니다. OpenSearch Serverless는 `point_in_time` 또는 `scroll`을 지원하지 않으므로 `search_after` 옵션을 사용합니다.

마이그레이션 프로세스 중에 새 인덱스가 생성되거나, 마이그레이션이 진행되는 동안 문서가 업데이트될 수 있습니다. 이 때문에 새 데이터나 업데이트된 데이터를 찾기 위해 도메인 인덱스 데이터의 단일 스캔이나 다중 스캔을 수행해야 할 수 있습니다.

파이프라인 구성에서 `index_read_count` 및 `interval`을 구성하여 스캔 실행 횟수를 지정합니다. 다음 예제에서는 다중 스캔을 수행하는 방법을 보여줍니다.

```
scheduling:
    interval: "PT2H"
    index_read_count: 3
    start_time: "2023-06-02T22:01:30.00Z"
```

OpenSearch Ingestion은 다음 구성을 사용하여 데이터를 동일한 인덱스에 쓰고 동일한 문서 ID를 유지하도록 합니다.

```
index: "${getMetadata(\"opensearch-index\")}"
document_id: "${getMetadata(\"opensearch-document_id\")}"
```

여러 OpenSearch Service 도메인 싱크 지정

여러 퍼블릭 OpenSearch Service 도메인을 데이터의 대상으로 지정할 수 있습니다. 이 기능을 사용하여 조건부 라우팅을 수행하거나 수신 데이터를 여러 OpenSearch Service 도메인으로 복제할 수 있습니다. 최대 10개의 서로 다른 퍼블릭 OpenSearch Service 도메인을 싱크로 지정할 수 있습니다.

다음 예제에서는 수신 데이터를 조건에 따라 서로 다른 OpenSearch Service 도메인으로 라우팅합니다.

```
...
  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
```

OpenSearch Serverless VPC 컬렉션으로 데이터 마이그레이션

OpenSearch Ingestion을 사용하여 Amazon OpenSearch Service 도메인 또는 OpenSearch Serverless VPC 컬렉션에서 VPC 컬렉션 싱크에서 데이터를 마이그레이션할 수 있습니다. 파이프라인 구성 내에서 네트워크 액세스 정책을 제공해야 합니다. OpenSearch Serverless VPC 컬렉션으로 데이터를 수집하는 방법에 대한 자세한 내용은 the section called “튜토리얼: 컬렉션에 데이터 수집” 섹션을 참조하세요.

VPC 컬렉션으로 데이터를 마이그레이션하는 방법

1. OpenSearch Serverless 컬렉션을 생성합니다. 지침은 the section called “튜토리얼: 컬렉션에 데이터 수집” 섹션을 참조하세요.
2. 컬렉션 엔드포인트와 대시보드 엔드포인트 모두에 VPC 액세스 권한을 지정하는 컬렉션에 대한 네트워크 정책을 생성합니다. 지침은 the section called “네트워크 액세스” 섹션을 참조하세요.
3. 아직 없는 경우 파이프라인 역할을 생성합니다. 지침은 the section called “파이프라인 역할” 섹션을 참조하세요.
4. 파이프라인을 생성합니다. 지침은 the section called “블루프린트 작업” 섹션을 참조하세요.

AWS SDKs를 사용하여 Amazon OpenSearch Ingestion과 상호 작용

이 섹션에는 AWS SDKs를 사용하여 Amazon OpenSearch Ingestion과 상호 작용하는 방법의 예가 포함되어 있습니다. 코드 예제는 도메인과 파이프라인을 생성한 다음 파이프라인으로 데이터를 수집하는 방법을 보여줍니다.

주제

- Python

Python

다음 샘플 스크립트는 AWS SDK for Python (Boto3)를 사용하여 IAM 파이프라인 역할, 데이터를 쓸 도메인, 데이터를 수집하는 파이프라인을 생성합니다. 그런 다음 `requests` HTTP 라이브러리를 사용하여 샘플 로그 파일을 파이프라인으로 수집합니다.

필요한 종속성을 설치하려면 다음 명령을 실행합니다.

```
pip install boto3
pip install botocore
pip install requests
pip install requests-auth-aws-sigv4
```

스크립트 내에서의 모든 인스턴스를 ID*account-id*로 바꿉니다 AWS 계정 .

```
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 Ingestion의 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 클라우드에서 AWS AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 OpenSearch Ingestion 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 OpenSearch Ingestion을 구성하는 방법을 보여줍니다. 또한 OpenSearch Ingestion 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- Amazon OpenSearch Ingestion 파이프라인에 대한 VPC 액세스 구성
- Amazon OpenSearch Ingestion에 대한 자격 증명 및 액세스 관리자

- AWS CloudTrail(을)를 사용하여 Amazon OpenSearch Ingestion API 직접 호출 로깅

Amazon OpenSearch Ingestion 파이프라인에 대한 VPC 액세스 구성

인터페이스 VPC 엔드포인트를 사용하여 Amazon OpenSearch Ingestion 파이프라인에 액세스할 수 있습니다. VPC는 전용 가상 네트워크입니다 AWS 계정. AWS 클라우드의 다른 가상 네트워크와 논리적으로 격리되어 있습니다. VPC 엔드포인트를 통해 파이프라인에 액세스하면 인터넷 게이트웨이, NAT 디바이스 또는 VPN을 연결하지 않아도 VPC 내부에서 OpenSearch Ingestion과 다른 서비스 간에 보안 통신이 가능합니다. 모든 트래픽은 AWS 클라우드 내에서 안전하게 유지됩니다.

OpenSearch Ingestion은 AWS PrivateLink에서 지원하는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 파이프라인 생성 중에 지정한 각 서브넷에서 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 OpenSearch Ingestion 파이프라인으로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다. 인터페이스 엔드포인트를 직접 생성하고 관리하도록 선택할 수도 있습니다.

VPC를 사용하면 공용 인터넷을 통하지 않고 VPC의 경계 내에서 OpenSearch Ingestion 파이프라인을 통해 데이터 흐름을 적용할 수 있습니다. VPC 내에 있지 않은 파이프라인은 공용 엔드포인트와 인터넷을 통해 데이터를 보내고 받습니다.

VPC 액세스 권한이 있는 파이프라인은 퍼블릭 또는 VPC OpenSearch Service 도메인과 퍼블릭 또는 VPC OpenSearch Serverless 컬렉션에 쓸 수 있습니다.

주제

- 고려 사항
- 제한 사항
- 사전 조건
- 파이프라인에 대한 VPC 액세스 구성
- 자체 관리형 VPC 엔드포인트
- VPC 액세스를 위한 서비스 연결 역할

고려 사항

파이프라인에 대한 VPC 액세스를 구성할 때 다음 사항을 고려하세요.

- 파이프라인은 싱크와 동일한 VPC에 있지 않아도 됩니다. 또한 두 VPC 간에 연결을 설정할 필요도 없습니다. OpenSearch Ingestion이 이들을 연결해 줍니다.

- 파이프라인에는 하나의 VPC만 지정할 수 있습니다.
- 퍼블릭 파이프라인과 달리 VPC 파이프라인은 쓰기 대상 도메인 또는 컬렉션 싱크와 동일한 AWS 리전 에 있어야 합니다.
- 파이프라인을 사용자의 VPC의 서브넷 1개, 2개 또는 3개에 배포하도록 선택할 수 있습니다. 서브넷은 통합 OpenSearch Compute Units(OCU)가 배포된 동일한 가용 영역에 분산되어 있습니다.
- 하나의 서브넷에만 파이프라인을 배포하고 가용 영역이 다운되면 데이터를 수집할 수 없습니다. 고가용성을 보장하려면 2개 또는 3개의 서브넷으로 파이프라인을 구성하는 것이 좋습니다.
- 보안 그룹 지정은 선택 사항입니다. 보안 그룹을 제공하지 않는 경우 OpenSearch Ingestion은 VPC에서 지정된 기본 보안 그룹을 사용합니다.

제한 사항

VPC 내 파이프라인에는 다음과 같은 제한 사항이 있습니다.

- 파이프라인 네트워크 구성을 생성한 후에는 해당 구성을 변경할 수 없습니다. VPC 내에서 파이프라인을 시작하는 경우 나중에 퍼블릭 엔드포인트로 변경할 수 없으며 그 반대의 경우도 마찬가지입니다.
- 인터페이스 VPC 엔드포인트 또는 퍼블릭 엔드포인트에서 파이프라인을 시작할 수도 있지만 두 방법을 동시에 사용할 수는 없습니다. 파이프라인을 만들 때 한 가지를 선택해야 합니다.
- VPC 액세스 권한이 있는 파이프라인을 프로비저닝한 후 다른 VPC로 이동할 수 없지만 해당 서브넷과 보안 그룹 설정은 변경할 수 있습니다.
- 파이프라인이 VPC 액세스를 사용하는 도메인 또는 컬렉션 싱크에 쓰는 경우, 파이프라인이 생성된 후에는 나중에 돌아가서 싱크(VPC 또는 퍼블릭)를 변경할 수 없습니다. 파이프라인을 삭제하고 새 싱크로 재생성해야 합니다. 여전히 퍼블릭 싱크에서 VPC 액세스를 사용하는 싱크로 전환할 수 있습니다.
- VPC 파이프라인에 계정 간 수집 액세스를 제공할 수 없습니다.

사전 조건

VPC 액세스 권한이 있는 파이프라인을 프로비저닝하려면 먼저 다음을 수행해야 합니다.

- VPC 생성

  VPC를 생성하려면 Amazon VPC 콘솔, AWS CLI 또는 AWS SDKs. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 작업을 참조하세요. VPC가 이미 있는 경우에는 이 단계를 건너뛸 수 있습니다.
- IP 주소 예약

OpenSearch Ingestion은 파이프라인 생성 중에 지정한 각 서브넷에 탄력적 네트워크 인터페이스를 배치합니다. 각 네트워크 인터페이스에는 IP 주소가 연결됩니다. 서브넷당 네트워크 인터페이스용 IP 주소 하나를 예약해야 합니다.

파이프라인에 대한 VPC 액세스 구성

OpenSearch Service 콘솔 내에서 또는 AWS CLI를 사용한 파이프라인에 대한 VPC 액세스 기능을 활성화할 수 있습니다.

콘솔

파이프라인 생성 중에 VPC 액세스를 구성합니다. 소스 네트워크 옵션에서 VPC 액세스를 선택하고 다음 설정을 구성합니다.

설정	설명
엔드포인트 관리	VPC 엔드포인트를 직접 생성할지 아니면 OpenSearch Ingestion에서 자동으로 생성할지 선택합니다.
VPC	사용하려는 Virtual Private Cloud(VPC)를 선택합니다. VPC와 파이프라인의 AWS 리전(은)는 동일해야 합니다.
서브넷	서브넷을 하나 이상 선택합니다. OpenSearch Service는 VPC 엔드포인트와 탄력적 네트워크 인터페이스를 서브넷에 배치합니다.
보안 그룹	필요한 애플리케이션이 파이프라인에 의해 노출된 포트(80 또는 443) 및 프로토콜(HTTP 또는 HTTPS)에서 OpenSearch Ingestion 파이프라인에 도달하도록 허용하는 VPC 보안 그룹을 하나 이상 선택합니다.
VPC 연결 옵션	소스에 Amazon DocumentDB, 자체 관리형 OpenSearch 또는 Confluent Kafka와 같은 VPC 간 통신이 필요한 경우 OpenSearch Ingestion은 이러한 소스에 연결하기 위해 지정한 서브넷에 탄력적 네트워크 인터페이스(ENIs)를 생성합니다. OpenSearch Ingestion은 각 가용 영역의 ENIs를 사용하여 지정된 소스에 도달합니다. VPC에 연결 옵션은 OpenSearch Ingestion 데이터 영역 VPC를 지정된 VPC에 연결합니다. 관리형 VPC에 대한 CIDR 예약을 선택하여 네트워크 인터페이스를 배포합니다.

CLI

를 사용하여 VPC 액세스를 구성하려면 `--vpc-options` 파라미터를 AWS CLI지정합니다.

```
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"
```

자체 관리형 VPC 엔드포인트

파이프라인을 생성할 때 엔드포인트 관리를 사용하여 자체 관리형 엔드포인트 또는 서비스 관리형 엔드포인트가 있는 파이프라인을 생성할 수 있습니다. 엔드포인트 관리는 선택 사항이며, 기본적으로 OpenSearch Ingestion에서 관리하는 엔드포인트로 설정됩니다.

에서 자체 관리형 VPC 엔드포인트가 있는 파이프라인을 생성하려면 OpenSearch Service 콘솔을 사용하여 파이프라인 생성을 AWS Management Console참조하세요. OpenSearch 에서 자체 관리형 VPC 엔드포인트가 있는 파이프라인을 생성하려면 create-pipeline 명령에서 `--vpc-options` 파라미터를 사용할 AWS CLI수 있습니다.

```
--vpc-options SubnetIds=subnet-abcdef01234567890,VpcEndpointManagement=CUSTOMER
```

엔드포인트 서비스를 지정할 때 파이프라인에 대해 엔드포인트를 직접 생성할 수 있습니다. 엔드포인트 서비스를 찾으려면 다음과 유사한 응답을 반환하는 get-pipeline 명령을 사용합니다.

```
"vpcEndpointService" : "com.amazonaws.osis.us-east-1.pipeline-
id-1234567890abcdef1234567890",
"vpcEndpoints" : [
  {
    "vpcId" : "vpc-1234567890abcdef0",
    "vpcOptions" : {
      "subnetIds" : [ "subnet-abcdef01234567890", "subnet-021345abcdef6789" ],
      "vpcEndpointManagement" : "CUSTOMER"
    }
  }
```

응답`vpcEndpointService`의를 사용하여 AWS Management Console 또는를 사용하여 VPC 엔드포인트를 생성합니다 AWS CLI.

자체 관리형 VPC 엔드포인트를 사용하는 경우 VPC에서 DNS 속성 `enableDnsSupport` 및 `enableDnsHostnames`를 활성화해야 합니다. 중지 후 다시 시작하는 자체 관리형 엔드포인트를 포함하는 파이프라인가 있으면 계정에서 VPC 엔드포인트를 다시 생성해야 합니다.

VPC 액세스를 위한 서비스 연결 역할

서비스 연결 역할은 서비스가 사용자를 대신하여 리소스를 생성하고 관리할 수 있도록 서비스에 권한을 위임하는 고유한 유형의 IAM 역할입니다. 서비스 관리형 VPC 엔드포인트를 선택하는 경우 OpenSearch Ingestion에서 VPC에 액세스하고, 파이프라인 엔드포인트를 생성하며, VPC의 서브넷에 네트워크 인터페이스를 배치하려면 AWSServiceRoleForAmazonOpenSearchIngestionService라고 하는 서비스 연결 역할이 필요합니다.

자체 관리형 VPC 엔드포인트를 선택하는 경우 OpenSearch Ingestion에는 AWSServiceRoleForOpensearchIngestionSelfManagedVpce라고 하는 서비스 연결 역할이 필요합니다. 이 역할, 해당 권한 및 삭제 방법에 대한 자세한 내용은 the section called “파이프라인 생성 역할” 섹션을 참조하세요.

수집 파이프라인을 생성할 때 OpenSearch Ingestion이 자동으로 역할을 생성합니다. 이 자동 생성이 성공하려면 계정에서 첫 번째 파이프라인을 생성하는 사용자에게 `iam:CreateServiceLinkedRole` 작업에 대한 권한이 있어야 합니다. 자세히 알아보려면 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요. 역할을 생성한 후 AWS Identity and Access Management (IAM) 콘솔에서 역할을 볼 수 있습니다.

Amazon OpenSearch Ingestion에 대한 자격 증명 및 액세스 관리자

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 어떤 사용자가 OpenSearch Ingestion 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- OpenSearch Ingestion에 대한 자격 증명 기반 정책
- OpenSearch Ingestion의 정책 작업
- OpenSearch Ingestion의 정책 리소스
- Amazon OpenSearch Ingestion의 정책 조건 키
- OpenSearch Ingestion을 통한 ABAC
- OpenSearch Ingestion에서 임시 보안 인증 정보 사용

- OpenSearch Ingestion에 대한 서비스 연결 역할
- OpenSearch Ingestion에 대한 자격 증명 기반 정책의 예

OpenSearch Ingestion에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 위탁자가 연결된 사용자 또는 역할에 적용되므로 위탁자를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

OpenSearch Ingestion에 대한 자격 증명 기반 정책의 예

OpenSearch Ingestion 자격 증명 기반 정책의 예를 보려면 the section called “자격 증명 기반 정책 예제”을(를) 참조하세요.

OpenSearch Ingestion의 정책 작업

정책 작업 지원: 예

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

OpenSearch Ingestion의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
osis
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
```

```
      "osis:action1",
      "osis:action2"
         ]
```

와일드카드 문자(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, List라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

```
"Action": "osis:List*"
```

OpenSearch Ingestion 자격 증명 기반 정책의 예를 보려면 OpenSearch Serverless에 대한 자격 증명 기반 정책 예시을(를) 참조하세요.

OpenSearch Ingestion의 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 문에는 Resource또는 NotResource요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

Amazon OpenSearch Ingestion의 정책 조건 키

서비스별 정책 조건 키 지원: 아니요

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Condition 요소(또는 Condition 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition 요소를 지정하거나 단일 Condition 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 AND 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 OR 작업을 사용하여 조건을 AWS 평가합니다. 문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

OpenSearch Ingestion 조건 키 목록을 보려면 서비스 승인 참조의 Amazon OpenSearch Ingestion에 대한 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스에 대해 알아보려면 Amazon OpenSearch Ingestion에서 정의한 작업을 참조하세요.

OpenSearch Ingestion을 통한 ABAC

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. 에서는 AWS이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연결할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 위탁자의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 aws:ResourceTag/*key-name*, aws:RequestTag/*key-name* 또는 aws:TagKeys 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

OpenSearch Ingestion 리소스 태그 지정에 대한 자세한 내용은 the section called “파이프라인 태그 지정” 섹션을 참조하세요.

OpenSearch Ingestion에서 임시 보안 인증 정보 사용

임시 자격 증명 지원: 예

일부 AWS 서비스 는 임시 자격 증명을 사용하여 로그인할 때 작동하지 않습니다. 임시 자격 증명으로 AWS 서비스 작업하는를 포함한 추가 정보는 AWS 서비스 IAM 사용 설명서의 IAM으로 작업하는 섹션을 참조하세요.

사용자 이름 및 암호를 제외한 방법을 AWS Management Console 사용하여에 로그인하는 경우 임시 자격 증명을 사용합니다. 예를 들어 회사의 SSO(Single Sign-On) 링크를 AWS 사용하여에 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 자격 증명을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다. 그런 다음 이러한 임시 자격 증명을 사용하여 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 access AWS. AWS recommends에 액세스할 수 있습니다. 자세한 정보는 IAM의 임시 보안 자격 증명 섹션을 참조하세요.

OpenSearch Ingestion에 대한 서비스 연결 역할

서비스 링크 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

OpenSearch Ingestion은 `AWSServiceRoleForAmazonOpenSearchIngestionService`라는 서비스 연결 역할을 사용합니다. `AWSServiceRoleForOpensearchIngestionSelfManagedVpce`라는 서비스 연결 역할은 자체 관리형 VPC 엔드포인트가 있는 파이프라인에서도 사용할 수 있습니다. OpenSearch Ingestion 서비스 연결 역할 생성 및 관리에 대한 자세한 내용은 the section called “파이프라인 생성 역할” 섹션을 참조하세요.

OpenSearch Ingestion에 대한 자격 증명 기반 정책의 예

기본적으로 사용자 및 역할에는 OpenSearch Ingestion 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN의 형식을 포함하여 Amazon OpenSearch Ingestion에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 Amazon OpenSearch Ingestion에 대한 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- 콘솔에서 OpenSearch Ingestion 사용
- OpenSearch Ingestion 파이프라인 관리
- OpenSearch Ingestion 파이프라인으로 데이터 수집

정책 모범 사례

자격 증명 기반 정책은 매우 강력합니다. 이 정책은 계정에서 사용자가 OpenSearch Ingestion 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

ID 기반 정책에 따라 계정에서 사용자가 OpenSearch Ingestion 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 AWS CloudFormation. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.

- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

콘솔에서 OpenSearch Ingestion 사용

OpenSearch Service 콘솔 내에서 OpenSearch Ingestion에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은 AWS 계정의 OpenSearch Ingestion 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하면 콘솔이 해당 정책이 있는 개체(예: IAM 역할)에 대해 의도한 대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 그 대신 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

다음 정책은 사용자가 OpenSearch Service 콘솔 내에서 OpenSearch Ingestion에 액세스할 수 있도록 허용합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ]
        }
```

```
    ]
}
```

또는의 모든 OpenSearch Ingestion 리소스에 대한 읽기 전용 액세스 권한을 부여하는 the section called “AmazonOpenSearchIngestionReadOnlyAccess” AWS 관리형 정책을 사용할 수 있습니다 AWS 계정.

OpenSearch Ingestion 파이프라인 관리

이 정책은 사용자가 Amazon OpenSearch Ingestion 파이프라인을 관리할 수 있도록 하는 파이프라인 관리자’ 정책의 예시입니다. 사용자는 파이프라인을 생성, 확인 및 삭제할 수 있습니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:us-east-1::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"
        }
    ]
```

```
}
```

OpenSearch Ingestion 파이프라인으로 데이터 수집

이 예시 정책을 통해 사용자 또는 기타 엔티티는 계정의 Amazon OpenSearch Ingestion 파이프라인으로 데이터를 수집할 수 있습니다. 사용자는 파이프라인을 수정할 수 없습니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:us-east-1:123456789012:pipeline/*",
            "Action": [
                "osis:Ingest"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CloudTrail(을)를 사용하여 Amazon OpenSearch Ingestion API 직접 호출 로깅

Amazon OpenSearch Ingestion은 OpenSearch Ingestion에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다.

CloudTrail은 OpenSearch Ingestion에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. 캡처된 호출에는 OpenSearch Service 콘솔의 OpenSearch Ingestion 섹션에서의 호출과 OpenSearch Ingestion API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면 OpenSearch Ingestion 에 대한 이벤트를 포함한 CloudTrail 이벤트를 지속해서 Amazon S3 버킷에 배포할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여 OpenSearch Ingestion 에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 OpenSearch Ingestion 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. OpenSearch Ingestion에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

OpenSearch Ingestion에 대한 이벤트를 AWS 계정포함하여에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다.

추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

모든 OpenSearch Ingestion 작업은 CloudTrail에 의해 기록되며 OpenSearch Serverless API 참조에 문서화됩니다. 예를 들어 `CreateCollection`, `ListCollections` 및 `DeleteCollection` 작업을 직접적으로 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명 정보를 이용하면 다음을 판단하는 데 도움이 됩니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

OpenSearch Ingestion 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다.

이벤트는 모든 소스로부터 단일 요청을 나타냅니다. 요청된 작업, 모든 파라미터, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 `DeletePipeline` 작업을 보여주는 CloudTrail 로그 항목이 나타낸 예시입니다.


```
{
    "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 Ingestion 파이프라인 태그 지정

태그를 사용하면 Amazon OpenSearch Service 도메인에 임의 정보를 할당할 수 있으므로 해당 정보를 분류하고 필터링할 수 있습니다. 태그는 사용자가 할당하거나가 AWS 리소스에 AWS 할당하는 메타데이터 레이블입니다. 각 태그는 키와 값으로 구성됩니다. 사용자가 할당하는 태그에 대해 키와 값을 정의합니다. 예를 들어 키를 `stage`로 정의하고 리소스 하나의 값을 `test`로 정의할 수 있습니다.

태그는 다음을 지원합니다.

- AWS 리소스를 식별하고 구성합니다. 많은 AWS 서비스가 태그 지정을 지원하므로 다른 서비스의 리소스에 동일한 태그를 할당하여 리소스가 관련이 있음을 나타낼 수 있습니다. 예를 들어 Amazon OpenSearch Service 도메인에 할당하는 것과 동일한 태그를 OpenSearch Serverless 컬렉션에 할당할 수 있습니다.
- AWS 비용을 추적합니다. AWS 결제 및 비용 관리 대시보드에서 이러한 태그를 활성화합니다.는 태그를 AWS 사용하여 비용을 분류하고 월별 비용 할당 보고서를 제공합니다. 자세한 내용은 AWS Billing 사용 설명서의 비용 할당 태그 사용을 참조하세요.
- 속성 기반 액세스 제어를 사용하여 파이프라인에 대한 액세스를 제한합니다. 예제 정책과 자세한 내용은 IAM 사용 설명서의 태그 키를 기반으로 액세스 제어 섹션을 참조하세요.

에서는 파이프라인이 기본 리소스입니다. OpenSearch Service 콘솔, AWS CLI, OpenSearch Ingestion APIs 또는 AWS SDKs를 사용하여 파이프라인에서 태그를 추가, 관리 및 제거할 수 있습니다.

주제

- 필요한 권한
- 태그 작업(콘솔)
- 태그 작업(AWS CLI)

필요한 권한

OpenSearch Ingestion은 파이프라인에 태그를 지정하는 데 다음 AWS Identity and Access Management Access Analyzer (IAM) 권한을 사용합니다.

- `osis:TagResource`
- `osis:ListTagsForResource`
- `osis:UntagResource`

각 권한에 대한 자세한 내용은 서비스 권한 부여 참조에서 OpenSearch Ingestion에 대한 작업, 리소스 및 조건 키에 대한 액션, 리소스 및 조건 키를 참조하세요.

태그 작업(콘솔)

콘솔은 도메인에 태그를 지정하는 가장 간단한 방법입니다.

태그를 만들려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 파이프라인을 선택합니다.
3. 태그를 추가할 파이프라인을 선택한 다음 태그 탭으로 이동합니다.
4. [관리(Manage)], [새 태그 추가(Add new tag)]를 선택합니다.
5. 태그 키와 선택 값을 입력합니다.
6. 저장을 선택합니다.

태그를 삭제하려면 동일한 단계를 따르고 [태그 관리(Manage tags)] 페이지에서 [제거(Remove)]를 선택합니다.

콘솔을 사용한 태그 작업에 대한 자세한 내용은 AWS 관리 콘솔 시작 안내서에서 Tag Editor를 참조하세요.

태그 작업(AWS CLI)

를 사용하여 파이프라인에 태그를 지정하려면 `TagResource` 요청을 AWS CLI보냅니다.

```
aws osis tag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
```

```
  --tags Key=service,Value=osis Key=source,Value=otel
```

UntagResource 명령을 사용하여 파이프라인에서 태그를 제거합니다.

```
aws osis untag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tag-keys service
```

ListTagsForResource 명령을 사용하여 파이프라인의 기존 태그를 확인합니다.

```
aws osis list-tags-for-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
```

Amazon CloudWatch를 사용한 Amazon OpenSearch Ingestion로깅 및 모니터링

Amazon OpenSearch Ingestion은 지표 및 로그를 Amazon CloudWatch에 게시합니다.

주제

- 파이프라인 모니터링
- 파이프라인 지표 모니터링

파이프라인 모니터링

Amazon OpenSearch Ingestion 파이프라인에 대한 로깅을 활성화하여 파이프라인 작업 및 수집 활동 중에 발생하는 오류 및 경고 메시지를 노출할 수 있습니다. OpenSearch Ingestion은 모든 로그를 Amazon CloudWatch Logs에 게시합니다. CloudWatch Logs는 로그 파일의 정보를 모니터링하고 특정 임곗값에 도달하면 사용자에게 알릴 수 있습니다. 또한 매우 내구력 있는 스토리지에 로그 데이터를 저장할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서를 참조하세요.

OpenSearch Ingestion의 로그에는 요청 처리 실패, 소스에서 싱크로의 인증 오류 및 문제 해결에 도움이 될 수 있는 기타 경고가 표시될 수 있습니다. 해당 로그의 경우 OpenSearch Ingestion은 INFO, WARN, ERROR, FATAL의 로그 수준을 사용합니다. 모든 파이프라인에 대해 로그 게시를 활성화하는 것이 좋습니다.

필요한 권한

OpenSearch Ingestion에서 CloudWatch Logs로 로그를 전송하도록 하려면 특정 IAM 권한을 가진 사용자로 로그인해야 합니다.

로그 전송 리소스를 생성하고 업데이트하려면 다음과 같은 CloudWatch Logs 권한이 필요합니다.

JSON

```
{a
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:GetLogDelivery",
                "logs:ListLogDeliveries"
            ]
        }
    ]
}
```

로그 게시 활성화

기존 파이프라인에서 또는 파이프라인을 생성하는 동안 로그 게시를 활성화할 수 있습니다. 파이프라인 생성 중에 로그 게시를 활성화하는 단계는 the section called “파이프라인 생성”(을)를 참조하세요.

콘솔

기존 파이프라인에서 로그 게시를 활성화하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 파이프라인을 선택합니다.
3. 로그를 활성화하려는 파이프라인을 열고 작업, 로그 게시 옵션 편집을 선택합니다.

4. CloudWatch Logs에 게시를 활성화합니다.
5. 새로운 로그 그룹을 생성하거나 기존 그룹을 선택합니다. 이름을 `/aws/vendedlogs/OpenSearchIngestion/pipeline-name/audit-logs`와 같은 경로 형식으로 지정하는 것이 좋습니다. 이 형식을 사용하면 `/aws/vendedlogs/OpenSearchIngestion`과 같은 특정 경로에 있는 모든 로그 그룹에 권한을 부여하는 CloudWatch 액세스 정책을 더 쉽게 적용할 수 있습니다.

   > **Important**
   >
   > 로그 그룹 이름에 접두사 `vendedlogs`를 포함해야 합니다. 그렇지 않으면 생성이 실패합니다.

6. 저장을 선택합니다.

CLI

를 사용하여 로그 게시를 활성화하려면 다음 요청을 AWS CLI보냅니다.

```
aws osis update-pipeline \
  --pipeline-name my-pipeline \
  --log-publishing-options  IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="/
aws/vendedlogs/OpenSearchIngestion/pipeline-name"}
```

파이프라인 지표 모니터링

원시 데이터를 수집하여 읽기 가능하며 실시간에 가까운 지표로 처리하는 Amazon CloudWatch를 통해 를 Amazon OpenSearch Ingestion 모니터링할 수 있습니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임곗값을 주시하다가 해당 임곗값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.

OpenSearch Ingestion 콘솔에는 CloudWatch의 원시 데이터를 기초로 하는 일련의 차트가 각 파이프라인의 성능 탭에 표시됩니다.

OpenSearch Ingestion은 대부분의 지원되는 플러그인의 지표를 보고합니다. 특정 플러그인에 아래의 자체 표가 없다면 플러그인별 지표를 보고하지 않는다는 뜻입니다. 이러한 지표는 `AWS/OSIS` 네임스페이스에 게시됩니다.

주제

- 공통 지표
- 버퍼 지표
- 서명 V4 지표
- 경계가 있는 차단 버퍼 지표
- Otel 추적 소스 지표
- Otel 지표 소스 지표
- Http 지표
- S3 ta 지표
- 집계 지표
- 날짜 지표
- Lambda 지표
- Grok 지표
- Otel 추적 원시 지표
- Otel 추적 그룹 지표
- 서비스 맵 스테이트풀 메트릭
- OpenSearch 지표
- 시스템 및 측정 지표

공통 지표

다음 지표는 모든 프로세서와 싱크에 공통입니다.

각 지표 앞에는 <*sub_pipeline_name*><*plugin*><*metric_name*> 형식으로 하위 파이프라인 이름과 플러그인 이름이 접두사로 붙습니다. 예를 들어, `my-pipeline`이라는 하위 파이프라인의 `recordsIn.count` 지표 전체 이름과 날짜 프로세서는 `my-pipeline.date.recordsIn.count`과 같습니다.

지표 접미사	설명
`recordsIn.count`	파이프라인 구성 요소로의 레코드 수신. 이 지표는 프로세서와 싱크에 적용됩니다. 관련 통계: 합계

지표 접미사	설명
	차원: PipelineName
recordsOut.count	파이프라인 구성 요소로의 레코드 송신. 이 지표는 프로세서와 소스에 적용됩니다. 관련 통계: 합계 차원: PipelineName
timeElapsed.count	파이프라인 구성 요소 실행 중에 기록된 데이터 포인트 수입니다. 이 지표는 프로세서와 싱크에 적용됩니다. 관련 통계: 합계 차원: PipelineName
timeElapsed.sum	파이프라인 구성 요소를 실행하는 동안 경과된 총 시간입니다. 이 지표는 프로세서와 싱크에 적용(밀리초)됩니다. 관련 통계: 합계 차원: PipelineName
timeElapsed.max	파이프라인 구성 요소를 실행하는 동안 경과된 최대 시간입니다. 이 지표는 프로세서와 싱크에 적용(밀리초)됩니다. 관련 통계: 최대 차원: PipelineName

버퍼 지표

다음 지표는 OpenSearch Ingestion이 모든 파이프라인에 대해 자동으로 구성하는 기본 경계가 있는 차단 버퍼에 적용됩니다.

각 지표 앞에는 <*sub_pipeline_name*><*plugin*><*metric_name*> 형식으로 하위 파이프라인 이름과 버퍼 이름이 접두사로 붙습니다. 예를 들어, my-pipeline이라는 하위 파이프라인의 recordsWritten.count 지표 전체 이름이 my-pipeline.BlockingBuffer.recordsWritten.count(와)과 같습니다.

지표 접미사	설명
recordsWritten.count	버퍼에 기록된 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
recordsRead.count	버퍼에서 읽은 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
recordsInFlight.value	버퍼에서 읽은 미확인된 레코드 수입니다. 관련 통계: Average 차원: PipelineName
recordsInBuffer.value	현재 버퍼에 있는 레코드 수입니다. 관련 통계: Average 차원: PipelineName
recordsProcessed.count	버퍼에서 읽고 파이프라인에서 처리한 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
recordsWriteFailed.count	파이프라인이 싱크에 기록하지 못한 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
writeTimeElapsed.count	버퍼에 쓰는 동안 기록된 데이터 포인트 수입니다. 관련 통계: 합계

지표 접미사	설명
	차원: PipelineName
writeTimeElapsed.sum	버퍼에 쓰는 동안 경과된 총 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
writeTimeElapsed.max	버퍼에 쓰는 동안 경과된 최대 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName
writeTimeouts.count	버퍼에 대한 쓰기 타임아웃 횟수입니다. 관련 통계: 합계 차원: PipelineName
readTimeElapsed.count	버퍼에 쓰는 동안 기록된 데이터 포인트 수입니다. 관련 통계: 합계 차원: PipelineName
readTimeElapsed.sum	버퍼에서 읽는 동안 경과된 총 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
readTimeElapsed.max	버퍼에서 읽는 동안 경과된 최대 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName

지표 접미사	설명
checkpointTimeElapsed.count	체크포인트를 수행하는 동안 기록된 데이터 포인트 수입니다. 관련 통계: 합계 차원: PipelineName
checkpointTimeElapsed.sum	체크포인트를 수행하는 동안 경과된 총 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
checkpointTimeElapsed.max	체크포인트를 수행하는 동안 경과된 최대 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName

서명 V4 지표

다음 지표는 파이프라인의 수집 엔드포인트에 적용되며 소스 플러그인(http, otel_trace, otel_metrics)과 연결됩니다. OpenSearch Ingestion에 대한 모든 요청은 서명 버전 4로 서명되어야 합니다. 이러한 지표를 통해 파이프라인에 연결할 때 권한 부여 문제를 식별하거나 성공적으로 인증되고 있는지 확인할 수 있습니다.

각 지표 앞에는 하위 파이프라인 이름 및 osis_sigv4_auth(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.osis_sigv4_auth.httpAuthSuccess.count입니다.

지표 접미사	설명
httpAuthSuccess.count	파이프라인에 대한 성공적인 서명 V4 요청 수입니다. 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
httpAuthFailure.count	파이프라인에 대한 실패한 서명 V4 요청 수입니다. 관련 통계: 합계 차원: PipelineName
httpAuthServerError.count	서버 오류를 반환한 파이프라인에 대한 서명 V4 요청 수입니다. 관련 통계: 합계 차원: PipelineName

경계가 있는 차단 버퍼 지표

다음 지표는 경계가 있는 차단 버퍼에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 BlockingBuffer(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.BlockingBuffer.bufferUsage.value입니다.

지표 접미사	설명
bufferUsage.value	버퍼에 있는 레코드 수를 기준으로 한 buffer_size 의 사용률입니다. buffer_size 는 버퍼에 기록된 최대 레코드 수와 확인되지 않은 기내 레코드를 나타냅니다. 관련 통계: Average 차원: PipelineName

Otel 추적 소스 지표

다음 지표는 oTel 추적 소스에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 otel_trace_source(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.otel_trace_source.requestTimeouts.count입니다.

지표 접미사	설명
requestTimeouts.count	시간을 초과한 요청 수입니다. 관련 통계: 합계 차원: PipelineName
requestsReceived.count	플러그인에서 수신된 요청 수입니다. 관련 통계: 합계 차원: PipelineName
successRequests.count	메시지 브로커가 성공적으로 처리한 구독 요청 수. 관련 통계: 합계 차원: PipelineName
badRequests.count	플러그인에서 처리한 잘못된 형식의 요청 수입니다. 관련 통계: 합계 차원: PipelineName
requestsTooLarge.count	콘텐츠의 스팬 수가 버퍼 용량보다 큰 요청의 수입니다. 관련 통계: 합계 차원: PipelineName
internalServerError.count	사용자 지정 예외 유형을 사용하여 플러그인에서 처리한 요청 수입니다. 관련 통계: 합계 차원: PipelineName
requestProcessDuration.count	플러그인의 요청을 처리하는 동안 기록된 데이터 포인트 수입니다.

지표 접미사	설명
	관련 통계: 합계 차원: PipelineName
requestProcessDuration.sum	플러그인에서 처리한 요청의 총 지연 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
requestProcessDuration.max	플러그인에서 처리한 요청의 최대 지연 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName
payloadSize.count	수신 요청의 페이로드 크기 분포 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
payloadSize.sum	수신 요청의 페이로드 크기 분포의 총 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
payloadSize.max	수신 요청의 페이로드 크기 분포의 최대 수(바이트)입니다. 관련 통계: 최대 차원: PipelineName

Otel 지표 소스 지표

다음 지표는 oTel 지표 소스에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 otel_metrics_source(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.otel_metrics_source.requestTimeouts.count입니다.

지표 접미사	설명
requestTimeouts.count	시간 초과된 플러그인에 대한 총 요청 수입니다. 관련 통계: 합계 차원: PipelineName
requestsReceived.count	플러그인에서 수신된 요청 총 수입니다. 관련 통계: 합계 차원: PipelineName
successRequests.count	플러그인이 성공적으로 처리한 요청 수(응답 상태 코드 200개)입니다. 관련 통계: 합계 차원: PipelineName
requestProcessDuration.count	플러그인에서 처리한 요청의 지연 시간 수(초)입니다. 관련 통계: 합계 차원: PipelineName
requestProcessDuration.sum	플러그인에서 처리한 요청의 총 지연 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
requestProcessDuration.max	플러그인에서 처리한 요청의 최대 지연 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName
payloadSize.count	수신 요청의 페이로드 크기 분포 수(바이트)입니다.

지표 접미사	설명
	관련 통계: 합계 차원: PipelineName
payloadSize.sum	수신 요청의 페이로드 크기 분포의 총 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
payloadSize.max	수신 요청의 페이로드 크기 분포의 최대 수(바이트)입니다. 관련 통계: 최대 차원: PipelineName

Http 지표

다음 지표는 HTTP 소스에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 http(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.http.requestsReceived.count입니다.

지표 접미사	설명
requestsReceived.count	/log/ingest 엔드포인트에서 수신된 바이트 수입니다. 관련 통계: 합계 차원: PipelineName
requestsRejected.count	플러그인이 거부한 요청 수(응답 상태 코드 429개)입니다. 관련 통계: 합계 차원: PipelineName
successRequests.count	플러그인이 성공적으로 처리한 요청 수(응답 상태 코드 200개)입니다. 관련 통계: 합계

지표 접미사	설명
	차원: PipelineName
badRequests.count	플러그인에서 처리한 콘텐츠 유형이나 형식이 잘못된 요청 수(응답 상태 코드 400개)입니다. 관련 통계: 합계 차원: PipelineName
requestTimeouts.count	HTTP 소스 서버에서 제한 시간이 초과된 요청 수(415 응답 상태 코드)입니다. 관련 통계: 합계 차원: PipelineName
requestsTooLarge.count	콘텐츠의 이벤트 수가 버퍼 용량보다 큰 요청의 수(413 응답 상태 코드)입니다. 관련 통계: 합계 차원: PipelineName
internalServerError.count	사용자 지정 예외 유형을 사용하여 플러그인에서 처리한 요청 수(500 응답 상태 코드)입니다. 관련 통계: 합계 차원: PipelineName
requestProcessDuration.count	플러그인에서 처리한 요청의 지연 시간 수(초)입니다. 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
`requestProcessDuration.sum`	플러그인에서 처리한 요청의 총 지연 시간(밀리초)입니다. 관련 통계: 합계 차원: `PipelineName`
`requestProcessDuration.max`	플러그인에서 처리한 요청의 최대 지연 시간(밀리초)입니다. 관련 통계: 최대 차원: `PipelineName`
`payloadSize.count`	수신 요청의 페이로드 크기 분포 수(바이트)입니다. 관련 통계: 합계 차원: `PipelineName`
`payloadSize.sum`	수신 요청의 페이로드 크기 분포의 총 수(바이트)입니다. 관련 통계: 합계 차원: `PipelineName`
`payloadSize.max`	수신 요청의 페이로드 크기 분포의 최대 수(바이트)입니다. 관련 통계: 최대 차원: `PipelineName`

S3 ta 지표

다음 지표는 S3 소스에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 `s3`(이)가 붙습니다. 예를 들어 *`sub_pipeline_name`*`.s3.s3ObjectsFailed.count`입니다.

지표 접미사	설명
`s3ObjectsFailed.count`	플러그인이 읽지 못한 S3 객체의 총 수입니다.

지표 접미사	설명
	관련 통계: 합계 차원: PipelineName
s3ObjectsNotFound.count	S3에서 Not Found 오류가 발생하여 플러그인이 읽지 못한 S3 객체의 수입니다. 이러한 지표도 s3ObjectsFailed 지표에 포함됩니다. 관련 통계: 합계 차원: PipelineName
s3ObjectsAccessDenied.count	S3에서 Access Denied 또는 Forbidden 오류가 발생하여 플러그인이 읽지 못한 S3 객체의 수입니다. 이러한 지표도 s3ObjectsFailed 지표에 포함됩니다. 관련 통계: 합계 차원: PipelineName
s3ObjectReadTimeElapsed.count	플러그인이 S3 객체에 대한 GET 요청을 수행하고, 객체를 구문 분석하고, 버퍼에 이벤트를 쓰는 데 걸리는 시간입니다. 관련 통계: 합계 차원: PipelineName
s3ObjectReadTimeElapsed.sum	플러그인이 S3 객체에 대한 GET 요청을 수행하고, 객체를 구문 분석하고, 버퍼에 이벤트를 쓰는 데 걸리는 시간입니다(밀리초). 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
s3ObjectReadTimeElapsed.max	플러그인이 S3 객체에 대한 GET 요청을 수행하고, 객체를 구문 분석하고, 버퍼에 이벤트를 쓰는 데 걸리는 최대 시간입니다(밀리초). 관련 통계: 최대 차원: PipelineName
s3ObjectSizeBytes.count	S3 객체 크기의 분포 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
s3ObjectSizeBytes.sum	S3 객체 크기의 분포 총 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
s3ObjectSizeBytes.max	S3 객체 크기의 최대 분포 수(바이트)입니다. 관련 통계: 최대 차원: PipelineName
s3ObjectProcessedBytes.count	플러그인에서 처리한 S3 객체의 배포 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
s3ObjectProcessedBytes.sum	플러그인에서 처리한 S3 객체의 총 배포 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
s3ObjectProcessedBytes.max	플러그인에서 처리한 S3 객체의 최대 배포 수(바이트)입니다. 관련 통계: 최대 차원: PipelineName
s3ObjectsEvents.count	플러그인이 수신한 S3 이벤트의 배포 수입니다. 관련 통계: 합계 차원: PipelineName
s3ObjectsEvents.sum	플러그인이 수신한 S3 이벤트의 총 배포입니다. 관련 통계: 합계 차원: PipelineName
s3ObjectsEvents.max	플러그인이 수신한 S3 이벤트의 최대 배포입니다. 관련 통계: 최대 차원: PipelineName
sqsMessageDelay.count	S3가 객체 생성에 필요한 이벤트 시간을 기록하여 객체가 완전히 구문 분석된 시점까지 기록하는 동안 기록된 데이터 포인트 수입니다. 관련 통계: 합계 차원: PipelineName
sqsMessageDelay.sum	S3가 객체 생성을 위한 이벤트 시간을 기록하는 시점부터 완전히 구문 분석된 시점까지의 총 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
sqsMessageDelay.max	S3가 객체 생성을 위한 이벤트 시간을 기록하는 시점부터 완전히 구문 분석된 시점까지의 최대 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName
s3ObjectsSucceeded.count	플러그인이 성공적으로 읽은 S3 객체 수입니다. 관련 통계: 합계 차원: PipelineName
sqsMessagesReceived.count	플러그인이 대기열에서 수신한 Amazon SQS 메시지 수입니다. 관련 통계: 합계 차원: PipelineName
sqsMessagesDeleted.count	플러그인이 대기열에서 삭제한 Amazon SQS 메시지 수입니다. 관련 통계: 합계 차원: PipelineName
sqsMessagesFailed.count	플러그인이 구문 분석하지 못한 Amazon SQS 메시지 수입니다. 관련 통계: 합계 차원: PipelineName

집계 지표

다음 지표는 집계 프로세서에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 aggregate(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.aggregate.actionHandleEventsOut.count입니다.

지표 접미사	설명
actionHandleEvents Out.count	구성된 작업에 대한 handleEvent 호출에서 반환된 이벤트 수입니다. 관련 통계: 합계 차원: PipelineName
actionHandleEvents Dropped.count	구성된 작업에 대한 handleEvent 호출에서 반환된 이벤트 수입니다. 관련 통계: 합계 차원: PipelineName
actionHandleEvents ProcessingErrors.count	오류가 발생한 구성된 작업에 대해 handleEvent 로 걸려온 호출 수입니다. 관련 통계: 합계 차원: PipelineName
actionConcludeGrou pEventsOut.count	구성된 작업에 대한 concludeGroup 호출에서 반환된 이벤트 수입니다. 관련 통계: 합계 차원: PipelineName
actionConcludeGrou pEventsDropped.count	구성된 작업에 대한 condludeGroup 호출에서 반환되지 않은 이벤트 수입니다. 관련 통계: 합계 차원: PipelineName
actionConcludeGrou pEventsProcessingE rrors.count	오류가 발생한 구성된 작업에 대해 concludeGroup 로 걸려온 호출 수입니다. 관련 통계: 합계

지표 접미사	설명
	차원: PipelineName
currentAggregateGroups.value	현재 그룹 수입니다. 이 게이지는 그룹이 종료되면 감소하고 이벤트에서 새 그룹이 생성되기 시작하면 증가합니다. 관련 통계: Average 차원: PipelineName

날짜 지표

다음 지표는 날짜 프로세서에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 date(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.date.dateProcessingMatchSuccess.count입니다.

지표 접미사	설명
dateProcessingMatchSuccess.count	match 구성 옵션에 지정된 패턴 중 최소 하나 이상과 일치하는 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
dateProcessingMatchFailure.count	match 구성 옵션에 지정된 패턴 중 어떤 것과도 일치하지 않는 레코드 수입니다. 관련 통계: 합계 차원: PipelineName

Lambda 지표

다음 지표가 AWS Lambda 프로세서에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 lambda(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.lambda.recordsSuccessfullySentToLambda.count입니다.

지표 접미사	설명
recordsSuccessfullySentToLambda.count	Lambda 함수에서 성공적으로 처리된 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
recordsFailedToSendToLambda.count	Lambda 함수로 전송하지 못한 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
lambdaFunctionLatency.avg lambdaFunctionLatency.max	Lambda 함수 호출의 지연 시간입니다. 관련 통계: 평균 및 최대 차원: PipelineName
numberOfRequestsSucceeded.count	성공한 Lambda 호출 요청의 총 수입니다. 관련 통계: 합계 차원: PipelineName
numberOfRequestsFailed.count	실패한 Lambda 호출 요청의 총 수입니다. 관련 통계: 합계 차원: PipelineName
requestPayloadSize.avg	Lambda로 전송된 요청 페이로드의 크기입니다. 관련 통계: Average 차원: PipelineName
responsePayloadSize.avg	Lambda에서 수신한 응답 페이로드의 크기입니다. 관련 통계: Average

지표 접미사	설명
	차원: PipelineName

Grok 지표

다음 지표는 Grok 프로세서에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 grok(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.grok.grokProcessingMatch.count입니다.

지표 접미사	설명
grokProcessingMatch.count	match 구성 옵션에 최소 하나 이상의 패턴이 검색된 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
grokProcessingMismatch.count	match 구성 옵션에 지정된 패턴 중 어떤 것과도 일치하지 않는 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
grokProcessingErrors.count	레코드 처리 오류 수입니다. 관련 통계: 합계 차원: PipelineName
grokProcessingTimeouts.count	매칭 중에 제한 시간이 초과된 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
grokProcessingTime.count	개별 레코드가 match 구성 옵션의 패턴과 일치하는 동안 기록된 데이터 포인트 수입니다. 관련 통계: 합계

지표 접미사	설명
	차원: PipelineName
grokProcessingTime.sum	각 개별 레코드가 match 구성 옵션의 패턴과 일치시키는 데 걸리는 총 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
grokProcessingTime.max	각 개별 레코드가 match 구성 옵션의 패턴과 일치시키는 데 걸리는 최대 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName

Otel 추적 원시 지표

다음 지표는 OTel 추적 원시 프로세서에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 otel_trace_raw(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.otel_trace_raw.traceGroupCacheCount.value입니다.

지표 접미사	설명
traceGroupCacheCount.value	추적 그룹 캐시의 추적 그룹 수입니다. 관련 통계: 합계 차원: PipelineName
spanSetCount.value	스팬 세트 컬렉션의 스팬 세트 수입니다. 관련 통계: 합계 차원: PipelineName

Otel 추적 그룹 지표

다음 지표는 OTel 추적 그룹 프로세서에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 `otel_trace_group`(이)가 붙습니다. 예를 들어 *`sub_pipeline_name`*`.otel_trace_group.recordsInMissingTraceGroup.count`입니다.

지표 접미사	설명
`recordsInMissingTraceGroup.count`	추적 그룹 필드가 누락된 수신 레코드 수입니다. 관련 통계: 합계 차원: `PipelineName`
`recordsOutFixedTraceGroup.count`	추적 그룹 필드가 성공적으로 채워진 수신 레코드의 수입니다. 관련 통계: 합계 차원: `PipelineName`
`recordsOutMissingTraceGroup.count`	추적 그룹 필드가 누락된 송신 레코드 수입니다. 관련 통계: 합계 차원: `PipelineName`

서비스 맵 스테이트풀 메트릭

다음 지표는 Service-Map 상태 저장 프로세서에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 `service-map-stateful`(이)가 붙습니다. 예를 들어 *`sub_pipeline_name`*`.service-map-stateful.spansDbSize.count`입니다.

지표 접미사	설명
`spansDbSize.value`	현재 및 이전 지속 기간에 대한 MapDB 내 스팬의 메모리 바이트 크기입니다. 관련 통계: Average

지표 접미사	설명
	차원: PipelineName
traceGroupDbSize.value	현재 및 이전 지속 기간에 대한 MapDB 내 추적 그룹의 메모리 바이트 크기입니다. 관련 통계: Average 차원: PipelineName
spansDbCount.value	현재 및 이전 지속 기간에 대한 MapDB 내 스팬의 수입니다. 관련 통계: 합계 차원: PipelineName
traceGroupDbCount.value	현재 및 이전 지속 기간에 대한 MapDB의 추적 그룹 수입니다. 관련 통계: 합계 차원: PipelineName
relationshipCount.value	현재 및 이전 지속 기간 동안 저장된 관계 수입니다. 관련 통계: 합계 차원: PipelineName

OpenSearch 지표

다음 지표는 OpenSearch 싱크에 적용됩니다. 각 지표 앞에는 하위 파이프라인 이름 및 opensearch(이)가 붙습니다. 예를 들어 *sub_pipeline_name*.opensearch.bulkRequestErrors.count입니다.

지표 접미사	설명
bulkRequestErrors.count	대량 요청을 보내는 동안 발생한 총 오류 수입니다. 관련 통계: 합계

지표 접미사	설명
	차원: PipelineName
documentsSuccess.count	대량 요청을 통해 OpenSearch Service에 성공적으로 전송된 문서 수(재시도 포함)입니다. 관련 통계: 합계 차원: PipelineName
documentsSuccessFirstAttempt.count	첫 시도에 대량 요청을 통해 OpenSearch Service에 성공적으로 전송된 문서 수입니다. 관련 통계: 합계 차원: PipelineName
documentErrors.count	대량 요청으로 전송하지 못한 문서 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestFailed.count	실패한 대량 요청 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestNumberOfRetries.count	대량 복원 요청의 수. 관련 통계: 합계 차원: PipelineName
bulkBadRequestErrors.count	대량 요청을 보내는 동안 발생한 Bad Request 오류 수입니다. 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
bulkRequestNotAllowedErrors.count	대량 요청을 보내는 동안 발생한 Request Not Allowed 오류 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestInvalidInputErrors.count	대량 요청을 보내는 동안 발생한 Invalid Input 오류 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestNotFoundErrors.count	대량 요청을 보내는 동안 발생한 Request Not Found 오류 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestTimeoutErrors.count	대량 요청을 보내는 동안 발생한 Request Timeout 오류 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestServerErrors.count	대량 요청을 보내는 동안 발생한 Server Error 오류 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestSizeBytes.count	대량 요청의 페이로드 크기 분포 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
bulkRequestSizeBytes.sum	대량 요청의 페이로드 크기 분포 총 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestSizeBytes.max	대량 요청의 페이로드 크기 분포 최대 수(바이트)입니다. 관련 통계: 최대 차원: PipelineName
bulkRequestLatency.count	재시도를 포함하여 요청이 플러그인으로 전송되는 동안 기록된 데이터 포인트 수입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestLatency.sum	재시도를 포함하여 플러그인으로 전송된 요청의 총 지연 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
bulkRequestLatency.max	재시도를 포함하여 플러그인으로 전송된 요청의 최대 지연 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName
s3.dlqS3RecordsSuccess.count	S3 DLQ(Dead Letter Queue)로 성공적으로 전송된 레코드 수입니다. 관련 통계: 합계 차원: PipelineName

지표 접미사	설명
s3.dlqS3RecordsFailed.count	S3 DLQ(Dead Letter Queue)로 전송되지 못한 레코드 수입니다. 관련 통계: 합계 차원: PipelineName
s3.dlqS3RequestSuccess.count	S3 DLQ(Dead Letter Queue)에 성공한 요청 수입니다. 관련 통계: 합계 차원: PipelineName
s3.dlqS3RequestFailed.count	S3 DLQ(Dead Letter Queue)에 실패한 요청 수입니다. 관련 통계: 합계 차원: PipelineName
s3.dlqS3RequestLatency.count	재시도를 포함하여 요청이 S3 DLQ(Dead Letter Queue)로 전송되는 동안 기록된 데이터 포인트 수입니다. 관련 통계: 합계 차원: PipelineName
s3.dlqS3RequestLatency.sum	재시도를 포함하여 S3 DLQ(Dead Letter Queue)로 전송된 요청의 총 지연 시간(밀리초)입니다. 관련 통계: 합계 차원: PipelineName
s3.dlqS3RequestLatency.max	재시도를 포함하여 S3 DLQ(Dead Letter Queue)로 전송된 요청의 최대 지연 시간(밀리초)입니다. 관련 통계: 최대 차원: PipelineName

지표 접미사	설명
s3.dlqS3RequestSizeBytes.count	S3 DLQ(Dead Letter Queue)에 대한 요청의 페이로드 크기 분포 수(바이트)입니다. 관련 통계: 합계 차원: PipelineName
s3.dlqS3RequestSizeBytes.sum	S3 DLQ(Dead Letter Queue)에 대한 요청의 총 페이로드 크기의 총 분포(바이트)입니다. 관련 통계: 합계 차원: PipelineName
s3.dlqS3RequestSizeBytes.max	S3 DLQ(Dead Letter Queue)에 대한 요청의 최대 페이로드 크기 분포(바이트)입니다. 관련 통계: 최대 차원: PipelineName

시스템 및 측정 지표

다음 지표는 전체 OpenSearch Ingestion 시스템에 적용됩니다. 이러한 지표 앞에는 아무 것도 붙지 않습니다.

지표	설명
system.cpu.usage.value	모든 데이터 노드의 사용 가능한 CPU 사용률입니다. 관련 통계: Average 차원: PipelineName , area, id
system.cpu.count.value	모든 데이터 노드의 총 CPU 사용량입니다. 관련 통계: Average

지표	설명
	차원: PipelineName , area, id
jvm.memory.max.value	메모리 관리에 사용할 수 있는 최대 메모리 용량(바이트)입니다. 관련 통계: Average 차원: PipelineName , area, id
jvm.memory.used.value	사용된 총 메모리 용량(바이트). 관련 통계: Average 차원: PipelineName , area, idsigna
jvm.memory.committed.value	Java 가상 머신(JVM)에서 사용하기 위해 커밋된 메모리의 용량(바이트)입니다. 관련 통계: Average 차원: PipelineName , area, id
computeUnits	파이프라인에서 사용 중인 Ingestion OpenSearch Compute Units (Ingestion OCUs)의 수입니다. 관련 통계: Maximum, Sum, Average 차원: PipelineName

Amazon OpenSearch Ingestion의 모범 사례

이 주제는 Amazon OpenSearch Ingestion 파이프라인 생성 및 관리에 대한 모범 사례를 제공하며, 많은 사용 사례에 적용되는 일반 지침을 포함하고 있습니다. 각 워크로드는 고유한 특성을 가지고 있으므로 모든 사용 사례에 적합한 일반적인 권장 사항은 없습니다.

주제

- 일반 모범 사례
- 권장되는 CloudWatch 경보

일반 모범 사례

파이프라인 생성 및 관리에는 다음과 같은 일반적인 모범 사례가 적용됩니다.

- 고가용성을 보장하려면 2개 또는 3개의 서브넷으로 VPC 파이프라인을 구성합니다. 하나의 서브넷에만 파이프라인을 배포하고 가용 영역이 다운되면 데이터를 수집할 수 없습니다.
- 각 파이프라인 내에서 하위 파이프라인 수를 5개 이하로 제한하는 것이 좋습니다.
- S3 소스 플러그인을 사용하는 경우 최적의 성능을 위해 균일한 크기의 S3 파일을 사용하세요.
- S3 소스 플러그인을 사용하는 경우 최적의 성능을 위해 S3 버킷에서 파일 크기 0.25GB마다 가시성 제한 시간을 30초씩 추가하세요.
- 실패한 이벤트를 오프로드하고 분석에 액세스할 수 있도록 파이프라인 구성에 DLQ(Dead Letter Queue)를 포함시키세요. 잘못된 매핑이나 기타 문제로 인해 싱크에서 데이터가 거부되는 경우 문제를 해결하고 수정하기 위해 데이터를 DLQ로 라우팅할 수 있습니다.

권장되는 CloudWatch 경보

CloudWatch 경보는 CloudWatch 지표가 일정 시간 동안 지정된 값을 초과하면 조치를 수행합니다. 예를 들어 클러스터 상태가 AWS 1분 이상 `red` 지속되면 이메일을 보낼 수 있습니다. 이 단원에는 Amazon OpenSearch Ingestion에 권장되는 몇 가지 경보와 이에 대응하는 방법이 포함되어 있습니다.

경보 구성에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 경보 생성을 참조하세요.

경보	문제
`computeUnits` 최대값은 15분, 연속 횟수 3번 동안 구성된 `maxUnits` 값임	파이프라인이 최대 용량에 도달했으며 `maxUnits` 업데이트가 필요할 수 있습니다. 파이프라인의 최대 용량을 늘리세요.
`opensearch.documentErrors.count` 합계는 1분, 연속 횟수 1번 동안 = *{sub_pipe*	파이프라인이 OpenSearch 싱크에 쓸 수 없습니다. 파이프라인 권한을 확인하고 도메인이나 컬렉션이 정상인지 확인하세요. DLQ(Dead Letter Queue)에 실패한 이벤트가 있는지 확인할 수도 있습니다 (구성된 경우).

경보	문제
line_name } .opensearch.recordsIn.count 임	
bulkRequestLatency.max 최대값은 1분, 연속 횟수 1번 동안 >= x임	파이프라인이 OpenSearch 싱크로 데이터를 전송하는 데 지연이 많이 발생합니다. 이는 싱크 크기가 너무 작거나 샤딩 전략이 잘못되어 싱크가 뒤처지고 있기 때문일 수 있습니다. 지연 시간이 오래 지속되면 파이프라인 성능에 영향을 줄 수 있으며 클라이언트의 역압으로 이어질 수 있습니다.
httpAuthFailure.count 합계는 1분, 연속 횟수 1번 동안 >= 1임	수집 요청은 인증되지 않습니다. 모든 클라이언트에 서명 버전 4 인증이 올바르게 활성화되어 있는지 확인하세요.
system.cpu.usage.value 평균값은 15분, 연속 횟수 3번 동안 >= 80%임	지속적으로 CPU 사용률이 높아지면 문제가 될 수 있습니다. 파이프라인의 최대 용량을 늘리는 것이 좋습니다.
bufferUsage.value 평균값은 15분, 연속 횟수 3번 동안 >= 80%임	지속적으로 높은 버퍼 사용량은 문제가 될 수 있습니다. 파이프라인의 최대 용량을 늘리는 것이 좋습니다.

고려할 만한 기타 경보

정기적으로 사용하는 Amazon OpenSearch Ingestion 기능에 따라 다음 경보 구성을 고려하세요.

경보	문제
dynamodb.exportJobFailure.count 합계가 1	Amazon S3로 내보내기를 트리거하는 시도가 실패했습니다.

경보	문제
opensearch.EndtoEndLatency.avg 평균값은 15분, 연속 횟수 4번 동안 > X임	EndtoEndLatency 는 DynamoDB 스트림에서 읽을 때 필요한 값보다 높습니다. 이는 OpenSearch 클러스터의 규모가 작거나 최대 파이프라인 OCU 용량이 DynamoDB 테이블의 WCU 처리량에 비해 너무 낮기 때문일 수 있습니다. EndtoEndLatency 는 내보내기 후에는 더 높지만 시간이 지나면 최근 DynamoDB 스트림을 따라잡기 때문에 감소할 것입니다.
dynamodb.changeEventsProcessed.count 합계는 X분 동안 == 0임	DynamoDB 스트림에서 수집되는 레코드가 없습니다. 테이블에 활동이 없거나 DynamoDB 스트림 액세스에 문제가 있을 수 있습니다.
opensearch.s3.dlqS3RecordsSuccess.count 합계는 1분, 연속 횟수 1번 동안 >= opensearch.documentSuccess.count 합계임	OpenSearch 싱크보다 많은 수의 레코드가 DLQ로 전송되고 있습니다. OpenSearch 싱크 플러그인 지표를 검토하여 근본 원인을 조사하고 결정하세요.
grok.grokProcessingTimeouts.count 합계는 = 1분, 연속 횟수 5번 동안 recordsIn.count 합계임	Grok 프로세서가 패턴 매칭을 시도하는 동안 모든 데이터의 타임아웃이 발생했습니다. 이로 인해 성능이 저하되고 파이프라인 속도가 느려질 수 있습니다. 패턴을 조정하여 타임아웃을 줄이는 것을 고려해 보세요.
grok.grokProcessingErrors.count 합계는 1분, 연속 횟수 1번 동안 >= 1임	Grok 프로세서가 파이프라인의 데이터와 패턴을 일치시키지 못해 오류가 발생했습니다. 데이터와 Grok 플러그인 구성을 검토하여 패턴 매칭이 예상되는지 확인하세요.

경보	문제
grok.grokProcessingMismatch.count 합계는 = 1분, 연속 횟수 5번 동안 recordsIn.count 합계임	Grok 프로세서가 파이프라인의 데이터와 패턴을 일치시키지 못했습니다. 데이터와 Grok 플러그인 구성을 검토하여 패턴 매칭이 예상되는지 확인하세요.
date.dateProcessingMatchFailure.count 합계는 = 1분, 연속 횟수 5번 동안 recordsIn.count 합계임	날짜 프로세서가 파이프라인의 데이터에 어떤 패턴도 일치시킬 수 없습니다. 데이터와 날짜 플러그인 구성을 검토하여 패턴 매칭이 예상되는지 확인하세요.
s3.s3ObjectsFailed.count 합계는 1분, 연속 횟수 1번 동안 >= 1임	이 문제는 S3 객체가 없거나 파이프라인에 충분한 권한이 없기 때문에 발생합니다. s3ObjectsNotFound.count 및 s3ObjectsAccessDenied.count 지표를 검토하여 근본 원인을 파악하세요. S3 객체가 존재하는지 확인하거나 권한을 업데이트하세요.
s3.sqsMessagesFailed.count 합계는 1분, 연속 횟수 1번 동안 >= 1임	S3 플러그인이 Amazon SQS 메시지를 처리하지 못했습니다. SQS 대기열에서 DLQ를 활성화한 경우 실패한 메시지를 검토하세요. 파이프라인이 처리하려는 잘못된 데이터를 대기열에 수신하고 있을 수 있습니다.
http.badRequests.count 합계는 1분, 연속 횟수 3번 동안 >= 1임	클라이언트가 잘못된 요청을 보내고 있습니다. 모든 클라이언트가 적절한 페이로드를 보내고 있는지 확인하세요.

경보	문제
`http.requestsTooLarge.count` 합계는 1분, 연속 횟수 1번 동안 >= 1임	HTTP 소스 플러그인의 요청에 너무 많은 데이터가 포함되어 있어 버퍼 용량을 초과합니다. 클라이언트의 배치 크기를 조정하세요.
`http.internalServerError.count` 합계는 1분, 연속 횟수 1번 동안 >= 0임	HTTP 소스 플러그인이 이벤트를 수신하는 데 문제가 있습니다.
`http.requestTimeouts.count` 합계는 1분, 연속 횟수 1번 동안 >= 0임	소스 타임아웃은 파이프라인이 제대로 프로비저닝되지 않았기 때문일 수 있습니다. 추가 워크로드를 처리하기 위해 파이프라인 `maxUnits`(을)를 늘리는 것을 고려해 보세요.
`otel_trace.badRequests.count` 합계는 1분, 연속 횟수 1번 동안 >= 1임	클라이언트가 잘못된 요청을 보내고 있습니다. 모든 클라이언트가 적절한 페이로드를 보내고 있는지 확인하세요.
`otel_trace.requestsTooLarge.count` 합계는 1분, 연속 횟수 1번 동안 >= 1임	Otel Trace 소스 플러그인의 요청에 너무 많은 데이터가 포함되어 있어 버퍼 용량을 초과합니다. 클라이언트의 배치 크기를 조정하세요.

경보	문제
otel_trace.internalServerError.count 합계는 1분, 연속 횟수 1번 동안 >= 0임	Otel Trace 소스 플러그인이 이벤트를 수신하는 데 문제가 있습니다.
otel_trace.requestTimeouts.count 합계는 1분, 연속 횟수 1번 동안 >= 0임	소스 타임아웃은 파이프라인이 제대로 프로비저닝되지 않았기 때문일 수 있습니다. 추가 워크로드를 처리하기 위해 파이프라인 maxUnits(을)를 늘리는 것을 고려해 보세요.
otel_metrics.requestTimeouts.count 합계는 1분, 연속 횟수 1번 동안 >= 0임	소스 타임아웃은 파이프라인이 제대로 프로비저닝되지 않았기 때문일 수 있습니다. 추가 워크로드를 처리하기 위해 파이프라인 maxUnits(을)를 늘리는 것을 고려해 보세요.

Amazon OpenSearch Serverless

Amazon OpenSearch Serverless는 Amazon OpenSearch Service를 위한 온디맨드 auto-scaling 구성입니다. 수동 용량 관리가 필요한 프로비저닝된 OpenSearch 도메인과 달리 OpenSearch Serverless 컬렉션은 애플리케이션의 필요에 따라 컴퓨팅 리소스를 자동으로 확장합니다.

OpenSearch Serverless는 빈번하지 않거나 간헐적이거나 예측할 수 없는 워크로드에 비용 효율적인 솔루션을 제공합니다. 애플리케이션의 사용량에 따라 컴퓨팅 용량을 자동으로 조정하여 비용을 최적화합니다. 서버리스 컬렉션은 프로비저닝된 OpenSearch Service 도메인과 동일한 대용량 분산 고가용성 스토리지 볼륨을 사용합니다.

OpenSearch Serverless 컬렉션은 항상 암호화됩니다. 암호화 키를 선택할 수 있지만 암호화를 비활성화할 수는 없습니다. 자세한 내용은 the section called “암호화” 섹션을 참조하세요.

이점

OpenSearch Serverless에는 다음과 같은 이점이 있습니다.

- 프로비저닝보다 간단함 – OpenSearch Serverless는 OpenSearch 클러스터와 용량 관리의 복잡성을 크게 줄입니다. 클러스터의 크기와 설정을 자동으로 조정하고 샤드 및 인덱스 수명 주기 관리를 처리합니다. 또한 서비스 소프트웨어 업데이트와 OpenSearch 버전 업그레이드도 관리합니다. 모든 업데이트와 업그레이드는 중단되지 않습니다.
- 비용 효율적 – OpenSearch Serverless를 사용하면 사용한 리소스에 대해서만 비용을 지불하면 됩니다. 따라서 피크 워크로드에 대한 사전 프로비저닝과 오버프로비저닝이 필요하지 않습니다.
- 고가용성 – OpenSearch Serverless는 중복성을 통해 프로덕션 워크로드를 지원하여 가용 영역 중단 및 인프라 장애로부터 보호합니다.
- 확장 가능 – OpenSearch Serverless는 리소스를 자동으로 확장하여 지속적으로 빠른 데이터 수집 속도와 쿼리 응답 시간을 유지합니다.

Amazon OpenSearch Serverless란 무엇인가요?

Amazon OpenSearch Serverless는 Amazon OpenSearch Service의 온디맨드 서버리스 옵션으로, OpenSearch 클러스터 프로비저닝, 구성 및 튜닝의 운영 복잡성을 제거합니다. 클러스터를 자체 관리하지 않거나 대규모 배포를 운영하기 위한 전용 리소스와 전문 지식이 부족한 조직에 적합합니다. OpenSearch Serverless를 사용하면 기본 인프라를 관리하지 않고도 대량의 데이터를 검색하고 분석할 수 있습니다.

OpenSearch Serverless 컬렉션은 특정 워크로드 또는 사용 사례를 지원하기 위해 함께 작동하는 OpenSearch 인덱스 그룹입니다. 컬렉션은 수동 프로비저닝이 필요한 자체 관리형 OpenSearch 클러스터에 비해 작업을 간소화합니다.

컬렉션은 프로비저닝된 OpenSearch Service 도메인과 동일한 대용량의 분산된 고가용성 스토리지를 사용하지만 수동 구성 및 튜닝을 제거하여 복잡성을 더욱 줄입니다. 컬렉션 내의 데이터는 전송 중에 암호화됩니다. 또한 OpenSearch Serverless는 OpenSearch Dashboards를 지원하여 데이터 분석을 위한 인터페이스를 제공합니다.

현재 서버리스 컬렉션은 OpenSearch 버전 2.17.x를 실행합니다. 새 버전이 릴리스되면 OpenSearch Serverless는 컬렉션을 자동으로 업그레이드하여 새로운 기능, 버그 수정 및 성능 개선을 통합합니다.

OpenSearch Serverless는 OpenSearch 오픈 소스 제품군과 동일한 수집 및 쿼리 API 작업을 지원하므로 기존 클라이언트와 애플리케이션을 계속 사용할 수 있습니다. OpenSearch Serverless를 사용하려면 클라이언트가 OpenSearch 2.x 버전과 호환되어야 합니다. 자세한 내용은 the section called “컬렉션으로 데이터 수집” 단원을 참조하십시오.

주제

- OpenSearch Serverless 사용 사례
- 작동 방법
- 컬렉션 유형 선택
- 요금
- 지원됨 AWS 리전
- 제한 사항
- OpenSearch Service와 OpenSearch Serverless 비교

OpenSearch Serverless 사용 사례

OpenSearch Serverless는 두 가지 기본 사용 사례를 지원합니다.

- 로그 분석 - 로그 분석 세그먼트는 운영 및 사용자 행동 인사이트를 얻기 위해 대량의 반구조화된 기계 생성 시계열 데이터를 분석하는 데 중점을 둡니다.
- 전체 텍스트 검색 - 전체 텍스트 검색 세그먼트는 내부 네트워크의 애플리케이션(컨텐츠 관리 시스템, 법률 문서)과 전자상거래 웹사이트 콘텐츠 검색과 같은 인터넷 경계 애플리케이션을 지원합니다.

컬렉션을 생성할 때 이러한 사용 사례 중 하나를 선택합니다. 자세한 내용은 the section called “컬렉션 유형 선택” 단원을 참조하십시오.

작동 방법

기존 OpenSearch 클러스터에는 인덱싱 및 검색 작업을 모두 수행하는 단일 인스턴스 세트가 있으며 인덱스 스토리지는 컴퓨팅 용량와 긴밀하게 결합되어 있습니다. 이와 달리 OpenSearch Serverless는 Amazon S3를 인덱스의 기본 데이터 스토리지로 사용하여 인덱싱(수집) 구성 요소를 검색(쿼리) 구성 요소와 분리하는 클라우드 네이티브 아키텍처를 사용합니다.

이 분리된 아키텍처를 사용하면 서로 독립적으로 그리고 S3의 인덱싱된 데이터와 독립적으로 검색 및 인덱싱 기능을 확장할 수 있습니다. 또한 이 아키텍처는 수집 및 쿼리 작업을 격리하여 리소스 경합 없이 동시에 실행할 수 있도록 합니다.

컬렉션에 데이터를 쓸 때 OpenSearch Serverless는 데이터를 인덱싱 컴퓨팅 유닛에 배포합니다. 인덱싱 컴퓨팅 유닛은 수신 데이터를 수집하고 인덱스를 S3로 이동합니다. 컬렉션 데이터에서 검색을 수행하면 OpenSearch Serverless는 쿼리 중인 데이터를 보유하고 있는 검색 컴퓨팅 유닛으로 요청을 라우팅합니다. 검색 컴퓨팅 유닛은 인덱싱된 데이터를 S3에서 직접 다운로드하고(아직 로컬에 캐시되지 않은 경우) 검색 작업을 실행하고 집계를 수행합니다.

다음 이미지는 이 분리된 아키텍처를 보여줍니다.

데이터 수집, 검색 및 쿼리를 위한 OpenSearch Serverless 컴퓨팅 용량은 OpenSearch 컴퓨팅 유닛(OCU)으로 측정됩니다. 각 OCU는 6GiB 메모리와 해당 가상 CPU(vCPU) 및 Amazon S3로의 데이터 전송의 조합입니다. 각 OCU에는 120GiB의 인덱스 데이터를 위한 충분한 핫 임시 스토리지가 포함되어 있습니다.

첫 번째 컬렉션을 생성할 때 OpenSearch Serverless는 두 개의 OCU(하나는 인덱싱용, 다른 하나는 검색용)를 인스턴스화합니다. 또한 고가용성을 보장하기 위해 다른 가용 영역에서 예비 노드 세트를 시작합니다. 개발 및 테스트를 위해 컬렉션에 대한 중복 활성화 설정을 비활성화할 수 있습니다. 그러면 두 개의 대기 복제본이 제거되고 두 개의 OCU만 인스턴스화됩니다. 기본적으로 중복 활성 복제본이 활성화됩니다. 즉, 계정의 첫 번째 컬렉션에 대해 총 4개의 OCU가 인스턴스화됩니다.

이러한 OCU는 컬렉션 엔드포인트에서 활동이 없는 경우에도 존재합니다. 이후의 모든 컬렉션은 이러한 OCU를 공유합니다. 동일한 계정에서 추가 컬렉션을 생성하면 OpenSearch Serverless는 사용자가 지정한 용량 제한에 따라 컬렉션을 지원하는 데 필요한 만큼만 검색 및 수집을 위한 추가 OCU를 추가합니다. 컴퓨팅 사용량이 감소하면 용량이 다시 스케일 다운됩니다.

이러한 OCU에 대해 요금이 청구되는 방식에 대한 자세한 내용은 the section called “요금” 섹션을 참조하세요.

컬렉션 유형 선택

OpenSearch Serverless는 세 가지 기본 컬렉션 유형을 지원합니다.

시계열 - 대량의 반정형 머신 생성 데이터를 실시간으로 분석하여 운영, 보안, 사용자 행동 및 비즈니스 성과에 대한 인사이트를 제공하는 로그 분석 세그먼트입니다.

검색 - 콘텐츠 관리 시스템 및 법률 문서 리포지토리와 같은 내부 네트워크 내의 애플리케이션과 전자 상거래 사이트 검색 및 콘텐츠 검색과 같은 인터넷 연결 애플리케이션을 지원하는 전체 텍스트 검색입니다.

벡터 검색 - 벡터 임베딩에 대한 의미 체계 검색은 벡터 데이터 관리를 간소화하고 기계 학습(ML) 증강 검색 환경을 활성화합니다. 챗봇, 개인 어시스턴트, 사기 탐지와 같은 생성형 AI 애플리케이션을 지원합니다.

컬렉션을 처음 생성할 때 컬렉션 유형을 선택합니다.

선택하는 컬렉션 유형은 컬렉션에 수집하려는 데이터의 종류와 해당 데이터를 쿼리하려는 방식에 따라 다릅니다. 컬렉션 유형을 생성한 후에는 변경할 수 없습니다.

컬렉션 유형에는 다음과 같은 눈에 띄는 차이점이 있습니다.

- 검색 및 벡터 검색 컬렉션의 경우 빠른 쿼리 응답 시간을 보장하기 위해 모든 데이터가 핫 스토리지에 저장됩니다. 시계열 컬렉션은 핫 스토리지와 웜 스토리지의 조합을 사용합니다. 최근 데이터는 핫 스토리지에 보관되어 더 자주 액세스하는 데이터에 대한 쿼리 응답 시간을 최적화합니다.
- 시계열 및 벡터 검색 컬렉션의 경우 사용자 지정 문서 ID별로 인덱싱하거나 업서트 요청별로 업데이트할 수 없습니다. 이 작업은 검색 사용 사례에만 사용됩니다. 대신 문서 ID로 업데이트할 수 있습니다. 자세한 내용은 the section called “지원되는 OpenSearch API 작업 및 권한” 단원을 참조하십시오.
- 검색 및 시계열 컬렉션의 경우 k-NN 유형 인덱스를 사용할 수 없습니다.

요금

AWS 는 다음 OpenSearch Serverless 구성 요소에 대해 요금을 부과합니다.

- 데이터 수집 컴퓨팅
- 검색 및 쿼리 컴퓨팅
- Amazon S3에 보관된 스토리지

OCU 1개는 6GB의 RAM, 해당 vCPU, GP3 스토리지 및 Amazon S3로의 데이터 전송으로 구성됩니다. 청구할 수 있는 가장 작은 단위는 시간 단위로 0.5 OCU. AWS bills OCU이며 초당 세분화됩니다. 계정 문에는 데이터 수집을 위한 레이블과 검색을 위한 레이블이 있는 OCU 시간 단위의 컴퓨팅 항목이 표시됩니다. AWS 또한는 Amazon S3에 저장된 데이터에 대해 매월 요금을 청구합니다. OpenSearch Dashboards 사용에 대해서는 요금이 부과되지 않습니다.

중복 활성 복제본이 있는 컬렉션을 생성하면 최소 2개의 OCUs

- 기본 및 대기를 모두 포함하여 수집용 OCU 1개(0.5 OCU × 2)
- 검색을 위한 OCU 1개(0.5 OCU × 2)

중복 활성 복제본을 비활성화하면 계정의 첫 번째 컬렉션에 대해 최소 1 OCU(0.5 OCU x 2)의 요금이 청구됩니다. 이후의 모든 컬렉션은 이러한 OCU를 공유할 수 있습니다.

OpenSearch Serverless는 컬렉션을 지원하는 데 필요한 컴퓨팅 성능 및 스토리지를 기반으로 추가 OCU(증분 단위: 1개의 OCU)를 추가합니다. 비용을 제어하기 위해 계정에 대한 최대 OCU 수를 구성할 수 있습니다.

> **Note**
>
> 고유한가 있는 컬렉션은 OCUs 다른 컬렉션과 공유할 AWS KMS keys 수 없습니다.

OpenSearch Serverless는 워크로드 변경에 필요한 최소 리소스를 사용하려고 시도합니다. 언제든지 프로비저닝되는 OCUs 수는 다를 수 있으며 정확하지 않습니다. 시간이 지남에 따라 OpenSearch Serverless가 사용하는 알고리즘은 시스템 사용량을 더 잘 최소화하기 위해 계속 개선됩니다.

자세한 내용은 Amazon OpenSearch Service 요금을 참조하세요.

지원됨 AWS 리전

OpenSearch Serverless는 OpenSearch Service를 사용할 수 AWS 리전 있는의 하위 집합에서 사용할 수 있습니다. 지원되는 리전 목록은 AWS 일반 참조의 Amazon OpenSearch Service 엔드포인트 및 할당량을 참조하세요.

제한 사항

OpenSearch Serverless에는 다음과 같은 제한 사항이 있습니다.

- 일부 OpenSearch API 작업은 지원되지 않습니다. the section called “지원되는 OpenSearch API 작업 및 권한”을(를) 참조하세요.
- 일부 OpenSearch 플러그인은 지원되지 않습니다. the section called “지원되는 OpenSearch 플러그인”을(를) 참조하세요.
- 현재 관리형 OpenSearch Service 도메인에서 서버리스 컬렉션으로 데이터를 자동으로 마이그레이션할 수 있는 방법은 없습니다. 도메인에서 컬렉션으로 데이터를 재인덱싱해야 합니다.
- 컬렉션에 대한 크로스 계정 액세스는 지원되지 않습니다. 암호화 또는 데이터 액세스 정책에 다른 계정의 컬렉션을 포함할 수 없습니다.
- 사용자 지정 OpenSearch 플러그인은 지원되지 않습니다.
- OpenSearch Serverless 컬렉션의 스냅샷을 생성하거나 복원할 수 없습니다.
- 교차 리전 간 검색 및 복제는 지원되지 않습니다.
- 단일 계정 및 리전에 보유할 수 있는 서버리스 리소스 수에는 제한이 있습니다. OpenSearch 서버리스 할당량을 참조하세요.
- 벡터 검색 컬렉션의 인덱스 새로 고침 간격은 약 60초입니다. 검색 및 시계열 컬렉션에서 인덱스의 새로 고침 간격은 약 10초입니다.
- 샤드 수, 간격 수, 새로 고침 간격은 수정할 수 없으며 OpenSearch Serverless에서 처리합니다. 샤딩 전략은 컬렉션 유형과 트래픽을 기반으로 합니다. 예를 들어 시계열 컬렉션은 쓰기 트래픽 병목 현상을 기반으로 기본 샤드의 규모를 조정합니다.
- OpenSearch 버전 최대 2.1에서 사용할 수 있는 지리공간 기능이 지원됩니다.

OpenSearch Service와 OpenSearch Serverless 비교

OpenSearch Serverless에서 일부 개념 및 기능은 프로비저닝된 OpenSearch Service 도메인의 해당 기능과 다릅니다. 예를 들어 한 가지 중요한 차이점은 OpenSearch Ingestion에는 클러스터 또는 노드 개념이 없다는 것입니다.

다음 표에서는 OpenSearch Serverless의 중요한 기능 및 개념이 프로비저닝된 OpenSearch Service 도메인의 동등한 기능과 어떻게 다른지 설명합니다.

Feature	OpenSearch Service	OpenSearch Serverless
도메인 대 컬렉션	인덱스는 사전 프로비저닝된 OpenSearch 클러스터인 도메인에 보관됩니다. 자세한 내용은 도메인 생성 및 관리 단원을 참조하십시오.	인덱스는 특정 워크로드 또는 사용 사례를 나타내는 인덱스를 논리적으로 그룹화한 컬렉션에 보관됩니다. 자세한 내용은 the section called “컬렉션 관리” 단원을 참조하십시오.
노드 유형 및 용량 관리	비용 및 성능 사양을 충족하는 노드 유형으로 클러스터를 구축합니다. 자체 스토리지 요구 사항을 계산하고 도메인의 인스턴스 유형을 선택해야 합니다. 자세한 내용은 the section called “도메인 크기 조정” 단원을 참조하십시오.	OpenSearch Serverless는 용량 사용량에 따라 계정에 대한 추가 컴퓨팅 유닛을 자동으로 확장하고 프로비저닝합니다. 자세한 내용은 the section called “용량 제한 관리” 단원을 참조하십시오.
결제	EC2 인스턴스의 사용 시간과 인스턴스에 연결된 EBS 스토리지 볼륨의 누적 크기에 대해 요금을 지불합니다. 자세한 내용은 the section called “요금” 단원을 참조하십시오.	데이터 수집을 위한 컴퓨팅, 검색 및 쿼리를 위한 컴퓨팅, S3에 보관된 스토리지에 대해서는 OCU 시간 단위로 요금이 청구됩니다. 자세한 내용은 the section called “요금” 단원을 참조하십시오.
암호화(Encryption)	저장된 암호화는 도메인에 대한 선택 사항입니다. 자세한 내용은 the section called “저장 시 암호화” 단원을 참조하십시오.	저장된 암호화는 컬렉션에 필수입니다. 자세한 내용은 the section called “암호화” 단원을 참조하십시오.
데이터 액세스 제어	도메인 내 데이터에 대한 액세스는 IAM 정책과 세분화된 액세스 제어에 따라 결정됩니다.	컬렉션 내 데이터에 대한 액세스는 데이터 액세스 정책에 따라 결정됩니다.

Feature	OpenSearch Service	OpenSearch Serverless
지원되는 OpenSearch 작업	OpenSearch Service는 모든 OpenSearch API 작업의 하위 집합을 지원합니다. 자세한 내용은 the section called “지원되는 연산자” 단원을 참조하십시오.	OpenSearch Serverless는 OpenSearch API 작업의 다른 하위 집합을 지원합니다. 자세한 내용은 the section called “지원되는 작업 및 플러그인” 단원을 참조하십시오.
대시보드 로그인	사용자 이름과 암호로 로그인합니다. 자세한 내용은 the section called “마스터 사용자로 OpenSearch 대시보드에 액세스” 단원을 참조하십시오.	AWS 콘솔에 로그인하고 대시보드 URL로 이동하면 자동으로 로그인됩니다. 자세한 내용은 the section called “OpenSearch 대시보드 액세스” 단원을 참조하십시오.
API	OpenSearch Service API 작업을 사용하여 프로그래밍 방식으로 OpenSearch Service와 상호 작용합니다.	OpenSearch Serverless API 작업을 사용하여 프로그래밍 방식으로 OpenSearch Serverless와 상호 작용합니다.
네트워크 액세스	도메인에 대한 네트워크 설정은 OpenSearch 대시보드 엔드포인트뿐만 아니라 도메인 엔드포인트에도 적용됩니다. 두 가지 모두에 대한 네트워크 액세스는 긴밀하게 연결되어 있습니다.	도메인 엔드포인트와 OpenSearch 대시보드 엔드포인트에 대한 네트워크 설정은 분리되어 있습니다. OpenSearch 대시보드에 대한 네트워크 액세스를 구성하지 않도록 선택할 수 있습니다. 자세한 내용은 the section called “네트워크 액세스” 단원을 참조하십시오.
요청에 서명하기	OpenSearch 상위 및 하위 수준 REST 클라이언트를 사용하여 요청에 서명합니다. 서비스 이름을 `es`로 지정합니다.	현재 OpenSearch Serverless는 OpenSearch Service에서 지원하는 클라이언트의 하위 집합을 지원합니다. 요청에 서명할 때 서비스 이름을 `aoss`로 지정합니다. `x-amz-content-sha256` 헤더는 필수입니다. 자세한 내용은 the section called “기타 클라이언트” 단원을 참조하십시오.

Feature	OpenSearch Service	OpenSearch Serverless
OpenSearch 버전 업그레이드	새 버전의 OpenSearch를 사용할 수 있게 되면 도메인을 수동으로 업그레이드해야 합니다. 도메인이 업그레이드 요구 사항을 충족하고 모든 주요 변경 사항을 해결했는지 확인할 책임은 귀하에게 있습니다.	OpenSearch Serverless는 컬렉션을 새 OpenSearch 버전으로 자동 업그레이드합니다. 새 버전이 출시되자마자 업그레이드가 반드시 이루어지는 것은 아닙니다.
서비스 소프트웨어 업데이트	서비스 소프트웨어 업데이트가 제공되면 도메인에 서비스 소프트웨어 업데이트를 수동으로 적용합니다.	OpenSearch Serverless는 컬렉션을 자동으로 업데이트하여 최신 버그 수정, 기능, 성능 개선 사항을 사용합니다.
VPC 액세스	VPC 내에서 도메인을 프로비저닝할 수 있습니다. 도메인에 액세스하기 위해 추가 OpenSearch Service 관리형 VPC 엔드포인트를 생성할 수도 있습니다.	계정에 대해 OpenSearch Serverless 관리형 VPC 엔드포인트를 하나 이상 생성합니다. 그런 다음 이러한 엔드포인트를 네트워크 정책에 포함합니다.
SAML 인증	도메인별로 SAML 인증을 활성화합니다. 자세한 내용은 the section called “OpenSearch 대시보드에 대한 SAML 인증” 단원을 참조하십시오.	계정 수준에서 하나 이상의 SAML 공급자를 구성한 다음 연결된 사용자 및 그룹 ID를 데이터 액세스 정책에 포함합니다. 자세한 내용은 the section called “SAML 인증” 단원을 참조하십시오.
전송 계층 보안(TLS)	OpenSearch Service는 TLS 1.2를 지원하지만 TLS 1.3을 사용하는 것이 좋습니다.	OpenSearch Serverless는 TLS 1.2를 지원하지만 TLS 1.3을 사용하는 것이 좋습니다.

자습서: Amazon OpenSearch Serverless 시작하기

이 자습서에서는 Amazon OpenSearch Serverless 검색 컬렉션을 신속하게 시작하고 실행하기 위한 기본 단계를 안내합니다. 검색 컬렉션을 사용하면 내부 네트워크의 애플리케이션과 전자상거래 웹사이트 검색 및 콘텐츠 검색과 같은 인터넷 경계 애플리케이션을 지원할 수 있습니다.

벡터 검색 컬렉션을 사용하는 방법을 알아보려면 the section called “벡터 검색 컬렉션 작업”을 참조하세요. 컬렉션 사용에 대한 자세한 내용은 이 설명서의 the section called “컬렉션 관리” 및 기타 주제 섹션을 참조하세요.

이 자습서에서는 다음 단계를 완료합니다.

1. 권한 구성
2. 컬렉션 생성
3. 데이터 업로드 및 검색
4. 컬렉션 삭제

> **Note**
>
> 에는 ASCII 문자만 사용하는 것이 좋습니다`IndexName`. `IndexName`에 ASCII 문자를 사용하지 않으면 CloudWatch 지표의 `IndexName`이 ASCII 문자가 아닌 문자인 경우 URL 인코딩 형식으로 변환됩니다.

1단계: 권한 구성

이 자습서를 완료하고 일반적으로 OpenSearch Serverless를 사용하려면 올바른 IAM 권한이 있어야 합니다. 이 자습서에서는 컬렉션을 생성하고 데이터를 업로드하고 검색한 다음 컬렉션을 삭제합니다.

사용자 또는 역할에는 다음과 같은 최소 권한이 포함된 연결된 자격 증명 기반 정책이 있어야 합니다.

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": "*"
    }
  ]
}
```

OpenSearch Serverless IAM 권한에 대한 자세한 내용은 the section called “ID 및 액세스 관리” 섹션을 참조하세요.

2단계: 컬렉션 생성

컬렉션은 특정 워크로드 또는 사용 사례를 지원하기 위해 함께 작동하는 OpenSearch 인덱스 그룹입니다.

OpenSearch Serverless 컬렉션 생성하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Collections(컬렉션)를 선택하고 Create collection(컬렉션 생성)을 선택합니다.
3. 컬렉션 이름을 movies(영화)로 지정합니다.
4. 컬렉션 유형에서 Search(검색)를 선택합니다. 자세한 내용은 컬렉션 유형 선택을 참조하세요.
5. 보안에서 표준 생성을 선택합니다.
6. 암호화에서 AWS 소유 키사용을 선택합니다. OpenSearch Serverless AWS KMS key 가 데이터를 암호화하는 데 사용하는 입니다.
7. Network(네트워크)에서 컬렉션에 대한 네트워크 설정을 구성합니다.
   - 액세스 유형으로 Public(퍼블릭)을 선택합니다.

- 리소스 유형의 경우 OpenSearch 엔드포인트에 대한 액세스와 OpenSearch 대시보드에 대한 액세스를 모두 활성화합니다. OpenSearch 대시보드를 사용하여 데이터를 업로드하고 검색하므로 둘 다 활성화해야 합니다.

8. 다음을 선택합니다.
9. Configure data access(데이터 액세스 구성)에서 컬렉션에 대한 액세스 설정을 지정합니다. 데이터 액세스 정책을 사용하면 사용자 및 역할이 컬렉션 내의 데이터에 액세스할 수 있습니다. 이 자습서에서는 단일 사용자에게 movies(영화) 컬렉션의 데이터를 인덱싱하고 검색하는 데 필요한 권한을 제공합니다.

   movies 컬렉션에 대한 액세스를 제공하는 단일 규칙을 생성합니다. 규칙 이름을 Movies collection access(Movies 컬렉션 액세스)로 지정합니다.
10. Add principals(보안 주체 추가), IAM users and roles(IAM 사용자 및 역할)를 선택하고 OpenSearch 대시보드에 로그인하고 데이터를 인덱싱하는 데 사용할 사용자 또는 역할을 선택합니다. 저장을 선택합니다.
11. Index permissions(인덱스 권한)에서 모든 권한을 선택합니다.
12. 다음을 선택합니다.
13. 액세스 정책 설정에서 Create a new data access policy(새 데이터 액세스 정책 생성)를 선택하고 정책 이름을 movies로 지정합니다.
14. 다음을 선택합니다.
15. 컬렉션 설정을 검토하고 Submit(제출)을 선택합니다. 컬렉션이 `Active` 상태가 될 때까지 몇 분 정도 기다립니다.

3단계: 데이터 업로드 및 검색

Postman 또는 cURL을 사용하여 OpenSearch Serverless 컬렉션에 데이터를 업로드할 수 있습니다. 간결하게 하기 위해 이러한 예시는 OpenSearch 대시보드 콘솔의 Dev Tools(개발 도구)를 사용합니다.

movies(영화) 컬렉션에서 데이터를 인덱싱하고 검색하기

1. 왼쪽 탐색 창에서 Collections(컬렉션)를 선택하고 movies(영화) 컬렉션을 선택하여 세부 정보 페이지를 엽니다.
2. 컬렉션에 대한 OpenSearch 대시보드 URL을 선택합니다. URL은 `https://dashboards.{region}.aoss.amazonaws.com/_login/?collectionId={collection-id}` 형식을 취합니다.
3. OpenSearch 대시보드에서 왼쪽 탐색 창을 열고 Dev Tools(개발 도구)를 선택합니다.

4. movies-index라는 단일 인덱스를 생성하려면 다음 요청을 보냅니다.

```
PUT movies-index
```

5. 단일 문서를 movies-index로 인덱싱하려면 다음 요청을 보냅니다.

```
PUT movies-index/_doc/1
{
  "title": "Shawshank Redemption",
  "genre": "Drama",
  "year": 1994
}
```

6. OpenSearch 대시보드에서 데이터를 검색하려면 하나 이상의 인덱스 패턴을 구성해야 합니다. OpenSearch는 이러한 패턴을 사용하여 분석할 인덱스를 식별하기 때문입니다. 왼쪽 탐색 창을 열고 Stack Management(스택 관리)를 선택하고 Index Patterns(인덱스 패턴)를 선택한 다음 Create index pattern(인덱스 패턴 생성)을 선택합니다. 본 자습서에서는 movies를 입력합니다.
7. 다음 단계(Next step)를 선택한 후 인덱스 패턴 생성(Create index pattern)을 선택합니다. 패턴이 생성되면 `title`, `genre`와 같은 다양한 문서 필드를 볼 수 있습니다.
8. 데이터 검색을 시작하려면 왼쪽 탐색 창을 다시 열고 Discover(검색)를 선택하거나 개발 도구 내의 검색 API를 사용합니다.

4단계: 컬렉션 삭제

movies(영화) 컬렉션은 테스트용이므로 실험을 마치면 삭제해야 합니다.

OpenSearch Serverless 컬렉션 삭제하기

1. Amazon OpenSearch Service 콘솔로 돌아갑니다.
2. 왼쪽 탐색 창에서 Collections(컬렉션)를 선택하고 movies(영화) 컬렉션을 선택합니다.
3. [삭제(Delete)]를 선택하고 삭제 의사를 확인합니다.

다음 단계

컬렉션과 인덱스 데이터를 생성하는 방법을 알았으므로 다음 연습을 시도해볼 수 있습니다.

- 컬렉션 생성을 위한 고급 옵션을 참조하세요. 자세한 내용은 the section called “컬렉션 관리” 단원을 참조하십시오.
- 컬렉션 보안을 대규모로 관리하기 위해 보안 정책을 구성하는 방법을 알아보세요. 자세한 내용은 the section called “OpenSearch Serverless의 보안” 단원을 참조하십시오.
- 데이터를 컬렉션으로 인덱싱하는 다른 방법을 알아보세요. 자세한 내용은 the section called “컬렉션으로 데이터 수집” 단원을 참조하십시오.

Amazon OpenSearch Serverless 컬렉션 생성 및 관리

콘솔, AWS CLI 및 API AWS , SDK 및를 사용하여 Amazon OpenSearch Serverless 컬렉션을 생성할 수 있습니다 AWS CloudFormation. SDKs

주제

- Amazon OpenSearch Serverless 컬렉션 관리
- 벡터 검색 컬렉션 작업
- Amazon OpenSearch Serverless를 통한 데이터 수명 주기 정책 사용
- AWS SDKs를 사용하여 Amazon OpenSearch Serverless와 상호 작용
- AWS CloudFormation 를 사용하여 Amazon OpenSearch Serverless 컬렉션 생성
- 스냅샷을 사용하여 컬렉션 백업

Amazon OpenSearch Serverless 컬렉션 관리

Amazon OpenSearch Serverless의 컬렉션은 분석 워크로드를 나타내는 하나 이상의 인덱스를 논리적으로 그룹화한 것입니다. OpenSearch Serverless는 수동 입력을 최소화하면서 컬렉션을 자동으로 관리하고 조정합니다.

주제

- 컬렉션에 대한 권한 구성
- 자동 의미 체계 보강 정보
- 컬렉션 생성
- OpenSearch 대시보드 액세스
- 컬렉션 보기
- 컬렉션 삭제

컬렉션에 대한 권한 구성

OpenSearch Serverless는 다음 AWS Identity and Access Management (IAM) 권한을 사용하여 컬렉션을 생성하고 관리합니다. 사용자를 특정 컬렉션으로 제한하도록 IAM 조건을 지정할 수 있습니다.

- `aoss:CreateCollection` – 컬렉션을 생성합니다.
- `aoss:ListCollections` – 현재 계정의 컬렉션을 나열합니다.
- `aoss:BatchGetCollection` – 하나 이상의 컬렉션에 대한 세부 정보를 가져옵니다.
- `aoss:UpdateCollection` – 컬렉션을 수정합니다.
- `aoss:DeleteCollection` – 컬렉션을 삭제합니다.

다음 샘플 자격 증명 기반 액세스 정책은 사용자가 `Logs`라는 단일 컬렉션을 관리하는 데 필요한 최소 권한을 제공합니다.

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

컬렉션이 제대로 작동하려면 암호화, 네트워크, 데이터 액세스 정책이 필요하기 때문에 `aoss:CreateAccessPolicy` 및 `aoss:CreateSecurityPolicy`가 포함됩니다. 자세한 내용은 the section called “ID 및 액세스 관리” 단원을 참조하십시오.

> **Note**
>
> 계정에서 첫 번째 컬렉션을 생성하려면 `iam:CreateServiceLinkedRole` 권한도 필요합니다. 자세한 내용은 the section called “컬렉션 생성 역할” 단원을 참조하십시오.

자동 의미 체계 보강 정보

컬렉션을 생성하거나 편집할 때 Amazon OpenSearch Service의 의미 체계 검색 구현 및 기능을 간소화하는 자동 의미 체계 보강을 구성할 수 있습니다. 의미 체계 검색은 키워드 일치뿐만 아니라 사용자 검색의 의도 및 컨텍스트 의미를 포함하는 쿼리 결과를 반환합니다. 예를 들어 사용자가 "두통 치료 방법"을 검색하는 경우 의미 체계 검색 시스템은 다음 결과를 반환할 수 있습니다.

- 세분화된 해결 방법
- 통증 관리 기법
- Over-the-counter 완화제
- 자연두통 완화 방법

시스템은 이러한 정확한 문구가 원래 쿼리에 없는 경우에도 기본 의도를 이해합니다.

자동 의미 체계 보강은 다음과 같은 이점을 제공합니다.

간소화된 구현

기계 학습(ML) 전문 지식이나 복잡한 통합이 필요하지 않습니다.

자동화된 프로세스

의미 체계 보강은 데이터 수집 중에 자동으로 수행됩니다.

검색 관련성 개선

의미 체계 보강은 검색 결과의 품질과 컨텍스트 정확도를 향상시킵니다.

확장성

의미 체계 보강은 수동 개입 없이 대규모 데이터 세트에 의미 체계 검색을 적용합니다.

작동 방법

자동 의미 체계 보강을 시작하려면 컬렉션을 생성하거나 편집하고 데이터의 어떤 필드에 의미 체계 검색 기능이 필요한지 지정합니다. 의미 체계 검색을 위한 필드를 식별한 후 데이터가 OpenSearch Service로 들어오면 자동 의미 체계 보강 프로세스가 이러한 필드를 자동으로 보강합니다. 보강된 데이터는 보다 지능적이고 컨텍스트 인식 검색을 지원합니다.

> **Note**
>
> 자동 의미 체계 보강을 구현할 때는 다음 요소를 고려하세요.
>
> - 처리 오버헤드: 보강 프로세스는 수집 중에 처리 시간을 추가할 수 있습니다.
> - 스토리지 영향: 데이터가 풍부해지면 추가 스토리지 공간이 필요합니다.
> - 언어 제한: 다국어 옵션이 필요한 언어를 지원하는지 확인합니다.

서버리스에 대한 자동 의미 체계 보강은 다음과 같은 언어 옵션을 제공합니다.

영어 전용 옵션

- 영어 콘텐츠에 최적화
- 주로 영어 텍스트를 처리하는 애플리케이션에 이상적

다국어 옵션

- 아랍어, 벵골어, 중국어, 영어, 핀란드어, 프랑스어, 힌디어, 인도네시아어, 일본어, 한국어, 페르시아어, 러시아어, 스페인어, 스와힐리어, 텔루구어 지원
- 다양한 국제 콘텐츠 또는 다국어 애플리케이션에 적합

자동 의미 체계 보강에 대한 권한 구성

자동 의미 체계 보강 인덱스를 생성하기 전에 필요한 권한을 구성해야 합니다. 이 섹션에서는 필요한 권한과 이를 설정하는 방법을 설명합니다.

IAM 정책 권한

다음 AWS Identity and Access Management (IAM) 정책을 사용하여 자동 의미 체계 보강 작업에 필요한 권한을 부여합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AutomaticSemanticEnrichmentPermissions",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateIndex",
                "aoss:GetIndex",
                "aoss:UpdateIndex",
                "aoss:DeleteIndex",
                "aoss:APIAccessAll"
            ],
            "Resource": "*"
        }
    ]
}
```

키 권한

- `aoss:*Index` 권한은 인덱스 관리를 활성화합니다.
- `aoss:APIAccessAll` 권한은 OpenSearch API 작업을 허용합니다.
- 특정 컬렉션에 대한 권한을 제한하려면를 컬렉션의 ARN`"Resource": "*"`으로 바꿉니다.

데이터 액세스 권한 구성

자동 의미 체계 보강을 위한 인덱스를 설정하려면 인덱스, 파이프라인 및 모델 컬렉션 리소스에 액세스할 수 있는 권한을 부여하는 적절한 데이터 액세스 정책이 있어야 합니다. 데이터 액세스 정책에 대한 자세한 내용은 섹션을 참조하세요Amazon OpenSearch Serverless를 위한 데이터 액세스 제어. 데이터 액세스 정책을 구성하는 절차는 섹션을 참조하세요데이터 액세스 정책 생성(콘솔).

데이터 액세스 권한

```
[
    {
        "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:CreateMLResources"]
            }
        ],
        "Principal": [
            "arn:aws:iam::account_id:role/role_name"
        ]
    },
```

```
]
```

네트워크 액세스 권한

서비스 APIs 프라이빗 컬렉션에 액세스하도록 허용하려면 서비스 API와 컬렉션 간에 필요한 액세스를 허용하는 네트워크 정책을 구성해야 합니다. 네트워크 정책에 대한 자세한 내용은 Amazon OpenSearch Serverless에 대한 네트워크 액세스를 참조하세요.

```
[
   {
      "Description":"Enable automatic semantic enrichment in a private collection",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/collection_name"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceServices":[
         "aoss.amazonaws.com"
      ],
   }
]
```

프라이빗 컬렉션에 대한 네트워크 액세스 권한을 구성하려면

1. https://console.aws.amazon.com/aos/home OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 네트워크 정책을 선택합니다. 그런 다음, 다음 중 하나를 수행하세요.
   - 기존 정책 이름을 선택하고 편집을 선택합니다.
   - 네트워크 정책 생성을 선택하고 정책 세부 정보를 구성합니다.
3. 액세스 유형 영역에서 프라이빗(권장)을 선택한 다음 AWS 서비스 프라이빗 액세스를 선택합니다.
4. 검색 필드에서 서비스를 선택한 다음 aoss.amazonaws.com 선택합니다.
5. 리소스 유형 영역에서 OpenSearch 엔드포인트에 대한 액세스 활성화 상자를 선택합니다.
6. 검색 컬렉션(들) 또는 특정 접두사 용어(들) 입력의 검색 필드에서 컬렉션 이름을 선택합니다. 그런 다음 네트워크 정책과 연결할 컬렉션의 이름을 입력하거나 선택합니다.
7. 새 네트워크 정책에 대해 생성 또는 기존 네트워크 정책에 대해 업데이트를 선택합니다.

컬렉션 생성

콘솔 또는를 사용하여 서버리스 컬렉션을 AWS CLI 생성할 수 있습니다. 다음 단계에서는 검색 또는 시계열 컬렉션을 만드는 방법을 다룹니다. 벡터 검색 컬렉션을 만들려면 the section called “벡터 검색 컬렉션 작업”(을)를 참조하세요.

주제

- 컬렉션 생성(콘솔)
- 컬렉션 생성(CLI)

컬렉션 생성(콘솔)

이 섹션의 절차에 따라를 사용하여 컬렉션을 생성합니다 AWS Management Console. 다음 단계에서는 검색 또는 시계열 컬렉션을 만드는 방법을 다룹니다. 벡터 검색 컬렉션을 만들려면 the section called “벡터 검색 컬렉션 작업”(을)를 참조하세요.

주제

- 컬렉션 설정 구성
- 추가 검색 필드 구성

컬렉션 설정 구성

다음 절차에 따라 컬렉션에 대한 정보를 구성합니다.

콘솔을 사용하여 컬렉션 설정을 구성하려면

1. https://console.aws.amazon.com/aos/home/에서 Amazon OpenSearch Service 콘솔로 이동합니다.
2. 왼쪽 탐색 창에서 Serverless(서버리스)를 확장하고 Collections(컬렉션)를 선택합니다.
3. Create collection(컬렉션 생성)을 선택합니다.
4. 컬렉션의 이름과 설명을 입력합니다. 이름은 다음 조건을 충족해야 합니다.
    - 계정 및에 고유합니다. AWS 리전
    - 소문자 a~z, 숫자 0~9 및 하이픈(-)만 포함할 것
    - 3~32자 사이일 것
5. 컬렉션 유형 선택:

- Time series(시계열) – 대량의 반구조화된 기계 생성 데이터를 분석하는 데 중점을 둔 로그 분석 세그먼트입니다. 최소 24시간 분량의 데이터는 핫 인덱스에 저장되고 나머지는 웜 스토리지에 유지됩니다.
- Search(검색) – 내부 네트워크 및 인터넷 연결 애플리케이션의 애플리케이션을 지원하는 전체 텍스트 검색입니다. 모든 검색 데이터는 빠른 쿼리 응답 시간을 보장하기 위해 핫 스토리지에 저장됩니다.

> **Note**
>
> 에 설명된 대로 자동 의미 체계 검색을 활성화하는 경우이 옵션을 선택합니다컬렉션 설정 구성.

- 벡터 검색 - 벡터 데이터 관리를 간소화하는 벡터 임베딩에 대한 시맨틱 검색입니다. 기계 학습(ML) 증강 검색 경험과 챗봇, 개인 비서, 사기 탐지와 같은 생성형 AI 애플리케이션을 강화합니다.

자세한 내용은 the section called “컬렉션 유형 선택” 단원을 참조하십시오.

6. 배포 유형에서 컬렉션의 중복성 설정을 선택합니다. 기본적으로 각 컬렉션에는 중복성이 있습니다. 즉, 인덱싱 및 검색 OpenSearch 컴퓨팅 유닛(OCUs)에는 각각 다른 가용 영역에 자체 대기 복제본이 있습니다. 개발 및 테스트를 위해 중복을 비활성화하도록 선택할 수 있습니다. 이렇게 하면 컬렉션의 OCU 수가 2개로 줄어듭니다. 자세한 내용은 the section called “작동 방법” 단원을 참조하십시오.
7. 보안에서 표준 생성을 선택합니다.
8. 암호화에서 데이터를 암호화할 AWS KMS 키를 선택합니다. OpenSearch Serverless는 입력한 컬렉션 이름이 암호화 정책에 정의된 패턴과 일치하는 경우 알려줍니다. 이 일치 항목을 유지하거나 고유한 암호화 설정으로 재정의하도록 선택할 수 있습니다. 자세한 내용은 the section called “암호화” 단원을 참조하십시오.
9. 네트워크 액세스 설정에서 컬렉션에 대한 네트워크 액세스를 구성합니다.

   - 액세스 유형에서 퍼블릭 또는 프라이빗을 선택합니다.

     프라이빗을 선택하는 경우 컬렉션에 액세스할 AWS 서비스 수 있는 VPC 엔드포인트와를 지정합니다.

     - 액세스를 위한 VPC 엔드포인트 - 액세스할 때 통과할 하나 이상의 VPC 엔드포인트를 지정합니다. VPC 엔드포인트를 생성하려면 the section called “VPC 엔드포인트”를 참조하세요.

  - AWS 서비스 프라이빗 액세스 - 액세스를 허용할 지원되는 서비스를 하나 이상 선택합니다.
- 리소스 유형에서 사용자가 OpenSearch 엔드포인트(cURL, Postman 등을 통해 API 직접 호출), OpenSearch Dashboards 엔드포인트(시각화 작업 및 콘솔을 통해 API 직접 호출) 또는 둘 다를 통해 컬렉션에 액세스할 수 있는지 여부를 선택합니다.

> **Note**
>
> AWS 서비스 프라이빗 액세스는 OpenSearch Dashboards 엔드포인트가 아닌 OpenSearch 엔드포인트에만 적용됩니다.

OpenSearch Serverless는 입력한 컬렉션 이름이 네트워크 정책에 정의된 패턴과 일치하는 경우 알려줍니다. 이 일치 항목을 유지하거나 사용자 지정 네트워크 설정으로 재정의하도록 선택할 수 있습니다. 자세한 내용은 the section called “네트워크 액세스” 단원을 참조하십시오.

10. (선택 사항) 컬렉션에 하나 이상의 태그를 추가합니다. 자세한 내용은 the section called “컬렉션 태그 지정” 단원을 참조하십시오.
11. 다음을 선택합니다.

추가 검색 필드 구성

컬렉션 생성 워크플로의 2페이지에 표시되는 옵션은 생성하는 컬렉션 유형에 따라 다릅니다. 이 섹션에서는 각 컬렉션 유형에 대해 추가 검색 필드를 구성하는 방법을 설명합니다. 이 섹션에서는 자동 의미 체계 보강을 구성하는 방법도 설명합니다. 컬렉션 유형에 적용되지 않는 섹션은 건너뜁니다.

주제

- 자동 의미 체계 보강 구성
- 시계열 검색 필드 구성
- 어휘 검색 필드 구성
- 벡터 검색 필드 구성

자동 의미 체계 보강 구성

컬렉션을 생성하거나 편집할 때 Amazon OpenSearch Service의 의미 체계 검색 구현 및 기능을 간소화하는 자동 의미 체계 보강을 구성할 수 있습니다. 의미 체계 검색은 키워드 일치뿐만 아니라 사용자 검색의 의도 및 컨텍스트 의미를 포함하는 쿼리 결과를 반환합니다. 자세한 내용은 자동 의미 체계 보강 정보 단원을 참조하십시오.

자동 의미 체계 보강을 구성하려면

1. 인덱스 세부 정보 섹션의 인덱스 이름에 이름을 지정합니다.
2. 자동 의미 체계 보강 필드 섹션에서 의미 체계 검색 필드 추가를 선택합니다.
3. 의미 체계 보강의 입력 필드 이름에 보강하려는 필드의 이름을 입력합니다.
4. 데이터 형식은 텍스트입니다. 이 값은 변경할 수 없습니다.
5. 언어에서 영어 또는 다국어를 선택합니다.
6. 필드 추가를 선택합니다.
7. 컬렉션에 대한 선택적 필드 구성을 완료한 후 다음을 선택합니다. 변경 사항을 검토하고 제출을 선택하여 컬렉션을 생성합니다.

시계열 검색 필드 구성

시계열 검색 필드 섹션의 옵션은 시계열 데이터 및 데이터 스트림과 관련이 있습니다. 이러한 주제에 대한 자세한 내용은 섹션을 참조하세요데이터 스트림을 사용하여 Amazon OpenSearch Service에서 시계열 데이터 관리.

시계열 검색 필드를 구성하려면

1. 시계열 검색 필드 섹션에서 시계열 필드 추가를 선택합니다.
2. 필드 이름에 이름을 입력합니다.
3. 데이터 유형의 경우 목록에서 유형을 선택합니다.
4. 필드 추가를 선택합니다.
5. 컬렉션에 대한 선택적 필드 구성을 완료한 후 다음을 선택합니다. 변경 사항을 검토하고 제출을 선택하여 컬렉션을 생성합니다.

어휘 검색 필드 구성

어휘 검색은 검색 쿼리와 인덱싱된 용어 또는 키워드 간의 정확한 일치를 찾습니다.

어휘 검색 필드를 구성하려면

1. 성 검색 필드 섹션에서 검색 필드 추가를 선택합니다.
2. 필드 이름에 이름을 입력합니다.
3. 데이터 유형의 경우 목록에서 유형을 선택합니다.
4. 필드 추가를 선택합니다.

5. 컬렉션에 대한 선택적 필드 구성을 완료한 후 다음을 선택합니다. 변경 사항을 검토하고 제출을 선택하여 컬렉션을 생성합니다.

벡터 검색 필드 구성

벡터 검색 필드를 구성하려면

1. 벡터 필드 섹션에서 벡터 필드 추가를 선택합니다.
2. 필드 이름에 이름을 입력합니다.
3. 엔진의 경우 목록에서 유형을 선택합니다.
4. 차원 수를 입력합니다.
5. 거리 지표의 경우 목록에서 유형을 선택합니다.
6. 컬렉션에 대한 선택적 필드 구성을 완료한 후 다음을 선택합니다.
7. 변경 사항을 검토하고 제출을 선택하여 컬렉션을 생성합니다.

컬렉션 생성(CLI)

이 섹션의 절차에 따라를 사용하여 OpenSearch Serverless 컬렉션을 생성합니다 AWS CLI.

주제

- 시작하기 전 준비 사항
- 컬렉션 생성
- 자동 의미 체계 보강 인덱스를 사용하여 컬렉션 생성

시작하기 전 준비 사항

를 사용하여 컬렉션을 생성하기 전에 다음 절차에 AWS CLI따라 컬렉션에 필요한 정책을 생성합니다.

> **Note**
>
> 다음 각 절차에서 컬렉션의 이름을 지정할 때 이름은 다음 기준을 충족해야 합니다.
>
> - 계정 및에 고유합니다. AWS 리전
> - 소문자 a~z, 숫자 0~9 및 하이픈(-)만 포함할 것
> - 3~32자 사이일 것

컬렉션에 필요한 정책을 생성하려면

1. 를 열고 다음 명령을 AWS CLI 실행하여 컬렉션의 의도한 이름과 일치하는 리소스 패턴으로 암호화 정책을 생성합니다.

```
aws opensearchserverless create-security-policy \
  --name policy name \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/collection name\"]}],\"AWSOwnedKey\":true}"
```

   예를 들어 컬렉션 로그 애플리케이션의 이름을 지정하려는 경우 다음과 같은 암호화 정책을 생성할 수 있습니다.

```
aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/logs-application\"]}],\"AWSOwnedKey\":true}"
```

   정책을 추가 컬렉션에 사용하려는 경우 `collection/logs*` 또는 `collection/*`과 같이 규칙을 더 광범위하게 만들 수 있습니다.

2. 다음 명령을 실행하여 네트워크 정책을 사용하여 컬렉션에 대한 네트워크 설정을 구성합니다. 컬렉션을 생성한 후에 네트워크 정책을 생성해도 되지만, 네트워크 정책은 컬렉션보다 먼저 생성하는 것이 좋습니다.

```
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}]"
```

   이전 로그 애플리케이션 예시를 사용하여 다음과 같은 네트워크 정책을 생성할 수 있습니다.

```
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}]"
```

컬렉션 생성

다음 절차에서는 CreateCollection API 작업을 사용하여 SEARCH 또는 유형의 컬렉션을 생성합니다TIMESERIES. 요청에 컬렉션 유형을 지정하지 않으면 기본값은 TIMESERIES입니다. 이러한 유형에 대한 자세한 내용은 섹션을 참조하세요컬렉션 유형 선택. 벡터 검색 컬렉션을 만들려면 the section called “벡터 검색 컬렉션 작업”(을)를 참조하세요.

컬렉션이 로 암호화된 경우 AWS 소유 키kmsKeyArn는 ARNauto이 아니라 입니다.

> **Important**
>
> 컬렉션을 생성한 후에는 데이터 액세스 정책과 일치하지 않는 한 컬렉션에 액세스할 수 없습니다. 자세한 내용은 Amazon OpenSearch Serverless를 위한 데이터 액세스 제어 단원을 참조하십시오.

컬렉션 생성

1. 에 설명된 필수 정책을 생성했는지 확인합니다시작하기 전 준비 사항.
2. 다음 명령을 실행합니다. 에 SEARCH 또는를 type 지정합니다TIMESERIES.

```
aws opensearchserverless create-collection --name "collection name" --
type collection type --description "description"
```

자동 의미 체계 보강 인덱스를 사용하여 컬렉션 생성

다음 절차에 따라 자동 의미 체계 보강을 위해 구성된 인덱스를 사용하여 새 OpenSearch Serverless 컬렉션을 생성합니다. 이 절차에서는 OpenSearch Serverless CreateIndex API 작업을 사용합니다.

인덱스가 자동 의미 체계 보강을 위해 구성된 새 컬렉션을 생성하려면

다음 명령을 실행하여 컬렉션과 인덱스를 생성합니다.

```
aws opensearchserverless create-index \
--region Region ID \
--id collection name --index-name index name \
--index-schema \
'mapping in json'
```

다음은 그 예입니다.

```
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"
                }
            }
        }
    }
}'
```

OpenSearch 대시보드 액세스

를 사용하여 컬렉션을 생성한 후 컬렉션의 OpenSearch Dashboards URL로 이동할 AWS Management Console수 있습니다. 왼쪽 탐색 창에서 컬렉션을 선택하고 해당 컬렉션을 선택한 다음 세부 정보 페이지를 열면 Dashboards URL을 찾을 수 있습니다. URL은 `https://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc` 형식을 취합니다. URL을 탐색하면 Dashboards에 자동으로 로그인됩니다.

OpenSearch 대시보드 URL을 이미 사용할 수 있지만에 없는 경우 브라우저에서 대시보드 URL을 AWS Management Console호출하면 콘솔로 리디렉션됩니다. 자격 AWS 증명을 입력하면 대시보드에 자동으로 로그인됩니다. SAML용 컬렉션에 액세스하는 방법에 대한 자세한 내용은 SAML을 통한 OpenSearch Dashboards 액세스를 참조하세요.

OpenSearch Dashboards 콘솔 제한 시간은 1시간이며 구성할 수 없습니다.

> **Note**
>
> 2023년 5월 10일, OpenSearch는 OpenSearch Dashboards를 위한 공통 글로벌 엔드포인트를 도입했습니다. 이제 브라우저에서 `https://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc` 형식을 취하는 URL을 사용하여 OpenSearch Dashboards로 이동할 수 있습니다. 이전 버전과의 호환성을 보장하기 위해 기존 컬렉션별 OpenSearch Dashboards 엔드포인트를 `https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/_dashboards` 형식으로 계속 지원할 예정입니다.

컬렉션 보기

Amazon OpenSearch Service 콘솔의 컬렉션 탭에서의 기존 AWS 계정 컬렉션을 볼 수 있습니다.

컬렉션을 ID와 함께 나열하려면 ListCollections 요청을 보냅니다.

```
aws opensearchserverless list-collections
```

샘플 응답

```
{
   "collectionSummaries":[
      {
         "arn":"arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "id":"07tjusf2h91cunochc",
         "name":"my-collection",
         "status":"CREATING"
      }
   ]
}
```

검색 결과를 제한하려면 컬렉션 필터를 사용합니다. 이 요청은 ACTIVE 상태의 컬렉션에 대한 응답을 필터링합니다.

```
aws opensearchserverless list-collections --collection-filters '{ "status": "ACTIVE" }'
```

OpenSearch 엔드포인트 및 OpenSearch 대시보드 엔드포인트를 포함하여 하나 이상의 컬렉션에 대한 자세한 정보를 보려면 BatchGetCollection 요청을 보냅니다.

```
aws opensearchserverless batch-get-collection --ids ["07tjusf2h91cunochc",
 "1iu5usc4rame"]
```

> **Note**
>
> 요청에 --names 또는 --ids를 포함할 수 있지만 둘 다 포함할 수는 없습니다.

샘플 응답

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


컬렉션 삭제

컬렉션을 삭제하면 컬렉션의 모든 데이터와 인덱스가 삭제됩니다. 컬렉션을 삭제한 후에는 복구할 수 없습니다.

콘솔을 사용하여 컬렉션 삭제하기

1. Amazon OpenSearch Service 콘솔의 Collections(컬렉션) 패널에서 삭제할 컬렉션을 선택합니다.
2. [삭제(Delete)]를 선택하고 삭제 의사를 확인합니다.

를 사용하여 컬렉션을 삭제하려면 DeleteCollection 요청을 AWS CLI보냅니다.

```
aws opensearchserverless delete-collection --id 07tjusf2h91cunochc
```

샘플 응답

```
{
   "deleteCollectionDetail":{
      "id":"07tjusf2h91cunochc",
      "name":"my-collection",
      "status":"DELETING"
   }
}
```

벡터 검색 컬렉션 작업

OpenSearch Serverless의 벡터 검색 컬렉션 유형은 확장 가능하고 성능이 우수한 유사성 검색 기능을 제공합니다. 이를 통해 기본 벡터 데이터베이스 인프라를 관리할 필요 없이 최신 기계 학습(ML) 증강 검색 경험과 생성형 AI애플리케이션을 쉽게 구축할 수 있습니다.

벡터 검색 컬렉션의 사용 사례에는 이미지 검색, 문서 검색, 음악 검색, 제품 추천, 동영상 검색, 위치 기반 검색, 사기 탐지, 이상 탐지 등이 있습니다.

OpenSearch Serverless의 벡터 엔진은 OpenSearch의 k-NN(k-최근접이웃) 검색 기능으로 구동되므로 서버리스 환경의 단순함과 함께 동일한 기능을 사용할 수 있습니다. 엔진은 k-NN 플러그인 API를 지원합니다. 이러한 작업을 통해 전체 텍스트 검색, 고급 필터링, 집계, 지리공간 쿼리, 데이터 검색 속도를 높이기 위한 중첩 쿼리, 향상된 검색 결과를 활용할 수 있습니다.

벡터 엔진은 유클리드 거리, 코사인 유사성, 점 곱 유사성과 같은 거리 측정법을 제공하며 16,000개의 차원을 수용할 수 있습니다. 숫자, 부울, 날짜, 키워드, 지오포인트 등 다양한 메타데이터 유형의 필드를 메타데이터에 저장할 수 있습니다. 설명 정보를 위한 텍스트와 함께 필드를 저장하여 저장된 벡터에 더 많은 컨텍스트를 추가할 수도 있습니다. 데이터 유형을 콜로케이션하면 복잡성이 줄어들고 유지 관리성이 향상되며 데이터 중복, 버전 호환성 문제 및 라이선스 문제를 피할 수 있습니다.

> **Note**
>
> Amazon OpenSearch Serverless는 32비트 부동 벡터와 16비트 벡터 간의 변환을 수행하는 데 사용할 수 있는 Faiss 16비트 스칼라 양자화를 지원합니다. 자세한 내용은 Faiss 16비트 스칼라 양자화를 참조하세요. 이진 벡터를 사용하여 메모리 비용을 줄일 수도 있습니다. 자세한 내용은 이진 벡터를 참조하세요.

벡터 검색 컬렉션 시작

이 자습서에서는 벡터 임베딩을 실시간으로 저장, 검색 및 불러오는 다음 단계를 완료합니다.

1. 권한 구성
2. 컬렉션 생성
3. 데이터 업로드 및 검색
4. 컬렉션 삭제

1단계: 권한 구성

이 자습서를 완료하려면(일반적으로 OpenSearch Serverless를 사용하려면) 올바른 AWS Identity and Access Management (IAM) 권한이 있어야 합니다. 이 자습서에서는 컬렉션을 생성하고 데이터를 업로드하고 검색한 다음 컬렉션을 삭제합니다.

사용자 또는 역할에는 다음과 같은 최소 권한이 포함된 연결된 자격 증명 기반 정책이 있어야 합니다.

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": "*"
    }
  ]
}
```

OpenSearch Serverless IAM 권한에 대한 자세한 내용은 the section called “ID 및 액세스 관리” 섹션을 참조하세요.

2단계: 컬렉션 생성

컬렉션은 특정 워크로드 또는 사용 사례를 지원하기 위해 함께 작동하는 OpenSearch 인덱스 그룹입니다.

OpenSearch Serverless 컬렉션 생성하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Collections(컬렉션)를 선택하고 Create collection(컬렉션 생성)을 선택합니다.
3. 컬렉션 하우징의 이름을 지정하세요.
4. 컬렉션 유형에서 벡터 검색을 선택합니다. 자세한 내용은 the section called “컬렉션 유형 선택” 단원을 참조하십시오.
5. 배포 유형에서 이중화 활성화(활성 복제본) 선택을 취소합니다. 이렇게 하면 개발 또는 테스트 모드에서 컬렉션이 생성되고 컬렉션의 OpenSearch 컴퓨팅 유닛(OCU) 수가 2개로 줄어듭니다. 이 자습서에서 프로덕션 환경을 생성하려면 이 확인란을 선택된 상태로 둡니다.
6. 보안에서 간편 생성을 선택하여 보안 구성을 간소화합니다. 벡터 엔진의 모든 데이터는 기본적으로 전송 및 저장 중에 암호화됩니다. 벡터 엔진은 세분화된 IAM 권한을 지원하므로 암호화, 네트워크, 컬렉션 및 인덱스를 생성, 업데이트 및 삭제할 수 있는 사용자를 정의할 수 있습니다.
7. 다음을 선택합니다.
8. 컬렉션 설정을 검토하고 Submit(제출)을 선택합니다. 컬렉션이 `Active` 상태가 될 때까지 몇 분 정도 기다립니다.

3단계: 데이터 업로드 및 검색

인덱스는 벡터 임베딩 및 기타 필드를 저장, 검색 및 불러올 수 있는 방법을 제공하는 공통 데이터 스키마를 포함하는 문서 컬렉션입니다. Postman 또는 awscurl과 같은 HTTP 도구를 사용하거나 OpenSearch 대시보드의 Dev Tools 콘솔을 사용하여 OpenSearch Serverless 컬렉션에서 데이터를 생성하고 인덱스로 업로드할 수 있습니다. 이 자습서에서는 Dev Tools를 사용합니다.

movies(영화) 컬렉션에서 데이터를 인덱싱하고 검색하기

1. 새 컬렉션에 대해 단일 인덱스를 생성하려면 Dev Tools 콘솔에서 다음 요청을 전송합니다. 기본적으로 이렇게 하면 `nmslib` 엔진과 유클리드 거리가 포함된 인덱스가 생성됩니다.

```
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. 단일 문서를 housing-index로 인덱싱하려면 다음 요청을 보냅니다.

```
POST housing-index/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. 인덱스에 있는 속성과 유사한 속성을 검색하려면 다음 쿼리를 보내세요.

```
GET housing-index/_search
{
    "size": 5,
    "query": {
```

```
        "knn": {
            "housing-vector": {
                "vector": [
                    10,
                    20,
                    30
                ],
                "k": 5
            }
        }
    }
}
```

4단계: 컬렉션 삭제

주택 컬렉션은 테스트용이므로 실험을 마치면 삭제해야 합니다.

OpenSearch Serverless 컬렉션 삭제하기

1. Amazon OpenSearch Service 콘솔로 돌아갑니다.
2. 왼쪽 탐색 창에서 컬렉션를 선택하고 속성 컬렉션을 선택합니다.
3. 삭제를 선택하여 삭제를 확인합니다.

필터링된 검색

필터를 사용하여 의미 체계 검색 결과를 구체화할 수 있습니다. 인덱스를 만들고 문서에서 필터링된 검색을 수행하려면 이전 자습서의 데이터 업로드 및 검색을 다음 지침으로 대체하세요. 다른 단계는 동일하게 유지됩니다. 필터에 대한 자세한 내용은 필터를 사용한 k-NN 검색을 참조하세요.

movies(영화) 컬렉션에서 데이터를 인덱싱하고 검색하기

1. 컬렉션에 대해 단일 인덱스를 생성하려면 Dev Tools 콘솔에서 다음 요청을 전송합니다.

```
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. 단일 문서를 housing-index-filtered로 인덱싱하려면 다음 요청을 보냅니다.

```
POST housing-index-filtered/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. 지정된 가격으로 지리적 지점으로부터 일정 거리 내에 있는 시애틀 아파트 데이터를 검색하려면 다음 요청을 보내세요.

```
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
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}
```

십억 개 규모의 워크로드

벡터 검색 컬렉션은 수십억 개 벡터로 구성된 워크로드를 지원합니다. Auto Scaling이 자동으로 작업을 수행하므로 크기 조정 목적으로 다시 인덱싱할 필요가 없습니다. 차원 수가 많은 벡터가 수백만 개(또는 그 이상) 있고 200개 이상의 OCU가 필요한 경우 AWS Support에 문의하여 계정의 최대 OpenSearch 컴퓨팅 유닛(OCU) 수를 늘리십시오.

제한 사항

벡터 검색 컬렉션에는 다음과 같은 제한 사항이 있습니다.

- 벡터 검색 컬렉션은 Apache Lucene ANN 엔진을 지원하지 않습니다.
- 벡터 검색 컬렉션은 Faiss의 HNSW 알고리즘만 지원하고 IVF 및 IVFQ는 지원하지 않습니다.
- 벡터 검색 컬렉션은 워밍업, 통계 및 모델 학습 API 작업을 지원하지 않습니다.
- 벡터 검색 컬렉션은 인라인 또는 저장된 스크립트를 지원하지 않습니다.
- 벡터 검색 컬렉션에 AWS Management Console 대한 인덱스 수 정보는에서 사용할 수 없습니다.
- 벡터 검색 컬렉션의 인덱스 새로 고침 간격은 60초입니다.

다음 단계

벡터 검색 컬렉션과 인덱스 데이터를 생성하는 방법을 알았으므로 다음 연습을 시도해볼 수 있습니다.

- OpenSearch Python 클라이언트를 사용하여 벡터 검색 컬렉션을 사용하세요. GitHub에서 이 튜토리얼을 참조하세요.
- OpenSearch Java 클라이언트를 사용하여 벡터 검색 컬렉션을 사용하세요. GitHub에서 이 튜토리얼을 참조하세요.
- OpenSearch를 벡터 저장소로 사용하도록 LangChain을 설정하세요. LangChain은 언어 모델을 기반으로 애플리케이션을 개발할 수 있는 오픈 소스 프레임워크입니다. 자세한 내용은 LangChain 설명서를 참조하세요.

Amazon OpenSearch Serverless를 통한 데이터 수명 주기 정책 사용

Amazon OpenSearch Serverless의 데이터 수명 주기 정책은 OpenSearch Serverless가 시계열 컬렉션에 데이터를 보존하는 기간을 정의합니다. 예를 들어 OpenSearch Serverless가 로그 데이터를 삭제하기 전에 30일 동안 로그 데이터를 보존하도록 정책을 설정할 수 있습니다.

의 각 시계열 컬렉션 내에서 각 인덱스에 대해 별도의 정책을 구성할 수 있습니다 AWS 계정. OpenSearch Serverless는 최소한 정책에서 지정한 기간 동안 문서를 보관합니다. 그런 다음 일반적으로 보존 기간의 48시간 또는 10% 중 더 긴 시간 내에 최선을 다해 문서를 자동으로 삭제합니다.

시계열 컬렉션만 데이터 수명 주기 정책을 지원합니다. 검색 및 벡터 검색 컬렉션은 그렇지 않습니다.

주제

- 데이터 수명 주기 정책
- 필수 권한
- 정책 우선순위
- 정책 구문
- 데이터 수명 주기 정책 생성
- 데이터 수명 주기 정책 업데이트
- 데이터 수명 주기 정책 삭제

데이터 수명 주기 정책

데이터 수명 주기 정책에서는 일련의 규칙을 지정합니다. 데이터 수명 주기 정책을 사용하면 이러한 규칙과 일치하는 인덱스 또는 컬렉션과 관련된 데이터의 보존 기간을 관리할 수 있습니다. 이러한 규칙은 인덱스 또는 인덱스 그룹에 있는 데이터의 보존 기간을 정의합니다. 각 규칙은 리소스 유형(`index`), 보존 기간, 보존 기간이 적용되는 리소스 목록(인덱스)으로 구성됩니다.

다음 형식 중 하나를 사용하여 보존 기간을 정의합니다.

- `"MinIndexRetention": "24h"`— OpenSearch Serverless는 지정된 기간 동안 인덱스 데이터를 시간 또는 일 단위로 보존합니다. 이 기간을 `24h`부터 `3650d`까지 설정할 수 있습니다.
- `"NoMinIndexRetention": true`— OpenSearch Serverless는 인덱스 데이터를 무기한 보존합니다.

다음 샘플 정책에서 첫 번째 규칙은 컬렉션 `marketing` 내 모든 인덱스의 보존 기간을 15일로 지정합니다. 두 번째 규칙은 `finance` 컬렉션에서 `log`로 시작하는 모든 인덱스 이름에 보존 기간을 설정하지 않고 무기한 보존하도록 지정합니다.

```
{
   "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
   }
}
```

다음 샘플 정책 규칙에서 OpenSearch Serverless는 계정 내 모든 컬렉션에 대해 모든 인덱스의 데이터를 무기한 보존합니다.

```
{
   "Rules": [
      {
         "ResourceType": "index",
         "Resource": [
            "index/*/*"
         ]
      }
   ],
   "NoMinIndexRetention": true
}
```

필수 권한

OpenSearch Serverless의 수명 주기 정책은 다음 AWS Identity and Access Management (IAM) 권한을 사용합니다. IAM 조건을 지정하여 사용자를 특정 컬렉션 및 인덱스와 연결된 데이터 수명 주기 정책으로 제한할 수 있습니다.

- `aoss:CreateLifecyclePolicy` - 데이터 수명 주기 정책 생성.
- `aoss:ListLifecyclePolicies` – 현재 계정의 모든 데이터 수명 주기 정책을 나열합니다.
- `aoss:BatchGetLifecyclePolicy`— 계정 또는 정책 이름과 관련된 데이터 수명 주기 정책을 확인합니다.
- `aoss:BatchGetEffectiveLifecyclePolicy`— 주어진 리소스(`index`는 지원되는 유일한 리소스임)에 대한 데이터 수명 주기 정책을 확인합니다.
- `aoss:UpdateLifecyclePolicy`— 주어진 데이터 수명 주기 정책을 수정하고 해당 보존 설정 또는 리소스를 변경합니다.
- `aoss:DeleteLifecyclePolicy` - 데이터 수명 주기 정책 삭제.

다음 자격 증명 기반 액세스 정책을 통해 사용자는 모든 데이터 수명 주기 정책을 보고 리소스 패턴 `collection/application-logs`로 정책을 업데이트할 수 있습니다.

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": "*"
        }
    ]
}
```

정책 우선순위

데이터 수명 주기 정책 규칙이 정책 내에서 또는 정책 간에 중복되는 상황이 있을 수 있습니다. 이 경우 인덱스에 대해 보다 구체적인 리소스 이름이나 패턴을 사용하는 규칙이 두 규칙에 모두 공통되는 모든 인덱스에 대해 보다 일반적인 리소스 이름 또는 패턴으로 규칙을 재정의합니다.

예를 들어, 다음 정책에서는 인덱스 `index/sales/logstash`에 두 가지 규칙이 적용됩니다. 이 경우 `index/sales/log*`이 `index/sales/logstash`와 가장 길게 일치하므로 두 번째 규칙이 우선시됩니다. 따라서 OpenSearch Serverless는 인덱스에 보존 기간을 설정하지 않습니다.

```
{
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/sales/*",
            ],
            "MinIndexRetention": "15d"
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/sales/log*",
            ],
            "NoMinIndexRetention": true
         }
      ]
   }
```

정책 구문

하나 이상의 규칙을 입력합니다. 이러한 규칙은 OpenSearch Serverless 인덱스의 데이터 수명 주기 설정을 정의합니다.

각 규칙에는 다음 요소가 포함됩니다. 각 규칙에 `MinIndexRetention` 또는 `NoMinIndexRetention`을 제공할 수 있지만 둘 다 제공할 수는 없습니다.

Element	설명
리소스 유형	규칙이 적용되는 리소스 유형입니다. 데이터 수명 주기 정책에 지원되는 유일한 옵션은 `index`입니다.
리소스	리소스 이름 및/또는 패턴 목록. 패턴은 접두사와 와일드카드(*)로 구성되며, 이를 통해 연결된 권한을 여러 리소스에 적용할 수 있습니다. 예를 들어 `index/<collection-name|pattern> /<index-name|pattern>` 입니다.
MinIndexRetention	문서를 인덱스에 보존하는 최소 기간은 d일 또는 h시간입니다. 하한은 `24h`이고 상한은 `3650d`입니다.
NoMinIndexRetention	`true`인 경우 OpenSearch Serverless는 문서를 무기한 보존합니다.

다음 예제에서 첫 번째 규칙은 `autoparts-inventory` 패턴(`index/autoparts-inventory/*`) 아래의 모든 인덱스에 적용되며 삭제 또는 아카이빙과 같은 작업이 발생하기 전에 최소 20일 동안 데이터를 보존해야 합니다.

두 번째 규칙은 `auto*/gear` 패턴(`index/auto*/gear`)과 일치하는 인덱스를 대상으로 하여 최소 보존 기간을 24시간으로 설정합니다.

세 번째 규칙은 특히 `tires` 인덱스에 적용되며 최소 보존 기간이 없습니다. 즉,이 인덱스의 데이터를 즉시 삭제하거나 보관하거나 다른 기준에 따라 보관할 수 있습니다. 이러한 규칙은 보존 기간이 다르거나 보존 제한이 없는 인덱스 데이터의 보존을 관리하는 데 도움이 됩니다.

```
{
  "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
    }
  ]
}
```

데이터 수명 주기 정책 생성

데이터 수명 주기 정책을 생성하려면 지정된 기준에 따라 데이터의 보존 및 삭제를 관리하는 규칙을 정의합니다.

콘솔

데이터 수명 주기 정책을 생성하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 데이터 수명 주기 정책을 선택합니다.
3. 데이터 수명 주기 정책 생성을 선택합니다.
4. 정책을 설명하는 이름을 입력합니다.
5. 데이터 수명 주기에서 추가를 선택하고 정책의 컬렉션 및 인덱스를 선택합니다.

먼저 인덱스가 속한 컬렉션을 선택합니다. 그런 다음 목록에서 인덱스를 선택하거나 인덱스 패턴을 입력합니다. 모든 컬렉션을 소스로 선택하려면 별표()를 입력합니다*.

6. 데이터 보존의 경우 데이터를 무기한 보존하도록 선택하거나 무제한(삭제하지 않음)을 선택 취소하고 OpenSearch Serverless가 Amazon S3에서 데이터를 자동으로 삭제하는 기간을 지정할 수 있습니다.
7. 저장을 선택한 다음 생성을 선택합니다.

AWS CLI

를 사용하여 데이터 수명 주기 정책을 생성하려면 다음 옵션과 함께 create-lifecycle-policy 명령을 AWS CLI사용합니다.

- `--name` - 정책의 이름입니다.
- `--type` - 정책의 유형입니다. 현재 사용 가능한 유일한 값은 입니다`retention`.
- `--policy` - 데이터 수명 주기 정책입니다. 이 파라미터는 인라인 정책과 .json 파일을 모두 허용합니다. 인라인 정책을 JSON 이스케이프 문자열로 인코딩해야 합니다. 파일에 정책을 제공하려면 형식을 사용합니다`--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}]}"
```

데이터 수명 주기 정책 업데이트

데이터 수명 주기 정책을 업데이트하려면 데이터 보존 또는 삭제 요구 사항의 변경 사항을 반영하도록 기존 규칙을 수정할 수 있습니다. 이를 통해 데이터 관리 요구 사항이 발전함에 따라 정책을 조정할 수 있습니다.

정책을 업데이트하는 시점과 OpenSearch Serverless가 새 보존 기간을 적용하기 시작하는 시점 사이에 몇 분의 지연 시간이 있을 수 있습니다.

콘솔

데이터 수명 주기 정책을 업데이트하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 데이터 수명 주기 정책을 선택합니다.
3. 업데이트할 데이터 수명 주기 정책을 선택한 다음 편집을 선택합니다.
4. 시각적 편집기 또는 JSON 편집기를 사용하여 정책을 수정합니다.
5. 저장을 선택합니다.

AWS CLI

를 사용하여 데이터 수명 주기 정책을 업데이트하려면 update-lifecycle-policy 명령을 AWS CLI사용합니다.

요청에 `--policy-version` 파라미터를 포함해야 합니다. list-lifecycle-policies 또는 batch-get-lifecycle-policy 명령을 사용하여 정책 버전을 검색할 수 있습니다. 다른 사용자가 변경한 내용을 실수로 덮어쓰지 않도록 최신 정책 버전을 포함하는 것이 좋습니다.

다음 요청은 데이터 수명 주기 정책을 새 정책 JSON 문서로 업데이트합니다.

Example

```
aws opensearchserverless update-lifecycle-policy \
  --name my-policy \
  --type retention \
  --policy-version MTY2MzY5MTY1MDA3Ml8x \
  --policy file://my-new-policy.json
```

데이터 수명 주기 정책 삭제

데이터 수명 주기 정책을 삭제하면 OpenSearch Serverless는 더 이상 일치하는 인덱스에 해당 정책을 적용하지 않습니다.

콘솔

데이터 수명 주기 정책을 삭제하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.

2. 왼쪽 탐색 창에서 데이터 수명 주기 정책을 선택합니다.
3. 삭제할 정책을 선택한 다음 삭제를 선택하고 삭제를 확인합니다.

AWS CLI

를 사용하여 데이터 수명 주기 정책을 삭제하려면 delete-lifecycle-policy 명령을 AWS CLI사용합니다.

Example

```
aws opensearchserverless delete-lifecycle-policy \
  --name my-policy \
  --type retention
```

AWS SDKs를 사용하여 Amazon OpenSearch Serverless와 상호 작용

이 섹션에는 AWS SDKs를 사용하여 Amazon OpenSearch Serverless와 상호 작용하는 방법의 예가 포함되어 있습니다. 이 코드 샘플은 보안 정책 및 컬렉션을 만드는 방법과 컬렉션을 쿼리하는 방법을 보여줍니다.

> **Note**
>
> 현재 이러한 코드 샘플을 빌드하고 있습니다. 코드 샘플(Java, Go 등)을 제공하려면 GitHub 리포지토리 내에서 직접 끌어오기 요청을 여세요.

주제

- Python
- JavaScript

Python

다음 샘플 스크립트는 AWS SDK for Python (Boto3)뿐만 아니라 Python용 opensearch-py 클라이언트를 사용하여 암호화, 네트워크, 데이터 액세스 정책을 생성하고 일치하는 컬렉션을 생성하고 일부 샘플 데이터를 인덱싱합니다.

필요한 종속성을 설치하려면 다음 명령을 실행합니다.

```
pip install opensearch-py
```

```
pip install boto3
pip install botocore
pip install requests-aws4auth
```

스크립트 내에서 Principal 요소를 요청에 서명하는 사용자 또는 역할의 Amazon 리소스 이름(ARN)으로 바꿉니다. 선택적으로 region을 수정할 수도 있습니다.

```
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

다음 샘플 스크립트는 Node.js의 JavaScript용 SDK와 JavaScript용 opensearch-js 클라이언트를 사용하여 암호화, 네트워크, 데이터 액세스 정책을 생성하고, 일치하는 컬렉션을 생성하고, 인덱스를 생성하고, 일부 샘플 데이터를 인덱싱합니다.

필요한 종속성을 설치하려면 다음 명령을 실행합니다.

```
npm i aws-sdk
npm i aws4
npm i @opensearch-project/opensearch
```

스크립트 내에서 `Principal` 요소를 요청에 서명하는 사용자 또는 역할의 Amazon 리소스 이름(ARN)으로 바꿉니다. 선택적으로 `region`을 수정할 수도 있습니다.

```
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()
```

AWS CloudFormation 를 사용하여 Amazon OpenSearch Serverless 컬렉션 생성

AWS CloudFormation 를 사용하여 컬렉션, 보안 정책 및 VPC 엔드포인트와 같은 Amazon OpenSearch Serverless 리소스를 생성할 수 있습니다. 포괄적인 OpenSearch Serverless

CloudFormation 참조는 AWS CloudFormation 사용 설명서의 Amazon OpenSearch Serverless를 참조하세요.

다음 샘플 CloudFormation 템플릿은 간단한 데이터 액세스 정책, 네트워크 정책, 보안 정책뿐만 아니라 일치하는 컬렉션을 생성합니다. Amazon OpenSearch Serverless를 빠르게 시작하고 실행하고 컬렉션을 생성하고 사용하는 데 필요한 요소를 프로비저닝할 수 있는 좋은 방법입니다.

> **Important**
>
> 이 예시에서는 프로덕션 워크로드에는 권장되지 않는 퍼블릭 네트워크 액세스를 사용합니다. 컬렉션을 보호하려면 VPC 액세스를 사용하는 것이 좋습니다. 자세한 내용은 AWS::OpenSearchServerless::VPC VpcEndpoint 및 the section called “VPC 엔드포인트”를 참조하세요.

```
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
```

스냅샷을 사용하여 컬렉션 백업

스냅샷은 재해 복구 기능을 제공하는 Amazon OpenSearch Serverless 컬렉션의 point-in-time 백업입니다. OpenSearch Serverless는 자동으로 컬렉션의 스냅샷을 생성하고 관리하여 비즈니스 연속성과 데이터 보호를 보장합니다. 각 스냅샷에는 다음이 포함됩니다.

- 인덱스 메타데이터: 인덱스에 대한 설정 및 매핑
- 클러스터 메타데이터: 인덱스 템플릿 및 별칭
- 인덱스 데이터: 인덱스에 저장된 모든 문서 및 데이터

주요 이점

- 수동 구성이 필요 없는 자동 시간별 백업
- 유지 관리 오버헤드 없음

- 추가 스토리지 비용 없음
- 실수로 인한 데이터 손실로부터 빠른 복구
- 스냅샷에서 특정 인덱스를 복원할 수 있는 기능

중요 고려 사항

- 스냅샷 생성은 즉각적으로 이루어지지 않으며 완료하는 데 시간이 필요합니다.
- 스냅샷 생성 중 새 문서 또는 업데이트가 스냅샷에 포함되지 않을 수 있습니다.
- 스냅샷은 원래 컬렉션에만 복원할 수 있으며 새 컬렉션에는 복원할 수 없습니다.
- 복원되면 인덱스는 원래 버전과 다른 새 UUIDs를 수신합니다.
- 한 번에 하나의 복원 작업만 실행할 수 있습니다.
- 동일한 컬렉션에서 동시에 여러 복원 작업을 시작할 수 없습니다. 활성 복원 작업 중에 인덱스를 복원하려고 하면 작업이 실패합니다.
- 복원 작업 중에 인덱스에 대한 요청이 실패합니다.

필수 권한

스냅샷을 사용하려면 데이터 액세스 정책에서 다음 권한을 구성합니다. 데이터 액세스 정책에 대한 자세한 내용은 섹션을 참조하세요데이터 액세스 정책 대 IAM 정책.

데이터 액세스 정책	API
aoss:DescribeSnapshot	GET /_cat/스냅샷 GET /_cat/snapshots/aoss-automated/ GET _snapshot/aoss-automated/*snapshot*/
aoss:RestoreSnapshot	POST /_snapshot/aoss-automated/*snapshot*/_restore
aoss:DescribeCollectionItems	GET /_cat/복구

다음 AWS CLI 명령을 사용하여 정책을 구성할 수 있습니다.

1. create-access-policy

2. delete-access-policy
3. get-access-policy
4. update-access-policy

다음은 액세스 정책을 생성하기 위한 샘플 CLI 명령입니다.

```
aws opensearchserverless create-access-policy \
--type data \
--name AWSExample-data-access-policy \
--region us-west-2 \
--policy '[
  {
    "Rules": [
      {
        "Resource": [
          "collection/AWSExample-collection"
        ],
        "Permission": [
          "aoss:DescribeSnapshot",
          "aoss:RestoreSnapshot",
          "aoss:DescribeCollectionItems"
        ],
        "ResourceType": "collection"
      }
    ],
    "Principal": [
      "arn:aws:iam::AWSExample-account-ID:user/AWSExample-user"
    ],
    "Description": "Data policy to support snapshot operations."
  }
]'
```

스냅샷 작업

기본적으로 새 컬렉션을 생성하면 OpenSearch Serverless는 매시간 스냅샷을 자동으로 생성합니다. 사용자는 아무 작업도 수행할 필요가 없습니다. 각 스냅샷에는 컬렉션의 모든 인덱스가 포함됩니다. OpenSearch Serverless가 스냅샷을 생성한 후 다음 명령을 사용하여 스냅샷을 나열하고 스냅샷의 세부 정보를 볼 수 있습니다.

스냅샷 나열

다음 명령을 사용하여 컬렉션의 모든 스냅샷을 나열합니다.

```
GET /_cat/snapshots/aoss-automated/
```

OpenSearch Serverless는 다음과 같은 응답을 반환합니다.

```
id                                status  start_epoch start_time end_epoch  end_time
  duration    indices successful_shards failed_shards total_shards
snapshot-AWSExampleSnapshotID1     SUCCESS 1737964331  07:52:11   1737964382 07:53:02
  50.4s       1
snapshot-AWSExampleSnapshotID2     SUCCESS 1737967931  08:52:11   1737967979 08:52:59
  47.7s       2
snapshot-AWSExampleSnapshotID3     SUCCESS 1737971531  09:52:11   1737971581 09:53:01
  49.1s       3
snapshot-AWSExampleSnapshotID4 IN_PROGRESS 1737975131  10:52:11   -          -
   4.8d       3
```

스냅샷 가져오기

스냅샷에 대한 정보를 검색합니다.

```
GET _snapshot/aoss-automated/snapshot/
```

요청 예시

```
GET _snapshot/aoss-automated/snapshot-AWSExampleSnapshotID1/
```

응답의 예

```
{
    "snapshots": [
        {
            "snapshot": "snapshot-AWSExampleSnapshotID1-5e01-4423-9833Example",
            "uuid": "AWSExample-5e01-4423-9833-9e9eb757Example",
            "version_id": 136327827,
            "version": "2.11.0",
            "remote_store_index_shallow_copy": true,
            "indices": [
                "AWSExample-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
            }
        }
    ]
}
```

스냅샷 응답 필드 이해

id

스냅샷 작업의 고유 식별자입니다.

status

스냅샷 작업의 현재 상태입니다. 가능한 값은 다음과 같습니다.

- SUCCESS
- IN_PROGRESS

기간

스냅샷 작업을 완료하는 데 걸린 시간입니다.

인덱스

스냅샷에 포함된 인덱스 수입니다.

스냅샷에서 복원

스냅샷에서 복원하면 이전에 생성한 백업에서 데이터를 복구할 수 있습니다. 이 프로세스는 OpenSearch Serverless의 재해 복구 및 데이터 관리에 매우 중요합니다.

중요 고려 사항

1. 복원된 인덱스의 UUIDs는 원래 버전과 다릅니다.
2. 스냅샷은 원래 컬렉션으로만 복원할 수 있습니다. 교차 수집 복원은 지원되지 않습니다.
3. 복원 작업은 클러스터 성능에 영향을 미칠 수 있습니다. 이에 따라 적절한 계획을 수립해야 합니다.

스냅샷에서 백업된 인시던트를 복원하려면

1. 다음 명령을 실행하여 적절한 스냅샷을 식별합니다.

   ```
   GET /_snapshot/aoss-automated/_all
   ```

   더 작은 스냅샷 목록을 보려면 다음 명령을 실행합니다.

   ```
   GET /_cat/snapshots/aoss-automated/
   ```

2. 복원하기 전에 다음 명령을 실행하여 스냅샷의 세부 정보를 확인합니다.

   ```
   GET _snapshot/aoss-automated/snapshot-AWSExampleSnapshotID1/
   ```

3. 다음 명령을 실행하여 특정 스냅샷에서 복원합니다.

   ```
   POST /_snapshot/aoss-automated/snapshot-ID/_restore
   ```

   요청 본문을 포함하여 복원 작업을 사용자 지정할 수 있습니다. 다음은 그 예입니다.

   ```
   POST /_snapshot/aoss-automated/snapshot-
   AWSExampleSnapshotID1-5e01-4423-9833Example/_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. 다음 명령을 실행하여 복원 진행 상황을 확인합니다.

   ```
   GET /_cat/recovery
   ```

Note

요청 본문이 포함된 명령으로 스냅샷을 복원할 때 다음 파라미터를 사용하여 복원 동작을 제어할 수 있습니다.

인덱스

복원할 인덱스를 지정합니다. 이 파라미터는 와일드카드 패턴을 지원합니다.

ignore_unavailable

스냅샷의 인덱스가 누락된 경우에도 복원 작업을 계속할 수 있습니다.

include_global_state

클러스터 상태를 복원할지 여부를 결정합니다.

include_aliases

연결된 별칭을 복원할지 여부를 제어합니다.

rename_pattern 및 rename_replacement

복원 작업 중에 인덱스의 이름을 바꿀 수 있습니다.

Amazon OpenSearch Serverless의 용량 제한 관리

Amazon OpenSearch Serverless에서는 용량을 직접 관리할 필요가 없습니다. OpenSearch Serverless는 현재 워크로드를 기반으로 계정의 컴퓨팅 용량을 자동으로 조정합니다. 서버리스 컴퓨팅 용량은 OpenSearch 컴퓨팅 유닛(OCU)으로 측정됩니다. 각 OCU는 6GiB 메모리와 해당 가상 CPU(vCPU) 및 Amazon S3로의 데이터 전송의 조합입니다. OpenSearch Serverless의 분리된 아키텍처에 대한 자세한 내용은 the section called “작동 방법” 섹션을 참조하세요.

첫 번째 컬렉션을 생성하면 OpenSearch Serverless는 중복성 설정을 기반으로 OCUs 인스턴스화합니다. 기본적으로 중복 활성 복제본이 활성화됩니다. 즉, 다른 가용 영역의 대기 노드에서 고가용성을 보장하기 위해 총 4개의 OCUs(인덱싱용 2개, 검색용 2개)가 인스턴스화됩니다. 개발 및 테스트를 위해 컬렉션에 대한 중복성 활성화 설정을 비활성화하여 대기 복제본을 제거하고 두 개의 OCUs(인덱싱용 OCU와 검색용 OCU)만 인스턴스화할 수 있습니다. 이러한 OCU는 인덱싱이나 검색 활동이 없는 경우에도 항상 존재합니다. 모든 후속 컬렉션은 이러한 OCUs 공유할 수 있습니다(자체 OCUs 세트를 인스턴스화하는 고유 AWS KMS 키가 있는 컬렉션은 제외). 필요한 경우 인덱싱 및 검색 사용량이 증가함에 따라 OpenSearch Serverless가 자동으로 OCU를 스케일 아웃하고 추가합니다. 컬렉션 엔드포인트의

트래픽이 감소하면 용량이 다시 데이터 크기에 필요한 최소 OCU 수로 다시 스케일 다운됩니다. 검색 및 시계열 컬렉션의 경우 유휴 시 필요한 OCUs 수는 데이터 크기 및 인덱스 수에 비례합니다. 벡터의 경우, 벡터 그래프를 저장하는 메모리(RAM)와 인덱스를 저장하는 디스크 공간 모두에 의존합니다. 유휴 상태가 아닌 경우 OCU 요구 사항은이 두 가지를 모두 고려합니다.

벡터 컬렉션은 인덱스 데이터를 OCU 로컬 스토리지에 보관합니다. OCU RAM 제한에 OCU 디스크 제한보다 빠르게 도달하여 RAM 공간에 의해 벡터 컬렉션이 제한됩니다. 중복성이 활성화된 경우 OCU 용량은 인덱싱의 경우 최소 1 OCU[0.5 OCU x 2], 검색의 경우 1 OCU[0.5 OCU x 2]로 스케일 다운됩니다. 중복을 비활성화하면 도메인이 인덱싱을 위해 0.5 OCU로 스케일 다운되고 검색을 위해 0.5 OCU로 스케일 다운될 수 있습니다. 또한 크기 조정은 컬렉션 또는 인덱스에 필요한 샤드 수를 고려합니다. 각 OCU는 지정된 수의 샤드를 지원할 수 있습니다. 인덱스 수는 샤드 수에 비례해야 합니다. 필요한 기본 OCUs의 총 수는 필요한 데이터, 메모리 및 샤드의 최대량입니다. 자세한 내용은 AWS 빅 데이터 블로그의 모든 규모에서 Amazon OpenSearch Serverless 비용 효율적인 검색 기능을 참조하세요.

검색 및 벡터 검색 컬렉션의 경우 빠른 쿼리 응답 시간을 보장하기 위해 모든 데이터가 핫 인덱스에 저장됩니다. 시계열 컬렉션은 핫 스토리지와 웜 스토리지의 조합을 사용하며, 최근 데이터는 핫 스토리지에 보관되어 더 자주 액세스하는 데이터에 대한 쿼리 응답 시간을 최적화합니다. 자세한 내용은 the section called “컬렉션 유형 선택” 단원을 참조하십시오.

> **Note**
>
> 벡터 검색 컬렉션이 검색 또는 시계열 컬렉션과 동일한 KMS 키를 사용하더라도 벡터 검색 컬렉션은 OCUs를 검색 및 시계열 컬렉션과 공유할 수 없습니다. 첫 번째 벡터 컬렉션에 대해 새 OCU 세트가 생성됩니다. 벡터 컬렉션의 OCU는 동일한 KMS 키 컬렉션 사이에서 공유됩니다.

컬렉션 용량을 관리하고 비용을 제어하려면 현재 계정 및 리전에 대한 전체 최대 인덱싱 및 검색 용량을 지정할 수 있으며 OpenSearch Serverless는 이러한 사양에 따라 컬렉션 리소스를 자동으로 스케일 아웃합니다.

인덱싱 및 검색 용량은 개별적으로 확장되므로 각각에 대해 계정 수준 제한을 지정합니다.

- 최대 인덱싱 용량 – OpenSearch Serverless는 인덱싱 용량을 이 OCU 수까지 늘릴 수 있습니다.
- 최대 검색 용량 – OpenSearch Serverless는 검색 용량을 이 OCU 수까지 늘릴 수 있습니다.

> **Note**
>
> 현재, 용량 설정은 계정 수준에만 적용됩니다. 컬렉션당 용량 제한은 구성할 수 없습니다.

목표는 최대 용량이 워크로드 급증을 처리할 수 있을 만큼 충분히 높은 용량이 되도록 하는 것입니다. 설정에 따라 OpenSearch Serverless는 인덱싱 및 검색 워크로드를 처리하기 위해 컬렉션의 OCU 수를 자동으로 스케일 아웃합니다.

주제

- 용량 설정 구성
- 최대 용량 제한
- 용량 사용량 모니터링

용량 설정 구성

OpenSearch Serverless 콘솔에서 용량 설정을 구성하려면 왼쪽 탐색 창에서 Serverless(서버리스)를 확장하고 Dashboard(대시보드)를 선택합니다. Capacity management(용량 관리)에서 최대 인덱싱 및 검색 용량을 지정합니다.

를 사용하여 용량을 구성하려면 UpdateAccountSettings 요청을 AWS CLI보냅니다.

```
aws opensearchserverless update-account-settings \
    --capacity-limits '{ "maxIndexingCapacityInOCU": 8,"maxSearchCapacityInOCU": 9 }'
```

최대 용량 제한

컬렉션에 포함될 수 있는 최대 인덱스 수는 1000개입니다. 세 가지 유형의 컬렉션 모두에서 기본 최대 OCU 용량은 인덱싱용 OCUs 10개와 검색용 OCUs. 계정에 허용되는 최소 OCU 용량은 인덱싱의 경우 1 OCU[0.5 OCU x 2], 검색의 경우 1 OCU[0.5 OCU x 2]입니다. 모든 컬렉션에서 허용되는 최대 용량은 인덱싱에 1,700 OCUs, 검색에 1,700 OCUs. OCU 수를 1개에서 최대 허용 용량(2의 배수)까지 원하는 수로 구성할 수 있습니다.

각 OCU에는 120GiB의 인덱스 데이터를 위한 충분한 핫 임시 스토리지가 포함되어 있습니다. OpenSearch Serverless는 검색 및 벡터 검색 컬렉션에서 인덱스당 최대 1TiB의 데이터와 시계열 컬렉션에서 인덱스당 100TiB의 핫 데이터를 지원합니다. 시계열 컬렉션의 경우 이보다 더 많은 데이터를 수집하여 S3에 웜 데이터로 저장할 수 있습니다.

모든 할당량 목록은 OpenSearch Serverless 할당량을 참조하세요.

용량 사용량 모니터링

`SearchOCU` 및 `IndexingOCU` 계정 수준 CloudWatch 지표를 모니터링하여 컬렉션이 어떻게 확장되고 있는지 이해할 수 있습니다. 계정이 용량과 관련된 지표의 임계값에 근접하면 알림을 받도록 경보를 구성하여 그에 따라 용량 설정을 조정하는 것이 좋습니다.

또한 이러한 지표를 사용하여 최대 용량 설정이 적절한지 아니면 조정이 필요한지 확인할 수 있습니다. 이러한 지표 분석을 통해 컬렉션의 효율성을 최적화하는 데 집중할 수 있을 것입니다. OpenSearch Serverless에서 CloudWatch로 보내는 지표에 대한 자세한 내용은 the section called “OpenSearch Serverless 모니터링” 섹션을 참조하세요.

Amazon OpenSearch Serverless 컬렉션으로 데이터 수집

이 섹션에서는 Amazon OpenSearch Serverless 컬렉션으로의 데이터 수집을 위해 지원되는 수집 파이프라인에 대한 세부 정보를 제공합니다. 또한 OpenSearch API 작업과 상호 작용하는 데 사용할 수 있는 일부 클라이언트도 다룹니다. OpenSearch Serverless와 통합하려면 클라이언트가 OpenSearch 2.x 버전과 호환되어야 합니다.

주제

- 최소 필수 권한
- OpenSearch Ingestion
- Fluent Bit
- Amazon Data Firehose
- Go
- Java
- JavaScript
- Logstash
- Python
- Ruby
- 다른 클라이언트로 HTTP 요청 서명

최소 필수 권한

데이터를 OpenSearch Serverless 컬렉션으로 수집하려면 데이터를 쓰는 보안 주체가 데이터 액세스 정책에 할당된 다음과 같은 최소 권한이 있어야 합니다.

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/target-collection/logs"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:WriteDocument",
               "aoss:UpdateIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

추가 인덱스에 쓰려는 경우 권한이 더 광범위할 수 있습니다. 예를 들어 단일 대상 인덱스를 지정하는 대신 모든 인덱스(index/*target-collection*/*) 또는 인덱스 하위 집합(index/*target-collection*/*logs**)에 대한 권한을 허용할 수 있습니다.

사용 가능한 모든 OpenSearch API 작업 및 관련 권한에 대한 참조는 the section called “지원되는 작업 및 플러그인”을(를) 참조하세요.

OpenSearch Ingestion

타사 클라이언트를 사용하여 OpenSearch Ingestion 컬렉션으로 직접 데이터를 보내는 대신 Amazon OpenSearch Ingestion을 사용할 수 있습니다. 데이터 생산자를 구성하면 데이터를 OpenSearch Ingestion으로 전송하고, 이를 통해 사용자가 지정한 컬렉션에 데이터를 자동으로 전송합니다. 전송 전에 데이터를 변환하도록 OpenSearch Ingestion을 구성할 수도 있습니다. 자세한 내용은 Amazon OpenSearch Ingestion 단원을 참조하십시오.

OpenSearch Ingestion 파이프라인에는 싱크로 구성된 OpenSearch Serverless 컬렉션에 쓸 수 있는 권한이 필요합니다. 이러한 권한에는 컬렉션을 설명하고 컬렉션에 HTTP 요청을 보내는 기능이 포함됩니다. OpenSearch Ingestion을 사용하여 컬렉션에 데이터를 추가하는 방법에 대한 지침은 the section called “컬렉션에 대한 액세스 권한을 파이프라인에 부여” 섹션을 참조하세요.

OpenSearch Ingestion을 시작하려면 the section called “튜토리얼: 컬렉션에 데이터 수집”(을)를 참조하세요.

Fluent Bit

AWS for Fluent Bit 이미지와 OpenSearch 출력 플러그인을 사용하여 데이터를 OpenSearch Serverless 컬렉션으로 수집할 수 있습니다.

> **Note**
>
> OpenSearch Serverless와 통합하려면 Fluent Bit AWS 용 이미지 버전 2.30.0 이상이 있어야 합니다.

구성의 예제:

구성 파일의 이 샘플 출력 섹션에서는 OpenSearch Serverless 컬렉션을 대상으로 사용하는 방법을 보여줍니다. 중요한 추가 사항은 `AWS_Service_Name` 파라미터인 `aoss`입니다. `Host`는 컬렉션 엔드포인트입니다.

```
[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

Firehose는 전송 대상으로 OpenSearch Serverless를 지원합니다. OpenSearch Serverless로 데이터를 전송하는 지침은 Amazon Data Firehose 개발자 안내서에서 Kinesis Data Firehose 전송 스트림 생성 및 대상에 대한 OpenSearch Serverless 선택을 참조하세요.

전송을 위해 Firehose에 제공하는 IAM 역할은 대상 컬렉션에 대한 `aoss:WriteDocument` 최소 권한이 있는 데이터 액세스 정책 내에서 지정되어야 하며 데이터를 전송할 기존 인덱스가 있어야 합니다. 자세한 내용은 the section called “최소 필수 권한” 단원을 참조하십시오.

OpenSearch Serverless로 데이터를 보내기 전에 데이터에 대한 변환을 수행해야 할 수 있습니다. Lambda 함수로 이 태스크를 수행하는 방법에 대한 자세한 내용은 동일한 안내서의 Amazon Kinesis Data Firehose Data 데이터 변환을 참조하세요.

Go

다음 샘플 코드는 Go용 opensearch-go 클라이언트를 사용하여 지정된 OpenSearch Serverless 컬렉션에 대한 보안 연결을 설정하고 단일 인덱스를 생성합니다. `region` 및 `host`의 값을 입력해야 합니다.

```
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

다음 샘플 코드는 Java용 opensearch-java 클라이언트를 사용하여 지정된 OpenSearch Serverless 컬렉션에 대한 보안 연결을 설정하고 단일 인덱스를 생성합니다. `region` 및 `host`의 값을 입력해야 합니다.

OpenSearch Service 도메인과 비교했을 때 중요한 차이점은 서비스 이름(`aoss` 대신 `es`)입니다.

```
// 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();
```

다음 샘플 코드는 다시 보안 연결을 설정한 다음, 인덱스를 검색합니다.

```
import org.opensearch.client.opensearch.OpenSearchClient;
>>>>>>> aoss-slr-update

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

다음 샘플 코드는 JavaScript용 opensearch-js 클라이언트를 사용하여 지정된 OpenSearch Serverless 컬렉션에 대한 보안 연결을 설정하고, 단일 인덱스를 생성하고, 문서를 추가하고, 인덱스를 삭제합니다. `node` 및 `region`의 값을 입력해야 합니다.

OpenSearch Service 도메인과 비교했을 때 중요한 차이점은 서비스 이름(`aoss` 대신 `es`)입니다.

Version 3

이 예시는 Node.js의 JavaScript용 SDK 버전 3을 사용합니다.

```
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

이 예시는 Node.js의 JavaScript용 SDK 버전 2를 사용합니다.

```
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

Logstash OpenSearch 플러그인을 사용하여 OpenSearch Serverless 컬렉션에 로그를 게시할 수 있습니다.

Logstash를 사용하여 OpenSearch Serverless로 데이터 보내기

1. Docker 또는 Linux를 사용하여 logstash-output-opensearch 플러그인 2.0.0 또는 이후 버전을 설치합니다.

   Docker

   Docker는 OpenSearch 출력 플러그인 opensearchproject/logstash-oss-with-opensearch-output-plugin이 사전 설치된 Logstash OSS 소프트웨어를 호스팅합니다. 다른 이미지와 마찬가지로 이미지를 가져올 수 있습니다.

```
docker pull opensearchproject/logstash-oss-with-opensearch-output-plugin:latest
```

Linux

아직 설치하지 않았다면 먼저 최신 버전의 Logstash를 설치합니다. 그런 다음 출력 플러그인 버전 2.0.0을 설치합니다.

```
cd logstash-8.5.0/
bin/logstash-plugin install --version 2.0.0 logstash-output-opensearch
```

플러그인이 이미 설치되어 있는 경우 최신 버전으로 업데이트합니다.

```
bin/logstash-plugin update logstash-output-opensearch
```

플러그인 버전 2.0.0부터 AWS SDK는 버전 3을 사용합니다. 8.4.0 이전 버전의 Logstash를 사용하는 경우 사전 설치된 AWS 플러그인을 제거하고 `logstash-integration-aws` 플러그인을 설치해야 합니다.

```
/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. OpenSearch 출력 플러그인이 OpenSearch Serverless와 함께 작동하려면 logstash.conf의 `opensearch` 출력 섹션을 다음과 같이 수정해야 합니다.

   - `aoss`를 `auth_type`의 `service_name`으로 지정합니다.
   - `hosts`에 대한 컬렉션 엔드포인트를 지정합니다.
   - 파라미터 `default_server_major_version` 및 `legacy_template`을 추가합니다. 이러한 파라미터는 플러그인이 OpenSearch Serverless와 함께 작동하는 데 필요합니다.

```
output {
```

```
  opensearch {
    hosts => "collection-endpoint:443"
    auth_type => {
      ...
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

이 예시 구성 파일은 S3 버킷의 파일에서 입력값을 가져와서 OpenSearch Serverless 컬렉션으로 보냅니다.

```
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. 그런 다음 새 구성으로 Logstash를 실행하여 플러그인을 테스트합니다.

```
bin/logstash -f config/test-plugin.conf
```

Python

다음 샘플 코드는 Python용 opensearch-py 클라이언트를 사용하여 지정된 OpenSearch Serverless 컬렉션에 대한 보안 연결을 설정하고, 단일 인덱스를 생성하고, 해당 인덱스를 검색합니다. `region` 및 `host`의 값을 입력해야 합니다.

OpenSearch Service 도메인과 비교했을 때 중요한 차이점은 서비스 이름(`aoss` 대신 `es`)입니다.

```
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 gem은 즉시 OpenSearch Service와 함께 OpenSearch Serverless에 대한 액세스 권한을 제공합니다. 이 gem의 종속 항목이므로 opensearch-ruby 클라이언트의 모든 기능을 가지고 있습니다.

Sigv4 서명자를 인스턴스화할 때 aoss를 서비스 이름으로 지정합니다.

```
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)
```

다른 클라이언트로 HTTP 요청 서명

다른 클라이언트로 HTTP 요청을 구성하는 경우 OpenSearch Serverless 컬렉션에 대한 요청에 서명할 때 다음 요구 사항이 적용됩니다.

- 서비스 이름을 aoss로 지정합니다.
- x-amz-content-sha256 헤더는 모든 AWS 서명 버전 4 요청에 필요합니다. 요청 페이로드의 해시를 제공합니다. 요청 페이로드가 있는 경우 값을 보안 해시 알고리즘(SHA) 암호화 해시(SHA256)로 설정합니다. 요청 페이로드가 없는 경우 값을 빈 문자열의 해시인 e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855(으)로 설정합니다.

주제

- cURL을 사용한 인덱싱
- Postman을 사용한 인덱싱

cURL을 사용한 인덱싱

다음 예제 요청은 클라이언트 URL 요청 라이브러리(cURL)를 사용하여 컬렉션 내에서 movies-index 인덱스로 단일 문서를 전송합니다.

```
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"}'
```

Postman을 사용한 인덱싱

다음 이미지는 Postman을 사용하여 컬렉션에 요청을 전송하는 방법을 보여줍니다. 인증 지침은 Postman의 Authenticate with AWS Signature authentication workflow를 참조하세요.

Amazon OpenSearch Serverless에서 Machine Learning 구성

기계 학습

Machine Learning(ML)은 ML 알고리즘 및 원격 모델 형태로 ML 기능을 제공합니다. 이러한 모델에 액세스하면 RAG 또는 시맨틱 검색과 같은 여러 AI 워크플로를 실행할 수 있습니다. ML은 커넥터로 구성할 수 있는 최신 외부 호스팅 모델을 사용하여 생성형 AI 사용 사례의 실험 및 프로덕션 배포를 지원합니다. 커넥터를 구성한 후 모델에 구성한 다음 배포하여 예측을 수행해야 합니다.

커넥터

커넥터를 사용하면 타사 ML 플랫폼에서 호스팅되는 모델에 쉽게 액세스할 수 있습니다. OpenSearch 클러스터와 원격 모델 간의 게이트웨이 역할을 합니다. 자세한 내용은 다음 설명서를 참조하세요.

- OpenSearch 설명서 웹 사이트에서 타사 ML 플랫폼용 커넥터 생성
- 외부 플랫폼용 커넥터
- 용 커넥터 AWS 서비스

> **Important**
>
> - 신뢰 정책을 생성할 때를 OpenSearch Service 원칙**ml.opensearchservice.amazonaws.com**으로 추가합니다.
> - 정책에서 도메인을 구성하는 방법을 표시하는 커넥터 페이지의 단계를 건너뜁니다.
> - 권한 구성 단계에서 `iam:PassRole` 문을 추가합니다.
> - OpenSearch Dashboards에서 ML 역할 매핑 단계를 건너뜁니다. 백엔드 역할 구성은 필요하지 않습니다. 이는 용 커넥터 AWS 서비스와 외부 플랫폼용 커넥터에 적용됩니다.
> - 컬렉션 엔드포인트에 대한 SigV4 요청에서 서비스 이름을 **aoss** 대신 로 설정합니다**es**.

모델

모델은 다양한 AI 워크플로에서 사용되는 핵심 기능입니다. 일반적으로 커넥터를 모델과 연결하여 커넥터를 사용하여 예측을 수행합니다. 모델이 배포된 상태이면 예측을 실행할 수 있습니다. 자세한 내용은 OpenSearch 설명서 웹 사이트의 타사 플랫폼에서 호스팅되는 모델 등록을 참조하세요.

> **Note**
>
> 로컬 모델과 같은 모든 모델 기능이 OpenSearch Serverless에서 지원되는 것은 아닙니다. 자세한 내용은 지원되지 않는 Machine Learning APIs 및 기능 단원을 참조하십시오.

Machine Learning에 대한 권한 구성

다음 섹션에서는 Machine Learning(ML)에 필요한 컬렉션 데이터 액세스 정책을 설명합니다. *## ### ##* 특정 정보로 바꿉니다. 자세한 내용은 지원되는 정책 권한 단원을 참조하십시오.

```
{
    "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 – 커넥터, 모델 및 모델 그룹을 검색하고 쿼리할 수 있는 권한을 부여합니다.
- aoss:CreateMLResource – 커넥터, 모델 및 모델 그룹을 생성할 수 있는 권한을 부여합니다.
- aoss:UpdateMLResource – 커넥터, 모델 및 모델 그룹을 업데이트할 수 있는 권한을 부여합니다.
- aoss:DeleteMLResource – 커넥터, 모델 및 모델 그룹을 삭제할 수 있는 권한을 부여합니다.
- aoss:ExecuteMLResource – 모델에 대한 예측을 수행할 수 있는 권한을 부여합니다.

지원되지 않는 Machine Learning APIs 및 기능

지원되지 않는 APIs

다음 Machine Learning(ML) APIs는 Amazon OpenSearch Serverless에서 지원되지 않습니다.

- 로컬 모델 기능
- 모델 훈련 API
- 모델 예측 알고리즘 API
- 모델 배치 예측 API
- 에이전트 API 및 해당 도구
- MCP 서버 APIs
- 메모리 APIs
- 컨트롤러 APIs
- 알고리즘 API 실행
- ML 프로파일 AP
- ML 통계 API

ML APIs에 대한 자세한 내용은 OpenSearch 설명서 웹 사이트의 ML APIs를 참조하세요.

지원되지 않는 기능

Amazon OpenSearch Serverless에서는 다음 ML 기능이 지원되지 않습니다.

- 에이전트 및 도구
- 로컬 모델

- 검색 및 수집 파이프라인 내의 ML 추론 프로세서
  - ML 추론 수집 프로세서
  - ML 추론 검색 응답 프로세서
  - ML 추론 검색 요청 프로세서

이러한 기능에 대한 자세한 내용은 OpenSearch 설명서 웹 사이트에서 다음 설명서를 참조하세요.

- 기계 학습
- ML 추론 프로세서
- 파이프라인 검색

OpenSearch Serverless에서 신경 검색 및 하이브리드 검색 구성

신경망 검색

Amazon OpenSearch Serverless는 데이터에 대한 의미 체계 검색 작업을 위해 신경 검색 기능을 지원합니다. 신경 검색은 기계 학습 모델을 사용하여 쿼리의 의미와 컨텍스트를 이해하므로 기존 키워드 기반 검색보다 더 관련성이 높은 검색 결과를 제공합니다. 이 섹션에서는 필요한 권한, 지원되는 프로세서 및 표준 OpenSearch 구현과의 주요 차이점을 포함하여 OpenSearch Serverless에서 신경 검색을 구성하는 방법을 설명합니다.

신경 검색을 사용하면 데이터에 대한 의미론 검색을 수행할 수 있습니다.이 경우 의미론적 의미를 고려하여 검색 쿼리의 의도를 이해할 수 있습니다. 이 기능은 다음 구성 요소로 구동됩니다.

- 텍스트 임베딩 수집 파이프라인 프로세서
- 신경 쿼리
- 신경 스파스 쿼리

하이브리드 검색

하이브리드 검색을 사용하면 키워드와 의미 체계 검색 기능을 결합하여 검색 관련성을 개선할 수 있습니다. 하이브리드 검색을 사용하려면 검색 결과를 처리하고 문서 점수를 결합하는 검색 파이프라인을 생성합니다. 자세한 내용은 OpenSearch 설명서 웹 사이트의 검색 파이프라인을 참조하세요. 다음 구성 요소를 사용하여 하이브리드 검색을 구현합니다.

- 정규화 검색 파이프라인 프로세서

지원되는 정규화 기법

- `min_max`
- `l2`

지원되는 조합 기법

- `arithmetic_mean`
- `geometric_mean`
- `harmonic_mean`

정규화 및 조합 기법에 대한 자세한 내용은 OpenSearch 설명서 웹 사이트의 요청 본문 필드를 참조하세요.

- Hybrid 쿼리

신경 및 하이브리드 쿼리

기본적으로 OpenSearch는 키워드가 포함된 검색 쿼리에 적합한 키워드 기반 Okapi BM25 알고리즘을 사용하여 문서 점수를 계산합니다. 신경 검색은 자연어 쿼리에 대한 새로운 쿼리 유형과 의미 체계 및 키워드 검색을 결합할 수 있는 기능을 제공합니다.

Example : **`neural`**

```
"neural": {
  "vector_field": {
    "query_text": "query_text",
    "query_image": "image_binary",
    "model_id": "model_id",
    "k": 100
  }
}
```

자세한 내용은 OpenSearch 설명서 웹 사이트의 신경 쿼리를 참조하세요.

Example : **`hybrid`**

```
"hybrid": {
      "queries": [
        array of lexical, neural, or combined queries
      ]
    }
```

자세한 내용은 OpenSearch 설명서 웹 사이트의 하이브리드 쿼리를 참조하세요.

Amazon OpenSearch Serverless에서 의미 체계 검색 구성 요소를 구성하려면 OpenSearch 설명서 웹 사이트의 신경 검색 자습서의 단계를 따릅니다. 다음과 같은 중요한 차이점에 유의하세요.

- OpenSearch Serverless는 원격 모델만 지원합니다. 원격으로 호스팅되는 모델로 커넥터를 구성해야 합니다. 원격 모델을 배포하거나 제거할 필요가 없습니다. 자세한 내용은 OpenSearch 설명서 웹 사이트의 의미 체계 및 하이브리드 검색 시작하기를 참조하세요.
- 벡터 인덱스를 검색하거나 최근에 생성된 검색 및 수집 파이프라인을 검색할 때 최대 15초의 지연 시간이 예상됩니다.

권한 구성

OpenSearch Serverless의 신경 검색에는 다음 권한이 필요합니다. 자세한 내용은 지원되는 정책 권한 단원을 참조하십시오.

Example : 신경 검색 정책

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "NeuralSearch",
            "Effect": "Allow",
            "Action": [
              "aoss:CreateIndex",
              "aoss:CreateCollectionItems",
              "aoss:CreateMLResource",
              "aoss:DescribeCollectionItems",
              "aoss:UpdateCollectionItems",
              "aoss:DeleteIndex",
              "aoss:DeleteCollectionItems",
              "aoss:DeleteMLResource",
              "aoss:APIAccessAll",
            ],
            "Resource": "*"
        }
    ]
}
```

- aoss:*Index – 텍스트 임베딩이 저장되는 벡터 인덱스를 생성합니다.
- aoss:*CollectionItems – 수집 및 검색 파이프라인을 생성합니다.
- aoss:*MLResource – 텍스트 임베딩 모델을 생성하고 등록합니다.
- aoss:APIAccessAll – 검색 및 수집 작업을 위해 OpenSearch APIs에 대한 액세스를 제공합니다.

다음은 신경 검색에 필요한 컬렉션 데이터 액세스 정책을 설명합니다. ## ### ## 특정 정보로 바꿉니다.

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

Amazon OpenSearch Serverless에서 워크플로 구성

워크플로

워크플로는 OpenSearch에서 AI 애플리케이션을 혁신하는 빌더를 지원합니다. 의미 체계 검색과 같은 OpenSearch에서 기계 학습(ML) 제품을 사용하는 현재 프로세스에는 복잡한 설정 및 사전 처리 작업과 상세 사용자 쿼리가 필요하며,이 두 가지 모두 시간이 많이 걸리고 오류가 발생하기 쉽습니다. 워크플로는 OpenSearch에 대한 여러 API 호출을 연결하는 간소화 프레임워크입니다.

설정 및 사용량은 OpenSearch 웹 사이트의 구성 자동화를 참조하세요. OpenSearch Serverless에서 워크플로를 사용하는 경우 다음과 같은 중요한 차이점을 고려하세요.

- OpenSearch Serverless는 워크플로 단계에서 원격 모델만 사용합니다. 이러한 모델은 배포할 필요가 없습니다.
- OpenSearch Serverless는 워크플로 재인덱스 단계를 지원하지 않습니다.
- 다른 API 호출 후 워크플로 및 워크플로 상태를 검색하는 경우 업데이트가 나타날 때까지 최대 15초의 지연 시간이 예상됩니다.

OpenSearch Serverless 컬렉션은 OpenSearch UI 애플리케이션에서 데이터 소스로 사용되는 워크플로를 지원합니다. 자세한 내용은 데이터 소스 연결 관리를 참조하세요.

권한 구성

템플릿을 생성하고 프로비저닝하기 전에 필요한 권한이 있는지 확인합니다. 도움이 필요한 경우 계정 관리자에게 문의하세요. OpenSearch Serverless 워크플로에는 다음 권한이 필요합니다. IAM 정책에서 컬렉션 리소스 ARN을 정의하여 특정 컬렉션에 대한 권한 범위를 지정할 수 있습니다.

Example : 워크플로 정책

JSON

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
              "aoss:CreateIndex",
              "aoss:CreateCollectionItems",
              "aoss:CreateMLResource",
              "aoss:DescribeCollectionItems",
              "aoss:UpdateCollectionItems",
              "aoss:DeleteIndex",
              "aoss:DeleteCollectionItems",
              "aoss:DeleteMLResource",
              "aoss:APIAccessAll",
            ],
            "Resource": "*"
        }
    ]
}
```

- aoss:*CollectionItems – 템플릿을 생성 및 관리하고 검색 및 수집 파이프라인을 프로비저닝할 수 있는 권한을 부여합니다.
- aoss:*Index – OpenSearch API 작업을 사용하여 인덱스를 생성하고 삭제할 수 있는 권한을 부여합니다.
- aoss:*MLResource – Machine Learning 구성을 사용하는 워크플로 단계를 프로비저닝할 수 있는 권한을 부여합니다.

Amazon OpenSearch Serverless 보안 개요

Amazon OpenSearch Serverless의 보안은 다음과 같은 점에서 Amazon OpenSearch Service의 보안과 근본적으로 다릅니다.

Feature	OpenSearch Service	OpenSearch Serverless
데이터 액세스 제어	데이터 액세스는 IAM 정책 및 세분화된 액세스 제어에 의해 결정됩니다.	데이터 액세스는 데이터 액세스 정책에 따라 결정됩니다.
저장된 데이터 암호화	저장된 암호화는 도메인에 대한 선택 사항입니다.	저장된 암호화는 컬렉션에 필수입니다.
보안 설정 및 관리	각 도메인에 대해 네트워크, 암호화, 데이터 액세스를 개별적으로 구성해야 합니다.	보안 정책을 사용하여 여러 컬렉션의 보안 설정을 대규모로 관리할 수 있습니다.

다음 다이어그램은 기능 컬렉션을 구성하는 보안 구성 요소를 보여줍니다. 컬렉션에는 할당된 암호화 키, 네트워크 액세스 설정, 해당 리소스에 권한을 부여하는 일치하는 데이터 액세스 정책이 있어야 합니다.

주제

- 암호화 정책
- 네트워크 정책
- 데이터 액세스 정책
- IAM 및 SAML 인증
- 인프라 보안
- Amazon OpenSearch Serverless에서 보안 시작하기
- Amazon OpenSearch Serverless에 대한 Identity and Access Management
- Amazon OpenSearch Serverless 암호화
- Amazon OpenSearch Serverless에 대한 네트워크 액세스
- Amazon OpenSearch Serverless를 위한 데이터 액세스 제어
- 인터페이스 엔드포인트(AWS PrivateLink)를 사용하여 Amazon OpenSearch Serverless에 액세스
- Amazon OpenSearch Serverless에 대한 SAML 인증
- Amazon OpenSearch Serverless에 대한 규정 준수 확인

암호화 정책

암호화 정책은 컬렉션이 AWS 소유 키 또는 고객 관리형 키로 암호화되는지 여부를 정의합니다. 암호화 정책은 리소스 패턴과 암호화 키라는 두 가지 구성 요소로 구성됩니다. 리소스 패턴은 정책이 적용되는 컬렉션을 정의합니다. 암호화 키는 관련 컬렉션을 보호하는 방법을 결정합니다.

정책을 여러 컬렉션에 적용하려면 정책 규칙에 와일드카드(*)를 포함해야 합니다. 예를 들어 다음 정책은 이름이 “log”로 시작하는 모든 컬렉션에 적용됩니다.

암호화 정책은 특히 프로그래밍 방식으로 컬렉션을 생성하고 관리하는 프로세스를 간소화합니다. 이름을 지정하여 컬렉션을 생성할 수 있으며 생성 시 암호화 키가 자동으로 할당됩니다.

네트워크 정책

네트워크 정책은 컬렉션에 프라이빗하게 액세스할 수 있는지 아니면 퍼블릭 네트워크에서 인터넷을 통해 액세스할 수 있는지 정의합니다. 프라이빗 컬렉션은 OpenSearch Serverless 관리형 VPC 엔드포인트 또는 AWS 서비스 프라이빗 액세스를 사용하는 Amazon Bedrock과 같은 특정 AWS 서비스 를 통해 액세스할 수 있습니다. 암호화 정책과 마찬가지로 네트워크 정책도 여러 컬렉션에 적용할 수 있으므로 여러 컬렉션에 대한 네트워크 액세스를 대규모로 관리할 수 있습니다.

네트워크 정책은 액세스 유형과 리소스 유형이라는 두 가지 구성 요소로 구성됩니다. 액세스 유형은 퍼블릭 또는 프라이빗 액세스일 수 있습니다. 리소스 유형에 따라 선택한 액세스가 컬렉션 엔드포인트, OpenSearch Dashboards 엔드포인트 또는 둘 다에 적용되는지가 결정됩니다.

네트워크 정책 내에서 VPC 액세스를 구성하려는 경우 먼저 하나 이상의 OpenSearch Serverless 관리형 VPC 엔드포인트를 생성해야 합니다. 이러한 엔드포인트를 사용하면 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 OpenSearch Serverless에 액세스할 수 있습니다.

에 대한 프라이빗 액세스는 컬렉션의 OpenSearch 엔드포인트에만 적용할 AWS 서비스 수 있으며 OpenSearch Dashboards 엔드포인트에는 적용할 수 없습니다. OpenSearch Dashboards에 대한 액세스 권한은 부여할 수 AWS 서비스 없습니다.

데이터 액세스 정책

데이터 액세스 정책은 사용자가 컬렉션 내 데이터에 액세스하는 방법을 정의합니다. 데이터 액세스 정책을 사용하면 특정 패턴과 일치하는 컬렉션 및 인덱스에 액세스 권한을 자동으로 할당하여 컬렉션을 대규모로 관리하는 데 도움이 됩니다. 단일 리소스에 여러 정책을 적용할 수 있습니다.

데이터 액세스 정책은 일련의 규칙으로 구성되며, 각 규칙에는 리소스 유형, 부여된 리소스, 권한 세트의 세 가지 구성 요소가 있습니다. 리소스 유형은 컬렉션 또는 인덱스일 수 있습니다. 부여된 리소스는 컬렉션/인덱스 이름 또는 와일드카드(*)가 있는 패턴일 수 있습니다. 권한 목록은 정책에서 액세스 권한을 부여하는 OpenSearch API 작업을 지정합니다. 또한 정책에는 액세스 권한을 부여할 IAM 역할, 사용자 및 SAML ID를 지정하는 보안 주체 목록이 포함되어 있습니다.

데이터 액세스 정책의 형식에 대한 자세한 내용은 정책 구문을 참조하세요.

데이터 액세스 정책을 생성하기 전에 정책에서 액세스를 제공할 하나 이상의 IAM 역할 또는 사용자나 SAML ID가 있어야 합니다. 자세한 내용은 다음 섹션을 참조하세요.

> **Note**
>
> 컬렉션의 퍼블릭 액세스에서 프라이빗 액세스로 전환하면 OpenSearch Serverless 컬렉션 콘솔에서 인덱스 탭이 제거됩니다.

IAM 및 SAML 인증

IAM 보안 주체 및 SAML 자격 증명은 데이터 액세스 정책의 구성 요소 중 하나입니다. 액세스 정책의 `principal` 설명에 IAM 역할, 사용자 및 SAML ID를 포함할 수 있습니다. 그러면 해당 보안 주체에게 관련 정책 규칙에 지정한 권한이 부여됩니다.

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

OpenSearch Serverless 내에서 SAML 인증을 직접 구성합니다. 자세한 내용은 the section called "SAML 인증" 단원을 참조하십시오.

인프라 보안

Amazon OpenSearch Serverless는 AWS 글로벌 네트워크 보안으로 보호됩니다. 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 AWS 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon OpenSearch Serverless에 액세스합니다. 클라이언트가 TLS(전송 계층 보안)를 지원해야 합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다. TLS 1.3에 대해 지원되는 암호 목록은 Elastic Load Balancing 설명서의 TLS protocols and ciphers를 참조하세요.

또한 요청은 액세스 키 ID 및 IAM 위탁자와 관련된 시크릿 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 자격 증명을 생성하여 요청에 서명할 수 있습니다.

Amazon OpenSearch Serverless에서 보안 시작하기

다음 자습서는 Amazon OpenSearch Serverless 사용을 시작하는 데 도움이 됩니다. 두 자습서 모두 동일한 기본 단계를 수행하지만 한 자습서에서는 콘솔을 사용하고 다른 자습서에서는 AWS CLI를 사용합니다.

이 자습서의 사용 사례는 단순화되었다는 점에 유의하세요. 네트워크와 보안 정책은 상당히 개방적입니다. 프로덕션 워크로드에서는 SAML 인증, VPC 액세스, 제한적인 데이터 액세스 정책과 같은 더욱 강력한 보안 기능을 구성하는 것이 좋습니다.

주제

- 자습서: Amazon OpenSearch Serverless에서 보안 시작하기(콘솔)
- 자습서: Amazon OpenSearch Serverless에서 보안 시작하기(CLI)

자습서: Amazon OpenSearch Serverless에서 보안 시작하기(콘솔)

이 자습서에서는 Amazon OpenSearch Serverless 콘솔을 사용하여 보안 정책을 생성하고 관리하는 기본 단계를 안내합니다.

이 자습서에서는 다음 단계를 완료합니다.

1. 권한 구성
2. 암호화 정책 생성
3. 네트워크 정책 생성
4. 데이터 액세스 정책 구성
5. 컬렉션 생성
6. 데이터 업로드 및 검색

이 자습서에서는를 사용하여 컬렉션을 설정하는 방법을 안내합니다 AWS Management Console. 를 사용하는 것과 동일한 단계는 단원을 AWS CLI참조하십시오the section called “자습서: 보안 시작하기(CLI)”.

1단계: 권한 구성

> **Note**
>
> `Action":"aoss:*"` 또는 `Action":"*"`과 같은 보다 광범위한 자격 증명 기반 정책을 이미 사용 중인 경우 이 단계를 건너뛸 수 있습니다. 하지만 프로덕션 환경에서는 최소 권한 원칙을 따르고 작업을 완료하는 데 필요한 최소 권한만 할당하는 것이 좋습니다.

이 자습서를 완료하려면 올바른 IAM 권한이 있어야 합니다. 사용자 또는 역할에는 다음과 같은 최소 권한이 포함된 연결된 자격 증명 기반 정책이 있어야 합니다.

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": "*"
    }
  ]
}
```

OpenSearch Serverless 권한의 전체 목록은 the section called “ID 및 액세스 관리” 섹션을 참조하세요.

2단계: 암호화 정책 생성

암호화 정책은 OpenSearch Serverless가 컬렉션을 암호화하는 데 사용할 AWS KMS 키를 지정합니다. AWS 관리형 키 또는 다른 키로 컬렉션을 암호화할 수 있습니다. 이 자습서에서는 간소화를 위해 컬렉션을 AWS 관리형 키로 암호화합니다.

암호화 정책 생성하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Serverless(서버리스)를 확장하고 Encryption policies(암호화 정책)를 선택합니다.
3. Create encryption policy(암호화 정책 생성)를 선택합니다.
4. 정책 이름을 books-policy로 지정합니다. 설명을 보려면 Encryption policy for books collection(책 컬렉션에 대한 암호화 정책)을 입력합니다.
5. Resources(리소스)에서 컬렉션 이름으로 지정할 books(책)를 입력합니다. 범위를 좀 더 넓히려면 별표(`books*`)를 포함하여 정책이 “books”(책)라는 단어로 시작하는 모든 컬렉션에 적용되도록 할 수 있습니다.
6. 암호화의 경우 AWS 소유 키 사용을 선택한 상태로 둡니다.
7. 생성(Create)을 선택합니다.

3단계: 네트워크 정책 생성

네트워크 정책은 퍼블릭 네트워크에서 인터넷을 통해 컬렉션에 액세스할 수 있는지 또는 OpenSearch Serverless 관리형 VPC 엔드포인트를 통해 액세스해야 하는지 여부를 결정합니다. 이 자습서에서는 퍼블릭 액세스를 구성해 보겠습니다.

네트워크 정책 생성하기

1. 왼쪽 탐색 창에서 Network policies(네트워크 정책)를 선택한 후 Create network policy(네트워크 정책 생성)를 선택합니다.
2. 정책 이름을 books-policy(책-정책)로 지정합니다. 설명을 보려면 Network policy for books collection(책 컬렉션에 대한 네트워크 정책)을 입력합니다.
3. Rule 1(규칙 1)에서 규칙의 이름을 Public access for books collection(책 컬렉션을 위한 퍼블릭 액세스)으로 지정합니다.
4. 이 자습서에서는 간소화를 위해 books(책) 컬렉션에 대한 공개 액세스를 구성해 보겠습니다. 액세스 유형으로 Public(퍼블릭)을 선택합니다.
5. OpenSearch Dashboards에서 컬렉션에 액세스할 것입니다. 이렇게 하려면 Dashboards 및 OpenSearch 엔드포인트에 대한 네트워크 액세스를 구성해야 합니다. 그러지 않으면 Dashboards가 작동하지 않습니다.

   리소스 유형의 경우 Access to OpenSearch endpoints(OpenSearch 엔드포인트에 대한 액세스)와 Access to OpenSearch Dashboards(OpenSearch Dashboards에 대한 액세스)를 모두 활성화합니다.
6. 두 입력 상자 모두에 Collection Name = books(컬렉션 이름 = 책)를 입력합니다. 이 설정은 단일 컬렉션(books)에만 적용되도록 정책의 범위를 축소합니다. 규칙은 다음과 같아야 합니다.

7. 생성(Create)을 선택합니다.

4단계: 데이터 액세스 정책 생성

데이터 액세스를 구성할 때까지는 컬렉션 데이터에 액세스할 수 없습니다. 데이터 액세스 정책은 1단계에서 구성한 IAM 자격 증명 기반 정책과는 별개입니다. 이를 통해 사용자는 컬렉션 내의 실제 데이터에 액세스할 수 있습니다.

이 자습서에서는 단일 사용자에게 books(책) 컬렉션의 데이터를 인덱싱하는 데 필요한 권한을 제공합니다.

데이터 액세스 정책 생성하기

1. 왼쪽 탐색 창에서 Data access policies(데이터 액세스 정책)를 선택하고 Create access policy(액세스 정책 생성)를 선택합니다.
2. 정책 이름을 books-policy(책-정책)로 지정합니다. 설명을 보려면 Data access policy for books collection(책 컬렉션에 대한 데이터 액세스 정책)을 입력합니다.
3. 정책 정의 방법으로 JSON을 선택하고 JSON 편집기에 다음 정책을 붙여 넣습니다.

   보안 주체 ARN을 OpenSearch Dashboards 및 인덱스 데이터에 로그인하는 데 사용할 계정의 ARN으로 바꿉니다.

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

   이 정책은 단일 사용자에게 books(책) 컬렉션에서 인덱스를 생성하고, 일부 데이터를 인덱싱하고, 검색하는 데 필요한 최소 권한을 제공합니다.
4. 생성(Create)을 선택합니다.

5단계: 컬렉션 생성

이제 암호화 및 네트워크 정책을 구성했으므로 일치하는 컬렉션을 생성할 수 있으며 보안 설정이 자동으로 적용됩니다.

OpenSearch Serverless 컬렉션 생성하기

1. 왼쪽 탐색 창에서 Collections(컬렉션)를 선택하고 Create collection(컬렉션 생성)을 선택합니다.
2. 컬렉션 이름을 books(책)로 지정합니다.
3. 컬렉션 유형에서 Search(검색)를 선택합니다.
4. Encryption(암호화)에서 OpenSearch Serverless는 컬렉션 이름이 `books-policy` 암호화 정책과 일치한다고 알려줍니다.
5. Network access settings(네트워크 액세스 설정)에서 OpenSearch Serverless는 컬렉션 이름이 `books-policy` 네트워크 정책과 일치한다고 알려줍니다.
6. 다음을 선택합니다.
7. Data access policy options(데이터 액세스 정책 옵션)에서 OpenSearch Serverless는 컬렉션 이름이 `books-policy` 데이터 액세스 정책과 일치한다고 알려줍니다.
8. 다음을 선택합니다.
9. 컬렉션 구성을 검토하고 Submit(제출)을 선택합니다. 컬렉션을 초기화하는 데 보통 1분도 채 걸리지 않습니다.

6단계: 데이터 업로드 및 검색

Postman 또는 curl을 사용하여 OpenSearch Serverless 컬렉션에 데이터를 업로드할 수 있습니다. 간결하게 하기 위해 이러한 예시는 OpenSearch Dashboards 콘솔의 Dev Tools(개발 도구)를 사용합니다.

컬렉션의 데이터를 인덱싱하고 검색하기

1. 왼쪽 탐색 창에서 Collections(컬렉션)를 선택하고 books(책) 컬렉션을 선택하여 세부 정보 페이지를 엽니다.
2. 컬렉션에 대한 OpenSearch Dashboards URL을 선택합니다. URL은 `https://`*`collection-id`*`.us-east-1.aoss.amazonaws.com/_dashboards` 형식을 취합니다.
3. 데이터 액세스 정책에서 지정한 보안 주체의 AWS 액세스 및 보안 키를 사용하여 OpenSearch Dashboards에 로그인합니다.
4. OpenSearch Dashboards에서 왼쪽 탐색 메뉴를 열고 Dev Tools(개발 도구)를 선택합니다.

5. books-index라는 단일 인덱스를 생성하려면 다음 명령을 실행합니다.

```
PUT books-index
```

6. books-index라는 단일 문서를 인덱싱하려면 다음 명령을 실행합니다.

```
PUT books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

7. OpenSearch Dashboards에서 데이터를 검색하려면 하나 이상의 인덱스 패턴을 구성해야 합니다. OpenSearch는 이러한 패턴을 사용하여 분석할 인덱스를 식별하기 때문입니다. Dashboards 주 메뉴를 열고 스택 관리(Stack Management)를 선택하고 인덱스 패턴(Index Patterns)을 선택한 다음 인덱스 패턴 생성(Create index pattern)을 선택합니다. 이 자습서에서는 books-index를 입력하세요.
8. 다음 단계(Next step)를 선택한 후 인덱스 패턴 생성(Create index pattern)을 선택합니다. 패턴이 생성되면 `author`, `title`와 같은 다양한 문서 필드를 볼 수 있습니다.
9. 데이터 검색을 시작하려면 기본 메뉴를 다시 열고 Discover(발견)를 선택하거나 검색 API를 사용합니다.

자습서: Amazon OpenSearch Serverless에서 보안 시작하기(CLI)

이 자습서에서는 보안을 위해 콘솔 시작하기 자습서에 설명된 단계를 안내하지만 OpenSearch Service 콘솔 AWS CLI 대신를 사용합니다.

이 튜토리얼에서는 다음 단계를 완료합니다.

1. IAM 권한 정책 생성
2. IAM 정책을 IAM 역할에 연결
3. 암호화 정책 생성
4. 네트워크 정책 생성
5. 컬렉션 생성
6. 데이터 액세스 정책 구성

7. 컬렉션 엔드포인트 검색
8. 연결에 데이터 업로드
9. 컬렉션의 검색 데이터

이 자습서의 목표는 매우 간단한 암호화, 네트워크, 데이터 액세스 설정을 사용하여 단일 OpenSearch Serverless 컬렉션을 설정하는 것입니다. 예를 들어 퍼블릭 네트워크 액세스, 암호화 AWS 관리형 키 용 및 단일 사용자에게 최소 권한을 부여하는 간소화된 데이터 액세스 정책을 구성합니다.

프로덕션 시나리오에서는 SAML 인증, 사용자 지정 암호화 키, VPC 액세스를 포함한 보다 강력한 구성을 구현하는 것이 좋습니다.

OpenSearch Serverless에서 보안 정책 시작하기

1.

> **Note**
>
> `Action":"aoss:*"` 또는 `Action":"*"`과 같은 보다 광범위한 자격 증명 기반 정책을 이미 사용 중인 경우 이 단계를 건너뛸 수 있습니다. 하지만 프로덕션 환경에서는 최소 권한 원칙을 따르고 작업을 완료하는 데 필요한 최소 권한만 할당하는 것이 좋습니다.

시작하려면이 자습서의 단계를 수행하는 데 필요한 최소 권한이 있는 AWS Identity and Access Management 정책을 생성합니다. 정책 이름을 `TutorialPolicy`로 지정하겠습니다.

```
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\": \"*\"}]}"
```

샘플 응답

```
{
    "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. 컬렉션에서 데이터를 인덱싱하고 검색할 IAM 역할에 TutorialPolicy를 연결합니다. 사용자 이름을 TutorialRole로 지정하겠습니다.

```
aws iam attach-role-policy \
  --role-name TutorialRole \
  --policy-arn arn:aws:iam::123456789012:policy/TutorialPolicy
```

3. 컬렉션을 생성하기 전에 이후 단계에서 생성하게 될 books(책) 컬렉션에 AWS 소유 키 를 할당하는 암호화 정책을 생성해야 합니다.

   books(책) 컬렉션에 대한 암호화 정책을 생성하려면 다음 요청을 보냅니다.

```
aws opensearchserverless create-security-policy \
  --name books-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/books\"]}],\"AWSOwnedKey\":true}"
```

   샘플 응답

```
{
    "securityPolicyDetail": {
        "type": "encryption",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDAwNTk5MF8x",
        "policy": {
            "Rules": [
                {
                    "Resource": [
                        "collection/books"
                    ],
                    "ResourceType": "collection"
                }
            ],
```

```
            "AWSOwnedKey": true
        },
        "createdDate": 1669240005990,
        "lastModifiedDate": 1669240005990
    }
}
```

4. books(책) 컬렉션에 대한 퍼블릭 액세스를 제공하는 네트워크 정책을 생성합니다.

```
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}]"
```

샘플 응답

```
{
    "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. books(책) 컬렉션 생성:

```
aws opensearchserverless create-collection --name books --type SEARCH
```

샘플 응답

```
{
    "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. books(책) 컬렉션에서 데이터를 인덱싱하고 검색할 수 있는 최소 권한을 제공하는 데이터 액세스 정책을 생성하세요. 보안 주체 ARN을 1단계의 `TutorialRole` ARN으로 바꿉니다.

```
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\"]}]"
```

샘플 응답

```
{
    "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
    }
}
```

TutorialRole는 이제 책 컬렉션에서 문서를 인덱싱하고 검색할 수 있을 것입니다.

7. OpenSearch API를 호출하려면 컬렉션 엔드포인트가 필요합니다. 다음 요청을 전송하여 collectionEndpoint 파라미터를 검색합니다.

```
aws opensearchserverless batch-get-collection --names books
```

샘플 응답

```
{
    "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**
>
> 컬렉션 상태가 ACTIVE로 변경될 때까지 컬렉션 엔드포인트를 볼 수 없습니다. 컬렉션이 성공적으로 생성될 때까지 상태를 확인하기 위해 여러 번 호출해야 할 수 있습니다.

8. Postman 또는 curl과 같은 HTTP 도구를 사용하여 books(책) 컬렉션에 데이터를 인덱싱합니다. books-index라는 색인을 생성하고 단일 문서를 추가하겠습니다.

   TutorialRole에 대한 보안 인증을 사용하여 이전 단계에서 검색한 컬렉션 엔드포인트에 다음 요청을 보냅니다.

```
PUT https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

   샘플 응답

```
{
  "_index" : "books-index",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
```

```
  "_shards" : {
    "total" : 0,
    "successful" : 0,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 0
}
```

9. 컬렉션에서 데이터 검색을 시작하려면 검색 API를 사용합니다. 다음 쿼리는 기본 검색을 수행합니다.

```
GET https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_search
```

샘플 응답

```
{
    "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 Serverless에 대한 Identity and Access Management

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 어떤 사용자가 OpenSearch Serverless 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- OpenSearch Serverless에 대한 자격 증명 기반 정책
- OpenSearch Serverless의 정책 작업
- OpenSearch Serverless에 대한 정책 리소스
- Amazon OpenSearch Serverless에 사용되는 정책 조건 키
- OpenSearch Serverless를 사용한 ABAC
- OpenSearch Serverless에서 임시 보안 인증 사용
- OpenSearch Serverless에 대한 서비스 연결 역할
- 기타 정책 타입
- OpenSearch Serverless에 대한 자격 증명 기반 정책 예시
- Amazon OpenSearch Serverless에 대한 IAM Identity Center 지원

OpenSearch Serverless에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 위탁자가 연결된 사용자 또는 역할에 적용되므로 위탁자를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

OpenSearch Serverless에 대한 자격 증명 기반 정책 예시

OpenSearch Serverless 자격 증명 기반 정책의 예를 보려면 the section called “자격 증명 기반 정책 예제”을(를) 참조하세요.

OpenSearch Serverless의 정책 작업

정책 작업 지원: 예

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

OpenSearch Serverless의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
aoss
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "aoss:action1",
      "aoss:action2"
         ]
```

와일드카드 문자(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `Describe`라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

```
"Action": "aoss:List*"
```

OpenSearch Serverless 자격 증명 기반 정책의 예를 보려면 OpenSearch Serverless에 대한 자격 증명 기반 정책 예시을(를) 참조하세요.

OpenSearch Serverless에 대한 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 문에는 Resource또는 NotResource요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

Amazon OpenSearch Serverless에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Condition 요소(또는 Condition 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition 요소를 지정하거나 단일 Condition 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 AND 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 OR 작업을 사용하여 조건을 AWS 평가합니다. 문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

ABAC(속성 기반 액세스 제어) 외에도 OpenSearch Serverless는 다음 조건 키를 지원합니다.

- aoss:collection
- aoss:CollectionId
- aoss:index

액세스 정책 및 보안 정책에 대한 권한을 제공하는 경우에도 이러한 조건 키를 사용할 수 있습니다. 예:

```
[
   {
      "Effect":"Allow",
      "Action":[
         "aoss:CreateAccessPolicy",
         "aoss:CreateSecurityPolicy"
      ],
      "Resource":"*",
      "Condition":{
         "StringLike":{
            "aoss:collection":"log"
         }
      }
   }
]
```

이 예시에서는 조건이 컬렉션 이름 또는 패턴과 일치하는 규칙이 포함된 정책에 적용됩니다. 조건에는 다음과 같은 동작이 있습니다.

- `StringEquals` - 정확한 리소스 문자열 “log”를 포함하는 규칙이 있는 정책에 적용됩니다(즉, `collection/log`).
- `StringLike` - “log” 문자열을 포함하는 리소스 문자열이 포함된 규칙이 있는 정책에 적용됩니다(즉, `collection/log` 또한 `collection/logs-application` 또는 `collection/applogs123`).

> **Note**
>
> 컬렉션 조건 키는 인덱스 수준에서 적용되지 않습니다. 예를 들어 위의 정책에서 조건은 리소스 문자열 `index/logs-application/*`을 포함하는 액세스 또는 보안 정책에 적용되지 않습니다.

OpenSearch Serverless 조건 키 목록을 보려면 서비스 승인 참조의 Amazon OpenSearch Serverless에 대한 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스에 대해 알아보려면 Amazon OpenSearch Serverless에서 정의한 작업을 참조하세요.

OpenSearch Serverless를 사용한 ABAC

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. 에서는 AWS이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연결할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 위탁자의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

OpenSearch Serverless 리소스 태그 지정에 대한 자세한 내용은 the section called “컬렉션 태그 지정” 섹션을 참조하세요.

OpenSearch Serverless에서 임시 보안 인증 사용

임시 자격 증명 지원: 예

일부 AWS 서비스 는 임시 자격 증명을 사용하여 로그인할 때 작동하지 않습니다. 임시 자격 증명으로 AWS 서비스 작업하는를 포함한 추가 정보는 AWS 서비스 IAM 사용 설명서의 IAM으로 작업하는 섹션을 참조하세요.

사용자 이름과 암호를 제외한 방법을 AWS Management Console 사용하여에 로그인하는 경우 임시 자격 증명을 사용합니다. 예를 들어 회사의 SSO(Single Sign-On) 링크를 AWS 사용하여에 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 자격 증명을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는 AWS API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다. 그런 다음 이러한 임시 자격 증명을 사용하여 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 access AWS. AWS recommends에 액세스할 수 있습니다. 자세한 정보는 IAM의 임시 보안 자격 증명 섹션을 참조하세요.

OpenSearch Serverless에 대한 서비스 연결 역할

서비스 링크 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

OpenSearch Serverless 서비스 연결 역할 생성 및 관리에 대한 자세한 내용은 the section called “컬렉션 생성 역할” 섹션을 참조하세요.

기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 서비스 제어 정책(SCPs) - SCPs는의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책입니다 AWS Organizations. AWS Organizations 는 기업이 소유한 여러 AWS 계정을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 계정 루트 사용자를 포함하여 멤버 AWS 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 Service control policies을 참조하세요.
- 리소스 제어 정책(RCP) - RCP는 소유한 각 리소스에 연결된 IAM 정책을 업데이트하지 않고 계정의 리소스에 대해 사용 가능한 최대 권한을 설정하는 데 사용할 수 있는 JSON 정책입니다. RCP는 멤버 계정의 리소스에 대한 권한을 제한하며 조직에 속하는지 여부에 관계없이 AWS 계정 루트 사용자를 포함한 자격 증명에 대한 유효 권한에 영향을 미칠 수 있습니다. RCP를 지원하는 AWS 서비스 목록을 포함하여 조직 및 RCPs에 대한 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCPs)을 참조하세요.

OpenSearch Serverless에 대한 자격 증명 기반 정책 예시

기본적으로 사용자 및 역할에는 OpenSearch Serverless 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN의 형식을 포함하여 Amazon OpenSearch Serverless에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 Amazon OpenSearch Serverless에 대한 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- 콘솔에서 OpenSearch Serverless 사용
- OpenSearch Serverless 컬렉션 관리
- OpenSearch Serverless 컬렉션 보기
- OpenSearch API 작업 사용
- OpenSearch API 작업을 위한 ABAC

정책 모범 사례

자격 증명 기반 정책은 매우 강력합니다. 이 정책은 계정에서 사용자가 OpenSearch Serverless 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

ID 기반 정책에 따라 계정에서 사용자가 OpenSearch Serverless 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 AWS CloudFormation. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하

여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.

- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

콘솔에서 OpenSearch Serverless 사용

OpenSearch Service 콘솔 내에서 OpenSearch Serverless에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은 AWS 계정의 OpenSearch Serverless 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하면 콘솔이 해당 정책이 있는 개체(예: IAM 역할)에 대해 의도한 대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 그 대신 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

다음 정책은 사용자가 OpenSearch Service 콘솔 내에서 OpenSearch Serverless에 액세스할 수 있도록 허용합니다.

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"
            ]
        }
    ]
}
```

OpenSearch Serverless 컬렉션 관리

이 정책은 사용자가 Amazon OpenSearch Serverless 컬렉션을 관리할 수 있도록 하는 “컬렉션 관리자” 정책의 예시입니다. 사용자는 컬렉션을 생성하고 보고 삭제할 수 있습니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:aoss:us-east-1::collection/*",
            "Action": [
                "aoss:CreateCollection",
                "aoss:DeleteCollection",
                "aoss:UpdateCollection"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:ListCollections",
                "aoss:CreateAccessPolicy",
                "aoss:CreateSecurityPolicy"
            ],
            "Effect": "Allow"
        }
    ]
}
```

OpenSearch Serverless 컬렉션 보기

이 예시 정책을 통해 사용자는 계정의 모든 Amazon OpenSearch Serverless 컬렉션에 대한 세부 정보를 볼 수 있습니다. 사용자는 컬렉션 또는 관련 보안 정책을 수정할 수 없습니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:ListCollections",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:BatchGetCollection"
            ],
            "Effect": "Allow"
        }
    ]
}
```

OpenSearch API 작업 사용

데이터 영역 API 작업은 OpenSearch Serverless 내 서비스에서 실시간 값을 도출하는 데 사용하는 함수로 구성됩니다. 컨트롤 플레인 API 작업은 환경을 설정하는 데 사용하는 함수로 구성됩니다.

브라우저에서 Amazon OpenSearch Serverless 데이터 영역 API와 OpenSearch Dashboards에 액세스하려면 수집 리소스에 대한 두 개의 IAM 권한을 추가해야 합니다. 이러한 권한은 `aoss:APIAccessAll` 및 `aoss:DashboardsAccessAll`입니다.

> **Note**
>
> 2023년 5월 10일부터 OpenSearch Serverless는 수집 리소스에 대해 이 두 가지 새로운 IAM 권한을 요구합니다. `aoss:APIAccessAll` 권한은 데이터 영역 액세스를 허용하고, `aoss:DashboardsAccessAll` 권한은 브라우저에서 OpenSearch Dashboards를 허용합니다. 두 개의 새 IAM 권한을 추가하지 않으면 403 오류가 발생합니다.

이 예시 정책을 통해 사용자는 계정의 지정된 컬렉션에 대한 데이터 영역 API에 액세스하고 계정의 모든 컬렉션에 대한 OpenSearch 대시보드에 액세스할 수 있습니다.

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` 및 `aoss:DashboardsAccessAll` 모두 수집 리소스에 전체 IAM 권한을 부여하는 반면, 대시보드 권한은 OpenSearch 대시보드 액세스 권한도 제공합니다. 각 권한은 독립적으로 작동하므로 `aoss:APIAccessAll`에서 명시적으로 거부해도 Dev Tools를 포함한 리소스에 대한 `aoss:DashboardsAccessAll` 액세스가 차단되지 않습니다. `aoss:DashboardsAccessAll`에서 거부하는 경우에도 마찬가지입니다. OpenSearch Serverless는 다음과 같은 전역 조건 키를 지원합니다.

- `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

다음은 데이터 영역 호출aws:SourceIp에 대한 보안 주체의 IAM 정책의 조건 블록에서를 사용하는 예입니다.

```
"Condition": {
    "IpAddress": {
         "aws:SourceIp": "203.0.113.0"
    }
}
```

다음은 데이터 영역 호출aws:SourceVpc에 대한 보안 주체의 IAM 정책의 조건 블록에서를 사용하는 예입니다.

```
"Condition": {
    "StringEquals": {
        "aws:SourceVpc": "vpc-0fdd2445d8EXAMPLE"
    }
```

```
}
```

또한 다음 OpenSearch Serverless 특정 키에 대한 지원이 제공됩니다.

- `aoss:CollectionId`
- `aoss:collection`

다음은 데이터 영역 호출`aoss:collection`에 대한 보안 주체의 IAM 정책의 조건 블록에서를 사용하는 예입니다.

```
"Condition": {
    "StringLike": {
         "aoss:collection": "log-*"
    }
}
```

OpenSearch API 작업을 위한 ABAC

자격 증명 기반 정책을 사용하면 태그를 사용하여 Amazon OpenSearch Serverless 데이터 영역 APIs. 다음 정책은 컬렉션에 `team:devops` 태그가 있는 경우 연결된 보안 주체가 데이터 영역 APIs에 액세스할 수 있도록 허용하는 예제입니다.

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"
                }
            }
        }
    ]
}
```

다음 정책은 컬렉션에 environment:production 태그가 있는 경우 연결된 보안 주체가 데이터 영역 APIs 및 대시보드 액세스에 액세스하는 것을 거부하는 예입니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "aoss:APIAccessAll",
                "aoss:DashboardsAccessAll",
            ],
            "Resource": "arn:aws:aoss:region:account-id:collection/collection-
id",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/environment": "production"
                }
            }
        }
    ]
}
```

Amazon OpenSearch Serverless는 데이터 영역 APIs에 대한 RequestTag 및 TagKeys 전역 조건 키를 지원하지 않습니다.

Amazon OpenSearch Serverless에 대한 IAM Identity Center 지원

Amazon OpenSearch Serverless에 대한 IAM Identity Center 지원

IAM Identity Center 보안 주체(사용자 및 그룹)를 사용하여 Amazon OpenSearch 애플리케이션을 통해 Amazon OpenSearch Serverless 데이터에 액세스할 수 있습니다. Amazon OpenSearch Serverless에 대한 IAM Identity Center 지원을 활성화하려면 IAM Identity Center 사용을 활성화해야 합니다. 이 작업을 수행하는 방법에 대한 자세한 내용은 IAM Identity Center란 무엇입니까?를 참조하세요.

IAM Identity Center 인스턴스가 생성된 후 고객 계정 관리자는 Amazon OpenSearch Serverless 서비스에 대한 IAM Identity Center 애플리케이션을 생성해야 합니다. CreateSecurityConfig:를 호출하여이

작업을 수행할 수 있습니다. 고객 계정 관리자는 요청 권한 부여에 사용할 속성을 지정할 수 있습니다. 사용되는 기본 속성은 `UserId` 및 입니다. `GroupId.`

Amazon OpenSearch Serverless용 IAM Identity Center 통합은 다음 AWS IAM Identity Center(IAM) 권한을 사용합니다.

- `aoss:CreateSecurityConfig` - IAM Identity Center 공급자 생성
- `aoss:ListSecurityConfig` - 현재 계정의 모든 IAM Identity Center 공급자를 나열합니다.
- `aoss:GetSecurityConfig` - IAM Identity Center 공급자 정보를 봅니다.
- `aoss:UpdateSecurityConfig` - 지정된 IAM Identity Center 구성 수정
- `aoss:DeleteSecurityConfig` - IAM Identity Centerprovider를 삭제합니다.

다음 아이덴티티 기반 액세스 정책을 사용하여 모든 IAM Identity Center 구성을 관리할 수 있습니다.

JSON

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Action": [
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> `Resource` 요소는 와일드카드여야 합니다.

IAM Identity Center 공급자 생성(콘솔)

IAM Identity Center 공급자를 생성하여 OpenSearch Application으로 인증을 활성화할 수 있습니다. OpenSearch Dashboards에 대한 IAM Identity Center 인증을 활성화하려면 다음 단계를 수행합니다.

1. Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 패널에서 서버리스를 확장하고 인증을 선택합니다.
3. IAM Identity Center 인증을 선택합니다.
4. 편집을 선택합니다.
5. IAM Identity Center로 인증 옆의 확인란을 선택합니다.
6. 드롭다운 메뉴에서 사용자 및 그룹 속성 키를 선택합니다. 사용자 속성은 `UserName`, `UserId`및 `Email`를 기반으로 사용자에게 권한을 부여하는 데 사용됩니다. 그룹 속성은 `GroupName` 및를 기반으로 사용자를 인증하는 데 사용됩니다`GroupId`.
7. IAM Identity Center 인스턴스를 선택합니다.
8. 저장을 선택합니다.

IAM Identity Center 공급자 생성(AWS CLI)

AWS Command Line Interface (AWS CLI)를 사용하여 IAM Identity Center 공급자를 생성하려면 다음 명령을 사용합니다.

```
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"
}'
```

IAM Identity Center가 활성화되면 고객은 사용자 및 그룹 속성만 수정할 수 있습니다.

```
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"
}'
```

를 사용하여 IAM Identity Center 공급자를 보려면 다음 명령을 AWS Command Line Interface사용합니다.

```
aws opensearchserverless list-security-configs --type iamidentitycenter
```

IAM Identity Center 공급자 삭제

IAM Identity Center는 두 개의 공급자 인스턴스를 제공합니다. 하나는 조직 계정용이고 다른 하나는 멤버 계정용입니다. IAM Identity Center 인스턴스를 변경해야 하는 경우 `DeleteSecurityConfig` API를 통해 보안 구성을 삭제하고 새 IAM Identity Center 인스턴스를 사용하여 새 보안 구성을 생성해야 합니다. 다음 명령을 사용하여 IAM Identity Center 공급자를 삭제할 수 있습니다.

```
aws opensearchserverless delete-security-config \
--region us-east-1 \
--id <id_from_list_security_configs>
```

IAM Identity Center에 컬렉션 데이터에 대한 액세스 권한 부여

IAM Identity Center 공급자가 활성화되면 IAM Identity Center 보안 주체를 포함하도록 컬렉션 데이터 액세스 정책을 업데이트할 수 있습니다. IAM Identity Center 보안 주체는 다음 형식으로 업데이트해야 합니다.

```
[
   {
"Rules":[
       ...
      ],
      "Principal":[
         "iamidentitycenter/<iamidentitycenter-instance-id>/user/<UserName>",
         "iamidentitycenter/<iamidentitycenter-instance-id>/group/<GroupId>"
      ]
   }
]
```

> **Note**
>
> Amazon OpenSearch Serverless는 모든 고객 컬렉션에 대해 하나의 IAM Identity Center 인스턴스만 지원하며 단일 사용자에 대해 최대 100개의 그룹을 지원할 수 있습니다. 허용된 인스턴스 수를 초과하여 사용하려고 하면 데이터 액세스 정책 권한 부여 처리에 불일치가 발생하여 403오류 메시지가 표시됩니다.

컬렉션, 인덱스 또는 둘 다에 대한 액세스 권한을 부여할 수 있습니다. 다른 사용자에게 다른 권한을 부여하려면 여러 규칙을 생성해야 합니다. 사용 가능한 권한 목록은 Amazon OpenSearch Service의 Identity and Access Management를 참조하세요. 액세스 정책의 형식을 지정하는 방법에 대한 자세한 내용은 수집 데이터에 대한 SAML 자격 증명 액세스 권한 부여를 참조하세요.

IAM Identity Center는 두 개의 공급자 인스턴스를 제공합니다. 하나는 조직 계정용이고 다른 하나는 멤버 계정용입니다. IAM Identity Center 인스턴스를 변경해야 하는 경우 `DeleteSecurityConfig` API를 통해 보안 구성을 삭제하고 새 IAM Identity Center 인스턴스를 사용하여 새 보안 구성을 생성해야 합니다. 다음 명령을 사용하여 IAM Identity Center 공급자를 삭제할 수 있습니다.

```
aws opensearchserverless delete-security-config \
--region us-east-1 \
--id <id_from_list_security_configs>
```

Amazon OpenSearch Serverless 암호화

저장 시 암호화

생성한 각 Amazon OpenSearch Serverless 컬렉션은 데이터에 대한 무단 액세스를 방지하는 보안 기능인 저장 데이터의 암호화로 보호됩니다. 유휴 시 암호화는 AWS Key Management Service (AWS KMS)를 사용하여 암호화 키를 저장하고 관리합니다. 암호화를 수행하기 위해 256비트 키(AES-256)가 있는 고급 암호화 표준 알고리즘을 사용합니다.

주제

- 암호화 정책
- 고려 사항
- 필요한 권한
- 고객 관리형 키에 대한 키 정책
- OpenSearch Serverless가 AWS KMS에서 권한 부여를 사용하는 방법

- 암호화 정책 생성(콘솔)
- 암호화 정책 생성(AWS CLI)
- 암호화 정책 보기
- 암호화 정책 업데이트
- 암호화 정책 삭제

암호화 정책

암호화 정책을 사용하면 특정 이름 또는 패턴과 일치하는 새로 생성한 컬렉션에 암호화 키를 자동으로 할당하여 여러 컬렉션을 대규모로 관리할 수 있습니다.

암호화 정책을 생성할 때 `MyCollection*`과 같은 와일드카드 기반 일치 규칙인 접두사를 지정하거나 단일 컬렉션 이름을 입력할 수 있습니다. 그런 다음 해당 이름 또는 접두사 패턴과 일치하는 컬렉션을 생성하면 정책과 해당 KMS 키가 자동으로 컬렉션에 할당됩니다.

암호화 정책에는 다음 요소가 포함됩니다.

- `Rules` – 각각 다음과 같은 하위 요소가 포함된 하나 이상의 컬렉션 일치 규칙:
  - `ResourceType` – 현재 유일한 옵션은 “컬렉션”입니다. 암호화 정책은 컬렉션 리소스에만 적용됩니다.
  - `Resource` – `collection/<collection name|pattern>` 형식으로 정책이 적용될 하나 이상의 컬렉션 이름 또는 패턴입니다.
- `AWSOwnedKey` – AWS 소유 키를 사용할지 여부.
- `KmsARN` – `AWSOwnedKey`를 false로 설정한 경우 연결된 컬렉션을 암호화할 KMS 키의 Amazon 리소스 이름(ARN)을 지정합니다. 이 파라미터를 포함하면 OpenSearch Serverless는 `AWSOwnedKey` 파라미터를 무시합니다.

다음 샘플 정책은 `autopartsinventory`라는 이름의 향후 컬렉션과 “sales”라는 용어로 시작하는 컬렉션에 고객 관리형 키를 할당합니다.

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
```

```
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

정책이 컬렉션 이름과 일치하더라도 리소스 패턴에 와일드카드(*)가 포함된 경우 컬렉션 생성 중에 이 자동 할당을 재정의하도록 선택할 수 있습니다. 자동 키 할당을 재정의하도록 선택하면 OpenSearch Serverless는 auto-<*collection-name*>이라는 암호화 정책을 생성하여 컬렉션에 연결합니다. 정책은 처음에는 단일 컬렉션에만 적용되지만 추가 컬렉션을 포함하도록 수정할 수 있습니다.

컬렉션과 더 이상 일치하지 않도록 정책 규칙을 수정하면 해당 컬렉션에서 연결된 KMS 키 할당이 취소되지 않습니다. 컬렉션은 항상 초기 암호화 키로 암호화된 상태로 유지됩니다. 컬렉션의 암호화 키를 변경하려면 컬렉션을 다시 생성해야 합니다.

여러 정책의 규칙이 컬렉션과 일치하는 경우 더 구체적인 규칙이 사용됩니다. 예를 들어 한 정책에 `collection/log*`에 대한 규칙이 포함되어 있고 다른 정책에 `collection/logSpecial`에 대한 규칙이 포함된 경우 더 구체적이기 때문에 두 번째 정책에 대한 암호화 키가 사용됩니다.

다른 정책에 이름이나 접두사가 이미 있는 경우 정책에서 이를 사용할 수 없습니다. 다른 암호화 정책에서 동일한 리소스 패턴을 구성하려고 하면 OpenSearch Serverless에 오류가 표시됩니다.

고려 사항

컬렉션의 암호화를 구성할 때 다음 사항을 고려하세요.

- 저장된 암호화는 모든 서버리스 컬렉션에 필수입니다.
- 고객 관리형 키 또는 AWS 소유 키를 사용할 수 있습니다. 고객 관리형 키를 선택하는 경우 자동 키 교체를 활성화하는 것이 좋습니다.
- 컬렉션을 생성한 후에는 컬렉션의 암호화 키를 변경할 수 없습니다. 컬렉션을 처음 설정할 때 AWS KMS 사용할 항목을 신중하게 선택합니다.
- 컬렉션은 단일 암호화 정책과만 일치할 수 있습니다.
- 고유한 KMS 키가 있는 컬렉션은 OpenSearch 컴퓨팅 유닛(OCU)을 다른 컬렉션과 공유할 수 없습니다. 고유 키가 있는 각 컬렉션에는 고유한 4개의 OCU가 필요합니다.
- 암호화 정책에서 KMS 키를 업데이트하는 경우 변경 사항은 KMS 키가 이미 할당된 일치하는 기존 컬렉션에 영향을 미치지 않습니다.

- OpenSearch Serverless는 고객 관리형 키에 대한 사용자 권한을 명시적으로 확인하지 않습니다. 사용자가 데이터 액세스 정책을 통해 컬렉션에 액세스할 권한이 있는 경우 연결된 키로 암호화된 데이터를 수집하고 쿼리할 수 있습니다.

필요한 권한

OpenSearch Serverless의 저장 시 암호화는 다음 AWS Identity and Access Management (IAM) 권한을 사용합니다. 사용자를 특정 컬렉션으로 제한하도록 IAM 조건을 지정할 수 있습니다.

- `aoss:CreateSecurityPolicy` – 암호화 정책을 생성합니다.
- `aoss:ListSecurityPolicies` – 모든 암호화 정책과 해당 정책이 연결된 컬렉션을 나열합니다.
- `aoss:GetSecurityPolicy` – 특정 암호화 정책의 세부 정보를 봅니다.
- `aoss:UpdateSecurityPolicy` – 암호화 정책을 수정합니다.
- `aoss:DeleteSecurityPolicy` – 암호화 정책을 삭제합니다.

다음 샘플 ID 기반 액세스 정책은 사용자가 `collection/application-logs` 리소스 패턴으로 암호화 정책을 관리하는 데 필요한 최소 권한을 제공합니다.

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":"*"
         }
      ]
}
```

고객 관리형 키에 대한 키 정책

컬렉션을 보호하기 위해 고객 관리형 키를 선택하면 OpenSearch Serverless는 선택한 보안 주체를 대신하여 KMS 키를 사용할 수 있는 권한을 얻습니다. 해당 보안 주체, 즉 사용자 또는 역할은 OpenSearch Serverless에 필요한 KMS 키에 대한 권한이 있어야 합니다. 키 정책 또는 IAM 정책에서 이러한 권한을 제공할 수 있습니다.

OpenSearch Serverless는 고객 관리형 키에 대해 최소한 다음 권한이 있어야 합니다.

- kms:DescribeKey
- kms:CreateGrant

예:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:CreateGrant"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "aoss.us-east-1.amazonaws.com"
        },
        "Bool": {
```

```
          "kms:GrantIsForAWSResource": "true"
        }
      }
    }
  ]
}
```

OpenSearch Serverless는 kms:GenerateDataKey 및 kms:Decrypt 권한으로 권한 부여를 생성합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서에서 AWS KMS의 키 정책 사용을 참조하세요.

OpenSearch Serverless가 AWS KMS에서 권한 부여를 사용하는 방법

OpenSearch Serverless는 고객 관리형 키를 사용하기 위해 권한 부여가 필요합니다.

새 키를 사용하여 계정에 암호화 정책을 생성하면 OpenSearch Serverless는 CreateGrant 요청을 에 전송하여 사용자를 대신하여 권한 부여를 생성합니다 AWS KMS. 의 권한 부여 AWS KMS 는 OpenSearch Serverless에 고객 계정의 KMS 키에 대한 액세스 권한을 부여하는 데 사용됩니다.

OpenSearch Serverless는 다음 내부 작업에 대해 고객 관리형 키를 사용할 수 있는 권한이 필요합니다.

- DescribeKey 요청을에 전송 AWS KMS 하여 제공된 대칭 고객 관리형 키 ID가 유효한지 확인합니다.
- GenerateDataKey 요청을 KMS 키로 전송하여 객체를 암호화하는 데 사용할 데이터 키를 생성합니다.
- AWS KMS 에 Decrypt 요청을 보내 암호화된 데이터 키를 복호화하여 데이터를 암호화하는 데 사용할 수 있도록 합니다.

언제든지 권한 부여에 대한 액세스 권한을 취소하거나 고객 관리형 키에 대한 서비스 액세스를 제거할 수 있습니다. 그렇게 하면 OpenSearch Serverless는 고객 관리형 키로 암호화된 데이터에 액세스할 수 없으며, 이는 해당 데이터에 의존하는 모든 작업에 영향을 미치고 비동기식 워크플로에서 `AccessDeniedException` 오류 및 실패로 이어집니다.

OpenSearch Serverless는 지정된 고객 관리형 키가 보안 정책 또는 컬렉션과 연결되지 않은 경우 비동기 워크플로에서 권한을 사용 중지합니다.

암호화 정책 생성(콘솔)

암호화 정책에서 정책이 적용될 KMS 키와 일련의 수집 패턴을 지정합니다. 정책에 정의된 패턴 중 하나와 일치하는 모든 새 컬렉션에는 컬렉션을 생성할 때 해당 KMS 키가 할당됩니다. 컬렉션을 생성하기 전에 암호화 정책을 생성하는 것이 좋습니다.

OpenSearch Serverless 암호화 정책 생성하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 패널에서 Serverless(서버리스)를 확장하고 Encryption policies(암호화 정책)를 선택합니다.
3. Create encryption policy(암호화 정책 생성)를 선택합니다.
4. 정책의 이름 및 설명을 입력합니다.
5. Resources(리소스)에서 이 암호화 정책에 대한 리소스 패턴을 하나 이상 입력합니다. 패턴 중 하나와 일치하는 현재 AWS 계정 및 리전에서 새로 생성된 컬렉션은 모두 자동으로 이 정책에 할당됩니다. 예를 들어 와일드카드 없이 `ApplicationLogs`를 입력하고 나중에 해당 이름으로 컬렉션을 생성하면 정책과 해당 KMS 키가 해당 컬렉션에 할당됩니다.

   이름이 `Logs`로 시작하는 새 컬렉션에 정책을 할당하는 `Logs*`와 같은 접두사를 입력할 수도 있습니다. 와일드카드를 사용하면 여러 컬렉션의 암호화 설정을 대규모로 관리할 수 있습니다.
6. Encryption(암호화)에서 사용할 KMS 키를 선택합니다.
7. 생성(Create)을 선택합니다.

다음 단계: 컬렉션 생성

하나 이상의 암호화 정책을 구성한 후 해당 정책에 정의된 규칙과 일치하는 컬렉션을 생성할 수 있습니다. 지침은 the section called “컬렉션 생성” 섹션을 참조하세요.

컬렉션 생성의 Encryptions(암호화) 단계에서 OpenSearch Serverless는 입력한 이름이 암호화 정책에 정의된 패턴과 일치함을 알리고 해당 KMS 키를 컬렉션에 자동으로 할당합니다. 리소스 패턴에 와일드카드(*)가 포함된 경우 일치 항목을 재정의하고 고유한 키를 선택할 수 있습니다.

암호화 정책 생성(AWS CLI)

OpenSearch Serverless API 작업을 사용하여 암호화 정책을 생성하려면 리소스 패턴과 암호화 키를 JSON 형식으로 지정합니다. CreateSecurityPolicy 요청은 인라인 정책과 .json 파일을 모두 허용합니다.

암호화 정책은 다음 형식을 사용합니다. 이 샘플 my-policy.json 파일은 이름이 sales로 시작하는 모든 컬렉션뿐만 아니라 이름이 autopartsinventory인 향후 모든 컬렉션과 일치합니다.

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

서비스 소유 키를 사용하려면 AWSOwnedKey를 true로 설정합니다.

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":true
}
```

다음 요청은 암호화 정책을 생성합니다.

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy file://my-policy.json
```

그런 다음 CreateCollection API 작업을 사용하여 리소스 패턴 중 하나와 일치하는 하나 이상의 컬렉션을 생성합니다.

암호화 정책 보기

컬렉션을 생성하기 전에 계정의 기존 암호화 정책을 미리 보고 컬렉션 이름과 일치하는 리소스 패턴을 가진 정책을 확인하는 것이 좋습니다. 다음 ListSecurityPolicies 요청은 계정의 모든 암호화 정책을 나열합니다.

```
aws opensearchserverless list-security-policies --type encryption
```

요청은 구성된 모든 암호화 정책에 대한 정보를 반환합니다. `policy` 요소의 콘텐츠를 사용하여 정책에 정의된 패턴 규칙을 봅니다.

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

KMS 키를 포함하여 특정 정책에 대한 자세한 정보를 보려면 GetSecurityPolicy 명령을 사용합니다.

암호화 정책 업데이트

암호화 정책에서 KMS 키를 업데이트하면 변경된 이름 또는 패턴과 일치하는 새로 생성한 컬렉션에만 변경 내용이 적용됩니다. KMS 키가 이미 할당된 기존 컬렉션에는 영향을 미치지 않습니다.

정책 일치 규칙에도 동일하게 적용됩니다. 규칙을 추가, 수정 또는 삭제하면 새로 생성한 컬렉션에만 변경 사항이 적용됩니다. 더 이상 컬렉션 이름과 일치하지 않도록 정책 규칙을 수정해도 기존 컬렉션에 할당된 KMS 키는 손실되지 않습니다.

OpenSearch Serverless 콘솔에서 암호화 정책을 업데이트하려면 Encryption policies(암호화 정책)를 선택하고 수정할 정책을 선택한 다음 Edit(편집)를 선택합니다. 변경하고 Save(저장)를 선택합니다.

OpenSearch Serverless API를 사용하여 암호화 정책을 업데이트하려면 UpdateSecurityPolicy 작업을 사용합니다. 다음 요청은 새 정책 JSON 문서로 암호화 정책을 업데이트합니다.

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy-version 2 \
    --policy file://my-new-policy.json
```

암호화 정책 삭제

암호화 정책을 삭제해도 정책에 정의된 KMS 키를 현재 사용하고 있는 컬렉션은 영향을 받지 않습니다. OpenSearch Serverless 콘솔에서 정책을 삭제하려면 정책을 선택하고 Delete(삭제)를 선택합니다.

DeleteSecurityPolicy 작업을 사용할 수도 있습니다.

```
aws opensearchserverless delete-security-policy --name my-policy --type encryption
```

전송 중 암호화

OpenSearch Serverless 내에서 컬렉션의 모든 경로는 업계 표준 AES-256 암호가 적용된 전송 계층 보안 1.2(TLS)를 사용하여 전송 중에 암호화됩니다. TLS 1.2를 통해서도 Opensearch의 모든 API와 Dashboards에 액세스할 수 있습니다. TLS는 네트워크를 통해 교환되는 정보를 암호화하는 데 사용되는 업계 표준 암호화 프로토콜 세트입니다.

Amazon OpenSearch Serverless에 대한 네트워크 액세스

Amazon OpenSearch Serverless 컬렉션에 대한 네트워크 설정에 따라 퍼블릭 네트워크에서 인터넷을 통해 컬렉션에 액세스할 수 있는지 또는 프라이빗 액세스로 액세스해야 하는지 여부가 결정됩니다.

프라이빗 액세스는 다음 중 하나 또는 둘 다에 적용될 수 있습니다.

- OpenSearch Serverless 관리형 VPC 엔드포인트
- Amazon Bedrock AWS 서비스 과 같이 지원됨

컬렉션의 OpenSearch 엔드포인트 및 해당 OpenSearch Dashboards 엔드포인트에 대해 네트워크 액세스를 개별적으로 구성할 수 있습니다.

네트워크 액세스는 다양한 소스 네트워크에서 액세스를 허용하기 위한 격리 메커니즘입니다. 예를 들어 컬렉션의 OpenSearch 대시보드 엔드포인트에 퍼블릭 액세스로 액세스할 수 있지만 OpenSearch API 엔드포인트에는 액세스할 수 없는 경우, 사용자는 퍼블릭 네트워크에서 연결할 때 대시보드를 통해서만 컬렉션 데이터에 액세스할 수 있습니다. 퍼블릭 네트워크에서 직접 OpenSearch API를 호출하

려고 하면 차단됩니다. 네트워크 설정은 이러한 소스에서 리소스 유형으로의 순열에 사용할 수 있습니다. Amazon OpenSearch Serverless는 IPv4 및 IPv6 연결을 모두 지원합니다.

주제

- 네트워크 정책
- 고려 사항
- 네트워크 정책을 구성하는 데 필요한 권한
- 정책 우선순위
- 네트워크 정책 생성(콘솔)
- 네트워크 정책 생성(AWS CLI)
- 네트워크 정책 보기
- 네트워크 정책 업데이트
- 네트워크 정책 삭제

네트워크 정책

네트워크 정책을 사용하면 정책에 정의된 규칙과 일치하는 컬렉션에 네트워크 액세스 설정을 자동으로 할당하여 많은 컬렉션을 대규모로 관리할 수 있습니다.

네트워크 정책에서는 일련의 규칙을 지정합니다. 이 규칙은 컬렉션 엔드포인트 및 OpenSearch Dashboards 엔드포인트에 대한 액세스 권한을 정의합니다. 각 규칙은 액세스 유형(퍼블릭 또는 프라이빗)과 리소스 유형(컬렉션 및/또는 OpenSearch 대시보드 엔드포인트)으로 구성됩니다. 각 리소스 유형(`collection` 및 `dashboard`)에 대해 정책을 적용할 컬렉션을 정의하는 일련의 규칙을 지정합니다.

이 샘플 정책에서 첫 번째 규칙은 `marketing*` 용어로 시작하는 모든 컬렉션에서 컬렉션 엔드포인트와 대시보드 엔드포인트 모두에 대한 VPC 엔드포인트 액세스를 지정합니다. 또한 Amazon Bedrock 액세스를 지정합니다.

> **Note**
>
> Amazon Bedrock AWS 서비스 과 같은에 대한 프라이빗 액세스는 컬렉션의 OpenSearch 엔드포인트에만 적용되며 OpenSearch Dashboards 엔드포인트에는 적용되지 않습니다. `ResourceType`가 인 경우에도 OpenSearch 대시보드에 대한 액세스 권한을 부여받을 수 `dashboard` AWS 서비스 없습니다.

두 번째 규칙은 finance 컬렉션에 대한 퍼블릭 액세스를 지정하지만 컬렉션 엔드포인트에 대해서만 (Dashboards 액세스 없음) 지정합니다.

```
[
   {
      "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
   }
]
```

이 정책은 “finance”(재무)로 시작하는 컬렉션에 대한 OpenSearch Dashboards에 대한 퍼블릭 액세스만 제공합니다. OpenSearch API에 직접 액세스하려는 모든 시도는 실패합니다.

```
[
  {
    "Description": "Dashboards access",
    "Rules": [
      {
        "ResourceType": "dashboard",
        "Resource": [
          "collection/finance*"
        ]
      }
    ],
    "AllowFromPublic": true
  }
]
```

네트워크 정책은 기존 컬렉션뿐만 아니라 향후 컬렉션에도 적용될 수 있습니다. 예를 들어 컬렉션을 만든 다음 컬렉션 이름과 일치하는 규칙을 사용하여 네트워크 정책을 생성할 수 있습니다. 컬렉션을 생성하기 전에 네트워크 정책을 먼저 생성해야 하는 것은 아닙니다.

고려 사항

컬렉션에 대한 네트워크 액세스를 구성할 때 다음 사항을 고려하세요.

- 컬렉션에 대한 VPC 엔드포인트 액세스를 구성하려면 먼저 OpenSearch Serverless 관리형 VPC 엔드포인트를 하나 이상 생성해야 합니다.
- 에 대한 프라이빗 액세스는 컬렉션의 OpenSearch 엔드포인트 AWS 서비스 에만 적용되며 OpenSearch Dashboards 엔드포인트에는 적용되지 않습니다. `ResourceType`가 인 경우에도 OpenSearch Dashboards에 대한 액세스 권한을 부여받을 수 `dashboard` AWS 서비스 없습니다.
- 퍼블릭 네트워크에서 컬렉션에 액세스할 수 있는 경우 모든 OpenSearch Serverless 관리형 VPC 엔드포인트 및 모든 AWS 서비스에서도 액세스할 수 있습니다.
- 단일 컬렉션에 여러 네트워크 정책을 적용할 수 있습니다. 자세한 내용은 the section called “정책 우선순위” 단원을 참조하십시오.

네트워크 정책을 구성하는 데 필요한 권한

OpenSearch Serverless에 대한 네트워크 액세스는 다음 AWS Identity and Access Management (IAM) 권한을 사용합니다. IAM 조건을 지정하여 사용자를 특정 컬렉션과 연결된 네트워크 정책으로 제한할 수 있습니다.

- `aoss:CreateSecurityPolicy` – 네트워크 액세스 정책을 생성합니다.
- `aoss:ListSecurityPolicies` – 현재 계정의 모든 네트워크 정책을 나열합니다.
- `aoss:GetSecurityPolicy` – 네트워크 액세스 정책 사양을 봅니다.
- `aoss:UpdateSecurityPolicy` – 주어진 네트워크 액세스 정책을 수정하고 VPC ID 또는 퍼블릭 액세스 지정을 변경합니다.
- `aoss:DeleteSecurityPolicy` – 모든 컬렉션에서 분리된 후 네트워크 액세스 정책을 삭제합니다.

다음 자격 증명 기반 액세스 정책을 통해 사용자는 모든 네트워크 정책을 보고 리소스 패턴 `collection/application-logs`로 정책을 업데이트할 수 있습니다.

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**
>
> 또한 OpenSearch Serverless에는 컬렉션 리소스에 대한 `aoss:APIAccessAll` 및 `aoss:DashboardsAccessAll` 권한이 필요합니다. 자세한 내용은 the section called "OpenSearch API 작업 사용" 단원을 참조하십시오.

정책 우선순위

네트워크 정책 규칙이 정책 내에서 또는 정책 간에 중복되는 상황이 있을 수 있습니다. 이 경우 퍼블릭 액세스를 지정하는 규칙이 두 규칙에 공통적인 모든 컬렉션에 대해 프라이빗 액세스를 지정하는 규칙보다 우선 적용됩니다.

예를 들어 다음 정책에서 두 규칙 모두 `finance` 컬렉션에 대한 네트워크 액세스를 할당하지만 한 규칙은 VPC 액세스를 지정하고 다른 규칙은 퍼블릭 액세스를 지정합니다. 이 상황에서 퍼블릭 액세스는 finance(재무) 컬렉션에 대해서만 VPC 액세스를 재정의하므로(두 규칙 모두에 존재하기 때문에) 퍼블릭 네트워크에서 finance(재무) 컬렉션에 액세스할 수 있습니다. sales 컬렉션은 지정된 엔드포인트에서 VPC 액세스 권한을 가집니다.

```
[
   {
      "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
   }
]
```

서로 다른 규칙의 여러 VPC 엔드포인트가 컬렉션에 적용되는 경우 규칙은 추가적으로 적용되며 지정된 모든 엔드포인트에서 컬렉션에 액세스할 수 있습니다. `AllowFromPublic`을 `true`로 설정했지만 하나 이상의 `SourceVPCEs` 또는 `SourceServices`를 제공하는 경우 OpenSearch Serverless는 VPC 엔드포인트 및 서비스 식별자를 무시하고 연결된 컬렉션에 퍼블릭 액세스 권한이 부여됩니다.

네트워크 정책 생성(콘솔)

네트워크 정책은 기존 정책뿐만 아니라 향후 정책에도 적용될 수 있습니다. 컬렉션을 생성하기 전에 네트워크 정책을 생성하는 것이 좋습니다.

OpenSearch Serverless 네트워크 정책 생성하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 패널에서 Serverless(서버리스)를 확장하고 Network policies(네트워크 정책)를 선택합니다.
3. Create network policy(네트워크 정책 생성)를 선택합니다.
4. 정책의 이름 및 설명을 입력합니다.
5. 하나 이상의 규칙을 입력합니다. 이러한 규칙은 OpenSearch Serverless 컬렉션 및 해당 OpenSearch Dashboards 엔드포인트에 대한 액세스 권한을 정의합니다.

   각 규칙에는 다음 요소가 포함됩니다.

Element	설명
규칙 이름	규칙의 내용을 설명하는 이름입니다. 예: “마케팅 팀을 위한 VPC 액세스”
액세스 유형	퍼블릭 또는 프라이빗 액세스를 선택합니다. 그리고 다음 중 하나 또는 둘 다를 선택합니다. • 액세스를 위한 VPC 엔드포인트 - 하나 이상의 OpenSearch Serverless 관리형 VPC 엔드포인트를 지정합니다. • AWS 서비스 프라이빗 액세스 - 지원되는를 하나 이상 선택합니다 AWS 서비스.
리소스 유형	OpenSearch 엔드포인트(OpenSearch API에 대한 호출 허용), OpenSearch Dashboards(시각화 및 OpenSearch 플러그인의 사용자 인터페이스에 대한 액세스 허용) 또는 둘 다에 대한 액세스를 제공할지 여부를 선택합니다. **Note** AWS 서비스 프라이빗 액세스는 컬렉션의 OpenSearch 엔드포인트에만 적용되며 OpenSearch Dashboards 엔드포인트에는 적용되지 않습니다. OpenSearch 대시보드를 선택하더라도 AWS 서비스 에는 엔드포인트 액세스 권한만 부여할 수 있습니다.

선택한 각 리소스 유형에 대해 기존 컬렉션을 선택하여 정책 설정을 적용하거나 하나 이상의 리소스 패턴을 생성할 수 있습니다. 리소스 패턴은 접두사와 와일드카드(*)로 구성되며 정책 설정이 적용될 컬렉션을 정의합니다.

예를 들어 `Marketing*`이라는 패턴을 포함하는 경우 이름이 “Marketing”으로 시작하는 새 컬렉션이나 기존 컬렉션에는 이 정책의 네트워크 설정이 자동으로 적용됩니다. 단일 와일드카드(*)는 모든 현재 및 향후 컬렉션에 정책을 적용합니다.

또한 `Finance`와 같이 와일드카드 없이 향후 컬렉션의 이름을 지정할 수 있습니다. OpenSearch Serverless는 정확히 동일한 이름으로 새로 생성된 컬렉션에 정책 설정을 적용합니다.

6. 정책 구성에 만족하면 Create(생성)를 선택합니다.

네트워크 정책 생성(AWS CLI)

OpenSearch Serverless API 작업을 사용하여 네트워크 정책을 생성하려면 JSON 형식으로 규칙을 지정합니다. CreateSecurityPolicy 요청은 인라인 정책과 .json 파일을 모두 허용합니다. 모든 컬렉션과 패턴은 `collection/<collection name|pattern>` 형식을 취해야 합니다.

> **Note**
>
> `dashboards` 리소스 유형은 OpenSearch Dashboards에 대한 권한만 허용하지만 OpenSearch Dashboards가 작동하려면 동일한 소스의 컬렉션 액세스도 허용해야 합니다. 아래 두 번째 정책을 참조하세요.

프라이빗 액세스를 지정하려면 다음 요소 중 하나 또는 둘 다를 포함합니다.

- `SourceVPCEs` - 하나 이상의 OpenSearch Serverless 관리형 VPC 엔드포인트를 지정합니다.
- `SourceServices` - 지원되는 하나 이상의 식별자를 지정합니다 AWS 서비스. 현재 다음 서비스 식별자가 지원됩니다.
  - `bedrock.amazonaws.com` - Amazon Bedrock

다음 샘플 네트워크 정책은 접두사 `log*`로 시작하는 컬렉션에만 컬렉션 엔드포인트(VPC 액세스 및 Amazon Bedrock)에 대한 프라이빗 액세스 권한에 제공합니다. 인증된 사용자는 OpenSearch Dashboards에 로그인할 수 없으며 프로그래밍 방식으로만 컬렉션 엔드포인트에 액세스할 수 있습니다.

```
[
   {
      "Description":"Private access for log collections",
      "Rules":[
```

```
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/log*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   }
]
```

다음 정책은 finance라는 이름이 지정된 단일 컬렉션에 대해 OpenSearch 엔드포인트 및 OpenSearch Dashboards에 대한 퍼블릭 액세스를 제공합니다. 컬렉션이 존재하지 않는 경우 컬렉션이 생성되면 네트워크 설정이 컬렉션에 적용됩니다.

```
[
   {
      "Description":"Public access for finance collection",
      "Rules":[
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/finance"
            ]
         },
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

다음 요청은 위의 네트워크 정책을 생성합니다.

```
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}]"
```

JSON 파일로 정책을 제공하려면 `--policy file://my-policy.json` 형식을 사용합니다.

네트워크 정책 보기

컬렉션을 생성하기 전에 계정의 기존 네트워크 정책을 미리 보고 컬렉션 이름과 일치하는 리소스 패턴을 가진 정책을 확인하는 것이 좋습니다. 다음 ListSecurityPolicies 요청은 계정의 모든 네트워크 정책을 나열합니다.

```
aws opensearchserverless list-security-policies --type network
```

요청은 구성된 모든 네트워크 정책에 대한 정보를 반환합니다. 특정 정책에 정의된 패턴 규칙을 보려면 응답의 `securityPolicySummaries` 요소 내용에서 정책 정보를 찾으십시오. 이 `name` 및 `type`를 기록하고 GetSecurityPolicy 요청에서 이러한 속성을 사용하여 다음 정책 세부 정보가 포함된 응답을 수신하세요.

```
{
    "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
        }
    ]
}
```

특정 정책에 대한 자세한 정보를 보려면 GetSecurityPolicy 명령을 사용합니다.

네트워크 정책 업데이트

네트워크에 대한 VPC 엔드포인트 또는 퍼블릭 액세스 지정을 수정하면 연결된 모든 컬렉션이 영향을 받습니다. OpenSearch Serverless 콘솔에서 네트워크 정책을 업데이트하려면 Network policies(네트워크 정책)를 확장하고 수정할 정책을 선택한 다음 Edit(편집)를 선택합니다. 변경하고 Save(저장)를 선택합니다.

OpenSearch Serverless API를 사용하여 네트워크 정책을 업데이트하려면 UpdateSecurityPolicy 명령을 사용합니다. 요청에 정책 버전을 포함해야 합니다. `ListSecurityPolicies` 또는`GetSecurityPolicy` 명령을 사용하여 정책 버전을 검색할 수 있습니다. 최신 정책 버전을 포함하면 다른 사람이 변경한 내용을 실수로 재정의하지 않습니다.

다음 요청은 새 정책 JSON 문서로 네트워크 정책을 업데이트합니다.

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type network \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

네트워크 정책 삭제

네트워크 정책을 삭제하려면 먼저 네트워크 정책을 모든 컬렉션에서 분리해야 합니다. OpenSearch Serverless 콘솔에서 정책을 삭제하려면 정책을 선택하고 Delete(삭제)를 선택합니다.

DeleteSecurityPolicy 명령을 사용할 수도 있습니다.

```
aws opensearchserverless delete-security-policy --name my-policy --type network
```

Amazon OpenSearch Serverless를 위한 데이터 액세스 제어

Amazon OpenSearch Serverless의 데이터 액세스 제어를 사용하면 액세스 메커니즘이나 네트워크 소스와 관계없이 사용자가 컬렉션 및 인덱스에 액세스하도록 허용할 수 있습니다. IAM 역할 및 SAML ID에 대한 액세스를 제공할 수 있습니다.

컬렉션 및 인덱스 리소스에 적용되는 데이터 액세스 정책을 통해 액세스 권한을 관리합니다. 데이터 액세스 정책을 사용하면 특정 패턴과 일치하는 컬렉션 및 인덱스에 액세스 권한을 자동으로 할당하여 컬렉션을 대규모로 관리하는 데 도움이 됩니다. 단일 리소스에 여러 데이터 액세스 정책을 적용할 수 있습니다. 단, OpenSearch Dashboards URL에 액세스하려면 컬렉션에 대한 데이터 액세스 정책이 있어야 합니다.

주제

- 데이터 액세스 정책 대 IAM 정책
- 데이터 액세스 정책을 구성하는 데 필요한 IAM 권한
- 정책 구문
- 지원되는 정책 권한
- OpenSearch Dashboards의 샘플 데이터 세트
- 데이터 액세스 정책 생성(콘솔)
- 데이터 액세스 정책 생성(AWS CLI)
- 데이터 액세스 정책 보기
- 데이터 액세스 정책 업데이트
- 데이터 액세스 정책 삭제
- 교차 계정 데이터 액세스

데이터 액세스 정책 대 IAM 정책

데이터 액세스 정책은 AWS Identity and Access Management (IAM) 정책과 논리적으로 구분됩니다. IAM 권한은 `CreateCollection` 및 `ListAccessPolicies`와 같은 서버리스 API 작업에 대한 액세스를 제어합니다. 데이터 액세스 정책은 OpenSearch Serverless가 지원하는 OpenSearch 작업(예: `PUT <index>` 또는 `GET _cat/indices`)에 대한 액세스를 제어합니다.

`aoss:CreateAccessPolicy` 및 `aoss:GetAccessPolicy`(다음 섹션에서 설명)와 같은 데이터 액세스 정책 API 작업에 대한 액세스를 제어하는 IAM 권한은 데이터 액세스 정책에 지정된 권한에 영향을 미치지 않습니다.

예를 들어 IAM 정책이 사용자의 `collection-a`에 대한 데이터 액세스 정책 생성을 거부하지만 모든 컬렉션(`*`)에 대한 데이터 액세스 정책을 생성할 수 있도록 허용한다고 가정해 보겠습니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "aoss:CreateAccessPolicy"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "collection-a"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

사용자가 모든 컬렉션(`collection/*` 또는 `index/*/*`)에 특정 권한을 허용하는 데이터 액세스 정책을 생성하면 해당 정책은 컬렉션 A를 포함한 모든 컬렉션에 적용됩니다.

> **Important**
>
> 데이터 액세스 정책 내에서 권한을 부여하는 것만으로는 OpenSearch Serverless 컬렉션의 데이터에 액세스하는 데 충분하지 않습니다. 또한 관련 보안 주체에게 IAM 권한 `aoss:APIAccessAll` 및 `aoss:DashboardsAccessAll`에 대한 액세스 권한을 부여해야 합니다. 두 권한 모두 컬렉션 리소스에 대한 전체 액세스 권한을 부여하는 반면, 대시보드 권한은 OpenSearch 대시보드 액세스 권한도 제공합니다. 보안 주체에게 이 두 IAM 권한이 모두 있지 않으면 컬렉션에 대한 요청을 보낼 때 403 오류가 발생합니다. 자세한 내용은 the section called “OpenSearch API 작업 사용” 단원을 참조하십시오.

데이터 액세스 정책을 구성하는 데 필요한 IAM 권한

OpenSearch Serverless의 데이터 액세스 제어는 다음과 같은 IAM 권한을 사용합니다. 사용자를 특정 액세스 정책 이름으로 제한하도록 IAM 조건을 지정할 수 있습니다.

- `aoss:CreateAccessPolicy` – 액세스 정책을 생성합니다.
- `aoss:ListAccessPolicies` – 모든 액세스 정책을 나열합니다.

- `aoss:GetAccessPolicy` – 특정 액세스 정책에 대한 세부 정보를 봅니다.
- `aoss:UpdateAccessPolicy` – 액세스 정책을 수정합니다.
- `aoss:DeleteAccessPolicy` – 액세스 정책을 삭제합니다.

다음 자격 증명 기반 액세스 정책은 사용자가 리소스 패턴 `collection/logs`를 포함하는 모든 액세스 정책 및 업데이트 정책을 볼 수 있도록 허용합니다.

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**
>
> 또한 OpenSearch Serverless에는 컬렉션 리소스에 대한 `aoss:APIAccessAll` 및 `aoss:DashboardsAccessAll` 권한이 필요합니다. 자세한 내용은 the section called "OpenSearch API 작업 사용" 단원을 참조하십시오.

정책 구문

데이터 액세스 정책에는 규칙 세트가 포함되어 있으며 각 규칙에는 다음 요소가 포함되어 있습니다.

Element	설명
`ResourceType`	권한이 적용되는 리소스 유형(컬렉션 또는 인덱스)입니다. 별칭 및 템플릿 권한은 컬렉션 수준에 있고 데이터 생성, 수정, 검색 권한은 인덱스 수준에 있습니다. 자세한 내용은 지원되는 정책 권한을 참조하세요.
`Resource`	리소스 이름 및/또는 패턴 목록. 패턴은 와일드카드(*)가 뒤따르는 접두사로 연결된 권한을 여러 리소스에 적용할 수 있도록 합니다. • 컬렉션은 `collection/` *`<name|pattern>`* 형식을 취합니다. • 인덱스는 `index/`*`<collection-name|pattern>`* `/`*`<index-name|pattern/>`* 형식을 취합니다.
`Permission`	지정된 리소스에 대해 부여할 권한 목록입니다. 권한 및 허용되는 API 작업의 전체 목록은 the section called "지원되는 OpenSearch API 작업 및 권한" 섹션을 참조하세요.
`Principal`	액세스 권한을 부여할 하나 이상의 보안 주체 목록입니다. 보안 주체는 IAM 역할 ARN 또는 SAML ID일 수 있습니다. 이러한 보안 주체는 현재 AWS 계정내에 있어야 합니다. 데이터 액세스 정책은 교차 계정 액세스를 직접 지원하지 않지만 다른의 사용자가 컬렉션 소유 계정에서 수임할 수 있는 역할을 정책에 포함할 AWS 계정 수 있습니다. 자세한 내용은 the section called "교차 계정 데이터 액세스" 단원을 참조하십시오.

다음 예시 정책은 autopartsinventory라는 컬렉션과 접두사 sales*로 시작하는 모든 컬렉션에 별칭 및 템플릿 권한을 부여합니다. 또한 autopartsinventory 컬렉션 내의 모든 인덱스와 접두사 orders*로 시작하는 salesorders 컬렉션의 모든 인덱스에 대한 읽기 및 쓰기 권한을 부여합니다.

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

정책 내에서는 액세스를 명시적으로 거부할 수 없습니다. 따라서 모든 정책 권한은 가산적입니다. 예를 들어 한 정책에서 사용자에게 aoss:ReadDocument 권한을 부여하고 다른 정책에서

aoss:WriteDocument 권한을 부여하면 사용자는 두 권한을 모두 가지게 됩니다. 세 번째 정책에서 동일한 사용자에게 aoss:* 권한을 부여하면 사용자는 연결된 인덱스에서 모든 작업을 수행할 수 있습니다. 더 제한적인 권한이 덜 제한적인 권한보다 우선하지는 않습니다.

지원되는 정책 권한

데이터 액세스 정책에서 지원되는 권한은 다음과 같습니다. 각 권한에서 허용하는 OpenSearch API 작업에 대한 내용은 the section called “지원되는 OpenSearch API 작업 및 권한”을 참조하세요.

컬렉션 권한

- aoss:CreateCollectionItems
- aoss:DeleteCollectionItems
- aoss:UpdateCollectionItems
- aoss:DescribeCollectionItems
- aoss:*

인덱스 권한

- aoss:ReadDocument
- aoss:WriteDocument
- aoss:CreateIndex
- aoss:DeleteIndex
- aoss:UpdateIndex
- aoss:DescribeIndex
- aoss:*

OpenSearch Dashboards의 샘플 데이터 세트

OpenSearch Dashboards는 데이터를 추가하기 전에 Dashboards를 탐색하는 데 도움이 되는 시각화, 대시보드 및 기타 도구와 더불어 샘플 데이터 세트를 제공합니다. 이 샘플 데이터로 인덱스를 만들려면 작업하려는 데이터 세트에 권한을 부여하는 데이터 액세스 정책이 필요합니다. 다음 정책은 와일드카드(*)를 사용하여 세 샘플 데이터 세트 모두에 권한을 부여합니다.

```
[
  {
```

```
    "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>"
    ]
  }
]
```

데이터 액세스 정책 생성(콘솔)

시각적 편집기를 사용하거나 JSON 형식으로 데이터 액세스 정책을 생성할 수 있습니다. 정책에 정의된 패턴 중 하나와 일치하는 모든 새 컬렉션에는 컬렉션을 생성할 때 해당 권한이 할당됩니다.

OpenSearch Serverless 데이터 액세스 정책 생성하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버리스를 확장하고 보안에서 데이터 액세스 정책을 선택합니다.
3. Create access policy(액세스 정책 생성)를 선택합니다.
4. 정책의 이름 및 설명을 입력합니다.
5. 정책의 첫 번째 규칙 이름을 입력합니다. 예: “로그 컬렉션 액세스”.
6. Add principals(보안 주체 추가)를 선택하고 데이터 액세스를 제공할 하나 이상의 IAM 역할 또는 SAML 사용자 및 그룹을 선택합니다.

> **Note**
>
> 드롭다운 메뉴에서 보안 주체를 선택하려면 `iam:ListUsers` 및 `iam:ListRoles` 권한(IAM 보안 주체의 경우)과 `aoss:ListSecurityConfigs` 권한(SAML 자격 증명의 경우)이 있어야 합니다.

7. Grant(부여)를 선택하고 별칭, 템플릿, 인덱스 권한을 선택하여 연관된 보안 주체에게 부여합니다. 전체 권한 및 해당 목록에서 허용되는 액세스는 the section called “지원되는 OpenSearch API 작업 및 권한” 섹션을 참조하세요.
8. (선택 사항) 정책에 대한 추가 규칙을 구성합니다.
9. 생성(Create)을 선택합니다. 정책을 만든 시점과 권한이 적용된 시점 사이에 약 1분의 지연 시간이 있을 수 있습니다. 5분 넘게 소요될 경우 지원에 문의하세요.

> **Important**
>
> 정책에 인덱스 권한만 포함되어 있고 컬렉션 권한은 없는 경우 `Collection cannot be accessed yet. Configure data access policies so that users can access the data within this collection`이라는 일치하는 컬렉션에 대한 메시지가 계속 표시될 수 있습니다. 이 경고는 무시해도 됩니다. 허용된 보안 주체는 여전히 컬렉션에서 할당된 인덱스 관련 작업을 수행할 수 있습니다.

데이터 액세스 정책 생성(AWS CLI)

OpenSearch Serverless API를 사용하여 데이터 액세스 정책을 생성하려면 `CreateAccessPolicy` 명령을 사용합니다. 이 명령은 인라인 정책과 .json 파일을 모두 허용합니다. 인라인 정책은 JSON 이스케이프 문자열로 인코딩해야 합니다.

다음 요청은 데이터 액세스 정책을 생성합니다.

```
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\"]}]"
```

.json 파일 내에 정책을 제공하려면 `--policy file://my-policy.json` 형식을 사용합니다.

정책에 포함된 보안 주체는 이제 액세스 권한이 부여된 OpenSearch 작업을 사용할 수 있습니다.

데이터 액세스 정책 보기

컬렉션을 생성하기 전에 계정의 기존 데이터 액세스 정책을 미리 보고 컬렉션 이름과 일치하는 리소스 패턴을 가진 정책을 확인하는 것이 좋습니다. 다음 ListAccessPolicies 요청은 계정의 모든 데이터 액세스 정책을 나열합니다.

```
aws opensearchserverless list-access-policies --type data
```

요청은 구성된 모든 데이터 액세스 정책에 대한 정보를 반환합니다. 특정 정책에 정의된 패턴 규칙을 보려면 응답의 `accessPolicySummaries` 요소 내용에서 정책 정보를 찾으십시오. 이 정책의 `name` 및 `type`를 기록하고 GetAccessPolicy 요청에서 이러한 속성을 사용하여 다음 정책 세부 정보가 포함된 응답을 수신하세요.

```
{
    "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
        }
    ]
}
```

리소스 필터를 포함하여 결과를 특정 컬렉션 또는 인덱스가 포함된 정책으로 제한할 수 있습니다.

```
aws opensearchserverless list-access-policies --type data --resource
 "index/autopartsinventory/*"
```

특정 정책에 대한 세부 정보를 보려면 GetAccessPolicy 명령을 사용합니다.

데이터 액세스 정책 업데이트

데이터 액세스 정책을 업데이트하면 모든 관련 컬렉션이 영향을 받습니다. OpenSearch Serverless 콘솔에서 데이터 액세스 정책을 업데이트하려면 Data access control(데이터 액세스 제어)을 선택하고 수정할 정책을 선택한 다음 Edit(편집)를 선택합니다. 변경하고 Save(저장)를 선택합니다.

OpenSearch Serverless API를 사용하여 데이터 액세스 정책을 업데이트하려면 `UpdateAccessPolicy` 요청을 전송하세요. `ListAccessPolicies` 또는 `GetAccessPolicy` 명령을 사용하여 검색할 수 있는 정책 버전을 포함해야 합니다. 최신 정책 버전을 포함하면 다른 사람이 변경한 내용을 실수로 재정의하지 않습니다.

다음 UpdateAccessPolicy 요청은 새 정책 JSON 문서로 데이터 액세스 정책을 업데이트합니다.

```
aws opensearchserverless update-access-policy \
    --name sales-inventory \
    --type data \
    --policy-version MTY2NDA1NDE4MDg10F8x \
    --policy file://my-new-policy.json
```

정책을 업데이트하는 시점과 새 권한이 적용되는 시점 사이에 몇 분의 지연 시간이 있을 수 있습니다.

데이터 액세스 정책 삭제

데이터 액세스 정책을 삭제하면 연결된 모든 컬렉션이 정책에 정의된 액세스 권한을 잃게 됩니다. 정책을 삭제하기 전에 IAM 및 SAML 사용자에게 컬렉션에 대한 적절한 액세스 권한이 있는지 확인하세요. OpenSearch Serverless 콘솔에서 정책을 삭제하려면 정책을 선택하고 Delete(삭제)를 선택합니다.

DeleteAccessPolicy 명령을 사용할 수도 있습니다.

```
aws opensearchserverless delete-access-policy --name my-policy --type data
```

교차 계정 데이터 액세스

교차 계정 자격 증명 또는 교차 계정 컬렉션으로 데이터 액세스 정책을 생성할 수는 없지만 여전히 수임 역할 옵션을 사용하여 교차 계정 액세스를 설정할 수 있습니다. 예를 들어 *account-b*에서 액세스해야 하는 컬렉션을 *account-a*에서 소유한 경우 *account-b*의 사용자가 *account-a*에서 역할을 수임할 수 있습니다. 역할은 IAM 권한 `aoss:APIAccessAll` 및 `aoss:DashboardsAccessAll`을 보유해야 하며 *account-a*의 데이터 액세스 정책에 포함되어야 합니다.

인터페이스 엔드포인트(AWS PrivateLink)를 사용하여 Amazon OpenSearch Serverless에 액세스

AWS PrivateLink 를 사용하여 VPC와 Amazon OpenSearch Serverless 간에 프라이빗 연결을 생성할 수 있습니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 OpenSearch Serverless에 액세스할 수 있습니다. VPC의 인스턴스는 OpenSearch Serverless에 액세스하기 위해 퍼블릭 IP 주소가 필요하지 않습니다. VPC 네트워크 액세스에 대한 자세한 내용은 Amazon OpenSearch Serverless의 네트워크 연결 패턴을 참조하세요.

AWS PrivateLink에서 제공되는 인터페이스 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 엔드포인트에 대해 지정하는 각 서브넷에 엔드포인트 네트워크 인터페이스를 생성합니다. 이는 OpenSearch Serverless로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

자세한 내용은AWS PrivateLink 가이드의 AWS PrivateLink를 통해 AWS 서비스 에 액세스를 참조하세요.

주제

- 수집 엔드포인트의 DNS 해결
- VPC 및 네트워크 액세스 정책
- VPC 및 엔드포인트 정책
- 고려 사항
- 필요한 권한
- OpenSearch Serverless용 인터페이스 엔드포인트 생성
- Amazon OpenSearch Serverless에 대한 공유 VPC 설정

수집 엔드포인트의 DNS 해결

VPC 엔드포인트를 생성하면 서비스가 새 Amazon Route 53 프라이빗 호스팅 영역을 생성하여 VPC에 연결합니다. 이 프라이빗 호스팅 영역은 OpenSearch Serverless 컬렉션(`*.aoss.us-east-1.amazonaws.com`)의 와일드카드 DNS 레코드를 엔드포인트에 사용되는 인터페이스 주소로 확인하기 위한 레코드로 구성됩니다. VPC에 OpenSearch Serverless VPC 엔드포인트 하나만 있으면 각 AWS 리전에 있는 모든 컬렉션과 Dashboards에 액세스할 수 있습니다. OpenSearch Serverless용 엔드포인트가 있는 모든 VPC에는 자체 프라이빗 호스팅 영역이 연결되어 있습니다.

또한 OpenSearch Serverless는 해당 리전의 모든 컬렉션에 대해 퍼블릭 Route 53 와일드카드 DNS 레코드를 생성합니다. DNS 이름은 OpenSearch Serverless 퍼블릭 IP 주소로 확인됩니다. OpenSearch

Serverless VPC 엔드포인트가 없는 VPC의 클라이언트 또는 공용 네트워크의 클라이언트는 퍼블릭 Route 53 해석기를 사용하고 해당 IP 주소로 컬렉션 및 Dashboards에 액세스할 수 있습니다. VPC 엔드포인트의 IP 주소 유형(IPv4, IPv6 또는 이중 스택)은 OpenSearch Serverless용 인터페이스 엔드포인트를 생성할 때 제공된 서브넷을 기반으로 결정됩니다.

> **Note**
>
> OpenSearch Serverless는 OpenSearch Service 도메인 확인을 위해 추가 Amazon Route 53 프라이빗 호스팅 영역(`<region>.opensearch.amazonaws.com`)을 생성합니다. AWS CLI의 update-vpc-endpoint 명령을 사용하여 기존 IPv4 VPC 엔드포인트를 이중 스택으로 업데이트할 수 있습니다.

특정 VPC의 DNS 해석기 주소는 VPC CIDR의 두 번째 IP 주소입니다. VPC의 모든 클라이언트는 해당 해석기를 사용하여 모든 컬렉션의 VPC 엔드포인트 주소를 가져와야 합니다. 해석기는 OpenSearch Serverless에서 만든 프라이빗 호스팅 영역을 사용합니다. 어떤 계정에서든 모든 컬렉션에 이 해석기를 사용하면 충분합니다. 일반적으로 필요하지는 않지만 일부 컬렉션 엔드포인트에는 VPC 해석기를 사용하고 다른 콜렉션 엔드포인트에는 퍼블릭 해석기를 사용할 수도 있습니다.

VPC 및 네트워크 액세스 정책

컬렉션의 OpenSearch API 및 Dashboards에 네트워크 권한을 부여하려면 OpenSearch Serverless 네트워크 액세스 정책을 사용할 수 있습니다. VPC 엔드포인트 또는 공용 인터넷에서 이 네트워크 액세스를 제어할 수 있습니다. 네트워크 정책은 트래픽 권한만 제어하므로 컬렉션 및 해당 인덱스의 데이터에 대한 운영 권한을 지정하는 데이터 액세스 정책도 설정해야 합니다. OpenSearch Serverless VPC 엔드포인트를 서비스에 대한 액세스 포인트로, 네트워크 액세스 정책을 컬렉션 및 Dashboards에 대한 네트워크 수준의 액세스 포인트로, 데이터 액세스 정책을 컬렉션의 데이터에 대한 모든 작업에 대해 세밀한 액세스 제어를 위한 액세스 포인트로 생각하세요.

네트워크 정책에서 여러 VPC 엔드포인트 ID를 지정할 수 있으므로 컬렉션에 액세스해야 하는 모든 VPC에 대해 VPC 엔드포인트를 만드는 것이 좋습니다. 이러한 VPCs OpenSearch Serverless 컬렉션 및 네트워크 정책을 소유한 AWS 계정과 다른 계정에 속할 수 있습니다. 한 계정의 VPC가 다른 계정의 VPC 엔드포인트를 사용할 수 있도록 두 계정 간에 VPC-VPC 피어링 또는 기타 프록시 솔루션을 생성하지 않는 것이 좋습니다. 이는 자체 엔드포인트가 있는 각 VPC보다 보안 및 비용 효율성이 떨어집니다. 네트워크 정책에서 해당 VPC의 엔드포인트에 대한 액세스 권한을 설정한 다른 VPC의 관리자는 첫 번째 VPC를 쉽게 볼 수 없습니다.

VPC 및 엔드포인트 정책

Amazon OpenSearch Serverless는 VPC에 대한 엔드포인트 정책을 지원하지 않습니다. 엔드포인트 정책은 엔드포인트를 사용하여 AWS 서비스에 액세스할 수 있는 보안 주체를 제어 AWS 하기 위해 VPC 엔드포인트에 연결하는 IAM 리소스 기반 정책입니다. 자세한 정보는 엔드포인트 정책을 사용하여 VPC 엔드포인트에 대한 액세스 제어를 참조하세요.

엔드포인트 정책을 사용하려면 먼저 인터페이스 엔드포인트를 생성해야 합니다. OpenSearch Serverless 콘솔 또는 OpenSearch Serverless API를 사용하여 인터페이스 엔드포인트를 생성할 수 있습니다. 인터페이스 엔드포인트를 생성한 후에는 엔드포인트에 엔드포인트 정책을 추가해야 합니다. 자세한 내용은 인터페이스 엔드포인트(AWS PrivateLink)를 사용하여 Amazon OpenSearch Serverless에 액세스를 참조하세요.

> **Note**
>
> OpenSearch Service 콘솔에서 직접 엔드포인트 정책을 정의할 수는 없습니다.

엔드포인트 정책은 사용자가 구성한 다른 자격 증명 기반 정책, 리소스 기반 정책, 네트워크 정책 또는 데이터 액세스 정책을 재정의하거나 대체하지 않습니다. 엔드포인트 정책 업데이트에 대한 자세한 내용은 엔드포인트 정책을 사용하여 VPC 엔드포인트에 대한 액세스 제어를 참조하세요.

기본적으로 엔드포인트 정책은 VPC 엔드포인트에 대한 전체 액세스 권한을 부여합니다.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

기본 VPC 엔드포인트 정책이 전체 엔드포인트 액세스 권한을 부여하지만 특정 역할 및 사용자에게만 액세스를 허용하도록 VPC 엔드포인트 정책을 구성할 수도 있습니다. 이렇게 하려면 다음 예제를 참조하세요.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012",
                    "987654321098"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

OpenSearch Serverless 컬렉션을 VPC 엔드포인트 정책에 조건부 요소로 포함하도록 지정할 수 있습니다. 이렇게 하려면 다음 예제를 참조하세요.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": [
                        "coll-abc"
                    ]
                }
            }
        }
```

```
    ]
}
```

에 대한 지원이 지원aoss:CollectionId됩니다.

```
Condition": {
         "StringEquals": {
               "aoss:CollectionId": "collection-id"
          }
}
```

VPC 엔드포인트 정책의 SAML 자격 증명을 사용하여 VPC 엔드포인트 액세스를 결정할 수 있습니다. VPC 엔드포인트 정책의 보안 주체 섹션에서 (*) 와일드카드를 사용해야 합니다. 이렇게 하려면 다음 예제를 참조하세요.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "saml:Groups": [
                        "saml/111122223333/idp123/group/football",
                        "saml/111122223333/idp123/group/soccer",
                        "saml/111122223333/idp123/group/cricket"
                    ]
                }
            }
        }
    ]
}
```

또한 특정 SAML 보안 주체 정책을 포함하도록 엔드포인트 정책을 구성할 수 있습니다. 이렇게 하려면 다음을 참조하세요.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalTag/Department": [
                        "Engineering"]
                    }
                }
            }
        ]
    }
```

Amazon OpenSearch Serverless에서 SAML 인증을 사용하는 방법에 대한 자세한 내용은 Amazon OpenSearch Serverless에 대한 SAML 인증을 참조하세요.

한 VPC 엔드포인트 정책에 IAM 사용자와 SAML 사용자를 모두 포함할 수 있습니다. 이렇게 하려면 다음 예제를 참조하세요.

JSON

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

```
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "saml:groups": [
                        "saml//idp123/group/football",
                        "saml//idp123/group/soccer",
                        "saml//idp123/group/cricket"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    ""
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

인터페이스 VPC 엔드포인트를 통해 Amazon EC2에서 Amazon OpenSearch Serverless 컬렉션에 액세스할 수도 있습니다. 자세한 내용은 (인터페이스 VPC 엔드포인트를 통해) Amazon EC2에서 OpenSearch Serverless 컬렉션에 액세스를 참조하세요.

고려 사항

OpenSearch Serverless에 대한 인터페이스 엔드포인트를 설정하기 전에 다음을 고려하세요.

- OpenSearch Serverless는 인터페이스 엔드포인트를 통해 지원되는 모든 OpenSearch API 작업(구성 API 작업 아님)에 대한 호출을 지원합니다.
- OpenSearch Serverless용 인터페이스 엔드포인트를 생성한 후에도 서버리스 컬렉션에 액세스하려면 이를 네트워크 액세스 정책 포함시켜야 합니다.
- 기본적으로 OpenSearch Serverless에 대한 전체 액세스는 인터페이스 엔드포인트를 통해 허용됩니다. 보안 그룹을 엔드포인트 네트워크 인터페이스와 연결하여 인터페이스 엔드포인트를 통해 OpenSearch 서버리스에 대한 트래픽을 제어할 수 있습니다.
- 단일 에는 최대 50개의 OpenSearch Serverless VPC 엔드포인트가 있을 AWS 계정 수 있습니다.

- 네트워크 정책에서 컬렉션의 API 또는 Dashboards에 대한 공용 인터넷 액세스를 활성화하면 모든 VPC와 공용 인터넷을 통해 컬렉션에 액세스할 수 있습니다.
- 온프레미스로 VPC 외부에 있는 경우 OpenSearch Serverless VPC 엔드포인트 확인에 DNS 해석기를 직접 사용할 수 없습니다. VPN 액세스가 필요한 경우 VPC에 외부 클라이언트가 사용할 DNS 프록시 해석기가 필요합니다. Route 53은 온프레미스 네트워크나 다른 VPC에서 사용자의 VPC로 DNS 쿼리를 보낼 때 사용할 수 있는 인바운드 엔드포인트 옵션을 제공합니다.
- OpenSearch Serverless가 생성하고 VPC에 연결하는 프라이빗 호스팅 영역은 서비스에서 관리하지만 Amazon Route 53 리소스에 표시되며 계정에 요금이 청구됩니다.
- 기타 고려 사항은 AWS PrivateLink 가이드의 고려 사항을 참조하세요.

필요한 권한

OpenSearch Serverless에 대한 VPC 액세스는 다음 AWS Identity and Access Management (IAM) 권한을 사용합니다. 사용자를 특정 컬렉션으로 제한하도록 IAM 조건을 지정할 수 있습니다.

- `aoss:CreateVpcEndpoint` – VPC 엔드포인트를 생성합니다.
- `aoss:ListVpcEndpoints` – 모든 VPC 엔드포인트를 나열합니다.
- `aoss:BatchGetVpcEndpoint` – VPC 엔드포인트의 하위 집합에 대한 세부 정보를 봅니다.
- `aoss:UpdateVpcEndpoint` – VPC 엔드포인트를 수정합니다.
- `aoss:DeleteVpcEndpoint` – VPC 엔드포인트를 삭제합니다.

또한 VPC 엔드포인트를 생성하려면 다음과 같은 Amazon EC2 및 Route 53 권한이 필요합니다.

- `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`

OpenSearch Serverless용 인터페이스 엔드포인트 생성

콘솔 또는 OpenSearch Serverless API를 사용하여 OpenSearch Serverless용 인터페이스 엔드포인트를 생성할 수 있습니다.

OpenSearch Serverless 컬렉션용 인터페이스 엔드포인트 생성하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Serverless(서버리스)를 확장하고 VPC endpoints(VPC 엔드포인트)를 선택합니다.
3. Create VPC endpoint(VPC 엔드포인트 생성)를 선택합니다.
4. 엔드포인트의 이름을 입력합니다.
5. VPC의 경우 OpenSearch Serverless에 액세스할 VPC를 선택합니다.
6. Subnets(서브넷)의 경우 OpenSearch Serverless에 액세스할 하나의 서브넷을 선택합니다.
   - 엔드포인트의 IP 주소 및 DNS 유형이 서브넷 유형을 기반으로 하는 경우
     - 이중 스택: 모든 서브넷에 IPv4 및 IPv6 주소 범위가 모두 있는 경우
     - IPv6: 모든 서브넷이 IPv6 전용 서브넷인 경우
     - IPv4: 모든 서브넷이 IPv4 주소 범위를 포함하는 경우
7. Security group(보안 그룹)의 경우 엔드포인트 네트워크 인터페이스에 연결할 보안 그룹을 선택합니다. 이것은 엔드포인트로 승인하는 인바운드 트래픽의 포트, 프로토콜, 소스를 제한하는 중요한 단계입니다. 보안 그룹 규칙이 VPC 엔드포인트를 사용하여 OpenSearch Serverless와 통신할 리소스가 엔드포인트 네트워크 인터페이스와 통신하도록 허용하는지 확인합니다.
8. Create endpoint(엔드포인트 생성)을 선택합니다.

OpenSearch Serverless API를 사용하여 VPC 엔드포인트를 생성하려면 `CreateVpcEndpoint` 명령을 사용합니다.

> **Note**
>
> 엔드포인트를 생성한 후에는 해당 ID를 기록해 둡니다(예: `vpce-abc123def4EXAMPLE`). 컬렉션에 대한 엔드포인트 액세스를 제공하려면 하나 이상의 네트워크 액세스 정책에 이 ID를 포함해야 합니다.

인터페이스 엔드포인트를 생성한 후에는 네트워크 액세스 정책을 통해 컬렉션에 대한 액세스를 제공해야 합니다. 자세한 내용은 the section called “네트워크 액세스” 단원을 참조하십시오.

Amazon OpenSearch Serverless에 대한 공유 VPC 설정

Amazon Virtual Private Cloud(VPC)를 사용하여 조직의 다른와 VPC 서브넷을 공유하고 여러 AWS 계정 의 리소스 간에 VPN과 같은 네트워킹 인프라를 공유할 수 있습니다 AWS 계정.

현재 Amazon OpenSearch Serverless는 사용자가 해당 VPC의 소유자가 아닌 한 공유 VPC에 AWS PrivateLink 대한 연결 생성을 지원하지 않습니다. AWS PrivateLink 또한는 간의 연결 공유를 지원하지 않습니다 AWS 계정.

그러나 OpenSearch Serverless의 유연한 모듈식 아키텍처를 기반으로 공유 VPC를 설정할 수 있습니다. 이는 OpenSearch Serverless 네트워킹 인프라가 개별 컬렉션(OpenSearch Service) 인프라와 분리되어 있기 때문입니다. 따라서 AWS PrivateLink VPC가 위치한 한 계정에 대해 VPCe 엔드포인트를 생성한 다음 다른 계정의 네트워크 정책에서 VPCe ID를 사용하여 해당 공유 VPC에서만 오는 트래픽을 제한할 수 있습니다.

다음 절차는 소유자 계정과 소비자 계정을 참조합니다.

소유자 계정은 VPC를 설정하고 다른 계정과 공유하는 공통 네트워킹 계정 역할을 합니다. 소비자 계정은 소유자 계정에서 공유한 VPC에서 OpenSearch Serverless 컬렉션을 생성하고 유지 관리하는 계정입니다.

사전 조건

공유 VPC를 설정하기 전에 다음 요구 사항이 충족되는지 확인합니다.

- 의도된 소유자 계정은 Amazon Virtual Private Cloud에서 VPC, 서브넷, 라우팅 테이블 및 기타 필수 리소스를 이미 설정했어야 합니다. 자세한 내용은 Amazon VPC 사용 설명서를 참조하세요.

- 의도한 소유자 계정과 소비자 계정은 동일한 조직에 속해야 합니다 AWS Organizations. 자세한 내용은 AWS Organizations 사용 설명서를 참조하십시오.

소유자 계정/공통 네트워킹 계정에서 공유 VPC를 설정하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. the section called “인터페이스 엔드포인트 생성” 섹션의 단계를 따르세요. 다음과 같이 선택합니다.
   - 조직의 소비자 계정과 공유되는 VPC 및 서브넷을 선택합니다.
3. 엔드포인트를 생성한 후 생성된 VPCe ID를 기록하여 소비자 계정에서 설정 작업을 수행할 관리자에게 제공합니다.

   VPCe IDs 형식입니다`vpce-abc123def4EXAMPLE`.

소비자 계정에서 공유 VPC를 설정하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 컬렉션이 아직 없는 경우의 정보를 사용하여 컬렉션을 the section called “컬렉션 관리” 생성합니다.
3. 의 정보를 사용하여 네트워크 정책을 the section called “네트워크 정책 생성(콘솔)” 생성합니다. 이렇게 할 때 다음과 같이 선택합니다.

   > **Note**
   >
   > 이를 위해 기존 네트워크 정책을 업데이트할 수도 있습니다.

   a. 액세스 유형에서 VPC(권장)를 선택합니다.

   b. 액세스용 VPC 엔드포인트에서 소유자 계정이 제공한 VPCe ID를 형식으로 선택합니다`vpce-abc123def4EXAMPLE`.

   c. 리소스 유형 영역에서 다음을 수행합니다.

      - OpenSearch 엔드포인트에 대한 액세스 활성화 상자를 선택한 다음 공유 VPC에서 액세스를 활성화하는 데 사용할 컬렉션 이름 또는 컬렉션 패턴을 선택합니다.

- OpenSearch Dashboard에 대한 액세스 활성화 상자를 선택한 다음 공유 VPC에서 액세스를 활성화하는 데 사용할 컬렉션 이름 또는 컬렉션 패턴을 선택합니다.

4. 새 정책의 경우 생성을 선택합니다. 기존 정책의 경우 업데이트를 선택합니다.

Amazon OpenSearch Serverless에 대한 SAML 인증

Amazon OpenSearch Serverless에 대한 SAML 인증을 사용하면 기존 자격 증명 공급자를 사용하여 서버리스 컬렉션의 OpenSearch Dashboards에 통합 인증(SSO)을 제공할 수 있습니다.

SAML 인증을 사용하면 타사 ID 공급자를 통해 OpenSearch Dashboards에 로그인하여 데이터를 인덱싱하고 검색할 수 있습니다. OpenSearch 서버리스는 IAM Identity Center, Okta, Keycloak, Active Directory Federation Services(ADFS) 및 Auth0과 같은 SAML 2.0 표준을 사용하는 공급자를 지원합니다. Okta, OneLogin 및 Microsoft Entra ID와 같은 다른 자격 증명 소스의 사용자 및 그룹을 동기화하도록 IAM Identity Center를 구성할 수 있습니다. IAM Identity Center에서 지원하는 자격 증명 소스 목록과 구성 단계는 IAM Identity Center 사용 설명서의 시작하기 자습서를 참조하세요.

> **Note**
>
> SAML 인증은 웹 브라우저를 통해 OpenSearch Dashboards에 액세스하는 용도로만 사용됩니다. 인증된 사용자는 OpenSearch Dashboards의 Dev Tools(개발 도구)를 통해서만 OpenSearch API 작업을 요청할 수 있습니다. SAML 자격 증명으로는 OpenSearch API 작업에 직접 HTTP 요청을 할 수 없습니다.

SAML 인증을 설정하려면 먼저 SAML 자격 증명 공급자(IdP)를 구성합니다. 그런 다음 데이터 액세스 정책에 해당 IdP의 사용자를 하나 이상 포함합니다. 이 정책은 컬렉션 및/또는 인덱스에 특정 권한을 부여합니다. 그러면 사용자는 OpenSearch Dashboards에 로그인하여 데이터 액세스 정책에서 허용되는 작업을 수행할 수 있습니다.

주제

- 고려 사항
- 필요한 권한
- SAML 공급자 생성(콘솔)
- OpenSearch 대시보드 액세스
- 컬렉션 데이터에 대한 SAML 자격 증명 액세스 권한 부여

- SAML 공급자 생성(AWS CLI)
- SAML 공급자 보기
- SAML 공급자 업데이트
- SAML 공급자 삭제

고려 사항

SAML 인증을 구성할 때는 다음 사항을 고려하세요.

- 서명 및 암호화된 요청은 지원되지 않습니다.
- 암호화된 어설션은 지원되지 않습니다.
- IdP 시작 인증 및 로그아웃은 지원되지 않습니다.
- 비 IAM 자격 증명(예: Amazon OpenSearch Serverless 및 SAML의 SAML 및 Amazon OpenSearch Service의 기본 내부 사용자 권한 부여)의 경우 서비스 제어 정책(SCP)이 적용되거나 평가되지 않습니다.

필요한 권한

OpenSearch Serverless에 대한 SAML 인증은 다음 AWS Identity and Access Management (IAM) 권한을 사용합니다.

- `aoss:CreateSecurityConfig` – SAML 공급자를 생성합니다.
- `aoss:ListSecurityConfig` – 현재 계정의 모든 SAML 공급자를 나열합니다.
- `aoss:GetSecurityConfig` – SAML 공급자 정보를 봅니다.
- `aoss:UpdateSecurityConfig` – XML 메타데이터를 포함하여 주어진 SAML 공급자 구성을 수정합니다.
- `aoss:DeleteSecurityConfig` – SAML 공급자를 삭제합니다.

다음 자격 증명 기반 액세스 정책을 통해 사용자는 모든 IdP 구성을 관리할 수 있습니다.

JSON

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

```
        {
            "Action": [
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

`Resource` 요소는 와일드카드여야 한다는 점에 유의하세요.

SAML 공급자 생성(콘솔)

이 단계에서는 SAML 공급자를 생성하는 방법을 설명합니다. 이를 통해 OpenSearch Dashboards에 대한 서비스 공급자(SP) 시작 인증을 통한 SAML 인증이 활성화됩니다. IdP 시작 인증은 지원되지 않습니다.

OpenSearch Dashboards에 대해 SAML 인증 활성화하기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 패널에서 Serverless(서버리스)를 확장하고 SAML authentication(SAML 인증)을 선택합니다.
3. Add SAML provider(SAML 공급자 추가)를 선택합니다.
4. 공급자의 이름 및 설명을 입력합니다.

> **Note**
>
> 지정하는 이름은 공개적으로 액세스할 수 있으며 사용자가 OpenSearch Dashboards에 로그인할 때 드롭다운 메뉴에 나타납니다. 이름을 쉽게 알아볼 수 있고 자격 증명 공급자에 대한 민감한 정보가 드러나지 않는지 확인하세요.

5. Configure your IdP(IDP 구성)에서 어설션 소비자 서비스(ACS) URL을 복사합니다.
6. 방금 복사한 ACS URL을 사용하여 자격 증명 공급자를 구성합니다. 용어 및 단계는 공급자마다 다릅니다. 공급자의 설명서를 참조하세요.

예를 들어 Okta에서는 “SAML 2.0 웹 애플리케이션”을 생성하고 ACS URL을 Single Sign On URL, Recipient URL(수신 URL), Destination URL(대상 URL)로 지정합니다. Auth0의 경우 Allowed Callback URLs(허용된 콜백 URL)에서 이를 지정합니다.

7. IdP에 대상 제한 필드가 있는 경우 이를 입력합니다. 대상 제한은 어설션의 대상을 지정하는 SAML 어설션 내의 값입니다. OpenSearch Serverless를 사용하면 다음을 수행할 수 있습니다. 다음 코드 예제의 *####* 자신의 AWS 계정 ID로 바꿔야 합니다.

   1. 기본 대상 제한을 사용합니다`:opensearch:`*`111122223333`*.
   2. (선택 사항)를 사용하여 사용자 지정 대상 제한을 구성합니다 AWS CLI. 자세한 내용은 SAML 공급자 생성(AWS CLI) 단원을 참조하십시오.

   대상 제한 필드의 이름은 공급자마다 다릅니다. Okta의 경우 Audience URI (SP Entity ID)(대상 URI(SP 엔터티 ID))입니다. IAM Identity Center의 경우 Application SAML audience(애플리케이션 SAML 대상)입니다.

8. IAM Identity Center를 사용하는 경우 `unspecified` 형식의 `Subject=${user:name}` 속성 매핑도 지정해야 합니다.

9. 자격 증명 공급자를 구성하면 IdP 메타데이터 파일이 생성됩니다. 이 XML 파일에는 TLS 인증서, 통합 인증 엔드포인트 및 자격 증명 공급자의 엔터티 ID와 같은 공급자에 대한 정보가 들어 있습니다.

   IdP 메타데이터 파일의 텍스트를 복사하여 Provide metadata from your IdP(IdP에서 메타데이터 제공) 필드에 붙여 넣습니다. 또는 XML 파일에서 가져오기(Import from XML file)를 선택하고 파일을 업로드합니다. 메타데이터 파일은 다음과 같아야 합니다.

```
<?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. 사용자 이름에 대한 SAML 어설션의 NameID 요소를 사용하려면 사용자 지정 사용자 ID 속성 필드를 비워 둡니다. 어설션에서 이 표준 요소를 사용하지 않고 사용자 이름을 사용자 지정 속성으로 포함하는 경우 여기에 해당 속성을 지정합니다. 속성은 대소문자를 구분합니다. 단일 사용자 속성만 지원됩니다.

    다음 예시는 SAML 어설션에서 NameID에 대한 재정의 속성을 보여줍니다.

    ```
    <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. (선택 사항) Group attribute(그룹 특성) 필드에 role 또는 group과 같은 사용자 지정 특성을 지정합니다. 단일 그룹 속성만 지원됩니다. 기본 그룹 속성은 없습니다. 지정하지 않는 경우 데이터 액세스 정책에는 사용자 보안 주체만 포함될 수 있습니다.

    다음 예시는 SAML 어설션의 그룹 특성을 보여줍니다.

    ```
    <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. 기본적으로 OpenSearch Dashboards는 24시간 후에 사용자를 로그아웃합니다. OpenSearch 대시보드 제한 시간을 지정하여 이 값을 1~12시간(15~720분) 사이의 숫자로 구성할 수 있습니다. 제한 시간을 15분 이하로 설정하려는 경우 세션이 1시간으로 재설정됩니다.

13. Create SAML provider(SAML 공급자 생성)를 선택합니다.

OpenSearch 대시보드 액세스

SAML 공급자를 구성한 후에는 해당 공급자와 연결된 모든 사용자 및 그룹이 OpenSearch Dashboards 엔드포인트로 이동할 수 있습니다. Dashboards URL에는 모든 컬렉션에 대한 *collection-endpoint*/_dashboards/ 형식이 있습니다.

SAML을 활성화한 경우에서 링크를 선택하면 SAML 자격 증명을 사용하여 로그인할 수 있는 IdP 선택 페이지로 AWS Management Console 이동합니다. 먼저 드롭다운을 사용하여 ID 공급자를 선택합니다.

그런 다음 IdP 보안 인증을 사용하여 로그인합니다.

SAML을 활성화하지 않은 경우에서 링크를 선택하면 SAML에 대한 옵션 없이 IAM 사용자 또는 역할로 로그인하라는 AWS Management Console 메시지가 표시됩니다.

컬렉션 데이터에 대한 SAML 자격 증명 액세스 권한 부여

SAML 공급자를 만든 후에도 기본 사용자 및 그룹에 컬렉션 내 데이터에 대한 액세스 권한을 부여해야 합니다. 데이터 액세스 정책을 통해 액세스 권한을 부여합니다. 사용자에게 액세스 권한을 부여하기까지는 사용자가 컬렉션 내 데이터를 읽거나 쓰거나 삭제할 수 없습니다.

액세스 권한을 부여하려면 데이터 액세스 정책을 생성하고 `Principal` 명령문에 SAML 사용자 및/또는 그룹 ID를 지정합니다.

```
[
   {
      "Rules":[
       ...
      ],
      "Principal":[
         "saml/987654321098/myprovider/user/Shaheen",
         "saml/987654321098/myprovider/group/finance"
      ]
   }
]
```

컬렉션, 인덱스 또는 둘 다에 대한 액세스 권한을 부여할 수 있습니다. 사용자마다 다른 권한을 가지게 하려면 규칙을 여러 개 만듭니다. 사용 가능한 권한 목록은 지원되는 정책 권한을 참조하세요. 액세스 정책의 형식 지정 방법에 대한 자세한 내용은 정책 구문을 참조하세요.

SAML 공급자 생성(AWS CLI)

OpenSearch Serverless API를 사용하여 SAML 공급자를 생성하려면 CreateSecurityConfig 요청을 보냅니다.

```
aws opensearchserverless create-security-config \
    --name myprovider \
    --type saml \
    --saml-options file://saml-auth0.json
```

.json 파일 내의 키-값 맵으로 메타데이터 XML을 포함하여 `saml-options`를 지정합니다. 메타데이터 XML은 JSON 이스케이프 문자열로 인코딩되어야 합니다.

```
{
   "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**
>
> (선택 사항)를 사용하여 사용자 지정 대상 제한을 구성합니다 AWS CLI. 자세한 내용은 SAML 공급자 생성(AWS CLI) 단원을 참조하십시오.

SAML 공급자 보기

다음 ListSecurityConfigs 요청은 계정의 모든 SAML 공급자를 나열합니다.

```
aws opensearchserverless list-security-configs --type saml
```

이 요청은 ID 공급자가 생성하는 전체 IdP 메타데이터를 포함하여 모든 기존 SAML 공급자에 대한 정보를 반환합니다.

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

향후 업데이트를 위한 configVersion을 포함하여 특정 공급자에 대한 세부 정보를 보려면 GetSecurityConfig 요청을 보냅니다.

SAML 공급자 업데이트

OpenSearch Serverless 콘솔을 사용하여 SAML 공급자를 업데이트하려면 SAML authentication(SAML 인증)을 선택하고 자격 증명 공급자를 선택한 다음 Edit(편집)를 선택합니다. 메타데이터 및 사용자 지정 속성을 포함하여 모든 필드를 수정할 수 있습니다.

OpenSearch Serverless API를 통해 공급자를 업데이트하려면 UpdateSecurityConfig 요청을 보내고 업데이트할 정책의 식별자를 포함합니다. ListSecurityConfigs 또는 GetSecurityConfig 명령을 사용하여 검색할 수 있는 구성 버전도 포함해야 합니다. 최신 버전을 포함하면 다른 사람이 변경한 내용을 실수로 재정의하지 않습니다.

다음 요청은 공급자의 SAML 옵션을 업데이트합니다.

```
aws opensearchserverless update-security-config \
    --id saml/123456789012/myprovider \
    --type saml \
    --saml-options file://saml-auth0.json \
    --config-version MTY2NDA1MjY4NDQ5M18x
```

SAML 구성 옵션을 .json 파일 내의 키-값 맵으로 지정합니다.

> **Important**
>
> SAML 옵션에 대한 업데이트는 증분되지 않습니다. 업데이트할 때 `SAMLOptions` 객체의 파라미터 값을 지정하지 않으면 기존 값이 빈 값으로 재정의됩니다. 예를 들어 현재 구성에 `userAttribute`에 대한 값이 포함된 경우 업데이트를 수행하고 이 값을 포함하지 않으면 해당 값이 구성에서 제거됩니다. `GetSecurityConfig` 작업을 호출하여 업데이트하기 전에 기존 값이 무엇인지 확인합니다.

SAML 공급자 삭제

SAML 공급자를 삭제하면 데이터 액세스 정책에서 연결된 사용자 및 그룹에 대한 모든 참조가 더 이상 작동하지 않습니다. 혼동을 피하려면 엔드포인트를 삭제하기 전에 액세스 정책에서 엔드포인트에 대한 모든 참조를 제거하는 것이 좋습니다.

OpenSearch Serverless 콘솔을 사용하여 SAML 공급자를 삭제하려면 Authentication(인증)을 선택하고 공급자를 선택한 다음 Delete(삭제)를 선택합니다.

OpenSearch Serverless API를 통해 공급자를 삭제하려면 DeleteSecurityConfig 요청을 보냅니다.

```
aws opensearchserverless delete-security-config --id saml/123456789012/myprovider
```

Amazon OpenSearch Serverless에 대한 규정 준수 확인

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon OpenSearch Serverless의 보안 및 규정 준수를 평가합니다. 이 프로그램에는 SOC, PCI 및 HIPAA가 포함됩니다.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 제공 범위 섹션을 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다.는 규정 준수를 지원하기 위해 다음 리소스를 AWS 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.

- HIPAA 적격 서비스 참조 - HIPAA 적격 서비스가 나열되어 있습니다. 모두 HIPAA 자격이 AWS 서비스 있는 것은 아닙니다.
- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 산업 및 위치에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 가이드에는 여러 프레임워크(미국 국립표준기술연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제표준화기구(ISO) 포함)의 보안 제어에 대한 지침을 보호하고 AWS 서비스 매핑하는 모범 사례가 요약되어 있습니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 -이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub - 내 보안 상태에 대한 포괄적인 보기를 AWS 서비스 제공합니다 AWS. Security Hub는 보안 컨트롤을 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 의심스러운 활동과 악의적인 활동이 있는지 환경을 모니터링하여 사용자, AWS 계정워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험과 규정 및 업계 표준 준수를 관리하는 방법을 간소화할 수 있습니다.

Amazon OpenSearch Serverless 컬렉션 태그 지정

태그를 사용하면 Amazon OpenSearch Serverless 컬렉션에 임의 정보를 할당할 수 있으므로 해당 정보를 분류하고 필터링할 수 있습니다. 태그는 사용자가 할당하거나가 AWS 리소스에 AWS 할당하는 메타데이터 레이블입니다.

각 태그는 키와 값으로 구성됩니다. 사용자가 할당하는 태그에 대해 키와 값을 정의합니다. 예를 들어 키를 `stage`로 정의하고 리소스 하나의 값을 `test`로 정의할 수 있습니다.

태그를 사용하면 AWS 리소스를 식별하고 구성할 수 있습니다. 많은 AWS 서비스가 태그 지정을 지원하므로 다른 서비스의 리소스에 동일한 태그를 할당하여 리소스가 관련이 있음을 나타낼 수 있습니다. 예를 들어 Amazon OpenSearch Service 도메인에 할당하는 것과 동일한 태그를 OpenSearch Serverless 컬렉션에 할당할 수 있습니다.

OpenSearch Serverless에서 기본 리소스는 컬렉션입니다. OpenSearch Service 콘솔, AWS CLI, OpenSearch Serverless API 작업 또는 AWS SDKs를 사용하여 컬렉션에서 태그를 추가, 관리 및 제거할 수 있습니다.

필요한 권한

OpenSearch Serverless는 컬렉션에 태그를 지정하는 데 다음 AWS Identity and Access Management Access Analyzer (IAM) 권한을 사용합니다.

- `aoss:TagResource`
- `aoss:ListTagsForResource`
- `aoss:UntagResource`

컬렉션 태그 지정(콘솔)

콘솔은 컬렉션에 태그를 지정하는 가장 간단한 방법입니다.

태그를 만들려면(콘솔)

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 Serverless(서버리스)를 확장하고 Collections(컬렉션)를 선택합니다.
3. 태그를 추가할 컬렉션을 선택한 다음 Tags(태그) 탭으로 이동합니다.
4. [관리(Manage)], [새 태그 추가(Add new tag)]를 선택합니다.
5. 태그 키와 선택 값을 입력합니다.
6. 저장을 선택합니다.

태그를 삭제하려면 동일한 단계를 따르고 [태그 관리(Manage tags)] 페이지에서 [제거(Remove)]를 선택합니다.

콘솔을 사용한 태그 작업에 대한 자세한 내용은 AWS 관리 콘솔 시작 안내서에서 Tag Editor를 참조하세요.

컬렉션 태그 지정(AWS CLI)

를 사용하여 컬렉션에 태그를 지정하려면 TagResource 요청을 AWS CLI보냅니다.

```
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
```

ListTagsForResource 명령을 사용하여 컬렉션의 기존 태그를 확인합니다.

```
aws opensearchserverless list-tags-for-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
```

UntagResource 명령을 사용하여 컬렉션에서 태그를 제거합니다.

```
aws opensearchserverless untag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tag-keys service
```

Amazon OpenSearch Serverless에서 지원되는 작업 및 플러그인

Amazon OpenSearch Serverless는 다양한 OpenSearch 플러그인은 물론 OpenSearch에서 사용할 수 있는 인덱싱, 검색, 메타데이터 API 작업의 하위 집합을 지원합니다. 특정 작업에 대한 액세스를 제한하기 위해 데이터 액세스 정책 내 테이블의 왼쪽 열에 권한을 포함할 수 있습니다.

주제

- 지원되는 OpenSearch API 작업 및 권한
- 지원되는 OpenSearch 플러그인

지원되는 OpenSearch API 작업 및 권한

다음 표에는 해당 데이터 액세스 정책 권한과 함께 OpenSearch Serverless가 지원하는 API 작업이 나와 있습니다.

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
`aoss:CreateIndex`	PUT <index>	인덱스를 생성합니다. 자세한 정보는 인덱스 생성을 참조하세요. **Note** 이 권한은 OpenSearch Dashboards의 샘플 데이터를 사용하여

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
		인덱스를 생성할 때도 적용됩니다.
`aoss:DescribeIndex`	• GET <index> • GET <index>/_mapping • GET <index>/_mappings • GET <index>/_setting • GET <index>/_setting/<setting> • GET <index>/_settings • GET <index>/_settings/<setting> • GET _cat/indices • GET _mapping • GET _mappings • GET _resolve/index/<index> • HEAD <index>	인덱스를 설명합니다. 자세한 정보는 다음 자료를 참조하세요. • 인덱스 가져오기 • 매핑 가져오기 • 설정 가져오기 • 인덱스가 존재함 • CAT 인덱스(응답에는 `health` 또는 `status` 필드가 포함되지 않음)

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
`aoss:WriteDocument`	• DELETE <index>/_doc/<id> • POST <index>/_bulk • POST <index>/_create/<id>(검색 컬렉션 유형에만 해당) • POST <index>/_doc • POST <index>/_update/<id>(검색 컬렉션 유형에만 해당) • POST _bulk • PUT <index>/_create/<id>(검색 컬렉션 유형에만 해당) • PUT <index>/_doc/<id>(검색 컬렉션 유형에만 해당)	문서를 작성하고 업데이트합니다. 자세한 정보는 다음 자료를 참조하세요. • 대량 • 데이터 인덱싱 **Note** 일부 작업은 SEARCH 유형의 컬렉션에만 허용됩니다. 자세한 내용은 the section called “컬렉션 유형 선택” 단원을 참조하십시오.

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
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	문서를 읽습니다. 자세한 정보는 다음 자료를 참조하세요. • 텍스트 분석 수행 • 문서 가져오기 • 개수 • DSL 쿼리 • 순위 평가 • API 분석 • 설명 • 특정 시점 • SQL 및 PPL

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
	• POST <index>/_search • POST _analyze • POST _field_caps • POST _search • /_search/point_in_time/_all 삭제 • /_search/point_in_time 삭제	
`aoss:DeleteIndex`	DELETE <target>	인덱스를 삭제합니다. 자세한 내용은 인덱스 삭제 섹션을 참조하세요.
`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	인덱스 설정을 업데이트합니다. 자세한 정보는 다음 자료를 참조하세요. • 매핑 • 설정 업데이트

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
`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>	• 인덱스 별칭, 파이프라인 및 템플릿을 생성합니다. 자세한 내용은 별칭 생성을 참조하세요. • * 템플릿 프로비저닝 또는 재프로비저닝. ML Commons Client 및 OpenSearch Serverless 서비스는 종속 정책을 관리합니다.

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
`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>	별칭, 인덱스 및 프레임워크 템플릿, 파이프라인을 사용하는 방법을 설명합니다. 자세한 정보는 다음 자료를 참조하세요. • 별칭 관리 • 인덱스 템플릿

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
	• 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>	별칭, 인덱스 템플릿 및 프레임워크 템플릿을 업데이트합니다. 자세한 정보는 다음 자료를 참조하세요. • 인덱스 별칭 • 인덱스 템플릿 * 템플릿을 프로비저닝 해제할 API입니다. ML Commons Client 및 OpenSearch Serverless 서비스는 종속 정책을 관리합니다.

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
`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>	별칭, 인덱스 및 프레임워크 템플릿, 파이프라인을 삭제합니다. 자세한 정보는 다음 자료를 참조하세요. • 별칭 삭제 • 템플릿 삭제

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
`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	모델 및 커넥터에 대한 정보를 검색하기 위한 GET 및 검색 APIs에 대해 설명합니다.
`aoss:CreateMLResource`	• POST /_plugins/_ml/models/_register • POST /_plugins/_ml/model_groups/_register • POST /_plugins/_ml/connectors/_create	ML 리소스를 생성할 수 있는 권한을 제공합니다.

데이터 액세스 정책 권한	OpenSearch API 작업	설명 및 주의 사항
`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>	기존 ML 리소스를 업데이트할 수 있는 권한을 제공합니다.
`aoss:DeleteMLResource`	• /_plugins/_ml/models/<model_id> 삭제 • DELETE _plugins/_ml/model_groups/<model_group_id> • DELETE /_plugins/_ml/connectors/<connector_id> • DELETE /_plugins/_ml/tasks/<task_id>	ML 리소스를 삭제할 수 있는 권한을 제공합니다.
`aoss:ExecuteMLResource`	• POST /_plugins/_ml/models/<model_id>/_predict	모델을 실행할 수 있는 권한을 제공합니다.

지원되는 OpenSearch 플러그인

OpenSearch Serverless 컬렉션에는 OpenSearch 커뮤니티의 다음 플러그인이 미리 포함되어 제공됩니다. Serverless는 자동으로 플러그인을 배포하고 관리합니다.

분석 플러그인

- ICU Analysis
- Japanese (kuromoji) Analysis
- Korean (Nori) Analysis

- Phonetic Analysis
- Smart Chinese Analysis
- Stempel Polish Analysis
- Ukrainian Analysis

Mapper plugins

- Mapper Size
- Mapper Murmur3
- Mapper Annotated Text

Scripting plugins

- Painless
- 표현식
- Mustache

또한 OpenSearch Serverless에는 모듈로 제공되는 모든 플러그인이 포함되어 있습니다.

Amazon OpenSearch Serverless 모니터링

모니터링은 Amazon OpenSearch Serverless 및 기타 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다.는 OpenSearch Serverless를 모니터링하고, 이상이 있을 때 보고하고, 필요한 경우 자동 조치를 취할 수 있도록 다음과 같은 모니터링 도구를 AWS 제공합니다.

- Amazon CloudWatch는 AWS 리소스와 AWS 실행 중인 애플리케이션을 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임곗값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다.

  예를 들어 CloudWatch에서 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 추적하고 필요할 때 자동으로 새 인스턴스를 시작할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.
- AWS CloudTrail는 AWS 계정에서 또는 이 계정을 대신하여 수행된 API 호출 및 관련 이벤트를 캡처합니다. 지정한 Amazon S3 버킷으로 로그 파일을 전달합니다. 호출한 사용자 및 계정 AWS, 호출이

수행된 소스 IP 주소, 호출이 발생한 시기를 식별할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

- Amazon EventBridge는 OpenSearch Service 도메인의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. 특정 이벤트를 감시하는 규칙을 생성하고 이러한 이벤트가 발생할 AWS 서비스 때 다른에서 자동화된 작업을 트리거할 수 있습니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

Amazon CloudWatch로 OpenSearch Serverless 모니터링

원시 데이터를 수집하고 읽을 수 있는 거의 실시간 지표로 처리하는 CloudWatch를 사용하여 Amazon OpenSearch Serverless를 모니터링할 수 있습니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다.

특정 임곗값을 주시하다가 해당 임곗값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.

OpenSearch Serverless는 `AWS/AOSS` 네임스페이스에서 다음 지표를 보고합니다.

지표	설명
`ActiveCollection`	컬렉션이 활성 상태인지 여부를 나타냅니다. 값이 1이면 컬렉션이 `ACTIVE` 상태임을 의미합니다. 이 값은 컬렉션 생성에 성공하면 내보내지며 컬렉션을 삭제할 때까지 1로 유지됩니다. 메트릭의 값은 0일 수 없습니다. 관련 통계: 최대 차원: `ClientId`, `CollectionId` , `CollectionName` 빈도: 60초
`DeletedDocuments`	삭제된 문서의 총 수입니다. 관련 통계: 평균, 합계 차원: `ClientId`, `CollectionId` , `CollectionName` , `IndexId`, `IndexName`

지표	설명
	빈도: 60초
IndexingOCU	컬렉션 데이터를 수집하는 데 사용되는 OpenSearch 컴퓨팅 유닛(OCU)의 수입니다. 이 지표는 계정 수준에서 적용됩니다. 관련 통계: 합계 차원: ClientId 빈도: 60초
IngestionDataRate	컬렉션 또는 인덱스에 대한 초당 GiB의 인덱싱 속도. 이 지표는 대량 인덱싱 요청에만 적용됩니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName , IndexId, IndexName 빈도: 60초
IngestionDocumentErrors	컬렉션 또는 인덱스에 대한 수집 중에 발생한 문서 오류의 총 수입니다. 대량 인덱싱 요청이 성공하면 작성자는 요청을 처리하고 요청 내의 모든 실패한 문서에 대해 오류를 내보냅니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName , IndexId, IndexName 빈도: 60초

지표	설명
IngestionDocumentRate	문서가 컬렉션 또는 인덱스로 수집되는 초당 속도입니다. 이 지표는 대량 인덱싱 요청에만 적용됩니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName , IndexId, IndexName 빈도: 60초
IngestionRequestErrors	컬렉션에 대한 대량 인덱싱 요청 오류의 총 수입니다. OpenSearch Serverless는 인증 또는 가용성 문제와 같은 어떤 이유로든 대량 인덱싱 요청이 실패할 때 이 메트릭을 내보냅니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName 빈도: 60초
IngestionRequestLatency	컬렉션에 대한 대량 쓰기 작업의 대기 시간(초). 관련 통계: 최소, 최대, 평균 차원: ClientId, CollectionId , CollectionName 빈도: 60초
IngestionRequestRate	컬렉션에서 수신한 대량 쓰기 작업의 총 수입니다. 관련 통계: 최소, 최대, 평균 차원: ClientId, CollectionId , CollectionName 빈도: 60초

지표	설명
IngestionRequestSuccess	컬렉션에 대한 성공적인 인덱싱 작업의 총 수입니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName 빈도: 60초
SearchableDocuments	컬렉션 또는 인덱스에서 검색 가능한 문서의 총 수입니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName , IndexId, IndexName 빈도: 60초
SearchRequestErrors	컬렉션에 대한 분당 쿼리 오류의 총 수입니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName 빈도: 60초
SearchRequestLatency	컬렉션에 대한 검색 작업을 완료하는 데 걸리는 평균 시간(밀리초). 관련 통계: 최소, 최대, 평균 차원: ClientId, CollectionId , CollectionName 빈도: 60초

지표	설명
SearchOCU	컬렉션 데이터를 검색하는 데 사용되는 OpenSearch 컴퓨팅 유닛(OCU)의 수입니다. 이 지표는 계정 수준에서 적용됩니다. 관련 통계: 합계 차원: ClientId 빈도: 60초
SearchRequestRate	컬렉션에 대한 분당 검색 요청의 총 수입니다. 관련 통계: 평균, 최대, 합계 차원: ClientId, CollectionId , CollectionName 빈도: 60초
StorageUsedInS3	Amazon S3 스토리지의 사용량(바이트)입니다. OpenSearch Serverless는 Amazon S3에 인덱싱된 데이터를 저장합니다. 정확한 값을 얻으려면 이 기간을 1분으로 선택해야 합니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName , IndexId, IndexName 빈도: 60초
2xx, 3xx, 4xx, 5xx	지정된 HTTP 응답 코드(2xx, 3xx, 4xx, 5xx)를 초래한 컬렉션에 대한 요청 수입니다. 관련 통계: 합계 차원: ClientId, CollectionId , CollectionName 빈도: 60초

를 사용하여 OpenSearch Serverless API 호출 로깅 AWS CloudTrail

Amazon OpenSearch Serverless는 Serverless에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다.

CloudTrail은 OpenSearch Serverless에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 OpenSearch Service 콘솔의 서버리스 섹션에서의 호출과 OpenSearch Serverless API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면 OpenSearch Serverless에 대한 이벤트를 포함한 CloudTrail 이벤트를 지속해서 Amazon S3 버킷에 배포할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여 OpenSearch Serverless에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 OpenSearch Serverless 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. OpenSearch Serverless에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

OpenSearch Serverless에 대한 이벤트를 AWS 계정포함하여에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다.

추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

모든 OpenSearch Serverless 작업은 CloudTrail에 의해 기록되며 OpenSearch Serverless API 참조에 문서화됩니다. 예를 들어 `CreateCollection`, `ListCollections` 및 `DeleteCollection` 작업을 직접적으로 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명 정보를 이용하면 다음을 판단하는 데 도움이 됩니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

OpenSearch Serverless 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다.

이벤트는 모든 소스로부터 단일 요청을 나타냅니다. 요청된 작업, 모든 파라미터, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 `CreateCollection` 작업을 보여주는 CloudTrail 로그 항목이 나타냅니다.


```
{
   "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"
   }
}
```

Amazon EventBridge를 사용하여 OpenSearch 서버리스 이벤트 모니터링

Amazon OpenSearch Service는 Amazon EventBridge와 통합하여 도메인에 영향을 주는 특정 이벤트를 사용자에게 알립니다. AWS 서비스의 이벤트는 거의 실시간으로 EventBridge로 전달됩니다. 동일한 이벤트가 Amazon EventBridge 이전 버전인 Amazon CloudWatch Events에도 전송됩니다. 원하는

이벤트만 표시하도록 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 자동으로 활성화할 수 있는 작업의 예는 다음과 같습니다.

- AWS Lambda 함수 호출
- Amazon EC2 Run Command 호출
- Amazon Kinesis Data Streams로 이벤트 릴레이
- AWS Step Functions 상태 시스템 활성화
- SNS 주제 또는 Amazon SQS 대기열 알림

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 시작하기를 참조하세요.

알림 설정

AWS 사용자 알림을 사용하여 OpenSearch Serverless 이벤트 발생 시 알림을 받을 수 있습니다. 이벤트는 OCU 사용량이 최대 한도에 도달했을 때와 같이 OpenSearch Serverless 환경의 변화를 나타내는 지표입니다. Amazon EventBridge 은 이벤트를 수신하고 AWS Management Console 알림 센터 및 선택한 전송 채널로 알림을 라우팅합니다. 이벤트가 지정한 규칙과 일치하면 알림을 받습니다.

OpenSearch Compute Units(OCU) 이벤트

OpenSearch 서버리스는 다음 OCU 관련 이벤트가 발생할 때 EventBridge에 이벤트를 보냅니다.

OCU 사용량이 최대 한도에 근접함

OpenSearch 서버리스는 검색 또는 인덱스 OCU 사용량이 용량 제한의 75%에 도달하면 이 이벤트를 전송합니다. OCU 사용량은 구성된 용량 제한과 현재 OCU 사용량을 기준으로 계산됩니다.

예제

다음은 이러한 유형의 이벤트 예입니다(검색 OCU).

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

다음은 이러한 유형의 이벤트 예입니다(인덱스 OCU).

```
{
  "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 사용량이 최대 한도에 도달했습니다.

OpenSearch 서버리스는 검색 또는 인덱스 OCU 사용량이 용량 제한의 100%에 도달하면 이 이벤트를 전송합니다. OCU 사용량은 구성된 용량 제한과 현재 OCU 사용량을 기준으로 계산됩니다.

예제

다음은 이러한 유형의 이벤트 예입니다(검색 OCU).

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

다음은 이러한 유형의 이벤트 예입니다(인덱스 OCU).

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

Amazon OpenSearch Service 도메인 생성 및 관리

이 장에서는 Amazon OpenSearch Service 도메인을 만들고 관리하는 방법에 대해 설명합니다. 도메인은 오픈 소스 OpenSearch 클러스터와 동등한 AWS에서 프로비저닝된 도메인입니다. 도메인을 생성할 때 해당 설정, 인스턴스 유형, 인스턴스 수 및 스토리지 할당을 지정합니다. 오픈 소스 클러스터에 대한 자세한 내용은 OpenSearch 설명서의 Creating a cluster를 참조하세요.

튜토리얼 시작하기의 간단한 지침과 달리 이 장에서는 모든 옵션에 대해 설명하고 관련 참조 정보를 제공합니다. OpenSearch Service 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDKs.

OpenSearch Service 도메인 생성

이 섹션에서는 OpenSearch Service 콘솔을 사용하거나 `create-domain` 명령과 AWS CLI 함께를 사용하여 OpenSearch Service 도메인을 생성하는 방법을 설명합니다.

OpenSearch Service 도메인(콘솔) 생성

콘솔에서 다음 절차에 따라 OpenSearch Service 도메인을 만듭니다.

OpenSearch Service 도메인(콘솔)을 만들려면

1. https://aws.amazon.com으로 이동하여 콘솔에 로그인(Sign In to the Console)을 선택합니다.
2. Analytics(분석)에서 Amazon OpenSearch Service를 선택합니다.
3. [도메인 생성(Create domain)]을 선택합니다.
4. 도메인 이름(Domain name)에 도메인 이름을 입력합니다. 이름은 다음 조건을 충족해야 합니다.
   - 계정 및에 고유 AWS 리전
   - 소문자로 시작할 것
   - 3~28자 사이일 것
   - 소문자 a~z, 숫자 0~9 및 하이픈(-)만 포함할 것
5. 도메인 생성 방법으로 [표준 생성]을 선택합니다.
6. 템플릿에서 도메인 목적에 가장 적합한 옵션을 선택합니다.
   - 고가용성과 성능이 필요한 워크로드용 프로덕션 도메인. 이러한 도메인은 가용성을 더 높이기 위해 Multi-AZ(대기 포함 또는 미포함)와 전용 프라이머리 노드를 사용합니다.

- 개발 또는 테스트용 개발/테스트. 이러한 도메인은 Multi-AZ(대기 포함 또는 대기 미포함) 또는 단일 가용 영역을 사용할 수 있습니다.

> **Important**
>
> 배포 유형이 다르면 다음 페이지에 표시되는 옵션도 다릅니다. 이 단계에는 모든 옵션이 포함됩니다.

7. 배포 옵션의 경우 대기 포함 도메인을 선택하여 3-AZ 도메인을 구성합니다. 이때 영역 중 하나에 있는 노드는 대기로 예약되어 있습니다. 이 옵션은 지정된 데이터 노드 수, 프라이머리 노드 수, 인스턴스 유형, 복제본 수, 소프트웨어 업데이트 설정과 같은 여러 모범 사례를 적용합니다.
8. 버전(Version)에서 사용할 OpenSearch 또는 레거시 Elasticsearch OSS의 버전을 선택합니다. OpenSearch의 최신 버전을 선택하는 것이 좋습니다. 자세한 내용은 the section called “지원되는 Elasticsearch 및 OpenSearch 버전” 섹션을 참조하세요.

   (선택 사항) 도메인에서 OpenSearch 버전을 선택한 경우 호환성 모드 활성화(Enable compatibility mode)를 선택하여 OpenSearch가 7.10으로 버전을 보고하도록 할 수 있습니다. 그러면 연결하기 전에 버전을 확인하는 특정 Elasticsearch OSS 클라이언트와 플러그인이 서비스 작업을 계속할 수 있습니다.
9. 인스턴스 유형(Instance type)에서 데이터 노드의 인스턴스 유형을 선택합니다. 자세한 내용은 the section called “지원되는 인스턴스 유형” 섹션을 참조하세요.

> **Note**
>
> 모든 가용 영역에서 모든 인스턴스 유형이 지원되는 것은 아닙니다. Multi-AZ를 선택할 경우 R5 또는 I3 등의 최신 세대 인스턴스 유형을 선택할 것을 권장합니다.

10. 노드 수에서 데이터 노드 수를 선택합니다.

    최대값은 OpenSearch Service 도메인 및 인스턴스 할당량을 참조하세요. 단일 노드 클러스터는 개발 및 테스트 용도로 적합할 뿐 프로덕션 워크로드에 사용해서는 안 됩니다. 자세한 지침은 the section called “도메인 크기 조정” 및 the section called “다중 AZ 도메인 구성” 섹션을 참조하세요.

> **Note**
>
> (선택 사항) 전용 조정자 노드는 모든 OpenSearch 버전 및 ElasticSearch 버전 6.8~7.10을 지원합니다. 전용 조정자 노드는 전용 클러스터 관리자가 활성화된 도메인에서 사용할 수 있습니다. 전용 조정자 노드를 활성화하려면 인스턴스 유형과 개수를 선택합니다. 모

범 사례는, 전용 조정자 노드의 인스턴스 패밀리를 데이터 노드(Intel 기반 인스턴스 또는 Graviton 기반 인스턴스)와 동일하게 유지하는 것입니다.

11. 스토리지 유형으로는 Amazon EBS를 선택합니다. 목록에서 사용 가능한 볼륨 유형은 선택한 인스턴스 유형에 따라 다릅니다. 매우 큰 도메인을 생성하기 위한 지침은 the section called “페타바이트 규모” 섹션을 참조하세요.
12. EBS 스토리지의 경우 다음 추가 설정을 구성합니다. 선택한 볼륨 유형에 따라 일부 설정이 표시되지 않을 수 있습니다.

설정	설명
EBS 볼륨 유형	범용(SSD) - gp3 및 범용(SSD) - gp2 또는 이전 세대 프로비저닝된 IOPS(SSD) 및 마그네틱(표준) 중에서 선택합니다.
노드당 EBS 스토리지 크기	각 데이터 노드에 연결할 EBS 볼륨 스토리지의 크기를 입력합니다. EBS 볼륨 크기는 노드당 크기입니다. 데이터 노드 수에 EBS 볼륨 크기를 곱하여 OpenSearch Service 도메인의 총 클러스터 크기를 계산할 수 있습니다. EBS 볼륨의 최소 크기 및 최대 크기는 지정된 EBS 볼륨 유형과 볼륨이 연결된 인스턴스 유형에 따라 달라집니다. 자세한 내용은 EBS 볼륨 크기 제한 섹션을 참조하세요.
프로비저닝된 IOPS	프로비저닝된 IOPS SSD 볼륨 유형을 선택한 경우, 볼륨에서 지원되는 초당 I/O(IOPS) 수를 입력합니다.

13. (선택 사항) gp3 볼륨 유형을 선택한 경우, 고급 설정을 확장하고 스토리지 가격에 포함된 것 이상인 추가 비용으로 추가 IOPS(데이터 노드당 프로비저닝한 3TiB의 볼륨 크기마다 최대 16,000) 및 추가 처리량(데이터 노드당 프로비저닝한 3TiB의 볼륨 크기마다 최대 1,000MiB/s)을 지정합니다. 자세한 정보는 Amazon OpenSearch Service 가격을 참조하세요.
14. (선택 사항) UltraWarm 스토리지를 활성화하려면 UltraWarm 데이터 노드 활성화(Enable UltraWarm data nodes)를 선택합니다. 각 인스턴스 유형별로 처리할 수 있는 최대 스토리지 용량이 있습니다. 주소 지정 가능한 총 웜 스토리지에 대한 웜 데이터 노드 수를 이 값에 곱합니다.
15. (선택 사항) 콜드 스토리지를 활성화하려면 콜드 스토리지 활성화(Enable cold storage)를 선택합니다. 콜드 스토리지를 사용하려면 UltraWarm Warm을 활성화해야 합니다.

16. Multi-AZ with Standby를 사용하는 경우 세 개의 전용 프라이머리 노드가 이미 활성화되어 있습니다. 원하는 프라이머리 노드 유형을 선택합니다. Multi-AZ without Standby 도메인을 선택한 경우 전용 프라이머리 노드 활성화를 선택하고 원하는 프라이머리 노드의 유형과 수를 선택합니다. 전용 프라이머리 노드는 클러스터 안정성을 높이고 인스턴스 개수가 10개보다 많은 도메인에 필요합니다. 프로덕션 도메인의 경우 3개의 전용 프라이머리 노드를 권장합니다.

> **Note**
>
> 전용 프라이머리 노드와 데이터 노드에 대해 다른 인스턴스 유형을 선택할 수 있습니다. 예를 들면 데이터 노드의 일반 목적 또는 스토리지 최적화 인스턴스를 선택할 수 있지만 전용 프라이머리 노드의 컴퓨팅에 최적화된 인스턴스는 선택할 수 없습니다.

17. (선택 사항) OpenSearch 또는 Elasticsearch 5.3 이상을 실행하는 도메인의 경우스냅샷 구성(Snapshot configuration)은 관련이 없습니다. 자동 스냅샷에 대한 자세한 내용은 the section called “인덱스 스냅샷 생성” 섹션을 참조하세요.

18. `https://search-`*`mydomain`*`-`*`1a2a3a4a5a6a7a8a9a0a9a8a7a`*`.`*`us-east-1`*`.es.amazonaws.com`의 표준 끝점이 아닌 사용자 지정 끝점을 사용하려는 경우 사용자 지정 엔드포인트 활성화(Enable custom endpoint)를 클릭하고 이름과 인증서를 제공합니다. 자세한 내용은 the section called “사용자 지정 엔드포인트 만들기” 단원을 참조하십시오.

19. [네트워크(Network)]에서 [VPC 액세스(VPC access)] 또는 [퍼블릭 액세스(Public access)]를 선택합니다. 퍼블릭 액세스(Public access)를 선택한 경우, 다음 단계로 건너뜁니다. VPC access(VPC 액세스)를 선택한 경우, 사전 조건이 충족되었는지 확인한 후 다음 설정을 구성합니다.

설정	설명
VPC	사용하려는 Virtual Private Cloud(VPC)를 선택합니다. VPC 및 도메인은 동일한 AWS 리전에 있어야 하며 테넌시가 기본값으로 설정된 VPC를 선택해야 합니다. OpenSearch Service는 전용 테넌시를 사용하는 VPC를 아직 지원하지 않습니다.
서브넷	서브넷을 선택합니다. 다중 AZ를 활성화한 경우, 서브넷을 두 개 또는 세 개 선택해야 합니다. OpenSearch Service가 서브넷에 VPC 엔드포인트와 탄력적 네트워크 인터페이스를 배치합니다. 서브넷에서 네트워크 인터페이스용 IP 주소를 충분히 예약해야 합니다. 자세한 내용은 VPC 서브넷에서 IP 주소 예약 섹션을 참조하세요.

설정	설명
보안 그룹	필요한 애플리케이션이 도메인에 의해 노출된 포트(80 또는 443) 및 프로토콜(HTTP 또는 HTTPS)에서 OpenSearch Service 도메인에 도달하도록 허용하는 VPC 보안 그룹을 하나 이상 선택합니다. 자세한 내용은 the section called “VPC 지원” 단원을 참조하십시오.
[IAM Role]	기본 역할을 유지합니다. OpenSearch Service가 이 사전 정의된 역할(서비스 연결 역할이라고도 함)을 사용하여 VPC에 액세스하고 VPC의 서브넷에 VPC 엔드포인트와 네트워크 인터페이스를 배치합니다. 자세한 내용은 VPC 액세스를 위한 서비스 연결 역할 섹션을 참조하세요.
IP 주소 유형	IP 주소 유형으로 이중 스택 또는 IPv4를 선택합니다. 이중 스택을 사용하면 IPv4 및 IPv6 주소 유형 간에 도메인 리소스를 공유할 수 있으며 권장되는 옵션입니다. IP 주소 유형을 이중 스택으로 설정하면 나중에 주소 유형을 변경할 수 없습니다.

20. 세분화된 액세스 제어 활성화 또는 비활성화:

    - 사용자 관리에 IAM을 사용하려면 IAM ARN을 마스터 사용자로 설정(Set IAM ARN as master user)을 선택하고 IAM 역할의 ARN을 지정합니다.
    - 내부 사용자 데이터베이스를 사용하려면 [기본 사용자 생성]을 선택하고 사용자 이름과 암호를 지정합니다.

    어느 옵션을 선택하든 마스터 사용자는 클러스터의 모든 인덱스와 모든 OpenSearch API에 액세스할 수 있습니다. 선택할 옵션에 대한 지침은 the section called “주요 개념” 섹션을 참조하세요.

    세분화된 액세스 제어를 비활성화한 경우에도 VPC 내에 배치하거나 제한적인 액세스 정책을 적용하거나 둘 다를 통해 도메인에 대한 액세스를 제어할 수 있습니다. 세분화된 액세스 제어를 사용하려면 노드 간 암호화 및 유휴 데이터 암호화를 활성화해야 합니다.

    > **Note**
    >
    > 매우 권장되는 사항으로 도메인의 데이터를 보호하기 위해 세분화된 액세스 제어를 활성화해야 합니다. 세분화된 액세스 제어는 클러스터, 인덱스, 문서 및 필드 수준에서 보안을 제공합니다.

21. (선택 사항) OpenSearch Dashboards에 SAML 인증을 사용하려면 SAML 인증 활성화를 선택하고 도메인에 대한 SAML 옵션을 구성합니다. 지침은 the section called “OpenSearch 대시보드에 대한 SAML 인증” 섹션을 참조하세요.
22. (선택 사항) OpenSearch 대시보드에 Amazon Cognito 인증을 사용하려면 Amazon Cognito 인증 활성화(Enable Amazon Cognito authentication)를 선택합니다. 그런 다음 OpenSearch 대시보드 인증에 사용할 Amazon Cognito 사용자 풀 및 자격 증명 풀을 선택합니다. 이러한 리소스를 만드는 방법은 the section called “OpenSearch Dashboards에 대한 Amazon Cognito 인증” 섹션을 참조하세요.
23. (선택 사항) IAM Identity Center(IDC) 인증을 사용하여 기존 ID 소스를 연결하고 AWS 애플리케이션에 사용자의 공통 보기를 제공하려면 IAM Identity Center로 인증된 API 액세스 활성화를 선택합니다. 자세한 내용은 IAM Identity Center 사용 설명서의 신뢰할 수 있는 자격 증명 전파 개요를 참조하세요.
24. (선택 사항) 이러한 기능을 사용하려면 고급 기능 섹션에서 자연어 쿼리 생성 및 Amazon Q Developer 기능 활성화를 선택한 상태로 둡니다.

    향상된 벡터 검색 옵션의 엔진 옵션으로 S3 벡터 활성화를 선택합니다. 자세한 내용은 (미리 보기) Amazon S3 벡터 엔진을 사용한 고급 검색 기능 단원을 참조하십시오.
25. 액세스 정책에서 액세스 정책을 선택하거나 사용자 고유의 액세스 정책을 구성합니다. 사용자 지정 정책을 생성하도록 선택한 경우 직접 구성하거나 다른 도메인에서 가져올 수 있습니다. 자세한 내용은 the section called “ID 및 액세스 관리” 섹션을 참조하세요.

    > **Note**
    >
    > VPC 액세스를 활성화한 경우 IP 기반 정책은 사용할 수 없습니다. 대신에 보안 그룹을 사용하여 어느 IP 주소가 도메인에 액세스할 수 있는지 제어할 수 있습니다. 자세한 내용은 the section called “VPC 도메인 액세스 정책에 대하여” 섹션을 참조하세요.

26. (선택 사항) 도메인에 대한 모든 요청이 HTTPS를 통해 도착하도록 하려면 도메인에 대한 모든 트래픽에 HTTPS 요구(Require HTTPS for all traffic to the domain)를 선택합니다. (선택 사항) 노드 간 암호화를 사용하려면 노드 간 암호화를 선택합니다. 자세한 내용은 the section called “노드 간 암호화” 단원을 참조하십시오. (선택 사항) 저장 데이터의 암호화를 활성화하려면 저장 데이터 암호화 활성화를 선택합니다. Multi-AZ with Standby 옵션을 선택한 경우 이러한 옵션이 미리 선택됩니다.
27. (선택 사항) AWS 소유 키 사용을 선택하여 OpenSearch Service가 사용자를 대신하여 AWS KMS 암호화 키를 생성하도록 합니다(또는 이미 생성한 암호화 키를 사용). 그렇지 않으면 자체 KMS 키를 선택합니다. 자세한 내용은 the section called “저장 시 암호화” 단원을 참조하십시오.

28. 사용량이 적은 기간의 경우 시작 시간을 선택하여 블루/그린 배포가 필요한 서비스 소프트웨어 업데이트 및 자동 조정 최적화를 예약하세요. 비수기 업데이트는 트래픽이 많은 기간 동안 클러스터의 전용 프라이머리 노드에 가해지는 부담을 최소화하는 데 도움이 됩니다.
29. 자동 조정(Auto-Tune)에서 속도와 안정성을 향상시키기 위해 OpenSearch Service가 도메인에 대한 메모리 관련 구성 변경을 제안하도록 허용할지를 선택합니다. 자세한 내용은 the section called “자동 조정” 단원을 참조하십시오.

    (선택 사항) 유지 관리 기간 추가를 선택하여 자동 조정이 도메인을 업데이트하는 반복 기간을 예약합니다.
30. (선택 사항) 자동 소프트웨어 업데이트를 선택하여 자동 소프트웨어 업데이트를 활성화합니다.
31. (선택 사항) 도메인을 설명하는 태그를 추가하여 해당 정보를 분류하고 필터링할 수 있습니다. 자세한 내용은 the section called “도메인 태그 지정” 단원을 참조하십시오.
32. (선택 사항) 고급 클러스터 설정(Advanced cluster settings)을 확장하고 구성합니다. 이러한 옵션에 대한 요약은 the section called “고급 클러스터 설정” 섹션을 참조하세요.
33. 생성(Create)을 선택합니다.

OpenSearch Service 도메인 생성(AWS CLI)

콘솔을 사용하여 OpenSearch Service 도메인을 만드는 대신 AWS CLI를 사용할 수 있습니다. 구문은 AWS CLI 명령 참조에서 Amazon OpenSearch Service를 참조하세요.

예시 명령

이 첫 번째 예제에서는 다음 OpenSearch Service 도메인 구성을 보여줍니다.

- OpenSearch 버전 1.2를 사용하여 mylogs라는 이름의 OpenSearch Service 도메인을 만듭니다.
- 인스턴스 유형이 `r6g.large.search`인 인스턴스 2개를 사용하여 도메인을 채웁니다.
- 각 데이터 노드의 저장에 100GiB 범용(SSD) gp3 EBS 볼륨을 사용합니다.
- 단일 IP 주소(192.0.2.0/32)의 익명 액세스만 허용합니다.

```
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"]}}}]}'
```

다음 예제에서는 아래 OpenSearch Service 도메인 구성을 보여줍니다.

- Elasticsearch 버전 7.10을 사용하여 mylogs라는 이름의 OpenSearch Service 도메인을 만듭니다.
- 인스턴스 유형이 `r6g.large.search`인 인스턴스 6개를 사용하여 도메인을 채웁니다.
- 각 데이터 노드의 저장에 100GiB 범용(SSD) gp2 EBS 볼륨을 사용합니다.
- 사용자 AWS 계정 ID로 식별되는 단일 사용자로 서비스에 대한 액세스를 제한합니다. 555555555555
- 가용 영역 세 개에 인스턴스 분산

```
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/*" } ] }'
```

다음 예제에서는 아래 OpenSearch Service 도메인 구성을 보여줍니다.

- OpenSearch 버전 1.0을 사용하여 mylogs라는 이름의 OpenSearch Service 도메인을 만듭니다.
- 인스턴스 유형이 `r6g.xlarge.search`인 인스턴스 10개를 사용하여 도메인을 채웁니다.
- 전용 프라이머리 노드의 역할을 위해 인스턴스 유형이 `r6g.large.search`인 인스턴스 세 개를 사용하여 도메인을 채웁니다.
- 각 데이터 노드에 대해 1000 IOPS의 기본 성능으로 구성된 100GiB 프로비저닝된 IOPS EBS 볼륨을 저장에 사용합니다.
- 사용자 한 명과 하위 리소스 하나(`_search` API)만 액세스할 수 있도록 제한합니다.

```
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**
>
> OpenSearch Service 도메인을 만들려고 하는데 같은 이름의 도메인이 이미 존재하는 경우, CLI는 오류를 보고하지 않습니다. 그 대신 기존 도메인에 대한 세부 정보가 표시됩니다.

OpenSearch Service 도메인(AWS SDKs) 생성

AWS SDKs(Android 및 iOS SDKs 제외)는를 포함하여 Amazon OpenSearch Service API 참조에 정의된 모든 작업을 지원합니다`CreateDomain`. 샘플 코드에 대한 내용은 the section called “ AWS SDKs 사용” 섹션을 참조하십시오. AWS SDKs 설치 및 사용에 대한 자세한 내용은 AWS 소프트웨어 개발 키트를 참조하세요.

OpenSearch Service 도메인 생성(AWS CloudFormation)

OpenSearch Service는 AWS 리소스 및 인프라를 생성하고 관리하는 데 소요되는 시간을 줄일 수 있도록 리소스를 모델링하고 설정하는 데 도움이 되는 AWS CloudFormation서비스와 통합됩니다. 생성하려는 OpenSearch 도메인을 설명하는 템플릿을 생성하면 CloudFormation이 도메인을 프로비저닝하고 구성합니다. OpenSearch 도메인에 대한 JSON 및 YAML 템플릿의 예를 비롯한 자세한 내용은 AWS CloudFormation 사용 설명서에서 Amazon OpenSearch Service 리소스 유형 참조를 참조하세요.

액세스 정책 구성

Amazon OpenSearch Service는 OpenSearch Service 도메인에 대한 액세스를 구성하는 여러 가지 방법을 제공합니다. 자세한 내용은 the section called “ID 및 액세스 관리” 및 the section called “세분화된 액세스 제어” 섹션을 참조하세요.

콘솔이 사용자가 도메인의 필요에 따라 사용자 지정할 수 있는 사전 구성된 액세스 정책을 제공합니다. 사용자가 다른 OpenSearch Service 도메인의 액세스 정책을 가져올 수도 있습니다. 이러한 액세스 정

책이 VPC 액세스와 상호 작용하는 방식에 대한 자세한 내용은 the section called “VPC 도메인 액세스 정책에 대하여” 섹션을 참조하세요.

액세스 정책을 구성하려면(콘솔)

1. https://aws.amazon.com으로 이동하여 콘솔에 로그인(Sign In to the Console)을 선택합니다.
2. 분석(Analytics)에서 Amazon OpenSearch Service를 선택합니다.
3. 탐색 창의 [도메인(Domains)]에서 업데이트할 도메인을 선택합니다.
4. [작업(Actions)], [보안 구성 편집(Edit security configuration)]을 선택합니다.
5. 액세스 정책 JSON을 편집하거나 미리 구성된 옵션을 가져옵니다.
6. 변경 사항 저장을 선택합니다.

고급 클러스터 설정

고급 옵션을 사용하여 다음을 구성합니다.

요청 본문의 인덱스

HTTP 요청의 본문에서 인덱스에 대한 명시적 참조를 허용할지를 지정합니다. 이 속성을 `false`로 설정하면 사용자가 하위 리소스에 대한 액세스 제어를 우회하는 것을 방지할 수 있습니다. 기본값은 `true`입니다. 자세한 내용은 the section called “고급 옵션 및 API 고려 사항” 섹션을 참조하세요.

필드데이터 캐시 할당

필드 데이터에 할당되는 Java 힙 공간의 백분율을 지정합니다. 기본적으로 이 설정은 JVM 힙의 20%입니다.

> **Note**
>
> 많은 고객이 일일 인덱스 교체를 문의합니다. 이러한 사용 사례는 대부분 JVM 힙의 40%로 구성된 `indices.fielddata.cache.size`를 사용하여 벤치마크 테스트를 시작하는 것이 좋습니다. 인덱스가 매우 큰 경우 큰 필드 데이터 캐시가 필요할 수 있습니다.

최대 절 개수

Lucene 부울 쿼리 하나에 허용되는 최대 절 수를 지정합니다. 기본값은 1,024입니다. 절 수가 허용되는 개수 이상인 쿼리는 `TooManyClauses` 오류를 일으킵니다. 자세한 내용은 Lucene 설명서를 참조하세요.

Amazon OpenSearch Service에서 구성 변경

Amazon OpenSearch Service는 도메인을 업데이트할 때 블루/그린 배포 프로세스를 사용합니다. 블루/그린 배포는 프로덕션 환경을 복제하고 업데이트가 완료되면 사용자를 새 환경으로 라우팅하는 도메인 업데이트용으로 유휴 환경을 만듭니다. 블루/그린 배포에서는 블루 환경이 현재 프로덕션 환경입니다. 그린 환경은 유휴 환경입니다.

데이터는 블루 환경에서 그린 환경으로 마이그레이션됩니다. 새 환경이 준비되면 OpenSearch Service가 환경을 전환하여 그린 환경을 새로운 프로덕션 환경으로 승격할 수 있습니다. 전환은 데이터 손실 없이 이루어집니다. 이렇게 하면 가동 중지가 최소화되고, 새로운 환경에 배포하는 데 실패하더라도 원래의 환경이 유지됩니다.

주제

- 블루/그린 배포의 원인이 되는 변경 사항
- 블루/그린 배포가 발생하지 않는 변경 사항
- 변경 사항으로 인해 블루/그린 배포가 발생하는지 판단
- 구성 변경 추적
- 구성 변경 단계
- 블루/그린 배포의 성능 영향
- 구성 변경 비용
- Troubleshooting validation errors(검증 오류 문제 해결 중)

블루/그린 배포의 원인이 되는 변경 사항

다음 작업에서는 블루/그린 배포가 사용됩니다.

- 인스턴스 유형 변경
- 세분화된 액세스 제어 활성화
- 서비스 소프트웨어 업데이트 수행
- 전용 프라이머리 노드 활성화 또는 비활성화
- Multi-AZ without Standby 활성화 또는 비활성화
- 스토리지 유형, 볼륨 유형 또는 볼륨 크기 변경
- 다른 VPC 서브넷 선택
- VPC 보안 그룹 추가 또는 제거

- 전용 조정자 노드 추가 또는 제거
- OpenSearch 대시보드에서 Amazon Cognito 인증 활성화 또는 비활성화
- 다른 Amazon Cognito 사용자 풀 또는 자격 증명 풀 선택
- 고급 설정 수정
- 새 OpenSearch 버전으로 업그레이드(업그레이드의 모두 또는 일부 중에 OpenSearch 대시보드를 사용할 수 없음)
- 저장된 데이터 암호화 또는 노드 간 암호화 활성화
- UltraWarm 또는 콜드 스토리지 활성화 또는 비활성화
- 자동 조정 사용 중지 및 변경 내용 롤백
- 선택적 플러그인을 도메인에 연결 및 선택적 플러그인을 도메인에서 분리
- 두 개의 전용 마스터 노드를 포함하는 다중 AZ 도메인에 대한 전용 마스터 노드 수 증가
- EBS 볼륨 크기 감소
- 마지막 변경이 진행 중이거나 6시간 이내에 발생한 경우 EBS 볼륨 크기, IOPS 또는 처리량 변경
- CloudWatch에 감사 로그 게시 활성화.

Multi-AZ with Standby 도메인의 경우 한 번에 하나의 변경 요청만 할 수 있습니다. 변경이 이미 진행 중인 경우 새 요청은 거부됩니다. `DescribeDomainChangeProgress` API로 현재 변경의 상태를 확인할 수 있습니다.

블루/그린 배포가 발생하지 않는 변경 사항

대부분의 경우 다음 작업에서는 블루/그린 배포가 사용되지 않습니다.

- 액세스 정책 수정
- 사용자 지정 엔드포인트 수정
- 전송 계층 보안(TLS) 정책 변경
- 자동 스냅샷 시간 변경
- HTTPS 요구 활성화 또는 비활성화
- 변경 사항을 롤백하지 않고 자동 조정 사용 설정 또는 사용 중지
- 도메인에 전용 마스터 노드가 있는 경우 데이터 노드 또는 UltraWarm 노드 개수 변경
- 도메인에 전용 마스터 노드가 있는 경우, 전용 마스터 인스턴스 유형 또는 개수 변경(두 개의 전용 마스터 노드가 있는 다중 AZ 도메인 제외)

- CloudWatch에 오류 로그 또는 느린 로그 게시 활성화 또는 비활성화
- CloudWatch에 감사 로그 게시 비활성화
- 데이터 노드당 최대 3TiB의 볼륨 크기 증가, 볼륨 유형, IOPS 또는 처리량 변경
- 태그 추가 및 삭제

> **Note**
>
> 서비스 소프트웨어 버전에 따라 몇 가지 예외가 있습니다. 변경 사항으로 인해 블루/그린 배포가 발생하지 않다는 점을 확인하려면 도메인을 업데이트하기 전 모의 실습(이 옵션이 사용 가능한 경우)을 수행합니다. 일부 변경은 모의 실행 옵션을 제공하지 않습니다. 일반적으로 피크 트래픽 시간이 아닐 때 클러스터를 변경하는 것이 좋습니다.

변경 사항으로 인해 블루/그린 배포가 발생하는지 판단

몇 가지 유형의 계획된 도메인 구성 변경을 테스트하여 해당 변경을 커밋하지 않고도 블루/그린 배포를 유발하는지 여부를 확인할 수 있습니다. 구성 변경을 시작하기 전에 콘솔 또는 API를 사용하여 검증 확인을 실행하여 도메인을 업데이트할 수 있는지 확인합니다.

Console

구성 변경을 검증하는 방법

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 Domains(도메인)를 선택합니다.
3. 구성을 변경할 도메인을 선택합니다. 그러면 도메인 세부 정보 페이지가 열립니다. Actions(작업) 드롭다운 메뉴를 선택한 다음 Edit cluster configuration(클러스터 구성 편집)을 선택합니다.
4. 인스턴스 유형이나 노드 수를 변경하는 등 도메인을 변경합니다.
5. 드라이 런 분석에서 실행을 선택합니다. 모의 실행은 구성 변경에서 오류를 검증하고 블루/그린 배포가 필요한지 여부를 결정합니다.
6. 모의 실행이 완료되면 페이지 하단에 모의 실행 ID와 함께 결과가 나타납니다. 분석은 구성 변경에 블루/그린 배포가 필요한지 여부를 나타냅니다.

   각 연습 실행은 이전 연습을 덮어씁니다. 각 실행의 세부 정보를 유지하려면 해당 모의 실행 ID를 저장합니다. 드라이 런은 90일 동안 또는 구성 업데이트를 수행할 때까지 사용할 수 있습니다.

7. 구성 업데이트를 계속하려면 Save changes(변경 사항 저장)를 선택합니다. 그렇지 않은 경우 취소를 선택합니다. 둘 중 어느 옵션을 선택하든 Cluster configuration(클러스터 구성) 탭으로 돌아갑니다. 이 탭에서 Dry run details(모의 실습 세부 정보)를 선택하여 최신 모의 실습의 세부 정보를 볼 수 있습니다. 이 페이지에는 모의 실습 전 구성과 모의 실습 구성 간의 나란히 비교도 포함되어 있습니다.

API

구성 API를 통해 모의 실습 검증을 수행할 수 있습니다. API로 변경 사항을 테스트하려면 `DryRun`을 `true`로 설정하고 `DryRunMode`를 `Verbose`로 설정합니다. 상세 표시 모드는 변경 사항이 블루/그린 배포를 시작할지 여부를 결정하는 것 외에도 검증 확인을 실행합니다. 예를 들어 이 UpdateDomainConfig 요청은 UltraWarm을 활성화하여 가져온 배포 유형을 테스트합니다.

```
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"
}
```

요청은 검증 확인을 실행하고 변경으로 인해 발생할 배포 유형을 반환하지만 실제로 업데이트를 수행하지는 않습니다.

```
{
   "ClusterConfig": {
     ...
    },
   "DryRunResults": {
      "DeploymentType": "Blue/Green",
      "Message": "This change will require a blue/green deployment."
    }
}
```

가능한 배포 유형은 다음과 같습니다.

- `Blue/Green` - 변경으로 인해 블루/그린 배포가 발생합니다.
- `DynamicUpdate` - 변경으로 인해 블루/그린 배포가 발생하지 않습니다.
- `Undetermined` - 도메인이 여전히 처리 중 상태이므로 배포 유형을 결정할 수 없습니다.
- `None` - 구성 변경이 없습니다.

검증에 실패하면 검증 실패 목록이 반환됩니다.

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

상태가 여전히 `pending`인 경우 후속 DescribeDryRunProgress 호출에서 UpdateDomainConfig 응답의 모의 실습 ID를 사용하여 검증 상태를 확인할 수 있습니다.

```
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."
    }
}
```

검증 확인 없이 모의 실습 분석을 실행하려면 구성 API를 사용할 때 `DryRunMode`를 `Basic`으로 설정합니다.

Python

다음 Python 코드는 UpdateDomainConfig API를 사용하여 테스트 실행 검증 확인을 수행하고, 검사가 성공하면 테스트 실행 없이 동일한 API를 호출하여 업데이트를 시작합니다. 검사에 실패하면 스크립트는 오류를 출력하고 중지합니다.

```
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)
```

구성 변경 추적

한 번에 하나의 구성 변경을 요청하거나 단일 요청으로 여러 변경 사항을 그룹화할 수 있습니다. 콘솔의 도메인 처리 상태 및 구성 변경 상태 필드를 사용하여 구성 변경을 추적합니다. 추가 변경을 요청하기 `Active` 전에 도메인 상태가 될 때까지 기다립니다.

도메인은 다음과 같은 처리 상태를 가질 수 있습니다.

- `Active` - 구성 변경이 진행 중이지 않습니다. 새 구성 변경 요청을 제출할 수 있습니다.
- `Creating` - 도메인이 생성 중입니다.
- `Modifying` - 새 데이터 노드, EBS, gp3, IOPS 프로비저닝 추가 또는 KMS 키 설정과 같은 구성 변경이 진행 중입니다.
- `Upgrading engine version` - 엔진 버전 업그레이드가 진행 중입니다.
- `Updating service software` - 서비스 소프트웨어 업데이트가 진행 중입니다.
- `Deleting` - 도메인이 삭제되고 있습니다.
- `Isolated` - 도메인이 일시 중지되었습니다.

도메인은 다음과 같은 구성 변경 상태를 가질 수 있습니다.

- `Pending` - 구성 변경 요청이 제출되었습니다.
- `Initializing` - 서비스가 구성 변경을 초기화하고 있습니다.
- `Validating` - 서비스가 요청된 변경 사항과 필요한 리소스를 검증하고 있습니다.
- `Awaiting user inputs` - 서비스는 인스턴스 유형 변경과 같은 구성 변경을 진행할 것으로 예상합니다. 구성 변경 사항을 편집할 수 있습니다.
- `Applying changes` - 서비스가 요청된 구성 변경을 적용하고 있습니다.
- `Cancelled` - 구성 변경이 취소되었습니다. 모든 변경 사항 롤백 취소를 선택합니다.
- `Completed` - 요청된 구성 변경이 성공적으로 완료되었습니다.
- `Validation failed` - 요청된 구성 변경을 완료하지 못했습니다. 구성 변경 사항이 적용되지 않았습니다.

> **Note**
>
> 검증 실패는 도메인에 있는 빨간색 인덱스, 선택한 인스턴스 유형을 사용할 수 없음 또는 디스크 공간 부족의 결과일 수 있습니다. 검증 오류 목록은 the section called “Troubleshooting validation errors(검증 오류 문제 해결 중)” 섹션을 참조하세요. 검증 실패 이벤트 중에 구성 변경을 취소, 재시도 또는 편집할 수 있습니다.

구성 변경이 완료되면 도메인 상태가 다시 로 변경됩니다`Active`.

모두 클러스터 상태와 Amazon CloudWatch 지표를 검토하여 도메인 업데이트가 진행되는 동안 클러스터의 노드 수가 일시적으로 증가하며, 종종 두 배가 되는 것을 확인할 수 있습니다. 다음 그림에 구성 변경 중 노드 수가 11개에서 22개로 두 배가 되었다가 업데이트가 완료되면 11개로 돌아가는 과정이 나와 있습니다.

이렇게 일시적인 증가로 인해 갑자기 관리해야 할 노드가 늘어난 클러스터의 전용 프라이머리 노드는 부담을 받을 수 있습니다. 또한 OpenSearch Service가 이전 클러스터에서 새 클러스터로 데이터를 복사하므로 검색 및 인덱싱 대기 시간이 늘어날 수 있습니다. 그러므로 블루/그린 배포에 따르는 오버헤드를 처리할 수 있을 만큼 클러스터에 충분한 용량을 유지해야 합니다.

> **Important**
>
> 구성 변경 및 서비스 유지 관리 중 추가로 발생하는 비용은 없습니다. 클러스터에 대해 요청한 노드 개수에 대해서만 비용이 청구됩니다. 구체적인 내용은 the section called “구성 변경 비용” 섹션을 참조하세요.

전용 프라이머리 노드의 오버로딩을 방지하기 위해, Amazon CloudWatch 지표를 통해 사용량을 모니터링할 수 있습니다. 권장 최댓값은the section called “권장되는 CloudWatch 경보” 섹션을 참조하세요.

구성 변경 단계

구성 변경을 시작한 후 OpenSearch Service에서는 도메인을 업데이트하는 일련의 단계를 거칩니다. 콘솔의 구성 변경 상태에서 구성 변경 진행 상황을 볼 수 있습니다. 업데이트가 수행되는 정확한 단계는 변경 유형에 따라 다릅니다. DescribeDomainChangeProgress API 작업을 사용하여 구성 변경을 모니터링할 수도 있습니다.

다음은 구성 변경 중에 업데이트가 진행될 수 있는 단계입니다.

단계 이름	설명
검증	도메인을 업데이트할 수 있는지 검증하고 필요한 경우 검증 문제를 표시합니다.
Creating a new environment(새 환경 생성 중)	블루/그린 배포를 시작하기 위해 필요한 필수 구성 요소를 완료하고 필요한 리소스를 생성합니다.

단계 이름	설명
Provisioning new nodes(새 노드 프로비저닝 중)	새로운 환경에서 새 인스턴스 집합 생성
Traffic routing on new nodes(새 노드의 트래픽 라우팅)	새로 생성된 데이터 노드로 트래픽을 리디렉션합니다.
Traffic routing on old nodes(이전 노드의 트래픽 라우팅)	이전 데이터 노드에서 트래픽을 사용 중지합니다.
Preparing nodes for removal(제거할 노드 준비 중)	노드 제거를 준비합니다. 이 단계는 도메인을 다운스케일링하는 경우에만 발생합니다(예: 8개 노드에서 6개 노드로).
Copying shards to new nodes(샤드를 새 노드에 복사 중)	이전 노드에서 새 노드로 샤드를 이동합니다.
Terminating nodes(노드 종료 중)	샤드가 제거된 후 이전 노드를 종료하고 삭제합니다.

단계 이름	설명
Deleting older resources(이전 리소스 삭제 중)	이전 환경과 연결된 리소스를 삭제합니다(예: 로드 밸런서).
Dynamic update(동적 업데이트)	업데이트에 블루/그린 배포가 필요하지 않고 동적으로 적용할 수 있는 경우에 표시됩니다.
전용 마스터 관련 변경 사항 적용	전용 마스터 인스턴스 유형 또는 개수가 변경될 때 표시됩니다.
볼륨 관련 변경 사항 적용	볼륨 크기, 유형, IOPS 및 처리량이 변경될 때 표시됩니다.

블루/그린 배포의 성능 영향

블루/그린 배포 중 수신 검색 및 인덱싱 요청에 대해 Amazon OpenSearch Service 클러스터를 사용할 수 있습니다. 그러나 다음과 같은 성능 문제가 발생할 수 있습니다.

- 클러스터에 관리할 노드가 더 많아지면 리더 노드의 사용량이 일시적으로 증가합니다.
- OpenSearch Service가 이전 노드의 데이터를 새 노드로 복사하여 검색 및 인덱싱 지연 시간이 늘어났습니다.
- 블루/그린 배포 중에 클러스터 로드가 증가하여 수신 요청에 대한 거부가 증가했습니다.
- 지연 시간 문제와 요청 거부를 방지하려면 클러스터가 정상이고 네트워크 트래픽이 낮을 때 블루/그린 배포를 실행해야 합니다.

구성 변경 비용

도메인에 대한 구성을 변경하는 경우 OpenSearch Service는 the section called “구성 변경”에 설명된 대로 새 클러스터를 생성합니다. 새 클러스터로 이전 클러스터를 마이그레이션하는 중 다음 비용이 발생합니다.

- 인스턴스 유형을 변경하면 처음에는 이전 및 새 클러스터 둘 다에 대한 비용이 청구됩니다. 그 이후에는 새 클러스터에 대한 비용만 청구됩니다. EBS 볼륨은 클러스터의 일부이므로 두 번 청구되지 않으며 인스턴스 결제에 따라 요금이 청구됩니다.

  예: `m3.xlarge` 인스턴스 세 개에서 `m4.large` 인스턴스 네 개로 구성을 변경합니다. 첫 1시간은 두 클러스터(3 * `m3.xlarge` + 4 * `m4.large`)에 대한 비용이 청구됩니다. 첫 1시간 이후부터는 새 클러스터(4 * `m4.large`)에 대한 비용만 청구됩니다.
- 인스턴스 유형을 변경하지 않으면 첫 1시간은 가장 큰 클러스터에 대한 비용만 청구됩니다. 첫 1시간 이후부터는 새 클러스터에 대한 비용만 청구됩니다.

  예: `m3.xlarge` 인스턴스 여섯 개에서 `m3.xlarge` 인스턴스 세 개로 구성을 변경합니다. 첫 1시간은 가장 큰 클러스터(6 * `m3.xlarge`)에 대한 비용이 청구됩니다. 첫 1시간 이후부터는 새 클러스터(3 * `m3.xlarge`)에 대한 비용만 청구됩니다.

Troubleshooting validation errors(검증 오류 문제 해결 중)

구성 변경을 시작하거나 OpenSearch 또는 Elasticsearch 버전 업그레이드를 수행하면 OpenSearch Service에서 먼저 일련의 검증 검사를 수행하여 도메인을 업데이트할 수 있는지 확인합니다. 이러한 검사 중 하나라도 실패하면 도메인을 업데이트하기 전에 수정해야 하는 특정 문제가 포함된 알림을 콘솔에서 받게 됩니다. 다음 표에는 OpenSearch Service에서 나타날 수 있는 도메인 문제와 해결 단계가 나열되어 있습니다.

문제	오류 코드	문제 해결 단계
보안 그룹을 찾을 수 없음	`SecurityGroupNotFound`	OpenSearch Service 도메인과 연결된 보안 그룹이 존재하지 않습니다. 이 문제를 해결하려면 지정된 이름으로 보안 그룹을 생성합니다.
서브넷을 찾을 수 없음	`SubnetNotFound`	OpenSearch Service 도메인과 연결된 서브넷이 존재하지 않습니다. 이 문제를 해결하려면 VPC에서 서브넷을 생성합니다.

문제	오류 코드	문제 해결 단계
서비스 연결 역할이 구성되지 않음	`SLRNotConfigured`	OpenSearch Service에 대한 서비스 연결 역할이 구성되지 않았습니다. 서비스 연결 역할은 OpenSearch Service에서 사전 정의하며 서비스에서 사용자를 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다. 역할이 없으면 수동으로 생성해야 할 수 있습니다.
IP 주소가 충분하지 않음	`InsufficientFreeIPsForSubnets`	하나 이상의 VPC 서브넷에 도메인을 업데이트하기에 충분한 IP 주소가 없습니다. 필요한 IP 주소 수를 계산하려면 the section called “VPC 서브넷에서 IP 주소 예약” 섹션을 참조하세요.
Cognito 사용자 풀이 존재하지 않음	`CognitoUserPoolNotFound`	OpenSearch Service에서 Amazon Cognito 사용자 풀을 찾을 수 없습니다. 자격 증명 풀을 생성했고 ID가 올바른지 확인합니다. ID를 찾으려면 Amazon Cognito 콘솔을 이용하거나 다음 AWS CLI 명령을 이용합니다. `aws cognito-idp list-user-pools --max-results 60 --region us-east-1`
Cognito ID 풀이 존재하지 않음	`CognitoIdentityPoolNotFound`	OpenSearch Service에서 Cognito ID 풀을 찾을 수 없습니다. 자격 증명 풀을 생성했고 ID가 올바른지 확인합니다. ID를 찾으려면 Amazon Cognito 콘솔을 이용하거나 다음 AWS CLI 명령을 이용합니다. `aws cognito-identity list-identity-pools --max-results 60 --region us-east-1`
사용자 풀에 대한 Cognito 도메인을 찾을 수 없음	`CognitoDomainNotFound`	사용자 풀에 도메인 이름이 없습니다. Amazon Cognito 콘솔 또는 다음 AWS CLI 명령을 사용하여 구성할 수 있습니다. `aws cognito-idp create-user-pool-domain --domain my-domain --user-pool-id id`

문제	오류 코드	문제 해결 단계
Cognito 역할이 구성되지 않음	`CognitoRoleNotConfigured`	Amazon Cognito 사용자 및 ID 풀을 구성하고 인증에 사용할 수 있는 권한을 OpenSearch Service에 부여하는 IAM 역할이 구성되지 않았습니다. 적절한 권한 세트와 신뢰 관계로 역할을 구성합니다. 기본 CognitoAccessForAmazonOpenSearch 역할을 생성하는 콘솔을 사용하거나 AWS CLI 또는 AWS SDK를 사용하여 역할을 수동으로 구성할 수 있습니다.
사용자 풀을 설명할 수 없음	`UserPoolNotDescribable`	지정된 Amazon Cognito 역할에 도메인과 연결된 사용자 풀을 설명할 권한이 없습니다. 역할 권한 정책이 `cognito-identity:DescribeUserPool` 작업을 허용하는지 확인합니다. 전체 권한 정책은 the section called “CognitoAccessForAmazonOpenSearch 역할 정보” 섹션을 참조하세요.
ID 풀을 설명할 수 없음	`IdentityPoolNotDescribable`	지정된 Amazon Cognito 역할에 도메인과 연결된 ID 풀을 설명할 권한이 없습니다. 역할 권한 정책이 `cognito-identity:DescribeIdentityPool` 작업을 허용하는지 확인합니다. 전체 권한 정책은 the section called “CognitoAccessForAmazonOpenSearch 역할 정보” 섹션을 참조하세요.
사용자 및 ID 풀을 설명할 수 없음	`CognitoPoolsNotDescribable`	지정된 Amazon Cognito 역할에 도메인과 연결된 사용자 및 ID 풀을 설명할 권한이 없습니다. 역할 권한 정책이 `cognito-identity:DescribeIdentityPool` 및 `cognito-identity:DescribeUserPool` 작업을 허용하는지 확인합니다. 전체 권한 정책은 the section called “CognitoAccessForAmazonOpenSearch 역할 정보” 섹션을 참조하세요.
KMS 키가 활성화되지 않음	`KMSKeyNotEnabled`	도메인을 암호화하는 데 사용되는 AWS Key Management Service (AWS KMS) 키가 비활성화되었습니다. 즉시 키를 다시 활성화합니다.

문제	오류 코드	문제 해결 단계
사용자 지정 인증서가 ISSUED(발급됨) 상태가 아님	`InvalidCertificate`	도메인에서 사용자 지정 엔드포인트를 사용하는 경우 AWS Certificate Manager (ACM)에서 SSL 인증서를 생성하거나 자체 인증서를 가져와서 보안을 유지합니다. 인증서가 Issued(발급됨) 상태여야 합니다. 이 오류가 발생하면 ACM 콘솔에서 인증서 상태를 확인합니다. 상태가 Expired(만료됨), Failed(실패), Inactive(비활성) 또는 Pending validation(검증 대기 중)인 경우 ACM 문제 해결 설명서를 참조하여 문제를 해결하세요.
선택한 인스턴스 유형을 시작하기에 용량이 충분하지 않음	`InsufficientInstanceCapacity`	요청한 인스턴스 유형 용량을 사용할 수 없습니다. 예를 들어, 5개의 `i3.16xlarge.search` 노드를 요청했지만 OpenSearch Service에 사용 가능한 `i3.16xlarge.search` 호스트가 충분하지 않아 요청을 이행할 수 없습니다. OpenSearch Service에서 지원되는 인스턴스 유형을 확인하고 다른 인스턴스 유형을 선택합니다.
클러스터의 빨간색 인덱스	`RedCluster`	클러스터에 있는 하나 이상의 인덱스가 빨간색 상태이므로 전체적으로 빨간색 클러스터 상태가 됩니다. 이 문제를 해결하고 수정하려면 the section called “빨간색 클러스터 상태” 섹션을 참조하세요.
메모리 회로 차단기, 요청 너무 많음	`TooManyRequests`	도메인에 대한 검색 및 쓰기 요청이 너무 많아 OpenSearch Service에서 구성을 업데이트할 수 없습니다. 요청 수를 줄이거나, 최대 64GiB RAM까지 인스턴스를 수직으로 확장하거나, 인스턴스를 추가하여 수평으로 확장할 수 있습니다.
새 구성에서 데이터를 보관할 수 없음(디스크 공간 부족)	`InsufficientStorageCapacity`	구성된 스토리지 크기가 도메인의 모든 데이터를 보관할 수 없습니다. 이 문제를 해결하려면 더 큰 볼륨을 선택하거나, 사용하지 않는 인덱스를 삭제하거나, 클러스터의 노드 수를 늘려 즉시 디스크 공간을 확보합니다.

문제	오류 코드	문제 해결 단계

특정 노드에 고정된 샤드	`ShardMovementBlocked`	도메인에 있는 하나 이상의 인덱스가 특정 노드에 연결되어 있으며 재할당할 수 없습니다. 특정 인덱스의 샤드를 호스팅할 수 있는 노드를 지정할 수 있는 샤드 할당 필터링을 구성했기 때문일 수 있습니다.
		
		이 문제를 해결하려면 영향을 받는 모든 인덱스에서 샤드 할당 필터를 제거합니다.
		
		`PUT my-index/_settings`
		`{`
		`  "settings": {`
		`  "index.routing.allocation.require._name": null`
		`  }`
		`}`

새 구성에서 모든 샤드를 보관할 수 없음(샤드 수)	`TooManyShards`	도메인의 샤드가 너무 많아 OpenSearch Service에서 샤드를 새 구성으로 이동할 수 없습니다. 이 문제를 해결하려면 현재 클러스터 노드와 동일한 구성 유형의 노드를 추가하여 도메인을 수평으로 확장합니다. 최대 EBS 볼륨 크기는 노드의 인스턴스 유형에 따라 다릅니다. 앞으로 이 문제를 방지하려면 the section called “샤드 수 선택” 섹션을 참조하고 사용 사례에 적합한 샤딩 전략을 정의합니다.
도메인과 연결된 서브넷이 IPv4 주소를 지원하지 않습니다.	`ResultCodeIPv4BlockNotExists`	이 문제를 해결하려면 도메인의 구성된 IP 주소 유형에 맞게 VPC에서 서브넷을 생성하거나 기존 서브넷을 업데이트하십시오. 도메인에서 IPv4 전용 주소 유형을 사용하는 경우 IPv4 전용 서브넷을 사용하십시오. 도메인에서 듀얼 스택 모드를 사용하는 경우 듀얼 스택 서브넷을 사용하십시오.

문제	오류 코드	문제 해결 단계
도메인과 연결된 서브넷이 IPv6 주소를 지원하지 않습니다.	`ResultCodeIPv6BlockNotExists`	이 문제를 해결하려면 도메인의 구성된 IP 주소 유형에 맞게 VPC에서 서브넷을 생성하거나 기존 서브넷을 업데이트하십시오. 도메인에서 IPv4 전용 주소 유형을 사용하는 경우 IPv4 전용 서브넷을 사용하십시오. 도메인에서 듀얼 스택 모드를 사용하는 경우 듀얼 스택 서브넷을 사용하십시오.

Amazon OpenSearch Service의 서비스 소프트웨어 업데이트

> **Note**
>
> 각 주요(패치 제외) 서비스 소프트웨어 업데이트 시 수행된 변경 사항 및 추가 사항에 대한 설명은 릴리스 정보를 참조하세요.

Amazon OpenSearch Service는 기능을 추가하거나 도메인을 개선하는 서비스 소프트웨어 업데이트를 정기적으로 릴리스합니다. 콘솔의 Notifications(알림) 패널은 업데이트가 있는지 확인하거나 업데이트 상태를 확인하는 가장 쉬운 방법입니다. 각 알림에는 서비스 소프트웨어 업데이트에 대한 세부 정보가 포함됩니다. 모든 서비스 소프트웨어 업데이트는 블루/그린 배포를 사용하여 가동 중단을 최소화합니다.

서비스 소프트웨어 업데이트는 OpenSearch 버전 업그레이드와 다릅니다. OpenSearch의 최신 버전 업그레이드에 대한 자세한 내용은 the section called “도메인 업그레이드” 섹션을 참조하세요.

선택적 업데이트와 필수 업데이트 비교

OpenSearch Service에는 다음의 두 가지 범주의 서비스 소프트웨어 업데이트가 있습니다.

선택적 업데이트

선택적 서비스 소프트웨어 업데이트에는 일반적으로 새로운 특징이나 기능에 대한 개선 사항 및 지원이 포함됩니다. 선택적 업데이트는 도메인에 적용되지 않으며 설치 기한도 정해져 있지 않습니다. 업데이트 사용 가능 여부는 이메일과 콘솔 알림을 통해 전달됩니다. 업데이트를 즉시 적용하도록 선택하거나 더 적절한 날짜 및 시간으로 다시 예약할 수 있습니다. 도메인의 사용량이 적은 기간 동안 일정을 잡을 수도 있습니다. 대부분의 소프트웨어 업데이트는 선택 사항입니다.

업데이트 예약 여부에 관계없이 블루/그린 배포를 유발하는 도메인을 변경하면 OpenSearch Service에서 자동으로 서비스 소프트웨어를 업데이트합니다.

사용량이 적은 기간에 선택적 업데이트를 자동으로 적용하도록 도메인을 구성할 수 있습니다. 이 옵션을 켜면 OpenSearch Service는 선택적 업데이트가 제공될 때로부터 최소 13일을 기다린 다음 72시간(3일) 후에 업데이트를 예약합니다. 업데이트가 예약되면 콘솔 알림을 받게 되며 나중에 업데이트하도록 일정을 조정할 수 있습니다.

자동 소프트웨어 업데이트를 켜려면 도메인을 만들거나 업데이트할 때 자동 소프트웨어 업데이트 활성화를 선택합니다. 를 사용하여 동일한 설정을 구성하려면 도메인을 생성하거나 업데이트할 `true` 때를 `--software-update-options`로 AWS CLI설정합니다.

필수 업데이트

필수 서비스 소프트웨어 업데이트에는 일반적으로 도메인의 지속적인 무결성과 기능을 보장하기 위한 중요한 보안 수정 사항이나 기타 필수 업데이트가 포함됩니다. 필수 업데이트 사항으로는 Log4j Common Vulnerabilities and Exposures(CVEs) 및 Instance Metadata Service Version 2(IMDSv2)의 적용 등이 있습니다. 연간 필수 업데이트 횟수는 보통 3회 미만입니다.

OpenSearch Service는 이러한 업데이트를 자동으로 예약하고 예정된 업데이트 72시간(3일) 전에 이메일과 콘솔 알림을 통해 알려줍니다. 업데이트를 즉시 적용하거나 허용된 기간 내에서 더 적절한 날짜 및 시간으로 업데이트를 다시 예약하도록 선택할 수 있습니다. 도메인의 다음 사용량이 적은 기간 동안 일정을 잡을 수도 있습니다. 필수 업데이트에 대해 아무 조치도 취하지 않고 블루/그린 배포를 야기하는 도메인 변경을 하지 않는 경우 OpenSearch Service는 지정된 기한(일반적으로 사용 가능 후 14일)이 지난 도메인의 사용량이 적은 기간 내에 언제든지 업데이트를 시작할 수 있습니다.

업데이트 예약 여부에 관계없이 블루/그린 배포를 유발하는 도메인을 변경하면 OpenSearch Service에서 자동으로 도메인을 업데이트합니다.

패치 업데이트

“-P”와 숫자로 끝나는 서비스 소프트웨어 버전(예: R20211203-*P4*)은 패치 릴리스입니다. 패치에는 성능 개선, 사소한 버그 수정, 보안 수정 또는 자세 개선이 포함될 수 있습니다. 패치 릴리스에는 새로운 기능이나 주요 변경 사항이 포함되어 있지 않으며 일반적으로 사용자에게 직접적이거나 눈에 띄는 영향을 미치지 않습니다. 서비스 소프트웨어 알림은 패치 릴리스가 선택 사항인지 필수인지 알려줍니다.

고려 사항

도메인 업데이트 여부를 결정할 때는 다음을 고려합니다.

- 도메인을 수동으로 업데이트하면 새로운 기능을 더욱 빠르게 활용할 수 있습니다. 업데이트(Update)를 선택하면 OpenSearch Service가 요청을 대기열에 배치하고 시간이 있을 때 업데이트를 시작합니다.
- 서비스 소프트웨어 업데이트를 시작하면 OpenSearch Service는 업데이트가 시작될 때와 완료될 때 알림을 보냅니다.
- 소프트웨어 업데이트는 블루/그린 배포를 사용하여 가동 중단을 최소화합니다. 업데이트는 클러스터의 전용 프라이머리 노드에 일시적으로 부담을 줄 수 있으므로 관련 오버헤드를 처리할 수 있는 충분한 용량을 유지해야 합니다.
- 업데이트는 일반적으로 몇 분 내에 완료되지만 시스템에 부하가 높은 경우 몇 시간 또는 며칠이 걸릴 수도 있습니다. 업데이트 기간이 길어지지 않도록 구성된 사용량이 적은 기간에 도메인을 업데이트하는 것이 좋습니다.

서비스 소프트웨어 업데이트 시작

OpenSearch Service 콘솔, AWS CLI또는 SDKs.

콘솔

서비스 소프트웨어 업데이트 요청

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 도메인 이름을 선택하여 구성을 엽니다.
3. 실행, 업데이트를 선택하고, 다음 옵션 중 하나를 선택합니다.

   - 지금 업데이트 적용 - 사용 가능한 용량이 있는 경우 현재 시간에 작업이 수행되도록 즉시 예약합니다. 용량을 사용할 수 없는 경우 선택할 수 있는 다른 시간대를 제공합니다.
   - 사용량이 적은 기간에 예약 — 도메인에 사용량이 적은 기간을 활성화한 경우에만 사용할 수 있습니다. 도메인에 구성된 사용량이 적은 기간에 업데이트가 수행되도록 예약합니다. 업데이트가 바로 다음 기간에 적용된다는 보장은 없습니다. 용량에 따라 다음 날에 발생할 수 있습니다. 자세한 내용은 the section called “사용량이 적은 기간” 단원을 참조하십시오.
   - 특정 날짜 및 시간 예약 - 특정 날짜 및 시간에 업데이트가 진행되도록 예약합니다. 용량상의 이유로 지정한 시간을 사용할 수 없는 경우 다른 시간대를 선택할 수 있습니다.

   업데이트를 나중 날짜(도메인의 사용량이 적은 기간 내 또는 외부)로 예약하는 경우 언제든지 다시 일정을 조정할 수 있습니다. 지침은 the section called “작업 일정 조정” 섹션을 참조하세요.
4. 확인을 선택합니다.

AWS CLI

start-service-software-update AWS CLI 요청을 보내 서비스 소프트웨어 업데이트를 시작합니다. 이 예제에서는 업데이트를 대기열에 즉시 추가합니다.

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "NOW"
```

응답:

```
{
    "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**
>
> 업데이트를 요청한 후에는 취소할 수 있는 기간 범위가 줄어들 수 있습니다. 이 PENDING_UPDATE 상태의 지속 시간은와 OpenSearch Service가 수행하는 동시 업데이트 수 AWS 리전 에 따라 크게 달라질 수 있습니다. 업데이트를 취소하려면 콘솔 또는 cancel-service-software-update AWS CLI 명령을 사용합니다.

BaseException를 통한 요청이 실패하면 용량상의 이유로 지정한 시간을 사용할 수 없으므로 다른 시간을 지정해야 합니다. OpenSearch Service는 응답에서 사용 가능한 대체 시간대를 제안합니다.

AWS SDKs

이 샘플 Python 스크립트는의 describe_domain 및 start_service_software_update 메서드를 AWS SDK for Python (Boto3) 사용하여 도메인이 서비스 소프트웨어 업데이트에 적합한지 확인하고, 적합한 경우 업데이트를 시작합니다. domain_name의 값을 제공해야 합니다.

```
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)
```

사용량이 적은 기간에 소프트웨어 업데이트 예약

2023년 2월 16일 이후에 생성된 각 OpenSearch Service 도메인에는 현지 시간으로 오후 10시에서 오전 8시 사이의 일일 10시간 기간이 있으며, 이 기간은 사용량이 적은 기간으로 간주됩니다. OpenSearch Service는 이 기간을 사용하여 도메인에 대한 서비스 소프트웨어 업데이트를 예약합니다. 비수기 업데이트는 트래픽이 많은 기간 동안 클러스터의 전용 프라이머리 노드에 가해지는 부담을 최소화하는 데 도움이 됩니다. OpenSearch Service는 사용자의 동의 없이는 이 10시간 기간 외에는 업데이트를 시작할 수 없습니다.

- 선택적 업데이트의 경우 OpenSearch Service에서 업데이트 사용 가능 여부를 알리고 다가오는 사용량이 적은 기간에 업데이트를 예약하라는 메시지를 표시합니다.
- 필수 업데이트의 경우 OpenSearch Service는 다가오는 사용량이 적은 기간에 자동으로 업데이트를 예약하고 3일 전에 알림을 보냅니다. (사용량이 적은 기간 내 또는 이외 기간의 경우) 업데이트를 완료하는 데 필요한 기간 내에만 업데이트 일정을 조정할 수 있습니다.

각 도메인에 대해 기본 오후 10시 시작 시간을 사용자 지정 시간으로 재정의하도록 선택할 수 있습니다. 지침은 the section called “사용량이 적은 사용자 지정 기간 구성” 섹션을 참조하세요.

콘솔

다가오는 사용량이 적은 기간에 업데이트를 예약하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 도메인 이름을 선택하여 구성을 엽니다.
3. [Actions], [Update Details]를 선택합니다.
4. 사용량이 적은 시간에 예약하기를 선택합니다.
5. 확인을 선택합니다.

사용량이 적은 기간 탭에서 예약된 작업을 확인하고 언제든지 일정을 조정할 수 있습니다. the section called “예약된 작업 보기”을(를) 참조하세요.

CLI

를 사용하여 예정된 사용량이 적은 기간 동안 업데이트를 예약하려면 StartServiceSoftwareUpdate 요청을 AWS CLI보내고 `--schedule-at` 파라미터`OFF_PEAK_WINDOW`에를 지정합니다.

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "OFF_PEAK_WINDOW"
```

서비스 소프트웨어 업데이트 이벤트 모니터링

OpenSearch Service는 서비스 소프트웨어 업데이트를 사용할 수 있거나 필요, 시작, 완료, 실패한 경우 알림을 전송합니다. OpenSearch Service 콘솔의 알림(Notifications) 패널에서 이러한 알림을 확인할 수 있습니다. 알림 심각도는 업데이트가 선택 사항인 경우 `Informational`, 필수인 경우 `High`입니다.

또한, OpenSearch Service는 서비스 소프트웨어 이벤트를 Amazon EventBridge로 보냅니다. EventBridge를 사용하여 이벤트 수신 시 이메일을 보내거나 특정 작업을 수행하는 규칙을 구성할 수 있습니다. 예제 연습은 the section called “자습서: 사용 가능한 업데이트에 대한 SNS 알림 보내기” 섹션을 참조하세요.

Amazon EventBridge로 전송되는 각 서비스 소프트웨어 이벤트 형식은 the section called “서비스 소프트웨어 업데이트 이벤트” 섹션을 참조하세요.

도메인이 업데이트에 적합하지 않은 경우

다음 상태에 있는 도메인은 서비스 소프트웨어 업데이트에 적합하지 않습니다.

상태	설명
처리 중 상태의 도메인	도메인이 구성 변경 도중에 있습니다. 작업이 완료된 후 업데이트 자격을 확인하세요.
빨간색 클러스터 상태	클러스터에서 하나 이상의 인덱스가 빨간색입니다. 문제 해결 단계는 the section called “빨간색 클러스터 상태” 섹션을 참조하세요.
높은 오류율	OpenSearch 클러스터가 요청을 처리하려고 시도할 때 다수의 5xx 오류를 반환합니다. 이 문제는 일반적으로 너무 많은 동시 읽기 또는 쓰기 요청의 결과입니다. 클러스터로 가는 트래픽을 줄이거나 도메인을 확장할 것을 고려하세요.
브레인 분할	브레인 분할은 OpenSearch 클러스터가 여러 개의 프라이머리 노드를 가지고 자체적으로는 절대로 다시 조인되지 않는 2개의 클러스터로 분할되어 있다는 의미입니다. 권장 수의 전용 프라이머리 노드를 사용하면 브레인 분할을 방지할 수 있습니다. 브레인 분할로부터 복구하기 위해 도움이 필요하면 지원에 문의하세요.
Amazon Cognito 통합 문제	도메인은 OpenSearch Dashboards에 대한 인증을 사용하며 OpenSearch Services는 하나 이상의 Amazon Cognito 리소스를 찾을 수 없습니다. 이 문제는 보통 Amazon Cognito 사용자 풀이 없는 경우에 발생합니다. 문제를 수정하려면 누락된 리소스를 다시 생성하고 이를 사용하도록 OpenSearch Service 도메인을 구성합니다.
기타 서비스 문제	OpenSearch Service 자체에 문제가 있을 경우 도메인이 업데이트 자격이 없는 것으로 표시될 수 있습니다. 도메인에 이전 조건이 하나도 적용되지 않지만 문제가 하루를 넘게 지속될 경우 지원에 문의하세요.

Amazon OpenSearch Service의 사용량이 적은 기간 정의

Amazon OpenSearch Service 도메인을 생성할 때는 사용량이 적은 시간으로 간주되는 일일 10시간 기간을 정의합니다. OpenSearch Service는 이 기간을 사용하여 가능하면 트래픽이 비교적 적은 시간에 블루/그린 배포가 필요한 서비스 소프트웨어 업데이트 및 자동 조정 최적화를 예약합니다. 의 경우

에는 도메인 업데이트용으로 새 환경을 만들고 업데이트가 완료되면 사용자를 새 환경으로 라우팅하는 관행에 따릅니다.

블루/그린 배포는 운영 중단이 없지만 블루/그린 배포에 리소스가 소비되는 동안 잠재적으로 성능에 미치는 영향을 최소화하려면 도메인의 구성된 사용량이 적은 기간에 이러한 배포를 예약하는 것이 좋습니다. 노드 교체와 같은 업데이트나 도메인에 즉시 배포해야 하는 업데이트는 사용량이 적은 기간을 사용하지 마하세요.

사용량이 적은 기간의 시작 시간은 수정할 수 있지만 기간의 길이는 수정할 수 없습니다.

> **Note**
>
> 사용량이 적은 기간은 2023년 2월 16일에 도입되었습니다. 이 날짜 이전에 생성된 모든 도메인은 사용량이 적은 기간이 기본적으로 비활성화되어 있습니다. 이러한 도메인의 사용량이 적은 기간을 수동으로 활성화하고 구성해야 합니다. 이 날짜 이후에 생성된 모든 도메인은 사용량이 적은 기간이 기본적으로 비활성화되어 있습니다. 도메인의 사용량이 적은 기간을 활성화한 후에는 비활성화할 수 없습니다.

사용량이 적은 기간 서비스 소프트웨어 업데이트

OpenSearch Service에는 선택형 및 필수라는 두 가지 범주의 서비스 소프트웨어 업데이트가 있습니다. 두 유형 모두 블루/그린 배포가 필요합니다. 선택적 업데이트는 도메인에 적용되지 않지만, 지정된 기한(일반적으로 출시 후 2주) 이전에 조치를 취하지 않으면 필수 업데이트가 자동으로 설치됩니다. 자세한 내용은 the section called “선택적 업데이트와 필수 업데이트 비교” 단원을 참조하십시오.

선택적 업데이트를 시작할 때는 업데이트를 즉시 적용하거나, 다음 사용량이 적은 기간으로 일정을 잡거나, 사용자 지정 날짜 및 시간을 지정하여 적용할 수 있습니다.

필수 업데이트의 경우 OpenSearch Service는 사용량이 적은 시간에 업데이트를 수행할 날짜 및 시간을 자동으로 예약합니다. 예정된 업데이트 3일 전에 알림을 받게 되며, 필요한 배포 기간 내에 나중에 업데이트하도록 일정을 조정할 수 있습니다. 지침은 the section called “작업 일정 조정” 섹션을 참조하세요.

사용량이 적은 자동 조정 최적화

이전에 자동 조정은 유지 관리 기간을 사용하여 블루/그린 배포가 필요한 변경 일정을 잡았습니다. 사용량이 적은 기간이 도입되기 전에 이미 자동 조정 및 유지 관리 기간을 사용하도록 설정한 도메인은

사용량이 적은 기간을 사용하도록 마이그레이션하지 않는 한 이러한 업데이트에 대한 유지 관리 기간을 계속 사용합니다.

서비스 소프트웨어 업데이트와 같은 도메인에서의 다른 활동을 예약하는 데 사용되므로 사용량이 적은 기간을 사용하여 도메인을 마이그레이션하는 것이 좋습니다. 지침은 the section called “자동 조정 유지 관리 기간에서 마이그레이션하기” 섹션을 참조하세요. 사용량이 적은 기간으로 도메인을 마이그레이션한 후에는 유지 관리 기간을 다시 사용하도록 되돌릴 수 없습니다.

2023년 2월 16일 이후에 생성된 모든 도메인은 레거시 유지 관리 기간 대신 사용량이 적은 기간을 사용하여 블루/그린 배포를 예약합니다. 도메인의 사용량이 적은 기간을 비활성화할 수 없습니다. 블루/그린 배포가 필요한 자동 조정 최적화 목록은 the section called “변경 유형”(을)를 참조하세요.

사용량이 적은 시간 활성화하기

사용량이 적은 기간이 도입된 2023년 2월 16일 이전에 생성된 모든 도메인은 기본적으로 이 기능이 비활성화되어 있습니다. 이러한 도메인에는 수동으로 활성화해야 합니다. 사용량이 적은 기간을 활성화한 후에는 비활성화할 수 없습니다.

콘솔

도메인의 사용량이 적은 기간을 활성화하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 도메인 이름을 선택하여 구성을 엽니다.
3. 사용량이 적은 기간 탭으로 이동하여 편집을 선택합니다.
4. 시간을 협정 세계시(UTC)로 지정합니다. 예를 들어, 미국 서부(오레곤) 지역에서 시작 시간을 오후 11시 30분으로 구성하려면 07:30을 지정합니다.
5. 변경 사항 저장을 선택합니다.

CLI

를 사용하여 사용량이 적은 기간을 수정하려면 UpdateDomainConfig 요청을 AWS CLI보냅니다.

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'Enabled=true,
 OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

사용자 지정 기간 시작 시간을 지정하지 않는 경우 기본값은 00:00 UTC입니다.

사용량이 적은 사용자 지정 기간 구성

도메인의 사용량이 적은 사용자 지정 기간은 UTC(협정 세계시)를 기준으로 지정합니다. 예를 들어 사용량이 적은 기간은 미국 동부(버지니아 북부) 지역의 도메인에서 오후 11시에 시작되도록 하려면 04:00 UTC를 지정합니다.

콘솔

도메인의 사용량이 적은 기간을 활성화하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 도메인 이름을 선택하여 구성을 엽니다.
3. 사용량이 적은 기간 탭으로 이동합니다. 구성된 사용량이 적은 기간과 도메인의 예정된 작업 목록을 볼 수 있습니다.
4. 편집을 선택하고 새 시작 시간을 UTC로 지정합니다. 예를 들어, 미국 동부(버지니아 북부) 지역에서 시작 시간을 오후 9시로 구성하려면 02:00 UCT를 지정합니다.
5. 변경 사항 저장을 선택합니다.

CLI

를 사용하여 사용량이 적은 사용자 지정 기간을 구성하려면 UpdateDomainConfig 요청을 AWS CLI보내고 24시간 형식으로 시간과 분을 지정합니다.

예를 들어, 다음 요청은 기간 시작 시간을 UTC 기준 오전 2:00로 변경합니다.

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

기간 시작 시간을 지정하지 않는 경우 도메인이 생성된 AWS 리전 의 기본 시간은 현지 시간으로 오후 10시입니다.

예약된 작업 보기

각 도메인에 대해 현재 예약되어 있거나 진행 중이거나 보류 중인 모든 작업을 볼 수 있습니다. 작업의 심각도는HIGH, MEDIUM, LOW일 수 있습니다.

파이프라인은 다음과 같은 상태일 수 있습니다.

- `Pending update`— 작업이 처리될 대기열에 있습니다.
- `In progress`— 작업이 현재 진행 중입니다.
- `Failed` - 작업을 완료하지 못했습니다.
- `Completed` – 작업이 성공적으로 완료되었습니다.
- `Not eligible`— 서비스 소프트웨어 업데이트에만 해당됩니다. 클러스터가 비정상 상태이므로 업데이트를 진행할 수 없습니다.
- `Eligible`— 서비스 소프트웨어 업데이트에만 해당됩니다. 도메인은 업데이트할 수 있습니다.

콘솔

OpenSearch Service 콘솔에는 도메인 구성 내에서 예약된 모든 작업이 각 작업의 심각도 및 현재 상태와 함께 표시됩니다.

도메인에 대한 예약된 작업 보기

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 도메인 이름을 선택하여 구성을 엽니다.
3. 사용량이 적은 기간 탭으로 이동합니다.
4. 예약된 작업에서 도메인에 대해 현재 예약되어 있거나 진행 중이거나 보류 중인 모든 작업을 볼 수 있습니다.

CLI

를 사용하여 예약된 작업을 보려면 ListScheduledActions 요청을 AWS CLI보냅니다.

```
aws opensearch list-scheduled-actions \
  --domain-name my-domain
```

응답:

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


작업 일정 조정

OpenSearch Service는 예정된 서비스 소프트웨어 업데이트 및 자동 조정 최적화를 알려줍니다. 변경 사항을 즉시 적용하도록 선택하거나 나중 날짜 및 시간으로 다시 예약할 수 있습니다.

> **Note**
>
> OpenSearch Service는 선택한 시간으로부터 1시간 이내에 작업을 예약할 수 있습니다. 예를 들어, 오후 5시에 업데이트를 적용하도록 선택하면 오후 5시에서 6시 사이에 업데이트를 적용할 수 있습니다.

콘솔

작업 일정 변경

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 도메인 이름을 선택하여 구성을 엽니다.

3. 사용량이 적은 기간 탭으로 이동합니다.
4. 예약된 작업을 선택한 다음, 작업, 삭제를 선택합니다.
5. 다음 옵션 중 하나를 선택하세요.

   - 지금 업데이트 적용 - 사용 가능한 용량이 있는 경우 현재 시간에 작업이 수행되도록 즉시 예약합니다. 용량을 사용할 수 없는 경우 선택할 수 있는 다른 시간대를 제공합니다.
   - 사용량이 적은 기간에 예약 - 다가오는 사용량이 적은 기간에 작업을 픽업하도록 표시합니다. 변경 사항이 바로 다음 기간에 적용된다는 보장은 없습니다. 용량에 따라 다음 날에 발생할 수 있습니다.
   - 이 업데이트 일정 조정 - 변경 사항을 적용할 사용자 지정 날짜 및 시간을 지정할 수 있습니다. 용량상의 이유로 지정한 시간을 사용할 수 없는 경우 다른 시간대를 선택할 수 있습니다.
   - 예약 업데이트 취소 - 업데이트를 취소합니다. 이 옵션은 선택적 서비스 소프트웨어 업데이트에만 사용할 수 있습니다. 자동 조정 작업 또는 필수 소프트웨어 업데이트에는 사용할 수 없습니다.
6. 변경 사항 저장을 선택합니다.

CLI

를 사용하여 작업을 다시 예약하려면 UpdateScheduledAction 요청을 AWS CLI전송합니다. 작업 ID를 검색하려면 `ListScheduledActions` 요청을 보내세요.

다음 요청은 서비스 소프트웨어 업데이트를 특정 날짜 및 시간으로 다시 예약합니다.

```
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
```

응답:

```
{
   "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"
   }
}
```

`SlotNotAvailableException`를 통한 요청이 실패하면 용량상의 이유로 지정한 시간을 사용할 수 없으므로 다른 시간을 지정해야 합니다. OpenSearch Service는 응답에서 사용 가능한 대체 시간대를 제안합니다.

자동 조정 유지 관리 기간에서 마이그레이션하기

도메인이 2023년 2월 16일 이전에 생성된 경우 유지 관리 기간을 사용하여 블루/그린 배포가 필요한 자동 조정 최적화를 예약할 수 있습니다. 사용량이 적은 기간을 대신 사용하도록 기존의 자동 조정 도메인을 마이그레이션할 수 있습니다.

> **Note**
>
> 사용량이 적은 기간으로 도메인을 마이그레이션한 후에는 유지 관리 기간을 다시 사용하도록 되돌릴 수 없습니다.

콘솔

사용량이 적은 기간을 사용하도록 도메인을 마이그레이션하려면

1. Amazon OpenSearch Service 콘솔에서 도메인 이름을 선택하여 해당 구성을 엽니다.
2. 자동 조정 탭으로 이동하여 편집을 선택합니다.
3. 사용량이 적은 기간으로 마이그레이션을 선택합니다.
4. 시작 시간(UTC)의 경우 사용량이 적은 기간의 일일 시작 시간을 협정 세계시(UTC)로 입력합니다.
5. 변경 사항 저장을 선택합니다.

CLI

를 사용하여 자동 조정 유지 관리 기간에서 사용량이 적은 기간으로 마이그레이션하려면 UpdateDomainConfig 요청을 AWS CLI전송합니다.

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=[]
```

자동 조정 유지 관리 기간에서 사용량이 적은 기간으로 도메인을 마이그레이션하려면 사용량이 적은 기간을 켜야 합니다. 사용량이 적은 기간은 별도의 요청이나 동일한 요청으로 활성화할 수 있습니다. 지침은 the section called “사용량이 적은 시간 활성화하기” 단원을 참조하세요.

Amazon OpenSearch Service의 알림

Amazon OpenSearch Service의 알림에는 도메인의 성능 및 상태에 대한 중요한 정보가 포함되어 있습니다. OpenSearch Service는 서비스 소프트웨어 업데이트, 자동 조정 기능 향상, 클러스터 상태 이벤트, 도메인 오류에 대해 알려줍니다. OpenSearch 및 Elasticsearch OSS의 모든 버전에서 알림을 사용할 수 있습니다.

OpenSearch Service 콘솔의 Notifications(알림) 패널에서 이러한 알림을 확인할 수 있습니다. OpenSearch Service에 대한 모든 알림은 Amazon EventBridge에도 표시됩니다. 알림 및 샘플 이벤트의 전체 목록은 the section called “이벤트 모니터링” 섹션을 참조하세요.

알림 시작하기

도메인을 만들 때 알림이 자동으로 활성화됩니다. OpenSearch Service 콘솔의 알림(Notifications) 패널로 이동하여 알림을 모니터링 및 확인합니다. 각 알림에는 게시된 시간, 관련 도메인, 심각도 및 상태 수준, 간단한 설명 등의 정보가 포함됩니다. 콘솔에서 최대 90일 동안의 기간별 알림을 볼 수 있습니다.

[알림(Notifications)] 패널에 액세스하거나 알림을 승인한 후, `es:ListNotificationsV2` 또는 `es:UpdateNotificationStatus`를 수행할 권한이 없다는 오류 메시지가 표시될 수 있습니다. 이 문제를 해결하려면 IAM에서 사용자 또는 역할에 다음 권한을 부여합니다.

JSON

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

```
      "es:ListDomainNames"
    ],
    "Resource": "arn:aws:es:*:111122223333:domain/*"
  }]
}
```

IAM 콘솔에서 무시해도 괜찮은 오류('IAM이 하나 이상의 작업을 인식하지 못합니다(IAM does not recognize one or more actions)')가 발생합니다. 또한 `es:UpdateNotificationStatus` 작업을 특정 도메인으로 제한할 수 있습니다. 자세한 내용은 the section called "정책 요소 참조"를 참조하세요.

알림 심각도

OpenSearch Service의 알림은 이미 취한 조치 또는 도메인의 운영과 관련한 정보 알림이거나 필수 보안 패치를 적용하는 등의 특정한 조치를 요하는 조치 가능 알림입니다. 각 알림에는 심각도(`Informational`, `Low`, `Medium`, `High` 또는 `Critical`)가 연결되어 있습니다. 다음 표에는 각 심각도가 요약되어 있습니다.

심각도	설명	예제
`Informational`	도메인 운영과 관련된 정보입니다.	• 서비스 소프트웨어 업데이트 사용 가능 • 자동 조정 시작
`Low`	권장되는 작업이지만 조치를 하지 않을 경우 도메인 가용성이나 성능에 부정적인 영향을 미치지 않습니다.	• 자동 조정 취소 • 샤드 수 높음 경고
`Medium`	권장 조치를 하지 않을 경우 영향이 있을 수 있지만 조치를 할 수 있는 기간이 연장됩니다.	• 서비스 소프트웨어 업데이트 실패 • 샤드 수 제한 초과됨
`High`	악영향을 피하기 위해서는 긴급한 조치가 필요합니다.	• 서비스 소프트웨어 업데이트 필요 • KMS 키에 액세스할 수 없음
`Critical`	악영향을 피하거나 복구하려면 즉각적인 조치가 필요합니다.	현재 사용할 수 없음

샘플 EventBridge 이벤트

다음 예제에서는 Amazon EventBridge로 전송된 OpenSearch Service 알림 이벤트를 보여줍니다. 업데이트는 선택 사항이기 때문에 알림의 심각도는 `Informational`입니다.

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

Amazon OpenSearch Service에서 다중 AZ 도메인 구성

서비스 중단 시 데이터 손실을 방지하고 Amazon OpenSearch Service 클러스터 가동 중지 시간을 최소화하기 위해 동일한 리전에 있는 두 개 또는 세 개의 가용 영역에 노드를 분산할 수 있습니다. 이를 다중 AZ 구성이라고 합니다. 가용 영역은 각 AWS 리전 내에서 격리된 위치입니다.

프로덕션 워크로드를 실행하는 도메인의 경우 다음과 같은 구성을 생성하는 Multi-AZ with Standby 배포 옵션을 사용하는 것이 좋습니다.

- 세 개 영역에 배포된 도메인.
- 전용 프라이머리 노드와 데이터 노드에 대해 최신 세대 인스턴스 유형.
- 세 개의 전용 프라이머리 노드와 세 개(또는 3의 배수)의 데이터 노드.
- 도메인의 각 인덱스에 대해 최소 두 개의 복제본 또는 세 개의 데이터 사본(기본 노드와 복제본 모두 포함)의 배수

이 단원의 나머지 부분에서는 이러한 구성에 대한 설명과 전후 관계를 제공합니다.

Multi-AZ with Standby

Multi-AZ with Standby는 99.99% 가용성, 프로덕션 워크로드를 위한 일관된 성능, 간소화된 도메인 구성 및 관리를 제공하는 Amazon OpenSearch Service 도메인용 배포 옵션입니다. Multi-AZ with Standby를 사용하면 성능이나 가용성에 영향을 주지 않고 도메인이 인프라 장애에도 복원력이 뛰어납니다. 이 배포 옵션은 지정된 데이터 노드 수, 프라이머리 노드 수, 인스턴스 유형, 복제본 수, 소프트웨어 업데이트 설정, 자동 조정 켜기 등 여러 모범 사례를 의무화하여 이 표준을 충족합니다.

> **Note**
>
> 대기 모드가 있는 다중 AZ는 OpenSearch 버전 1.3 이상에서 사용할 수 있으며 가용 영역이 3개 이상인 리전이 필요합니다.

Multi-AZ with Standby로 사용하는 경우 OpenSearch Service는 세 가용 영역에 도메인을 생성하며, 각 영역에는 전체 데이터 사본이 포함되고 데이터는 각 영역에 균등하게 분산됩니다. 도메인은 이러한 영역 중 하나에 있는 노드를 대기 모드로 예약하므로 검색 요청을 처리하지 않습니다. OpenSearch Service는 기본 인프라에서 장애를 감지하면 1분 이내에 대기 노드를 자동으로 활성화합니다. 도메인은 계속해서 인덱싱 및 검색 요청을 처리하므로 장애 조치를 수행하는 데 걸리는 시간으로 영향이 제한됩니다. 데이터나 리소스를 재분배하지 않으므로 클러스터 성능에 영향을 주지 않고 가용성이 저하될 위험도 없습니다. Multi-AZ with Standby는 추가 비용 없이 사용할 수 있습니다.

AWS Management Console에서 대기 모드로 도메인을 생성할 수 있는 두 가지 옵션이 있습니다. 먼저, 간편 생성 방법으로 도메인을 생성할 수 있으며, OpenSearch Service는 다음을 포함하는 미리 결정된 구성을 자동으로 사용합니다.

- 세 개의 가용 영역(하나는 대기 모드로 사용)
- 세 개의 전용 프라이머리 노드 및 데이터 노드
- 도메인에서 활성화된 자동 조정
- 데이터 노드용 GP3 스토리지

표준 생성 방법을 선택하고 배포 옵션으로 대기 모드가 있는 도메인을 선택할 수도 있습니다. 이렇게 하면 영역 3개와 프라이머리 노드 3개와 같은 대기 모드의 주요 기능은 그대로 유지하면서 도메인을 사용자 지정할 수 있습니다. 데이터 노드 수를 3의 배수(가용 영역 수)로 선택하는 것이 좋습니다.

도메인을 생성한 후에는 도메인 세부 정보 페이지로 이동하여 클러스터 구성 탭에서 가용 영역 아래에 대기 모드가 있는 3-AZ가 나타나는지 확인할 수 있습니다.

기존 도메인을 Multi-AZ with Standby로 마이그레이션하는 데 문제가 있는 경우 문제 해결 안내서에서 대기 모드로 Multi-AZ with Standby로의 마이그레이션 오류를 참조하세요.

제한 사항

Multi-AZ with Standby로 도메인을 설정할 때는 다음 제한 사항을 고려하세요.

- 노드의 총 샤드 수는 1000개를 초과할 수 없고, 클러스터의 총 샤드 수는 75000개를 초과할 수 없으며, 단일 샤드의 크기는 65GB를 초과할 수 없습니다.
- 대기 모드가 있는 다중 AZ는 `m5`, , `c5`, `r5`, `r6gr7g`, 및 `c6g` `m6g` `r6gd` `i3` 인스턴스 유형에서만 작동합니다. 지원되는 인스턴스에 대한 자세한 내용은 지원되는 인스턴스 유형을 참조하세요.
- 프로비저닝된 IOPS SSD, 범용 SSD(GP3) 또는 대기 모드가 있는 인스턴스 지원 스토리지만 사용할 수 있습니다.
- Multi-AZ with Standby 도메인에서 에서 UltraWarm을 활성화하는 경우 웜 노드 수는 사용 중인 가용 영역 수의 배수여야 합니다.

Multi-AZ without Standby

OpenSearch Service는 99.9% 가용성을 제공하는 multi-AZ without Standby를 계속 지원합니다. 노드는 가용 영역 전체에 분산되어 있으며 가용성은 가용 영역 수와 데이터 사본에 따라 달라집니다. 대기 모드에서는 모범 사례에 따라 도메인을 구성해야 하지만 대기 모드 외에서는 가용 영역, 노드, 복제본 수를 직접 선택할 수 있습니다. 대기 모드가 있는 도메인을 생성하여 중단될 수 있는 기존 워크플로가 있는 경우가 아니면 이 옵션을 사용하지 않는 것이 좋습니다.

이 옵션을 선택하는 경우에도 노드, 디스크 및 단일 AZ 장애에 대한 복원력을 유지하려면 가용 영역 3개를 선택하는 것이 좋습니다. 장애가 발생하면 클러스터는 가용성과 중복성을 유지하기 위해 나머지 리소스에 데이터를 재분배합니다. 이러한 데이터 이동은 클러스터의 리소스 사용량을 증가시키고 성능에 영향을 미칠 수 있습니다. 클러스터 크기가 적절하지 않으면 가용성이 저하될 수 있으며, 이는 다중 AZ의 목적을 크게 저해합니다.

에서 대기 없이 도메인을 구성하는 유일한 방법은 표준 생성 방법을 AWS Management Console 선택하고 배포 옵션으로 대기 없이 도메인을 선택하는 것입니다.

샤드 배포

Multi-AZ without Standby를 활성화하는 경우, 클러스터의 인덱스당 복제본을 한 개 이상 생성해야 합니다. 복제본이 없으면 OpenSearch Service가 다른 가용 영역에 데이터 사본을 배포할 수 없습니다.

다행히 모든 인덱스의 기본 구성은 복제본 1개입니다. 다음 다이어그램에서 볼 수 있듯이 OpenSearch Service에서는 기본 샤드와 해당 복제본 샤드를 최대한 서로 다른 영역에 분산합니다.

가용 영역별로 샤드를 분산시키는 것 외에도 OpenSearch Service는 샤드를 노드별로 분산시킵니다. 그러나 특정 도메인 구성은 샤드 수가 불균형해질 수 있습니다. 다음 도메인을 생각해 보세요.

- 데이터 노드 5개
- 기본 샤드 5개
- 복제본 2개
- 가용 영역 3개

이 경우, OpenSearch는 다음 다이어그램과 같이 기본 샤드와 복제본 샤드를 영역에 분산시키기 위해 한 개의 노드를 오버로드해야 합니다.

개별 노드의 부하를 늘리고 성능을 저하시킬 수 있는 이러한 상황을 피하려면, 인덱스당 두 개 이상의 복제본을 사용하려는 경우 Multi-AZ with Standby를 선택하거나 인스턴스 수를 3의 배수로 선택하는 것이 좋습니다.

전용 프라이머리 노드 분산

도메인을 구성할 때 가용 영역을 두 개 선택하더라도 OpenSearch Service는 전용 프라이머리 노드를 세 개의 가용 영역에 자동으로 분산시킵니다. 이러한 분산은 영역에 서비스 중단이 발생할 경우 클러스터 가동 중지를 방지합니다. 권장되는 세 개의 전용 프라이머리 노드를 사용하면 하나의 가용 영역이 중단되더라도 클러스터가 여전히 전용 프라이머리 노드의 쿼럼(2)을 유지하므로 새 마스터를 선택할 수 있습니다. 다음 다이어그램은 이 구성을 보여줍니다.

세 개의 가용 영역에서 사용할 수 없는 전 세대 인스턴스 유형을 선택하는 경우 다음 시나리오가 적용됩니다.

- 도메인에 가용 영역을 세 개 선택하면 OpenSearch Service가 오류를 발생시킵니다. 다른 인스턴스 유형을 선택하고 다시 시도하세요.
- 도메인에 가용 영역을 두 개 선택하면 OpenSearch Service가 두 개의 영역에 전용 프라이머리 노드를 분산시킵니다.

가용 영역 중단

가용 영역 중단은 드문 경우지만 발생할 수 있습니다. 다음 표에는 중단 시 다양한 다중 AZ 구성과 동작이 나와 있습니다. 표의 마지막 행은 Multi-AZ with Standby에 적용되는 반면, 다른 모든 행은 Multi-AZ without Standby에만 적용되는 구성을 포함합니다.

리전의 가용 영역 수	선택한 가용 영역 수	전용 프라이머리 노드 수	한 개의 가용 영역에 중단이 발생할 경우의 동작
2 이상	2	0	가동 중지. 클러스터에서 데이터 노드의 절반이 손실되고 마스터를 선택하기 전에 가용 영역에서 하나 이상의 노드를 교체해야 합니다.
2	2	3	50/50의 가동 중지 확률. OpenSearch Service는 하나의 가용 영역에 두 개의 전용 프라이머리 노드를 분산시키고 하나를 다른 영역에 분산시킵니다. • 하나의 전용 프라이머리 노드가 있는 가용 영역에 장애가 발생하면 나머지 가용 영역에 있는 두 개의 전용 프라이머리 노드가 마스터로 선택될 수 있습니다. • 두 개의 전용 프라이머리 노드가 있는 가용 영역에 장애가 발생하면 나머지 가용 영역이 복구될 때까지 클러스터를 사용할 수 없습니다.
3 이상	2	3	가동 중지 없음. OpenSearch Service는 세 개의 가용 영역에 전용 프라이머리 노드를 자동으로 분산시키므로 나머지 두 개의 전용 프라이머리 노드가 마스터로 선택될 수 있습니다.
3 이상	3	0	가동 중지 없음. 약 3분의 2의 데이터 노드가 여전히 마스터로 선택될 수 있습니다.
3 이상	3	3	가동 중지 없음. 나머지 두 개의 전용 프라이머리 노드가 마스터로 선택될 수 있습니다.

모든 구성에서 원인과 상관없이 노드 장애는 클러스터의 나머지 데이터 노드에서 일정 기간에 부하 증가를 유발하며, OpenSearch Service는 현재 누락된 노드를 교체하도록 새 노드를 자동으로 구성할 수 있습니다.

예를 들어, 3개 영역 구성에서 가용 영역 장애가 발생하는 경우 데이터 노드의 최대 2/3가 클러스터에 대한 요청을 최대한 많이 처리해야 합니다. 이에 따라 나머지 노드들도 온라인 상태가 될 때 새 노드에 샤드를 복제하므로 성능에 더 큰 영향을 미칠 수 있습니다. 워크로드에 가용성이 중요한 경우 이 문제를 최소화하기 위해 클러스터에 리소스를 추가하는 것을 고려합니다.

> **Note**
>
> OpenSearch Service에서는 다중 AZ 도메인을 투명하게 관리하므로 가용 영역 중단을 수동으로 시뮬레이션할 수 없습니다.

VPC 내에서 Amazon OpenSearch Service 도메인 시작

Amazon OpenSearch Service 도메인과 같은 AWS 리소스를 Virtual Private Cloud(VPC)로 시작할 수 있습니다. VPC는 전용 가상 네트워크입니다 AWS 계정. AWS 클라우드의 다른 가상 네트워크와 논리적으로 격리되어 있습니다. OpenSearch Service 도메인을 VPC 안에 배치하면 인터넷 게이트웨이, NAT 디바이스 또는 VPN 연결 없이 VPC 내부에서 OpenSearch Service와 다른 서비스 간에 보안 통신이 가능합니다. 모든 트래픽은 AWS 클라우드 내에서 안전하게 유지됩니다.

> **Note**
>
> VPC에 OpenSearch Service 도메인을 배치하는 경우 컴퓨터가 VPC에 연결할 수 있어야 합니다. 이 연결은 종종 VPN, Transit Gateway, 관리형 네트워크 또는 프록시 서버의 형식을 따릅니다. VPC 밖에서는 도메인에 직접 액세스할 수 없습니다.

VPC 대 퍼블릭 도메인

다음은 VPC 도메인과 퍼블릭 도메인의 몇 가지 차이점입니다. 각 차이점은 추후 보다 자세히 설명합니다.

- 논리적 격리로 인해 퍼블릭 엔드포인트를 사용할 때에 비해, VPC에 상주하는 도메인에는 보안 계층이 하나 추가됩니다.

- 퍼블릭 도메인은 인터넷에 연결된 디바이스에서 액세스할 수 있지만 VPC 도메인에는 특정 형태의 VPN 또는 프록시가 필요합니다.
- 퍼블릭 도메인과 비교하면, VPC 도메인은 콘솔에 더 적은 정보를 표시합니다. 특히 클러스터 상태(Cluster health) 탭에는 샤드 정보가 포함되지 않으며 인덱스(Indices) 탭은 표시되지 않습니다.
- 도메인 엔드포인트의 형식이 다릅니다(`https://search-domain-name` 대 `https://vpc-domain-name`).
- 보안 그룹은 이미 IP 기반 액세스 정책을 사용하므로 VPC에 상주하는 도메인에 IP 기반 액세스 정책을 적용할 수는 없습니다.

제한 사항

현재, VPC에서 OpenSearch Service 도메인을 운영할 경우 다음과 같은 제약이 있습니다.

- VPC에서 새 도메인을 시작할 경우 나중에 해당 도메인이 퍼블릭 엔드포인트를 사용하도록 변경할 수 없습니다. 반대도 마찬가지입니다. 퍼블릭 엔드포인트를 사용하여 도메인을 만들 경우 나중에 해당 도메인을 VPC 안에 배치할 수 없습니다. 대신에 새로운 도메인을 만들어 데이터를 마이그레이션해야 합니다.
- VPC에서 도메인을 시작할 수도 있고 퍼블릭 엔드포인트를 사용할 수도 있지만 두 방법을 동시에 사용할 수는 없습니다. 도메인을 만들 때 한 가지를 선택해야 합니다.
- 전용 테넌시를 사용하는 VPC 내에서 도메인을 실행할 수 없습니다. Default로 설정된 테넌시와 함께 VPC를 사용해야 합니다.
- 도메인을 VPC 내부에 배치한 후 다른 VPC로 이전할 수 없지만 서브넷과 보안 그룹 설정은 변경할 수 있습니다.
- VPC에 상주하는 도메인에 대한 OpenSearch 대시보드의 기본 설치에 액세스하려면 사용자가 VPC에 액세스할 수 있어야 합니다. 이 프로세스는 네트워크 구성에 따라 다르지만, VPN 또는 관리형 네트워크에 연결하거나 프록시 서버 또는 Transit Gateway를 사용해야 할 수 있습니다. 자세한 내용은 the section called “VPC 도메인 액세스 정책에 대하여”, Amazon VPC 사용 설명서 및 the section called “대시보드에 대한 액세스 제어 ” 섹션을 참조하세요.

아키텍처

VPC를 지원하기 위해 OpenSearch Service는 엔드포인트를 사용자의 VPC의 서브넷 1개, 2개 또는 3개에 배치합니다. 도메인에서 다중 가용 영역을 활성화하려는 경우 동일한 리전의 다른 가용 영역에 각 서브넷이 있어야 합니다. 가용 영역을 한 개만 사용하는 경우 OpenSearch Service가 엔드포인트를 서브넷 한 개에만 배치합니다.

다음 그림은 가용 영역 한 개에 대한 VPC 아키텍처를 보여줍니다.

다음 그림은 가용 영역 두 개에 대한 VPC 아키텍처를 보여줍니다.

또한 OpenSearch Service는 각 데이터 노드에 대해 VPC에 탄력적 네트워크 인터페이스(ENI)를 VPC에 설치합니다. OpenSearch Service는 ENI마다 서브넷의 IPv4 주소 범위에서 프라이빗 IP 주소를 할당합니다. 또한 IP 주소에 퍼블릭 DNS 호스트 이름(도메인 엔드포인트)을 할당합니다. 퍼블릭 DNS 서비스를 사용하여 엔드포인트(즉 DNS 호스트 이름)을 데이터 노드의 적절한 IP 주소로 확인해야 합니다.

- `enableDnsSupport` 옵션을 `true`(기본값)로 설정하여 VPC가 Amazon 제공 DNS 서버를 사용하는 경우, OpenSearch Service 엔드포인트 확인이 성공합니다.
- VPC가 프라이빗 DNS 서버를 사용하고 서버가 신뢰할 수 있는 퍼블릭 DNS 서버에 도달하여 DNS 호스트 이름을 확인할 수 있는 경우에도 OpenSearch Service 엔드포인트 확인이 성공합니다.

IP 주소는 변경될 수 있으므로, 항상 올바른 데이터 노드에 액세스할 수 있도록 주기적으로 도메인 엔드포인트를 확인해야 합니다. DNS 확인 주기를 1분으로 설정할 것을 권장합니다. 클라이언트를 사용하는 경우, 클라이언트 내 DNS 캐시도 삭제되는지 확인해야 합니다.

퍼블릭 액세스에서 VPC 액세스로 마이그레이션

도메인을 만들 때 도메인이 퍼블릭 엔드포인트를 사용할지 또는 VPC에 상주할지를 지정합니다. 도메인을 만든 후에는 이 옵션을 변경할 수 없습니다. 대신에 새 도메인을 만들고 수동으로 다시 인덱싱하거나 데이터를 마이그레이션할 수 있습니다. 스냅샷이 데이터를 마이그레이션하는 편리한 방법을 제공합니다. 스냅샷 생성 및 복원에 대한 자세한 내용은 the section called “인덱스 스냅샷 생성” 섹션을 참조하세요.

VPC 도메인 액세스 정책에 대하여

OpenSearch Service 도메인을 VPC에 배치할 경우 강력한 내부 보안 계층이 제공됩니다. 퍼블릭 액세스를 통해 도메인을 생성할 때는 엔드포인트가 다음과 같은 형식을 따릅니다.

```
https://search-domain-name-identifier.region.es.amazonaws.com
```

"퍼블릭"이라는 단어에서 알 수 있듯이 이 엔드포인트는 사용자가 액세스 권한을 제어할 수 있더라도(제어해야 합니다) 인터넷에 연결된 디바이스라면 모두 액세스할 수 있습니다. 웹 브라우저에서 엔드

포인트에 액세스하는 경우에는 `Not Authorized` 메시지가 수신될 수도 있지만 요청이 도메인에 전달됩니다.

VPC 액세스를 통해 도메인을 생성할 때도 엔드포인트가 아래와 같이 퍼블릭 엔드포인트와 비슷한 모습을 보입니다.

```
https://vpc-domain-name-identifier.region.es.amazonaws.com
```

하지만 웹 브라우저에서 엔드포인트에 액세스하려고 하면 요청 시간 제한에 걸릴 수도 있습니다. 따라서 기본적인 GET 요청을 실행할 때도 컴퓨터를 VPC에 연결할 수 있어야 합니다. 이 연결은 종종 VPN, Transit Gateway, 관리형 네트워크 또는 프록시 서버의 형식을 따릅니다. 사용할 수 있는 다양한 형식에 대한 자세한 내용은 Amazon VPC 사용 설명서의 VPC에 대한 예를 참조하세요. 개발 중심 예제는 the section called “VPC 도메인 테스트” 섹션을 참조하세요.

위와 같은 연결 요건에 더하여 VPC에서는 보안 그룹을 통해 도메인 액세스를 관리할 수 있습니다. 많은 사용 사례에서 이러한 보안 기능의 조합이면 충분하며 도메인에 개방적 액세스 정책을 적용하는 데 염려가 없을 것입니다.

오픈 액세스 정책이 포함된 작업은 인터넷에 접속하는 모든 사용자가 OpenSearch 서비스 도메인에 액세스할 수 있도록 하지는 않습니다. 오히려 요청이 OpenSearch Service 도메인에 전달되어 연결된 보안 그룹이 요청을 허용하는 경우에 한해 도메인이 해당 요청을 수락한다는 것을 의미합니다. 유일한 예외는 세분화된 액세스 제어 또는 IAM 역할을 지정하는 액세스 정책을 사용하는 경우입니다. 이러한 상황에서 도메인이 해당 요청을 수락하려면 보안 그룹이 요청을 허용하고, 또한 유효한 자격 증명으로 서명되어야 합니다.

> **Note**
>
> 보안 그룹은 이미 IP기반 액세스 정책을 사용하므로 VPC에 상주하는 OpenSearch Service 도메인에 IP 기반 액세스 정책을 적용할 수는 없습니다. 퍼블릭 액세스를 사용하는 경우에도 IP 기반 정책을 사용할 수 있습니다.

시작하기 전에: VPC 액세스 사전 요구 사항

VPC와 새 OpenSearch Service 도메인 간 연결을 활성화하기 전에 다음 작업을 수행해야 합니다.

- VPC 생성

  VPC를 생성하려면 Amazon VPC 콘솔, AWS CLI 또는 AWS SDKs. 자세한 내용은 Amazon VPC 사용 설명서의 VPC 작업을 참조하세요. VPC가 이미 있는 경우에는 이 단계를 건너뛸 수 있습니다.

- IP 주소 예약

  OpenSearch Service는 VPC의 서브넷에 네트워크 인터페이스를 배치하여 도메인과 VPC 간 연결을 활성화합니다. 각 네트워크 인터페이스에는 IP 주소가 연결됩니다. 서브넷에서 네트워크 인터페이스용 IP 주소를 충분히 예약해야 합니다. 자세한 내용은 VPC 서브넷에서 IP 주소 예약 섹션을 참조하세요.

VPC 도메인 테스트

VPC의 향상된 보안 기능을 통해 도메인 연결 및 기본 테스트 실행 작업을 실현할 수 있습니다. 이미 OpenSearch Service VPC 도메인이 있고 VPN 서버를 생성하지 않으려면 다음 프로세스를 시도합니다.

1. 도메인의 액세스 정책에 대해 [세분화된 액세스 제어만 사용(Only use fine-grained access control)]을 선택합니다. 테스트를 마친 후에는 언제든 이 설정을 업데이트할 수 있습니다.
2. Amazon Linux Amazon EC2 인스턴스를 OpenSearch Service 도메인과 동일한 VPC, 서브넷 및 보안 그룹에 생성합니다.

   이 인스턴스는 테스트용이며 거의 작업할 필요가 없으므로 `t2.micro`와 같은 저렴한 비용의 인스턴스 유형을 선택합니다. 인스턴스에 퍼블릭 IP 주소를 할당하고 새 키 페어를 생성하거나 기존 키 페어를 선택합니다. 새 키를 생성하는 경우 `~/.ssh` 디렉터리로 다운로드합니다.

   인스턴스 생성에 대한 자세한 내용은 Amazon EC2 Linux 인스턴스 시작하기를 참조하세요.
3. VPC에 인터넷 게이트웨이를 추가합니다.
4. VPC의 라우팅 테이블에서 새 라우팅을 추가합니다. 대상 주소(Destination)에서 컴퓨터의 퍼블릭 IP 주소를 포함하는 CIDR 블록을 지정합니다. 대상(Target)에서 방금 생성한 인터넷 게이트웨이를 지정합니다.

   예를 들어 컴퓨터에 `123.123.123.123/32`를 지정하거나 컴퓨터 범위에 `123.123.123.0/24`를 지정할 수 있습니다.
5. 보안 그룹의 경우 두 가지 인바운드 규칙을 지정합니다.

유형	프로토콜	포트 범위	소스
SSH(22)	TCP(6)	22	*your-cidr-block*

유형	프로토콜	포트 범위	소스
HTTPS(443)	TCP(6)	443	*your-security-group-id*

첫 번째 규칙은 SSH를 EC2 인스턴스로 가져옵니다. 두 번째는 EC2 인스턴스가 HTTPS를 통해 OpenSearch Service 도메인과 통신할 수 있게 허용합니다.

6. 터미널에서 다음 명령을 실행합니다.

```
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
```

이 명령은 https://localhost:9200에 대한 요청을 EC2 인스턴스를 통해 OpenSearch Service 도메인으로 전달하는 SSH 터널을 생성합니다. 명령에 포트 9200을 지정하면 로컬 OpenSearch 설치를 시뮬레이션하지만 원하는 포트를 사용합니다. OpenSearch Service는 포트 80(HTTP) 또는 443(HTTPS)을 통한 연결만 허용합니다.

이 명령은 피드백을 제공하지 않고 무제한 실행됩니다. 이를 중단하려면 `Ctrl + C`를 누릅니다.

7. 웹 브라우저에서 https://localhost:9200/_dashboards/로 이동합니다. 보안 예외를 인정해야 할 수 있습니다.

또는 curl, Postman 또는 좋아하는 프로그래밍 언어를 사용하여 https://localhost:9200에 요청을 보낼 수 있습니다.

> **Tip**
>
> 인증서 불일치로 인해 curl 오류가 발생하는 경우 `--insecure` 플래그를 시도합니다.

VPC 서브넷에서 IP 주소 예약

OpenSearch Service는 VPC의 서브넷(다중 가용 영역을 활성화한 경우 VPC의 서브넷 여러 개)에 네트워크 인터페이스를 배치하여 도메인을 VPC에 연결합니다. 각 네트워크 인터페이스에는 IP 주소가 연결됩니다. OpenSearch Service 도메인을 만들기 전에 각 서브넷에서 네트워크 인터페이스를 수용하는 데 충분한 IP 주소가 있어야 합니다.

기본 공식은 다음과 같습니다. 각 서브넷에서 OpenSearch Service가 예약한 IP 주소 수 = 데이터 노드 수 곱하기 3, 나누기 가용 영역 수.

예시

- 도메인에 3개의 가용 영역에 걸쳐 9개의 데이터 노드가 있는 경우, 서브넷당 IP 개수는 9 * 3 / 3 = 9입니다.
- 도메인에 2개의 가용 영역에 걸쳐 8개의 데이터 노드가 있는 경우, 서브넷당 IP 개수는 8 * 3 / 2 = 12입니다.
- 도메인의 가용 영역 하나에 6개의 데이터 노드가 있는 경우, 서브넷당 IP 개수는 6 * 3 / 1 = 18입니다.

도메인을 생성할 때 OpenSearch Service는 IP 주소를 예약한 후 도메인에 일부를 사용하고 나머지는 블루/그린 배포를 위해 예약합니다. Amazon EC2 콘솔의 네트워크 인터페이스 단원에서 네트워크 인터페이스 및 연결된 IP 주소를 확인할 수 있습니다. 설명(Description) 열은 네트워크 인터페이스와 연결된 OpenSearch Service 도메인을 보여줍니다.

> **Tip**
>
> OpenSearch Service 예약 IP 주소에 전용 서브넷을 생성하는 것이 좋습니다. 전용 서브넷을 사용하면 다른 애플리케이션 및 서비스와 중복을 방지하고 향후 클러스터를 확장해야 할 경우 추가 IP 주소 예약이 가능하도록 할 수 있습니다. 자세히 알아보려면 VPC에서 서브넷 만들기 섹션을 참조하세요.

전용 조정자 노드를 프로비저닝하여 VPC 도메인에 필요한 프라이빗 IP 주소 예약 수를 줄이는 방법도 고려해 볼 수 있습니다. OpenSearch는 데이터 노드 대신 전용 조정자 노드에 탄력적 네트워크 인터페이스(ENI)를 연결합니다. 전용 조정자 노드는 일반적으로 총 데이터 노드의 약 10%를 차지합니다. 따라서 더 적은 수의 프라이빗 IP 주소가 VPC 도메인에 대해 예약됩니다.

VPC 액세스를 위한 서비스 연결 역할

서비스 연결 역할은 서비스가 사용자를 대신하여 리소스를 생성하고 관리할 수 있도록 서비스에 권한을 위임하는 고유한 유형의 IAM 역할입니다. OpenSearch Service에서 VPC에 액세스하고, 도메인 엔드포인트를 생성하고, VPC의 서브넷에 네트워크 인터페이스를 배치하려면 서비스 연결 역할이 필요합니다.

OpenSearch Service 콘솔을 사용하여 VPC에서 도메인을 생성할 때 OpenSearch Service에서 자동으로 역할이 생성됩니다. 이 자동 생성이 성공하려면 사용자가 `iam:CreateServiceLinkedRole` 작업에 대한 권한을 보유해야 합니다. 자세히 알아보려면 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

OpenSearch Service가 역할을 생성하면 IAM 콘솔을 사용하여 이 역할(`AWSServiceRoleForAmazonOpenSearchService`)을 볼 수 있습니다.

이 역할 권한에 대한 모든 정보와 삭제하는 방법은 the section called “서비스 연결 역할 사용” 섹션을 참조하세요.

Amazon OpenSearch Service에서 인덱스 스냅샷 생성

Amazon OpenSearch Service의 스냅샷은 클러스터의 인덱스와 상태의 백업입니다. 상태에는 클러스터 설정, 노드 정보, 인덱스 설정 및 샤드 할당이 포함됩니다.

OpenSearch Service 스냅샷은 다음 형식으로 제공됩니다.

- 자동 스냅샷은 클러스터 복구 전용입니다. 빨간색 클러스터 상태 또는 데이터 손실이 발생할 경우 이 옵션을 사용하여 도메인을 복원할 수 있습니다. 자세한 내용은 아래 스냅샷 복원을 참조하세요. OpenSearch Service는 추가 요금 없이 미리 구성된 Amazon S3 버킷에 자동 스냅샷을 저장합니다.
- 수동 스냅샷은 클러스터 복구 또는 한 클러스터에서 다른 클러스터로 데이터 이동 시 사용합니다. 수동 스냅샷을 시작해야 합니다. 이러한 스냅샷은 자체 Amazon S3 버킷에 저장되며 표준 S3 요금이 적용됩니다. 자체 관리형 OpenSearch 클러스터의 스냅샷이 있는 경우 해당 스냅샷을 사용하여 OpenSearch Service 도메인으로 마이그레이션할 수도 있습니다. 자세한 내용은 Amazon OpenSearch Service로 마이그레이션을 참조하세요.

모든 OpenSearch Service 도메인은 자동 스냅샷을 생성하지만 빈도는 다음과 같은 방법으로 다릅니다.

- OpenSearch 또는 Elasticsearch 5.3 이상을 실행하는 도메인의 경우 OpenSearch Service는 시간별 자동 스냅샷을 생성하고 최대 336개의 스냅샷을 14일 동안 보관합니다. 시간당 스냅샷은 증분 특성으로 인해 중단이 적습니다. 또한 도메인 문제가 발생할 경우 보다 최근의 복구 시점을 제공합니다.
- Elasticsearch 5.1 이하를 실행하는 도메인의 경우 OpenSearch Service는 지정한 시간 동안 일별 자동 스냅샷을 생성하고 최대 14개의 스냅샷을 보관하며 30일 이상 스냅샷 데이터를 보관하지 않습니다.

클러스터가 빨간색 상태가 되면 클러스터 상태가 지속되는 동안 모든 자동 스냅샷이 실패합니다. 2주 내에 문제를 해결하지 않으면 클러스터의 데이터가 영구적으로 손실될 수 있습니다. 문제 해결 단계는 the section called “빨간색 클러스터 상태” 섹션을 참조하세요.

사전 조건

스냅샷을 수동으로 생성하려면 IAM 및 Amazon S3를 사용해야 합니다. 스냅샷을 생성하기 전에 다음 필수 조건을 충족해야 합니다.

전제 조건	설명
S3 버킷	S3 버킷을 생성하여 OpenSearch Service 도메인에 대한 수동 스냅샷을 저장합니다. 지침은 Amazon Simple Storage Service 사용 설명서의 범용 버킷 생성을 참조하세요. 버킷의 이름을 기억해야 다음 위치에서 사용할 수 있습니다. • IAM 역할에 연결된 IAM 정책의 `Resource` 설명문 • 스냅샷 리포지토리를 등록하는 데 사용되는 Python 클라이언트 (이 메서드를 사용하는 경우) **Important** S3 Glacier 수명 주기 규칙을 이 버킷에 적용하지 마세요. 수동 스냅샷은 S3 Glacier 스토리지 클래스를 지원하지 않습니다.
IAM 역할	OpenSearch Service에 대한 권한을 위임할 IAM 역할을 생성합니다. 지침은 IAM 사용 설명서에서 IAM 역할 생성(콘솔)을 참조하세요. 이 장의 나머지 부분에서는 이 역할을 `TheSnapshotRole` 이라고 부릅니다. IAM 정책 연결 다음 정책을 `TheSnapshotRole` 에 연결하여 S3 버킷에 대한 액세스를 허용하려면:

전제 조건	설명

	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 /*"
	]
	}
	]
	}
	
	역할에 정책을 연결하는 지침은 IAM 사용 설명서의 IAM 자격 증명 권한 추가(콘솔)를 참조하세요.
	
	신뢰 관계 편집
	
	다음 예제에서와 같이 `Principal` 설명문에서 OpenSearch Service를 지정하려면 `TheSnapshotRole` 의 신뢰 관계를 편집합니다.
	
	JSON
	
	{

전제 조건	설명

	"Version": "2012-10-17",
	"Statement": [{
	"Sid": "",
	"Effect": "Allow",
	"Principal": {
	"Service": "es.amazonaws.com"
	},
	"Action": "sts:AssumeRole"
	}]
	
	}
	
	신뢰 관계를 편집하는 지침은 IAM 사용 설명서의 역할 신뢰 정책 업데이트를 참조하세요.

전제 조건	설명

권한	스냅샷 리포지토리를 등록하려면 TheSnapshotRole 을 OpenSearch Service에 전달할 수 있어야 합니다. es:ESHttpPut 작업에도 액세스해야 합니다. 이러한 두 권한을 모두 부여하려면 요청에 서명하기 위해 자격 증명이 사용되는 IAM 역할에 다음 정책을 연결합니다.
	
	JSON
	
	{
	"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* /*"
	}
	]
	}
	
	사용자 또는 역할에 TheSnapshotRole 을 전달할 iam:PassRole 권한이 없는 경우 다음 단계에서 리포지토리를 등록하려고 할 때 다음과 같은 일반적인 오류가 발생할 수 있습니다.
	
	$ 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* "}

수동 스냅샷 리포지토리 등록

수동 인덱스 스냅샷을 생성하려면 OpenSearch Service를 사용해 스냅샷 리포지토리를 등록해야 합니다. 이 일회성 작업을 수행하려면에 설명된 TheSnapshotRole대로에 액세스할 수 있는 자격 증명으로 AWS 요청에 서명해야 합니다the section called “사전 조건”.

1단계: OpenSearch 대시보드에서 스냅샷 역할 매핑(세분화된 액세스 제어를 사용하는 경우)

세분화된 액세스 제어를 사용하면 리포지토리를 등록할 때 추가 단계가 있습니다. 다른 모든 목적으로 HTTP 기본 인증을 사용하더라도 TheSnapshotRole을 전달할 `iam:PassRole` 권한이 있는 IAM 역할에 `manage_snapshots` 역할을 매핑해야 합니다.

1. OpenSearch Service 도메인에 대한 OpenSearch 대시보드 플러그인으로 이동합니다. OpenSearch Service 콘솔의 도메인 대시보드에서 Dashboards 엔드포인트를 찾을 수 있습니다.
2. 주 메뉴에서 보안(Security), 역할(Roles)을 선택하고 manage_snapshots 역할을 선택합니다.
3. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다.
4. TheSnapshotRole을 전달할 권한이 있는 역할의 ARN을 추가합니다. Backend roles(백엔드 역할) 아래에 역할 ARN을 배치합니다.

   ```
   arn:aws:iam::123456789123:role/role-name
   ```

5. 맵(Map)을 선택하고 매핑된 사용자(Mapped users)에 사용자 또는 역할이 나타나는지 확인합니다.

2단계: 리포지토리 등록

다음 스냅샷 탭은 스냅샷 디렉토리를 등록하는 방법을 보여줍니다. 수동 스냅샷을 암호화하고 새 도메인으로 마이그레이션한 후 스냅샷을 등록하는 것과 관련된 옵션은 관련 탭을 참조하세요.

Snapshots

스냅샷 리포지토리를 등록하려면 PUT 요청을 OpenSearch Service 도메인 엔드포인트로 보냅니다. 대신 샘플 Python 클라이언트, Postman이나 다른 방법으로 서명된 요청을 전송해 스냅샷 리포지토리를 등록합니다. OpenSearch 대시보드 콘솔에서 리포지토리를 등록하는 데 PUT 요청을 사용할 수 없습니다.

요청은 다음과 같은 형식을 사용합니다.

```
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**
>
> 리포지토리 이름은 “cs-”로 시작할 수 없습니다. 또한 여러 도메인에서 동일한 리포지토리에 쓰면 안 됩니다. 하나의 도메인에만 리포지토리에 대한 쓰기 액세스 권한이 있어야 합니다.

도메인이 Virtual Private Cloud(VPC)에 상주하는 경우, 요청이 스냅샷 리포지토리를 등록하려면 컴퓨터가 VPC에 연결되어야 합니다. VPC 액세스는 네트워크 구성에 따라 다르지만, VPN 또는 회사 네트워크 연결이 필요할 수 있습니다. OpenSearch Service 도메인에 도달할 수 있는지 알아보려면 웹 브라우저에서 `https://your-vpc-domain.region.es.amazonaws.com`으로 이동하여 기본 JSON 응답을 받을 수 있는지 확인합니다.

Amazon S3 버킷이 OpenSearch 도메인 AWS 리전 이 아닌 다른에 있는 경우 `"endpoint": "s3.amazonaws.com"` 요청에 파라미터를 추가합니다.

Encrypted snapshots

현재 AWS Key Management Service (KMS) 키를 사용하여 수동 스냅샷을 암호화할 수 없지만 서버 측 암호화(SSE)를 사용하여 보호할 수 있습니다.

스냅샷 리포지토리로 사용하는 버킷에 대해 S3 관리형 키로 SSE를 활성화하려면 PUT 요청의 `"settings"` 블록에 `"server_side_encryption": true`를 추가합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3 관리형 키(SSE-S3)로 서버 측 암호화 사용을 참조하세요.

또는 스냅샷 리포지토리로 사용하는 S3 버킷에서 서버 측 암호화에 AWS KMS 키를 사용할 수 있습니다. 이 접근 방식을 사용하는 경우 S3 버킷을 암호화하는 데 사용되는 AWS KMS 키에 대한

TheSnapshotRole 권한을 제공해야 합니다. 자세한 내용은 AWS KMS에서 키 정책을 참조하세요.

Domain migration

스냅샷 리포지토리 등록은 일회성 작업입니다. 하지만 한 도메인에서 다른 도메인으로 마이그레이션하려면 기존 도메인과 새 도메인에서 동일한 스냅샷 리포지토리를 등록해야 합니다. 리포지토리 이름은 임의의 이름입니다.

새 도메인으로 마이그레이션하거나 동일한 리포지토리를 여러 도메인으로 등록할 때는 다음 지침을 고려합니다.

- 새 도메인에 리포지토리를 등록하는 경우 PUT 요청의 "settings" 블록에 "readonly": true를 추가합니다. 이 설정을 사용하면 실수로 이전 도메인의 데이터를 덮어쓰지 않을 수 있습니다. 하나의 도메인에만 리포지토리에 대한 쓰기 액세스 권한이 있어야 합니다.
- 데이터를 다른 AWS 리전의 도메인으로 마이그레이션하는 경우(예: us-east-2에 있는 이전 도메인 및 버킷에서 us-west-2의 새 도메인으로 마이그레이션하는 경우) PUT 문에서 "region": "*region*"(을)를 "endpoint": "s3.amazonaws.com"(으)로 대체하고 해당 요청을 다시 시도합니다.

샘플 Python 클라이언트 사용하기

Python 클라이언트는 간단한 HTTP 요청보다 자동화가 쉽고 재사용성이 뛰어납니다. 이 메서드를 사용하여 스냅샷 리포지토리를 등록하려면 다음 샘플 Python 코드를 register-repo.py와 같은 Python 파일로 저장합니다. 클라이언트는 AWS SDK for Python (Boto3), requests 및 requests-aws4auth 패키지를 요구합니다. 클라이언트는 다른 스냅샷 작업을 위한 주석 처리된 예제를 포함하고 있습니다.

샘플 코드에서 변수 host, region, path, payload를 업데이트합니다.

```
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)
```

수동 스냅샷 생성

스냅샷은 즉각적으로 이루어지지 않습니다. 완료하는 데 시간이 걸리며 클러스터의 완벽한 특정 시점 보기를 나타내지 않습니다. 스냅샷이 진행 중인 동안에도 문서를 인덱스 처리하고 클러스터에 다른 요청을 할 수 있지만 일반적으로 새로운 문서 및 기존 문서의 업데이트는 해당 스냅샷에 포함되지 않습니다. 스냅샷은 OpenSearch에서 해당 스냅샷을 시작한 시점에 존재한 기본 샤드를 포함합니다. 스냅샷 스레드 풀의 크기에 따라 서로 다른 시간에 스냅샷에 다른 샤드가 포함될 수 있습니다. 모범 사례는 the section called “스냅샷 성능 개선” 단원을 참조하세요.

스냅샷 스토리지 및 성능

OpenSearch 스냅샷은 증분식이며, 마지막으로 성공한 스냅샷 이후로 변경된 데이터만 저장합니다. 이 증분적 특성은 자주 사용되는 스냅샷과 그 반대의 스냅샷 간의 디스크 사용량 차이가 거의 없는 경우가 많다는 의미이기도 합니다. 즉, 일주일에 한 번 시간별로 스냅샷을 가져올 경우(총 168개의 스냅샷) 주말에 단일 스냅샷을 가져오는 것보다 훨씬 많은 디스크 공간을 사용할 수는 없습니다. 또한 스냅샷을 자주 가져올수록 완료하는 데 걸리는 시간이 줄어듭니다. 예를 들어 일일 스냅샷은 완료하는 데 20~30

분이 소요될 수 있지만 시간당 스냅샷은 몇 분 안에 완료될 수 있습니다. 일부 OpenSearch 사용자는 30분마다 스냅샷을 가져옵니다.

스냅샷 만들기

스냅샷을 생성할 때 다음 정보를 지정합니다.

- 스냅샷 리포지토리의 이름
- 스냅샷의 이름

이 장의 예제에서는 편의상 그리고 간단하게 하기 위해 일반적인 HTTP 클라이언트인 curl을 사용합니다. curl 요청에 사용자 이름과 암호를 전달하려면 튜토리얼 시작하기를 참조하세요.

하지만 액세스 정책이 사용자 또는 역할을 지정하는 경우에는 스냅샷 요청에 서명해야 합니다. curl의 경우 버전 7.75.0 이상에서 `--aws-sigv4` 옵션을 사용할 수 있습니다. 샘플 Python 클라이언트의 주석 처리된 예제를 사용하여 curl 명령이 사용하는 동일한 엔드포인트에 서명된 HTTP 요청을 할 수 있습니다.

수동 스냅샷을 생성하려면 다음 단계를 수행합니다.

1. 현재 스냅샷 생성이 진행 중인 경우 스냅샷을 생성할 수 없습니다. 확인하려면 다음 명령을 실행합니다.

   ```
   curl -XGET 'domain-endpoint/_snapshot/_status'
   ```

2. 수동 스냅샷을 생성하려면 다음 명령을 실행합니다.

   ```
   curl -XPUT 'domain-endpoint/_snapshot/repository-name/snapshot-name'
   ```

   특정 인덱스를 포함하거나 제외하고 다른 설정을 지정하려면 요청 본문을 추가합니다. 요청 구조는 OpenSearch 설명서의 Take snapshots(스냅샷 만들기) 섹션을 참조하세요.

> **Note**
>
> 스냅샷 생성에 필요한 시간은 OpenSearch Service 도메인의 크기에 따라 늘어납니다. 스냅샷 작업이 길게 실행되면 경우에 따라 `504 GATEWAY_TIMEOUT` 같은 오류가 발생합니다. 이러한 오류는 무시하고 작업이 성공적으로 완료될 때까지 기다릴 수 있습니다. 다음 명령을 실행하여 도메인의 모든 스냅샷 상태를 확인합니다.

```
curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
```

스냅샷 복원

스냅샷을 복원하기 전에 대상 도메인에서 Multi-AZ with Standby를 사용하지 않는지 확인하십시오. 대기가 활성화되어 있으면 복원 작업이 실패합니다.

> **Warning**
>
> 인덱스 별칭을 사용하는 경우, 별칭에 요청 쓰기를 중단하거나 인덱스를 삭제하기 전에 그 별칭을 다른 인덱스로 전환합니다. 쓰기 중단 요청은 다음과 같은 상황을 피하도록 해 줍니다.
>
> 1. 인덱스를 삭제하면 별칭도 삭제됩니다.
> 2. 현재 지워진 별칭에 잘못된 쓰기 요청 때문에 그 별칭과 동일한 이름을 가진 새 인덱스가 생성됩니다.
> 3. 새 인덱스에 지정하는 이름과 충돌하기 때문에 그 별칭을 더 이상 사용할 수 없습니다. 별칭을 다른 인덱스로 전환하는 경우 스냅샷에서 복원할 때 `"include_aliases": false`를 지정합니다.

스냅샷을 복원하려면

1. 복원할 스냅샷을 식별합니다. 사용자 지정 분석기 패키지 또는 할당 요구 사항 설정과 같은 이 인덱스의 모든 설정이 도메인과 호환되는지 확인하세요. 모든 스냅샷 리포지토리를 보려면 다음 명령을 실행합니다.

   ```
   curl -XGET 'domain-endpoint/_snapshot?pretty'
   ```

   리포지토리를 식별한 후, 다음 명령을 실행하여 모든 스냅샷을 봅니다.

   ```
   curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
   ```

> **Note**
>
> 대부분의 자동 스냅샷은 `cs-automated` 리포지토리에 저장됩니다. 도메인이 저장된 데이터를 암호화하는 경우 `cs-automated-enc` 리포지토리에 저장됩니다. 찾고 있는 수동 스냅샷 리포지토리가 보이지 않으면 도메인에 이 수동 스냅샷 리포지토리를 등록했는지 확인합니다.

2. (선택 사항) 클러스터의 인덱스와 스냅샷의 인덱스 간에 이름 충돌이 있는 경우 OpenSearch Service 도메인에서 하나 이상의 인덱스를 삭제하거나 이름을 변경합니다. 이미 같은 이름의 인덱스가 있는 OpenSearch 클러스터로 인덱스 스냅샷을 복원할 수는 없습니다.

   인덱스 이름 충돌이 있는 경우 다음 옵션이 있습니다.

   - 기존 OpenSearch Service 도메인에서 인덱스를 삭제한 후 스냅샷을 복원합니다.
   - 스냅샷에서 인덱스를 복원할 때 인덱스 이름을 변경하고 나중에 다시 인덱스를 만듭니다. 인덱스의 이름을 바꾸는 방법을 알아보려면 OpenSearch 설명서의 this example request를 참조하세요.
   - 스냅샷을 다른 OpenSearch Service 도메인에 복원합니다(수동 스냅샷만 가능).

   다음 명령은 도메인의 모든 기존 인덱스를 삭제합니다.

   ```
   curl -XDELETE 'domain-endpoint/_all'
   ```

   그러나 모든 인덱스를 복원하지 않으려는 경우 하나를 삭제할 수 있습니다.

   ```
   curl -XDELETE 'domain-endpoint/index-name'
   ```

3. 스냅샷을 복원하려면 다음 명령을 실행합니다.

   ```
   curl -XPOST 'domain-endpoint/_snapshot/repository-name/snapshot-name/_restore'
   ```

   OpenSearch 대시보드 및 세분화된 액세스 제어 인덱스에 대한 특별한 권한 때문에 모든 인덱스를 복원하려는 시도가 실패할 수 있으며, 자동화된 스냅샷에서 복원을 시도할 경우 특히 그렇습니다. 다음 예제에서는 `cs-automated` 스냅샷 리포지토리에 있는 `2020-snapshot`에서 인덱스 `my-index`만 복원합니다.

   ```
   curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
   ```

```
-d '{"indices": "my-index"}' \
-H 'Content-Type: application/json'
```

또는 Dashboards 및 세분화된 액세스 제어 인덱스를 제외한 모든 인덱스를 복원할 수 있습니다.

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "-.kibana*,-.opendistro*"}' \
-H 'Content-Type: application/json'
```

`rename_pattern` 및 `rename_replacement` 파라미터를 사용하여 데이터를 삭제하지 않고 스냅샷을 복원할 수 있습니다. 이러한 파라미터에 대한 자세한 내용은 OpenSearch 설명서의 Restore Snapshot API 요청 필드 및 예제 요청을 참조하세요.

> **Note**
>
> 기본 샤드 중 일부만 관련 인덱스에 사용할 수 있는 경우, 스냅샷에 `state`의 `PARTIAL`이(가) 있을 수 있습니다. 이 값은 최소한 샤드 하나의 데이터가 제대로 저장되지 않았음을 의미합니다. 부분 스냅샷에서도 복원할 수는 있지만, 그보다 오래된 스냅샷을 사용하여 누락된 인덱스를 복원해야 합니다.

수동 스냅샷 삭제

수동으로 스냅샷을 삭제하려면 다음 명령을 실행합니다.

```
DELETE _snapshot/repository-name/snapshot-name
```

Snapshot Management를 사용한 스냅샷 자동화

OpenSearch Dashboards에서 Snapshot Management(SM) 정책을 설정하여 주기적인 스냅샷 생성 및 삭제를 자동화할 수 있습니다. SM은 인덱스 그룹의 스냅샷을 생성할 수 있는 반면 인덱스 상태 관리는 인덱스당 하나의 스냅샷만 만들 수 있습니다. OpenSearch Service에서 SM을 사용하려면 자체 Amazon S3 리포지토리를 등록해야 합니다. 리포지토리 등록에 대한 지침은 수동 스냅샷 리포지토리 등록을 참조하세요.

SM 이전에 OpenSearch Service는 기본적으로 켜져 있는 자동 스냅샷 기능을 무료로 제공했습니다. 이 기능은 스냅샷을 서비스가 관리하는 `cs-*` 리포지토리로 전송합니다. 기능을 비활성화하려면 지원에 문의하세요.

SM 기능에 대한 자세한 내용은 OpenSearch 설명서의 스냅샷 관리를 참조하세요.

SM은 현재 여러 인덱스 유형에 대한 스냅샷 생성을 지원하지 않습니다. 예를 들어 *로 일부 인덱스에서 스냅샷을 생성하려고 하거나 일부 인덱스가 웜 티어에 속해 있는 경우 스냅샷 생성이 실패합니다. 스냅샷에 여러 인덱스 유형을 포함해야 하는 경우 SM에서 이 옵션을 지원할 때까지 ISM 스냅샷 작업을 사용하세요.

권한 구성

이전 OpenSearch Service 도메인 버전에서 2.5로 업그레이드한 경우 스냅샷 관리 보안 권한이 도메인에 정의되어 있지 않을 수 있습니다. 관리자가 아닌 사용자는 이 역할에 매핑되어 세분화된 액세스 제어를 사용하는 도메인의 스냅샷 관리를 사용해야 합니다. 수동으로 스냅샷 관리 역할을 생성하려면 다음 단계를 수행합니다.

1. OpenSearch 대시보드에서 보안(Security)으로 이동하여 권한(Permissions)을 선택합니다.
2. 작업 그룹 생성(Create action group)을 선택하고 다음 그룹을 구성합니다.

그룹 이름	권한
`snapshot_management_full_access`	• `cluster:admin/opensearch/snapshot_management/*` • `cluster:admin/opensearch/notifications/feature/publish` • `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. 역할(Roles)과 역할 생성(Create role)을 차례로 선택합니다.
4. 역할 이름을 snapshot_management_role로 지정합니다.

5. 클러스터 권한에서 `snapshot_management_full_access` 및 `snapshot_management_read_access`를 선택합니다.
6. 생성(Create)을 선택합니다.
7. 역할을 생성한 후, 스냅샷을 관리할 사용자 또는 백엔드 역할에 매핑합니다.

고려 사항

스냅샷 관리를 구성할 때 다음 사항을 고려하세요.

- 리포지토리당 하나의 정책이 허용됩니다.
- 정책 하나에 최대 400개의 스냅샷이 허용됩니다.
- 도메인이 빨간색 상태이거나, JVM 압력이 높거나(85% 이상), 스냅샷 기능이 중단된 경우에는 이 기능이 실행되지 않습니다. 클러스터의 전체 인덱싱 및 검색 성능이 영향을 받는 경우 SM도 영향을 받을 수 있습니다.
- 스냅샷 작업은 이전 작업이 완료된 후에만 시작되므로 한 정책으로 동시 스냅샷 작업이 활성화되지 않습니다.
- 일정이 동일한 정책이 여러 개 있을 경우 리소스 스파이크가 발생할 수 있습니다. 정책의 스냅샷 인덱스가 겹치는 경우 샤드 수준 스냅샷 작업은 순차적으로만 실행될 수 있으며, 이로 인해 연쇄적인 성능 문제가 발생할 수 있습니다. 정책이 리포지토리를 공유하는 경우 해당 리포지토리에 대한 쓰기 작업이 급증할 수 있습니다.
- 특별한 사용 사례가 없는 한 스냅샷 작업 자동화를 시간당 1회 이하로 예약하는 것이 좋습니다.

인덱스 상태 관리를 사용한 스냅샷 자동화

인덱스 상태 관리(ISM) 스냅샷 작업을 사용하여 수명, 크기 또는 문서 수의 변화에 따라 인덱스의 스냅샷을 자동으로 트리거할 수 있습니다. ISM은 인덱스당 하나의 스냅샷이 필요한 경우에 가장 적합합니다. 인덱스 그룹의 스냅샷이 필요한 경우 Snapshot Management를 사용한 스냅샷 자동화(을)를 참조하세요.

OpenSearch Service에서 SM을 사용하려면 자체 Amazon S3 리포지토리를 등록해야 합니다. `snapshot` 작업을 사용한 ISM 정책의 예는 샘플 정책을 참조하세요.

스냅샷에 Curator 사용

ISM이 인덱스 및 스냅샷 관리를 위해 작동하지 않는 경우 Curator를 대신 사용할 수 있습니다. 이는 복잡한 클러스터에서 관리 작업을 간소화하는 데 도움이 될 수 있는 고급 필터링 기능을 제공합니다. pip를 사용하여 Curator를 설치합니다.

```
pip install elasticsearch-curator
```

명령줄 인터페이스(CLI) 또는 Python API로서 Curator를 사용할 수 있습니다. Python API를 사용하는 경우 버전 7.13.4 또는 그 이전의 레거시 elasticsearch-py 클라이언트를 사용해야 합니다. 이는 opensearch-py 클라이언트를 지원하지 않습니다.

CLI를 사용하는 경우 명령줄에서 자격 증명을 내보내고 다음과 같이 `curator.yml`을 구성합니다.

```
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
```

Amazon OpenSearch Service 도메인 업그레이드

> **Note**
>
> OpenSearch 및 Elasticsearch 버전 업그레이드는 서비스 소프트웨어 업데이트와 다릅니다. OpenSearch Service 도메인에 대한 서비스 소프트웨어 업데이트에 대한 자세한 내용은 the section called “서비스 소프트웨어 업데이트” 섹션을 참조하세요.

Amazon OpenSearch Service는 OpenSearch 1.0 이상 또는 Elasticsearch 5.1 이상을 실행하는 도메인에 대해 현재 위치 업그레이드를 제공합니다. Amazon Data Firehose 또는 Amazon CloudWatch Logs 같은 서비스를 사용하여 데이터를 OpenSearch로 스트리밍하는 경우, 마이그레이션하기 전에 이러한 서비스가 새 버전의 OpenSearch를 지원하는지 확인합니다.

지원되는 업그레이드 경로

현재 OpenSearch Service는 다음 업그레이드 경로를 지원합니다.

구 버전	새 버전
OpenSearch 1.3 또는 2.x	OpenSearch 2.x 도메인이 다음 조건을 충족하는 경우 OpenSearch 2.17은 기본적으로 자동 모드로 동시 세그먼트 검색을 활성화합니다. • 이전 동시 검색 설정은 명시적으로 설정되지 않습니다. • 모든 데이터 인스턴스(핫 및 웜)는 인스턴스 유형이 2.xl 이상입니다. • 1주일 이상 데이터 인스턴스(핫 및 웜)의 평균 p90 cpu 사용률은 45% 미만입니다. 동시 세그먼트 검색 설정에 대한 자세한 내용은 동시 세그먼트 검색을 참조하세요. 버전 2.3에는 다음과 같은 주요 변경 사항이 있습니다. • 이 `type` 파라미터는 버전 2.0의 모든 OpenSearch API 엔드포인트에서 제거되었습니다. 자세한 내용은 주요 변경 사항을 참조하세요. • 도메인에 원래 Elasticsearch 6.8에서 생성된 인덱스(핫, UltraWarm 또는 콜드)가 포함된 경우 해당 인덱스는 OpenSearch 2.3과 호환되지 않습니다. 버전 2.3으로 업그레이드하기 전에 호환되지 않는 인덱스를 재인덱싱해야 합니다. 호환되지 않는 UltraWarm 또는 콜드 인덱스의 경우 핫 스토리지로 마이그레이션하고 데이터를 재인덱싱한 다음 웜 또는 콜드 스토리지로 다시 마이그레이션합니다. 또는 인덱스가 더 이상 필요하지 않은 경우 인덱스를 삭제할 수 있습니다. 이러한 단계를 먼저 수행하지 않고 실수로 도메인을 버전 2.3으로 업그레이드한 경우 호환되지 않는 인덱스를 현재 스토리지 계층에서 마이그레이션할 수 없습니다. 유일한 방법은 삭제하는 것입니다.
OpenSearch 1.x	OpenSearch 1.x

구 버전	새 버전
Elasticsearch 7.x	Elasticsearch 7.x 또는 OpenSearch 1.x **Important** OpenSearch 1.x에서는 많은 중요한 변경 사항이 도입되었습니다. 자세한 내용은 Amazon OpenSearch Service 이름 변경을 참조하세요.
Elasticsearch 6.8	Elasticsearch 7.x 또는 OpenSearch 1.x **Important** Elasticsearch 7.0 및 OpenSearch 1.0에는 수많은 주요 변경 사항이 포함되어 있습니다. 인플레이스 업그레이드를 시작하기 전에 6.x 도메인의 수동 스냅샷을 생성하여 테스트 7.x 또는 OpenSearch 1.x 도메인에서 복원하고, 해당 테스트 도메인을 사용하여 잠재적인 업그레이드 문제를 식별하는 것이 좋습니다. OpenSearch 1.0의 주요 변경 사항은 Amazon OpenSearch Service 이름 변경 섹션을 참조하세요. Elasticsearch 6.x와 같이 인덱스에는 하나의 매핑 유형만 포함될 수 있지만 해당 유형의 이름은 `_doc`여야 합니다. 결과적으로 특정 API(예: `_bulk` API)는 더 이상 요청 본문에 매핑 유형이 필요하지 않습니다. 새 인덱스의 경우 자체 호스팅된 Elasticsearch 7.x 및 OpenSearch 1.x의 기본 샤드 수는 1입니다. Elasticsearch 7.x 이상의 OpenSearch Service 도메인은 이전 기본값인 5를 유지합니다.
Elasticsearch 6.x	Elasticsearch 6.x

구 버전	새 버전
Elasticsearch 5.6	Elasticsearch 6.x **Important** 버전 6.x에서 생성된 인덱스는 더 이상 여러 개의 매핑 유형을 지원하지 않습니다. 버전 5.x에서 생성된 인덱스는 6.x 클러스터로 복원될 때 계속 여러 매핑 유형을 지원합니다. 클라이언트 코드를 통해 인덱스당 매핑 유형 하나만 생성할 수 있는지 확인합니다. Elasticsearch 5.6에서 6.x로 업그레이드할 때 가동 중지를 최소화하기 위해 OpenSearch Service는 `.kibana` 인덱스를 `.kibana-6` 으로 다시 인덱싱하고, `.kibana`를 삭제하고, `.kibana`라는 별칭을 생성하여 새 인덱스를 새 별칭에 매핑합니다.
Elasticsearch 5.x	Elasticsearch 5.x

업그레이드 프로세스는 세 단계로 구성됩니다.

1. 업그레이드 전 점검 – OpenSearch Service가 업그레이드를 차단할 수 있는 문제가 있는지 확인하고 이러한 점검이 성공하지 않을 경우 다음 단계를 진행하지 않습니다.
2. 스냅샷 – OpenSearch Service가 OpenSearch 또는 Elasticsearch 클러스터의 스냅샷을 만들고 스냅샷이 성공하지 않을 경우 다음 단계를 진행하지 않습니다. 업그레이드가 실패할 경우 OpenSearch Service가 이 스냅샷을 사용해 클러스터를 원래 상태로 복원합니다. 자세한 내용은 the section called “업그레이드 후 다운그레이드할 수 없음”을(를) 참조하세요.
3. 업그레이드 – OpenSearch Service가 업그레이드를 시작합니다. 15분에서 몇 시간까지 걸릴 수 있습니다. 일부 또는 모든 업그레이드 도중 OpenSearch Dashboards를 사용하지 못할 수 있습니다.

도메인 업그레이드(콘솔)

업그레이드 프로세스는 되돌릴 수 없으며 일시 중지 또는 취소할 수 없습니다. 업그레이드 도중에는 도메인에서 구성을 변경할 수 없습니다. 업그레이드를 시작하기 전에 진행해도 좋은지 다시 한번 확인하세요. 동일한 단계를 사용해 실제로 업그레이드를 시작하지 않고 업그레이드 전 점검을 수행할 수 있습니다.

클러스터에 전용 프라이머리 노드가 있는 경우 OpenSearch 업그레이드는 가동 중지 없이 완료됩니다. 그렇지 않으면 클러스터가 업그레이드 후 프라이머리 노드를 선택하는 몇 초 동안 응답하지 않을 수도 있습니다.

최신 버전의 OpenSearch 또는 Elasticsearch로 도메인 업그레이드

1. 도메인의 수동 스냅샷을 생성합니다. 이전 OpenSearch 버전을 다시 사용하려는 경우 이 스냅샷은 새 도메인에 복원할 수 있는 백업으로 사용할 수 있습니다.
2. https://aws.amazon.com으로 이동하여 Sign In to the Console(콘솔에 로그인)을 선택합니다.
3. Analytics(분석)에서 Amazon OpenSearch Service를 선택합니다.
4. 탐색 창의 Domains(도메인)에서 업그레이드할 도메인을 선택합니다.
5. Actions(작업), Upgrade(업그레이드)를 선택합니다.
6. 업그레이드할 버전을 선택합니다. OpenSearch 버전으로 업그레이드하는 경우 Enable compatibility mode(호환성 모드 사용 설정) 옵션이 표시됩니다. 이 설정을 활성화하면 OpenSearch는 Elasticsearch OSS 클라이언트 및 Logstash와 같은 플러그인이 Amazon OpenSearch Service를 계속 사용할 수 있도록 해당 버전을 7.10으로 보고합니다. 나중에 이 설정을 비활성화할 수 있습니다
7. Upgrade(업그레이드)를 선택합니다.
8. 도메인 대시보드에서 Status(상태)를 확인하여 업그레이드 상태를 모니터링합니다.

도메인 업그레이드(CLI)

다음 작업을 사용하여 도메인에 올바른 OpenSearch 또는 Elasticsearch 버전을 식별하고, 인플레이스 업그레이드를 시작하고, 업그레이드 전 점검을 수행하고, 진행 상태를 확인할 수 있습니다.

- `get-compatible-versions` (`GetCompatibleVersions`)
- `upgrade-domain` (`UpgradeDomain`)
- `get-upgrade-status` (`GetUpgradeStatus`)
- `get-upgrade-history` (`GetUpgradeHistory`)

자세한 내용은 AWS CLI 명령 참조 및 Amazon OpenSearch Service API 참조를 참조하세요.

도메인 업그레이드(SDK)

이 샘플은의 OpenSearchService 하위 수준 Python 클라이언트 AWS SDK for Python (Boto) 를 사용하여 도메인이 특정 버전으로 업그레이드할 수 있는지 확인하고, 업그레이드하고, 업그레이드 상태를 지속적으로 확인합니다.

```
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()
```

검증 장애 문제 해결

OpenSearch 또는 Elasticsearch 버전 업그레이드를 시작하면 OpenSearch Service에서 먼저 일련의 검증 검사를 수행하여 도메인을 업그레이드할 수 있는지 확인합니다. 이러한 검사 중 하나라도 실패하면 도메인을 업그레이드하기 전에 수정해야 하는 특정 문제가 포함된 알림을 받게 됩니다. 잠재적 문제 목록 및 문제 해결 단계는 the section called “Troubleshooting validation errors(검증 오류 문제 해결 중)”을 참조하세요.

업그레이드 문제 해결

인플레이스 업그레이드는 정상 상태 도메인이 필요합니다. 도메인은 업그레이드 자격이 없거나 매우 다양한 이유로 업그레이드가 실패할 수 있습니다. 다음 표에는 가장 일반적인 문제가 나와 있습니다.

문제	설명
선택적 플러그인은 지원되지 않음	선택적 플러그인으로 도메인을 업그레이드하면 OpenSearch Service에서 플러그인도 자동으로 업그레이드합니다. 따라서 도메인의 대상 버전도 이러한 선택적 플러그인을 지원해야 합니다. 대상 버전에서 사용할 수 없는 선택적 플러그인이 도메인에 설치된 경우 업그레이드 요청이 실패합니다.
노드당 샤드가 너무 많음	OpenSearch뿐만 아니라 Elasticsearch 7.x 버전의 기본 설정에는 노드당 1,000개 이하의 샤드가 있습니다. 현재 클러스터의 노드가 이 설정을 초과하면 OpenSearch Service에서 업그레이드할 수 없습니다. 문제 해결 옵션은 the section called “최대 샤드 제한 초과” 섹션을 참조하세요.
처리 중 상태의 도메인	도메인이 구성 변경 도중에 있습니다. 작업이 완료된 후 업그레이드 자격을 확인하세요.
빨간색 클러스터 상태	클러스터에서 하나 이상의 인덱스가 빨간색입니다. 문제 해결 단계는 the section called “빨간색 클러스터 상태” 섹션을 참조하세요.
높은 오류율	클러스터가 요청을 처리하려고 시도할 때 다수의 5xx 오류를 반환합니다. 이 문제는 일반적으로 너무 많은 동시 읽기 또는 쓰기 요청의 결과입니다. 클러스터로 가는 트래픽을 줄이거나 도메인을 확장할 것을 고려하세요.
브레인 분할	브레인 분할은 클러스터가 여러 개의 프라이머리 노드를 가지고 자체적으로는 절대로 다시 조인되지 않는 2개의 클러스터로 분할되어 있다는 의미입니다. 권장 수의 전용 프라이머리 노드를 사용하면 브레인 분할을 방지할 수 있습니다. 브레인 분할로부터 복구하기 위해 도움이 필요하면 지원에 문의하세요.
프라이머리 노드가 없음	OpenSearch Service가 클러스터의 프라이머리 노드를 찾을 수 없습니다. 도메인에서 다중 AZ를 사용하는 경우 가용 영역 장애로 인해 클러스터가 쿼럼을 상실하고 새 프라이머리 노드를 선택하지 못할 수 있습니다. 문제가 자체적으로 해결되지 않을 경우 지원에 문의하세요.

문제	설명
대기 중 작업이 너무 많음	프라이머리 노드에 부하가 너무 높아 대기 중 작업이 많습니다. 클러스터로 가는 트래픽을 줄이거나 도메인을 확장할 것을 고려하세요.
손상된 스토리지 볼륨	하나 이상의 노드의 디스크 볼륨이 제대로 기능하지 않습니다. 이 문제는 흔히 높은 오류율, 대기 작업이 너무 많음 등 다른 문제와 함께 발생합니다. 이 문제가 단독으로 발생하고 자체적으로 해결되지 않을 경우 지원에 문의하세요.
KMS 키 문제	도메인을 암호화하는 데 사용된 KMS 키가 액세스 불가능하거나 없습니다. 자세한 내용은 the section called “저장된 데이터를 암호화하는 도메인 모니터링” 섹션을 참조하세요.
진행 중인 스냅샷	도메인이 현재 스냅샷을 생성하고 있습니다. 스냅샷이 완료된 후 업그레이드 자격을 확인하세요. 또한 수동 스냅샷 리포지토리를 나열하고, 해당 리포지토리에서 스냅샷을 나열하고, 수동 스냅샷을 생성할 수 있는지도 확인하세요. OpenSearch Service가 스냅샷이 진행 중인지 확인할 수 없는 경우 업그레이드가 실패할 수 있습니다.
스냅샷 시간 초과 또는 실패	업그레이드 전 스냅샷 생성이 너무 오래 걸렸거나 실패했습니다. 클러스터 상태를 확인한 후 다시 시도하세요. 문제가 지속될 경우 지원에 문의하세요.
호환되지 않는 인덱스	하나 이상의 인덱스가 대상 버전과 호환되지 않습니다. 인덱스를 이전 버전의 OpenSearch 또는 Elasticsearch에서 마이그레이션한 경우 이 문제가 발생할 수 있습니다. 인덱스를 다시 생성한 후 다시 시도하세요.
높은 디스크 사용량	클러스터의 디스크 사용량이 90%를 초과합니다. 데이터를 삭제하거나 도메인을 확장한 후 다시 시도하세요.
높은 JVM 사용량	JVM 메모리 압력이 75%를 초과합니다. 클러스터로 가는 트래픽을 줄이거나 도메인을 확장한 후 다시 시도하세요.
OpenSearch Dashboards 별칭 문제	`.dashboards` 가 이미 별칭으로 구성되고 호환되지 않는 인덱스(예를 들어 OpenSearch Dashboards의 이전 버전 중 하나)에 매핑되어 있습니다. 다시 인덱싱한 후 다시 시도하세요.

문제	설명
빨간색 Dashboards 상태	OpenSearch Dashboards 상태가 빨간색입니다. 업그레이드가 완료되면 Dashboards를 사용해보세요. 상태가 지속될 경우 수동으로 해결한 후 다시 시도하세요.
클러스터 간 호환성	업그레이드 후 소스 도메인과 대상 도메인 간 교차 클러스터 호환성이 유지되는 경우에만 업그레이드할 수 있습니다. 업그레이드 프로세스 중에 호환되지 않는 모든 연결이 식별됩니다. 계속하려면 원격 도메인을 업그레이드하거나 호환되지 않는 연결을 삭제하세요. 도메인에서 복제가 활성 상태인 경우 연결을 삭제한 후에는 복제를 재개할 수 없다는 점을 참조하세요.
기타 OpenSearch Service 서비스 문제	OpenSearch Service 자체에 문제가 있을 경우 도메인이 업그레이드 자격이 없는 것으로 표시될 수 있습니다. 도메인에 상기 조건이 하나도 적용되지 않지만 문제가 하루를 넘게 지속될 경우 지원에 문의하세요.

스냅샷을 사용하여 데이터 마이그레이션

인플레이스 업그레이드는 도메인을 새 OpenSearch 또는 Elasticsearch 버전으로 업그레이드하는 더 쉽고 빠르며 안정적인 방법입니다. 스냅샷은 5.1 이전 버전의 Elasticsearch에서 마이그레이션하거나 완전히 새 클러스터로 마이그레이션하려는 경우 적합한 옵션입니다.

다음 표에는 스냅샷을 통해 데이터를 다른 OpenSearch 또는 Elasticsearch 버전을 사용하는 도메인으로 마이그레이션하는 방법이 나와 있습니다. 스냅샷 생성 및 복원에 대한 자세한 내용은 the section called “인덱스 스냅샷 생성” 섹션을 참조하세요.

구 버전	새 버전	마이그레이션 프로세스
OpenSearch 1.3 또는 2.x	OpenSearch 2.x	1. OpenSearch 2.3에 대한 주요 변경 사항을 검토하여 인덱스 또는 애플리케이션을 조정할 필요가 있는지 확인하세요. 2. 1.3 또는 2.x 도메인의 수동 스냅샷을 생성합니다. 3. 원래 1.3 또는 2.x 도메인보다 더 높은 버전의 2.x 도메인을 생성합니다. 4. 원래 도메인의 스냅샷을 2.x 도메인에 복원합니다. 다음과 같이 작업 중에 새 이름으로 `.opensearch` 인덱스를 복원해야 할 수도 있습니다.

구 버전	새 버전	마이그레이션 프로세스

```
POST _snapshot/ <repository-name> /<snapshot-
name>/_restore
{
  "indices": "*",
  "ignore_unavailable": true,
  "rename_pattern": ".opensearch",
  "rename_replacement": ".backup-opensearc
h"
}
```

그런 다음 새 도메인에서 `.backup-opensearch` 를 다시 인덱싱하고 `.opensearch` 에 별칭을 지정할 수 있습니다. `_restore`의 기본값이 false이므로 `_restore` REST 호출에는 `include_global_state` 가 포함되지 않습니다. 따라서 테스트 도메인에는 인덱스 템플릿이 포함되지 않으며 백업의 전체 상태가 반영되지 않습니다.

5. 원래 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.

구 버전	새 버전	마이그레이션 프로세스

OpenSearch 1.x	OpenSearch 1.x	1. 1.x 도메인의 수동 스냅샷을 생성합니다.
		2. 원래 1.x 도메인보다 더 높은 버전의 1.x 도메인을 생성합니다.
		3. 원래 도메인의 스냅샷을 새로운 1.x 도메인에 복원합니다. 다음과 같이 작업 중에 새 이름으로 `.opensearch` 인덱스를 복원해야 할 수도 있습니다.
		
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".opensearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		
		그런 다음 새 도메인에서 `.backup-opensearch` 를 다시 인덱싱하고 `.opensearch` 에 별칭을 지정할 수 있습니다. `_restore`의 기본값이 false이므로 `_restore` REST 호출에는 `include_global_state` 가 포함되지 않습니다. 따라서 테스트 도메인에는 인덱스 템플릿이 포함되지 않으며 백업의 전체 상태가 반영되지 않습니다.
		4. 원래 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.

구 버전	새 버전	마이그레이션 프로세스

Elasticsearch 6.x 또는 7.x	OpenSearch 1.x	1. OpenSearch 1.0에 대한 주요 변경 사항을 검토하여 인덱스 또는 애플리케이션을 조정할 필요가 있는지 확인하세요.
		2. Elasticsearch 7.x 또는 6.x 도메인의 수동 스냅샷을 생성합니다.
		3. OpenSearch 1.x 도메인을 생성합니다.
		4. Elasticsearch 도메인의 스냅샷을 OpenSearch 도메인에 복원합니다. 다음과 같이 작업 중에 새 이름으로 `.elasticsearch` 인덱스를 복원해야 할 수도 있습니다.
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		그런 다음 새 도메인에서 `.backup-opensearch` 를 다시 인덱싱하고 `.elasticsearch` 에 별칭을 지정할 수 있습니다. `_restore`의 기본값이 false이므로 `_restore` REST 호출에는 `include_global_state` 가 포함되지 않습니다. 따라서 테스트 도메인에는 인덱스 템플릿이 포함되지 않으며 백업의 전체 상태가 반영되지 않습니다.
		5. 원래 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.

구 버전	새 버전	마이그레이션 프로세스

Elasticsearch 6.x	Elasticsearch 7.x	1. 7.0에 대한 주요 변경 사항을 검토하여 인덱스 또는 애플리케이션을 조정할 필요가 있는지 확인하세요.
		2. 6.x 도메인의 수동 스냅샷을 생성합니다.
		3. 7.x 도메인을 생성합니다.
		4. 원래 도메인의 스냅샷을 7.x 도메인에 복원합니다. 다음과 같이 작업 중에 새 이름으로 `.opensearch` 인덱스를 복원해야 할 수도 있습니다.
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-elasticsearch"`
		`}`
		그런 다음 새 도메인에서 `.backup-elasticsearch` 를 다시 인덱싱하고 `.elasticsearch` 에 별칭을 지정할 수 있습니다. `_restore`의 기본값이 false이므로 `_restore` REST 호출에는 `include_global_state` 가 포함되지 않습니다. 따라서 테스트 도메인에는 인덱스 템플릿이 포함되지 않으며 백업의 전체 상태가 반영되지 않습니다.
		5. 원래 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.

구 버전	새 버전	마이그레이션 프로세스
Elasticsearch 6.x	Elasticsearch 6.8	1. 6.x 도메인의 수동 스냅샷을 생성합니다. 2. 6.8 도메인을 생성합니다. 3. 원래 도메인의 스냅샷을 6.8 도메인에 복원합니다. 4. 원래 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.
Elasticsearch 5.x	Elasticsearch 6.x	1. 6.0에 대한 주요 변경 사항을 검토하여 인덱스 또는 애플리케이션을 조정할 필요가 있는지 확인하세요. 2. 5.x 도메인의 수동 스냅샷을 생성합니다. 3. 6.x 도메인을 생성합니다. 4. 원래 도메인의 스냅샷을 6.x 도메인에 복원합니다. 5. 5.x 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.
Elasticsearch 5.x	Elasticsearch 5.6	1. 5.x 도메인의 수동 스냅샷을 생성합니다. 2. 5.6 도메인을 생성합니다. 3. 원래 도메인의 스냅샷을 5.6 도메인에 복원합니다. 4. 원래 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.
Elasticsearch 2.3	Elasticsearch 6.x	Elasticsearch 2.3 스냅샷은 6.x와 호환되지 않습니다. 데이터를 2.3에서 6.x로 직접 마이그레이션하려면 새 도메인에서 인덱스를 수동으로 다시 만들어야 합니다. 또는 이 표의 2.3~5.x 단계에 따라 새 5.x 도메인에서 `_reindex` 작업을 수행하여 2.3 인덱스를 5.x 인덱스로 변환한 다음, 5.x~6.x 단계를 따르세요.

구 버전	새 버전	마이그레이션 프로세스
Elasticsearch 2.3	Elasticsearch 5.x	1. 5.0에 대한 주요 변경 사항을 검토하여 인덱스 또는 애플리케이션을 조정할 필요가 있는지 확인하세요. 2. 2.3 도메인의 수동 스냅샷을 생성합니다. 3. 5.x 도메인을 생성합니다. 4. 2.3 도메인의 스냅샷을 5.x 도메인에 복원합니다. 5. 2.3 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.
Elasticsearch 1.5	Elasticsearch 5.x	Elasticsearch 1.5 스냅샷은 5.x와 호환되지 않습니다. 데이터를 1.5에서 5.x로 마이그레이션하려면 새 도메인에서 인덱스를 수동으로 다시 만들어야 합니다. **Important** 1.5 스냅샷은 2.3과 호환되지만, OpenSearch Service 2.3 도메인에서는 `_reindex` 작업을 지원하지 않습니다. 인덱스를 다시 만들 수는 없기 때문에 1.5 도메인에서 만든 인덱스는 2.3 스냅샷에서 5.x 도메인으로 복원할 수 없습니다.

구 버전	새 버전	마이그레이션 프로세스
Elasticsearch 1.5	Elasticsearch 2.3	1. 마이그레이션 플러그인을 사용하여 2.3 버전으로 직접 업그레이드할 수 있는지 확인하세요. 마이그레이션 전에 데이터를 변경해야 할 수 있습니다. a. 웹 브라우저에서 `http://domain-endpoint /_plugin/migration/` 을 엽니다. b. Run checks now(지금 확인 실행)를 선택합니다. c. 결과를 검토하고, 필요하면 지침에 따라 데이터를 변경합니다. 2. 1.5 도메인의 수동 스냅샷을 생성합니다. 3. 2.3 도메인을 생성합니다. 4. 1.5 도메인의 스냅샷을 2.3 도메인에 복원합니다. 5. 1.5 도메인이 더 이상 필요 없는 경우에는 삭제합니다. 삭제하지 않으면 해당 도메인에 계속 요금이 부과될 수 있습니다.

Amazon OpenSearch Service에 대한 사용자 지정 엔드포인트 만들기

Amazon OpenSearch Service 도메인에 대한 사용자 지정 엔드포인트를 생성하면 더 쉽게 OpenSearch 및 OpenSearch Dashboards URL을 참조할 수 있습니다. 회사의 브랜딩을 포함하거나 표준 브랜드보다 짧고 기억하기 쉬운 엔드포인트를 사용할 수 있습니다.

새 도메인으로 전환해야 하는 경우 새 URL을 가리키도록 DNS를 업데이트하고 이전과 동일한 엔드포인트를 계속 사용합니다.

AWS Certificate Manager (ACM)에서 인증서를 생성하거나 자체 인증서를 가져와 사용자 지정 엔드포인트를 보호합니다.

새 도메인에 대한 사용자 지정 엔드포인트

OpenSearch Service 콘솔 AWS CLI또는 구성 API를 사용하여 새 OpenSearch Service 도메인에 대한 사용자 지정 엔드포인트를 활성화할 수 있습니다.

엔드포인트를 사용자 지정하려면(콘솔)

1. OpenSearch Service 콘솔에서 [도메인 생성(Create domain)]을 선택하고 도메인 이름을 입력합니다.
2. 사용자 지정 엔드포인트(Custom endpoint)에서 사용자 지정 엔드포인트 활성화(Enable custom endpoint)를 선택합니다.
3. 사용자 지정 호스트 이름(Custom hostname)에 원하는 사용자 지정 엔드포인트 호스트 이름을 입력합니다. 호스트 이름은 정규화된 도메인 이름(FQDN)이어야 합니다(예: www.yourdomain.com 또는 example.yourdomain.com).

   > **Note**
   >
   > 와일드카드 인증서가 없는 경우 사용자 지정 엔드포인트의 하위 도메인에 대한 새 인증서를 받아야 합니다.

4. AWS 인증서에서 도메인에 사용할 SSL 인증서를 선택합니다. 사용할 수 있는 인증서가 없는 경우 인증서를 ACM으로 가져오거나 ACM을 사용하여 인증서를 프로비저닝할 수 있습니다. 자세한 내용은 AWS Certificate Manager 사용 설명서의 인증서 발급 및 관리를 참조하세요.

   > **Note**
   >
   > 인증서에는 사용자 지정 엔드포인트 이름이 있어야 하며 OpenSearch Service 도메인과 동일한 계정에 있어야 합니다. 인증서 상태는 발급됨(ISSUED)이어야 합니다.

   - 나머지 단계에 따라 도메인을 생성하고 [생성(Create)]을 선택합니다.
   - 처리가 완료되면 도메인을 선택하여 사용자 지정 엔드포인트를 확인합니다.

   CLI 또는 구성 API를 사용하려면 `CreateDomain` 및 `UpdateDomainConfig` 작업을 수행합니다. 자세한 내용은 AWS CLI 명령 참조 및 Amazon OpenSearch Service API 참조를 참조하세요.

기존 도메인에 대한 사용자 지정 엔드포인트

기존 OpenSearch Service 도메인에 사용자 지정 엔드포인트를 추가하려면 [편집(Edit)]을 클릭하고 위의 2~4단계를 수행합니다.

CNAME 매핑

OpenSearch Service 도메인에 대해 사용자 지정 엔드포인트를 활성화한 후에는 Amazon Route 53(또는 선호하는 DNS 서비스 공급자)에서 CNAME 매핑을 생성할 수 있습니다. CNAME 매핑을 생성하면 트래픽을 사용자 지정 엔드포인트 및 해당 하위 도메인으로 라우팅할 수 있습니다. 이 매핑이 없으면 트래픽을 사용자 지정 엔드포인트로 라우팅할 수 없습니다. Route 53에서 이 매핑을 생성하는 단계는 Configuring DNS routing for a new domain 및 Creating a new hosted zone for a subdomain을 참조하세요. 다른 공급자의 경우 해당 설명서를 참조하세요.

사용자 지정 엔드포인트가 자동 생성된 도메인 엔드포인트를 가리키는 CNAME 레코드를 만듭니다. 도메인이 이중 스택인 경우 CNAME 레코드가 두 서비스 생성 엔드포인트 중 하나를 가리키도록 할 수 있습니다. 사용자 지정 엔드포인트의 이중 스택 기능은 CNAME 레코드가 가리키는 서비스 생성 엔드포인트에 따라 달라집니다. 사용자 지정 엔드포인트 호스트 이름은 CNAME 레코드의 이름이고 및 도메인 엔드포인트 호스트 이름은 CNAME 레코드의 값입니다.

OpenSearch Dashboards에 대한 SAML 인증을 사용하려면 새 SSO URL을 사용하여 IdP를 업데이트해야 합니다.

Amazon Route 53을 사용하여 별칭 레코드 유형을 생성해 도메인의 사용자 지정 엔드포인트에서 이중 스택 검색 엔드포인트를 가리킬 수 있습니다. 별칭 레코드 유형을 생성하려면 이중 스택 IP 주소 유형을 사용하도록 도메인을 구성해야 합니다. Route 53 API 작업을 사용하여 이 작업을 수행할 수 있습니다.

Route 53 API를 사용하여 별칭 레코드 유형을 생성하려면 도메인의 별칭 대상을 지정합니다. OpenSearch Service 콘솔의 사용자 지정 엔드포인트 섹션에 있는 호스팅 영역(이중 스택) 필드에서 또는 `DescribeDomain` API를 사용하고 `DomainEndpointV2HostedZoneId`의 값을 복사하여 도메인의 별칭 대상을 찾을 수 있습니다.

Amazon OpenSearch Service에 대한 자동 조정

Amazon OpenSearch Service의 자동 조정은 OpenSearch 클러스터의 성능 및 사용량 지표를 사용하여 대기열 및 캐시 크기, 노드의 JVM(Java 가상 머신) 설정 등 메모리 관련 구성 변경을 제안합니다. 이러한 선택적 변경 사항은 클러스터 속도와 안정성을 향상시킵니다.

일부 변경 사항은 즉시 배포되지만 다른 변경 사항은 도메인의 사용량이 적은 기간을 예약해야 합니다. 언제든지 기본 OpenSearch Service 설정으로 되돌릴 수 있습니다. 자동 조정은 도메인에 대한 성능 메트릭을 수집하고 분석하므로 알림(Notifications) 페이지의 OpenSearch Service 콘솔에서 권장 사항을 볼 수 있습니다.

Auto-Tune은 지원되는 인스턴스 유형으로 모든 OpenSearch 버전 또는 Elasticsearch 6.7 이상을 실행하는 도메인의 상용 AWS 리전 에서 사용할 수 있습니다.

변경 유형

자동 조정에는 크게 두 가지 범주의 변경 사항이 있습니다.

- 클러스터가 실행될 때 적용되는 비중단 변경 사항.
- 블루/그린 배포가 필요한 변경 사항은 도메인의 사용량이 적은 기간에 적용됩니다.

도메인의 성능 지표에 따라 자동 조정은 다음 설정에 대한 조정을 제안할 수 있습니다.

유형 변경	범주	설명
JVM 힙 크기	블루/그린	기본적으로 OpenSearch Service는 JVM 힙에 인스턴스 RAM의 50%를 사용합니다(최대 힙 크기 32GiB). 이 비율을 늘리면 OpenSearch에 더 많은 메모리가 제공되지만 운영 체제 및 기타 프로세스에서는 더 적은 양의 메모리를 사용할 수 있습니다. 값이 클수록 가비지 수집 일시 중지 횟수는 줄어들 수 있지만 일시 중지 시간은 늘어납니다.
JVM 신세대 설정	블루/그린	JVM “신세대” 설정은 사소한 가비지 수집의 빈도에 영향을 미칩니다. 사소한 수집이 더 자주 발생하면 주요 수집 및 일시 중지 수가 줄어들 수 있습니다.
대기열 크기	비중단	기본적으로 검색 대기열 크기는 `1000`이고 쓰기 대기열 크기는 `10000`입니다. 자동 조정은 요청을 처리하는 데 추가 힙을 사용할 수 있는 경우 검색 및 쓰기 대기열의 크기를 자동으로 조정합니다.
캐시 크기	비중단	이 필드 캐시는 힙 데이터 구조를 모니터링하므로 캐시 사용을 모니터링하는 것이 중요합니다. 자동 조정은 메모리 부족 및 회로 차단기 문제를 방지하기 위해 필드 데이터 캐시 크기를 조정합니다. 이 샤드 요청 캐시는 노드 수준에서 관리되며 기본 최대 크기는 힙의 1%입니다. 자동 조정은 구성된 클러스터가 처리할 수 있는 것보다 더 많은 검색 및 인덱스 요청을 허용하도록 샤드 요청 캐시 크기를 조정합니다.

유형 변경	범주	설명
요청 크기	비중단	기본적으로 진행 중인 요청의 집계된 크기가 전체 JVM의 10% 를 초과하는 경우(t2 인스턴스 타입일 경우 2%, t3.small일 경우 1%), OpenSearch는 기존 요청이 완료될 때까지 모든 새로운 _search 및 _bulk 요청을 제한합니다. 자동 조정은 현재 시스템에 사용되고 있는 JVM의 양에 따라 이 임계값(일반적으로 5~15%)을 자동으로 조정합니다. 예를 들어, JVM 메모리 부담이 크면 자동 조정이 임계값을 5%로 줄일 수 있습니다. 이때 클러스터가 안정화되고 임계값이 증가할 때까지 거부가 더 많이 표시될 수 있습니다.

자동 조정 활성화 또는 비활성화

OpenSearch Service는 새 도메인에서 기본적으로 자동 조정을 활성화합니다. 기존 도메인에서 자동 조정을 활성화하거나 비활성화하려면 콘솔을 사용하는 것이 좋습니다. 이렇게 하면 프로세스가 크게 간소화됩니다. 자동 조정을 활성화해도 블루/그린 배포는 발생하지 않습니다.

현재 AWS CloudFormation을 사용하여 자동 조정을 활성화 또는 비활성화할 수 없습니다.

콘솔

기존 도메인에서 자동 조정을 활성화하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 탐색 창의 도메인에서 도메인 이름을 선택하여 클러스터 구성을 엽니다.
3. 자동 조정이 아직 활성화되지 않은 경우 켜기를 선택합니다.
4. 필요에 따라 사용량이 적은 기간을 선택하여 도메인에 구성된 사용량이 적은 기간에 블루/그린 배포가 필요한 최적화를 예약할 수도 있습니다. 자세한 내용은 the section called “자동 조정 강화 예약” 단원을 참조하십시오.
5. 변경 사항 저장을 선택합니다.

CLI

를 사용하여 자동 조정을 활성화하려면 UpdateDomainConfig 요청을 AWS CLI전송합니다.

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options DesiredState=ENABLED
```

자동 조정 강화 예약

2023년 2월 16일 이전에 자동 조정은 유지 관리 기간을 사용하여 블루/그린 배포가 필요한 변경 일정을 잡았습니다. 유지 관리 기간은 이제 더 이상 사용되지 않으며, 일반적으로 도메인의 트래픽이 적은 일일 10시간의 시간대인 사용량이 적은 기간으로 대체되었습니다. 사용량이 적은 기간의 기본 시작 시간은 수정할 수 있지만 길이는 수정할 수 없습니다.

2023년 2월 16일에 사용량이 적은 기간이 도입되기 전에 자동 조정 유지 관리 기간을 활성화한 도메인에서는 중단 없이 기존 유지 관리 기간을 계속 사용할 수 있습니다. 단, 대신 도메인 유지 관리를 위해 사용량이 적은 기간을 사용하도록 기존 도메인을 마이그레이션하는 것이 좋습니다. 지침은 the section called “자동 조정 유지 관리 기간에서 마이그레이션하기” 섹션을 참조하세요.

콘솔

자동 조정 작업을 예약하려면 사용량이 적은 시간대에

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 탐색 창의 도메인에서 도메인 이름을 선택하여 클러스터 구성을 엽니다.
3. 자동 조정 탭으로 이동하여 편집을 선택합니다.
4. 자동 조정이 아직 활성화되지 않은 경우 켜기를 선택합니다.
5. 사용량이 적은 기간 중에 최적화 예약에서 사용량이 적은 기간을 선택합니다.
6. 변경 사항 저장을 선택합니다.

CLI

구성된 사용량이 적은 기간에 자동 조정 작업을 예약하도록 도메인을 구성하려면 UpdateDomainConfig 요청에 `UseOffPeakWindow`(을)를 포함하세요.

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=null
```

자동 조정 변경 사항 모니터링

Amazon CloudWatch에서 자동 조정 통계를 모니터링할 수 있습니다. 전체 지표 목록은 the section called “지표 자동 조정” 섹션을 참조하세요.

OpenSearch Service는 자동 조정 이벤트를 Amazon EventBridge로 보냅니다. EventBridge를 사용하여 이벤트 수신 시 이메일을 보내거나 특정 작업을 수행하는 규칙을 구성할 수 있습니다. EventBridge로 전송되는 각 자동 조정 이벤트 형식은 the section called “이벤트 자동 조정”(을)를 참조하세요.

Amazon OpenSearch Service 도메인 태그 지정

태그를 사용하면 Amazon OpenSearch Service 도메인에 임의 정보를 할당할 수 있으므로 해당 정보를 분류하고 필터링할 수 있습니다. 태그는 사용자가 정의하고 OpenSearch Service 도메인과 연결하는 키-값 페어입니다. 이러한 태그를 사용하여 비슷한 태그가 지정된 리소스에 대한 비용을 그룹화하여 비용을 추적할 수 있습니다. AWS 는 태그에 의미론적 의미를 적용하지 않습니다. 태그는 엄격히 문자열로 해석됩니다. 모든 태그에는 다음 요소가 포함되어 있습니다.

태그 요소	설명	필수
태그 키	태그 키는 태그의 이름입니다. 키는 연결된 OpenSearch Service 도메인에 대해 고유해야 합니다. 태그 키 및 값에 대한 기본 제한 사항 목록은 사용자 정의 태그 제한을 참조하세요.	예
태그 값	태그 값은 태그의 문자열 값입니다. 태그 값은 태그 세트에서 고유할 필요는 없으며 null일 수 있습니다. 예를 들어, project/Trinity 및 cost-center/Trinity의 태그 세트에 키-값 페어가 있을 수 있습니다. 태그 키 및 값에 대한 기본 제한 사항 목록은 사용자 정의 태그 제한을 참조하세요.	아니요

각 OpenSearch Service 도메인에는 해당 OpenSearch Service 도메인에 할당된 모든 태그를 포함하는 태그 세트가 있습니다. AWS 는 OpenSearch Service 도메인에 태그를 자동으로 할당하지 않습니다. 태그 세트는 0에서 50 사이의 태그를 포함할 수 있습니다. 기존 태그와 동일한 키가 있는 도메인에 태그를 추가하면 새 값이 이전 값을 덮어씁니다.

태그 예제

키를 사용하여 범주를 정의할 수 있으며 값은 해당 범주의 항목일 수 있습니다. 예를 들어, 태그 키를 `project`로 정의하고 태그 값을 `Salix`로 정의하여 OpenSearch Service 도메인이

Salix project에 배정됨을 나타낼 수 있습니다. 태그를 사용하여 `environment=test` 또는 `environment=production` 등의 키를 사용해 OpenSearch Service 도메인을 테스트나 프로덕션에 사용되도록 지정할 수도 있습니다. OpenSearch Service 도메인과 연결된 메타데이터를 더 쉽게 추적할 수 있게 일관성 있는 태그 키 세트를 사용합니다.

태그를 사용하여 자체 비용 구조를 반영하도록 AWS 청구서를 구성할 수도 있습니다. 이렇게 하려면 가입하여 태그 키 값이 포함된 AWS 계정 청구서를 가져옵니다. 그런 다음 같은 태그 키 값을 가진 리소스에 따라 결제 정보를 구성하여 리소스 비용의 합을 볼 수 있습니다. 예를 들어, 키-값 페어로 OpenSearch Service 도메인에 태그를 지정한 다음 결제 정보를 구성하여 여러 서비스에 걸친 각 도메인의 총비용을 볼 수 있습니다. 자세한 내용은 AWS Billing and Cost Management 설명서의 비용 할당 태그 사용을 참조하세요.

> **Note**
>
> 권한 부여 목적으로 태그가 캐시됩니다. 이 때문에 OpenSearch Service 도메인의 태그에 대한 추가나 업데이트가 제공되는 데 몇 분 정도 걸릴 수 있습니다.

도메인 태그 지정(콘솔)

콘솔은 도메인에 태그를 지정하는 가장 간단한 방법입니다.

태그를 만들려면(콘솔)

1. https://aws.amazon.com으로 이동하여 콘솔에 로그인(Sign In to the Console)을 선택합니다.
2. 분석(Analytics)에서 Amazon OpenSearch Service를 선택합니다.
3. 태그를 추가할 도메인을 선택한 다음 [태그(Tags)] 탭으로 이동합니다.
4. [관리(Manage)], [새 태그 추가(Add new tag)]를 선택합니다.
5. 태그 키와 선택 값을 입력합니다.
6. 저장을 선택합니다.

태그를 삭제하려면 동일한 단계를 따르고 [태그 관리(Manage tags)] 페이지에서 [제거(Remove)]를 선택합니다.

콘솔을 사용한 태그 작업에 대한 자세한 내용은 AWS 관리 콘솔 시작 안내서에서 Tag Editor를 참조하세요.

도메인 태그 지정(AWS CLI)

--add-tags 명령과 AWS CLI 함께를 사용하여 리소스 태그를 생성할 수 있습니다.

구문

```
add-tags --arn=<domain_arn> --tag-list Key=<key>,Value=<value>
```

파라미터	설명
`--arn`	태그를 연결하려는 OpenSearch Service 도메인의 Amazon 리소스 이름입니다.
`--tag-list`	공백으로 구분된 키-값 페어 세트로, 형식은 다음과 같습니다. `Key=<key>,Value=<value>`

예제

다음 예제에서는 logs 도메인에 대해 태그 2개를 생성합니다.

```
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
```

--remove-tags 명령을 사용하여 OpenSearch Service 도메인에서 태그를 제거할 수 있습니다.

구문

```
remove-tags --arn=<domain_arn> --tag-keys Key=<key>,Value=<value>
```

파라미터	설명
`--arn`	태그를 연결하려는 OpenSearch Service 도메인의 Amazon 리소스 이름(ARN)입니다.
`--tag-keys`	OpenSearch Service 도메인에서 제거하려는 공백으로 구분된 키-값 페어 세트입니다.

예제

다음 예제에서는 이전 예제에서 생성한 logs 도메인에서 태그 2개를 제거합니다.

```
aws opensearch remove-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-
keys service instances
```

--list-tags 명령을 사용하여 OpenSearch Service 도메인에 대한 기존 태그를 볼 수 있습니다.

구문

```
list-tags --arn=<domain_arn>
```

파라미터	설명
`--arn`	태그를 연결하려는 OpenSearch Service 도메인의 Amazon 리소스 이름(ARN)입니다.

예제

다음 예제에서는 logs 도메인에 대한 리소스 태그를 모두 나열합니다.

```
aws opensearch list-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs
```

도메인 태그 지정(AWS SDKs)

AWS SDKs(Android 및 iOS SDKs 제외)는 , `AddTags` `ListTags`및 작업을 포함하여 Amazon OpenSearch Service API 참조에 정의된 모든 `RemoveTags` 작업을 지원합니다. AWS SDKs 설치 및 사용에 대한 자세한 내용은 AWS 소프트웨어 개발 키트를 참조하세요.

Python

이 예제에서는 AWS SDK for Python(Boto)의 OpenSearchService 하위 수준 Python 클라이언트를 사용하여 도메인에 태그를 추가하고, 도메인에 연결된 태그를 나열하며, 도메인에서 태그를 제거합니다. `DOMAIN_ARN`, `TAG_KEY` 및 `TAG_VALUE`의 값을 입력해야 합니다.

```
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
```

Amazon OpenSearch Service 도메인에 대한 관리 작업 수행

Amazon OpenSearch Service는 도메인 관련 문제를 해결해야 하는 경우 세분화된 제어를 제공하는 여러 관리 옵션을 제공합니다. 이러한 옵션에는 데이터 노드에서 OpenSearch 프로세스를 다시 시작하는 기능과 데이터 노드를 다시 시작하는 기능이 포함됩니다.

OpenSearch Service는 노드 상태 파라미터를 모니터링하고 이상이 있을 경우 수정 조치를 취하여 도메인을 안정적으로 유지합니다. 노드에서 OpenSearch 프로세스를 다시 시작하고 노드 자체를 다시 시작하는 관리 옵션을 사용하면 이러한 완화 조치 중 일부를 제어할 수 있습니다.

AWS Management Console AWS CLI또는 AWS SDK를 사용하여 이러한 작업을 수행할 수 있습니다. 다음 섹션에서는 콘솔에서 이러한 작업을 수행하는 방법을 설명합니다.

Amazon OpenSearch Service의 노드에서 OpenSearch 프로세스 다시 시작

노드에서 OpenSearch 프로세스를 다시 시작하려면

1. OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 도메인을 선택합니다. 사용하려는 도메인의 이름을 선택합니다.
3. 도메인 세부 정보 페이지가 열리면 인스턴스 상태 탭으로 이동합니다.
4. 데이터 노드에서 처리를 재시작하려는 노트 옆에 있는 버튼을 선택합니다.
5. 작업 드롭다운을 선택하고 OpenSearch/ElasticSearch 프로세스 재시작을 선택합니다.
6. 모달에서 확인을 선택합니다.
7. 시작한 작업의 상태를 보려면 노드 이름을 선택하세요. 노드 세부 정보 페이지가 열린 후 노드 이름 아래에 있는 이벤트 탭을 선택하면 해당 노드와 관련된 이벤트 목록이 표시됩니다.

Amazon OpenSearch Service에서 데이터 노드 재부팅

데이터 노드를 재부팅하려면

1. OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 도메인을 선택합니다. 사용하려는 도메인의 이름을 선택합니다.
3. 도메인 세부 정보 페이지가 열리면 인스턴스 상태 탭으로 이동합니다.
4. 데이터 노드에서 처리를 재시작하려는 노트 옆에 있는 버튼을 선택합니다.
5. 작업 드롭다운을 선택하고 노드 재부팅을 선택합니다.
6. 모달에서 확인을 선택합니다.
7. 시작한 작업의 상태를 보려면 노드 이름을 선택하세요. 노드 세부 정보 페이지가 열린 후 노드 이름 아래에 있는 이벤트 탭을 선택하면 해당 노드와 관련된 이벤트 목록이 표시됩니다.

Amazon OpenSearch Service의 노드에서 OpenSearch Dashboards 프로세스 다시 시작

OpenSearch Dashboards(이전 Kibana) 프로세스를 다시 시작하여 고정 인터페이스, 로드 실패 또는 응답하지 않는 시각화와 같은 문제를 복구할 수 있습니다. OpenSearch Dashboards를 다시 시작하는 옵션은 Dashboards 프로세스를 적극적으로 실행 중인 노드에서만 사용할 수 있습니다. 대부분의 OpenSearch Service 도메인에서이 프로세스는 데이터 노드가 아닌 전용 조정자 노드에서 실행됩니다. 따라서 콘솔에서 작업 드롭다운을 열면 일반적으로 조정자 노드에만 옵션이 나타납니다. 자세한 내용은 the section called “전용 조정자 노드” 단원을 참조하십시오.

이 동작은 도메인 구성 방식에 따라 달라집니다.

- 전용 조정자 노드가 있는 도메인 - 대시보드 프로세스는 해당 노드에서만 실행되며 다시 시작 옵션만 표시합니다.
- 전용 조정자 노드가 없는 도메인 - 더 간단하거나 이전 설정에서는 대시보드가 대신 데이터 노드에서 실행될 수 있으며 재시작 옵션이 여기에 나타납니다.
- 마스터 노드 - 이러한 노드는 클러스터 상태 및 선택 관리 전용입니다. 대시보드를 실행하지 않으며 재시작 옵션을 표시하지 않습니다.

대시보드 프로세스를 실행하는 노드를 확인하려면 도메인의 클러스터 구성 섹션으로 이동하여 노드 역할을 검토합니다. 재시작 옵션은 대시보드 프로세스를 호스팅하는 노드에서만 사용할 수 있습니다.

노드에서 대시보드 프로세스를 다시 시작하려면

1. OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 도메인을 선택합니다. 사용하려는 도메인의 이름을 선택합니다.
3. 도메인 세부 정보 페이지가 열리면 인스턴스 상태 탭으로 이동합니다.
4. 대시보드 프로세스를 실행 중인 노드의 섹션에서 프로세스를 다시 시작하려는 노드 옆의 버튼을 선택합니다.
5. 작업 드롭다운을 선택하고 Dashboard/Kibana 프로세스 재시작을 선택합니다.
6. 모달에서 확인을 선택합니다.
7. 시작한 작업의 상태를 보려면 노드 이름을 선택하세요. 노드 세부 정보 페이지가 열린 후 노드 이름 아래에 있는 이벤트 탭을 선택하면 해당 노드와 관련된 이벤트 목록이 표시됩니다.

제한 사항

관리 옵션에는 다음과 같은 제한 사항이 있습니다.

- 관리 옵션은 Elasticsearch 버전 7.x 이상에서 지원됩니다.
- 관리 옵션은 Multi-AZ with Standby가 활성화된 도메인을 지원하지 않습니다.
- OpenSearch 및 Elasticsearch 프로세스 재시작과 데이터 노드 재부팅은 3개 이상의 데이터 노드가 있는 도메인에서 지원됩니다.
- Dashboards 및 Kibana 프로세스 지원은 2개 이상의 데이터 노드가 있는 도메인에서 지원됩니다.
- 노드에서 OpenSearch 프로세스를 다시 시작하거나 노드를 재부팅하려면 도메인이 빨간색 상태가 아니어야 하고 모든 인덱스에 복제본이 구성되어 있어야 합니다.

Amazon OpenSearch Service 직접 쿼리 작업

Amazon OpenSearch Service 다이렉트 쿼리를 사용하여 수집 파이프라인을 구축하지 않고도 Amazon CloudWatch Logs, Amazon S3 및 Amazon Security Lake의 데이터를 분석할 수 있습니다. 이제로 ETL 통합을 사용하면 OpenSearch SQL 또는 PPL을 사용하여 데이터를 쿼리하고 Discover에서 탐색할 수 있습니다.

시작하려면 OpenSearch Service 콘솔에서 데이터 소스를 구성합니다. Amazon S3의 경우 도메인 연결을 사용하고 Query Workbench에서 SQL을 사용하여 테이블을 생성합니다. CloudWatch Logs 및 Security Lake는 미리 구성된 소스와 AWS Glue Data Catalog 테이블을 사용합니다.

직접 쿼리 요금

OpenSearch Service 직접 쿼리를 사용하면 OpenSearch Service와 Amazon S3, Amazon CloudWatch Logs 및 Amazon Security Lake에서 데이터를 처리하고 저장하는 데 사용되는 리소스에 대해 별도의 요금이 발생합니다. 직접 쿼리를 실행하면 청구서에 DirectQuery OCUs 사용 유형으로 나열된 시간당 OpenSearch 컴퓨팅 단위(OCU) 요금이 표시됩니다.

직접 쿼리는 대화형 보기 쿼리와 인덱싱된 보기 쿼리의 두 가지 유형으로 구성됩니다.

- 대화형 쿼리는 데이터 선택기를 채우고 S3, CloudWatch Logs 또는 Security Lake의 데이터에 대한 분석을 수행하는 데 사용됩니다.

  Amazon S3 직접 쿼리의 경우 Discover에서 새 쿼리를 실행하면 OpenSearch Service는 최소 3분 동안 지속되는 새 세션을 시작합니다. OpenSearch Service는 후속 쿼리가 빠르게 실행되도록이 세션을 활성 상태로 유지합니다.

  CloudWatch Logs 및 Security Lake 쿼리의 경우 OpenSearch Service는 확장 세션을 유지하지 않고 별도의 사전 워밍된 작업으로 각 쿼리를 처리합니다.
- 인덱싱된 뷰 쿼리는 컴퓨팅을 사용하여 OpenSearch Service에서 인덱싱된 뷰를 유지합니다. 이러한 쿼리는 다양한 양의 데이터를 명명된 인덱스로 수집하기 때문에 일반적으로 시간이 더 오래 걸립니다. 데이터를 인덱싱하면 향후 대화형 쿼리를 더 빠르게 실행하거나 참조할 인덱스가 필요한 대시보드 또는 알림과 같은 고급 분석 기능을 잠금 해제할 수 있습니다.

  Amazon S3 데이터 소스의 경우 인덱싱된 데이터는 구매한 인스턴스 유형에 따라 도메인에 저장됩니다. CloudWatch Logs 및 Security Lake 연결 데이터 소스의 경우 인덱싱된 데이터는 OpenSearch Serverless 컬렉션에 저장되며, 여기서 인덱싱된 데이터(IndexingOCU), 검색된 데이터(SearchOCU) 및 GB로 저장된 데이터에 대한 요금이 부과됩니다.

자세한 내용은 Amazon OpenSearch Service 요금 내의 직접 쿼리 및 서버리스 섹션을 참조하세요.

직접 쿼리 제한 사항

일반 제한 사항

OpenSearch Service 직접 쿼리에는 다음 제한 사항이 적용됩니다.

- 일부 데이터 유형은 지원되지 않습니다. 지원되는 데이터 유형은 Parquet, CSV 및 JSON으로 제한됩니다.
- 시간이 지남에 따라 데이터 구조가 변경되는 경우 데이터 구조 변경을 고려하여 인덱싱된 뷰 또는 out-of-the-box 제공 통합을 업데이트해야 합니다.
- AWS CloudFormation 템플릿은 아직 지원되지 않습니다.
- OpenSearch SQL 및 OpenSearch PPL 문은 직접 쿼리를 사용하는 것과 비교하여 OpenSearch 인덱스로 작업할 때 서로 다른 제한 사항이 있습니다. 직접 쿼리는 JOINs, 하위 쿼리 및 조회와 같은 고급 명령을 지원하는 반면 OpenSearch 인덱스에서 이러한 명령에 대한 지원은 제한되거나 존재하지 않습니다. 자세한 내용은 the section called “지원되는 SQL 및 PPL 명령” 단원을 참조하십시오.

Amazon S3에 대한 제한 사항

Amazon S3에서 데이터를 직접 쿼리하는 경우 다음과 같은 추가 제한 사항이 적용됩니다.

- S3에 대한 직접 쿼리는 OpenSearch 버전 2.13 이상을 실행하는 OpenSearch Service 도메인에서만 사용할 수 있으며에 액세스해야 합니다 AWS Glue Data Catalog. OpenSearch Query Workbench에서 SQL을 사용하여 기존 AWS Glue Data Catalog 테이블을 다시 생성해야 합니다.
- S3에 대한 직접 쿼리를 사용하려면 Amazon S3에서 체크포인트 버킷을 지정해야 합니다. 이 버킷은 마지막 새로 고침 시간과 가장 최근에 수집된 데이터를 포함하여 인덱싱된 뷰의 상태를 유지합니다.
- OpenSearch 도메인 및는 동일한에 있어야 AWS Glue Data Catalog 합니다 AWS 계정. S3 버킷은 다른 계정에 있을 수 있지만(IAM 정책에 조건을 추가해야 함) 도메인 AWS 리전 과 동일한에 있어야 합니다.
- S3를 사용하는 OpenSearch Service 직접 쿼리는 Query Workbench에서 생성된 Spark 테이블만 지원합니다. AWS Glue Data Catalog 또는 Athena 내에서 생성된 테이블은 인덱싱된 뷰를 유지하는 데 필요한 Spark 스트리밍에서 지원되지 않습니다.
- OpenSearch 인스턴스 유형에는 선택한 특정 인스턴스 유형에 따라 10MiB 또는 100MiB의 네트워크 페이로드 제한이 있습니다.

Amazon CloudWatch Logs에 대한 제한 사항

CloudWatch Logs에서 데이터를 직접 쿼리하는 경우 다음과 같은 추가 제한 사항이 적용됩니다.

- CloudWatch Logs와의 직접 쿼리 통합은 OpenSearch Service 컬렉션 및 OpenSearch 사용자 인터페이스에서만 사용할 수 있습니다.
- OpenSearch Serverless 컬렉션에는 100MiB의 네트워크 페이로드 제한이 있습니다.
- CloudWatch Logs는 콘솔에서 설치된 VPC 흐름, CloudTrail 및 AWS WAF 대시보드 통합을 지원합니다.

Amazon Security Lake에 대한 제한 사항

Security Lake에서 데이터를 직접 쿼리하는 경우 다음과 같은 추가 제한 사항이 적용됩니다.

- Security Lake와의 직접 쿼리 통합은 OpenSearch Service 컬렉션 및 OpenSearch 사용자 인터페이스에서만 사용할 수 있습니다.
- OpenSearch Serverless 컬렉션에는 100MiB의 네트워크 페이로드 제한이 있습니다.
- Security Lake에 대한 테이블 관리는 Lake Formation에서 수행됩니다.
- Security Lake는 구체화된 뷰만 인덱싱된 뷰로 지원합니다. 커버링 인덱스는 지원되지 않습니다.

Amazon OpenSearch Service에서 직접 쿼리를 사용하기 위한 권장 사항

이 페이지에서는 Amazon OpenSearch Service 다이렉트 쿼리를 사용하여 CloudWatch Logs, Amazon S3 및 Amazon Security Lake의 데이터를 분석하기 위한 권장 사항을 제공합니다. 이러한 모범 사례를 통해 데이터 수집이나 중복 없이 성능을 최적화하고 효율적인 쿼리를 보장할 수 있습니다.

주제

- 일반 권장 사항
- Amazon S3 권장 사항
- CloudWatch Logs 권장 사항
- Security Lake 권장 사항

일반 권장 사항

직접 쿼리를 사용할 때는 다음을 수행하는 것이 좋습니다.

- COALESCE SQL 함수를 사용하여 누락된 열을 처리하고 결과가 반환되는지 확인합니다.

- 쿼리에서 제한을 사용하여 너무 많은 데이터를 가져오지 않도록 합니다.
- 동일한 데이터 세트를 여러 번 분석하려는 경우 인덱싱된 뷰를 생성하여 데이터를 완전히 수집하여 OpenSearch로 인덱싱하고 분석을 완료하면 삭제합니다.
- 더 이상 필요하지 않은 경우 가속화 작업과 인덱스를 삭제합니다.
- 동일하지만 경우에 따라서만 다른 필드 이름을 포함하는 쿼리(예: field1 및 FIELD1)는 지원되지 않습니다.

  예를 들어 다음 쿼리는 지원되지 않습니다.

  ```
  Select AWSAccountId, AwsAccountId from LogGroup
  Select a.@LogStream, b.@logStream from Table A INNER Join Table B ona.id = b.id
  ```

  그러나 필드 이름(@logStream)이 두 로그 그룹에서 동일하기 때문에 다음 쿼리가 지원됩니다.

  ```
  Select a.@logStream, b.@logStream from Table A INNER Join Table B on a.id = b.id
  ```

- 함수와 표현식은 필드 이름에서 작동해야 하며 FROM절에 지정된 로그 그룹이 있는 SELECT 문의 일부여야 합니다.

  예를 들어이 쿼리는 지원되지 않습니다.

  ```
  SELECT cos(10) FROM LogGroup
  ```

  이 쿼리는 다음과 같이 지원됩니다.

  ```
  SELECT cos(field1) FROM LogGroup
  ```

Amazon S3 권장 사항

Amazon OpenSearch Service를 사용하여 Amazon S3에서 쿼리 데이터를 전달하는 경우 다음 사항도 권장합니다.

- 연도, 월, 일, 시간의 파티션 형식을 사용하여 Amazon S3에 데이터를 수집하여 쿼리 속도를 높입니다.
- 건너뛰기 인덱스를 빌드할 때 카디널리티가 높은 필드에는 블룸 필터를 사용하고 값 범위가 큰 필드에는 최소/최대 인덱스를 사용합니다. 카디널리티가 높은 필드의 경우 값 기반 접근 방식을 사용하여 쿼리 효율성을 개선하는 것이 좋습니다.
- 인덱스 상태 관리를 사용하여 구체화된 뷰 및 커버링 인덱스에 대한 스토리지를 유지합니다.

CloudWatch Logs 권장 사항

Amazon OpenSearch Service를 사용하여 CloudWatch Logs에서 쿼리 데이터를 전달하는 경우 다음 사항도 권장합니다.

- 하나의 쿼리에서 여러 로그 그룹을 검색할 때는 적절한 구문을 사용합니다. 자세한 내용은 the section called “다중 로그 그룹 함수” 단원을 참조하십시오.
- SQL 또는 PPL 명령을 사용하는 경우 특정 필드를 백틱에 묶어 성공적으로 쿼리합니다. 백틱은 특수 문자(비영문자 및 비숫자)가 있는 필드에 필요합니다. 예를 들어 `@message`, `Operation.Export`, 및를 백틱`Test::Field`으로 묶습니다. 순전히 알파벳 이름의 열을 백틱으로 묶을 필요는 없습니다.

  간단한 필드가 있는 쿼리의 예:

  ```
  SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
  LIMIT 1000;
  ```

  백틱이 추가된 유사한 쿼리:

  ```
  SELECT `@SessionToken`, `@Operation`, `@StartTime` FROM `LogGroup-A`
  LIMIT 1000;
  ```

Security Lake 권장 사항

Amazon OpenSearch Service를 사용하여 Security Lake에서 쿼리 데이터를 전달하는 경우 다음 사항도 권장합니다.

- Security Lake 상태를 확인하고 문제 없이 원활하게 실행되고 있는지 확인합니다. 자세한 문제 해결 단계는 Amazon Security Lake 사용 설명서의 데이터 레이크 상태 문제 해결을 참조하세요.
- 쿼리 액세스를 확인합니다.

  - Security Lake 위임된 관리자 계정과 다른 계정에서 Security Lake를 쿼리하는 경우 Security Lake에서 쿼리 액세스 권한이 있는 구독자를 설정합니다.
  - 동일한 계정에서 Security Lake를 쿼리하는 경우 관리형 S3 버킷을 LakeFormation에 등록하는 방법에 대한 Security Lake의 메시지가 있는지 확인합니다.
- 쿼리 템플릿과 사전 구축된 대시보드를 탐색하여 분석을 바로 시작할 수 있습니다.
- Open Cybersecurity Schema Framework(OCSF) 및 Security Lake에 대해 알아봅니다.
  - OCSF GitHub 리포지토리의 AWS 소스에 대한 스키마 매핑 예제 검토
  - AWS 소스 버전 2(OCSF 1.1.0)에 대한 Security Lake 쿼리를 방문하여 Security Lake를 효과적으로 쿼리하는 방법을 알아봅니다.
  - `accountid`, `region`및 파티션을 사용하여 쿼리 성능 개선 `time_dt`
- Security Lake가 쿼리를 지원하는 SQL 구문에 익숙해지세요. 자세한 내용은 the section called “지원되는 SQL 명령” 단원을 참조하십시오.

직접 쿼리 할당량

계정에는 OpenSearch Service 직접 쿼리와 관련된 다음과 같은 할당량이 있습니다.

Amazon S3 할당량

Amazon S3 데이터 소스에 대한 쿼리를 시작할 때마다 OpenSearch Service는 세션을 열고 최소 3분 동안 연결을 유지합니다. 이렇게 하면 후속 쿼리에서 세션 시작 시간을 제거하여 쿼리 지연 시간을 줄일 수 있습니다.

설명	Maximum	재정의 가능
도메인당 연결 수	10	예
도메인당 데이터 소스 수	20	예
도메인당 인덱스 수	5	예
데이터 소스별 동시 세션 수	10	예
쿼리당 최대 OCU	60	예
최대 쿼리 실행 시간(분)	30	예

설명	Maximum	재정의 가능
가속화당 최대 OCU	20	예
최대 임시 스토리지	20	예

CloudWatch Logs 할당량

> **Note**
>
> CloudWatch Logs Insights를 사용하여 직접 쿼리를 수행하려는 경우를 참조해야 합니다the section called “CloudWatch Logs Insights”.

설명	값	소프트 제한?	Notes
직접 쿼리 APIs 전반의 계정 수준 TPS 제한	3TPS	예	
최대 데이터 소스 수	20	예	한도는 당입니다 AWS 계정.
최대 자동 새로 고침 인덱스 또는 구체화된 뷰	30	예	제한은 데이터 소스당입니다.
최대 동시 쿼리 수	15	예	한도는 보류 중이거나 실행 중인 상태의 쿼리에 적용됩니다. 대화형 쿼리(예:와 같은 데이터 검색 명령SELECT) 및 인덱스 쿼리(예: CREATE/ALTER/와 같은 작업)를 포함합니다DROP.
쿼리당 최대 동시 OCU	512	예	OpenSearch 컴퓨팅 유닛(OCU). 각각 vCPU 16개와 32GB 메모리가 있는 실행기 15개와 드라이버 1개를 기준으로 제한합니다. 동시 처리 능력을 나타냅니다.

설명	값	소프트 제한?	Notes
분 단위의 최대 쿼리 실행 시간	60	아니요	CloudWatch Logs Insights의 OpenSearch PPL/SQL 쿼리에는 제한이 적용됩니다.
오래된 쿼리 IDs를 제거하는 기간	90일	예	OpenSearch Service가 이전 항목에 대한 쿼리 메타데이터를 제거하는 기간입니다. 예를 들어 90일이 지난 쿼리의 경우 GetDirectQuery 또는 GetDirectQueryResult 호출이 실패합니다.

Security Lake 할당량

설명	값	소프트 제한?	Notes
직접 쿼리 APIs 전반의 계정 수준 TPS 제한	3TPS	예	
최대 데이터 소스 수	20	예	한도는 당입니다 AWS 계정.
최대 자동 새로 고침 인덱스 또는 구체화된 뷰	30	예	데이터 소스당 제한이 적용됩니다. 자동 새로 고침이 true로 설정된 인덱스 및 구체화된 뷰(MVs)만 포함합니다.
최대 동시 쿼리 수	30	예	한도는 보류 중이거나 실행 중인 상태의 쿼리에 적용됩니다. 대화형 쿼리(예:와 같은 데이터 검색 명령SELECT) 및 인덱스 쿼리(예: CREATE/ALTER/와 같은 작업)를 포함합니다DROP.
쿼리당 최대 동시 OCU	512	예	OpenSearch 컴퓨팅 유닛(OCU). 각각 vCPU 16개와 32GB 메모리가 있는 실행기 15개와 드라이버 1개를 기준으로 제한합니다. 동시 처리 능력을 나타냅니다.

설명	값	소프트 제한?	Notes
분 단위의 최대 쿼리 실행 시간	30	아니요	대화형 쿼리(예:와 같은 데이터 검색 명령SELECT)에만 적용됩니다. REFRESH 쿼리의 경우 제한은 6시간입니다.
오래된 쿼리 IDs를 제거하는 기간	90일	예	OpenSearch Service가 이전 항목에 대한 쿼리 메타데이터를 제거하는 기간입니다. 예를 들어 90일이 지난 쿼리의 경우 GetDirectQuery 또는 GetDirectQueryResult 호출이 실패합니다.

지원됨 AWS 리전

Amazon S3, CloudWatch Logs 및 Security Lake의 OpenSearch Service 직접 쿼리에 대해 다음이 지원 AWS 리전 됩니다.

Amazon S3 AWS 리전 에서 사용 가능

- 아시아 태평양(홍콩)
- 아시아 태평양(뭄바이)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(스톡홀름)
- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(오리건)

CloudWatch Logs에 사용 가능 AWS 리전

- 아시아 태평양(뭄바이)
- 아시아 태평양(홍콩)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(스톡홀름)
- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(오리건)
- 유럽(파리)
- 유럽(런던)
- 남아메리카(상파울루)

Security Lake에 사용 가능 AWS 리전

- 아시아 태평양(뭄바이)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(스톡홀름)
- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(오리건)

- 유럽(파리)
- 유럽(런던)
- 남아메리카(상파울루)

OpenSearch Service에서 Amazon S3 데이터 직접 쿼리

이 섹션에서는 Amazon OpenSearch Service에서 데이터 소스 통합을 생성하고 구성하는 프로세스를 안내하므로 Amazon S3에 저장된 데이터를 효율적으로 쿼리하고 분석할 수 있습니다.

다음 페이지에서는 Amazon S3 직접 쿼리 데이터 소스를 설정하고, 필요한 사전 조건을 탐색하고, AWS Management Console 및 OpenSearch Service API를 모두 사용하여 step-by-step 절차를 따르는 방법을 알아봅니다. 또한 AWS Glue Data Catalog 역할 매핑 및 OpenSearch Dashboards에서 액세스 제어 구성을 비롯한 중요한 다음 단계도 다룹니다.

주제

- OpenSearch Service에서 Amazon S3 데이터 소스 통합 생성
- OpenSearch Dashboards에서 S3 데이터 소스 구성 및 쿼리

OpenSearch Service에서 Amazon S3 데이터 소스 통합 생성

AWS Management Console 또는 API를 통해 OpenSearch Service에 대한 새 Amazon S3 직접 쿼리 데이터 소스를 생성할 수 있습니다. 각 새 데이터 소스는를 사용하여 Amazon S3 버킷 AWS Glue Data Catalog 을 나타내는 테이블을 관리합니다.

주제

- 사전 조건
- 절차
- 다음 단계
- AWS Glue Data Catalog 역할 매핑
- 추가 리소스

사전 조건

시작하기 전에 다음 설명서를 검토했는지 확인합니다.

- the section called “Amazon S3에 대한 제한 사항”

- the section called “Amazon S3 권장 사항”
- the section called “Amazon S3 할당량”

데이터 소스를 생성하려면 먼저에 AWS 계정다음 리소스가 있어야 합니다.

- 버전이 2.13 이상인 OpenSearch 도메인. 이는 직접 쿼리 통합을 설정하기 위한 기반입니다. 이를 설치하는 지침은 the section called “ OpenSearch Service 도메인 생성” 섹션을 참조하세요.
- 하나 이상의 S3 버킷. 쿼리하려는 데이터가 포함된 버킷과 쿼리 체크포인트를 저장할 버킷을 지정해야 합니다. S3 버킷 생성에 대한 지침은 Amazon S3 사용 설명서의 버킷 생성을 참조하세요.
- (선택 사항) 하나 이상의 AWS Glue 테이블. Amazon S3에서 데이터를 쿼리하려면 S3 데이터를 가리키 AWS Glue Data Catalog 도록에 테이블이 설정되어 있어야 합니다. OpenSearch Query Workbench를 사용하여 테이블을 생성해야 합니다. 기존 Hive 테이블은 호환되지 않습니다.

  Amazon S3 데이터 소스를 처음 설정하는 경우 관리자 데이터 소스를 생성하여 모든 AWS Glue Data Catalog 테이블을 구성해야 합니다. OpenSearch out-of-the-box 통합을 설치하거나 OpenSearch Query Workbench를 사용하여 고급 사용 사례를 위한 사용자 지정 SQL 테이블을 생성하여이 작업을 수행할 수 있습니다. VPC, CloudTrail 및 AWS WAF 로그용 테이블 생성에 대한 예제는 VPC, CloudTrail 및 용 GitHub의 설명서를 참조하세요AWS WAF. 테이블을 생성한 후 새 Amazon S3 데이터 소스를 생성하고 제한된 테이블로 액세스를 제한할 수 있습니다.
- (선택 사항) 수동으로 생성한 IAM 역할입니다. 이 역할을 사용하여 데이터 소스에 대한 액세스를 관리할 수 있습니다. 또는 필요한 권한을 사용하여 OpenSearch Service가 자동으로 역할을 생성하도록 할 수 있습니다. 수동으로 생성한 IAM 역할을 사용하도록 선택한 경우의 지침을 따릅니다the section called “수동으로 생성한 IAM 역할에 필요한 권한”.

절차

AWS Management Console 또는 OpenSearch Service API를 사용하여 도메인에 직접 쿼리 데이터 소스를 설정할 수 있습니다.

를 사용하여 데이터 소스를 설정하려면 AWS Management Console

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 Domains(도메인)를 선택합니다.
3. 새 데이터 소스를 설정하려는 도메인을 선택합니다. 그러면 도메인 세부 정보 페이지가 열립니다.
4. 일반 도메인 세부 정보 아래에서 연결 탭을 선택하고 직접 쿼리 섹션을 찾습니다.
5. 데이터 소스 구성을 선택합니다.

6. 새 데이터 소스의 이름과 선택적 설명을 입력합니다.
7. Amazon S3 with AWS Glue Data Catalog를 선택합니다.
8. IAM 권한 액세스 설정에서 액세스 관리 방법을 선택합니다.

   a. 이 데이터 소스에 대한 역할을 자동으로 생성하려면 다음 단계를 따르세요.

      i. 새 역할 생성을 선택합니다.
      ii. IAM 역할의 이름을 입력합니다.
      iii. 쿼리하려는 데이터가 포함된 S3 버킷을 하나 이상 선택합니다.
      iv. 쿼리 체크포인트를 저장할 체크포인트 S3 버킷을 선택합니다.
      v. 하나 이상의 AWS Glue 데이터베이스 또는 테이블을 선택하여 쿼리할 데이터를 정의합니다. 테이블이 아직 생성되지 않은 경우 기본 데이터베이스에 대한 액세스를 제공합니다.

   b. 직접 관리하는 기존 역할을 사용하려면 다음 단계를 따르세요.

      i. 기존 역할 사용을 선택합니다.
      ii. 드롭다운 메뉴에서 기존 역할을 선택합니다.

> **Note**
>
> 자체 역할을 사용하는 경우 IAM 콘솔에서 필요한 정책을 연결하여 필요한 모든 권한이 있는지 확인해야 합니다. 자세한 내용은의 샘플 정책을 참조하세요the section called “수동으로 생성한 IAM 역할에 필요한 권한”.

9. 구성을 선택합니다. 그러면 OpenSearch 대시보드 URL이 포함된 데이터 소스 세부 정보 화면이 열립니다. 이 URL로 이동하여 다음 단계를 완료할 수 있습니다.

OpenSearch Service API

AddDataSource API 작업을 사용하여 도메인에 새 데이터 소스를 생성합니다.

```
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"
}
```

다음 단계

OpenSearch 대시보드 방문

데이터 소스를 생성한 후 OpenSearch Service는 OpenSearch 대시보드 링크를 제공합니다. 이를 사용하여 액세스 제어를 구성하고, 테이블을 정의하고, out-of-the-box 통합을 설치하고, 데이터를 쿼리할 수 있습니다.

자세한 내용은 the section called “S3 데이터 소스 구성” 단원을 참조하십시오.

AWS Glue Data Catalog 역할 매핑

데이터 소스를 생성한 후 세분화된 액세스 제어를 활성화한 경우 직접 쿼리를 실행하려면 관리자가 아닌 사용자를 AWS Glue Data Catalog 액세스 권한이 있는 IAM 역할에 매핑해야 합니다. IAM `glue_access` 역할에 매핑할 수 있는 백엔드 역할을 수동으로 생성하려면 다음 단계를 수행합니다.

> **Note**
>
> 인덱스는 데이터 소스에 대한 모든 쿼리에 사용됩니다. 지정된 데이터 소스의 요청 인덱스에 대한 읽기 액세스 권한이 있는 사용자는 해당 데이터 소스에 대한 모든 쿼리를 읽을 수 있습니다. 결과 인덱스에 대한 읽기 액세스 권한이 있는 사용자는 해당 데이터 소스에 대한 모든 쿼리의 결과를 읽을 수 있습니다.

1. OpenSearch 대시보드의 기본 메뉴에서 보안, 역할 및 역할 생성을 선택합니다.
2. 역할 이름을 glue_access로 지정합니다.
3. 클러스터 권한에서 `indices:data/write/bulk*`, `indices:data/read/scroll`, `indices:data/read/scroll/clear`를 선택합니다.
4. 인덱스에는 역할 액세스 권한이 있는 사용자에게 부여하려는 다음과 같은 인덱스를 입력합니다.
   - `.query_execution_request_<name of data source>`
   - `query_execution_result_<name of data source>`

- `.async-query-scheduler`
- `flint_*`

5. 인덱스 권한에서 `indices_all`을 선택합니다.
6. 생성(Create)을 선택합니다.
7. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다.
8. 백엔드 역할에서 도메인 호출 권한이 필요한 AWS Glue 역할의 ARN을 추가합니다.

```
arn:aws:iam::account-id:role/role-name
```

9. 맵을 선택하고 매핑된 사용자에 역할이 나타나는지 확인합니다.

역할 매핑에 대한 자세한 내용은 the section called “사용자에 역할 매핑” 섹션을 참조하십시오.

추가 리소스

수동으로 생성한 IAM 역할에 필요한 권한

도메인에 대한 데이터 소스를 생성할 때 데이터에 대한 액세스를 관리할 IAM 역할을 선택합니다. 여기에는 두 가지 옵션이 있습니다.

1. 새 IAM 역할 자동 생성
2. 수동으로 생성한 기존 IAM 역할 사용

수동으로 생성한 역할을 사용하는 경우 역할에 올바른 권한을 연결해야 합니다. 권한은 특정 데이터 소스에 대한 액세스를 허용하고 OpenSearch Service가 역할을 수임하도록 허용해야 합니다. 이는 OpenSearch Service가 데이터에 안전하게 액세스하고 상호 작용할 수 있도록 하기 위해 필요합니다.

다음 샘플 정책은 데이터 소스를 생성하고 관리하는 데 필요한 최소 권한을 보여줍니다. `s3:*` 또는 `AdminstratorAccess` 정책과 같이 더 광범위한 권한이 있는 경우 이러한 권한에는 샘플 정책의 최소 권한 권한이 포함된다는 점에 유의하시기 바랍니다.

다음 샘플 정책에서 *## ### ####* 자신의 정보로 바꿉니다.

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/*"
         ]
      }
   ]
```

```
}
```

다른 계정의 Amazon S3 버킷을 지원하려면 Amazon S3 정책에 조건을 포함하고 적절한 계정을 추가해야 합니다.

다음 샘플 조건에서 *## ### ####* 자신의 정보로 바꿉니다.

```
"Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{accountId}}"
                }
```

역할에는 대상 ID를 지정하는 다음과 같은 신뢰 정책도 있어야 합니다.

JSON

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

역할을 생성하기 위한 지침은 사용자 지정 신뢰 정책을 사용하여 역할 생성을 참조하세요.

OpenSearch Service에서 세분화된 액세스 제어를 활성화한 경우 데이터 소스에 대해 새 OpenSearch 세분화된 액세스 제어 역할이 자동으로 생성됩니다. 새 세분화된 액세스 제어 역할의 이름은 `AWSOpenSearchDirectQuery` *<name of data source>*입니다.

기본적으로 역할에는 직접 쿼리 데이터 소스 인덱스에 대한 액세스만 있습니다. 데이터 소스에 대한 액세스 권한을 제한하거나 부여하도록 역할을 구성할 수 있지만 이 역할의 액세스 권한을 조정하지 않는 것이 좋습니다. 데이터 소스를 삭제하면 이 역할이 삭제됩니다. 조정할 경우 다른 사용자가 역할에 매핑된 경우 다른 사용자의 액세스 권한이 제거됩니다.

OpenSearch Dashboards에서 S3 데이터 소스 구성 및 쿼리

이제 데이터 소스를 생성했으므로 보안 설정을 구성하거나, Amazon S3 테이블을 정의하거나, 가속화된 데이터 인덱싱을 설정할 수 있습니다. 실제로 데이터를 쿼리하기 전에, 이 섹션에서 OpenSearch 대시보드의 다양한 데이터 소스 사용 사례를 살펴보겠습니다.

다음 섹션을 구성하려면 먼저 OpenSearch 대시보드에서 데이터 소스로 이동해야 합니다. 왼쪽 탐색에서 데이터 관리 아래에 있는 데이터 소스를 선택합니다. 데이터 소스 관리에서 콘솔에서 생성한 데이터 소스의 이름을 선택합니다.

쿼리 워크벤치를 사용하여 Spark 테이블 생성

OpenSearch Service에서 Amazon S3로의 직접 쿼리는 AWS Glue Data Catalog내의 Spark 테이블을 사용합니다. OpenSearch 대시보드를 종료하지 않고도 쿼리 워크벤치 내에서 테이블을 생성할 수 있습니다.

데이터 소스의 기존 데이터베이스 및 테이블을 관리하거나 직접 쿼리를 사용할 새 테이블을 생성하려면 왼쪽 탐색에서 쿼리 워크벤치를 선택하고 데이터 소스 드롭다운에서 Amazon S3 데이터 소스를 선택합니다.

S3에 Parquet 형식으로 저장된 VPC 흐름 로그에 대한 테이블을 설정하려면 다음 쿼리를 실행합니다.

```
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"
```

테이블을 생성한 후 다음 쿼리를 실행하여 직접 쿼리와 호환되는지 확인합니다.

```
MSCK REPAIR TABLE  datasourcename.databasename.vpclogstable
```

널리 사용되는 AWS 로그 유형에 대한 통합 설정

Amazon S3에 저장된 AWS 로그 유형을 OpenSearch Service와 통합할 수 있습니다. OpenSearch 대시보드를 사용하여 AWS Glue Data Catalog 테이블, 저장된 쿼리 및 대시보드를 생성하는 통합을 설치합니다. 이러한 통합은 인덱싱된 보기를 사용하여 대시보드를 최신 상태로 유지합니다.

통합 설치 지침은 OpenSearch 설명서의 통합 자산 설치를 참조하세요.

통합을 선택할 때 `S3 Glue` 태그가 있는지 확인합니다.

통합을 설정할 때 연결 유형에 S3 연결을 지정합니다. 그런 다음 통합을 위한 데이터 소스, 데이터의 Amazon S3 위치, 가속 인덱싱을 관리할 체크포인트, 사용 사례에 필요한 자산을 선택합니다.

> **Note**
>
> 체크포인트의 S3 버킷에 체크포인트 위치에 대한 쓰기 권한이 있는지 확인합니다. 이러한 권한이 없으면 통합의 가속화가 실패합니다.

액세스 제어 설정

데이터 소스의 세부 정보 페이지에서 액세스 제어 섹션을 찾아 편집을 선택합니다. 도메인에 세분화된 액세스 제어가 활성화된 경우 제한을 선택하고 새 데이터 소스에 대한 액세스 권한을 제공할 역할을 선택합니다. 관리자만 데이터 소스에 액세스하도록 하려는 경우 관리자 전용을 선택할 수도 있습니다.

> **Important**
>
> 인덱스는 데이터 소스에 대한 모든 쿼리에 사용됩니다. 지정된 데이터 소스의 요청 인덱스에 대한 읽기 액세스 권한이 있는 사용자는 해당 데이터 소스에 대한 모든 쿼리를 읽을 수 있습니다. 결과 인덱스에 대한 읽기 액세스 권한이 있는 사용자는 해당 데이터 소스에 대한 모든 쿼리의 결과를 읽을 수 있습니다.

OpenSearch Discover에서 S3 데이터 쿼리

테이블을 설정하고 원하는 선택적 쿼리 가속화를 구성한 후 데이터 분석을 시작할 수 있습니다. 데이터를 쿼리하려면 드롭다운 메뉴에서 데이터 소스를 선택합니다. Amazon S3 및 OpenSearch Dashboards를 사용하는 경우 검색으로 이동하여 데이터 소스 이름을 선택합니다.

건너뛰기 인덱스를 사용하거나 인덱스를 생성하지 않은 경우 SQL 또는 PPL을 사용하여 데이터를 쿼리할 수 있습니다. 구체화된 뷰 또는 커버링 인덱스를 구성한 경우에는 이미 인덱스가 있으므로 대시보드 전체에서 대시보드 쿼리 언어(DQL)를 사용할 수 있습니다. 관찰성 플러그인과 함께 PPL을 사용하고 쿼리 워크벤치 플러그인과 함께 SQL을 사용할 수도 있습니다. 현재는 관찰성 플러그인과 쿼리 워크벤치 플러그인만 PPL 및 SQL을 지원합니다. OpenSearch Service API를 사용하여 데이터를 쿼리하려면 비동기 API 설명서를 참조하세요.

> **Note**
>
> 일부 SQL 및 PPL 문, 명령 및 함수는 지원되지 않습니다. 지원되는 명령 목록은 섹션을 참조하세요the section called “지원되는 SQL 및 PPL 명령”.
> 구체화된 뷰 또는 커버링 인덱스를 생성한 경우 내에 인덱싱한 경우 DQL을 사용하여 데이터를 쿼리할 수 있습니다.

문제 해결

결과가 예상대로 반환되지 않는 경우가 있을 수 있습니다. 문제가 발생하는 경우를 따르고 있는지 확인합니다the section called “추천”.

OpenSearch Service에서 Amazon CloudWatch Logs 데이터 직접 쿼리

이 섹션에서는 Amazon OpenSearch Service에서 데이터 소스 통합을 생성하고 구성하는 프로세스를 안내하므로 CloudWatch Logs에 저장된 데이터를 효율적으로 쿼리하고 분석할 수 있습니다.

다음 페이지에서는 CloudWatch Logs 직접 쿼리 데이터 소스를 설정하고, 필요한 사전 조건을 탐색하고,를 사용하여 step-by-step 절차를 따르는 방법을 알아봅니다 AWS Management Console.

주제

- OpenSearch Service에서 Amazon CloudWatch Logs 데이터 소스 통합 생성
- OpenSearch Dashboards에서 CloudWatch Logs 데이터 소스 구성 및 쿼리

OpenSearch Service에서 Amazon CloudWatch Logs 데이터 소스 통합 생성

관찰성 요구 사항에 Amazon OpenSearch Serverless를 사용하는 경우 이제 OpenSearch Service에 데이터를 복사하거나 수집하지 않고도 Amazon CloudWatch Logs를 분석할 수 있습니다. 이 기능은

OpenSearch Service에서 Amazon S3의 데이터를 분석하는 것과 마찬가지로 직접 쿼리를 활용하여 데이터를 쿼리합니다. AWS Management Console에서 연결된 새 데이터 소스를 생성하여 시작할 수 있습니다.

CloudWatch Logs에서 운영 로그를 직접 쿼리하기 위해 Amazon OpenSearch Serverless를 빌드하지 않고도 CloudWatch Logs 데이터를 분석하는 새 데이터 소스를 생성할 수 있습니다. 이렇게 하면 OpenSearch Service 외부에 있는 액세스된 운영 데이터를 분석할 수 있습니다. OpenSearch Service 및 CloudWatch Logs에서 쿼리하면 CloudWatch Logs의 로그 분석을 시작한 다음 도구를 전환할 필요 없이 OpenSearch의 데이터 소스 모니터링으로 다시 이동할 수 있습니다.

이 기능을 사용하려면 AWS 관리 콘솔을 통해 OpenSearch Service에 대한 CloudWatch Logs 다이렉트 쿼리 데이터 소스를 생성합니다.

주제

- 사전 조건
- 절차
- 다음 단계
- 추가 리소스

사전 조건

시작하기 전에 다음 설명서를 검토했는지 확인합니다.

- the section called “Amazon CloudWatch Logs에 대한 제한 사항”
- the section called “CloudWatch Logs 권장 사항”
- the section called “CloudWatch Logs 할당량”

데이터 소스를 생성하려면 먼저에 AWS 계정다음 리소스가 있어야 합니다.

- CloudWatch Logs를 활성화합니다. OpenSearch 리소스 AWS 계정 와 동일한에서 로그를 수집하도록 CloudWatch Logs를 구성합니다. 지침은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs 시작하기를 참조하세요. Amazon CloudWatch
- 하나 이상의 CloudWatch 로그 그룹. 쿼리하려는 데이터가 포함된 로그 그룹을 지정할 수 있습니다. 로그 그룹 생성에 대한 지침은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs에서 로그 그룹 생성을 참조하세요. Amazon CloudWatch

- (선택 사항) 수동으로 생성한 IAM 역할입니다. 이 역할을 사용하여 데이터 소스에 대한 액세스를 관리할 수 있습니다. 또는 필요한 권한을 사용하여 OpenSearch Service가 자동으로 역할을 생성하도록 할 수 있습니다. 수동으로 생성한 IAM 역할을 사용하도록 선택한 경우의 지침을 따릅니다the section called “수동으로 생성한 IAM 역할에 필요한 권한”.

절차

를 사용하여 컬렉션 수준 쿼리 데이터 소스를 설정할 수 있습니다 AWS Management Console.

를 사용하여 컬렉션 수준 데이터 소스를 설정하려면 AWS Management Console

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 중앙 관리로 이동하여 연결된 데이터 소스를 선택합니다.
3. 연결을 선택합니다.
4. 데이터 소스 유형으로 CloudWatch를 선택합니다.
5. 다음을 선택합니다.
6. 데이터 연결 세부 정보에서 이름과 선택적 설명을 입력합니다.
7. IAM 역할에서 로그 그룹에 대한 액세스를 관리하는 방법을 선택합니다.

   a. 이 데이터 소스에 대한 역할을 자동으로 생성하려면 다음 단계를 따르세요.

      i. 새 역할 생성을 선택합니다.
      ii. IAM 역할의 이름을 입력합니다.
      iii. 하나 이상의 로그 그룹을 선택하여 쿼리할 데이터를 정의합니다.

   b. 직접 관리하는 기존 역할을 사용하려면 다음 단계를 따르세요.

      i. 기존 역할 사용을 선택합니다.
      ii. 드롭다운 메뉴에서 기존 역할을 선택합니다.

   > **Note**
   >
   > 자체 역할을 사용하는 경우 IAM 콘솔에서 필요한 정책을 연결하여 필요한 모든 권한이 있는지 확인해야 합니다. 자세한 내용은 the section called “수동으로 생성한 IAM 역할에 필요한 권한” 단원을 참조하십시오.

8. (선택 사항) 태그에서 데이터 소스에 태그를 추가합니다.

9. 다음을 선택합니다.
10. OpenSearch 설정에서 OpenSearch 설정 방법을 선택합니다.

    a. 기본 설정을 사용합니다.

    - 기본 리소스 이름 및 데이터 보존 설정을 검토합니다. 사용자 지정 이름을 사용하는 것이 좋습니다.

      기본 설정을 사용하면 추가 비용 없이 새 OpenSearch 애플리케이션과 Essentials 워크스페이스가 생성됩니다. OpenSearch를 사용하면 여러 데이터 소스를 분석할 수 있습니다. 여기에는 인기 있는 사용 사례에 맞는 맞춤형 경험을 제공하는 워크스페이스가 포함되어 있습니다. Workspace는 액세스 제어를 지원하므로 사용 사례에 맞는 프라이빗 공간을 생성하고 공동 작업자와만 공유할 수 있습니다.

    b. 사용자 지정 설정 사용:

      i. 사용자 지정을 선택합니다.
      ii. 필요에 따라 컬렉션 이름과 데이터 보존 설정을 편집합니다.
      iii. 사용할 OpenSearch 애플리케이션 및 워크스페이스를 선택합니다.
11. 다음을 선택합니다.
12. 선택 사항을 검토하고 변경해야 하는 경우 편집을 선택합니다.
13. 연결을 선택하여 데이터 소스를 설정합니다. 데이터 소스가 생성되는 동안이 페이지를 유지합니다. 준비가 되면 데이터 소스 세부 정보 페이지로 이동합니다.

다음 단계

OpenSearch 대시보드 방문

데이터 소스가 생성된 후 OpenSearch Service에서 OpenSearch 대시보드 URL을 제공합니다. 이를 사용하여 액세스 제어를 구성하고, 테이블을 정의하고, 널리 사용되는 로그 유형에 대한 로그 유형 기반 대시보드를 설정하고, SQL 또는 PPL을 사용하여 데이터를 쿼리할 수 있습니다.

자세한 내용은 the section called “CloudWatch Logs 데이터 소스 구성” 단원을 참조하십시오.

추가 리소스

수동으로 생성한 IAM 역할에 필요한 권한

데이터 소스를 생성할 때 데이터에 대한 액세스를 관리할 IAM 역할을 선택합니다. 여기에는 두 가지 옵션이 있습니다.

1. 새 IAM 역할 자동 생성
2. 수동으로 생성한 기존 IAM 역할 사용

수동으로 생성한 역할을 사용하는 경우 역할에 올바른 권한을 연결해야 합니다. 권한은 특정 데이터 소스에 대한 액세스를 허용하고 OpenSearch Service가 역할을 수임하도록 허용해야 합니다. 이는 OpenSearch Service가 데이터에 안전하게 액세스하고 상호 작용할 수 있도록 하기 위해 필요합니다.

다음 샘플 정책은 데이터 소스를 생성하고 관리하는 데 필요한 최소 권한을 보여줍니다. `logs:*` 또는 `AdminstratorAccess` 정책과 같이 더 광범위한 권한이 있는 경우 이러한 권한에는 샘플 정책의 최소 권한 권한이 포함된다는 점에 유의하시기 바랍니다.

다음 샘플 정책에서 *## ### ####* 자신의 정보로 바꿉니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AmazonOpenSearchDirectQueryAllLogsAccess",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:StartQuery",
                "logs:GetLogGroupFields"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "accountId"
                }
            },
            "Resource": [
                "arn:aws:logs:region:accountId:log-group:*"
            ]
        }
    ]
}

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

```
        {
            "Sid": "AmazonOpenSearchDirectQueryServerlessAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll",
                "aoss:DashboardsAccessAll"
            ],
            "Resource": [
                "arn:aws:aoss:region:accountId:collection/ARN/*",
                "arn:aws:aoss:region:accountId:collection/ARN"
            ]
        }
    ]
}
```

역할에는 대상 ID를 지정하는 다음과 같은 신뢰 정책도 있어야 합니다.

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::datasource/rolename"
                }
            }
        }
    ]
}
```

역할을 생성하기 위한 지침은 사용자 지정 신뢰 정책을 사용하여 역할 생성을 참조하세요.

기본적으로 역할에는 직접 쿼리 데이터 소스 인덱스에 대한 액세스만 있습니다. 데이터 소스에 대한 액세스 권한을 제한하거나 부여하도록 역할을 구성할 수 있지만 이 역할의 액세스 권한을 조정하지 않는 것이 좋습니다. 데이터 소스를 삭제하면 이 역할이 삭제됩니다. 조정할 경우 다른 사용자가 역할에 매핑된 경우 다른 사용자의 액세스 권한이 제거됩니다.

OpenSearch Dashboards에서 CloudWatch Logs 데이터 소스 구성 및 쿼리

이제 데이터 소스를 생성했으므로 OpenSearch Dashboards를 시작할 수 있습니다. 이 섹션에서는 OpenSearch Dashboards에서 데이터 소스의 다양한 사용 사례를 안내합니다.

검색 페이지에서 로그 그룹 쿼리

OpenSearch 검색 페이지에서 CloudWatch Logs 로그 그룹을 쿼리하도록 구성한 새 직접 쿼리 데이터 소스를 사용할 수 있습니다. 이렇게 하려면 로그 탐색을 선택한 다음 검색 창을 사용하여 SQL 또는 PPL을 사용하여 쿼리를 빌드합니다. 로그 그룹에서 반환된 데이터를 필터링, 정렬 및 시각화할 수 있습니다. CloudWatch Logs 통합에 지원되는 명령문, 명령 및 제한 사항을 이해하려면 섹션을 참조하세요the section called “지원되는 SQL 및 PPL 명령”.

데이터 소스에 대한 대시보드 보기 생성

OpenSearch Service를 사용하면 사전 구축된 대시보드 템플릿을 사용하여 널리 사용되는 AWS 로그 유형을 빠르게 분석할 수 있습니다. CloudWatch Logs에는 VPC, CloudTrail AWS WAF및 Network Firewall 로그에 대한 템플릿이 있습니다. 이러한 템플릿을 사용하면 특정 데이터에 맞는 대시보드를 빠르게 생성할 수 있습니다. 여기에는 해당 특정 로그 유형에 맞게 조정된 대시보드가 포함됩니다. 이렇게 하면 처음부터 모든 것을 빌드할 필요 없이 널리 사용되는 로그 AWS 소스를 빠르게 분석하여 실행할 수 있습니다.

> **Note**
>
> 대시보드는 직접 쿼리 OpenSearch 컴퓨팅 유닛(OCUs)과 서버리스 컬렉션 인덱싱OCU, searchingOCUs 및 스토리지를 사용하여 CloudWatch Logs에서 데이터를 수집하는 인덱싱된 뷰를 사용합니다. indexingOCUs

다음 단계에 따라 이러한 사전 구축된 템플릿 중 하나를 사용하여 대시보드를 생성하면 데이터를 즉시 탐색하고 분석할 수 있습니다.

대시보드 보기를 생성하려면

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.

2. 왼쪽 탐색 창에서 중앙 관리를 선택한 다음 연결된 데이터 소스를 선택합니다.
3. 데이터 소스를 선택하여 세부 정보 페이지를 엽니다.
4. 대시보드 생성(Create dashboard)을 선택합니다.
5. 생성할 대시보드 유형을 선택합니다.
6. 대시보드의 이름을 입력합니다.
7. 대시보드에 대한 선택적 설명을 입력합니다.
8. 대시보드에서 볼 로그 그룹을 하나 이상 선택합니다.
9. 대시보드에서 데이터를 새로 고칠 빈도를 선택합니다.
10. 사용할 OpenSearch 워크스페이스를 선택합니다.
    a. 새 워크스페이스를 생성하려면 새 워크스페이스 생성을 선택하고 이름을 입력합니다.
    b. 기존 워크스페이스를 사용하려면 기존 워크스페이스 선택을 선택합니다.
11. 대시보드 생성(Create dashboard)을 선택합니다.

OpenSearch Discover에서 CloudWatch Logs 데이터 쿼리

데이터를 쿼리하려면 드롭다운 메뉴에서 데이터 소스를 선택합니다. CloudWatch Logs를 사용하는 경우 Essentials 워크스페이스에서 검색으로 이동하여 OpenSearch SQL 또는 Piped Processing Language(PPL)를 사용하여 데이터 쿼리를 시작합니다. 지원되는 명령 목록은 섹션을 참조하세요the section called “지원되는 SQL 및 PPL 명령”.

> **Note**
>
> 구체화된 뷰를 생성한 경우 DQL을 사용하여 안에 인덱싱한 데이터를 쿼리할 수 있습니다.

문제 해결

결과가 예상대로 반환되지 않는 경우가 있을 수 있습니다. 문제가 발생하는 경우를 따르고 있는지 확인합니다the section called “추천”.

OpenSearch Service에서 Amazon Security Lake 데이터 직접 쿼리

이 섹션에서는 Amazon OpenSearch Service에서 데이터 소스 통합을 생성하고 구성하는 프로세스를 안내하므로 Security Lake에 저장된 데이터를 효율적으로 쿼리하고 분석할 수 있습니다.

다음 페이지에서는 Security Lake 직접 쿼리 데이터 소스를 설정하고, 필요한 사전 조건을 탐색하고,를 사용하여 step-by-step 절차를 따르는 방법을 알아봅니다 AWS Management Console.

주제

- OpenSearch Service에서 Amazon Security Lake 데이터 소스 통합 생성
- OpenSearch Dashboards에서 Security Lake 데이터 소스 구성 및 쿼리

OpenSearch Service에서 Amazon Security Lake 데이터 소스 통합 생성

Amazon OpenSearch Serverless를 사용하여 Amazon Security Lake에서 보안 데이터를 직접 쿼리할 수 있습니다. 이렇게 하려면 Security Lake 데이터에 OpenSearch 제로 ETL 기능을 사용할 수 있는 데이터 소스를 생성합니다. 데이터 소스를 생성할 때 Security Lake에 저장된 데이터를 직접 검색하고, 인사이트를 얻고, 분석할 수 있습니다. 온디맨드 인덱싱을 사용하여 일부 Security Lake 데이터 세트에서 쿼리 성능을 가속화하고 고급 OpenSearch 분석을 사용할 수 있습니다.

주제

- 사전 조건
- 절차
- 다음 단계
- 추가 리소스

사전 조건

시작하기 전에 다음 설명서를 검토했는지 확인합니다.

- the section called “Amazon Security Lake에 대한 제한 사항”
- the section called “Security Lake 권장 사항”
- the section called “Security Lake 할당량”

데이터 소스를 생성하기 전에 Security Lake에서 다음 작업을 수행합니다.

- Security Lake를 활성화합니다. OpenSearch 리소스 AWS 리전 와 동일한에서 로그를 수집하도록 Security Lake를 구성합니다. 지침은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 시작하기를 참조하세요.

- Security Lake 권한을 설정합니다. 리소스 관리에 대한 서비스 연결 역할 권한을 수락했으며 콘솔의 문제 페이지에 문제가 표시되지 않는지 확인합니다. 자세한 내용은 Amazon Security Lake 사용 설명서의 Security Lake에 대한 서비스 연결 역할을 참조하세요.
- Security Lake 데이터 소스를 공유합니다. Security Lake와 동일한 계정 내에서 OpenSearch에 액세스할 때 Security Lake 콘솔에서 Security Lake 버킷을 Lake Formation에 등록하라는 메시지가 없는지 확인합니다. 교차 계정 OpenSearch 액세스를 위해 Security Lake 콘솔에서 Lake Formation 쿼리 구독자를 설정합니다. OpenSearch 리소스와 연결된 계정을 구독자로 사용합니다. 자세한 내용은 Amazon Security Lake 사용 설명서의 Security Lake의 구독자 관리를 참조하세요.

또한에 AWS 계정다음 리소스도 있어야 합니다.

- (선택 사항) 수동으로 생성한 IAM 역할입니다. 이 역할을 사용하여 데이터 소스에 대한 액세스를 관리할 수 있습니다. 또는 필요한 권한을 사용하여 OpenSearch Service에서 자동으로 역할을 생성하도록 할 수 있습니다. 수동으로 생성한 IAM 역할을 사용하도록 선택한 경우의 지침을 따릅니다the section called “수동으로 생성된 IAM 역할에 필요한 권한”.

절차

내에서 Security Lake 데이터베이스에 연결하도록 데이터 소스를 설정할 수 있습니다 AWS Management Console.

를 사용하여 데이터 소스를 설정하려면 AWS Management Console

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 중앙 관리로 이동하여 연결된 데이터 소스를 선택합니다.
3. 연결을 선택합니다.
4. 데이터 소스 유형으로 Security Lake를 선택합니다.
5. 다음을 선택합니다.
6. 데이터 연결 세부 정보에서 이름과 선택적 설명을 입력합니다.
7. IAM 권한 액세스 설정에서 데이터 소스에 대한 액세스를 관리하는 방법을 선택합니다.

    a. 이 데이터 소스에 대한 역할을 자동으로 생성하려면 다음 단계를 따르세요.

        i. 새 역할 생성을 선택합니다.
        ii. IAM 역할의 이름을 입력합니다.
        iii. 하나 이상의 AWS Glue 테이블을 선택하여 쿼리할 데이터를 정의합니다.

b. 직접 관리하는 기존 역할을 사용하려면 다음 단계를 따르세요.

i. 기존 역할 사용을 선택합니다.

ii. 드롭다운 메뉴에서 기존 역할을 선택합니다.

> **Note**
>
> 자체 역할을 사용할 때는 IAM 콘솔에서 필요한 정책을 연결하여 필요한 모든 권한이 있는지 확인해야 합니다. 자세한 내용은 the section called “수동으로 생성된 IAM 역할에 필요한 권한” 단원을 참조하십시오.

8. (선택 사항) 태그에서 데이터 소스에 태그를 추가합니다.
9. 다음을 선택합니다.
10. OpenSearch 설정에서 OpenSearch 설정 방법을 선택합니다.

    - 기본 리소스 이름 및 데이터 보존 설정을 검토합니다.

      기본 설정을 사용하면 추가 비용 없이 새 OpenSearch 애플리케이션과 Essentials 워크스페이스가 생성됩니다. OpenSearch를 사용하면 여러 데이터 소스를 분석할 수 있습니다. 여기에는 인기 있는 사용 사례에 맞는 맞춤형 경험을 제공하는 워크스페이스가 포함되어 있습니다. Workspace는 액세스 제어를 지원하므로 사용 사례에 맞는 프라이빗 공간을 생성하고 공동 작업자와만 공유할 수 있습니다.

11. 사용자 지정 설정 사용:

    a. 사용자 지정을 선택합니다.

    b. 필요에 따라 컬렉션 이름과 데이터 보존 설정을 편집합니다.

    c. 사용할 OpenSearch 애플리케이션 및 워크스페이스를 선택합니다.

12. 다음을 선택합니다.
13. 선택 사항을 검토하고 변경해야 하는 경우 편집을 선택합니다.
14. 연결을 선택하여 데이터 소스를 설정합니다. 데이터 소스가 생성되는 동안이 페이지를 유지합니다. 준비가 되면 데이터 소스 세부 정보 페이지로 이동합니다.

다음 단계

OpenSearch Dashboards를 방문하여 대시보드 생성

데이터 소스가 생성된 후 OpenSearch Service에서 OpenSearch 대시보드 URL을 제공합니다. 이를 사용하여 SQL 또는 PPL을 사용하여 데이터를 쿼리합니다. Security Lake 통합에는 로그 분석을 시작할 수 있도록 SQL 및 PPL용 사전 패키징된 쿼리 템플릿이 함께 제공됩니다.

자세한 내용은 the section called “Security Lake 데이터 소스 구성” 단원을 참조하십시오.

추가 리소스

수동으로 생성된 IAM 역할에 필요한 권한

데이터 소스를 생성할 때 데이터에 대한 액세스를 관리할 IAM 역할을 선택합니다. 여기에는 두 가지 옵션이 있습니다.

1. 새 IAM 역할 자동 생성
2. 수동으로 생성한 기존 IAM 역할 사용

수동으로 생성한 역할을 사용하는 경우 역할에 올바른 권한을 연결해야 합니다. 권한은 특정 데이터 소스에 대한 액세스를 허용하고 OpenSearch Service가 역할을 수임하도록 허용해야 합니다. 이는 OpenSearch Service가 데이터에 안전하게 액세스하고 상호 작용할 수 있도록 하기 위해 필요합니다.

다음 샘플 정책은 데이터 소스를 생성하고 관리하는 데 필요한 최소 권한을 보여줍니다. `AdminstratorAccess` 정책과 같은 더 광범위한 권한이 있는 경우 이러한 권한은 샘플 정책의 최소 권한 권한을 포함합니다.

다음 샘플 정책에서 `## ### ####` 자신의 정보로 바꿉니다.

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": [
                "*"
            ]
        }
    ]
}
```

역할에는 대상 ID를 지정하는 다음과 같은 신뢰 정책도 있어야 합니다.

JSON

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

역할을 생성하기 위한 지침은 사용자 지정 신뢰 정책을 사용하여 역할 생성을 참조하세요.

기본적으로 역할에는 직접 쿼리 데이터 소스 인덱스에 대한 액세스만 있습니다. 데이터 소스에 대한 액세스 권한을 제한하거나 부여하도록 역할을 구성할 수 있지만 이 역할의 액세스 권한을 조정하지 않는 것이 좋습니다. 데이터 소스를 삭제하면 이 역할이 삭제됩니다. 조정할 경우 다른 사용자가 역할에 매핑된 경우 다른 사용자의 액세스 권한이 제거됩니다.

고객 관리형 키로 암호화된 Security Lake 데이터 쿼리

데이터 연결과 연결된 Security Lake 버킷이 고객 관리형를 사용한 서버 측 암호화를 사용하여 암호화된 경우 키 정책에 LakeFormation 서비스 역할을 추가 AWS KMS key해야 합니다. 이렇게 하면 서비스가 쿼리에 대한 데이터에 액세스하고 읽을 수 있습니다.

다음 샘플 정책에서 *## ### ####* 자신의 정보로 바꿉니다.

```
{
    "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": "*"
}
```

OpenSearch Dashboards에서 Security Lake 데이터 소스 구성 및 쿼리

이제 데이터 소스를 생성했으므로 OpenSearch Dashboards에서 설정할 수 있습니다.

실제로 데이터를 쿼리하기 전에, 이 섹션에서 OpenSearch 대시보드의 다양한 데이터 소스 사용 사례를 살펴보겠습니다. 시작하려면 OpenSearch Dashboards에서 데이터 소스로 이동해야 합니다. 왼쪽 메뉴의 관리에서 데이터 소스를 선택합니다. 그런 다음 OpenSearch Service 콘솔에서 이전에 생성한 데이터 소스의 이름을 선택합니다.

Discover에서 Security Lake 테이블 쿼리

Security Lake 로그를 기반으로 테이블을 생성한 경우 이제 OpenSearch Discover에서 해당 테이블을 직접 쿼리할 수 있습니다. 이를 통해 익숙한 Discover 인터페이스에서 직접 Security Lake에 저장된 데이터에 원활하게 액세스하고 분석할 수 있습니다. Discover에서 직접 Security Lake를 쿼리하면 데이터를 수동으로 추출, 변환 및 별도의 검색 인덱스로 로드할 필요가 없습니다. 로그 분석을 빠르게 시작하기 위해 Discover에는 PPL 및 SQL 저장 쿼리 세트가 포함되어 있습니다.

먼저 구성한 데이터 소스를 선택합니다. 쿼리하려는 연결된 데이터베이스와 테이블을 선택한 다음 검색 창을 사용하여 테이블에 대한 쿼리를 작성합니다. Security Lake 통합에 지원되는 문, 명령 및 제한 사항을 이해하려면 섹션을 참조하세요the section called “지원되는 SQL 및 PPL 명령”.

Security Lake에서 사용할 수 있는 사전 구축된 쿼리를 활용하려면 검색 오른쪽 상단의 ...로 이동하여 쿼리 열기를 선택한 다음 템플릿을 선택합니다. Security Lake에서 지원되는 로그 소스에 사용할 수 있는 사전 구축된 쿼리가 많습니다. 사용 사례와 일치하는 템플릿을 검색하고, 검색 창에 사용할 쿼리를 복사하고, 템플릿 필드(예: 리전 및 작업)를 자체 정보로 바꿉니다.

Discover에서 데이터 가속화

OpenSearch에서 성능을 개선하고 후속 쿼리 및 분석을 더 빠르게 활성화하기 위해 Discover의 쿼리 결과를 OpenSearch 인덱싱 보기로 수집할 수 있습니다.

인덱싱된 뷰를 생성하려면

1. 검색에서 인덱싱된 보기 생성을 선택합니다.

2. 쿼리 편집기에서 원하는 쿼리를 입력합니다. 여기에서 새 쿼리를 생성하거나 이전 검색에서 기존 쿼리를 사용할 수 있습니다.
3. 새 인덱싱된 뷰의 이름을 지정합니다. 나중에 뷰를 식별하는 데 도움이 되는 설명이 포함된 이름을 선택합니다.
4. 인덱싱된 뷰에 대한 데이터 보존 설정을 구성합니다. 데이터를 인덱스에 보관하는 기간을 지정하여 성능과 스토리지 비용의 균형을 맞출 수 있습니다.
5. 인덱싱된 뷰를 생성합니다. 생성된 후에는 더 빠른 쿼리 및 분석을 위해 인덱싱된 뷰를 사용할 수 있습니다.

이전에 인덱싱된 뷰를 생성한 경우 검색에서 액세스할 수 있습니다.

기존 인덱스 보기를 사용하려면

1. 검색에서 인덱싱된 보기 선택을 선택하여 Security Lake에 대한 기존 인덱싱된 보기 목록을 확인합니다.
2. 사용할 인덱싱된 뷰를 선택합니다. 이렇게 하면 현재 쿼리에 보기가 적용되므로 데이터 검색 및 분석 속도가 크게 빨라질 수 있습니다.

데이터 소스에 대한 대시보드 보기 생성

OpenSearch Service를 사용하는 경우 사전 구축된 대시보드 템플릿을 사용하여 널리 사용되는 AWS 로그 유형을 분석할 수 있습니다. Security Lake에는 VPC, CloudTrail 및 WAF 로그에 대한 템플릿이 있습니다. 이러한 템플릿을 사용하면 특정 데이터에 맞는 대시보드를 생성할 수 있습니다. 여기에는 해당 특정 로그 유형에 맞게 조정된 사전 구축된 쿼리 및 대시보드가 포함됩니다. 이렇게 하면 처음부터 모든 것을 빌드할 필요 없이 이러한 인기 있는 AWS 로그 소스를 빠르게 분석하여 실행할 수 있습니다.

> **Note**
>
> 대시보드는 Security Lake에서 데이터를 수집하고 직접 쿼리 및 수집 컴퓨팅에 기여하는 인덱싱된 뷰를 사용합니다.

다음 단계에 따라 이러한 사전 구축된 템플릿 중 하나를 사용하여 대시보드를 생성하면 데이터를 즉시 탐색하고 분석할 수 있습니다.

대시보드 보기를 생성하려면

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 중앙 관리를 선택한 다음 연결된 데이터 소스를 선택합니다.
3. 데이터 소스를 선택하여 세부 정보 페이지를 엽니다.
4. 대시보드 생성(Create dashboard)을 선택합니다.
5. 생성할 대시보드 유형을 선택합니다.
6. 대시보드의 이름을 입력합니다.
7. 대시보드에 대한 선택적 설명을 입력합니다.
8. 대시보드에서 볼 AWS Glue 테이블을 하나 이상 선택합니다.
9. 대시보드에서 데이터를 새로 고칠 빈도를 선택합니다.
10. 사용할 OpenSearch 워크스페이스를 선택합니다.
    a. 새 워크스페이스를 생성하려면 새 워크스페이스 생성을 선택합니다.
    b. 기존 워크스페이스를 사용하려면 기존 워크스페이스 선택을 선택합니다.
11. 워크스페이스의 이름을 입력합니다.
12. 대시보드 생성(Create dashboard)을 선택합니다.

문제 해결

결과가 예상대로 반환되지 않는 경우가 있을 수 있습니다. 문제가 발생하는 경우를 따르고 있는지 확인합니다the section called “추천”.

Amazon OpenSearch Service에서 데이터 소스 관리

데이터 소스 관리는 직접 쿼리 데이터 소스 및 기타 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다.는 모니터링, 문제 발생 시 보고 및 적절한 경우 자동 조치를 취할 수 있는 다음 도구를 AWS 제공합니다.

주제

- CloudWatch 지표 데이터 소스를 사용한 모니터링
- 데이터 소스 활성화 및 비활성화
- AWS Budget을 사용한 모니터링
- 데이터 소스 삭제

CloudWatch 지표 데이터 소스를 사용한 모니터링

CloudWatch를 사용하여 직접 쿼리를 모니터링할 수 있습니다. CloudWatch는 원시 데이터를 수집하여 읽기 쉽고 실시간에 가까운 지표로 처리합니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다.

특정 임계치를 모니터링하다가 해당 임계치가 충족될 때 알림을 전송하거나 작업을 수행하도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch란 무엇인가요?를 참조하세요.

Amazon S3는 다음 지표를 보고합니다.

지표	설명
AsyncQueryCreateAPI	비동기 쿼리를 생성하기 위해 API에 대해 수행된 총 요청 수. 관련 통계: 평균, 최대, 합계 차원: `ClientId`, `DomainName` 빈도: 60초
AsyncQueryGetApiRequestCount	비동기 쿼리 결과를 검색하기 위해 API에 대해 수행된 총 요청 수. 관련 통계: 평균, 최대, 합계 차원: `ClientId`, `DomainName` 빈도: 60초
AsyncQueryCancelApiRequestCount	비동기 쿼리를 취소하기 위해 API에 대해 수행된 총 요청 수. 관련 통계: 평균, 최대, 합계 차원: `ClientId`, `DomainName` 빈도: 60초
AsyncQueryGetApiFailedRequestCusErrCount	고객 관련 오류(예: 잘못된 쿼리 ID)로 인해 비동기 쿼리 결과를 검색할 때 실패한 요청 수.

지표	설명
	관련 통계: 평균, 최대, 합계 차원: `ClientId`, `DomainName` 빈도: 60초
AsyncQueryCancelApiFailedRequestCusErrCount	고객 관련 오류(예: 잘못된 쿼리 ID)로 인해 비동기 쿼리 결과를 검색할 때 실패한 요청 수. 관련 통계: 평균, 최대, 합계 차원: `ClientId`, `DomainName` 빈도: 60초
AsyncQueryCancelApiFailedRequestSysErrCount	고객 관련 오류로 인해 비동기 쿼리를 생성할 때 실패한 요청 수. 관련 통계: 평균, 최대, 합계 차원: `ClientId`, `DomainName` 빈도: 60초
AsyncQueryGetApiFailedRequestSysErrCount	시스템 관련 오류로 인해 비동기 쿼리 결과를 검색할 때 실패한 요청 수. 관련 통계: 평균, 최대, 합계 차원: `ClientId`, `DomainName` 빈도: 60초

CloudWatch Logs 및 Security Lake는 다음 지표를 보고합니다.

지표	설명
DirectQueryRate	데이터 소스에 대한 요청 속도입니다.

지표	설명
	관련 통계: 합계, 최대값, 최소값, 평균 차원: `DataSourceName` 빈도: 60초
DirectQueryLatency	데이터 소스에서 쿼리를 실행할 때 관찰된 지연 시간입니다. 관련 통계: Average, P90, P99, Sum, Minimum, Maximum 차원: `DataSourceName` 빈도: 60초
FailedDirectQueries	데이터 소스 쿼리에서 관찰된 총 쿼리 실패 수입니다. 관련 통계: 합계, 최대값, 최소값, 평균 차원: `DataSourceName` 빈도: 60초
DirectQueryConsumedOCU	데이터 소스에서 쿼리를 실행하는 데 사용되는 OCUs 수입니다. 관련 통계: Average, P90, P99, Sum, Minimum, Maximum 차원: `DataSourceName` 빈도: 60초

데이터 소스 활성화 및 비활성화

> **Note**
>
> 다음 정보는 Amazon S3 데이터 소스에만 적용됩니다.

데이터 소스에 대한 직접 쿼리 사용을 중지하려는 경우 데이터 소스를 비활성화하도록 선택할 수 있습니다. 데이터 소스를 비활성화하면 기존 쿼리 실행이 완료되고 모든 새 쿼리가 실행되지 않습니다.

인덱스 건너뛰기, 구체화된 뷰, 커버링 인덱스와 같은 쿼리 성능을 높이기 위한 가속화 설정은 데이터 소스가 비활성화되면 수동으로 설정됩니다. 비활성화된 후 데이터 소스가 활성으로 설정되면 사용자 쿼리가 예상대로 실행됩니다. 이전에 설정하고 수동으로 설정한 가속화는 일정에 따라 다시 실행하도록 수동으로 구성해야 합니다.

AWS Budget을 사용한 모니터링

Amazon OpenSearch Service는 계정 수준에서 OCU 사용량 데이터를 Billing and Cost Management의 Cost Explorer에 채웁니다. 계정 수준에서 OCU 사용량을 고려하고 임계값을 초과하면 임계값과 알림을 설정할 수 있습니다.

Cost Explorer에서 필터링할 사용량 유형의 형식은 RegionCode -DirectQueryOCU(OCU-시간)와 같습니다. DirectQueryOCU(OCU 시간) 사용량이 임계값을 충족할 때 알림을 받으려면 Budgets 계정을 생성하고 AWS 설정한 임계값을 기반으로 알림을 구성할 수 있습니다. Amazon S3의 경우 선택적으로 Amazon SNS 주제를 설정할 수 있습니다. 그러면 임계값 기준이 충족될 경우 데이터 소스가 꺼집니다.

> **Note**
>
> AWS Budgets의 사용량 데이터는 실시간이 아니며 최대 8시간까지 지연될 수 있습니다.

데이터 소스 삭제

데이터 소스를 삭제하면 Amazon OpenSearch Service가 도메인 또는 컬렉션에서 데이터 소스를 제거합니다. 또한 OpenSearch Service는 데이터 소스와 관련된 인덱스를 제거합니다. 트랜잭션 데이터는 다른 트랜잭션 데이터에서 삭제되지 AWS 서비스않지만 다른 트랜잭션 데이터는 OpenSearch Service로 새 데이터를 보내지 AWS 서비스 않습니다.

AWS Management Console 또는 OpenSearch Service API를 사용하여 데이터 소스 통합을 삭제할 수 있습니다.

AWS Management Console

Amazon S3 데이터 소스를 삭제하려면

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 도메인을 선택합니다.

3. 데이터 소스를 삭제하려는 도메인을 선택합니다. 그러면 도메인 세부 정보 페이지가 열립니다. 일반 정보 아래에서 연결 탭을 선택하고 직접 쿼리 섹션을 찾습니다.
4. 삭제하려는 데이터 소스를 선택한 다음 삭제를 선택하고 삭제를 확인합니다.

CloudWatch Logs 또는 Security Lake 데이터 소스를 삭제하려면

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 중앙 관리를 선택한 다음 연결된 데이터 소스를 선택합니다.
3. 삭제하려는 데이터 소스를 선택한 다음 삭제를 선택하고 삭제를 확인합니다.

OpenSearch Service API

Amazon S3 데이터 소스를 삭제하려면 DeleteDataSource API 작업을 사용합니다.

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource/data-source-name
```

CloudWatch Logs 또는 Security Lake 데이터 소스를 삭제하려면 DeleteDirectQueryDataSource API 작업을 사용합니다.

Amazon OpenSearch Service 데이터 소스의 쿼리 성능 최적화

외부 데이터 소스에 액세스할 때 Amazon OpenSearch Service의 쿼리 성능이 저하될 수 있습니다. 이는 네트워크 지연 시간, 데이터 변환 또는 대용량 데이터 볼륨과 같은 요인 때문일 수 있습니다. 성능을 개선하려면 사용 사례에 따라 선택한 양의 데이터를 인덱싱하는 것이 좋습니다.

- Amazon S3에서 직접 쿼리 속도 향상(인덱스 건너뛰기)
- 및 Security Lake에서 대시보드 시각화 구축(자재화된 뷰)
- 오프라인 검토를 위해 인덱싱된 뷰를 사용하여 쿼리 결과 수집 또는 Security Lake에서 성능 향상(자재화된 뷰)

예제 쿼리를 포함하여 가속화된 쿼리에 대한 전체 설명서는 오픈 소스 설명서의 Optimize query performance using OpenSearch indexing을 참조하세요.

주제

- 건너뛰기 인덱스

- 구체화된 뷰
- 커버링 인덱스

건너뛰기 인덱스

건너뛰기 인덱스는 Amazon S3에 저장된 데이터의 메타데이터만 수집합니다. 건너뛰는 인덱스로 테이블을 쿼리하면 쿼리 플래너는 인덱스를 사용하여 쿼리를 다시 작성하므로 모든 파티션과 파일을 스캔하지 않고도 데이터의 위치를 효율적으로 식별할 수 있습니다. 이 접근 방식은 저장된 데이터의 정확한 위치를 좁히는 데 도움이 됩니다.

건너뛰기 인덱스를 생성하는 방법에는 두 가지가 있습니다. 첫 번째 방법은 데이터 소스 세부 정보 내에서 건너뛰기 인덱스를 자동으로 생성하는 것입니다. 두 번째는 Query Workbench를 사용하여 SQL 문을 사용하여 건너뛰기 인덱스를 수동으로 생성하는 것입니다.

데이터 소스에서 건너뛰기 인덱스를 자동 생성하려면 대시보드 관리 및 데이터 가속화로 이동한 다음 데이터베이스와 테이블을 선택합니다(최신 데이터베이스와 테이블을 가져오려면 새로 고쳐야 할 수 있음). 그런 다음 생성을 선택하여 건너뛰기 인덱스를 자동 생성하거나 인덱싱하려는 각 필드를 수동으로 선택하고 가속(인덱스 유형 건너뛰기)을 지정할 수 있습니다. 마지막으로 가속 생성을 선택하여 새 건너뛰기 인덱스를 채우는 반복 작업을 생성합니다.

인덱스 건너뛰기는 Amazon S3 데이터 소스에서만 지원됩니다.

Query Workbench를 사용하여 인덱스 건너뛰기를 설정하는 방법에 대한 자세한 내용은 OpenSearch 설명서의 인덱스 건너뛰기를 참조하세요.

구체화된 뷰

구체화된 뷰는 집계와 같은 복잡한 쿼리를 사용하여 OpenSearch Dashboards 시각화를 지원합니다. 쿼리를 기반으로 데이터의 하위 집합을 수집하여 OpenSearch 인덱스에 저장합니다. 그런 다음이 인덱스를 사용하여 시각화를 생성할 수 있습니다.

구체화된 뷰는 Amazon S3 및 Security Lake 데이터 소스에 대해 지원됩니다.

Query Workbench를 사용하여 구체화된 뷰를 설정하는 방법에 대한 자세한 내용은 OpenSearch 설명서의 구체화된 뷰를 참조하세요.

커버링 인덱스

커버링 인덱스는 테이블의 지정된 열에서 데이터를 수집하며 OpenSearch는이 데이터를 기반으로 새 인덱스를 생성합니다. 이 새 인덱스를 시각화 및 이상 탐지 또는 지리 공간 분석과 같은 기타 OpenSearch 기능에 사용할 수 있습니다.

커버링 인덱스는 Amazon S3 데이터 소스에서만 지원됩니다.

커버링 인덱스 설정에 대한 자세한 내용은 OpenSearch 설명서의 커버링 인덱스를 참조하세요.

지원되는 SQL 및 PPL 명령

OpenSearch SQL 및 OpenSearch 파이프라인 처리 언어(PPL)는 OpenSearch, CloudWatch Logs Insights 및 Security Lake에서 데이터를 쿼리, 분석 및 처리하기 위한 언어입니다. OpenSearch Discover에서 OpenSearch SQL 및 OpenSearch PPL을 사용하여 CloudWatch Logs, Amazon S3 또는 Security Lake 내에서 데이터를 쿼리할 수 있습니다. 또한 CloudWatch Logs Insights는 CloudWatch Logs 분석을 위해 특별히 구축된 쿼리 언어인 Logs Insights QL 외에도 OpenSearch PPL 및 OpenSearch SQL 쿼리 언어를 모두 지원합니다.

- OpenSearch SQL: OpenSearch SQL은 관계형 데이터베이스 작업에 익숙한 옵션을 제공합니다. OpenSearch SQL은 SQL 기능의 하위 집합을 제공하므로 임시 쿼리 및 데이터 분석 작업을 수행하는 데 적합합니다. OpenSearch SQL을 사용하면 SELECT, FROM, WHERE, GROUP BY, HAVING 등의 명령과 SQL에서 사용할 수 있는 기타 다양한 SQL 명령 및 함수를 사용할 수 있습니다. 테이블(또는 로그 그룹) 간에 JOINs 실행하고, 하위 쿼리를 사용하여 테이블(또는 로그 그룹) 간에 데이터를 상호 연관시키고, 풍부한 JSON, 수학, 문자열, 조건부 및 기타 SQL 함수 세트를 사용하여 로그 및 보안 데이터에 대한 강력한 분석을 수행할 수 있습니다.
- OpenSearch PPL(Piped Processing Language): OpenSearch PPL을 사용하면 파이프 조합 명령을 사용하여 데이터를 검색, 쿼리 및 분석할 수 있으므로 복잡한 쿼리를 더 쉽게 이해하고 구성할 수 있습니다. 구문은 Unix 파이프를 기반으로 하며 데이터를 변환하고 처리하기 위한 명령 체인을 활성화합니다. PPL을 사용하면 데이터를 필터링하고 집계할 수 있으며, 분석을 위해 JOINs, 하위 쿼리, LookUP 및 풍부한 수학, 문자열, 날짜, 조건부 및 기타 함수 집합과 같은 명령을 사용할 수 있습니다.

OpenSearch PPL 및 OpenSearch SQL 쿼리 언어의 대부분의 명령은 CloudWatch Logs 및 OpenSearch에서 공통적이지만 이러한 각 서비스에서 지원되는 명령 및 함수 집합에는 몇 가지 차이점이 있습니다. 자세한 내용은 다음 페이지의 표를 참조하세요.

- the section called “지원되는 SQL 명령”

- the section called “CloudWatch Logs Insights”
- the section called “일반 SQL 제한 사항”
- the section called “지원되는 PPL 명령”
- the section called “OpenSearch PPL을 사용하는 CloudWatch Logs Insights 사용자를 위한 추가 정보”

지원되는 OpenSearch SQL 명령 및 함수

다음 참조 표에는 Amazon S3, Security Lake 또는 CloudWatch Logs에서 데이터 쿼리를 위해 OpenSearch Discover에서 지원되는 SQL 명령과 CloudWatch Logs Insights에서 지원되는 SQL 명령이 나와 있습니다. CloudWatch Logs Insights에서 지원되는 SQL 구문과 CloudWatch Logs 쿼리를 위해 OpenSearch Discover에서 지원되는 SQL 구문은 동일하며 다음 표에서 CloudWatch Logs로 참조됩니다.

> **Note**
>
> 또한 OpenSearch는 OpenSearch에서 수집되어 인덱스에 저장된 데이터를 쿼리하기 위한 SQL 지원을 제공합니다. 이 SQL 언어는 직접 쿼리에 사용되는 SQL과 다르며 인덱스에서는 OpenSearch SQL이라고 합니다.

주제

- 명령
- 함수
- 일반 SQL 제한 사항
- OpenSearch SQL을 사용하는 CloudWatch Logs Insights 사용자를 위한 추가 정보

명령

> **Note**
>
> 예제 명령 열에서 쿼리하려는 데이터 소스에 따라 *<tableName/logGroup>* 필요에 따라를 바꿉니다.
>
> - 명령 예: `SELECT Body , Operation FROM` **`<tableName/logGroup>`**

- Amazon S3 또는 Security Lake를 쿼리하는 경우 다음을 사용합니다. `SELECT Body , Operation FROM table_name`
- CloudWatch Logs를 쿼리하는 경우 다음을 사용합니다. `SELECT Body , Operation FROM `LogGroupA``

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “SELECT 절”	프로젝션된 값을 표시합니다.	지원됨	지원됨	지원됨	`SELECT method, status FROM <tableName/logGroup>`
the section called “WHERE 절”	제공된 필드 기준에 따라 로그 이벤트를 필터링합니다.	지원됨	지원됨	지원됨	`SELECT * FROM <tableName/logGroup> WHERE status = 100`
the section called “GROUP BY 절”	범주를 기반으로 로그 이벤트를 그룹화하고 통계를 기반으로 평균을	지원됨	지원됨	지원됨	`SELECT method, status, COUNT(*) AS request_count, SUM(bytes) AS total_bytes FROM <tableName/logGroup> GROUP BY method, status`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
	찾습니다.				
the section called “HAVING 절”	그룹화 조건을 기준으로 결과를 필터링합니다.	지원됨	지원됨	지원됨	`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 절”	순서 절의 필드를 기반으로 결과를 정렬합니다. 내림차순 또는 오름차순으로 정렬할 수 있습니다.	지원됨	지원됨	지원됨	`SELECT * FROM <tableName/logGroup> ORDER BY status DESC`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “JOIN 절” (INNER | CROSS | LEFT OUTER)	공통 필드를 기반으로 두 테이블의 결과를 조인합니다.	지원됨(조인에 Inner 및 Left Outer 키워드를 사용해야 함, SELECT 문에서는 하나의 JOIN 작업만 지원됨)	지원됨(조인에 내부, 왼쪽 외부 및 교차 키워드를 사용해야 함)	지원됨(조인에 내부, 왼쪽 외부 및 교차 키워드를 사용해야 함)	`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 절”	결과를 첫 번째 N개 행으로 제한합니다.	지원됨	지원됨	지원됨	`SELECT * FROM <tableName/logGroup> LIMIT 10`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:

the section called “CASE 절”	조건을 평가하고 첫 번째 조건이 충족되면 값을 반환합니다.	지원됨	지원됨	지원됨	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>*

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:

the section called “공통 테이블 표현식”	SELECT INSERT, UPDATE DELETE 또는 MERGE 문 내에 명명된 임시 결과 세트를 생성합니다.	지원되지 않음	지원됨	지원됨	`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”	SQL 문의 실행 계획을 실제로 실행하지 않고 표시합니다.	지원되지 않음	지원됨	지원됨	`EXPLAIN`
					`SELECT`
					`  k,`
					`  SUM(v)`
					`FROM`
					`  VALUES`
					`  (1, 2),`
					`  (1, 3) AS t(k, v)`
					`GROUP BY`
					`  k`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “LATERAL SUBQUERY 절”	FROM 절의 하위 쿼리가 동일한 FROM 절의 이전 항목에서 열을 참조하도록 허용합니다.	지원되지 않음	지원됨	지원됨	`SELECT * FROM tableName LATERAL ( SELECT * FROM t2 WHERE t1.c1 = t2.c1 )`
the section called “LATERAL VIEW 절”	기본 테이블의 각 행에 테이블 생성 함수를 적용하여 가상 테이블을 생성합니다.	지원되지 않음	지원됨	지원됨	`SELECT * FROM tableName LATERAL VIEW EXPLODE(ARRAY(30, 60)) tableName AS c_age LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “LIKE 조건자”	와일드카드 문자를 사용하여 문자열을 패턴과 일치시킵니다.	지원됨	지원됨	지원됨	`SELECT method, status, request, host FROM <tableName/logGroup> WHERE method LIKE 'D%'`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “OFFSET”	쿼리에서 행을 반환하기 전에 건너뛸 행 수를 지정합니다.	쿼리에서 LIMIT 절과 함께 사용할 때 지원됩니다. 예: • 지원되는 항목: `SELECT * FROM Table LIMIT 100 OFFSET 10` • 지원되지 않음: `SELECT * FROM Table OFFSET 10`	지원됨	지원됨	`SELECT method, status, bytes, datetime FROM <tableName/logGroup> ORDER BY datetime OFFSET 10`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “PIVOT 절”	행을 열로 변환하여 행 기반 형식의 데이터를 열 기반 형식으로 바꿉니다.	지원되지 않음	지원됨	지원됨	`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 “집합 연산자”	둘 이상의 SELECT 문(예: UNION, INTERSECT, EXCEPT)의 결과를 결합합니다.	지원됨	지원됨	지원됨	`SELECT method, status, bytes FROM <tableName/logGroup> WHERE status = '416' UNION SELECT method, status, bytes FROM <tableName/logGroup> WHERE bytes > 20000`

Command	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “SORT BY 절”	쿼리 결과를 반환할 순서를 지정합니다.	지원됨	지원됨	지원됨	`SELECT method, status, bytes FROM <tableName/logGroup> SORT BY bytes DESC`
the section called “UNPIVOT”	열을 행으로 변환하여 데이터를 열 기반 형식에서 행 기반 형식으로 바꿉니다.	지원되지 않음	지원됨	지원됨	`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`

함수

> **Note**
>
> 예제 명령 열에서 쿼리하려는 데이터 소스에 따라 *<tableName/logGroup>* 필요에 따라를 바꿉니다.

- 명령 예: `SELECT Body , Operation FROM <tableName/logGroup>`
- Amazon S3 또는 Security Lake를 쿼리하는 경우 다음을 사용합니다. `SELECT Body , Operation FROM table_name`
- CloudWatch Logs를 쿼리하는 경우 다음을 사용합니다. `SELECT Body , Operation FROM `LogGroupA``

사용 가능한 SQL 문법	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “문자열 함수”	SQL 쿼리 내에서 문자열 및 텍스트 데이터를 조작하고 변환할 수 있는 내장 함수입니다. 예를 들어 변환 사례, 문자열 결합, 부분 추출, 텍스트 정리 등이 있습니다.	지원됨	지원됨	지원됨	`SELECT UPPER(method) AS upper_method, LOWER(host) AS lower_hos t FROM <tableName/logGroup>`
the section called “날짜 및 시간 함수”	쿼리에서 날짜 및 타임스탬프 데이터를 처리하고 변환하기 위한 내장 함수입니다. 예:	지원됨	지원됨	지원됨	`SELECT TO_TIMESTAMP(datetime) AS timestamp, TIMESTAMP_SECONDS( UNIX_TIMESTAMP(datetime)) AS from_seconds, UNIX_TIMESTAMP(datetime) AS to_unix,`

사용 가능한 SQL 문법	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
	date_add, date_format, datediff, current_date.				`FROM_UTC_TIMESTAMP (datetime, 'PST') AS to_pst, TO_UTC_TIMESTAMP(datetime, 'EST') AS from_est FROM <tableName/logGroup>`
the section called “집계 함수”	여러 행에서 계산을 수행하여 단일 요약 값을 생성하는 내장 함수입니다. 예를 들어 합계, 개수, 평균, 최대 및 최소입니다.	지원됨	지원됨	지원됨	`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 “조건 함수”	지정된 조건을 기반으로 작업을 수행하거나 조건부로 표현식을 평가하는 내장 함수입니다. 예: CASE 및 IF.	지원됨	지원됨	지원됨	`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>`

사용 가능한 SQL 문법	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:

the section called “JSON 함수”	데이터 세트에서 JSON 구조를 조작할 수 있도록 SQL 쿼리(예: from_json, to_json, get_json_object, json_tuple) 내에서 JSON 형식의 데이터를 구문 분석, 추출, 수정 및 쿼리하는 기본 제공 함수입니다.	지원됨	지원됨	지원됨	<pre>SELECT
					    FROM_JSON(
					        @message,
					        'STRUCT<
					            host: STRING,
					            user-identifier: STRING,
					            datetime: STRING,
					            method: STRING,
					            status: INT,
					            bytes: INT
					        >'
					    ) AS parsed_json
					FROM
					    <tableName/logGroup></pre>

사용 가능한 SQL 문법	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “배열 함수”	SQL 쿼리에서 배열 유형 열로 작업하기 위한 내장 함수를 사용하면 배열 데이터 액세스, 수정 및 분석과 같은 작업(예: 크기, 폭발, array_contains)을 수행할 수 있습니다.	지원됨	지원됨	지원됨	`SELECT scores, size(scores) AS length, array_contains(scores, 90) AS has_90 FROM <tableName/logGroup>`
the section called “윈도 함수”	현재 행(창)과 관련된 지정된 행 집합에서 계산을 수행하여 순위, 합계 실행 및 이동 평균(예: ROW_NUMBER, RANK, LAG, LEAD)과 같은 작업을 활성화하는 내장 함수	지원됨	지원됨	지원됨	`SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) AS field2Rank FROM <tableName/logGroup>`

사용 가능한 SQL 문법	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “변환 함수”	SQL 쿼리 내에서 데이터를 한 유형에서 다른 유형으로 변환하여 데이터 유형 변환 및 형식 변환을 지원하는 내장 함수(예: CAST, TO_DATE, TO_TIMESTAMP, BINARY)	지원됨	지원됨	지원됨	`SELECT CAST('123' AS INT) AS converted_number, CAST(123 AS STRING) AS converted_string FROM <tableName/logGroup>`
the section called “조건자 함수”	조건을 평가하고 지정된 기준 또는 패턴(예: IN, LIKE, BETWEEN, IS NULL, EXISTS)을 기반으로 부울 값(true/false)을 반환하는 내장 함수	지원됨	지원됨	지원됨	`SELECT * FROM <tableName/logGroup> WHERE id BETWEEN 50000 AND 75000`

사용 가능한 SQL 문법	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “맵 함수”	컬렉션의 각 요소에 지정된 함수를 적용하여 데이터를 새 값 집합으로 변환합니다.	지원되지 않음	지원됨	지원됨	`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`
the section called “수학 함수”	평균, 합계 또는 삼각형 값 계산과 같은 숫자 데이터에 대한 수학 작업을 수행합니다.	지원됨	지원됨	지원됨	`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>`

사용 가능한 SQL 문법	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “다중 로그 그룹 함수”	사용자가 SQL SELECT 문에서 여러 로그 그룹을 지정할 수 있습니다.	지원됨	해당 사항 없음	해당 사항 없음	`SELECT lg1.Column1, lg1.Column2 FROM `logGroups(logGroupIdentifier: ['LogGroup1', 'LogGroup2'])` AS lg1 WHERE lg1.Column3 = "Success"`
the section called “생성기 함수”	값 시퀀스를 생성하는 반복자 객체를 생성하여 대규모 데이터 세트에서 효율적인 메모리 사용을 허용합니다.	지원되지 않음	지원됨	지원됨	`SELECT explode(array(10, 20))`

일반 SQL 제한 사항

CloudWatch Logs, Amazon S3 및 Security Lake와 함께 OpenSearch SQL을 사용할 때는 다음 제한이 적용됩니다.

1. SELECT 문에는 하나의 JOIN 작업만 사용할 수 있습니다.
2. 중첩 하위 쿼리는 한 수준만 지원됩니다.
3. 세미콜론으로 구분된 여러 문 쿼리는 지원되지 않습니다.
4. 동일하지만 경우에 따라서만 다른 필드 이름을 포함하는 쿼리(예: field1 및 FIELD1)는 지원되지 않습니다.

예를 들어 다음 쿼리는 지원되지 않습니다.

```
Select AWSAccountId, awsaccountid from LogGroup
```

그러나 필드 이름(@logStream)이 두 로그 그룹에서 동일하기 때문에 다음 쿼리는 입니다.

```
Select a.`@logStream`, b.`@logStream` from Table A INNER Join Table B on a.id = b.id
```

5. 함수와 표현식은 필드 이름에서 작동해야 하며 FROM 절에 지정된 로그 그룹이 있는 SELECT 문의 일부여야 합니다.

   예를 들어이 쿼리는 지원되지 않습니다.

   ```
   SELECT cos(10) FROM LogGroup
   ```

   이 쿼리는 다음과 같이 지원됩니다.

   ```
   SELECT cos(field1) FROM LogGroup
   ```

OpenSearch SQL을 사용하는 CloudWatch Logs Insights 사용자를 위한 추가 정보

CloudWatch Logs는 Logs Insights 콘솔, API 및 CLI에서 OpenSearch SQL 쿼리를 지원합니다. SELECT, FROM, WHERE, GROUP BY, HAVING, JOINS 및 중첩 쿼리를 비롯한 대부분의 명령과 JSON, 수학, 문자열 및 조건 함수를 지원합니다. 그러나 CloudWatch Logs는 읽기 작업만 지원하므로 DDL 또는 DML 문을 허용하지 않습니다. 지원되는 명령 및 함수의 전체 목록은 이전 섹션의 표를 참조하세요.

다중 로그 그룹 함수

CloudWatch Logs Insights는 여러 로그 그룹을 쿼리하는 기능을 지원합니다. SQL에서이 사용 사례를 해결하려면 `logGroups` 명령을 사용할 수 있습니다. 이 명령은 하나 이상의 로그 그룹과 관련된 CloudWatch Logs Insights의 데이터 쿼리에만 해당됩니다. 이 구문을 사용하면 각 로그 그룹에 대한 쿼리를 작성하고 `UNION` 명령과 결합하는 대신 명령에 지정하여 여러 로그 그룹을 쿼리할 수 있습니다.

구문:

```
`logGroups(
    logGroupIdentifier: ['LogGroup1','LogGroup2', ...'LogGroupn']
```

```
)
```

이 구문에서는 logGroupIndentifier 파라미터에 최대 50개의 로그 그룹을 지정할 수 있습니다. 모니터링 계정의 로그 그룹을 참조하려면 LogGroup 이름 대신 ARNs 사용합니다.

쿼리 예제:

```
SELECT LG1.Column1, LG1.Column2 from `logGroups(
    logGroupIdentifier: ['LogGroup1', 'LogGroup2']
)` as LG1
WHERE LG1.Column1 = 'ABC'
```

CloudWatch Logs를 쿼리할 때 FROM 문 뒤에 여러 로그 그룹이 포함된 다음 구문은 지원되지 않습니다.

```
SELECT Column1, Column2 FROM 'LogGroup1', 'LogGroup2', ...'LogGroupn'
WHERE Column1 = 'ABC'
```

제한 사항

SQL 또는 PPL 명령을 사용하는 경우 특정 필드를 백틱으로 묶어 쿼리합니다. 특수 문자(비영숫자 및 비숫자)가 있는 필드에는 백틱이 필요합니다. 예를 들어 @message, Operation.Export, 및를 백틱Test::Field으로 묶습니다. 순전히 알파벳 이름의 열을 백틱으로 묶을 필요는 없습니다.

간단한 필드가 있는 쿼리의 예:

```
SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
LIMIT 1000;
```

백틱이 추가된 동일한 쿼리:

```
SELECT `SessionToken`, `Operation`, `StartTime`  FROM `LogGroup-A`
LIMIT 1000;
```

CloudWatch Logs에만 국한되지 않는 추가 일반 제한 사항은 섹션을 참조하세요the section called “일반 SQL 제한 사항”.

샘플 쿼리 및 할당량

> **Note**
>
> 다음은 CloudWatch 데이터를 쿼리하는 CloudWatch Logs Insights 사용자와 OpenSearch 사용자 모두에게 적용됩니다.

CloudWatch Logs에서 사용할 수 있는 샘플 SQL 쿼리는 Amazon CloudWatch Logs Insights 콘솔의 저장된 쿼리 및 샘플 쿼리에서 예제를 참조하세요.

OpenSearch Service에서 CloudWatch Logs를 쿼리할 때 적용되는 제한에 대한 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs 할당량을 참조하세요. Amazon CloudWatch 제한에는 쿼리할 수 있는 CloudWatch Log 그룹 수, 실행할 수 있는 최대 동시 쿼리 수, 최대 쿼리 실행 시간 및 결과에 반환된 최대 행 수가 포함됩니다. CloudWatch Logs를 쿼리하는 데 사용하는 언어(즉, OpenSearch PPL, SQL 및 Logs Insights)에 관계없이 제한은 동일합니다.

SQL 명령

주제

- 문자열 함수
- 날짜 및 시간 함수
- 집계 함수
- 조건 함수
- JSON 함수
- 배열 함수
- 윈도 함수
- 변환 함수
- 조건자 함수
- 맵 함수
- 수학 함수
- 생성기 함수
- SELECT 절
- WHERE 절
- GROUP BY 절

- HAVING 절
- ORDER BY 절
- JOIN 절
- LIMIT 절
- CASE 절
- 공통 테이블 표현식
- EXPLAIN
- LATERAL SUBQUERY 절
- LATERAL VIEW 절
- LIKE 조건자
- OFFSET
- PIVOT 절
- 집합 연산자
- SORT BY 절
- UNPIVOT

문자열 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
ascii(str)	의 첫 번째 문자의 숫자 값을 반환합니다`str`.
base64(빈)	인수를 바이너리에서 기본 64 문자열`bin`로 변환합니다.
bit_length(expr)	문자열 데이터의 비트 길이 또는 바이너리 데이터의 비트 수를 반환합니다.

함수	설명
btrim(str)	에서 선행 및 후행 공백 문자를 제거합니다`str`.
btrim(str, trimStr)	에서 선행 및 후행 `trimStr` 문자를 제거합니다`str`.
char(expr)	와 동일한 바이너리를 가진 ASCII 문자를 반환합니다`expr`. n이 256보다 큰 경우 결과는 chr(n % 256)과 같습니다.
char_length(expr)	문자열 데이터의 문자 길이 또는 이진 데이터의 바이트 수를 반환합니다. 문자열 데이터의 길이에는 후행 공백이 포함됩니다. 바이너리 데이터의 길이에는 바이너리 0이 포함됩니다.
character_length(expr)	문자열 데이터의 문자 길이 또는 이진 데이터의 바이트 수를 반환합니다. 문자열 데이터의 길이에는 후행 공백이 포함됩니다. 바이너리 데이터의 길이에는 바이너리 0이 포함됩니다.
chr(expr)	와 동일한 바이너리를 가진 ASCII 문자를 반환합니다`expr`. n이 256보다 큰 경우 결과는 chr(n % 256)과 같습니다.
concat_ws(sep[, str | array(str)]+)	null 값을 `sep`건너뛰고 로 구분된 문자열의 연결을 반환합니다.
포함(왼쪽, 오른쪽)	부울을 반환합니다. 오른쪽이 왼쪽 내에 있는 경우 값은 True입니다. 입력 표현식 중 하나가 NULL인 경우 NULL을 반환합니다. 그렇지 않으면가 False를 반환합니다. 왼쪽 또는 오른쪽 모두 STRING 또는 BINARY 유형이어야 합니다.
decode(bin, charset)	두 번째 인수 문자 세트를 사용하여 첫 번째 인수를 디코딩합니다.

함수	설명
decode(expr, search, result [, search, result] ... [, default])	expr을 각 검색 값과 순서대로 비교합니다. expr이 검색 값과 같으면 디코딩은 해당 결과를 반환합니다. 일치하는 항목이 없으면 기본값을 반환합니다. 기본값을 생략하면 null을 반환합니다.
elt(n, input1, input2, ...)	n-번째 입력을 반환합니다. 예를 들어가 2일 `input2` 때 n를 반환합니다.
인코딩(str, 문자 집합)	두 번째 인수 문자 집합을 사용하여 첫 번째 인수를 인코딩합니다.
endswith(왼쪽, 오른쪽)	부울을 반환합니다. 왼쪽이 오른쪽으로 끝나는 경우 값은 True입니다. 입력 표현식 중 하나가 NULL인 경우 NULL을 반환합니다. 그렇지 않으면가 False를 반환합니다. 왼쪽 또는 오른쪽 모두 STRING 또는 BINARY 유형이어야 합니다.
find_in_set(str, str_array)	쉼표로 구분된 목록()에서 지정된 문자열()의 인덱스(1 기반`str`)를 반환합니다`str_array` . 문자열을 찾을 수 없거나 지정된 문자열(`str`)에 쉼표가 포함된 경우 0을 반환합니다.
format_number(expr1, expr2)	'#,###,###.##'과 `expr1` 같은 숫자의 형식을 `expr2` 소수 자릿수로 반올림합니다. `expr2`가 0이면 결과에 소수점이나 소수 부분이 없는 것입니다.는 사용자 지정 형식`expr2`도 수락합니다. MySQL의 FORMAT처럼 작동한다고 가정합니다.
format_string(strfmt, obj, ...)	printf 스타일 형식 문자열에서 형식이 지정된 문자열을 반환합니다.
initcap(str)	대문자로 된 각 단어의 첫 번째 문자와 `str` 함께를 반환합니다. 다른 모든 문자는 소문자입니다. 단어는 공백으로 구분됩니다.

함수	설명
instr(str, 하위 문자열)	에서의 첫 번째 발생에 대한 (1 기반) 인덱스substr를 반환합니다str.
lcase(str)	모든 문자가 소문자로 변경된 str 상태로를 반환합니다.
left(str, len)	문자열에서 가장 왼쪽len(len문자열 유형일 수 있음) 문자를 반환합니다str. len가 0보다 작거나 같으면 결과가 빈 문자열입니다.
len(expr)	문자열 데이터의 문자 길이 또는 이진 데이터의 바이트 수를 반환합니다. 문자열 데이터의 길이에는 후행 공백이 포함됩니다. 바이너리 데이터의 길이에는 바이너리 0이 포함됩니다.
length(expr)	문자열 데이터의 문자 길이 또는 이진 데이터의 바이트 수를 반환합니다. 문자열 데이터의 길이에는 후행 공백이 포함됩니다. 바이너리 데이터의 길이에는 바이너리 0이 포함됩니다.
levenshtein(str1, str2[, 임계값])	지정된 두 문자열 사이의 Levenshtein 거리를 반환합니다. 임계값이 설정되고 거리가 그보다 크면 -1을 반환합니다.
locate(substr, str[, pos])	위치 substr str 뒤에 있는의 첫 번째 발생 위치를 반환합니다pos. 지정된 pos 및 반환 값은 1 기반입니다.
lower(str)	모든 문자가 소문자로 변경된 str 상태로를 반환합니다.

함수	설명
lpad(str, len[, pad])	로 str왼쪽 패딩된를 길이pad로 반환합니다len. str가 보다 길면 len반환 값이 len 문자 또는 바이트로 단축됩니다. pad이 지정되지 않은 경우 str는 문자열인 경우 공백 문자로, 바이트 시퀀스인 경우 0으로 왼쪽에 패딩됩니다.
ltrim(str)	에서 선행 공백 문자를 제거합니다str.
luhn_check(str)	Luhn 알고리즘에 따라 문자열이 유효한지 확인합니다. 이 체크섬 함수는 신용 카드 번호 및 정부 식별 번호에 광범위하게 적용되어 유효한 번호를 잘못 입력되거나 잘못된 번호와 구분합니다.
mask(input[, upperChar, lowerChar, digitChar, otherChar])	는 지정된 문자열 값을 마스킹합니다. 함수는 문자를 'X' 또는 'x'로 바꾸고 숫자는 'n'으로 바꿉니다. 이는 민감한 정보가 제거된 테이블의 복사본을 생성하는 데 유용할 수 있습니다.
octet_length(expr)	문자열 데이터의 바이트 길이 또는 바이너리 데이터의 바이트 수를 반환합니다.
overlay(입력, 교체, pos[, len])	input를 로 시작하고 길이replace가 인 로 바꿉pos니다len.
position(substr, str[, pos])	위치 substr str 뒤에 있는의 첫 번째 발생 위치를 반환합니다pos. 지정된 pos 및 반환 값은 1 기반입니다.
printf(strfmt, obj, ...)	printf 스타일 형식 문자열에서 형식이 지정된 문자열을 반환합니다.
regexp_count(str, regexp)	문자열에서 정규식 패턴이 regexp 일치하는 횟수를 반환합니다str.

함수	설명
regexp_extract(str, regexp[, idx])	regexp 표현식과 str 일치하고 정규식 그룹 인덱스에 해당하는의 첫 번째 문자열을 추출합니다.
regexp_extract_all(str, regexp[, idx])	regexp 표현식과 str 일치하고 정규식 그룹 인덱스에 해당하는의 모든 문자열을 추출합니다.
regexp_instr(str, regexp)	문자열에서 정규식을 검색하고 일치하는 하위 문자열의 시작 위치를 나타내는 정수를 반환합니다. 위치는 0이 아닌 1 기반입니다. 일치하는 항목이 없으면가 0을 반환합니다.
regexp_replace(str, regexp, rep[, position])	일치하는의 모든 하위 문자열str을 regexp로 바꿉니다rep.
regexp_substr(str, regexp)	문자열 내의 정규식과 일치하는 하위 문자열regexp을 반환합니다str. 정규식을 찾을 수 없는 경우 결과는 null입니다.
repeat(str, n)	지정된 문자열 값을 n번 반복하는 문자열을 반환합니다.
replace(str, search[, replace])	의 모든 발생을 search로 바꿉니다replace.
right(str, len)	문자열에서 가장 오른쪽len(len문자열 유형일 수 있음) 문자를 반환합니다str. len가 0보다 작거나 같으면 결과가 빈 문자열입니다.
rpad(str, len[, pad])	로 str오른쪽 패딩된 pad를 길이로 반환합니다len. 이 보다 str 길면 len반환 값이 len 문자로 단축됩니다. pad이 지정되지 않은 경우 str는 문자열인 경우 공백 문자로, 이진 문자열인 경우 0으로 오른쪽으로 패딩됩니다.
rtrim(str)	에서 후행 공백 문자를 제거합니다str.

함수	설명
문장(str[, lang, country])	단어 배열str로 분할합니다.
Soundex(str)	문자열의 Soundex 코드를 반환합니다.
공백(n)	n 공백으로 구성된 문자열을 반환합니다.
split(str, regex, limit)	일치하는 str 발생을 분할regex하고 최대 길이의 배열을 반환합니다. limit
split_part(str, 구분 기호, partNum)	구분 기호str로 분할하고 분할의 요청된 부분(1 기반)을 반환합니다. 입력이 null이면는 null을 반환합니다. partNum가 분할된 부분의 범위를 벗어나면는 빈 문자열을 반환합니다. partNum가 0이면에서 오류가 발생합니다. partNum가 음수인 경우 부분은 문자열 끝에서 역순으로 계산됩니다. delimiter 가 빈 문자열인 경우 str는 분할되지 않습니다.
startwith(왼쪽, 오른쪽)	부울을 반환합니다. 왼쪽이 오른쪽으로 시작하는 경우 값은 True입니다. 입력 표현식 중 하나가 NULL인 경우 NULL을 반환합니다. 그렇지 않으면가 False를 반환합니다. 왼쪽 또는 오른쪽 모두 STRING 또는 BINARY 유형이어야 합니다.
substr(str, pos[, len])	에서 시작하고 길이str가 pos인의 하위 문자열len또는에서 시작하고 길이pos가 인 바이트 배열 조각을 반환합니다len.
substr(str FROM pos[FOR len]])	에서 시작하고 길이str가 pos인의 하위 문자열len또는에서 시작하고 길이pos가 인 바이트 배열 조각을 반환합니다len.
하위 문자열(str, pos[, len])	에서 시작하고 길이str가 pos인의 하위 문자열len또는에서 시작하고 길이pos가 인 바이트 배열 조각을 반환합니다len.

함수	설명
하위 문자열(str FROM pos[FOR len]])	에서 시작하고 길이str가 pos인의 하위 문자열 len또는에서 시작하고 길이pos가 인 바이트 배열 조각을 반환합니다len.
substring_index(str, delim, count)	구분 기호가 count 발생str하기 전에 하위 문자열을 반환합니다delim. count가 양수이면 최종 구분 기호 왼쪽에 있는 모든 항목(왼쪽에서 계산)이 반환됩니다. count가 음수이면 최종 구분 기호 오른쪽에 있는 모든 항목(오른쪽에서 계산)이 반환됩니다. 함수 substring_index는를 검색할 때 대/소문자를 구분하여 일치를 수행합니다delim.
to_binary(str[, fmt])	제공된를 기반으로 입력을 str 이진 값으로 변환합니다fmt.는 "hex", "utf-8", "utf8" 또는 "base64"의 대소문자를 구분하지 않는 문자열 리터럴일 fmt 수 있습니다. 기본적으로 변환을 위한 이진 형식은이 생략된 경우 "hex"fmt입니다. 하나 이상의 입력 파라미터가 NULL인 경우 함수는 NULL을 반환합니다.

함수	설명
to_char(numberExpr, formatExpr)	numberExpr 를 기반으로 문자열로 변환합니다formatExpr . 변환에 실패하면 예외가 발생합니다. 형식은 대소문자를 구분하지 않는 '0' 또는 '9': 0에서 9 사이의 예상 숫자를 지정합니다. 형식 문자열의 시퀀스 0 또는 9는 입력 값의 숫자 시퀀스와 일치하여 형식 문자열의 해당 시퀀스와 동일한 길이의 결과 문자열을 생성합니다. 0/9 시퀀스가 10진수 값의 일치하는 부분보다 많은 숫자를 포함하고, 0으로 시작하고, 10진수 앞에 있는 경우 결과 문자열은 0으로 왼쪽 패딩됩니다. 그렇지 않으면 공백으로 채워집니다. '.' 또는 'D': 소수점의 위치를 지정합니다(선택 사항, 한 번만 허용). ',' 또는 'G': 그룹화(천) 구분자(,)의 위치를 지정합니다. 각 그룹화 구분자의 왼쪽과 오른쪽에 0 또는 9가 있어야 합니다. '
to_number(expr, fmt)	문자열 형식 'fmt'를 기반으로 문자열 'expr'을 숫자로 변환합니다. 변환에 실패하면 예외가 발생합니다. 형식은 대소문자를 구분하지 않는 '0' 또는 '9': 0에서 9 사이의 예상 숫자를 지정합니다. 형식 문자열의 시퀀스 0 또는 9는 입력 문자열의 숫자 시퀀스와 일치합니다. 0/9 시퀀스가 0으로 시작하고 소수점 앞에 있는 경우 동일한 크기의 숫자 시퀀스만 일치시킬 수 있습니다. 그렇지 않으면 시퀀스가 9로 시작하거나 소수점 뒤에 있는 경우 크기가 같거나 작은 숫자 시퀀스와 일치할 수 있습니다. '.' 또는 'D': 소수점의 위치를 지정합니다(선택 사항, 한 번만 허용). ',' 또는 'G': 그룹화(천) 구분자(,)의 위치를 지정합니다. 각 그룹화 구분자의 왼쪽과 오른쪽에 0 또는 9가 있어야 합니다. 'expr'은 숫자 크기와 관련된 그룹화 구분자와 일치해야 합니다. '

함수	설명
to_varchar(numberExpr, formatExpr)	numberExpr 를 기반으로 문자열로 변환합니다formatExpr . 변환에 실패하면 예외가 발생합니다. 형식은 대소문자를 구분하지 않는 '0' 또는 '9': 0에서 9 사이의 예상 숫자를 지정합니다. 형식 문자열의 시퀀스 0 또는 9는 입력 값의 숫자 시퀀스와 일치하여 형식 문자열의 해당 시퀀스와 동일한 길이의 결과 문자열을 생성합니다. 0/9 시퀀스가 10진수 값의 일치하는 부분보다 많은 숫자를 포함하고, 0으로 시작하고, 10진수 앞에 있는 경우 결과 문자열은 0으로 왼쪽 패딩됩니다. 그렇지 않으면 공백으로 채워집니다. '.' 또는 'D': 소수점의 위치를 지정합니다(선택 사항, 한 번만 허용). ',' 또는 'G': 그룹화(천) 구분자(,)의 위치를 지정합니다. 각 그룹화 구분자의 왼쪽과 오른쪽에 0 또는 9가 있어야 합니다. '
translate(입력, 시작, 종료)	input 문자열에 있는 문자를 from 문자열의 해당 문자로 바꾸어 to 문자열을 변환합니다.
trim(str)	에서 선행 및 후행 공백 문자를 제거합니다str.
trim(둘 다 str에서)	에서 선행 및 후행 공백 문자를 제거합니다str.
trim(LEADING FROM str)	에서 선행 공백 문자를 제거합니다str.
trim(TRAILING FROM str)	에서 후행 공백 문자를 제거합니다str.
trim(trimStr FROM 문자열)	에서 선행 및 후행 trimStr 문자를 제거합니다str.
trim(양쪽 trimStr FROM str)	에서 선행 및 후행 trimStr 문자를 제거합니다str.
trim(LEADING trimStr FROM str)	에서 선행 trimStr 문자를 제거합니다str.
trim(TRAILING trimStr FROM str)	에서 후행 trimStr 문자를 제거합니다str.

함수	설명
try_to_binary(str[, fmt])	동일한 작업을 to_binary 수행하는의 특수 버전이지만 변환을 수행할 수 없는 경우 오류를 발생시키는 대신 NULL 값을 반환합니다.
try_to_number(expr, fmt)	문자열 형식에 따라 문자열 'expr'을 숫자로 변환합니다fmt. 문자열 'expr'이 예상 형식과 일치하지 않으면 NULL을 반환합니다. 형식은 to_number 함수와 동일한 의미 체계를 따릅니다.
ucase(str)	모든 문자가 대문자로 변경된 str 상태로를 반환합니다.
unbase64(str)	인수를 기본 64 문자열에서 바이너리str로 변환합니다.
upper(str)	모든 문자가 대문자로 변경된 str 상태로를 반환합니다.

예시

```
-- 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|
+---------------+
```

날짜 및 시간 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
add_months(start_date, num_months)	num_months 이후 날짜를 반환합니다start_date .
convert_timezone([sourceTz,]targetTz, sourceTs)	시간대가 없는 타임스탬프를 sourceTz 시간대sourceTs에서 로 변환합니다targetTz.
curdate()	쿼리 평가 시작 시 현재 날짜를 반환합니다. 동일한 쿼리 내의 모든 커데이트 호출은 동일한 값을 반환합니다.
current_date()	쿼리 평가 시작 시 현재 날짜를 반환합니다. 동일한 쿼리 내에서 current_date의 모든 호출은 동일한 값을 반환합니다.
현재_날짜	쿼리 평가 시작 시 현재 날짜를 반환합니다.
current_timestamp()	쿼리 평가 시작 시 현재 타임스탬프를 반환합니다. 동일한 쿼리 내에서 current_timestamp의 모든 호출은 동일한 값을 반환합니다.
current_timestamp	쿼리 평가 시작 시 현재 타임스탬프를 반환합니다.
current_timezone()	현재 세션 현지 시간대를 반환합니다.
date_add(start_date, num_days)	num_days 이후 날짜를 반환합니다start_date .
date_diff(endDate, startDate)	에서 까지의 일수startDate 를 반환합니다endDate.
date_format(타임스탬프, fmt)	날짜 형식 로 지정된 형식의 문자열 timestamp 값으로 변환합니다fmt.
date_from_unix_date(일)	1970-01-01 이후 일수에서 날짜를 생성합니다.

함수	설명
date_part(필드, 소스)	날짜/타임스탬프 또는 간격 소스의 일부를 추출합니다.
date_sub(start_date, num_days)	num_days 이전 날짜를 반환합니다start_date .
date_trunc(fmt, ts)	형식 모델에서 지정한 단위로 ts 잘린 타임스탬프를 반환합니다fmt.
dateadd(start_date, num_days)	num_days 이후 날짜를 반환합니다start_date .
datediff(endDate, startDate)	에서 까지의 일수startDate 를 반환합니다endDate.
datepart(필드, 소스)	날짜/타임스탬프 또는 간격 소스의 일부를 추출합니다.
일(날짜)	날짜/타임스탬프의 요일을 반환합니다.
dayofmonth(날짜)	날짜/타임스탬프의 요일을 반환합니다.
dayofweek(날짜)	날짜/타임스탬프의 요일을 반환합니다(1 = 일요일, 2 = 월요일, ..., 7 = 토요일).
dayofyear(날짜)	날짜/타임스탬프의 연도 날짜를 반환합니다.
extract(field FROM 소스)	날짜/타임스탬프 또는 간격 소스의 일부를 추출합니다.
from_unixtime(unix_time[, fmt])	지정된 unix_time 에서를 반환합니다fmt.
from_utc_timestamp(타임스탬프, 시간대)	'2017-07-14 02:40:00.0'과 같은 타임스탬프가 주어지면는 이를 UTC의 시간으로 해석하고 해당 시간을 지정된 시간대의 타임스탬프로 렌더링합니다. 예를 들어 'GMT+1'은 '2017-07-14 03:40:00.0'을 생성합니다.

함수	설명
시간(타임스탬프)	문자열/타임스탬프의 시간 구성 요소를 반환합니다.
last_day(날짜)	날짜가 속한 달의 마지막 날을 반환합니다.
localtimestamp()	쿼리 평가 시작 시 시간대가 없는 현재 타임스탬프를 반환합니다. 동일한 쿼리 내의 모든 localtimestamp 호출은 동일한 값을 반환합니다.
localtimestamp	쿼리 평가 시작 시 세션 시간대의 현재 로컬 날짜-시간을 반환합니다.
make_date(년, 월, 일)	연도, 월 및 일 필드에서 날짜를 생성합니다.
make_dt_interval([days[, hours[, mins[, secs]]]])	DayTimeIntervalType 기간을 일, 시간, 분 및 초로 설정합니다.
make_interval([years[, months[, weeks[, days[, hours[, mins[, secs]]]]]]])	년, 월, 주, 일, 시간, 분 및 초부터 간격을 둡니다.
make_timestamp(년, 월, 일, 시간, 분, 초[, 시간대])	연도, 월, 일, 시간, 분, 초 및 시간대 필드에서 타임스탬프를 생성합니다.
make_timestamp_ltz(연도, 월, 일, 시간, 분, 초[, 시간대])	연도, 월, 일, 시간, 분, 초 및 시간대 필드의 현지 시간대로 현재 타임스탬프를 생성합니다.
make_timestamp_ntz(년, 월, 일, 시간, 분, 초)	년, 월, 일, 시간, 분, 초 필드에서 로컬 날짜-시간을 생성합니다.
make_ym_interval([years[, months]])	년, 월에서 년-월 간격을 설정합니다.
minute(타임스탬프)	문자열/타임스탬프의 분 구성 요소를 반환합니다.
month(날짜)	날짜/타임스탬프의 월 구성 요소를 반환합니다.

함수	설명
months_between(timestamp1, timestamp2[, roundOff])	`timestamp1` 가 보다 이후인 경우 `timestamp2` 결과는 양수입니다. `timestamp1` 및 `timestamp2` 가 같은 월의 날짜이거나 둘 다 마지막 월의 날짜인 경우 시간대는 무시됩니다. 그렇지 않으면 차이는 매월 31일을 기준으로 계산되며 roundOff=false가 아니면 8자리로 반올림됩니다.
next_day(start_date, day_of_week)	표시된 대로 `start_date` 이후이고 이름이 지정된 첫 번째 날짜를 반환합니다. 입력 파라미터 중 하나 이상이 NULL인 경우 함수는 NULL을 반환합니다.
지금()	쿼리 평가 시작 시 현재 타임스탬프를 반환합니다.
quarter(날짜)	날짜의 분기를 1~4 범위로 반환합니다.
second(타임스탬프)	문자열/타임스탬프의 두 번째 구성 요소를 반환합니다.
session_window(time_column, gap_duration)	열 및 간격 기간을 지정하는 타임스탬프가 지정된 세션 기간을 생성합니다. 자세한 설명과 예제는 구조화 스트리밍 가이드 문서의 '시간 범위 유형'을 참조하세요.
timestamp_micros(마이크로초)	UTC 에포크 이후 마이크로초 수에서 타임스탬프를 생성합니다.
timestamp_millis(밀리초)	UTC 에포크 이후 밀리초 수에서 타임스탬프를 생성합니다.
timestamp_seconds(초)	UTC 에포크 이후 초(분수일 수 있음)부터 타임스탬프를 생성합니다.

함수	설명
to_date(date_str[, fmt])	`date_str` 표현식이 포함된 `fmt` 표현식을 날짜로 구문 분석합니다. 잘못된 입력과 함께 null을 반환합니다. 기본적으로가 생략`fmt`된 경우 캐스팅 규칙을 따라 날짜를 지정합니다.
to_timestamp(timestamp_str[, fmt])	`timestamp_str` 표현식이 포함된 `fmt` 표현식을 타임스탬프로 구문 분석합니다. 잘못된 입력과 함께 null을 반환합니다. 기본적으로가 생략`fmt`된 경우 캐스팅 규칙을 따라 타임스탬프로 이동합니다.
to_timestamp_ltz(timestamp_str[, fmt])	`timestamp_str` 표현식이 있는 `fmt` 표현식을 현지 시간대가 있는 타임스탬프로 구문 분석합니다. 잘못된 입력과 함께 null을 반환합니다. 기본적으로가 생략`fmt`된 경우 캐스팅 규칙을 따라 타임스탬프로 이동합니다.
to_timestamp_ntz(timestamp_str[, fmt])	`timestamp_str` 표현식이 있는 `fmt` 표현식을 시간대가 없는 타임스탬프로 구문 분석합니다. 잘못된 입력과 함께 null을 반환합니다. 기본적으로가 생략`fmt`된 경우 캐스팅 규칙을 따라 타임스탬프로 이동합니다.
to_unix_timestamp(timeExp[, fmt])	지정된 시간의 UNIX 타임스탬프를 반환합니다.
to_utc_timestamp(타임스탬프, 시간대)	'2017-07-14 02:40:00.0'과 같은 타임스탬프가 주어지면는 이를 지정된 시간대의 시간으로 해석하고 해당 시간을 UTC의 타임스탬프로 렌더링합니다. 예를 들어 'GMT+1'은 '2017-07-14 01:40:00.0'을 생성합니다.
trunc(날짜, fmt)	형식 모델에서 지정한 단위로 잘린 날짜의 `date` 시간 부분을 반환합니다`fmt`.
try_to_timestamp(timestamp_str[, fmt])	`timestamp_str` 표현식이 포함된 `fmt` 표현식을 타임스탬프로 구문 분석합니다.

함수	설명
unix_date(날짜)	1970-01-01 이후 일수를 반환합니다.
unix_micros(타임스탬프)	1970-01-01 00:00:00 UTC 이후 마이크로초 수를 반환합니다.
unix_millis(타임스탬프)	1970-01-01 00:00:00 UTC 이후 밀리초 수를 반환합니다. 더 높은 수준의 정밀도를 자릅니다.
unix_seconds(타임스탬프)	1970-01-01 00:00:00 UTC 이후의 초 수를 반환합니다. 더 높은 수준의 정밀도를 자릅니다.
unix_timestamp([timeExp[, fmt]])	현재 또는 지정된 시간의 UNIX 타임스탬프를 반환합니다.
weekday(날짜)	날짜/타임스탬프의 요일을 반환합니다(0 = 월요일, 1 = 화요일, ..., 6 = 일요일).
weekofyear(날짜)	지정된 날짜의 연도 요일을 반환합니다. 일주일은 월요일에 시작되는 것으로 간주되며, 1주차는 >3일의 첫 번째 주입니다.
window(time_column, window_duration[, slide_duration[, start_time]])	열을 지정하는 타임스탬프가 주어지면 행을 하나 이상의 기간으로 버킷화합니다. 창 시작은 포함되지만 창 끝은 제외됩니다. 예를 들어 12:05는 [12:05,12:10) 창에 있지만 [12:00,12:05)에는 없습니다. Windows는 마이크로초 정밀도를 지원할 수 있습니다. 월 단위의 Windows는 지원되지 않습니다. 자세한 설명과 예제는 구조화된 스트리밍 가이드 문서의 '이벤트 시간에 대한 창 작업'을 참조하세요.

함수	설명
window_time(window_column)	기간의 이벤트 시간 값에 사용할 수 있는 시간/세션 기간 열에서 시간 값을 추출합니다. 추출된 시간은 (window.end - 1)이며, 이는 집계 창에 배타적인 상한이 있다는 사실을 반영합니다. [start, end) 자세한 설명과 예제는 구조화 스트리밍 가이드 문서의 '이벤트 시간에 대한 창 작업'을 참조하세요.
연도(날짜)	날짜/타임스탬프의 연도 구성 요소를 반환합니다.

예시

```
-- 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|
+----------------+
```

집계 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

집계 함수는 행 전반의 값에 대해 작동하여 합계, 평균, 계산, 최소값/최대값, 표준 편차, 추정과 같은 수학 계산과 일부 비수학 연산을 수행합니다.

구문

```
aggregate_function(input1 [, input2, ...]) FILTER (WHERE boolean_expression)
```

파라미터

- boolean_expression - 결과 유형 부울로 평가되는 표현식을 지정합니다. 논리 연산자(AND, OR)를 사용하여 둘 이상의 표현식을 결합할 수 있습니다.

순서가 지정된 집계 함수

이러한 집계 함수는 다른 집계 함수와 다른 구문을 사용하여 값을 정렬할 표현식(일반적으로 열 이름)을 지정합니다.

구문

```
{ PERCENTILE_CONT | PERCENTILE_DISC }(percentile) WITHIN GROUP (ORDER BY
 { order_by_expression [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [ , ... ] }) FILTER
 (WHERE boolean_expression)
```

파라미터

- percentile - 찾으려는 값의 백분위수입니다. 백분위수는 0.0에서 1.0 사이의 상수여야 합니다.
- order_by_expression - 값을 집계하기 전에 정렬할 표현식(일반적으로 열 이름)입니다.
- boolean_expression - 결과 유형 부울로 평가되는 표현식을 지정합니다. 논리 연산자(AND, OR)를 사용하여 둘 이상의 표현식을 결합할 수 있습니다.

예시

```
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|
+----------+-------+--------+-------+--------+-----+-----+-----+-----+
```

조건 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
coalesce(expr1, expr2, ...)	존재하는 경우 첫 번째 null이 아닌 인수를 반환합니다. 그렇지 않은 경우 null입니다.
if(expr1, expr2, expr3)	가 true로 expr1 평가되면를 반환하고, expr2그렇지 않으면를 반환합니다expr3.
ifnull(expr1, expr2)	expr1가 null이면를 반환하고, expr1 그렇지 expr2 않으면를 반환합니다.
nanvl(expr1, expr2)	NaN이 아니면 expr1를 반환합니다expr2.
nullif(expr1, expr2)	가와 expr1같expr2거나 expr1 그렇지 않으면 null을 반환합니다.
nvl(expr1, expr2)	expr1가 null이면를 반환하고, expr1 그렇지 expr2 않으면를 반환합니다.
nvl2(expr1, expr2, expr3)	가 nullexpr2expr1이 아니면를 반환하고, expr3 그렇지 않으면를 반환합니다.

함수	설명
CASE WHEN expr1 THEN expr2 [WHEN expr3 THEN expr4]* [ELSE expr5] END	expr1 = true이면를 반환하고expr2, expr3 = true이면를 반환하고, expr4그렇지 않으면를 반환합니다expr5.

예시

```
-- 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 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
from_json (jsonStr, 스키마[, 옵션])	지정된 `jsonStr` 및 `schema`가 있는 구조체 값을 반환합니다.
get_json_object(json_txt, 경로)	`path`에서 json 객체를 추출합니다.
json_array_length(jsonArray)	가장 바깥쪽 JSON 배열의 요소 수를 반환합니다.
json_object_keys(json_object)	가장 바깥쪽 JSON 객체의 모든 키를 배열로 반환합니다.
json_tuple(jsonStr, p1, p2, ..., pn)	함수 get_json_object와 같은 튜플을 반환하지만 여러 이름을 사용합니다. 모든 입력 파라미터와 출력 열 유형은 문자열입니다.
schema_of_json(json[, 옵션])	JSON 문자열의 DDL 형식으로 스키마를 반환합니다.
to_json(expr[, 옵션])	지정된 구조체 값이 있는 JSON 문자열을 반환합니다.

예시

```
-- 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}]               |
+-------------------------+
```

배열 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
array(expr, ...)	지정된 요소가 있는 배열을 반환합니다.
array_append(배열, 요소)	첫 번째 인수로 전달된 배열 끝에 요소를 추가합니다. 요소 유형은 배열의 요소 유형과 유사해야 합니다. Null 요소도 배열에 추가됩니다. 그러나 배열이 전달되면는 NULL 출력이 NULL입니다.
array_compact(배열)	배열에서 null 값을 제거합니다.
array_contains(배열, 값)	배열에 값이 포함된 경우 true를 반환합니다.
array_distinct(배열)	배열에서 중복 값을 제거합니다.

함수	설명
array_except(array1, array2)	중복 없이 array2가 아닌 array1의 요소 배열을 반환합니다.
array_insert(x, pos, val)	배열 x의 인덱스 위치에 val을 배치합니다. 배열 인덱스는 1부터 시작합니다. 함수가 현재 마지막 요소 뒤에 새 요소를 삽입하는 최대 음수 인덱스는 -1입니다. 배열 크기 위의 인덱스는 배열을 추가하거나 인덱스가 음수인 경우 배열 앞에 'null' 요소를 추가합니다.
array_intersect(array1, array2)	중복 없이 array1과 array2의 교차점에 있는 요소의 배열을 반환합니다.
array_join(배열, 구분 기호[, nullReplacement])	null을 대체하기 위해 구분 기호와 선택적 문자열을 사용하여 지정된 배열의 요소를 연결합니다. nullReplacement에 값을 설정하지 않으면 null 값이 필터링됩니다.
array_max(배열)	배열의 최대값을 반환합니다. NaN은 이중/부동 소수점 유형에 대한 비NaN 요소보다 큽니다. NULL 요소는 건너뜁니다.
array_min(배열)	배열의 최소값을 반환합니다. NaN은 이중/부동 소수점 유형에 대한 비NaN 요소보다 큽니다. NULL 요소는 건너뜁니다.
array_position(배열, 요소)	배열의 첫 번째 일치 요소의 (1 기반) 인덱스를 길게 반환하거나 일치하는 요소가 없으면 0을 반환합니다.
array_prepend(배열, 요소)	첫 번째 인수로 전달된 배열의 시작 부분에 요소를 추가합니다. 요소 유형은 배열의 요소 유형과 동일해야 합니다. Null 요소도 배열 앞에 추가됩니다. 그러나 전달된 배열이 NULL인 경우 출력은 NULL입니다.

함수	설명
array_remove(배열, 요소)	배열에서 요소와 동일한 모든 요소를 제거합니다.
array_repeat(요소, 개수)	요소 수 시간이 포함된 배열을 반환합니다.
array_union(array1, array2)	중복 없이 array1과 array2의 조합에 있는 요소의 배열을 반환합니다.
arrays_overlap(a1, a2)	a1에 a2에도 null이 아닌 요소가 하나 이상 포함된 경우 true를 반환합니다. 배열에 공통 요소가 없고 둘 다 비어 있지 않고 둘 중 하나에 null 요소 null이 포함된 경우, 그렇지 않으면 false가 반환됩니다.
arrays_zip(a1, a2, ...)	N번째 구조체에 입력 배열의 모든 N번째 값이 포함된 구조체의 병합된 배열을 반환합니다.
flatten(arrayOfArrays)	배열 배열을 단일 배열로 변환합니다.
get(배열, 인덱스)	지정된 (0 기반) 인덱스에서 배열의 요소를 반환합니다. 인덱스가 배열 경계 외부를 가리키는 경우이 함수는 NULL을 반환합니다.
sequence(시작, 중지, 단계)	시작부터 중지(포함)까지 단계별로 증가하는 요소 배열을 생성합니다. 반환된 요소의 유형은 인수 표현식의 유형과 동일합니다. 지원되는 유형은 바이트, 짧은, 정수, 긴, 날짜, 타임스탬프입니다. 시작 및 중지 표현식은 동일한 유형으로 확인되어야 합니다. 시작 및 중지 표현식이 'date' 또는 'timestamp' 유형으로 확인되면 단계 표현식은 'interval' 또는 'year-month interval' 또는 'day-time interval' 유형으로 확인되어야 하며, 그렇지 않으면 시작 및 중지 표현식과 동일한 유형으로 확인되어야 합니다.
셔플(배열)	지정된 배열의 임의 순열을 반환합니다.

함수	설명
slice(x, 시작, 길이)	지정된 길이의 인덱스 시작부터 시작하여(배열 인덱스는 1에서 시작하거나 시작이 음수인 경우 끝부터 시작) 배열 x를 하위 설정합니다.
sort_array(array[, ascendingOrder])	배열 요소의 자연스러운 순서에 따라 입력 배열을 오름차순 또는 내림차순으로 정렬합니다. NaN은 이중/부동 소수점 유형에 대한 비NaN 요소보다 큽니다. Null 요소는 반환된 배열의 시작 부분에 오름차순으로 배치되거나 반환된 배열의 끝에 내림차순으로 배치됩니다.

예시

```
-- 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]|
+-----------------------------------------+
```

윈도 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

창 함수는 창이라고 하는 행 그룹에서 작동하며 행 그룹을 기반으로 각 행의 반환 값을 계산합니다. 창 함수는 이동 평균 계산, 누적 통계 계산 또는 현재 행의 상대 위치가 지정된 행 값에 액세스하는 등의 작업을 처리하는 데 유용합니다.

구문

```
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 ] )
```

파라미터

- 순위 함수

  구문: `RANK | DENSE_RANK | PERCENT_RANK | NTILE | ROW_NUMBER`

  분석 함수

  구문: `CUME_DIST | LAG | LEAD | NTH_VALUE | FIRST_VALUE | LAST_VALUE`

  집계 함수

  구문: `MAX | MIN | COUNT | SUM | AVG | ...`
- `nulls_option` - 창 함수를 평가할 때 null 값을 건너뛸지 여부를 지정합니다. RESPECT NULLS는 null 값을 건너뛰지 않고 IGNORE NULLS는 건너뛰기를 의미합니다. 지정하지 않으면 기본값은 RESPECT NULLS입니다.

  구문: `{ IGNORE | RESPECT } NULLS`

  참고: `Only LAG | LEAD | NTH_VALUE | FIRST_VALUE |`는와 함께 사용할 수 `LAST_VALUE` 있습니다`IGNORE NULLS`.
- `window_frame` - 창을 시작할 행과 창을 종료할 위치를 지정합니다.

  구문: `{ RANGE | ROWS } { frame_start | BETWEEN frame_start AND frame_end }`

  frame_start 및 frame_end의 구문은 다음과 같습니다.

  구문: `UNBOUNDED PRECEDING | offset PRECEDING | CURRENT ROW | offset FOLLOWING | UNBOUNDED FOLLOWING`

  offset: 현재 행의 위치에서 오프셋을 지정합니다.

참고 frame_end를 생략하면 기본적으로 현재 ROW로 설정됩니다.

예시

```
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|
+--+----+----+----+---------+-----------+----------+
```

변환 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
bigint(expr)	'expr' 값을 대상 데이터 유형 'bigint'로 캐스팅합니다.
바이너리(expr)	'expr' 값을 대상 데이터 유형 'binary'로 캐스팅합니다.
부울(expr)	`expr` 값을 대상 데이터 유형 `boolean`으로 캐스팅합니다.
cast(expr AS 유형)	'expr' 값을 대상 데이터 유형 'type'으로 캐스팅합니다.
date(expr)	'expr' 값을 대상 데이터 유형 'date'로 캐스팅합니다.
10진수(expr)	'expr' 값을 대상 데이터 유형 'decimal'로 캐스팅합니다.
double(expr)	`expr` 값을 대상 데이터 유형 `double`으로 캐스팅합니다.
float(expr)	`expr` 값을 대상 데이터 유형 `float`으로 캐스팅합니다.
int(expr)	`expr` 값을 대상 데이터 유형 `int`로 캐스팅합니다.
smallint(expr)	`expr` 값을 대상 데이터 유형 `smallint`로 캐스팅합니다.
string(expr)	'expr' 값을 대상 데이터 유형 'string'으로 캐스팅합니다.
timestamp(expr)	'expr' 값을 대상 데이터 유형 'timestamp'로 캐스팅합니다.
tinyint(expr)	'expr' 값을 대상 데이터 유형 'tinyint'로 캐스팅합니다.

예시

```
-- cast
SELECT cast(field as int);
+---------------+
|CAST(field AS INT)|
+---------------+
|             10|
+---------------+
```

조건자 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
! expr	논리 not.
expr1 < expr2	`expr1`이 `expr2`보다 작으면 true를 반환합니다.
expr1 <= expr2	`expr1`이 `expr2`보다 작거나 같으면 true를 반환합니다.
expr1 <=> expr2	null이 아닌 피연산자에 대해 EQUAL(=) 연산자와 동일한 결과를 반환하지만 둘 다 null이면 true, 둘 중 하나가 null이면 false를 반환합니다.
expr1 = expr2	`expr1`이 `expr2`와 같으면 true를 반환하고, 그렇지 않으면 false를 반환합니다.
expr1 == expr2	`expr1`이 `expr2`와 같으면 true를 반환하고, 그렇지 않으면 false를 반환합니다.
expr1 > expr2	`expr1`이 `expr2`보다 큰 경우 true를 반환합니다.
expr1 >= expr2	`expr1`이 `expr2`보다 크거나 같으면 true를 반환합니다.
expr1 및 expr2	논리적 AND.

함수	설명
str 유사 패턴[ESCAPE 이스케이프]	str이 `pattern`과 대/소문자를 구분하지 않고 일치하면 true를 반환하고, 인수가 null이면 null을 반환하고, 그렇지 않으면 false를 반환합니다.
expr1 in(expr2, expr3, ...)	`expr`가 임의의 valN과 같으면 true를 반환합니다.
isan(expr)	`expr`가 NaN이면 true를 반환하고 그렇지 않으면 false를 반환합니다.
isnotnull(expr)	`expr`이 null이 아니면 true를 반환하고 그렇지 않으면 false를 반환합니다.
isnull(expr)	`expr`이 null이면 true를 반환하고 그렇지 않으면 false를 반환합니다.
str like pattern[ESCAPE 이스케이프]	str이 `pattern`과 `escape` 일치하면 true를 반환하고, 인수가 null이면 null을 반환하고, 그렇지 않으면 false를 반환합니다.
expr 아님	논리 not.
expr1 또는 expr2	논리적 OR.
regexp(str, regexp)	`str`이 `regexp`와 일치하면 true를 반환하고, 그렇지 않으면 false를 반환합니다.
regexp_like(str, regexp)	`str`이 `regexp`와 일치하면 true를 반환하고, 그렇지 않으면 false를 반환합니다.
rlike(str, regexp)	`str`이 `regexp`와 일치하면 true를 반환하고, 그렇지 않으면 false를 반환합니다.

예시

```
-- !
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%)|
+-----------------------------------------------------------+
|                                                       true|
+-----------------------------------------------------------+
-- in
SELECT 1 in(1, 2, 3);
+----------------+
|(1 IN (1, 2, 3))|
+----------------+
|            true|
+----------------+
SELECT 1 in(2, 3, 4);
+----------------+
|(1 IN (2, 3, 4))|
+----------------+
|           false|
+----------------+
```

```
SELECT named_struct('a', 1, 'b', 2) in(named_struct('a', 1, 'b', 1), named_struct('a',
 1, 'b', 3));
+-----------------------------------------------------------------------------------+
|(named_struct(a, 1, b, 2) IN (named_struct(a, 1, b, 1), named_struct(a, 1, b, 3)))|
+-----------------------------------------------------------------------------------+
|                                                                              false|
+-----------------------------------------------------------------------------------+
SELECT named_struct('a', 1, 'b', 2) in(named_struct('a', 1, 'b', 2), named_struct('a',
 1, 'b', 3));
+-----------------------------------------------------------------------------------+
|(named_struct(a, 1, b, 2) IN (named_struct(a, 1, b, 2), named_struct(a, 1, b, 3)))|
+-----------------------------------------------------------------------------------+
|                                                                               true|
+-----------------------------------------------------------------------------------+
-- isnan
SELECT isnan(cast('NaN' as double));
+--------------------------+
|isnan(CAST(NaN AS DOUBLE))|
+--------------------------+
|                      true|
+--------------------------+
-- isnotnull
SELECT isnotnull(1);
+---------------+
|(1 IS NOT NULL)|
+---------------+
|           true|
+---------------+
-- isnull
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|
+---------------+
```

맵 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
element_at(배열, 인덱스)	지정된 (1 기반) 인덱스에서 배열의 요소를 반환합니다.
element_at(맵, 키)	지정된 키의 값을 반환합니다. 키가 맵에 포함되지 않은 경우 함수는 NULL을 반환합니다.
map(key0, value0, key1, value1, ...)	지정된 키/값 페어로 맵을 생성합니다.
map_concat(map, ...)	지정된 모든 맵의 결합을 반환합니다.
map_contains_key(맵, 키)	맵에 키가 포함된 경우 true를 반환합니다.
map_entries(맵)	지정된 맵에 있는 모든 항목의 정렬되지 않은 배열을 반환합니다.
map_from_array(키, 값)	지정된 키/값 배열 쌍으로 맵을 생성합니다. 키의 모든 요소는 null이 아니어야 합니다.
map_from_entries(arrayOfEntries)	지정된 항목 배열에서 생성된 맵을 반환합니다.
map_keys(맵)	맵의 키가 포함된 정렬되지 않은 배열을 반환합니다.
map_values(맵)	맵 값을 포함하는 정렬되지 않은 배열을 반환합니다.
str_to_map(text[, pairDelim[, keyValueDelim]])	구분 기호를 사용하여 텍스트를 키/값 페어로 분할한 후 맵을 생성합니다. 기본 구분 기호는 'pairDelim'의 경우 ', 'keyValueDelim'의 경우

함수	설명
	':'입니다. `pairDelim`과 `keyValueDelim`은 모두 정규식으로 처리됩니다.
try_element_at(배열, 인덱스)	지정된 (1 기반) 인덱스에서 배열의 요소를 반환합니다. 인덱스가 0이면 시스템에서 오류가 발생합니다. 인덱스가 < 0인 경우는 마지막에서 첫 번째 까지 요소에 액세스합니다. 인덱스가 배열 길이를 초과하는 경우 함수는 항상 NULL을 반환합니다.
try_element_at(맵, 키)	지정된 키의 값을 반환합니다. 키가 맵에 포함되지 않은 경우 함수는 항상 NULL을 반환합니다.

예시

```
-- 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|
+---------------------------------+
```

수학 함수

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
expr1 % expr2	`expr1`/`expr2` 뒤에 나머지를 반환합니다.
expr1 * expr2	`expr1`*`expr2`를 반환합니다.
expr1 + expr2	`expr1`+`expr2`를 반환합니다.
expr1 - expr2	`expr1`-`expr2`를 반환합니다.
expr1/expr2	`expr1`/`expr2`를 반환합니다. 항상 부동 소수점 분할을 수행합니다.
abs(expr)	숫자 또는 간격 값의 절대값을 반환합니다.
acos(expr)	`java.lang.Math.acos`로 계산된 것처럼 `expr`의 역코사인(a.k.a. 아크 코사인)을 반환합니다.
acosh(expr)	`expr`의 역 하이퍼볼릭 코사인을 반환합니다.
asin(expr)	`java.lang.Math.asin`으로 계산되는 것처럼 `expr`의 아크 사인(a.k.a. 아크 사인)을 반환합니다.
asinh(expr)	`expr`의 역 하이퍼볼릭 사인을 반환합니다.
atan(expr)	`java.lang.Math.atan`으로 계산된 것처럼 `expr`의 역탄젠트(일명 arc tangent)를 반환합니다.
atan2(exprY, exprX)	`java.lang.Math.atan2`로 계산된 것처럼 평면의 양수 x축과 좌표(`exprX`, `exprY`)가 지정한 점 사이의 각도를 라디안 단위로 반환합니다.
atanh(expr)	`expr`의 역 하이퍼볼릭 탄젠트를 반환합니다.
bin(expr)	바이너리로 표시되는 긴 값 `expr`의 문자열 표현을 반환합니다.
bround(expr, d)	HALF_EVEN 반올림 모드를 사용하여 `d` 소수 자릿수로 반올림된 `expr`를 반환합니다.

함수	설명
cbrt(expr)	`expr`의 큐브 루트를 반환합니다.
ceil(expr[, scale])	반올림 후 `expr`보다 작지 않은 가장 작은 숫자를 반환합니다. 선택적 `scale` 파라미터를 지정하여 반올림 동작을 제어할 수 있습니다.
천장(expr[, scale])	반올림 후 `expr`보다 작지 않은 가장 작은 숫자를 반환합니다. 선택적 `scale` 파라미터를 지정하여 반올림 동작을 제어할 수 있습니다.
conv(num, from_base, to_base)	`from_base`에서 `to_base`로 `num`을 변환합니다.
cos(expr)	`java.lang.Math.cos`로 계산된 것처럼 `expr`의 코사인을 반환합니다.
cosh(expr)	`java.lang.Math.cosh`로 계산된 것처럼 `expr`의 하이퍼볼릭 코사인을 반환합니다.
cot(expr)	`1/java.lang.Math.tan`으로 계산된 것처럼 `expr`의 코탄젠트를 반환합니다.
csc(expr)	`1/java.lang.Math.sin`으로 계산되는 것처럼 `expr`의 코사컨트를 반환합니다.
도(expr)	라디안을 도로 변환합니다.
expr1 div expr2	`expr1`을 `expr2`로 나눕니다. 피연산자가 NULL이거나 `expr2`가 0인 경우 NULL을 반환합니다. 결과는 길게 캐스팅됩니다.
e()	Euler의 번호 e를 반환합니다.
exp(expr)	e를 `expr`의 출력으로 반환합니다.
expm1(expr) - exp(`expr`)를 반환합니다.	1

함수	설명
factorial(expr)	`expr`의 인수를 반환합니다. `expr`는 [0..20]입니다. 그렇지 않은 경우 null입니다.
floor(expr[, scale])	반올림 후 `expr`보다 크지 않은 가장 큰 숫자를 반환합니다. 선택적 `scale` 파라미터를 지정하여 반올림 동작을 제어할 수 있습니다.
가장 큼(expr, ...)	null 값을 건너뛰면서 모든 파라미터의 최대 값을 반환합니다.
16진수(expr)	`expr`를 16진수로 변환합니다.
hypot(expr1, expr2)	sqrt(`expr1`**2 + `expr2`**2)를 반환합니다.
least(expr, ...)	null 값을 건너뛰면서 모든 파라미터의 최소 값을 반환합니다.
ln(expr)	`expr`의 자연 로그(베이스 e)를 반환합니다.
로그(기본, expr)	`base`와 함께 `expr`의 로그를 반환합니다.
log10(expr)	기본값이 10인 `expr`의 로그를 반환합니다.
log1p(expr)	log(1 + `expr`)를 반환합니다.
log2(expr)	기본값이 2인 `expr`의 로그를 반환합니다.
expr1 mod expr2	`expr1`/`expr2` 뒤에 나머지를 반환합니다.
negative(expr)	`expr`의 부정 값을 반환합니다.
pi()	pi를 반환합니다.
pmod(expr1, expr2)	`expr1` mod `expr2`의 양수 값을 반환합니다.
positive(expr)	`expr`의 값을 반환합니다.
pow(expr1, expr2)	`expr1`을 `expr2`의 출력으로 높입니다.

함수	설명
power(expr1, expr2)	`expr1`을 `expr2`의 출력으로 높입니다.
라디안(expr)	각도를 라디안으로 변환합니다.
rand([seed])	독립적이고 동일하게 분산된(i.i.d.) 값이 균일하게 분산된 임의 값을 [0, 1)로 반환합니다.
randn([seed])	표준 정규 분포에서 가져온 독립적이고 동일하게 분산된(i.i.d.) 값이 있는 임의 값을 반환합니다.
random([seed])	독립적이고 동일하게 분산된(i.i.d.) 값이 균일하게 분산된 임의 값을 [0, 1)로 반환합니다.
rint(expr)	인수에 대한 값에서 가장 가깝고 수학 정수와 동일한 이중 값을 반환합니다.
round(expr, d)	HALF_UP 반올림 모드를 사용하여 `d` 소수 자릿수로 반올림된 `expr`를 반환합니다.
sec(expr)	`1/java.lang.Math.cos`로 계산된 것처럼 `expr`의 시컨트를 반환합니다.
shiftleft(기본, expr)	비트 단위 왼쪽 이동.
sign(expr)	`expr`이 음수, 0 또는 양수이므로 -1.0, 0.0 또는 1.0을 반환합니다.
signum(expr)	`expr`이 음수, 0 또는 양수이므로 -1.0, 0.0 또는 1.0을 반환합니다.
sin(expr)	`java.lang.Math.sin`으로 계산된 것처럼 `expr`의 사인을 반환합니다.
sinh(expr)	`java.lang.Math.sinh`로 계산된 것처럼 `expr`의 하이퍼볼릭 사인을 반환합니다.
sqrt(expr)	`expr`의 제곱근을 반환합니다.

함수	설명
tan(expr)	`java.lang.Math.tan`으로 계산된 것처럼 `expr`의 탄젠트를 반환합니다.
tanh(expr)	`java.lang.Math.tanh`로 계산된 것처럼 `expr`의 하이퍼볼릭 탄젠트를 반환합니다.
try_add(expr1, expr2)	`expr1`과 `expr2`의 합계를 반환하고 오버플로 시 결과는 null입니다. 허용되는 입력 유형은 `+` 연산자와 동일합니다.
try_divide(dividend, divisor)	`dividend`/`divisor`를 반환합니다. 항상 부동 소수점 분할을 수행합니다. `expr2`가 0인 경우 결과는 항상 null입니다. `dividend`는 숫자 또는 간격이어야 합니다. `divisor`는 숫자여야 합니다.
try_multiply(expr1, expr2)	`expr1`*`expr2`를 반환하고 오버플로 시 결과가 null입니다. 허용되는 입력 유형은 `*` 연산자와 동일합니다.
try_subtract(expr1, expr2)	`expr1`-`expr2`를 반환하고 오버플로 시 결과가 null입니다. 허용되는 입력 유형은 `-` 연산자와 동일합니다.
unhex(expr)	16진수 `expr`를 바이너리로 변환합니다.
width_bucket(값, min_value, max_value, num_bucket)	`min_value`~`max_value` 범위의 `num_bucket` 버킷이 있는 등가 히스토그램에서 `value`가 할당될 버킷 번호를 반환합니다.

예시

```
-- %
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|
```

```
+----------------------------------------------------------------------+
```

생성기 함수

> **Note**
>
> 이러한 SQL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

함수	설명
explode(expr)	배열 `expr`의 요소를 여러 행으로 구분하거나 `expr`의 요소를 여러 행과 열로 매핑합니다. 달리 지정하지 않는 한는 배열 요소에 기본 열 이름 `col`을 사용하거나 맵 요소에 `key` 및 `value`를 사용합니다.
explode_outer(expr)	배열 `expr`의 요소를 여러 행으로 구분하거나 `expr`의 요소를 여러 행과 열로 매핑합니다. 달리 지정하지 않는 한는 배열 요소에 기본 열 이름 `col`을 사용하거나 맵 요소에 `key` 및 `value`를 사용합니다.
인라인(expr)	구조체 배열을 테이블로 탐색합니다. 달리 지정하지 않는 한 열 이름 col1, col2 등을 기본적으로 사용합니다.
inline_outer(expr)	구조체 배열을 테이블로 탐색합니다. 달리 지정하지 않는 한 열 이름 col1, col2 등을 기본적으로 사용합니다.
posexplode(expr)	배열 `expr`의 요소를 위치가 있는 여러 행으로 구분하거나 `expr`의 요소를 위치가 있는 여러 행과 열로 구분합니다. 달리 지정하지 않는 한는 위치의 열 이름 `pos`, 배열의 요소에 `col` 또는 맵의 요소에 `key` 및 `value`를 사용합니다.
posexplode_outer(expr)	배열 `expr`의 요소를 위치가 있는 여러 행으로 구분하거나 `expr`의 요소를 위치가 있는 여러 행과 열로 구분합니다. 달리 지정하지 않는 한는 위치의 열 이름 `pos`, 배열의 요소에 `col` 또는 맵의 요소에 `key` 및 `value`를 사용합니다.
스택(n, expr1, ..., exprk)	`expr1`, ..., `exprk`를 `n` 행으로 구분합니다. 달리 지정하지 않는 한 기본적으로 열 이름 col0, col1 등을 사용합니다.

예시

```
-- 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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

OpenSearch SQL은 하나 이상의 테이블에서 결과 세트를 검색하는 데 사용되는 SELECT 문을 지원합니다. 다음 섹션에서는 전체 쿼리 구문과 쿼리의 다양한 구문에 대해 설명합니다.

구문

```
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 } ]
```

select_statement는 다음과 같이 정의됩니다.

```
SELECT [ ALL | DISTINCT ] { [ [ named_expression ] [ , ... ] ] }
FROM { from_item [ , ... ] }
[ PIVOT clause ]
[ UNPIVOT clause ]
[ LATERAL VIEW clause ] [ ... ]
[ WHERE boolean_expression ]
[ GROUP BY expression [ , ... ] ]
[ HAVING boolean_expression ]
```

파라미터

- 모두

  관계에서 일치하는 모든 행을 선택하고 기본적으로 활성화됩니다.
- DISTINCT

  결과에서 중복 항목을 제거한 후 관계에서 일치하는 모든 행을 선택합니다.
- named_expression

  이름이 할당된 표현식입니다. 일반적으로 열 표현식을 나타냅니다.

  구문: `expression [[AS] alias]`
- from_item

  테이블 관계

  조인 관계

  피벗 관계

피벗 해제 관계

테이블-값 함수

인라인 테이블

[LATERAL] (Subquery)

- PIVOT

  PIVOT 절은 데이터 관점에 사용됩니다. 특정 열 값을 기반으로 집계된 값을 가져올 수 있습니다.

- UNPIVOT

  UNPIVOT 절은 열을 행으로 변환합니다. 값의 집계를 PIVOT제외하고의 반대입니다.

- LATERAL VIEW

  LATERAL VIEW 절은 하나 이상의 행EXPLODE이 포함된 가상 테이블을 생성하는와 같은 생성기 함수와 함께 사용됩니다.

  LATERAL VIEW는 각 원래 출력 행에 행을 적용합니다.

- WHERE

  제공된 조건자를 기반으로 FROM 절의 결과를 필터링합니다.

- 그룹화 기준

  행을 그룹화하는 데 사용되는 표현식을 지정합니다.

  이는 집계 함수(MIN, , MAX, COUNTSUM, AVG등)와 함께 사용되어 각 그룹의 그룹화 표현식 및 집계 값을 기반으로 행을 그룹화합니다.

  FILTER 절이 집계 함수에 연결되면 일치하는 행만 해당 함수에 전달됩니다.

- 보유

  에서 생성된 행이 GROUP BY 필터링되는 조건자를 지정합니다.

  HAVING 절은 그룹화가 수행된 후 행을 필터링하는 데 사용됩니다.

  없이 HAVING를 지정하면 그룹화 표현식(글로벌 집계)이 GROUP BY 없는 GROUP BY를 나타냅니다.

- 순서 기준

쿼리의 전체 결과 집합 행 순서를 지정합니다.

출력 행은 파티션에 걸쳐 정렬됩니다.

이 파라미터는 SORT BY 및와 상호 배타DISTRIBUTE BY적이며 함께 지정할 수 없습니다.

- SORT BY

  각 파티션 내에서 행이 정렬되는 순서를 지정합니다.

  이 파라미터는 ORDER BY와 상호 배타적이며 함께 지정할 수 없습니다.

- LIMIT

  문 또는 하위 쿼리에서 반환할 수 있는 최대 행 수를 지정합니다.

  이 절은 대부분와 함께 결정적 결과를 생성하는 ORDER BY 데 사용됩니다.

- 부울 표현식

  결과 유형 부울로 평가되는 표현식을 지정합니다.

  논리 연산자(,)를 사용하여 둘 이상의 표현식을 결합할 수 있습니다ANDOR.

- expression

  값으로 평가되는 하나 이상의 값, 연산자 및 SQL 함수의 조합을 지정합니다.

- named_window

  하나 이상의 소스 창 사양에 대한 별칭을 지정합니다.

  소스 창 사양은 쿼리의 창 정의에서 참조할 수 있습니다.

WHERE 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

WHERE 절은 지정된 조건에 따라 쿼리 또는 하위 쿼리의 FROM 절 결과를 제한하는 데 사용됩니다.

구문

```
WHERE boolean_expression
```

파라미터

- 부울 표현식

  결과 유형 부울로 평가되는 표현식을 지정합니다.

  논리 연산자(,)를 사용하여 둘 이상의 표현식을 결합할 수 있습니다ANDOR.

예시

```
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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

GROUP BY 절은 지정된 그룹화 표현식 세트를 기반으로 행을 그룹화하고 하나 이상의 지정된 집계 함수를 기반으로 행 그룹에 대한 집계를 계산하는 데 사용됩니다.

또한 시스템은 GROUPING SETS, CUBE, ROLLUP 절을 통해 동일한 입력 레코드 세트에 대해 여러 집계를 수행합니다. 그룹화 표현식과 고급 집계는 절에서 혼합하고 GROUP BY 절에서 중첩할 수 있습니다GROUPING SETS . 자세한 내용은 Mixed/Nested Grouping Analytics 단원을 참조하십시오.

FILTER 절이 집계 함수에 연결되면 일치하는 행만 해당 함수에 전달됩니다.

구문

```
GROUP BY group_expression [ , group_expression [ , ... ] ] [ WITH { ROLLUP | CUBE } ]
GROUP BY { group_expression | { ROLLUP | CUBE | GROUPING SETS } (grouping_set
 [ , ...]) } [ , ... ]
```

집계 함수는 다음과 같이 정의됩니다.

```
aggregate_name ( [ DISTINCT ] expression [ , ... ] ) [ FILTER ( WHERE
 boolean_expression ) ]
```

파라미터

- group_expression

  행이 함께 그룹화되는 기준에 따라 기준을 지정합니다. 행 그룹화는 그룹화 표현식의 결과 값을 기반으로 수행됩니다.

  그룹화 표현식은와 같은 열 이름GROUP BY a,와 같은 열 위치 GROUP BY 0또는와 같은 표현식일 수 있습니다GROUP BY a + b.
- 그룹화_설정

그룹화 세트는 괄호 안에 0개 이상의 쉼표로 구분된 표현식으로 지정됩니다. 그룹화 세트에 요소가 하나만 있는 경우 괄호를 생략할 수 있습니다.

예를 들어 `GROUPING SETS ((a), (b))`는 `GROUPING SETS (a, b)`와 동일합니다.

구문: `{ ( [ expression [ , ... ] ] ) | expression }`

- 그룹화 세트

  뒤에 지정된 각 그룹화 세트의 행을 그룹화합니다`GROUPING SETS`.

  예를 들어, `GROUP BY GROUPING SETS ((warehouse), (product))`는 의미상 `GROUP BY warehouse` 및의 결과 조합과 동일합니다`GROUP BY product`. 이 절은 연`UNION ALL`산자의 각 레그가 `GROUPING SETS` 절에서 지정된 각 그룹화 세트의 집계를 수행하는 UNION ALL의 약어입니다.

  마찬가지로 `GROUP BY GROUPING SETS ((warehouse, product), (product), ())`는 의미상 및 전역 집계의 결과 조합`GROUP BY warehouse, product, GROUP BY product`과 동일합니다.

- ROLLUP

  단일 문에서 여러 수준의 집계를 지정합니다. 이 절은 여러 그룹화 세트를 기반으로 집계를 계산하는 데 사용됩니다. `ROLLUP`는의 약어입니다`GROUPING SETS`.

  예를 들어, `GROUP BY warehouse, product WITH ROLLUP` or `GROUP BY ROLLUP(warehouse, product)`은 `GROUP BY GROUPING SETS((warehouse, product), (warehouse), ())`과 같습니다.

  `GROUP BY ROLLUP(warehouse, product, (warehouse, location))`는와 동일합니다`GROUP BY GROUPING SETS((warehouse, product, location), (warehouse, product), (warehouse), ())`.

  ROLLUP 사양의 N 요소는 N+1 GROUPING SETS가 됩니다.

- CUBE

  CUBE 절은 GROUP BY 절에 지정된 그룹화 열의 조합을 기반으로 집계를 수행하는 데 사용됩니다. CUBE는 GROUPING SETS의 약어입니다.

예를 들어, GROUP BY warehouse, product WITH CUBE or GROUP BY CUBE(warehouse, product)은 GROUP BY GROUPING SETS((warehouse, product), (warehouse), (product), ())과 같습니다.

GROUP BY CUBE(warehouse, product, (warehouse, location))는와 동일합니다GROUP BY GROUPING SETS((warehouse, product, location), (warehouse, product), (warehouse, location), (product, warehouse, location), (warehouse), (product), (warehouse, product), ()). CUBE 사양의 N 요소는 2^N이 됩니다GROUPING SETS.

- 혼합/중첩 그룹화 분석

  GROUP BY 절에는 여러 group_expressions 및 여러가 포함될 수 있습니다CUBE|ROLLUP|GROUPING SETS. 에는 , GROUPING SETS(ROLLUP(warehouse, location),와 같은 중첩된 CUBE|ROLLUP|GROUPING SETS 절도 있을 GROUPING SETS 수 있습니다CUBE(warehouse, location))GROUPING SETS(warehouse, GROUPING SETS(location, GROUPING SETS(ROLLUP(warehouse, location),CUBE(warehouse, location)))).

  CUBE|ROLLUP는에 대한 구문 설탕일 뿐입니다GROUPING SETS. 로 변환하는 방법은 위의 섹션을 참조CUBE|ROLLUP하세요GROUPING SETS.는이 컨텍스트GROUPING SETS에서 단일 그룹으로 취급될 group_expression 수 있습니다.

  GROUP BY 절GROUPING SETS의 여러에 대해 원래의 교차 곱을 GROUPING SETS 수행하여 단일를 생성합니다GROUPING SETS. GROUPING SETS 절GROUPING SETS에서 중첩된의 경우 그룹화 세트를 가져와 제거합니다.

  예를 들어, GROUP BY warehouse, GROUPING SETS((product), ()), GROUPING SETS((location, size), (location), (size), ()) and GROUP BY warehouse, ROLLUP(product), CUBE(location, size)은 GROUP BY GROUPING SETS((warehouse, product, location, size), (warehouse, product, location), (warehouse, product, size), (warehouse, product), (warehouse, location, size), (warehouse, location), (warehouse, size), (warehouse))과 같습니다.

  GROUP BY GROUPING SETS(GROUPING SETS(warehouse), GROUPING SETS((warehouse, product)))는와 동일합니다GROUP BY GROUPING SETS((warehouse), (warehouse, product)).

- aggregate_name

집계 함수 이름(MIN, MAX, COUNT, SUM, AVG등)을 지정합니다.

- DISTINCT

  집계 함수로 전달되기 전에 입력 행의 중복을 제거합니다.

- 필터

  boolean_expression WHERE 절의가 true로 평가되는 입력 행을 필터링하여 집계 함수로 전달하고 다른 행은 삭제됩니다.

예시

```
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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

HAVING 절은 지정된 조건을 GROUP BY 기반으로에서 생성된 결과를 필터링하는 데 사용됩니다. 절과 함께 사용되는 경우가 많습니다GROUP BY.

구문

```
HAVING boolean_expression
```

파라미터

- 부울 표현식

  결과 유형 부울로 평가되는 표현식을 지정합니다. 논리 연산자(,)를 사용하여 둘 이상의 표현식을 결합할 수 있습니다ANDOR.

  참고 HAVING 절에서 지정된 표현식은 다음 항목만 참조할 수 있습니다.

  1. 상수
  2. 에 표시되는 표현식 GROUP BY
  3. 집계 함수

예시

```
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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

ORDER BY 절은 사용자 지정 순서로 정렬된 방식으로 결과 행을 반환하는 데 사용됩니다. SORT BY 절과 달리이 절은 출력의 총 순서를 보장합니다.

구문

```
ORDER BY { expression [ sort_direction | nulls_sort_order ] [ , ... ] }
```

파라미터

- 순서 기준

  행을 정렬nulls_sort_order하는 데 사용되는 선택적 파라미터 sort_direction 및와 함께 쉼표로 구분된 표현식 목록을 지정합니다.

- sort_direction

  선택적으로 행을 오름차순 또는 내림차순으로 정렬할지 여부를 지정합니다.

  정렬 방향의 유효한 값은 ASC 오름차순과 DESC 내림차순입니다.

정렬 방향이 명시적으로 지정되지 않은 경우 기본적으로 행은 오름차순으로 정렬됩니다.

구문: [ASC | DESC]

- nulls_sort_order

선택적으로 NULL이 아닌 NULL 값 이전/이후에 값을 반환할지 여부를 지정합니다.

null_sort_order를 지정하지 않은 경우 NULLs 정렬 순서가 인 경우 먼저 정렬ASC하고 정렬 순서가 인 경우 NULLS 정렬을 마지막으로 정렬합니다DESC.

1. NULLS FIRST를 지정하면 정렬 순서에 관계없이 NULL 값이 먼저 반환됩니다.

2. NULLS LAST를 지정하면 정렬 순서에 관계없이 NULL 값이 마지막으로 반환됩니다.

구문: [NULLS { FIRST | LAST }]

예시

```
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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

SQL 조인은 조인 기준에 따라 두 관계의 행을 결합하는 데 사용됩니다. 다음 섹션에서는 예제와 함께 전체 조인 구문과 다양한 유형의 조인에 대해 설명합니다.

구문

```
relation INNER JOIN relation [ join_criteria ]
```

파라미터

- 관계

  조인할 관계를 지정합니다.
- join_type

  조인 유형을 지정합니다.

  구문: `INNER | CROSS | LEFT OUTER`
- join_criteria

  한 관계의 행을 다른 관계의 행과 결합하는 방법을 지정합니다.

  구문: `ON boolean_expression | USING ( column_name [ , ... ] )`
- 부울 표현식

  반환 유형이 부울인 표현식을 지정합니다.

조인 유형

- 내부 조인

내부 조인을 명시적으로 지정해야 합니다. 두 관계에서 일치하는 값이 있는 행을 선택합니다.

구문: `relation INNER JOIN relation [ join_criteria ]`

- 왼쪽 조인

  왼쪽 조인은 왼쪽 관계의 모든 값과 오른쪽 관계의 일치하는 값을 반환하거나 일치하는 값이 없는 경우 NULL을 추가합니다. 이를 왼쪽 외부 조인이라고도 합니다.

  구문: `relation LEFT OUTER JOIN relation [ join_criteria ]`

- 교차 조인

  교차 조인은 두 관계의 데카르트 곱을 반환합니다.

  구문: `relation CROSS JOIN relation [ join_criteria ]`

예시

```
-- 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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

LIMIT 절은 SELECT 문에서 반환되는 행 수를 제한하는 데 사용됩니다. 일반적으로이 절은 결과가 결정적이 되도록 하기 ORDER BY 위해와 함께 사용됩니다.

구문

```
LIMIT { ALL | integer_expression }
```

파라미터

- 모두

  지정된 경우 쿼리는 모든 행을 반환합니다. 즉,이 옵션을 지정하면 제한이 적용되지 않습니다.

- integer_expression

  정수를 반환하는 폴더블 표현식을 지정합니다.

예시

```
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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

CASE 절은 규칙을 사용하여 다른 프로그래밍 언어의 if/else 문과 마찬가지로 지정된 조건을 기반으로 특정 결과를 반환합니다.

구문

```
CASE [ expression ] { WHEN boolean_expression THEN then_expression } [ ... ]
[ ELSE else_expression ]
END
```

파라미터

- 부울 표현식

  결과 유형 부울로 평가되는 표현식을 지정합니다.

  논리 연산자(,)를 사용하여 둘 이상의 표현식을 결합할 수 있습니다ANDOR.

- then_expression

  boolean_expression 조건을 기반으로 then 표현식을 지정합니다.

  then_expression 및는 모두 동일한 유형이거나 공통 유형과 강제적이어야 else_expression 합니다.

- else_expression

  기본 표현식을 지정합니다.

  then_expression 및 else_expression는 모두 동일한 유형이거나 공통 유형과 강제적이어야 합니다.

예시

```
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
                |
+------
+-----------------------------------------------------------------------------------------------------
+
```

공통 테이블 표현식

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

공통 테이블 표현식(CTE)은 사용자가 SQL 문의 범위 내에서 여러 번 참조할 수 있는 임시 결과 세트를 정의합니다. CTE는 주로 SELECT 문에 사용됩니다.

구문

```
WITH common_table_expression [ , ... ]
```

common_table_expression는 다음과 같이 정의됩니다.

```
Syntexpression_name [ ( column_name [ , ... ] ) ] [ AS ] ( query )
```

파라미터

- expression_name

  공통 테이블 표현식의 이름을 지정합니다.

- query

  SELECT 문입니다.

예시

```
-- 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**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

EXPLAIN 문은 입력 문에 대한 논리적/물리적 계획을 제공하는 데 사용됩니다. 기본적으로이 절은 물리적 계획에 대한 정보만 제공합니다.

구문

```
EXPLAIN [ EXTENDED | CODEGEN | COST | FORMATTED ] statement
```

파라미터

- 확장됨

구문 분석된 논리적 계획, 분석된 논리적 계획, 최적화된 논리적 계획 및 물리적 계획을 생성합니다.

구문 분석된 논리적 계획은 쿼리에서 추출한 해결되지 않은 계획입니다.

분석된 논리적 계획은 unresolvedAttribute 및를 완전한 형식의 객체unresolvedRelation로 변환합니다.

최적화된 논리적 계획은 최적화 규칙 세트를 통해 변환되므로 물리적 계획이 생성됩니다.

- CODEGEN

  및 물리적 계획이 있는 경우 문에 대한 코드를 생성합니다.

- 비용

  계획 노드 통계를 사용할 수 있는 경우는 논리적 계획과 통계를 생성합니다.

- 형식 지정됨

  물리적 계획 개요와 노드 세부 정보라는 두 섹션을 생성합니다.

- 설명

  설명할 SQL 문을 지정합니다.

예시

```
-- 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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

LATERAL SUBQUERY는 키워드 앞에 오는 하위 쿼리입니다LATERAL. 이전 FROM 절의 열을 참조하는 방법을 제공합니다. LATERAL 키워드가 없으면 하위 쿼리는 외부 쿼리의 열만 참조할 수 있지만 FROM 절에서는 참조할 수 없습니다.는 복잡한 쿼리를 더 간단하고 효율적으로 LATERAL SUBQUERY 만듭니다.

구문

```
[ LATERAL ] primary_relation [ join_relation ]
```

파라미터

- primary_relation

  기본 관계를 지정합니다. 다음 중 하나가 될 수 있습니다.

  1. 테이블 관계
  2. 별칭이 지정된 쿼리

     구문: `( query ) [ [ AS ] alias ]`
  3. 별칭이 지정된 관계

     `Syntax: ( relation ) [ [ AS ] alias ]`

예시

```
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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

`LATERAL VIEW` 절은 하나 이상의 행`EXPLODE`이 포함된 가상 테이블을 생성하는와 같은 생성기 함수와 함께 사용됩니다. `LATERAL VIEW`는 각 원래 출력 행에 행을 적용합니다.

구문

```
LATERAL VIEW [ OUTER ] generator_function ( expression [ , ... ] ) [ table_alias ] AS
 column_alias [ , ... ]
```

파라미터

- 외부

OUTER 지정된 경우 입력 배열/맵이 비어 있거나 null인 경우 null을 반환합니다.

- generator_function

  생성기 함수(EXPLODE, INLINE등)를 지정합니다.

- table_alias

  선택 사항generator_function인의 별칭입니다.

- column_별칭

  출력 행에 사용할 수 generator_function있는의 열 별칭을 나열합니다.

  에 출력 열이 여러 개 있는 경우 별칭generator_function을 여러 개 가질 수 있습니다.

예시

```
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 조건자

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

LIKE 조건자는 특정 패턴을 검색하는 데 사용됩니다. 이 조건자는 ANY, SOME및를 포함한 한정자가 있는 여러 패턴도 지원합니다ALL.

구문

```
[ NOT ] { LIKE search_pattern [ ESCAPE esc_char ] | [ RLIKE | REGEXP ] regex_pattern }
[ NOT ] { LIKE quantifiers ( search_pattern [ , ... ]) }
```

파라미터

- search_pattern

  LIKE 절로 검색할 문자열 패턴을 지정합니다. 다음과 같은 특수 패턴 일치 문자를 포함할 수 있습니다.

  - %는 0개 이상의 문자와 일치합니다.
  - _는 정확히 하나의 문자와 일치합니다.
- esc_char

  이스케이프 문자를 지정합니다. 기본 이스케이프 문자는 입니다\.
- regex_pattern

  RLIKE 또는 REGEXP 절로 검색할 정규식 검색 패턴을 지정합니다.
- 한정자

  ANY, SOME 및를 포함하는 조건자 한정자를 지정합니다ALL.

  ANY 또는는 패턴 중 하나가 입력과 일치하는 경우 true를 반환함을 SOME 의미합니다.

  ALL는 모든 패턴이 입력과 일치하는 경우 true를 반환함을 의미합니다.

예시

```
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**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

OFFSET 절은 SELECT 문에서 반환된 행을 반환하기 시작하기 전에 건너뛸 행 수를 지정하는 데 사용됩니다. 일반적으로이 절은 결과가 결정적이 되도록 하기 ORDER BY 위해와 함께 사용됩니다.

구문

```
OFFSET integer_expression
```

파라미터

- integer_expression

  정수를 반환하는 폴더블 표현식을 지정합니다.

예시

```
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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

PIVOT 절은 데이터 관점에 사용됩니다. 특정 열 값을 기반으로 집계된 값을 가져올 수 있으며, 이는 SELECT 절에서 사용되는 여러 열로 전환됩니다. PIVOT 절은 테이블 이름 또는 하위 쿼리 뒤에 지정할 수 있습니다.

구문

```
PIVOT ( { aggregate_expression [ AS aggregate_expression_alias ] } [ , ... ] FOR
 column_list IN ( expression_list ) )
```

파라미터

- aggregate_expression

집계 표현식 (SUM(a), COUNT(DISTINCT b)등을 지정합니다.)

- aggregate_expression_별칭

  집계 표현식의 별칭을 지정합니다.

- column_list

  FROM 절에 새 열로 바꿀 열을 지정하는 열이 포함되어 있습니다. 대괄호를 사용하여와 같은 열을 묶을 수 있습니다(c1, c2).

- expression_list

  의 값을 column_list 집계 조건으로 일치시키는 데 사용되는 새 열을 지정합니다. 별칭을 추가할 수도 있습니다.

예시

```
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  |
+------+-----------+-------+-------+-------+-------+
```

집합 연산자

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

집합 연산자는 두 입력 관계를 하나의 입력 관계로 결합하는 데 사용됩니다. OpenSearch SQL은 세 가지 유형의 집합 연산자를 지원합니다.

- EXCEPT 또는 MINUS
- INTERSECT
- UNION

입력 관계는 동일한 수의 열과 각 열에 대해 호환되는 데이터 형식을 가져야 합니다.

예외

EXCEPT 및는 한 관계에서 찾을 수 있지만 다른 관계에서는 찾을 수 없는 행을 EXCEPT ALL 반환합니다. EXCEPT (또는 EXCEPT DISTINCT)는 고유한 행만 가져오고 EXCEPT ALL는 결과 행에서 중복을 제거하지 않습니다. MINUS는의 별칭입니다EXCEPT.

구문

```
[ ( ] relation [ ) ] EXCEPT | MINUS [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

예시

```
-- 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 및는 두 관계에서 발견되는 행을 INTERSECT ALL 반환합니다. INTERSECT (또는 INTERSECT DISTINCT)는 고유한 행만 가져오고 INTERSECT ALL는 결과 행에서 중복을 제거하지 않습니다.

구문

```
 [ ( ] relation [ ) ] INTERSECT [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

예시

```
(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 및는 두 관계 중 하나에 있는 행을 UNION ALL 반환합니다. UNION (또는 UNION DISTINCT) 는 고유한 행만 가져오고 UNION ALL는 결과 행에서 중복을 제거하지 않습니다.

구문

```
[ ( ] relation [ ) ] UNION [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

예시

```
(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 절

> **Note**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

SORT BY 절은 각 파티션 내에서 사용자 지정 순서로 정렬된 결과 행을 반환하는 데 사용됩니다. 파티션이 두 개 이상 있는 경우 부분적으로 정렬된 결과를 반환할 SORT BY 수 있습니다. 이는 출력의 총 순서를 보장하는 ORDER BY 절과 다릅니다.

구문

```
SORT BY { expression [ sort_direction | nulls_sort_order ] [ , ... ] }
```

파라미터

- SORT BY

  각 파티션 내의 행을 정렬하는 데 사용되는 선택적 파라미터 sort_direction 및 nulls_sort_order와 함께 쉼표로 구분된 표현식 목록을 지정합니다.

- sort_direction

  선택적으로 행을 오름차순 또는 내림차순으로 정렬할지 여부를 지정합니다.

  정렬 방향의 유효한 값은 ASC 오름차순과 DESC 내림차순입니다.

  정렬 방향이 명시적으로 지정되지 않은 경우 기본적으로 행은 오름차순으로 정렬됩니다.

  구문: [ASC | DESC]

- nulls_sort_order

  선택적으로 NULL 값이 NULL이 아닌 값 이전/이후에 반환되는지 여부를 지정합니다.

  null_sort_order를 지정하지 않으면 정렬 순서가 NULLs이 먼저 정렬ASC되고 정렬 순서가 인 경우 NULLS 정렬이 마지막으로 정렬됩니다DESC.

1. NULLS FIRST를 지정하면 정렬 순서에 관계없이 NULL 값이 먼저 반환됩니다.

2. NULLS LAST를 지정하면 정렬 순서에 관계없이 NULL 값이 마지막으로 반환됩니다.

구문: [NULLS { FIRST | LAST }]

예시

```
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**
>
> 이 SQL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “지원되는 SQL 명령”.

UNPIVOT 절은 여러 열을 SELECT 절에서 사용되는 여러 행으로 변환합니다. UNPIVOT 절은 테이블 이름 또는 하위 쿼리 뒤에 지정할 수 있습니다.

구문

```
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] [, ...])
```

파라미터

- unpivot_column

  피벗을 해제하려는 열을 지정하는 FROM 절의 열을 포함합니다.

- name_열

  피벗되지 않은 열의 이름을 포함하는 열의 이름입니다.

- values_column

  피벗되지 않은 열의 값을 포함하는 열의 이름입니다.

예시

```
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           |
+------+------------------+---------------+----------------+
```

지원되는 PPL 명령

다음 표에는 OpenSearch Dashboards가 CloudWatch Logs, Amazon S3 또는 Security Lake 쿼리를 지원하는 PPL 명령과 CloudWatch Logs Insights가 지원하는 명령이 나와 있습니다. CloudWatch Logs Insights는 CloudWatch Logs를 쿼리할 때 OpenSearch Dashboards와 동일한 PPL 구문을 사용하며, 테이블은 둘 다 CloudWatch Logs라고 합니다.

> **Note**
>
> OpenSearch Service 외부에서 데이터를 분석하면 명령이 OpenSearch 인덱스와 다르게 실행될 수 있습니다.

주제

- 명령
- 함수
- OpenSearch PPL을 사용하는 CloudWatch Logs Insights 사용자를 위한 추가 정보

명령

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “필드”	프로젝션이 필요한 필드 세트를 표시합니다.	지원됨	지원됨	지원됨	`fields field1, field2`
the section called “여기서 각 항목은 다음과 같습니다.”	지정한 조건에 따라 데이터를 필터링합니다.	지원됨	지원됨	지원됨	`where field1="success" | where field2 != "i-023fe0a90929d8822" | fields field3,`

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
					`col4, col5, col6 | head 1000`
the section called “stats”	집계 및 계산을 수행합니다.	지원됨	지원됨	지원됨	`stats count(), count(`field1`), min(`field1`), max(`field1`), avg(`field1`) by field2 | head 1000`
the section called “parse”	문자열에서 정규식(정규식) 패턴을 추출하고 추출된 패턴을 표시합니다. 추출된 패턴을 사용하여 새 필드를 생성하거나 데이터를 필터링할 수 있습니다.	지원됨	지원됨	지원됨	`parse `field1` ".*/(?<field2>[^/]+$)" | where field2 = "requestId" | fields field2, `field2` | head 1000`

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “패턴”	텍스트 필드에서 로그 패턴을 추출하고 검색 결과에 결과를 추가합니다. 로그를 패턴별로 그룹화하면 분석 및 문제 해결을 위해 대량의 로그 데이터에서 통계를 더 쉽게 집계할 수 있습니다.	지원되지 않음	지원됨	지원됨	`patterns new_field='no_numbers' pattern='[0-9]' message | fields message, no_numbers`
the section called “정렬”	필드 이름을 기준으로 표시된 결과를 정렬합니다. sort -FieldName을 사용하여 내림차순으로 정렬합니다.	지원됨	지원됨	지원됨	`stats count(), count(`field1`), min(`field1`) as field1Alias, max(`field1`), avg(`field1`) by field2 | sort -field1Alias | head 1000`

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “평가”	필드 값을 수정 또는 처리하고 다른 필드에 저장합니다. 이는 열을 수학적으로 수정하거나, 열에 문자열 함수를 적용하거나, 열에 날짜 함수를 적용하는 데 유용합니다.	지원됨	지원됨	지원됨	`eval field2 = `field1` * 2 | fields field1, field2 | head 20`
the section called “이름 바꾸기”	검색 결과에서 하나 이상의 필드 이름을 바꿉니다.	지원됨	지원됨	지원됨	`rename field2 as field1 | fields field1`
the section called “head”	표시된 쿼리 결과를 frst N 행으로 제한합니다.	지원됨	지원됨	지원됨	`fields `@message` | head 20`
the section called “grok”	정규식을 기반으로 grok 패턴으로 텍스트 필드를 구문 분석하고 검색 결과에 결과를 추가합니다.	지원됨	지원됨	지원됨	`grok email '.+@%{HOSTNAME:host}' | fields email`

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “top”	필드의 가장 빈번한 값을 찾습니다.	지원됨	지원됨	지원됨	`top 2 Field1 by Field2`
the section called “dedup”	지정한 필드를 기반으로 중복 항목을 제거합니다.	지원됨	지원됨	지원됨	`dedup field1 | fields field1, field2, field3`
the section called “join”	두 데이터 세트를 함께 조인합니다.	지원됨	지원됨	지원됨	`source=customer | join ON c_custkey = o_custkey orders | head 10`

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “조회”	조회 인덱스(차원 테이블)의 데이터를 추가하거나 대체하여 검색 데이터를 강화합니다. 인덱스의 필드를 차원 테이블의 값으로 확장하거나 조회 조건이 일치할 때 값을 추가하거나 바꿀 수 있습니다.	지원되지 않음	지원됨	지원됨	`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 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “subquery”	파이프 처리 언어(PPL) 문 내에서 복잡한 중첩 쿼리를 수행합니다.	지원됨	지원됨	지원됨	`where id in [ subquery source=users | where user in [ subquery source=actions | where action="login" | fields user ] | fields uid ]`
the section called “드물게”	필드 목록에 있는 모든 필드의 가장 빈도가 낮은 값을 찾습니다.	지원됨	지원됨	지원됨	`rare Field1 by Field2`
the section called “추세선”	필드의 이동 평균을 계산합니다.	지원됨	지원됨	지원됨	`trendline sma(2, field1) as field1Alias`

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “eventstats”	계산된 요약 통계로 이벤트 데이터를 강화합니다. 이벤트 내에서 지정된 필드를 분석하고 다양한 통계 측정값을 계산한 다음 이러한 결과를 각 원래 이벤트에 새 필드로 추가합니다.	지원됨(제외count( ))	지원됨	지원됨	`eventstats sum(field1) by field2`
the section called “flatten”	필드를 평면화합니다. 필드는 다음 유형이어야 합니다. `struct<?,?> or array<struct<?,?>>`	지원됨	지원됨	지원됨	`source=table | flatten field1`
the section called “fieldsummary”	각 필드(개수, 고유 개수, 최소, 최대, 평균, stddev 및 평균)에 대한 기본 통계를 계산합니다.	지원됨(쿼리당 필드 1개)	지원됨	지원됨	`where field1 != 200 | fieldsummary includefields=field1 nulls=true`

PPL 명령	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “fillnull”	입력한 값으로 null 필드를 채웁니다. 하나 이상의 필드에서 사용할 수 있습니다.	지원됨	지원됨	지원됨	`fields field1 | eval field2=field1 | fillnull value=0 field1`
the section called “확장”	여러 값이 포함된 필드를 별도의 행으로 분할하여 지정된 필드의 각 값에 대해 새 행을 생성합니다.	지원됨	지원됨	지원됨	`expand employee | stats max(salary) as max by state, company`
the section called “describe”	테이블, 스키마 및 카탈로그의 구조 및 메타데이터에 대한 자세한 정보를 가져옵니다.	지원되지 않음	지원됨	지원됨	`describe schema.table`

함수

PPL 함수	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “String” (CONCAT, CONCAT_WS ,	PPL 쿼리 내에서 문자열 및 텍스트 데이터를 조작	지원됨	지원됨	지원됨	`eval col1Len = LENGTH(col1) | fields col1Len`

PPL 함수	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
LENGTH, LOWER, LTRIM, POSITION, REVERSE, RIGHT, RTRIM, SUBSTRING, TRIM, UPPER)	하고 변환할 수 있는 PPL의 내장 함수입니다. 예를 들어 변환 사례, 문자열 결합, 부분 추출, 텍스트 정리 등이 있습니다.				

PPL 함수	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “날짜 및 시간” (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,	PPL 쿼리에서 날짜 및 타임스탬프 데이터를 처리하고 변환하기 위한 내장 함수입니다. 예: date_add, date_format, datediff, current_date.	지원됨	지원됨	지원됨	`eval newDate = ADDDATE(DATE('2020-08-26'), 1) | fields newDate`

PPL 함수	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
DATE_SUB, TIMESTAMP ADD , TIMESTAMP DIFF , UTC_TIMESTAMP , CURRENT_TIMEZONE)					
the section called “Condition” (EXISTS, IF, IFNULL, ISNOTNULL , ISNULL, NULLIF)	여러 행에 대해 계산을 수행하여 단일 요약 값을 생성하는 내장 함수입니다. 예를 들어 합계, 개수, 평균, 최대 및 최소입니다.	지원됨	지원됨	지원됨	`eval field2 = isnull(col1) | fields field2, col1, field3`

PPL 함수	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “수학 연산” (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)	PPL 쿼리에서 수학 계산 및 변환을 수행하기 위한 내장 함수입니다. 예: abs(절대값), round(숫자 반올림), sqrt(제곱근), pow(검정력 계산), ceil(가장 가까운 정수로 반올림).	지원됨	지원됨	지원됨	`eval field2 = ACOS(col1) | fields col1`
the section called “Expressions” (산술 연산자(+, -, *), 조건자 연산자 (>. <, IN))	표현식, 특히 값 표현식에 대한 내장 함수는 스칼라 값을 반환합니다. 표현식의 유형과 형식은 다릅니다.	지원됨	지원됨	지원됨	`where age > (25 + 5) | fields age`

PPL 함수	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “IP 주소” (CIDRMATCH)	CIDR과 같은 IP 주소를 처리하기 위한 내장 함수입니다.	지원됨	지원됨	지원됨	`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)	배열, 추출 및 검증을 포함하여 JSON을 처리하기 위한 내장 함수입니다.	지원됨	지원됨	지원됨	`eval` `json_extract('{"a":"b"}', '$.a')` = `json_extract('{"a":"b"}', '$a')`
the section called “Lambda” (EXISTS, FILTER, REDUCE, TRANSFORM)	배열, 추출 및 검증을 포함하여 JSON을 처리하기 위한 내장 함수입니다.	지원되지 않음	지원됨	지원됨	`eval array = json_array(1, -1, 2), result = filter(array, x -> x > 0) | fields result`

PPL 함수	설명	CloudWatch Logs	Amazon S3	Security Lake	명령 예제:
the section called “암호화” (MD5, SHA1, SHA2)	검증, 비교 또는 더 복잡한 보안 프로토콜의 일부로 사용할 수 있는 고유한 데이터 지문을 생성할 수 있는 내장 함수입니다.	지원됨	지원됨	지원됨	`eval `MD5('hello')` = MD5('hello') | fields `MD5('hello')``

OpenSearch PPL을 사용하는 CloudWatch Logs Insights 사용자를 위한 추가 정보

CloudWatch Logs Insights는 대부분의 OpenSearch PPL 명령 및 함수를 지원하지만 일부 명령 및 함수는 현재 지원되지 않습니다. 예를 들어 현재 PPL에서 조회 명령을 지원하지 않습니다. 2025년 6월 2일부터 CloudWatch Logs Insights는 이제 PPL에서 JOIN, 하위 쿼리, Flatten, Fillnull, Expand, Cidrmatch 및 JSON 함수를 지원합니다. 지원되는 쿼리 명령 및 함수의 전체 목록은 위 표의 Amazon CloudWatch Logs 열을 참조하세요.

샘플 쿼리 및 할당량

다음은 CloudWatch 데이터를 쿼리하는 CloudWatch Logs Insights 사용자와 OpenSearch 사용자 모두에게 적용됩니다.

OpenSearch Service에서 CloudWatch Logs를 쿼리할 때 적용되는 제한에 대한 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs 할당량을 참조하세요. Amazon CloudWatch 제한에는 쿼리할 수 있는 CloudWatch Log 그룹 수, 실행할 수 있는 최대 동시 쿼리 수, 최대 쿼리 실행 시간 및 결과에 반환된 최대 행 수가 포함됩니다. CloudWatch Logs를 쿼리하는 데 사용하는 언어(즉, OpenSearch PPL, SQL 및 Logs Insights QL)에 관계없이 제한은 동일합니다.

PPL 명령

주제

- 설명
- 상관 관계 명령
- dedup 명령
- describe 명령
- eval 명령
- eventstats 명령
- 명령 확장
- 설명 명령
- fillnull 명령
- 필드 명령
- flatten 명령
- grok 명령
- 헤드 명령
- 조인 명령
- 조회 명령
- 구문 분석 명령
- 패턴 명령
- 희귀 명령
- 이름 바꾸기 명령
- 검색 명령
- sort 명령
- stats 명령
- 하위 쿼리 명령
- 상위 명령
- 추세선 명령
- 여기서 명령은
- 필드 요약
- 명령 확장
- PPL 함수

설명

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

PPL은 줄 주석과 블록 주석을 모두 지원합니다. 시스템은 주석 텍스트를 평가하지 않습니다.

줄 설명

줄 주석은 두 개의 슬래시 //로 시작하고 새 줄로 끝납니다.

예시

```
os> source=accounts | top gender // finds most common gender of all the accounts
fetched rows / total rows = 2/2
+----------+
| gender   |
|----------|
| M        |
| F        |
+----------+
```

블록 설명

블록 주석은 슬래시 뒤에 별표 *로 시작하고, 별표 뒤에 슬래시 */로 끝납니다.

예시

```
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        |
+------------------+----------+
```

상관 관계 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

일반적인 차원과 기간에 따라 다양한 데이터 소스를 상호 연관시킬 수 있습니다.

이 상관관계는 동일한 기간을 공유하지만 공식적으로 동기화되지 않은 다양한 수직 요소의 대량 데이터를 처리할 때 매우 중요합니다.

이러한 다양한 데이터 소스를 기간 및 유사한 차원에 따라 상호 연관시켜 데이터를 보강하고 중요한 인사이트를 발견할 수 있습니다.

예시

관찰성 도메인에는 세 가지 개별 데이터 소스가 있습니다.

- 로그
- Metrics
- 트레이스

이러한 데이터 소스는 공통 차원을 공유할 수 있습니다. 한 데이터 소스에서 다른 데이터 소스로 전환하려면 데이터 소스의 상관 관계를 올바르게 지정해야 합니다. 의미 체계 명명 규칙을 사용하여 로그, 트레이스 및 지표에서 공유 요소를 식별할 수 있습니다.

예시

```
{
  "@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"
}
```

이 예제는 AWS 에 있는 서비스에서 도착하는 ELB 로그를 보여줍니다 AWS. 상태 코드가 500인 백엔드 HTTP 응답을 보여 오류를 나타냅니다. 이로 인해 알림이 트리거되거나 정기적인 모니터링 프로세스의 일부가 될 수 있습니다. 다음 단계는 철저한 조사를 위해이 이벤트와 관련된 데이터를 수집하는 것입니다.

기간과 관련된 모든 데이터를 쿼리하려는 유혹이 있을 수 있지만이 접근 방식은 압도적일 수 있습니다. 결국 정보가 너무 많아 근본 원인을 식별하는 것보다 관련 없는 데이터를 필터링하는 데 더 많은 시간을 소비할 수 있습니다.

대신 다른 소스의 데이터를 상호 연관시켜 보다 대상화된 접근 방식을 사용할 수 있습니다. 상관관계에 다음 차원을 사용할 수 있습니다.

- IP - `"ip": "10.0.0.1" | "ip": "**************"`

- 포트 - "port": 2817 | "target_port": "10.0.0.1:80"

추가 추적 및 지표 인덱스에 액세스할 수 있고 스키마 구조에 익숙하다면 보다 정확한 상관관계 쿼리를 생성할 수 있습니다.

다음은 상호 연관시킬 수 있는 HTTP 정보가 포함된 추적 인덱스 문서의 예입니다.

```
{
  "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": {
       ...
      }
    }
  }
```

```
  }
}
```

이 접근 방식에서는 elb 로그와 상호 연관될 수 ip 있는 traceId 및 http의 클라이언트/서버를 확인하여 시스템의 동작과 조건을 더 잘 이해할 수 있습니다.

새 상관관계 쿼리 명령

다음은 이러한 유형의 조사를 허용하는 새로운 명령입니다.

```
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 )
```

명령의 각 부분이 수행하는 작업은 다음과 같습니다.

1. source alb_logs, traces - 상호 연관시킬 데이터 소스를 선택합니다.
2. where ip="10.0.0.1" AND cloud.provider="aws" - 이렇게 하면 검색 범위가 좁아집니다.
3. correlate exact fields(traceId, ip) - 다음 필드의 정확한 일치를 기반으로 데이터를 상호 연관시키도록 시스템에 지시합니다.
   - ip 필드에는 명시적 필터 조건이 있으므로 모든 데이터 소스의 상관관계에 사용됩니다.
   - traceId 필드에는 명시적 필터가 없으므로 모든 데이터 소스에서 동일한 traceIds 일치합니다.

필드 이름은 상관관계 명령 내에서 함수의 논리적 의미를 나타냅니다. 실제 조인 조건은 사용자가 제공하는 매핑 문에 따라 달라집니다.

이 용어는 상관관계 문이 쿼리 문을 이행하기 위해 모든 필드가 일치해야 함을 exact 의미합니다.

이 용어는 모범 사례 시나리오에서 일치를 approximate 시도하며 부분 일치가 있는 행을 거부하지 않습니다.

다른 필드 매핑 처리

데이터 소스 간에 동일한 논리 필드(예: ip)의 이름이 다른 경우 경로 필드의 명시적 매핑을 제공해야 합니다. 이를 해결하기 위해 상관관계 조건을 확장하여 서로 다른 필드 이름과 유사한 논리적 의미를 일치시킬 수 있습니다. 이 작업을 수행하는 방법은 다음과 같습니다.

```
alb_logs.ip = traces.attributes.http.server.address, alb_logs.traceId = traces.traceId
```

상관관계 조인에 참여하는 각 필드에 대해이 상관관계 명령으로 조인할 모든 테이블이 포함된 관련 매핑 문을 제공해야 합니다.

예시

이 예제에는 두 가지 소스가 있습니다. `alb_logs, traces`

2개의 필드가 있습니다. `traceId, ip`

2개의 매핑 문이 있습니다. `alb_logs.ip = traces.attributes.http.server.address, alb_logs.traceId = traces.traceId`

상관관계 기간 범위 지정

실행 엔진(드라이버)에서 수행하는 작업을 간소화하기 위해 범위 문을 추가할 수 있습니다. 이렇게 하면 조인 쿼리가이 검색의 범위를 지정해야 하는 시간에 대해 명시적으로 지시됩니다.

`scope(@timestamp, 1D)` i

이 예제에서는 검색 범위가 매일 초점을 맞추므로 같은 날에 나타나는 상관관계가 함께 그룹화됩니다. 이 범위 지정 메커니즘은 결과를 간소화하고 더 잘 제어할 수 있도록 하여 필요에 따라 증분 검색 해상도를 지원합니다.

드라이버 지원

새 상관관계 명령은 실제로 '숨겨진' 조인 명령입니다. 따라서 다음 PPL 드라이버만이 명령을 지원합니다. 이러한 동인에서 상관관계 명령은 적절한 Catalyst Join 논리적 계획으로 직접 변환됩니다.

예시

```
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)
```

논리적 계획:

```
'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]
```

촉매 엔진은 가장 효율적인 조인 순서에 따라이 쿼리를 최적화합니다.

dedup 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

dedup 명령을 사용하여 지정된 필드를 기반으로 검색 결과에서 동일한 문서를 제거합니다.

구문

다음 구문을 사용합니다.

```
dedup [int] <field-list> [keepempty=<bool>] [consecutive=<bool>]
```

`int`

- 선택 사항.
- dedup 명령은 <int>를 지정할 때 각 조합에 대해 여러 이벤트를 유지합니다. <int>의 숫자는 0보다 커야 합니다. 숫자를 지정하지 않으면 처음 발생한 이벤트만 유지됩니다. 다른 모든 중복 항목은 결과에서 제거됩니다.
- 기본값: 1

`keepempty`

- 선택 사항.

- true인 경우는 필드 목록의 필드에 NULL 값이 있거나 MISSING인 문서를 보관합니다.
- 기본값: false

consecutive

- 선택 사항.
- true인 경우 값의 연속 중복 조합이 있는 이벤트만 제거합니다.
- 기본값: false

field-list

- 필수.
- 쉼표로 구분된 필드 목록입니다. 하나 이상의 필드가 필요합니다.

예제 1: 필드 1개 기준 중복

이 예제에서는 성별 필드를 사용하여 문서를 삭제하는 방법을 보여줍니다.

PPL 쿼리:

```
os> source=accounts | dedup gender | fields account_number, gender;
fetched rows / total rows = 2/2
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 13               | F        |
+------------------+----------+
```

예제 2: 중복 문서 2개 유지

이 예제에서는 성별 필드가 있는 문서를 중복된 두 개를 유지하면서 중복 제거하는 방법을 보여줍니다.

PPL 쿼리:

```
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        |
+------------------+----------+
```

예제 3: 기본적으로 빈 필드 유지 또는 무시

이 예제에서는 null 값 필드를 유지하여 문서를 중복 제거하는 방법을 보여줍니다.

PPL 쿼리:

```
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   |
+------------------+-----------------------+
```

이 예제에서는 빈 값 필드를 무시하여 문서를 중복 제거하는 방법을 보여줍니다.

PPL 쿼리:

```
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   |
+------------------+-----------------------+
```

예제 4: 연속 문서 중복

이 예제에서는 연속 문서를 중복 제거하는 방법을 보여줍니다.

PPL 쿼리:

```
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        |
+------------------+----------+
```

추가 예제

- 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 (연속 중복 제거는 지원되지 않음)

제한 사항

- | dedup 2 a, b keepempty=false의 경우

```
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
```

- `| dedup 2 a, b keepempty=true`의 경우

```
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 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`describe` 명령을 사용하여 테이블, 스키마 및 카탈로그의 구조 및 메타데이터에 대한 자세한 정보를 가져옵니다. 다음은 `describe` 명령의 다양한 예제와 사용 사례입니다.

설명

- `describe table`이 명령은 `DESCRIBE EXTENDED table` SQL 명령과 같습니다.
- `describe schema.table`
- `describe schema.`table``
- `describe catalog.schema.table`
- `describe catalog.schema.`table``
- `describe `catalog`.`schema`.`table``

eval 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`eval` 명령은 표현식을 평가하고 검색 결과에 결과를 추가합니다.

구문

다음 구문을 사용합니다.

```
eval <field>=<expression> ["," <field>=<expression> ]...
```

- `field`: 필수. 필드 이름이 없으면 새 필드가 추가됩니다. 필드 이름이 이미 있는 경우 재정의됩니다.
- `expression`: 필수. 시스템에서 지원하는 모든 표현식입니다.

예제 1: 새 필드 생성

이 예제에서는 각 문서에 대해 새 `doubleAge` 필드를 생성하는 방법을 보여줍니다. 새 `doubleAge`는 연령에 2를 곱한 평가 결과입니다.

PPL 쿼리:

```
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          |
+-------+-------------+
```

예제 2: 기존 필드 재정의

이 예제에서는 기존 연령 필드를 연령에 1을 더한 값으로 재정의하는 방법을 보여줍니다.

PPL 쿼리:

```
os> source=accounts | eval age = age + 1 | fields age ;
fetched rows / total rows = 4/4
+-------+
| age   |
|-------|
| 33    |
| 37    |
| 29    |
| 34    |
+-------+
```

예제 3: eval에 정의된 필드를 사용하여 새 필드 생성

이 예제에서는 eval 명령에 정의된 ddAge 필드를 사용하여 새 필드를 생성하는 방법을 보여줍니다. 새 필드는 평가 결과에 2를 doubleAge곱한 값ddAge입니다. 여기서 doubleAge는 eval 명령에 정의되어 있습니다.

PPL 쿼리:

```
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     |
+-------+-------------+---------+
```

가정: a, b는의 기존 필드c입니다. `table`

추가 예제

- `source = table | eval f = 1 | fields a,b,c,f`
- `source = table | eval f = 1` (출력 a,b,c,f 필드)
- `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`

사례로 평가 예제:

```
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')
```

다른 사례 예제로 평가:

가정: a, b는의 기존 필드c입니다. `table`

추가 예제

- `source = table | eval f = 1 | fields a,b,c,f`

- source = table | eval f = 1 (출력 a,b,c,f 필드)
- 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

사례로 평가 예제:

```
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')
```

다른 사례 예제로 평가:

```
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
```

제한 사항

- 기존 필드 재정의는 지원되지 않습니다. 이렇게 하려고 시도하는 쿼리는 '참조 'a'가 모호함'이라는 메시지와 함께 예외가 발생합니다.

```
- `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 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`eventstats` 명령을 사용하여 계산된 요약 통계로 이벤트 데이터를 보강합니다. 이벤트 내에서 지정된 필드를 분석하고 다양한 통계 측정값을 계산한 다음 이러한 결과를 각 원래 이벤트에 새 필드로 추가하여 작동합니다.

이벤트 통계의 주요 측면

1. 전체 결과 집합 또는 정의된 그룹 내에서 계산을 수행합니다.
2. 원래 이벤트는 그대로 유지되며 통계 결과를 포함하도록 새 필드가 추가되었습니다.
3. 이 명령은 비교 분석, 이상치 식별 또는 개별 이벤트에 대한 추가 컨텍스트 제공에 특히 유용합니다.

통계와 이벤트 통계의 차이

stats 및 eventstats 명령은 통계를 계산하는 데 모두 사용되지만 작동 방식과 생성되는 항목에는 몇 가지 주요 차이점이 있습니다.

출력 형식

- stats: 계산된 통계만 포함된 요약 테이블을 생성합니다.
- eventstats: 계산된 통계를 기존 이벤트에 새 필드로 추가하여 원본 데이터를 보존합니다.

이벤트 보존

- stats: 결과 세트를 통계 요약으로만 줄여 개별 이벤트를 삭제합니다.
- eventstats: 모든 원래 이벤트를 유지하고 계산된 통계와 함께 새 필드를 추가합니다.

사용 사례

- stats: 요약 보고서 또는 대시보드를 생성하는 데 가장 적합합니다. 결과를 요약하는 최종 명령으로 자주 사용됩니다.
- eventstats: 추가 분석 또는 필터링을 위해 통계 컨텍스트로 이벤트를 보강해야 할 때 유용합니다. 검색 중 후속 명령에 사용할 수 있는 통계를 추가하는 데 사용할 수 있습니다.

구문

다음 구문을 사용합니다.

```
eventstats <aggregation>... [by-clause]
```

aggregation

- 필수.
- 집계 함수입니다.
- 집계 인수는 필드여야 합니다.

섹션별

- 선택 사항.
- 구문: by [span-expression,] [field,]...

- by 절에는 스칼라 함수 및 집계 함수와 같은 필드와 표현식이 포함될 수 있습니다. 스팬 절을 사용하여 특정 필드를 동일한 간격의 버킷으로 분할할 수도 있습니다. 그런 다음 eventstats 명령은 이러한 스팬 버킷을 기반으로 집계를 수행합니다.
- 기본값: by 절을 지정하지 않으면 eventstats 명령이 전체 결과 집합에 걸쳐 집계됩니다.

스팬 표현식

- 선택 사항, 최대 1개.
- 구문: `span(field_expr, interval_expr)`
- 간격 표현식의 단위는 기본적으로 자연 단위입니다. 그러나 날짜 및 시간 유형 필드의 경우 날짜/시간 단위를 사용할 때 간격 표현식에 단위를 지정해야 합니다.

  예를 들어 필드를 버킷age으로 10년 분할하려면를 사용합니다`span(age, 10)`. 시간 기반 필드의 경우를 사용하여 `timestamp` 필드를 시간별 간격으로 분할할 수 있습니다`span(timestamp, 1h)`.

사용 가능한 시간 단위

범위 간격 단위
밀리초(ms)
초(초)
분(m, 대/소문자 구분)
시간(h)
일(d)
주(w)
month(M, 대/소문자 구분)
분기(q)
연도(y)

집계 함수

COUNT

COUNT는 SELECT 문으로 검색된 행의 expr 수를 반환합니다.

CloudWatch Logs에서 쿼리를 사용하는 경우 COUNT는 지원되지 않습니다.

예시

```
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)는 expr의 합계를 반환합니다.

예시

```
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)는 expr의 평균 값을 반환합니다.

예시

```
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)` expr의 최대값을 반환합니다.

예시

```
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)` expr의 최소값을 반환합니다.

예시

```
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) expr의 샘플 표준 편차를 반환합니다.

예시

```
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) expr의 모집단 표준 편차를 반환합니다.

예시

```
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 또는 PERCENTILE_APPROX

PERCENTILE(expr, percent) 또는 expr의 대략적인 백분위수 값을 지정된 백분율로 PERCENTILE_APPROX(expr, percent) 반환합니다.

%

- 숫자는 0에서 100 사이의 상수여야 합니다.

예시

```
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
       |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+--------------------------+--------+-------
+--------------------------------+
```

예제 1: 그룹별 필드의 평균, 합계 및 개수 계산

이 예제에서는 성별을 기준으로 모든 계정 그룹의 평균 연령, 합계 연령 및 이벤트 수를 계산합니다.

```
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     |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+-----------+-----------+--------+
```

예제 2: 범위별로 개수 계산

이 예제에서는 연령을 10년 간격으로 가져옵니다.

```
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         |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+-----------+
```

예제 3: 성별 및 범위별로 개수 계산

이 예제에서는 연령을 5년 간격으로 구하고 그룹을 성별로 구합니다.

```
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   |
+----------------+----------+-----------+----------+-----+--------
+----------------------+-------------+---------------------------+--------+-------
+-----+
```

사용법

- 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)

스팬을 사용한 집계

- 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`

기간 범위를 사용한 집계(텀블 윈도우 함수)

- `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`

여러 수준별 집계 그룹

- `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`

명령 확장

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`expand` 명령을 사용하여 유형의 필드를 평면화합니다.

- `Array<Any>`
- `Map<Any>`

구문

다음 구문을 사용합니다.

```
expand <field> [As alias]
```

필드

- 확장(확장)할 필드입니다. 지원되는 유형이어야 합니다.

별칭

- 선택 사항. 원래 필드 이름 대신 사용할 이름입니다.

사용법

이 expand 명령은 지정된 배열 또는 맵 필드의 각 요소에 대한 행을 생성합니다.

- 배열 요소는 개별 행이 됩니다.
- 맵 키-값 페어는 별도의 행으로 나뉘며 각 키-값은 행으로 표시됩니다.

- 별칭이 제공되면 분해된 값이 원래 필드 이름 대신 별칭 아래에 표시됩니다.
- 이는 , stats eval및와 같은 다른 명령과 함께 사용하여 확장 후 데이터를 조작하거나 추출parse할 수 있습니다.

예시

- `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`

설명 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`explain` 명령은 쿼리 실행 계획을 이해하는 데 도움이 되므로 쿼리를 분석하고 최적화하여 성능을 높일 수 있습니다. 이 소개에서는 쿼리 최적화에서 설명 명령의 용도와 중요도에 대한 간략한 개요를 제공합니다.

설명

- `source=accounts | top gender // finds most common gender of all the accounts` (줄 주석)
- `source=accounts | dedup 2 gender /* dedup the document with gender field keep 2 duplication */ | fields account_number, gender` (설명 차단)

설명

- `describe table`이 명령은 `DESCRIBE EXTENDED table` SQL 명령과 같습니다.
- `describe schema.table`
- `describe schema.`table``
- `describe catalog.schema.table`
- `describe catalog.schema.`table``
- `describe `catalog`.`schema`.`table``

설명

- `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

필드

- 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

필드 요약

- 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

중첩 필드

- 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

필터

- 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` - 참고:이 값은 >= 1 및 <= 4, 즉 [1, 4]를 반환합니다.
- `source = table | where b not between '2024-09-10' and '2025-09-10'` - 참고: b >= '**********' 및 b <= '2025-09-10'을 반환합니다.
- `source = table | where cidrmatch(ip, '***********/24')`
- `source = table | where cidrmatch(ipv6, '2003:db8::/32')`
- `source = table | trendline sma(2, temperature) as temp_trend`

IP 관련 쿼리

- `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`

복잡한 필터

```
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
```

논리적 조건이 있는 필터

- `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`

평가

가정: a, b, c는의 기존 필드입니다. `table`

- `source = table | eval f = 1 | fields a,b,c,f`
- `source = table | eval f = 1` (출력 a,b,c,f 필드)
- `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 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

설명

`fillnull` 명령을 사용하여 검색 결과의 하나 이상의 필드에서 null 값을 지정된 값으로 바꿉니다.

구문

다음 구문을 사용합니다.

```
fillnull [with <null-replacement> in <nullable-field>["," <nullable-field>]] | [using
 <source-field> = <null-replacement> ["," <source-field> = <null-replacement>]]
```

- null 대체: 필수. null 값을 대체하는 데 사용되는 값입니다.
- nullable-field: 필수. 필드 참조. 이 필드의 null 값은 null 대체에 지정된 값으로 대체됩니다.

예제 1: 필드 하나 채우기

이 예제는 단일 필드에서 fillnull을 사용하는 방법을 보여줍니다.

```
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         |
```

예제 2: 여러 필드에 적용된 Fillnull

이 예제는 여러 필드에 적용된 fillnull을 보여줍니다.

```
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    |
```

예제 3: 다양한 null 대체 값을 사용하여 여러 필드에 적용된 Fillnull입니다.

이 예제에서는 null을 대체하는 데 사용되는 다양한 값으로 fillnull을 보여줍니다.

- `/error` `request_path` 필드 내
- `1970-01-01 00:00:00` `timestamp` 필드 내

```
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    |
```

필드 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`fields` 명령을 사용하여 검색 결과에서 필드를 유지하거나 제거합니다.

구문

다음 구문을 사용합니다.

```
field [+|-] <field-list>
```

- index: 선택 사항입니다.

  더하기(+)를 사용하는 경우 필드 목록에 지정된 필드만 유지됩니다.

  마이너스(-)를 사용하면 필드 목록에 지정된 모든 필드가 제거됩니다.

  기본값: +
- field list: 필수. 유지하거나 제거할 쉼표로 구분된 필드 목록입니다.

예제 1: 결과에서 지정된 필드 선택

이 예제에서는 검색 결과에서 account_number, firstname및 lastname 필드를 가져오는 방법을 보여줍니다.

PPL 쿼리:

```
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         |
+------------------+-------------+------------+
```

예제 2: 결과에서 지정된 필드 제거

이 예제에서는 검색 결과에서 account_number 필드를 제거하는 방법을 보여줍니다.

PPL 쿼리:

```
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         |
+-------------+------------+
```

추가 예제

- `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`

중첩 필드 예제:

```
`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`
```

flatten 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

플랫 명령을 사용하여 다음 유형의 필드를 확장합니다.

- `struct<?,?>`
- `array<struct<?,?>>`

구문

다음 구문을 사용합니다.

```
flatten <field>
```

- 필드: 평면화할 필드입니다. 필드는 지원되는 유형이어야 합니다.

스키마

col_name	data_type
_시간	문자열
브리지	array<struct<length:bigint,name:string>>
city	문자열
쿠어	struct<alt:bigint,lat:double,long:double>
country	문자열

Data

_시간	브리지	city	쿠어	country
2024-09-13T12:00:00	[{801, Tower Bridge}, {928, London Bridge}]	런던	{35, 51.5074, -0.1278}	영국
2024-09-13T12:00:00	[{232, Pont Neuf}, {160, Pont	파리	{35, 48.8566, 2.3522}	프랑스

_시간	브리지	city	쿠어	country
	Alexandre III}]			
2024-09-13T12:00:00	[{48, Rialto Bridge}, {11, Bridge of Sighs}]	베네치아	{2, 45.4408, 12.3155}	이탈리아
2024-09-13T12:00:00	[{***, Charles Bridge}, {343, Legion Bridge}]	프라하	{200, 50.0755, 14.4378}	체코 공화국
2024-09-13T12:00:00	[{375, Chain Bridge}, {333, Liberty Bridge}]	부다페스트	{96, 47.4979, 19.0402}	헝가리
1990-09-13T12:00:00	NULL	바르샤바어	NULL	폴란드

예제 1: 구조체 평면화

이 예제에서는 구조체 필드를 평면화하는 방법을 보여줍니다.

PPL 쿼리:

```
source=table | flatten coor
```

_시간	브리지	city	country	alt	lat	long
2024-09-13T12:00:00	[{801, Tower	런던	영국	35	51.5074	-0.1278

_시간	브리지	city	country	alt	lat	long
	Bridge}, {928, London Bridge}]					
2024-09-13T12:00:00	[{232, Pont Neuf}, {160, Pont Alexandre III}]	파리	프랑스	35	48.8566	2.3522
2024-09-13T12:00:00	[{48, Rialto Bridge}, {11, Bridge of Sighs}]	베네치아	이탈리아	2	45.4408	12.3155
2024-09-13T12:00:00	[{516, Charles Bridge}, {343, Legion Bridge}]	프라하	체코 공화국	200	50.0755	14.4378
2024-09-13T12:00:00	[{375, Chain Bridge}, {333, Liberty Bridge}]	부다페스트	헝가리	96	47.4979	19.0402

_시간	브리지	city	country	alt	lat	long
1990-09-13T12:00:00	NULL	바르샤바어	폴란드	NULL	NULL	NULL

예제 2: 배열 평면화

이 예제에서는 구조체 필드의 배열을 평면화하는 방법을 보여줍니다.

PPL 쿼리:

```
source=table | flatten bridges
```

_시간	city	Coor	country	length	name
2024-09-13T12:00:00	런던	{35, 51.5074, -0.1278}	영국	801	타워 브리지
2024-09-13T12:00:00	런던	{35, 51.5074, -0.1278}	영국	928	런던 브리지
2024-09-13T12:00:00	파리	{35, 48.8566, 2.3522}	프랑스	232	Pont Neuf
2024-09-13T12:00:00	파리	{35, 48.8566, 2.3522}	프랑스	160	퐁 알렉산드레 III
2024-09-13T12:00:00	베네치아	{2, 45.4408, 12.3155}	이탈리아	48	Rialto 브리지
2024-09-13T12:00:00	베네치아	{2, 45.4408, 12.3155}	이탈리아	11	한숨의 브리지
2024-09-13T12:00:00	프라하	{200, 50.0755, 14.4378}	체코 공화국	516	찰스 브리지
2024-09-13T12:00:00	프라하	{200, 50.0755, 14.4378}	체코 공화국	343	리전 브리지

_시간	city	Coor	country	length	name
2024-09-13T12:00:00	부다페스트	{96, 47.4979, 19.0402}	헝가리	375	체인 브리지
2024-09-13T12:00:00	부다페스트	{96, 47.4979, 19.0402}	헝가리	333	자유 브리지
1990-09-13T12:00:00	바르샤바어	NULL	폴란드	NULL	NULL

예제 3: 배열 및 구조체 평면화

이 예제에서는 여러 필드를 평면화하는 방법을 보여줍니다.

PPL 쿼리:

```
source=table | flatten bridges | flatten coor
```

_시간	city	country	length	name	alt	lat	long
2024-09-13T12:00:00	런던	영국	801	타워 브리지	35	51.5074	-0.1278
2024-09-13T12:00:00	런던	영국	928	런던 브리지	35	51.5074	-0.1278
2024-09-13T12:00:00	파리	프랑스	232	Pont Neuf	35	48.8566	2.3522
2024-09-13T12:00:00	파리	프랑스	160	퐁 알렉산드레 III	35	48.8566	2.3522

_시간	city	country	length	name	alt	lat	long
2024-09-13T12:00:00	베네치아	이탈리아	48	Rialto 브리지	2	45.4408	12.3155
2024-09-13T12:00:00	베네치아	이탈리아	11	한숨의 브리지	2	45.4408	12.3155
2024-09-13T12:00:00	프라하	체코 공화국	516	찰스 브리지	200	50.0755	14.4378
2024-09-13T12:00:00	프라하	체코 공화국	343	리전 브리지	200	50.0755	14.4378
2024-09-13T12:00:00	부다페스트	헝가리	375	체인 브리지	96	47.4979	19.0402
2024-09-13T12:00:00	부다페스트	헝가리	333	자유 브리지	96	47.4979	19.0402
1990-09-13T12:00:00	바르샤바	폴란드	NULL	NULL	NULL	NULL	NULL

grok 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

grok 명령은 grok 패턴으로 텍스트 필드를 구문 분석하고 검색 결과에 결과를 추가합니다.

구문

다음 구문을 사용합니다.

```
grok <field> <pattern>
```

필드

- 필수.
- 필드는 텍스트 필드여야 합니다.

패턴

- 필수.
- 지정된 텍스트 필드에서 새 필드를 추출하는 데 사용되는 grok 패턴입니다.
- 새 필드 이름이 이미 있는 경우 원래 필드를 대체합니다.

Grok 패턴

grok 패턴은 각 문서의 텍스트 필드와 일치시켜 새 필드를 추출하는 데 사용됩니다.

예제 1: 새 필드 생성

이 예제에서는 각 문서에 host 대해 새 필드를 생성하는 방법을 보여줍니다. host는 @ email 필드에서 뒤에 오는 호스트 이름이 됩니다. null 필드를 구문 분석하면 빈 문자열이 반환됩니다.

```
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 |
+-------------------------+-------------+
```

예제 2: 기존 필드 재정의

이 예제에서는 거리 번호가 제거된 기존 address 필드를 재정의하는 방법을 보여줍니다.

```
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    |
+------------------+
```

예제 3: grok를 사용하여 로그 구문 분석

이 예제에서는 grok를 사용하여 원시 로그를 구문 분석하는 방법을 보여줍니다.

```
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
    |
+-------------------------------------------------------------------------------------------------------
+----------------------------+------------+---------+
```

제한 사항

grok 명령에는 구문 분석 명령과 동일한 제한 사항이 있습니다.

헤드 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

head 명령을 사용하여 선택적 오프셋 뒤에 지정된 결과의 첫 번째 N개를 검색 순서로 반환합니다.

구문

다음 구문을 사용합니다.

```
head [<size>] [from <offset>]
```

<크기>

- 선택적 정수입니다.
- 반환할 결과 수.
- 기본값: 10

<오프셋>

- 선택 사항인 이후의 정수입니다`from`.
- 건너뛸 결과 수입니다.
- 기본값: 0

예제 1: 처음 10개의 결과 가져오기

이 예제는 계정 인덱스에서 최대 10개의 결과를 검색하는 방법을 보여줍니다.

PPL 쿼리:

```
os> source=accounts | fields firstname, age | head;
fetched rows / total rows = 4/4
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| Jane        | 32    |
| John        | 36    |
| Jorge       | 28    |
| Juan        | 33    |
+-------------+-------+
```

예제 2: 첫 번째 N개의 결과 가져오기

이 예제는 계정 인덱스의 첫 번째 N 결과를 보여줍니다.

PPL 쿼리:

```
os> source=accounts | fields firstname, age | head 3;
fetched rows / total rows = 3/3
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| Jane        | 32    |
| John        | 36    |
| Jorge       | 28    |
+-------------+-------+
```

예제 3: 오프셋 M 후 첫 번째 N 결과 가져오기

이 예제는 계정 인덱스에서 M 결과를 건너뛴 후 첫 번째 N 결과를 검색하는 방법을 보여줍니다.

PPL 쿼리:

```
os> source=accounts | fields firstname, age | head 3 from 1;
fetched rows / total rows = 3/3
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| John        | 36    |
| Jorge       | 28    |
| Juan        | 33    |
+-------------+-------+
```

조인 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

조인 명령을 사용하면 공통 필드를 기반으로 여러 소스의 데이터를 결합할 수 있으므로 복잡한 분석을 수행하고 분산 데이터 세트에서 심층적인 인사이트를 얻을 수 있습니다.

스키마

인덱스는 `otel-v1-apm-span-*` (large)와 `otel-v1-apm-service-map` (small) 두 개 이상 있습니다.

인덱스의 관련 필드:

`otel-v1-apm-span-*`

- traceId - 추적의 고유 식별자입니다. 동일한 트레이스의 모든 스팬은 동일한 traceId를 공유합니다.
- spanId - 추적 내의 스팬에 대한 고유 식별자로, 스팬이 생성될 때 할당됩니다.
- parentSpanId -이 스팬의 상위 스팬의 spanId입니다. 루트 범위인 경우이 필드는 비어 있어야 합니다.
- durationInNanos - startTime과 endTime 간의 나노초 차이입니다(`UIlatency`에 있음).
- serviceName - 스팬이 시작되는 리소스입니다.
- traceGroup - 추적의 루트 범위 이름입니다.

`otel-v1-apm-service-map`

- serviceName - 스팬을 내보낸 서비스의 이름입니다.
- destination.domain -이 클라이언트가 호출하는 서비스의 serviceName입니다.
- destination.resource -이 클라이언트가 호출하는 스팬 이름(API, 작업 등)입니다.
- target.domain - 클라이언트가 호출하는 서비스의 serviceName입니다.
- target.resource - 클라이언트가 호출하는 스팬 이름(API, 작업 등)입니다.
- traceGroupName - 요청 체인을 시작한 최상위 스팬 이름입니다.

요구 사항

다음을 계산join할 수 있도록 지원합니다.

각 서비스에 대해 서비스 맵 인덱스의 스팬 인덱스를 조인하여 다양한 유형의 필터에서 지표를 계산합니다.

이 샘플 쿼리는 order 서비스에 client_cancel_order 대한 추적 그룹별로 필터링할 때 지연 시간을 계산합니다.

```
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`)
```

PPL로 마이그레이션

join 명령의 구문

```
SEARCH source=<left-table>
| <other piped command>
| [joinType] JOIN
    [leftAlias]
    ON joinCriteria
    <right-table>
| <other piped command>
```

재작성

```
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

- 구문: `INNER | LEFT OUTER | CROSS`
- 선택 사항
- 수행할 조인의 유형입니다. 지정하지 `INNER` 않으면 기본값은 입니다.

leftAlias

- 구문: `left = <leftAlias>`
- 선택 사항
- 모호한 이름 지정을 방지하기 위해 왼쪽 조인 측에 사용할 하위 쿼리 별칭입니다.

joinCriteria

- 구문: `<expression>`
- 필수
- 구문은 로 시작합니다`ON`. 모든 비교 표현식일 수 있습니다. 일반적으로 조인 기준은와 같습니다`<leftAlias>.<leftField>=<rightAlias>.<rightField>`.

  예를 들어 `l.id = r.id`입니다. 조인 기준에 여러 조건이 포함된 경우 각 비교 표현식 간에 `AND` 및 `OR` 연산자를 지정할 수 있습니다. 예를 들어 `l.id = r.id AND l.email = r.email AND (r.age > 65 OR r.age < 18)`입니다.

추가 예제

SQL 쿼리에서 마이그레이션(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;
```

PPL 조인 쿼리로 다시 작성:

```
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
```

제한: 오른쪽 조인에서는 하위 검색이 지원되지 않습니다.

하위 검색이 지원되는 경우 다음과 같이 위의 PPL 쿼리를 다시 작성할 수 있습니다.

```
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
```

조회 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

lookup 명령을 사용하여 조회 인덱스(차원 테이블)의 데이터를 추가하거나 대체하여 검색 데이터를 보강합니다. 이 명령을 사용하면 인덱스의 필드를 차원 테이블의 값으로 확장할 수 있습니다. 조회 조건이 충족될 때 값을 추가하거나 대체하는 데에도 사용할 수 있습니다. lookup 명령은 정적 데이터 세트로 소스 데이터를 보강하는 Join 명령보다 더 적합합니다.

구문

다음 구문을 사용합니다.

```
SEARCH source=<sourceIndex>
| <other piped command>
| LOOKUP <lookupIndex> (<lookupMappingField> [AS <sourceMappingField>])...
    [(REPLACE | APPEND) (<inputField> [AS <outputField>])...]
| <other piped command>
```

lookupIndex

- 필수 사항입니다.
- 조회 인덱스의 이름(차원 테이블).

lookupMappingField

- 필수 사항입니다.
- 오른쪽 테이블의 조인 키와 유사한 조회 인덱스의 매핑 키입니다. 쉼표로 구분된 여러 필드를 지정할 수 있습니다.

sourceMappingField

- 선택 사항.
- 기본값: <lookupMappingField>.

- 왼쪽의 조인 키와 유사한 소스 쿼리의 매핑 키입니다.

inputField

- 선택 사항.
- 기본값: 일치하는 값이 있는 조회 인덱스의 모든 필드입니다.
- 일치하는 값이 결과 출력에 적용되는 조회 인덱스의 필드입니다. 쉼표로 구분된 여러 필드를 지정할 수 있습니다.

outputField

- 선택 사항.
- 기본값: <inputField>.
- 출력의 필드입니다. 여러 출력 필드를 지정할 수 있습니다. 소스 쿼리에서 기존 필드 이름을 지정하면 해당 값이 inputField의 일치하는 값으로 대체되거나 추가됩니다. 새 필드 이름을 지정하면 결과에 추가됩니다.

REPLACE | APPEND

- 선택 사항.
- 기본값: REPLACE
- 일치하는 값을 처리하는 방법을 지정합니다. REPLACE를 지정하면 <lookupIndex> 필드의 일치하는 값이 결과 값을 덮어씁니다. 를 지정하는 경우 <lookupIndex> 필드의 APPEND일치하는 값은 결과에서 누락된 값에만 추가됩니다.

사용법

- 룩업 <lookupIndex> ID AS ID REPLACE 메일 AS 이메일
- 룩업 <lookupIndex> 이름 REPLACE mail AS email
- 룩업 <lookupIndex> ID AS ID, 이름 APPEND 주소, 메일 AS 이메일
- 룩업 <lookupIndex> ID

예시

다음 예시를 참조하세요.

```
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
```

구문 분석 명령

`parse` 명령은 정규식을 사용하여 텍스트 필드를 구문 분석하고 검색 결과에 결과를 추가합니다.

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

구문

다음 구문을 사용합니다.

```
parse <field> <pattern>
```

`field`

- 필수.
- 필드는 텍스트 필드여야 합니다.

`pattern`

- 필수 문자열입니다.

- 지정된 텍스트 필드에서 새 필드를 추출하는 데 사용되는 정규식 패턴입니다.
- 새 필드 이름이 이미 있는 경우 원래 필드를 대체합니다.

정규식

정규식 패턴은 Java 정규식 엔진이 있는 각 문서의 전체 텍스트 필드를 일치시키는 데 사용됩니다. 표현식의 각 명명된 캡처 그룹은 새 STRING 필드가 됩니다.

예제 1: 새 필드 생성

이 예제에서는 각 문서에 host 대해 새 필드를 생성하는 방법을 보여줍니다. host는 @ email 필드에서 뒤에 오는 호스트 이름이 됩니다. null 필드를 구문 분석하면 빈 문자열이 반환됩니다.

PPL 쿼리:

```
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 |
+-----------------------+-------------+
```

예제 2: 기존 필드 재정의

이 예제에서는 거리 번호가 제거된 기존 address 필드를 재정의하는 방법을 보여줍니다.

PPL 쿼리:

```
os> source=accounts | parse address '\d+ (?<address>.+)' | fields address ;
fetched rows / total rows = 4/4
+------------------+
| address          |
|------------------|
| Example Lane     |
| Example Street   |
| Example Avenue   |
| Example Court    |
```

```
+------------------+
```

예제 3: 캐스팅된 구문 분석 필드를 기준으로 필터링 및 정렬

이 예제는 address 필드에서 500보다 높은 거리 번호를 정렬하는 방법을 보여줍니다.

PPL 쿼리:

```
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   |
+----------------+----------------+
```

제한 사항

구문 분석 명령에는 몇 가지 제한 사항이 있습니다.

- 구문 분석으로 정의된 필드는 다시 구문 분석할 수 없습니다.

  다음 명령은 작동하지 않습니다.

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | parse street '\w+ (?<road>\w+)'
  ```

- 구문 분석으로 정의된 필드는 다른 명령으로 재정의할 수 없습니다.

  where는 재정의할 street 수 없으므로 문서와 일치하지 않습니다.

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | eval street='1' | where
   street='1' ;
  ```

- 구문 분석에 사용되는 텍스트 필드는 재정의할 수 없습니다.

  street는 재정address의되므로 성공적으로 구문 분석되지 않습니다.

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | eval address='1' ;
  ```

- 구문 분석으로 정의된 필드는 stats 명령에서 사용한 후 필터링하거나 정렬할 수 없습니다.

  where 다음 명령의는 작동하지 않습니다.

```
source=accounts | parse email '.+@(?<host>.+)' | stats avg(age) by host | where
 host=pyrami.com ;
```

패턴 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

patterns 명령은 텍스트 필드에서 로그 패턴을 추출하고 검색 결과에 결과를 추가합니다. 로그를 패턴별로 그룹화하면 분석 및 문제 해결을 위해 대량의 로그 데이터에서 통계를 더 쉽게 집계할 수 있습니다.

구문

다음 구문을 사용합니다.

```
patterns [new_field=<new-field-name>] [pattern=<pattern>] <field>
```

new-field-name

- 선택적 문자열입니다.
- 추출된 패턴에 대한 새 필드의 이름입니다.
- 기본값은 patterns_field입니다.
- 이름이 이미 있는 경우 원래 필드를 대체합니다.

패턴

- 선택적 문자열입니다.
- 텍스트 필드에서 필터링해야 하는 정규식 문자 패턴입니다.
- 없는 경우 기본 패턴은 영숫자 문자([a-zA-Z\d])입니다.

필드

- 필수.
- 필드는 텍스트 필드여야 합니다.

예제 1: 새 필드 생성

이 예제에서는 각 문서에 `email` 대해에서 추출 구두점을 사용하는 방법을 보여줍니다. null 필드를 구문 분석하면 빈 문자열이 반환됩니다.

PPL 쿼리:

```
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   | @.               |
+-----------------------+------------------+
```

예제 2: 로그 패턴 추출

이 예제에서는 기본 패턴을 사용하여 원시 로그 필드에서 구두점을 추출하는 방법을 보여줍니다.

PPL 쿼리:

```
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
                                       | ... - - [//::: -] " / /."       |
+-------------------------------------------------------------------------------------------
+--------------------------------+
```

예제 3: 사용자 지정 정규식 패턴을 사용하여 로그 패턴 추출

이 예제에서는 사용자 정의 패턴을 사용하여 원시 로그 필드에서 구두점을 추출하는 방법을 보여줍니다.

PPL 쿼리:

```
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/."
                                       |
+-------------------------------------------------------------------------------------------
+-----------------------------------------------------------------------------
+
```

제한 사항

패턴 명령에는 구문 분석 명령과 동일한 제한 사항이 있습니다.

희귀 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`rare` 명령을 사용하여 필드 목록에서 모든 필드 값의 가장 덜 일반적인 튜플을 찾습니다.

> **Note**
>
> 그룹별 필드의 개별 값 튜플마다 최대 10개의 결과가 반환됩니다.

구문

다음 구문을 사용합니다.

```
rare [N] <field-list> [by-clause] rare_approx [N] <field-list> [by-clause]
```

필드 목록

- 필수.
- 쉼표로 구분된 필드 이름 목록입니다.

섹션별

- 선택 사항.
- 결과를 그룹화할 하나 이상의 필드입니다.

N

- 반환할 결과 수.
- 기본값: 10

rare_ 대략

- HyperLogLog++ 알고리즘의 예상 카디널리티를 사용하여 드문(n) 필드의 대략적인 수입니다.

예제 1: 필드에서 가장 덜 일반적인 값 찾기

이 예제에서는 모든 계정의 가장 덜 일반적인 성별을 찾습니다.

PPL 쿼리:

```
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        |
+----------+
```

예제 2: 성별별로 구성된 가장 덜 일반적인 값 찾기

이 예제에서는 성별을 기준으로 모든 계정 그룹의 가장 덜 일반적인 연령을 찾습니다.

PPL 쿼리:

```
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    |
+----------+-------+
```

이름 바꾸기 명령

`rename` 명령을 사용하여 검색 결과에서 하나 이상의 필드 이름을 변경합니다.

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

구문

다음 구문을 사용합니다.

```
rename <source-field> AS <target-field>["," <source-field> AS <target-field>]...
```

소스 필드

- 필수.
- 이름을 바꾸려는 필드의 이름입니다.

대상 필드

- 필수.
- 이름을 바꾸려는 이름입니다.

예제 1: 필드 이름 바꾸기

이 예제에서는 단일 필드의 이름을 바꾸는 방법을 보여줍니다.

PPL 쿼리:

```
os> source=accounts | rename account_number as an | fields an;
fetched rows / total rows = 4/4
+------+
| an   |
|------|
| 1    |
| 6    |
| 13   |
| 18   |
+------+
```

예제 2: 여러 필드 이름 바꾸기

이 예제에서는 여러 필드의 이름을 바꾸는 방법을 보여줍니다.

PPL 쿼리:

```
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    |
+------+---------+
```

제한 사항

- 기존 필드 재정의는 지원되지 않습니다.

```
source=accounts | grok address '%{NUMBER} %{GREEDYDATA:address}' | fields address
```

검색 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`search` 명령을 사용하여 인덱스에서 문서를 검색합니다. `search` 명령은 PPL 쿼리에서 첫 번째 명령으로만 사용할 수 있습니다.

구문

다음 구문을 사용합니다.

```
search source=[<remote-cluster>:]<index> [boolean-expression]
```

search

- 선택 사항.
- 검색 키워드는 생략할 수 있습니다.

인덱스

- 필수.
- 검색 명령은 쿼리할 인덱스를 지정해야 합니다.
- 클러스터 간 검색을 `<cluster name>:` 위해에서 인덱스 이름 앞에 접두사를 붙일 수 있습니다.

부울 표현식

- 선택 사항.
- 부울 값으로 평가되는 표현식입니다.

예제 1: 모든 데이터 가져오기

이 예제는 계정 인덱스에서 모든 문서를 가져오는 방법을 보여줍니다.

PPL 쿼리:

```
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         |
```

```
+------------------+-------------+----------------------+-----------+----------
+--------+---------------+---------+-------+-----------------------+------------+
```

예제 2: 조건을 사용하여 데이터 가져오기

이 예제는를 사용하여 계정 인덱스에서 모든 문서를 가져오는 방법을 보여줍니다.

PPL 쿼리:

```
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 명령

`sort` 명령을 사용하여 지정된 필드별로 검색 결과를 정렬합니다.

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

구문

다음 구문을 사용합니다.

```
sort <[+|-] sort-field>...
```

[+|-]

- 선택 사항.
- 더하기 [+]는 NULL/MISSING 값이 먼저 있는 오름차순을 나타냅니다.
- 빼기 [-]는 NULL/MISSING 값이 마지막인 내림차순을 나타냅니다.
- 기본값: NULL/MISSING 값이 먼저 있는 오름차순입니다.

정렬 필드

- 필수.
- 정렬에 사용되는 필드입니다.

예제 1: 하나의 필드로 정렬

이 예제에서는 연령 필드를 사용하여 문서를 오름차순으로 정렬하는 방법을 보여줍니다.

PPL 쿼리:

```
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    |
+------------------+-------+
```

예제 2: 하나의 필드를 기준으로 정렬하고 모든 결과를 반환합니다.

이 예제에서는 연령 필드를 사용하여 문서를 오름차순으로 정렬하는 방법을 보여줍니다.

PPL 쿼리:

```
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    |
+-------------------+-------+
```

예제 3: 한 필드를 기준으로 내림차순으로 정렬

이 예제에서는 연령 필드를 사용하여 문서를 내림차순으로 정렬하는 방법을 보여줍니다.

PPL 쿼리:

```
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    |
+-------------------+-------+
```

예제 4: 여러 필드를 기준으로 정렬

이 예제에서는 성별 필드를 오름차순으로 정렬하고 연령 필드를 내림차순으로 정렬하는 방법을 보여줍니다.

PPL 쿼리:

```
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    |
+-------------------+----------+-------+
```

예제 5: 정렬 기준 필드에 null 값 포함

이 예제에서는 기본 옵션(오름차순 및 null 우선)을 기준으로 고용주 필드를 정렬하는 방법을 보여줍니다. 결과는 null 값이 첫 번째 행에 있음을 보여줍니다.

PPL 쿼리:

```
os> source=accounts | sort employer | fields employer;
fetched rows / total rows = 4/4
+------------+
| employer   |
|------------|
| null       |
| AnyCompany |
| AnyCorp    |
| AnyOrgty   |
+------------+
```

stats 명령

`stats` 명령을 사용하여 검색 결과에서 집계를 계산합니다.

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

NULL/MISSING 값 처리

NULL/MISSING 값 처리

함수	NULL	MISSING
COUNT	계산되지 않음	계산되지 않음
SUM	무시	무시
AVG	무시	무시
MAX	무시	무시
MIN	무시	무시

구문

다음 구문을 사용합니다.

```
stats <aggregation>... [by-clause]
```

aggregation

- 필수.
- 필드에 적용되는 집계 함수입니다.

섹션별

- 선택 사항.
- 구문: `by [span-expression,] [field,]...`
- 집계 결과를 그룹화하기 위한 필드와 표현식을 지정합니다. 분류를 사용하면 필드와 표현식을 사용하여 집계 결과를 그룹화할 수 있습니다. 스칼라 함수, 집계 함수 및 스팬 표현식을 사용하여 특정 필드를 동일한 간격의 버킷으로 분할할 수 있습니다.
- 기본값:를 지정하지 않으면 stats 명령<by-clause>은 전체 결과 집합에 대한 집계를 나타내는 단일 행을 반환합니다.

스팬 표현식

- 선택 사항, 최대 1개.
- 구문: `span(field_expr, interval_expr)`
- 간격 표현식의 단위는 기본적으로 자연 단위입니다. 필드가 날짜 및 시간 유형 필드이고 간격이 날짜/시간 단위인 경우 간격 표현식에 단위를 지정합니다.
- 예를 들어 age 필드를 버킷으로 10년 분할하면 처럼 보입니다`span(age, 10)`. 타임스탬프 필드를 시간별 간격으로 분할하려면를 사용합니다`span(timestamp, 1h)`.

사용 가능한 시간 단위

범위 간격 단위
밀리초(ms)

범위 간격 단위
초(초)
분(m, 대/소문자 구분)
시간(h)
일(d)
주(w)
month(M, 대/소문자 구분)
분기(q)
연도(y)

집계 함수

COUNT

SELECT 문으로 검색된 행의 expr 수를 반환합니다.

예시

```
os> source=accounts | stats count();
fetched rows / total rows = 1/1
+-----------+
| count()   |
|-----------|
| 4         |
+-----------+
```

SUM

SUM(expr)를 사용하여 expr의 합계를 반환합니다.

예시

```
os> source=accounts | stats sum(age) by gender;
```

```
fetched rows / total rows = 2/2
+------------+----------+
| sum(age)   | gender   |
|------------+----------|
| 28         | F        |
| 101        | M        |
+------------+----------+
```

AVG

expr의 평균 값을 반환AVG(expr)하는 데 사용합니다.

예시

```
os> source=accounts | stats avg(age) by gender;
fetched rows / total rows = 2/2
+--------------------+----------+
| avg(age)           | gender   |
|--------------------+----------|
| 28.0               | F        |
| 33.666666666666664 | M        |
+--------------------+----------+
```

MAX

MAX(expr)를 사용하여 expr의 최대값을 반환합니다.

예시

```
os> source=accounts | stats max(age);
fetched rows / total rows = 1/1
+------------+
| max(age)   |
|------------|
| 36         |
+------------+
```

MIN

MIN(expr)를 사용하여 expr의 최소값을 반환합니다.

예시

```
os> source=accounts | stats min(age);
fetched rows / total rows = 1/1
+------------+
| min(age)   |
|------------|
| 28         |
+------------+
```

STDDEV_SAMP

expr의 샘플 표준 편차를 반환STDDEV_SAMP(expr)하는 데 사용합니다.

예시

```
os> source=accounts | stats stddev_samp(age);
fetched rows / total rows = 1/1
+--------------------+
| stddev_samp(age)   |
|--------------------|
| 3.304037933599835  |
+--------------------+
```

STDDEV_POP

expr의 모집단 표준 편차를 반환STDDEV_POP(expr)하는 데 사용합니다.

예시

```
os> source=accounts | stats stddev_pop(age);
fetched rows / total rows = 1/1
+--------------------+
| stddev_pop(age)    |
|--------------------|
| 2.***************** |
+--------------------+
```

테이크

TAKE(field [, size])를 사용하여 필드의 원래 값을 반환합니다. 값의 순서를 보장하지는 않습니다.

필드

- 필수.
- 필드는 텍스트 필드여야 합니다.

size

- 선택적 정수입니다.
- 값 수를 반환해야 합니다.
- 기본값은 10입니다.

예시

```
os> source=accounts | stats take(firstname);
fetched rows / total rows = 1/1
+-----------------------------+
| take(firstname)             |
|-----------------------------|
| [Jane, Mary, Nikki, Juan    |
+-----------------------------+
```

PERCENTILE 또는 PERCENTILE_APPROX

PERCENTILE(expr, percent) 또는 PERCENTILE_APPROX(expr, percent)를 사용하여 expr의 대략적인 백분위수 값을 지정된 백분율로 반환합니다.

%

- 숫자는 0에서 100 사이의 상수여야 합니다.

예시

```
os> source=accounts | stats percentile(age, 90) by gender;
fetched rows / total rows = 2/2
+-----------------------+----------+
| percentile(age, 90)   | gender   |
|-----------------------+----------|
| 28                    | F        |
```

```
| 36                    | M        |
+-----------------------+----------+
```

예제 1: 이벤트 수 계산

이 예제는 계정의 이벤트 수를 계산하는 방법을 보여줍니다.

```
os> source=accounts | stats count();
fetched rows / total rows = 1/1
+-----------+
| count()   |
|-----------|
| 4         |
+-----------+
```

예제 2: 필드의 평균 계산

이 예제에서는 모든 계정의 평균 수명을 계산하는 방법을 보여줍니다.

```
os> source=accounts | stats avg(age);
fetched rows / total rows = 1/1
+------------+
| avg(age)   |
|------------|
| 32.25      |
+------------+
```

예제 3: 그룹별 필드의 평균 계산

이 예제는 성별로 그룹화된 모든 계정의 평균 연령을 계산하는 방법을 보여줍니다.

```
os> source=accounts | stats avg(age) by gender;
fetched rows / total rows = 2/2
+--------------------+----------+
| avg(age)           | gender   |
|--------------------+----------|
| 28.0               | F        |
| 33.666666666666664 | M        |
+--------------------+----------+
```

예제 4: 그룹별 필드의 평균, 합계 및 개수 계산

이 예제에서는 성별별로 그룹화된 모든 계정의 평균 연령, 합계 연령 및 이벤트 수를 계산하는 방법을 보여줍니다.

```
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        |
+--------------------+------------+-----------+----------+
```

예제 5: 필드의 최대값 계산

이 예제에서는 모든 계정의 최대 수명을 계산합니다.

```
os> source=accounts | stats max(age);
fetched rows / total rows = 1/1
+------------+
| max(age)   |
|------------|
| 36         |
+------------+
```

예제 6: 그룹별 필드의 최대 및 최소 계산

이 예제에서는 성별별로 그룹화된 모든 계정의 최대 및 최소 연령 값을 계산합니다.

```
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        |
+------------+------------+----------+
```

예제 7: 필드의 고유 개수 계산

필드의 고유 값 수를 가져오려면 대신 DISTINCT_COUNT (또는 DC) 함수를 사용할 수 있습니다COUNT. 이 예제에서는 모든 계정의 성별 필드 수와 고유 수를 모두 계산합니다.

```
os> source=accounts | stats count(gender), distinct_count(gender);
fetched rows / total rows = 1/1
+-----------------+--------------------------+
| count(gender)   | distinct_count(gender)   |
|-----------------+--------------------------|
| 4               | 2                        |
+-----------------+--------------------------+
```

예제 8: 범위별로 개수 계산

이 예제에서는 연령을 10년 간격으로 가져옵니다.

```
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         |
+--------------+------------+
```

예제 9: 성별 및 범위별로 개수 계산

이 예제에서는 5년의 성별 및 연령 범위별로 그룹화된 레코드를 계산합니다.

```
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        |
+-------+------------+----------+
```

스팬 표현식은 명령에 지정된 순서에 관계없이 항상 첫 번째 그룹화 키로 표시됩니다.

```
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        |
+-------+------------+----------+
```

예제 10: 성별 및 범위별로 개수를 계산하고 이메일 목록 가져오기

이 예제에서는 10년 간격으로 연령을 구하고 성별별로 그룹을 구합니다. 또한 각 행에 대해 최대 5개의 이메일 목록을 가져옵니다.

```
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        |
+-------+-----------------------------------------------------+------------+----------+
```

예제 11: 필드의 백분위수 계산

이 예제에서는 모든 계정의 백분위수 90세를 계산하는 방법을 보여줍니다.

```
os> source=accounts | stats percentile(age, 90);
fetched rows / total rows = 1/1
+-----------------------+
| percentile(age, 90)   |
|-----------------------|
| 36                    |
+-----------------------+
```

예제 12: 그룹별로 필드의 백분위수 계산

이 예제에서는 성별별로 모든 계정 그룹의 백분위수 90세를 계산하는 방법을 보여줍니다.

```
os> source=accounts | stats percentile(age, 90) by gender;
fetched rows / total rows = 2/2
+-----------------------+----------+
| percentile(age, 90)   | gender   |
```

```
|----------------------+----------|
| 28                   | F        |
| 36                   | M        |
+----------------------+----------+
```

예제 13: 성별 및 범위별로 백분위수 계산

이 예제에서는 10년 간격으로 백분위수 90세를 구하고 성별로 그룹화합니다.

```
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)`
```

스팬이 있는 집계

```
- `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`
```

기간 범위를 사용한 집계(텀블 윈도우 함수)

```
- `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`
```

여러 수준별 집계 그룹

```
- `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`
```

하위 쿼리 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

subquery 명령을 사용하여 Piped Processing Language(PPL) 문 내에서 복잡한 중첩 쿼리를 수행합니다.

```
source=logs | where field in [ subquery source=events | where condition | fields
 field ]
```

이 예제에서 기본 검색(source=logs)은 하위 쿼리()의 결과를 기준으로 필터링됩니다source=events.

하위 쿼리 명령은 복잡한 데이터 분석을 위해 여러 수준의 중첩을 지원합니다.

중첩 하위 쿼리 예제

```
source=logs | where id in [ subquery source=users | where user in [ subquery
 source=actions | where action="login" | fields user] | fields uid ]
```

InSubquery 사용

- `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 ]` (하위 쿼리로 필터링 검색)
- `source = outer a not in [ source = inner | fields b ]` (하위 쿼리로 필터링 검색)
- `source = outer | where a in [ source = inner1 | where b not in [ source = inner2 | fields c ] | fields b ]` (중첩)
- `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` (조인 필터로)

IN-Subquery PPL을 사용한 SQL 마이그레이션 예제

TPC-H Q4(집계를 사용한 하위 쿼리)

```
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
```

PPL InSubquery 쿼리로 다시 작성:

```
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(하위 쿼리에 중첩됨)

```
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
```

PPL InSubquery 쿼리로 다시 작성:

```
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 사용

가정: a, b는 테이블 외부의 필드, c, d는 테이블 내부 필드, ef는 테이블 내부 필드2.

- `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 ]` (하위 쿼리로 필터링 검색)
- `source = outer not exists [ source = inner | where a = c ]` (하위 쿼리로 필터링 검색)
- `source = table as t1 exists [ source = table as t2 | where t1.a = t2.a ]` (테이블 별칭은 기존 하위 쿼리에서 유용함)
- `source = outer | where exists [ source = inner1 | where a = c and exists [ source = inner2 | where c = e ] ]` (중첩)
- `source = outer | where exists [ source = inner1 | where a = c | where exists [ source = inner2 | where c = e ] ]` (중첩)
- `source = outer | where exists [ source = inner | where c > 10 ]` (관련 없음)
- `source = outer | where not exists [ source = inner | where c > 10 ]` (관련 없음)
- `source = outer | where exists [ source = inner ] | eval l = "nonEmpty" | fields l` (상관 관계가 없는 특별한 존재)

ScalarSubquery 사용

가정: a, b 테이블 외부의 필드, c, 테이블 내부 d 필드, e, 테이블 중첩 f 필드

상관 관계가 없는 스칼라 하위 쿼리

선택에서:

- `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`

에서:

- `source = outer | where a > [ source = inner | stats min(c) ] | fields a`

검색 필터에서:

- `source = outer a > [ source = inner | stats min(c) ] | fields a`

상관관계가 있는 스칼라 하위 쿼리

선택에서:

- `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`

에서:

- `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`

검색 필터에서:

- `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`

중첩된 스칼라 하위 쿼리

- `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 ]`

(관계) 하위 쿼리

`InSubquery`, `ExistsSubquery` 및 `ScalarSubquery`는 모두 하위 쿼리 표현식입니다. 그러나 `RelationSubquery`는 하위 쿼리 표현식이 아니며 Join 또는 From 절에 일반적으로 사용되는 하위 쿼리 계획입니다.

- `source = table1 | join left = l right = r [ source = table2 | where d > 10 | head 5 ]` (오른쪽 조인에서 하위 쿼리)
- `source = [ source = table1 | join left = l right = r [ source = table2 | where d > 10 | head 5 ] | stats count(a) by b ] as outer | head 1`

추가 컨텍스트

`InSubquery`, `ExistsSubquery`및 `ScalarSubquery`는 `where` 절 및 검색 필터에 일반적으로 사용되는 하위 쿼리 표현식입니다.

여기서 명령은 다음과 같습니다.

```
| where <boolean expression> | ...
```

검색 필터:

```
search source=* <boolean expression> | ...
```

부울 표현식에 하위 쿼리 표현식을 사용할 수 있습니다.

```
| where orders.order_id in [ source=returns | where return_reason="damaged" | field
 order_id ]
```

`orders.order_id in [ source=... ]`는 입니다`<boolean expression>`.

일반적으로 이러한 종류의 하위 쿼리 절에는 InSubquery 표현식이라는 이름이 지정됩니다. 입니다`<boolean expression>`.

조인 유형이 다른 하위 쿼리

`ScalarSubquery`를 사용하는 예:

```
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 ]
```

InSubquery, ExistsSubquery 및 ScalarSubquery와 달리 RelationSubquery는 하위 쿼리 표현식이 아닙니다. 대신 하위 쿼리 계획입니다.

```
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 ...
```

상위 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

top 명령을 사용하여 필드 목록에서 모든 필드의 가장 일반적인 값 튜플을 찾습니다.

구문

다음 구문을 사용합니다.

```
top [N] <field-list> [by-clause] top_approx [N] <field-list> [by-clause]
```

N

- 반환할 결과 수.
- 기본값: 10

필드 목록

- 필수.
- 쉼표로 구분된 필드 이름 목록입니다.

분류별

- 선택 사항.
- 결과를 그룹화할 하나 이상의 필드입니다.

상단_대략

- HyperLogLog++ 알고리즘별 예상 카디널리티를 사용하여 (n) 상위 필드의 대략적인 개수입니다.

예제 1: 필드에서 가장 일반적인 값 찾기

이 예제에서는 모든 계정에서 가장 일반적인 성별을 찾습니다.

PPL 쿼리:

```
os> source=accounts | top gender;
os> source=accounts | top_approx gender;
fetched rows / total rows = 2/2
+----------+
| gender   |
```

```
|----------|
| M        |
| F        |
+----------+
```

예제 2: 필드에서 가장 일반적인 값 찾기(1로 제한)

이 예제에서는 모든 계정에 대해 가장 일반적인 단일 성별을 찾습니다.

PPL 쿼리:

```
os> source=accounts | top_approx 1 gender;
fetched rows / total rows = 1/1
+----------+
| gender   |
|----------|
| M        |
+----------+
```

예제 3: 성별별로 그룹화된 가장 일반적인 값 찾기

이 예제에서는 성별별로 그룹화된 모든 계정의 가장 일반적인 연령을 찾습니다.

PPL 쿼리:

```
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    |
+----------+-------+
```

추세선 명령

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

trendline 명령을 사용하여 필드의 이동 평균을 계산합니다.

구문

다음 구문 사용

```
TRENDLINE [sort <[+|-] sort-field>] SMA(number-of-datapoints, field) [AS alias]
 [SMA(number-of-datapoints, field) [AS alias]]...
```

[+|-]

- 선택 사항.
- 더하기 [+]는 NULL/MISSING 값이 먼저 있는 오름차순을 나타냅니다.
- 빼기 [-]는 NULL/MISSING 값이 마지막인 내림차순을 나타냅니다.
- 기본값: NULL/MISSING 값이 먼저 있는 오름차순입니다.

정렬 필드

- 정렬을 사용할 때 필수입니다.
- 정렬에 사용되는 필드입니다.

number-of-datapoints

- 필수.
- 이동 평균을 계산하는 데이터 포인트 수입니다.
- 0보다 커야 합니다.

필드

- 필수.
- 이동 평균을 계산해야 하는 필드의 이름입니다.

별칭

- 선택 사항.
- 이동 평균을 포함하는 결과 열의 이름입니다.

Simple Moving Average(SMA) 유형만 지원됩니다. 다음과 같이 계산됩니다.

```
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
```

예제 1: 온도의 시간 단위에 대한 단순 이동 평균 계산

이 예제에서는 두 개의 데이터 포인트를 사용하여 온도에 대한 단순 이동 평균을 계산합니다.

PPL 쿼리:

```
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|
+-----------+---------+--------------------+----------+
```

예제 2: 정렬을 사용하여 온도의 시간 단위에 대한 단순 이동 평균 계산

이 예제에서는 device-id별로 내림차순으로 정렬된 2개 및 3개의 데이터 포인트를 사용하여 온도에 대한 단순 이동 평균 2개를 계산합니다.

PPL 쿼리:

```
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|
+-----------+---------+--------------------+------------+------------------+
```

여기서 명령은

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`where` 명령은 부울 표현식을 사용하여 검색 결과를 필터링합니다. 부울 표현식이 true로 평가될 때만 결과를 반환합니다.

구문

다음 구문을 사용합니다.

```
where <boolean-expression>
```

부울 표현식

- 선택 사항.
- 부울 값으로 평가할 수 있는 모든 표현식입니다.

예제 1: 조건을 사용하여 결과 세트 필터링

이 예제는 특정 조건을 충족하는 계정 인덱스에서 문서를 가져오는 방법을 보여줍니다.

PPL 쿼리:

```
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        |
+------------------+----------+
```

추가 예제

논리적 조건이 있는 필터

- `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` - 참고:이 값은 >= 1 및 <= 4, 즉 [1, 4]를 반환합니다.
- `source = table | where b not between '2024-09-10' and '2025-09-10'` - 참고: b >= '**********' 및 b <= '2025-09-10'을 반환합니다.
- `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
```

필드 요약

> **Note**
>
> 이 PPL 명령을 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “명령”.

`fieldsummary` 명령을 사용하여 각 필드(개수, 고유 개수, 최소, 최대, 평균, stddev, 평균)에 대한 기본 통계를 계산하고 각 필드의 데이터 유형을 결정합니다. 이 명령은 이전 파이프와 함께 사용할 수 있으며 이를 고려합니다.

구문

다음 구문을 사용합니다. CloudWatch Logs 사용 사례의 경우 쿼리에서 하나의 필드만 지원됩니다.

```
... | fieldsummary <field-list> (nulls=true/false)
```

includefields

- 통계를 사용하여 통합 결과 집합으로 수집할 모든 열의 목록입니다.

NULL

- 선택 사항.

- true로 설정된 경우 집계 계산에 null 값을 포함합니다(숫자 값의 경우 null을 0으로 대체).

예시 1

PPL 쿼리:

```
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"  |
+------------------+-------------+------------+------------+------------+------------
+------------+------------+----------------|
```

예시 2

PPL 쿼리:

```
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"|
+------------------+-------------+------------+------------+------------+------------
+------------+------------+----------------|
```

명령 확장

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

expand 명령을 사용하여 Array<Any> 또는 Map<Any> 유형의 필드를 평면화하여 각 요소 또는 키-값 페어에 대해 개별 행을 생성합니다.

구문

다음 구문을 사용합니다.

```
expand <field> [As alias]
```

필드

- 확장(확장)할 필드입니다.
- 필드는 지원되는 유형이어야 합니다.

별칭

- 선택 사항.
- 원래 필드 이름 대신 사용할 이름입니다.

사용 지침

expand 명령은 지정된 배열 또는 맵 필드의 각 요소에 대한 행을 생성합니다. 여기서

- 배열 요소는 개별 행이 됩니다.
- 맵 키-값 페어는 별도의 행으로 나뉘며 각 키-값은 행으로 표시됩니다.
- 별칭이 제공되면 분해된 값이 원래 필드 이름 대신 별칭 아래에 표시됩니다.

이 명령을 통계, 평가 및 구문 분석과 같은 다른 명령과 함께 사용하여 확장 후 데이터를 조작하거나 추출할 수 있습니다.

예시

- `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`

확장 명령을 eval, stats 등과 같은 다른 명령과 함께 사용할 수 있습니다. 여러 확장 명령을 사용하면 각 복합 배열 또는 맵 내의 모든 내부 요소의 데카르트 곱이 생성됩니다.

효과적인 SQL 푸시다운 쿼리

확장 명령은 LATERAL VIEW 폭발을 사용하여 동등한 SQL 작업으로 변환되므로 SQL 쿼리 수준에서 배열 또는 맵을 효율적으로 폭발할 수 있습니다.

```
SELECT customer exploded_productId
FROM table
LATERAL VIEW explode(productId) AS exploded_productId
```

explode 명령은 다음 기능을 제공합니다.

- 새 열을 반환하는 열 작업입니다.
- 분해된 열의 모든 요소에 대해 새 행을 생성합니다.
- 내부 null은 분해된 필드의 일부로 무시됩니다(널에 대한 행이 생성되거나 분해되지 않음).

PPL 함수

주제

- PPL 조건 함수
- PPL 암호화 해시 함수
- PPL 날짜 및 시간 함수
- PPL 표현식
- PPL IP 주소 함수
- PPL JSON 함수
- PPL Lambda 함수
- PPL 수학 함수
- PPL 문자열 함수
- PPL 유형 변환 함수

PPL 조건 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

ISNULL

설명: 필드가 null인 경우 true를 `isnull(field)` 반환합니다.

인수 유형:

- 지원되는 모든 데이터 형식입니다.

반환 유형:

- BOOLEAN

예:

```
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

설명: 필드가 null이 아닌 경우 true를 `isnotnull(field)` 반환합니다.

인수 유형:

- 지원되는 모든 데이터 형식입니다.

반환 유형:

- BOOLEAN

예:

```
os> source=accounts | where not isnotnull(employer) | fields account_number, employer
fetched rows / total rows = 1/1
+------------------+------------+
| account_number   | employer   |
|------------------+------------|
| 18               | null       |
+------------------+------------+
```

exists

예:

```
os> source=accounts | where exists(email) | fields account_number, email
fetched rows / total rows = 1/1
```

IFNULL

설명:가 null`field2`이면 `field1`를 `ifnull(field1, field2)` 반환합니다.

인수 유형:

- 지원되는 모든 데이터 형식입니다.
- 두 파라미터의 유형이 서로 다른 경우 함수는 의미 체계 검사에 실패합니다.

반환 유형:

- 임의

예:

```
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

설명: 두 파라미터가 동일한 경우 `nullif(field1, field2)` null을 반환하고, 그렇지 않으면 field1을 반환합니다.

인수 유형:

- 지원되는 모든 데이터 형식입니다.
- 두 파라미터의 유형이 서로 다른 경우 함수는 의미 체계 검사에 실패합니다.

반환 유형:

- 임의

예:

```
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

설명: 조건이 true`expr1`이면를 `if(condition, expr1, expr2)` 반환하고, 그렇지 않으면를 반환합니다`expr2`.

인수 유형:

- 지원되는 모든 데이터 형식입니다.
- 두 파라미터의 유형이 서로 다른 경우 함수는 의미 체계 검사에 실패합니다.

반환 유형:

- 임의

예:

```
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 암호화 해시 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

MD5

MD5는 MD5 다이제스트를 계산하고 값을 32자 16진수 문자열로 반환합니다.

사용량: `md5('hello')`

인수 유형:

- STRING

반환 유형:

- STRING

예:

```
os> source=people | eval `MD5('hello')` = MD5('hello') | fields `MD5('hello')`
fetched rows / total rows = 1/1
+----------------------------------+
| MD5('hello')                     |
|----------------------------------|
| <32 character hex string>        |
+----------------------------------+
```

SHA1

SHA1은 SHA-1의 16진수 문자열 결과를 반환합니다.

사용량: `sha1('hello')`

인수 유형:

- STRING

반환 유형:

- STRING

예:

```
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

SHA2는 SHA-2 해시 함수 패밀리(SHA-224, SHA-256, SHA-384 및 SHA-512)의 16진수 문자열 결과를 반환합니다. numBits는 결과의 원하는 비트 길이를 나타내며, 값은 224, 256, 384, 512여야 합니다.

사용량:

- `sha2('hello',256)`
- `sha2('hello',512)`

인수 유형:

- 문자열, 정수

반환 유형:

- STRING

예:

```
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 날짜 및 시간 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

DAY

사용량:는 날짜의 날짜를 1~31 범위로 DAY(date) 추출합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: DAYOFMONTH, DAY_OF_MONTH

예:

```
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

사용량:는 날짜의 날짜를 1~31 범위로 DAYOFMONTH(date) 추출합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: DAY, DAY_OF_MONTH

예:

```
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

사용량:는 날짜의 날짜를 1~31 범위로 DAY_OF_MONTH(DATE) 추출합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: DAY, DAYOFMONTH

예:

```
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

사용량: 날짜의 평일 인덱스를 DAYOFWEEK(DATE) 반환합니다(1 = 일요일, 2 = 월요일, ..., 7 = 토요일).

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: DAY_OF_WEEK

예:

```
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

사용량: 날짜의 평일 인덱스를 DAY_OF_WEEK(DATE) 반환합니다(1 = 일요일, 2 = 월요일, ..., 7 = 토요일).

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: DAYOFWEEK

예:

```
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

사용량: 1~366 범위의 날짜에 대한 연도의 날짜를 DAYOFYEAR(DATE) 반환합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: DAY_OF_YEAR

예:

```
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

사용량: 1~366 범위의 날짜에 대한 연도의 날짜를 DAY_OF_YEAR(DATE) 반환합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: DAYOFYEAR

예:

```
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

사용량:는 월요일, 화요일, 수요일, 목요일, 금요일, 토요일, 일요일을 포함한 날짜의 평일 이름을 DAYNAME(DATE) 반환합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: STRING

예:

```
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

사용량: 타임스탬프 또는 문자열 값으로 지정된 인수의 표현을 FROM_UNIXTIME 반환합니다. 이 함수는 UNIX_TIMESTAMP 함수의 역변환을 수행합니다.

두 번째 인수를 제공하는 경우는 이를 FROM_UNIXTIME 사용하여 DATE_FORMAT 함수와 유사한 결과의 형식을 지정합니다.

타임스탬프가 1970-01-01 00:00:00~3001-01-18 23:59:59.999999(0~32536771199.999999 epoch 시간) 범위를 벗어나는 경우 함수는를 반환합니다NULL.

인수 유형: DOUBLE, STRING

반환 유형 맵:

DOUBLE -> 타임스탬프

DOUBLE, STRING -> STRING

예:

```
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

사용량: 시간에 대한 시간 값을 HOUR(TIME) 추출합니다.

표준 시간대와 달리이 함수의 시간 값은 23보다 큰 범위를 가질 수 있습니다. 따라서의 반환 값은 23보다 클 HOUR(TIME) 수 있습니다.

인수 유형: STRING/TIME/TIMESTAMP

반환 유형: INTEGER

동의어: HOUR_OF_DAY

예:

```
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

사용량:는 지정된 시간에서 시간 값을 HOUR_OF_DAY(TIME) 추출합니다.

표준 시간대와 달리이 함수의 시간 값은 23보다 큰 범위를 가질 수 있습니다. 따라서의 반환 값은 23보다 클 HOUR_OF_DAY(TIME) 수 있습니다.

인수 유형: STRING/TIME/TIMESTAMP

반환 유형: INTEGER

동의어: HOUR

예:

```
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

사용량: 지정된 날짜 인수에 대한 DATE 값으로 해당 월의 마지막 날짜를 LAST_DAY 반환합니다.

인수 유형: DATE/STRING/TIMESTAMP/TIME

반환 유형: DATE

예:

```
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

사용량:는의 동의어LOCALTIMESTAMP()입니다NOW().

예:

```
> source=people | eval `LOCALTIMESTAMP()` = LOCALTIMESTAMP() | fields
 `LOCALTIMESTAMP()`
fetched rows / total rows = 1/1
+---------------------+
| LOCALTIMESTAMP()    |
|---------------------|
| 2022-08-02 15:54:19 |
+---------------------+
```

LOCALTIME

사용량: LOCALTIME()는의 동의어입니다NOW().

예:

```
> source=people | eval `LOCALTIME()` = LOCALTIME() | fields `LOCALTIME()`
fetched rows / total rows = 1/1
+---------------------+
| LOCALTIME()         |
|---------------------|
| 2022-08-02 15:54:19 |
+---------------------+
```

MAKE_DATE

사용량: 지정된 연도, 월 및 일 값을 기준으로 날짜 값을 MAKE_DATE 반환합니다. 모든 인수는 정수로 반올림됩니다.

사양: 1. MAKE_DATE(정수, 정수, 정수) -> 날짜

인수 유형: INTEGER, INTEGER, INTEGER

반환 유형: DATE

예:

```
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

사용량: 지정된 시간의 분 구성 요소를 0~59 범위의 정수로 MINUTE(TIME) 반환합니다.

인수 유형: STRING/TIME/TIMESTAMP

반환 유형: INTEGER

동의어: MINUTE_OF_HOUR

예:

```
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

사용량: 지정된 시간의 분 구성 요소를 0~59 범위의 정수로 MINUTE_OF_HOUR(TIME) 반환합니다.

인수 유형: STRING/TIME/TIMESTAMP

반환 유형: INTEGER

동의어: MINUTE

예:

```
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

사용량:는 지정된 날짜의 월을 1~12 범위의 정수로 MONTH(DATE) 반환합니다(여기서 1은 1월을 나타내고 12는 12월을 나타냄).

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: MONTH_OF_YEAR

예:

```
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

사용량:는 지정된 날짜의 월을 1~12 범위의 정수로 MONTHNAME(DATE) 반환합니다(여기서 1은 1월을 나타내고 12는 12월을 나타냄).

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: MONTH_OF_YEAR

예:

```
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

사용량:는 지정된 날짜의 월을 1~12 범위의 정수로 MONTH_OF_YEAR(DATE)반환합니다(여기서 1은 1월을 나타내고 12는 12월을 나타냄).

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

동의어: MONTH

예:

```
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

사용량: 현재 날짜 및 시간을 'YYYY-MM-DD hh:mm:ss' 형식의 TIMESTAMP 값으로 NOW 반환합니다. 값은 클러스터 시간대로 표시됩니다.

> **Note**
>
> NOW()는 문이 실행되기 시작한 시간을 나타내는 일정한 시간을 반환합니다. 이는 정확한 실행 시간을 반환SYSDATE()하는와 다릅니다.

반환 유형: TIMESTAMP

사양: NOW() -> TIMESTAMP

예:

```
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

사용량: 지정된 날짜의 분기를 1~4 범위의 정수로 QUARTER(DATE) 반환합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

예:

```
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

사용량:는 지정된 시간의 두 번째 구성 요소를 0~59 범위의 정수로 SECOND(TIME) 반환합니다.

인수 유형: STRING/TIME/TIMESTAMP

반환 유형: INTEGER

동의어: SECOND_OF_MINUTE

예:

```
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

사용량:는 지정된 시간의 두 번째 구성 요소를 0~59 범위의 정수로 SECOND_OF_MINUTE(TIME) 반환합니다.

인수 유형: STRING/TIME/TIMESTAMP

반환 유형: INTEGER

동의어: SECOND

예:

```
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

사용량: 지정된 날짜에서 두 번째 인수(예: DATE 또는 DAYS)를 SUBDATE(DATE, DAYS)뺍니다.

인수 유형: DATE/TIMESTAMP, LONG

반환 유형 맵: (DATE, LONG) -> DATE

주석: ADDDATE

예:

```
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

사용량: 현재 날짜 및 시간을 'YYYY-MM-DD hh:mm:ss.nnnnnn' 형식의 TIMESTAMP 값으로 SYSDATE() 반환합니다.

SYSDATE()는 실행되는 정확한 시간을 반환합니다. 이는 문이 실행되기 시작한 시점을 나타내는 일정한 시간을 반환하는 NOW()와 다릅니다.

선택적 인수 유형: INTEGER(0~6) - 반환 값에서 소수 초의 자릿수를 지정합니다.

반환 유형: TIMESTAMP

예:

```
os> source=people | eval `SYSDATE()` = SYSDATE() | fields `SYSDATE()`
fetched rows / total rows = 1/1
+----------------------------+
| SYSDATE()                  |
|----------------------------|
| 2022-08-02 15:39:05.123456 |
+----------------------------+
```

TIMESTAMP

사용량: 입력 문자열을 타임스탬프expr로 사용하여 타임스탬프 유형을 TIMESTAMP(EXPR) 구성합니다.

단일 인수를 사용하여는 입력에서 타임스탬프를 TIMESTAMP(expr) 구성합니다. expr가 문자열인 경우 타임스탬프로 해석됩니다. 문자열이 아닌 인수의 경우 함수는 UTC 시간대expr를 사용하여 타임스탬프로 캐스팅합니다. expr가 TIME 값인 경우 함수는 캐스팅 전 오늘 날짜를 적용합니다.

두 인수와 함께 사용할 경우는 날짜 또는 타임스탬프 표현식(expr2)에 시간 표현식(expr1)을 TIMESTAMP(expr1, expr2) 추가하고 결과를 타임스탬프 값으로 반환합니다.

인수 유형: STRING/DATE/TIME/TIMESTAMP

반환 유형 맵:

(STRING/DATE/TIME/타임스탬프) -> 타임스탬프

(STRING/DATE/TIME/TIMESTAMP, STRING/DATE/TIME/TIMESTAMP) -> 타임스탬프

예:

```
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

사용량: 지정된 날짜 인수를 Unix 시간(1970년 초에 시작된 Epoch 이후 초)으로 UNIX_TIMESTAMP 변환합니다. 인수가 제공되지 않으면 현재 Unix 시간을 반환합니다.

날짜 인수는 DATE, TIMESTAMP 문자열 또는 , , 또는 형식 중 하나의 숫자일 수 있습니다YYMMDDYYMMDDhhmmssYYYYMMDDYYYYMMDDhhmmss. 인수에 시간 구성 요소가 포함된 경우 선택적으로 소수 초를 포함할 수 있습니다.

인수가 잘못된 형식이거나 1970-01-01 00:00:00~3001-01-18 23:59:59.999999(에포크 시간 기준 0~32536771199.999999) 범위를 벗어나는 경우 함수는를 반환합니다NULL.

함수는 DATE, TIMESTAMP또는를 인수 유형DOUBLE으로 수락하거나 인수를 수락하지 않습니다. 항상 Unix 타임스탬프를 나타내는 DOUBLE 값을 반환합니다.

역변환의 경우 FROM_UNIXTIME 함수를 사용할 수 있습니다.

인수 유형: <NONE>/DOUBLE/DATE/TIMESTAMP

반환 유형: DOUBLE

예:

```
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

사용량: 지정된 날짜의 주 번호를 WEEK(DATE) 반환합니다.

인수 유형: DATE/TIMESTAMP/STRING

반환 유형: INTEGER

동의어: WEEK_OF_YEAR

예:

```
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

사용량: 날짜의 평일 인덱스를 WEEKDAY(DATE) 반환합니다(0 = 월요일, 1 = 화요일, ..., 6 = 일요일).

함수와 비슷dayofweek하지만 매일 다른 인덱스를 반환합니다.

인수 유형: STRING/DATE/TIME/TIMESTAMP

반환 유형: INTEGER

예:

```
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

사용량: 지정된 날짜의 주 번호를 WEEK_OF_YEAR(DATE) 반환합니다.

인수 유형: DATE/TIMESTAMP/STRING

반환 유형: INTEGER

동의어: WEEK

예:

```
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

사용량:는 1000~9999 범위의 날짜로 연도를 반환하거나 "0" 날짜로 0을 YEAR(DATE) 반환합니다.

인수 유형: STRING/DATE/TIMESTAMP

반환 유형: INTEGER

예:

```
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

사용량:는 지정된 날짜에 지정된 간격을 DATE_ADD(date, INTERVAL expr unit) 추가합니다.

인수 유형: DATE, INTERVAL

반환 유형: DATE

약어: DATE_SUB

예:

```
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

사용량: 날짜에서 간격 expr를 DATE_SUB(date, INTERVAL expr unit) 뺍니다.

인수 유형: DATE, INTERVAL

반환 유형: DATE

약어: DATE_ADD

예:

```
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

사용량: 지정된 날짜에 지정된 시간 간격을 추가한 후 TIMESTAMP 값을 반환합니다.

인수:

- 간격: 간격(초, 분, 시간, 일, 주, 월, 분기, 연도)
- 정수: INTEGER
- 날짜: DATE, TIMESTAMP 또는 STRING

를 날짜 인수STRING로 제공하는 경우 유효한 로 형식을 지정합니다TIMESTAMP. 함수는 DATE 인수를 로 자동 변환합니다TIMESTAMP.

예:

```
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

사용량: 지정된 간격 단위로 시작 날짜와 종료 날짜/시간 간의 차이를 `TIMESTAMPDIFF(interval, start, end)` 반환합니다.

인수:

- 간격: 간격(초, 분, 시간, 일, 주, 월, 분기, 연도)
- 시작: DATE, TIMESTAMP 또는 STRING
- 종료: DATE, TIMESTAMP 또는 STRING

함수는 적절한 `TIMESTAMP` 경우 인수를 로 자동 변환합니다. `STRING` 인수를 유효한 로 포맷`TIMESTAMP`합니다.

예:

```
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

사용량: 현재 UTC 타임스탬프를 'YYYY-MM-DD hh:mm:ss'의 값으로 `UTC_TIMESTAMP` 반환합니다.

반환 유형: TIMESTAMP

사양: UTC_TIMESTAMP() -> TIMESTAMP

예:

```
> 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

사용량: 현재 현지 시간대를 CURRENT_TIMEZONE 반환합니다.

반환 유형: STRING

예:

```
> source=people | eval `CURRENT_TIMEZONE()` = CURRENT_TIMEZONE() | fields
 `CURRENT_TIMEZONE()`
fetched rows / total rows = 1/1
+------------------------+
| CURRENT_TIMEZONE()     |
|------------------------|
| America/Chicago        |
+------------------------+
```

PPL 표현식

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

표현식, 특히 값 표현식은 스칼라 값을 반환합니다. 표현식의 유형과 형식은 다릅니다. 예를 들어 원자 표현식과 그 위에 구축된 산술, 조건자 및 함수 표현식으로 리터럴 값이 있습니다. Filter 및 Stats 명령에서 산술 표현식을 사용하는 등 다양한 절에서 표현식을 사용할 수 있습니다.

연산자

산술 표현식은 다음과 같이 숫자 리터럴과 이진 산술 연산자로 구성된 표현식입니다.

1. +:를 추가합니다.
2. -: 빼기.
3. *: 곱하기.
4. /: 나누기(정수의 경우 결과는 소수 부분이 삭제된 정수)
5. %: Modulo(정수에만 사용, 결과는 나머지 분할)

우선 순위

괄호를 사용하여 산술 연산자의 우선 순위를 제어합니다. 그렇지 않으면 우선 순위가 더 높은 연산자가 먼저 수행됩니다.

유형 변환

연산자 서명을 조회할 때 암시적 유형 변환이 수행됩니다. 예를 들어 정수+는 서명과 일치+(double,double)하여 실수가 됩니다. 이 규칙은 함수 호출에도 적용됩니다.

다양한 유형의 산술 표현식의 예:

```
os> source=accounts | where age > (25 + 5) | fields age ;
fetched rows / total rows = 3/3
+-------+
| age   |
|-------|
| 32    |
| 36    |
| 33    |
+-------+
```

조건자 연산자

조건자 연산자는 true로 평가되는 표현식입니다. MISSING 및 NULL 값 비교는 다음 규칙을 따릅니다.

- MISSING 값은 MISSING 값과 같을 뿐이며 다른 값보다 작습니다.
- NULL 값은 NULL 값과 같고 값보다 크지만 다른 모든 MISSING 값보다 작습니다.

연산자

조건자 연산자

명칭	설명
>	보다 큼 연산자
>=	보다 크거나 같은 연산자
<	연산자 미만
!=	같지 않은 연산자
<=	이하 연산자
=	같음 연산자
LIKE	간단한 패턴 일치
IN	NULL 값 테스트
AND	AND 연산자
OR	OR 연산자
XOR	XOR 연산자
NOT	NOT NULL 값 테스트

날짜/시간을 비교할 수 있습니다. 서로 다른 날짜/시간 유형(예: DATE 및 TIME)을 비교할 때 둘 다 로 변환됩니다DATETIME. 변환에는 다음 규칙이 적용됩니다.

- TIME는 오늘 날짜에 적용됩니다.
- DATE는 자정에 해석됩니다.

기본 조건자 연산자

비교 연산자의 예:

```
os> source=accounts | where age > 33 | fields age ;
fetched rows / total rows = 1/1
+-------+
```

```
| age   |
|-------|
| 36    |
+-------+
```

IN

값 목록의 IN 연산자 테스트 필드의 예:

```
os> source=accounts | where age in (32, 33) | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 32    |
| 33    |
+-------+
```

OR

OR 연산자의 예:

```
os> source=accounts | where age = 32 OR age = 33 | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 32    |
| 33    |
+-------+
```

NOT

NOT 연산자의 예:

```
os> source=accounts | where age not in (32, 33) | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 36    |
| 28    |
```

```
+-------+
```

PPL IP 주소 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

`CIDRMATCH`

사용량: 지정된 IP 주소가 지정된 cidr 범위 내에 있는지 `CIDRMATCH(ip, cidr)` 확인합니다.

인수 유형:

- 문자열, 문자열
- 반환 유형: BOOLEAN

예:

```
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`는 IPv4 또는 IPv6 주소일 수 있습니다.

- `cidr`는 IPv4 또는 IPv6 블록일 수 있습니다.
- `ip` 및는 IPv4 또는 IPv6 둘 다여야 `cidr` 합니다.
- `ip` 및는 모두 유효하고 비어 있지 않거나 null이 아니어야 `cidr` 합니다.

PPL JSON 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

`JSON`

사용량: 문자열을 JSON 형식으로 구문 분석할 수 있는지 여부를 `json(value)` 평가합니다. 함수는 유효한 JSON인 경우 원본 문자열을 반환하고 유효하지 않은 경우 null을 반환합니다.

인수 유형: STRING

반환 유형: STRING/NULL. 유효한 JSON 객체 형식의 STRING 표현식입니다.

예:

```
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

사용량: 키-값 페어의 멤버에서 JSON 객체를 `json_object(<key>, <value>[, <key>, <value>]...)` 반환합니다.

인수 유형:

- <key>는 STRING이어야 합니다.
- <value>는 모든 데이터 형식일 수 있습니다.

반환 유형: JSON_OBJECT. 유효한 JSON 객체의 StructType 표현식입니다.

예:

```
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

사용량: 값 목록을 사용하여 JSON ARRAY를 `json_array(<value>...)` 생성합니다.

인수 유형:는 문자열, 숫자 또는 부울과 같은 모든 종류의 값일 <value> 수 있습니다.

반환 유형: ARRAY. 유효한 JSON 배열에 대해 지원되는 모든 데이터 유형의 배열입니다.

예:

```
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

사용량: 지정된 json 객체 값을 가진 JSON 문자열을 `to_json_string(jsonObject)` 반환합니다.

인수 유형: JSON_OBJECT

반환 유형: STRING

예:

```
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

사용량: 가장 바깥쪽 배열의 요소 수를 `array_length(jsonArray)` 반환합니다.

인수 유형: ARRAY. ARRAY 또는 JSON_ARRAY 객체입니다.

반환 유형: INTEGER

예:

```
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

사용량:는 지정된 JSON 경로를 기반으로 JSON 문자열에서 JSON 객체를 `json_extract(jsonStr, path)` 추출합니다. 입력 JSON 문자열이 유효하지 않은 경우 함수는 null을 반환합니다.

인수 유형: STRING, STRING

반환 유형: STRING

- 유효한 JSON 객체 형식의 STRING 표현식입니다.
- NULL 잘못된 JSON의 경우가 반환됩니다.

예:

```
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

사용량: 가장 바깥쪽 JSON 객체의 모든 키를 배열로 `json_keys(jsonStr)` 반환합니다.

인수 유형: STRING. 유효한 JSON 객체 형식의 STRING 표현식입니다.

반환 유형: ARRAY[STRING]. 함수는 다른 유효한 JSON 문자열, 빈 문자열 또는 잘못된 JSON에 NULL 대해를 반환합니다.

예:

```
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

사용량: JSON 문자열이 유효한 JSON 구문을 사용하고 TRUE 또는 FALSE를 반환하는지 여부를 `json_valid(jsonStr)` 평가합니다.

인수 유형: STRING

반환 유형: BOOLEAN

예:

```
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 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

EXISTS

사용량: Lambda 조건자가 배열의 하나 이상의 요소에 대해 보유하는지 여부를 `exists(array, lambda)` 평가합니다.

인수 유형: ARRAY, LAMBDA

반환 유형: BOOLEAN. 배열의 하나 이상의 요소가 Lambda 조건자를 충족하는 TRUE 경우를 반환하고, 그렇지 않으면를 반환합니다FALSE.

예:

```
 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

사용량: 지정된 Lambda 함수를 사용하여 입력 배열을 `filter(array, lambda)` 필터링합니다.

인수 유형: ARRAY, LAMBDA

반환 유형: ARRAY. Lambda 조건자를 충족하는 입력 배열의 모든 요소를 포함하는 ARRAY입니다.

예:

```
 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

사용량: Lambda `transform(array, lambda)` 변환 함수를 사용하여 배열의 요소를 변환합니다. 두 번째 인수는 바이너리 Lambda 함수를 사용하는 경우 요소의 인덱스를 의미합니다. 이는 기능 프로그래밍`map`의와 유사합니다.

인수 유형: ARRAY, LAMBDA

반환 유형: ARRAY. 입력 배열의 각 요소에 Lambda 변환 함수를 적용한 결과가 포함된 ARRAY입니다.

예:

```
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

사용량: lambda 함수를 적용하여 배열을 단일 값으로 `reduce(array, start, merge_lambda, finish_lambda)` 줄입니다. 함수는 merge_lambda를 시작 값과 모든 배열 요소에 적용한 다음 `finish_lambda`를 결과에 적용합니다.

인수 유형: ARRAY, ANY, LAMBDA, LAMBDA

반환 유형: ANY. Lambda 함수를 시작 값과 입력 배열에 적용한 최종 결과입니다.

예:

```
 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 수학 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

ABS

사용량: x의 절대값을 ABS(x) 계산합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: INTEGER/LONG/FLOAT/DOUBLE

예:

```
os> source=people | eval `ABS(-1)` = ABS(-1) | fields `ABS(-1)`
fetched rows / total rows = 1/1
+-----------+
| ABS(-1)   |
|-----------|
| 1         |
+-----------+
```

ACOS

사용량: x의 아크 코사인을 ACOS(x) 계산합니다. NULL x가 -1~1 범위에 속하지 않으면를 반환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `ACOS(0)` = ACOS(0) | fields `ACOS(0)`
fetched rows / total rows = 1/1
+--------------------+
| ACOS(0)            |
|--------------------|
| 1.5707963267948966 |
+--------------------+
```

ASIN

사용량: x의 아크 사인을 asin(x) 계산합니다. NULL x가 -1~1 범위에 속하지 않으면를 반환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `ASIN(0)` = ASIN(0) | fields `ASIN(0)`
fetched rows / total rows = 1/1
+-----------+
| ASIN(0)   |
|-----------|
| 0.0       |
+-----------+
```

ATAN

사용량:는 x의 아크 탄젠트를 ATAN(x) 계산합니다.는 두 인수의 기호에 따라 결과의 사분면이 결정된다는 점을 제외하고 y/x의 아크 탄젠트를 atan(y, x) 계산합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
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

사용량: 두 인수의 기호에 따라 결과의 사분면이 결정된다는 점을 제외하고 y/x의 아크 탄젠트를 ATAN2(y, x) 계산합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
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

사용량: 숫자의 큐브 루트를 CBRT 계산합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE:

INTEGER/LONG/FLOAT/DOUBLE -> DOUBLE

예:

```
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

사용량: CEILING 함수의 별칭입니다. CEILING(T)는 값 T의 상한을 사용합니다.

제한: 저장 시 IEEE 754 이중 유형이 소수를 표시할 때CEILING만 예상대로 작동합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: LONG

예:

```
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

사용량: 숫자 x를 기본에서 b 기본으로 CONV(x, a, b) 변환합니다.

인수 유형: x: STRING, a: INTEGER, b: INTEGER

반환 유형: STRING

예:

```
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

사용량: x의 코사인을 COS(x) 계산합니다. 여기서 x는 라디안 단위로 제공됩니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `COS(0)` = COS(0) | fields `COS(0)`
fetched rows / total rows = 1/1
+----------+
| COS(0)   |
|----------|
| 1.0      |
+----------+
```

COT

사용량: x의 코탄젠트를 COT(x) 계산합니다. x가 0과 같으면 out-of-range 오류가 반환됩니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `COT(1)` = COT(1) | fields `COT(1)`
fetched rows / total rows = 1/1
+--------------------+
| COT(1)             |
|--------------------|
| 0.6420926159343306 |
+--------------------+
```

CRC32

사용량: 순환 중복 검사 값을 CRC32 계산하고 부호 없는 32비트 값을 반환합니다.

인수 유형: STRING

반환 유형: LONG

예:

```
os> source=people | eval `CRC32('MySQL')` = CRC32('MySQL') | fields `CRC32('MySQL')`
fetched rows / total rows = 1/1
+------------------+
| CRC32('MySQL')   |
|------------------|
| 3259397556       |
+------------------+
```

DEGREES

사용량: x를 라디안에서 도로 DEGREES(x) 변환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
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

사용량: Euler의 번호를 E() 반환합니다.

반환 유형: DOUBLE

예:

```
os> source=people | eval `E()` = E() | fields `E()`
fetched rows / total rows = 1/1
+-------------------+
| E()               |
|-------------------|
| 2.718281828459045 |
+-------------------+
```

EXP

사용량: x의 출력으로 증가된 e를 EXP(x) 반환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `EXP(2)` = EXP(2) | fields `EXP(2)`
fetched rows / total rows = 1/1
+------------------+
| EXP(2)           |
|------------------|
| 7.38905609893065 |
+------------------+
```

FLOOR

사용량: FLOOR(T)는 값 T의 바닥을 사용합니다.

제한: 저장 시 IEEE 754 이중 유형이 소수를 표시할 때FLOOR만 예상대로 작동합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: LONG

예:

```
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

사용량: x의 자연 로그를 LN(x) 반환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `LN(2)` = LN(2) | fields `LN(2)`
fetched rows / total rows = 1/1
+--------------------+
| LN(2)              |
|--------------------|
| 0.6931471805599453 |
+--------------------+
```

LOG

사용량: x의 기본 e 로그인 x의 자연 로그를 LOG(x) 반환합니다. log(B, x)는 log(x)/log(B)와 동일합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
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

사용량: LOG2(x)는 log(x)/와 같습니다log(2).

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `LOG2(8)` = LOG2(8) | fields `LOG2(8)`
fetched rows / total rows = 1/1
+-----------+
| LOG2(8)   |
|-----------|
| 3.0       |
+-----------+
```

LOG10

사용량: LOG10(x)는 log(x)/와 같습니다log(10).

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `LOG10(100)` = LOG10(100) | fields `LOG10(100)`
fetched rows / total rows = 1/1
+--------------+
| LOG10(100)   |
|--------------|
| 2.0          |
+--------------+
```

MOD

사용량: 숫자 n의 나머지 부분을 m으로 나눈 값을 MOD(n, m) 계산합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: m이 0이 아닌 값인 경우 n과 m 유형 사이에서 더 넓은 유형입니다. m이 0인 경우는 NULL을 반환합니다.

예:

```
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

사용량: 상수 pi를 PI() 반환합니다.

반환 유형: DOUBLE

예:

```
os> source=people | eval `PI()` = PI() | fields `PI()`
fetched rows / total rows = 1/1
+-------------------+
| PI()              |
|-------------------|
| 3.141592653589793 |
+-------------------+
```

POW

사용량: x의 값을 y의 곱으로 POW(x, y) 계산합니다. 잘못된 입력은 NULL 결과를 반환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

동의어: POWER(_, _)

예:

```
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

사용량: x의 값을 y의 곱으로 POWER(x, y) 계산합니다. 잘못된 입력은 NULL 결과를 반환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

동의어: POW(_, _)

예:

```
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

사용량: x를 각도에서 라디안으로 RADIANS(x) 변환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `RADIANS(90)` = RADIANS(90) | fields `RADIANS(90)`
fetched rows / total rows  = 1/1
+--------------------+
| RADIANS(90)        |
|--------------------|
| 1.5707963267948966 |
+--------------------+
```

RAND

사용량: RAND()/RAND(N)는 0 <= 값 < 1.0 범위의 임의의 부동 소수점 값을 반환합니다. 정수 N을 지정하면 함수가 실행 전에 시드를 초기화합니다. 이 동작의 한 가지 의미는 동일한 인수 N을 사용하면가 매번 동일한 값을 rand(N) 반환하여 반복 가능한 열 값 시퀀스를 생성한다는 것입니다.

인수 유형: INTEGER

반환 유형: FLOAT

예:

```
os> source=people | eval `RAND(3)` = RAND(3) | fields `RAND(3)`
fetched rows / total rows = 1/1
+------------+
| RAND(3)    |
|------------|
| 0.73105735 |
+------------+
```

ROUND

사용량: 인수 x를 소수점 이하 자릿수로 ROUND(x, d) 반올림합니다. d를 지정하지 않으면 기본값은 0입니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형 맵:

- (정수/긴 [,정수]) -> 긴
- (FLOAT/DOUBLE [, 정수]) -> LONG

예:

```
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

사용량: 숫자가 음수인지, 0인지 또는 양수인지에 따라 인수의 기호를 -1, 0 또는 1로 SIGN 반환합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: INTEGER

예:

```
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

사용량: x의 사인을 `sin(x)` 계산합니다. 여기서 x는 라디안 단위로 제공됩니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형: DOUBLE

예:

```
os> source=people | eval `SIN(0)` = SIN(0) | fields `SIN(0)`
fetched rows / total rows = 1/1
+----------+
| SIN(0)   |
|----------|
| 0.0      |
+----------+
```

SQRT

사용량: 음수가 아닌 숫자의 제곱근을 SQRT 계산합니다.

인수 유형: INTEGER/LONG/FLOAT/DOUBLE

반환 유형 맵:

- (음수가 아님) INTEGER/LONG/FLOAT/DOUBLE -> DOUBLE

- (음수) INTEGER/LONG/FLOAT/DOUBLE -> NULL

예:

```
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 문자열 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

CONCAT

사용량:는 최대 9개의 문자열을 함께 CONCAT(str1, str2, ..., str_9) 추가합니다.

인수 유형:

- 문자열, 문자열, ..., 문자열
- 반환 유형: STRING

예:

```
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

사용량:는 지정된 구분 기호를 사용하여 둘 이상의 문자열을 CONCAT_WS(sep, str1, str2) 연결합니다.

인수 유형:

- 문자열, 문자열, ..., 문자열
- 반환 유형: STRING

예:

```
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

사용량: 바이트 단위로 측정된 입력 문자열의 길이를 length(str) 반환합니다.

인수 유형:

- STRING
- 반환 유형: INTEGER

예:

```
os> source=people | eval `LENGTH('helloworld')` = LENGTH('helloworld') | fields
 `LENGTH('helloworld')`
fetched rows / total rows = 1/1
+------------------------+
| LENGTH('helloworld')   |
|------------------------|
| 10                     |
```

```
+------------------------+
```

LOWER

사용량: 입력 문자열을 소문자로 `lower(string)` 변환합니다.

인수 유형:

- STRING
- 반환 유형: STRING

예:

```
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

사용량: 입력 문자열에서 선행 공백 문자를 `ltrim(str)` 제거합니다.

인수 유형:

- STRING
- 반환 유형: STRING

예:

```
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

사용량: 문자열에서 하위 문자열이 처음 발생한 위치를 POSITION(substr IN str) 반환합니다. 하위 문자열이 문자열에 없는 경우 0을 반환합니다. 인수가 NULL이면 NULL을 반환합니다.

인수 유형:

- 문자열, 문자열
- 반환 유형 INTEGER

예:

```
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

사용량: 입력 문자열의 반전된 문자열을 REVERSE(str) 반환합니다.

인수 유형:

- STRING
- 반환 유형: STRING

예:

```
os> source=people | eval `REVERSE('abcde')` = REVERSE('abcde') | fields
 `REVERSE('abcde')`
fetched rows / total rows = 1/1
+--------------------+
| REVERSE('abcde')   |
```

```
|--------------------|
| edcba              |
+--------------------+
```

RIGHT

사용량: 입력 문자열에서 가장 오른쪽 문자를 `right(str, len)` 반환합니다. 하위 문자열이 문자열에 없는 경우 0을 반환합니다. 인수가 NULL이면 NULL을 반환합니다.

인수 유형:

- 문자열, 정수
- 반환 유형: STRING

예:

```
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

사용량: 입력 문자열에서 후행 공백 문자를 `rtrim(str)`자릅니다.

인수 유형:

- STRING
- 반환 유형: STRING

예:

```
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

사용량: substring(str, start) 또는는 입력 문자열의 하위 문자열을 substring(str, start, length) 반환합니다. 길이를 지정하지 않으면 시작 위치에서 전체 문자열을 반환합니다.

인수 유형:

- 문자열, 정수, 정수
- 반환 유형: STRING

예:

```
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

사용량: 입력 문자열에서 선행 및 후행 공백을 trim(string) 제거합니다.

인수 유형:

- STRING
- 반환 유형: STRING

예:

```
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

사용량: 입력 문자열을 대문자로 `upper(string)` 변환합니다.

인수 유형:

- STRING
- 반환 유형: STRING

예:

```
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 유형 변환 함수

> **Note**
>
> 이 PPL 함수를 지원하는 AWS 데이터 소스 통합을 확인하려면 섹션을 참조하세요the section called “함수”.

TRIM

사용량:는를 `expr`로 `cast(expr as dateType)` 캐스팅`dataType`하고의 값을 반환합니다`dataType`.

다음과 같은 변환 규칙이 적용됩니다.

유형 변환 규칙

Src/Target	STRING	NUMBER	BOOLEAN	TIMESTAMP	날짜	TIME
STRING		Note1	Note1	TIMESTAMP()	날짜()	TIME()
NUMBER	Note1		v!=0	N/A	해당 사항 없음	N/A
BOOLEAN	Note1	v?1:0		N/A	해당 사항 없음	N/A
TIMESTAMP	Note1	N/A	N/A		날짜()	TIME()
날짜	Note1	N/A	해당 사항 없음	해당 사항 없음		N/A
TIME	Note1	N/A	해당 사항 없음	해당 사항 없음	N/A	

문자열로 캐스팅 예제:

```
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 |
+---------+--------+------------+
```

숫자로 캐스팅 예제:

```
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         |
+---------+-----------+
```

현재까지 캐스팅 예제:

```
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 |
+------------+----------+---------------------+
```

체인 캐스트 예제:

```
os> source=people | eval `cbool` = CAST(CAST(true as string) as boolean) | fields
 `cbool`
fetched rows / total rows = 1/1
+---------+
| cbool   |
|---------|
| True    |
+---------+
```

Amazon OpenSearch Service 도메인 모니터링

모니터링은 Amazon OpenSearch Service 및 기타 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다.는 OpenSearch Service 리소스를 모니터링하고, 문제를 보고하고, 필요한 경우 자동 조치를 취할 수 있는 다음 도구를 AWS 제공합니다.

Amazon CloudWatch

Amazon CloudWatch는 OpenSearch Service 리소스를 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지표가 특정 임계값에 도달하면 사용자에게 알리거나 조치를 하도록 경보를 설정할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.

Amazon CloudWatch Logs

Amazon CloudWatch Logs를 사용하면 OpenSearch 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다. CloudWatch Logs는 로그 파일의 정보를 모니터링하고 특정 임계값에 도달하면 사용자에게 알릴 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서를 참조하세요.

Amazon EventBridge

Amazon EventBridge는 OpenSearch Service 도메인의 변경 사항을 설명하는 시스템 이벤트의 스트림을 거의 실시간으로 제공합니다. 특정 이벤트를 감시하는 규칙을 생성하고 이러한 이벤트가 발생할 때 다른 AWS 서비스에서 자동화된 작업을 트리거할 수 있습니다. 자세한 내용은 Amazon EventBridge 사용 설명서를 참조하세요.

AWS CloudTrail

AWS CloudTrail 는 OpenSearch Service에 대한 구성 API 호출을 이벤트로 캡처합니다. 사용자가 지정한 Amazon S3 버킷에 이러한 이벤트를 전송할 수 있습니다. 이 정보를 사용하면 어떤 사용자 및 계정이 요청했는지, 어떤 소스 IP 주소에서 요청했는지, 언제 요청이 발생했는지 확인할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

주제

- Amazon CloudWatch로 OpenSearch 클러스터 지표 모니터링
- Amazon CloudWatch Logs를 사용하여 OpenSearch 로그 모니터링
- Amazon OpenSearch Service의 감사 로그 모니터링
- Amazon EventBridge를 사용하여 OpenSearch Service 이벤트 모니터링
- AWS CloudTrail을 사용한 Amazon OpenSearch Service API 호출 모니터링

Amazon CloudWatch로 OpenSearch 클러스터 지표 모니터링

Amazon OpenSearch Service는 도메인의 데이터를 Amazon CloudWatch에 게시합니다. CloudWatch를 사용하면 이러한 데이터 포인트에 대한 통계를 지표라는 정렬된 시계열 데이터 세트로 검색할 수 있습니다. OpenSearch Service는 60초 간격으로 CloudWatch에 대부분의 지표를 전송합니다. 범용 또는 마그네틱 EBS 볼륨을 사용하는 경우에는 EBS 볼륨 지표만 5분마다 업데이트됩니다. 모든 누적 지표(예: `ThreadpoolWriteRejected`, `ThreadpoolSearchRejected`)는 메모리 내에 있으며 상태가 손실됩니다. 노드 삭제, 노드 반송, 노드 교체 및 블루/그린 배포 중에 지표가 재설정됩니다. Amazon CloudWatch에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.

OpenSearch Service 콘솔에는 CloudWatch의 원시 데이터를 기반으로 하는 일련의 차트가 표시됩니다. 필요에 따라 콘솔의 그래프 대신에 CloudWatch에서 클러스터 데이터를 확인하는 것을 선호할 수 있습니다. 지표는 2주 동안 보관된 후 삭제됩니다. 메트릭은 추가 요금 없이 제공되지만 CloudWatch는 여전히 대시보드 및 경보 생성 시 요금이 청구됩니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하세요.

OpenSearch Service는 다음 지표를 CloudWatch에 게시합니다.

- the section called “클러스터 지표”
- the section called “전용 프라이머리 노드 지표입니다.”
- the section called “EBS 볼륨 지표입니다.”
- the section called “인스턴스 지표”
- the section called “UltraWarm 지표”
- the section called “전용 조정자 노드 지표”
- the section called “콜드 스토리지 지표”
- the section called “알림 지표”
- the section called “이상 탐지 지표”
- the section called “비동기 검색 지표”
- the section called “SQL 지표”
- the section called “k-NN 지표”
- the section called “클러스터 간 검색 지표”
- the section called “클러스터 간 복제 지표”
- the section called “순위 학습 지표”
- the section called “파이프 처리 언어 지표”

CloudWatch에서 지표 보기

CloudWatch 지표는 먼저 서비스 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다.

CloudWatch 콘솔을 사용하여 지표를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Metrics(지표)를 찾은 다음 All metrics(모든 지표)를 선택합니다. ES/OpenSearchService 네임스페이스를 선택합니다.
3. 해당 지표를 보려면 차원을 선택합니다. 개별 노드에 대한 지표는 `ClientId, DomainName, NodeId` 차원에 있습니다. 클러스터 지표는 `Per-Domain, Per-Client Metrics` 차원에 있습니다. 일부 노드 지표는 클러스터 수준에서 집계되므로 두 차원 모두에 포함됩니다. 샤드 지표는 `ClientId, DomainName, NodeId, ShardRole` 차원에 있습니다.

를 사용하여 지표 목록을 보려면 AWS CLI

다음 명령 실행:

```
aws cloudwatch list-metrics --namespace "AWS/ES"
```

OpenSearch Service의 상태 차트 해석

OpenSearch Service에서 지표를 보려면 Cluster health(클러스터 상태) 및 Instance health(인스턴스 상태) 탭을 선택합니다. 이 Instance health(인스턴스 상태) 탭은 박스 차트를 사용하여 각 OpenSearch 노드의 상태를 한눈에 파악할 수 있도록 합니다.

- 각 색 상자는 지정된 기간에 노드의 값 범위를 보여줍니다.
- 파란색 상자는 다른 노드와 일관적인 값을 나타냅니다. 빨간색 상자는 이상값을 나타냅니다
- 각 상자 내의 흰색 선은 노드의 현재 값을 보여줍니다.
- 각 상자의 양쪽에 있는 “수염”은 일정 기간에 모든 노드의 최솟값과 최댓값을 보여줍니다.

도메인의 구성을 변경하는 경우 Cluster health(클러스터 상태) 및 Instance health(인스턴스 상태) 탭의 개별 인스턴스 목록이 정확한 수로 반환되기 전에 짧은 기간에 두 배의 크기로 증가하곤 합니다. 이 동작에 대한 설명은 the section called “구성 변경” 섹션을 참조하세요.

클러스터 지표

Amazon OpenSearch Service는 다음 클러스터에 대한 지표를 제공합니다.

지표	설명
ClusterStatus.green	값이 1이면 클러스터의 노드에 모든 인덱스 샤드가 할당되었음을 나타냅니다. 관련 통계: Maximum
ClusterStatus.yellow	값이 1이면 모든 인덱스의 기본 샤드가 클러스터의 노드에 할당되어 있지만 하나 이상의 인덱스에 대해 복제본 샤드가 할당되어 있지 않음을 나타냅니다. 자세한 내용은 the section called “노란색 클러스터 상태” 단원을 참조하십시오. 관련 통계: Maximum
ClusterStatus.red	값이 1이면 인덱스 하나 이상의 기본 및 복제본 샤드가 클러스터의 노드에 할당되지 않았음을 나타냅니다. 자세한 내용은 the section called “빨간색 클러스터 상태” 섹션을 참조하세요. 관련 통계: Maximum
Shards.active	활성 기본 및 복제본 샤드의 총 수입니다. 관련 통계: Maximum, Sum
Shards.unassigned	클러스터의 노드에 할당되지 않은 샤드 수입니다. 관련 통계: Maximum, Sum
Shards.delayedUnassigned	제한 시간 설정으로 노드 할당이 지연된 샤드 수입니다. 관련 통계: Maximum, Sum
Shards.activePrimary	활성 기본 샤드 수입니다. 관련 통계: Maximum, Sum

지표	설명
Shards.initializing	초기화 중인 샤드 수입니다. 관련 통계: 합계
Shards.relocating	재배치 중인 샤드 수입니다. 관련 통계: 합계
Nodes	전용 프라이머리 노드 및 UltraWarm 노드를 포함하여 OpenSearch Service 클러스터에 있는 노드 수입니다. 자세한 내용은 the section called “구성 변경” 섹션을 참조하세요. 관련 통계: Maximum
SearchableDocuments	클러스터의 모든 데이터 노드에서 검색 가능한 총 문서 수입니다. 관련 통계: 최소, 최대, 평균
DeletedDocuments	클러스터의 모든 데이터 노드에서 삭제 표시된 총 문서 수입니다. 이들 문서는 더 이상 검색 결과에 나타나지 않지만, OpenSearch는 세그먼트 병합 시에만 삭제된 문서를 디스크에서 제거합니다. 이 지표는 삭제 요청 후 증가하고 세그먼트 병합 후 감소합니다. 관련 통계: 최소, 최대, 평균
CPUUtilization	클러스터의 데이터 노드에 대한 CPU 사용량 백분율입니다. 최대는 CPU 사용량이 가장 높은 노드를 나타냅니다. 평균은 클러스터의 모든 노드를 나타냅니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 관련 통계: Maximum, Average

지표	설명
FreeStorageSpace	클러스터에서 사용할 수 있는 데이터 노드 공간입니다. Sum은 클러스터의 사용 가능한 전체 공간을 표시하지만, 정확한 값을 얻으려면 이 기간을 1분으로 두어야 합니다. Minimum, Maximum은 사용 가능한 공간이 가장 작은 노드와 가장 큰 노드를 각각 표시합니다. 이 지표는 개별 노드에도 사용할 수 있습니다. OpenSearch Service는 이 지표가 0에 도달하는 경우 ClusterBlockException 를 발생시킵니다. 복구하려면 인덱스를 삭제하거나, 더 큰 인스턴스를 추가하거나 기존 인스턴스에 EBS 기반 스토리지를 추가해야 합니다. 자세한 내용은 the section called “사용 가능한 스토리지 공간 부족” 섹션을 참조하세요. OpenSearch Service 콘솔은 이 값을 GiB로 표시합니다. Amazon CloudWatch 콘솔은 이 값을 MiB로 표시합니다. **Note** FreeStorageSpace 는 항상 OpenSearch _cluster/stats 및 _cat/allocation API가 제공하는 값보다 낮습니다. OpenSearch Service는 내부 작업을 위해 각 인스턴스에 스토리지 공간의 일정 비율을 예약합니다. 자세한 내용은 스토리지 요구 사항 계산을 참조하세요. 관련 통계: Minimum, Maximum, Average, Sum
ClusterUsedSpace	클러스터의 총 사용 공간입니다. 정확한 값을 얻으려면 이 기간을 1분으로 두어야 합니다. OpenSearch Service 콘솔은 이 값을 GiB로 표시합니다. Amazon CloudWatch 콘솔은 이 값을 MiB로 표시합니다. 관련 통계: Minimum, Maximum

지표	설명
ClusterIndexWritesBlocked	수신되는 쓰기 요청에 대한 클러스터의 허용 또는 차단 여부를 나타냅니다. 값이 0이면 클러스터가 요청을 허용하고 있다는 것을 의미합니다. 값이 1이면 클러스터가 요청을 차단하고 있다는 것을 의미합니다. 몇 가지 공통적인 요인을 꼽자면 FreeStorageSpace 가 너무 낮은 경우 또는 JVMMemoryPressure 가 너무 높은 경우가 있습니다. 이러한 문제를 줄이려면 디스크 공간을 추가하거나 클러스터를 확장하는 것이 좋습니다. 관련 통계: Maximum
JVMMemoryPressure	클러스터의 모든 데이터 노드에 사용된 Java 힙의 최대 비율입니다. OpenSearch Service는 Java 힙에 인스턴스 RAM의 절반을 사용합니다(최대 힙 크기 32GiB). 인스턴스를 최대 64GiB의 RAM까지 수직 확장할 수 있으며 인스턴스를 추가하면 수평 확장도 가능합니다. the section called “권장되는 CloudWatch 경보” 섹션을 참조하세요. 관련 통계: Maximum **Note** 서비스 소프트웨어 R20220323에서 이 지표에 대한 로직이 변경되었습니다. 자세한 내용은 릴리스 정보를 참조하세요.
OldGenJVMMemoryPressure	클러스터의 모든 데이터 노드에서 '구세대'에 사용된 Java 힙의 최대 비율입니다. 이 지표는 노드 수준에도 사용할 수 있습니다. 관련 통계: Maximum
AutomatedSnapshotFailure	클러스터에 대해 실패한 자동 스냅샷 수입니다. 값 1은 지난 36시간 동안 도메인에 대해 생성된 자동 스냅샷이 없음을 나타냅니다. 관련 통계: Minimum, Maximum

지표	설명
CPUCreditBalance	클러스터의 데이터 노드에 사용할 수 있는 잔여 CPU 크레딧입니다. CPU 크레딧은 1분 동안 CPU 코어의 전체 성능을 제공합니다. 자세한 내용은 Amazon EC2 개발자 안내서의 CPU 크레딧을 참조하세요. 이 지표는 T2 인스턴스 유형에 대해서만 확인할 수 있습니다. 관련 통계: Minimum
OpenSearchDashboardsHealthyNodes	OpenSearch 대시보드의 상태 확인입니다. 최솟값, 최댓값 및 평균이 모두 1과 같으면 Dashboards가 정상적으로 동작하고 있습니다. 최대 1, 최소 0, 평균 0.7인 노드가 10개 있는 경우 이는 노드 7개(70%)가 정상이고 노드 3개(30%)가 비정상임을 의미합니다. 관련 통계: 최소, 최대, 평균
OpensearchDashboardsReportingFailedRequestSysErrCount	서버 문제 또는 기능 제한으로 인해 실패한 OpenSearch 대시보드 보고서 생성에 대한 요청 수입니다. 관련 통계: 합계
OpensearchDashboardsReportingFailedRequestUserErrCount	클라이언트 문제로 인해 실패한 OpenSearch 대시보드 보고서 생성에 대한 요청 수입니다. 관련 통계: 합계
OpensearchDashboardsReportingRequestCount	OpenSearch 대시보드 보고서 생성에 대한 총 요청 수입니다. 관련 통계: 합계
OpensearchDashboardsReportingSuccessCount	OpenSearch 대시보드 보고서 생성에 대해 성공한 요청 수입니다. 관련 통계: 합계

지표	설명
KMSKeyError	값이 1이면 저장 데이터를 암호화하는 데 사용된 AWS KMS 키가 비활성화되었음을 나타냅니다. 도메인을 정상 작동으로 복원하려면 키를 다시 활성화해야 합니다. 콘솔에는 저장된 데이터를 암호화하는 도메인에 대해서만 이 지표가 표시됩니다. 관련 통계: Minimum, Maximum
KMSKeyInaccessible	값이 1이면 저장 데이터를 암호화하는 데 사용된 AWS KMS 키가 삭제되었거나 OpenSearch Service에 대한 권한 부여가 취소되었음을 나타냅니다. 이 상태의 도메인은 복원할 수 없습니다. 하지만 수동 스냅샷이 있는 경우 해당 스냅샷을 사용하여 도메인의 데이터를 새 도메인으로 마이그레이션할 수 있습니다. 콘솔에는 저장된 데이터를 암호화하는 도메인에 대해서만 이 지표가 표시됩니다. 관련 통계: Minimum, Maximum
InvalidHostHeaderRequests	잘못된(또는 누락된) 호스트 헤더를 포함하여 OpenSearch 클러스터에 수행된 HTTP 요청 수입니다. 유효한 요청에는 도메인 호스트 이름이 호스트 헤더 값으로 포함됩니다. OpenSearch Service는 제한적인 액세스 정책이 없는 퍼블릭 액세스 도메인에 대한 잘못된 요청을 거부합니다. 모든 도메인에 제한적인 액세스 정책을 적용하는 것을 권장합니다. 이 지표에 대한 값이 클 경우, 사용자의 OpenSearch 클라이언트가 요청에 도메인 호스트 이름이(예를 들어, IP 주소 아님) 포함되었는지 확인합니다. 관련 통계: 합계
OpenSearchRequests (previously ElasticsearchRequests)	OpenSearch 클러스터에 수행된 요청 수입니다. 관련 통계: 합계

지표	설명
2xx, 3xx, 4xx, 5xx	해당 HTTP 응답 코드(2xx, 3xx, 4xx, 5xx)를 발생시킨 도메인에 대한 요청 건수입니다. 관련 통계: 합계
ThroughputThrottle	디스크가 제한되었는지 여부를 나타냅니다. 제한은 ReadThroughputMicroBursting 및 WriteThroughputMicroBursting 의 총 처리량이 최대 처리량 MaxProvisionedThroughput 보다 높을 때 발생합니다. MaxProvisionedThroughput 는 인스턴스 처리량 또는 프로비저닝된 볼륨 처리량 중 더 낮은 값입니다. 값이 1이면 디스크가 제한되었음을 나타냅니다. 값이 0이면 정상적인 동작 상태를 나타냅니다. 인스턴스 처리량에 대한 자세한 내용은 Amazon EBS 최적화 인스턴스를 참조하세요. 볼륨 처리량에 대한 자세한 내용은 Amazon EBS 볼륨 유형을 참조하세요. 관련 통계: Minimum, Maximum
IopsThrottle	도메인에서 초당 입출력 작업량(IOPS)이 스로틀링되었는지 여부를 나타냅니다. 스로틀링은 데이터 노드의 IOPS가 EBS 볼륨의 최대 허용 한도 또는 데이터 노드의 EC2 인스턴스를 위반할 때 발생합니다. 인스턴스 IOPS에 대한 자세한 내용은 Amazon EBS 최적화 인스턴스를 참조하세요. 볼륨 IOPS에 대한 자세한 내용은 Amazon EBS 볼륨 유형을 참조하세요. 관련 통계: Minimum, Maximum
HighSwapUsage	값이 1이면 페이지 오류로 인한 스왑으로 인해 특정 기간 기본 디스크 사용량이 급증할 수 있음을 나타냅니다. 관련 통계: Maximum

전용 프라이머리 노드 지표입니다.

Amazon OpenSearch Service는 전용 프라이머리 노드에 대한 다음 지표를 제공합니다.

지표	설명
MasterCPUUtilization	전용 프라이머리 노드에서 사용하는 최대 CPU 리소스 비율. 이 지표가 60%에 도달하면 인스턴스 유형의 크기를 늘리는 것이 좋습니다. 관련 통계: Maximum
MasterFreeStorageSpace	이 지표는 관련이 없으므로 무시해도 좋습니다. 이 서비스에서는 프라이머리 노드를 데이터 노드로 사용하지 않습니다.
MasterJVMMemoryPressure	클러스터의 모든 전용 프라이머리 노드에 사용되는 Java 힙의 최대 비율. 이 지표가 85%에 도달하면 더 큰 인스턴스 유형으로 이전하는 것이 좋습니다. 관련 통계: Maximum Note 서비스 소프트웨어 R20220323에서 이 지표에 대한 로직이 변경되었습니다. 자세한 내용은 릴리스 정보를 참조하세요.
MasterOldGenJVMMemoryPressure	프라이머리 노드당 '구세대'에 사용된 Java 힙의 최대 비율입니다. 관련 통계: Maximum
MasterCPUCreditBalance	클러스터의 전용 프라이머리 노드에 사용할 수 있는 잔여 CPU 크레딧입니다. CPU 크레딧은 1분 동안 CPU 코어의 전체 성능을 제공합니다. 자세한 내용은 Amazon EC2 개발자 안내서의 CPU 크레딧을 참조하세요. 이 지표는 T2 인스턴스 유형에 대해서만 확인할 수 있습니다. 관련 통계: Minimum

지표	설명
MasterReachableFromNode	MasterNotDiscovered 예외에 대한 상태 확인입니다. 값이 1이면 정상적인 동작 상태를 나타냅니다. 값이 0이면 /_cluster/health/가 오류를 일으킨 것을 나타냅니다. 여기에서 오류란 소스 노드에서 프라이머리 노드에 도달할 수 없다는 것을 의미합니다. 일반적으로 네트워크 연결 문제 또는 AWS 종속성 문제의 결과입니다. 관련 통계: Maximum
MasterSysMemoryUtilization	사용 중인 프라이머리 노드 메모리의 비율입니다. 관련 통계: Maximum

전용 조정자 노드 지표

Amazon OpenSearch Service는 전용 조정자 노드에 대한 다음 지표를 제공합니다.

지표	설명
CoordinatorCPUUtilization	전용 조정자 노드에서 사용하는 최대 CPU 리소스 비율. 이 지표가 80%에 도달하면 인스턴스 유형의 크기를 늘리는 것이 좋습니다. 관련 통계: Maximum
CoordinatorJVMMemoryPressure	클러스터의 모든 전용 조정자 노드에 사용되는 Java 힙의 최대 비율. 이 지표가 85%에 도달하면 더 큰 인스턴스 유형으로 이전하는 것이 좋습니다. 관련 통계: Maximum
CoordinatorOldGenJVMMemoryPressure	프라이머리 노드당 '구세대'에 사용된 Java 힙의 최대 비율입니다. 관련 통계: Maximum
CoordinatorSysMemoryUtilization	사용 중인 조정자 노드 메모리의 비율.

지표	설명
	관련 통계: Maximum
CoordinatorFreeStorageSpace	이 지표는 서비스가 조정자 노드를 데이터 노드로 사용하지 않음을 나타냅니다.

EBS 볼륨 지표입니다.

Amazon OpenSearch Service는 다음 EBS 볼륨에 대한 지표를 제공합니다.

지표	설명
ReadLatency	EBS 볼륨에 대한 읽기 작업의 대기 시간(초)입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 관련 통계: 최소, 최대, 평균
WriteLatency	EBS 볼륨에 대한 쓰기 작업의 대기 시간(초)입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 관련 통계: 최소, 최대, 평균
ReadThroughput	EBS 볼륨에 대한 읽기 작업의 처리량(바이트/초)입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 관련 통계: 최소, 최대, 평균
ReadThroughputMicroBursting	마이크로 버스팅을 고려할 때 EBS 볼륨의 읽기 작업 처리량(초당 바이트)입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 마이크로 버스팅은 EBS 볼륨이 상당히 짧은 시간(1분 미만) 동안 높은 IOPS 또는 처리량을 버스팅할 때 발생합니다. 관련 통계: 최소, 최대, 평균
WriteThroughput	EBS 볼륨에 대한 쓰기 작업의 처리량(바이트/초)입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 관련 통계: 최소, 최대, 평균

지표	설명
WriteThroughputMicroBursting	마이크로 버스팅을 고려할 때 EBS 볼륨의 쓰기 작업 처리량(초당 바이트)입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 마이크로 버스팅은 EBS 볼륨이 상당히 짧은 시간(1분 미만) 동안 높은 IOPS 또는 처리량을 버스팅할 때 발생합니다. 관련 통계: 최소, 최대, 평균
DiskQueueDepth	EBS 볼륨에 대해 대기 중인 I/O 요청 수입니다. 관련 통계: 최소, 최대, 평균
ReadIOPS	EBS 볼륨에 대한 읽기 작업의 초당 I/O 작업 수입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 관련 통계: 최소, 최대, 평균
ReadIOPSMicroBursting	마이크로 버스팅을 고려할 때 EBS 볼륨에 대한 읽기 작업의 초당 I/O 작업 수입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 마이크로 버스팅은 EBS 볼륨이 상당히 짧은 시간(1분 미만) 동안 높은 IOPS 또는 처리량을 버스팅할 때 발생합니다. 관련 통계: 최소, 최대, 평균
WriteIOPS	EBS 볼륨에 대한 쓰기 작업의 초당 I/O 작업 수입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 관련 통계: 최소, 최대, 평균
WriteIOPSMicroBursting	마이크로 버스팅을 고려할 때 EBS 볼륨에 대한 쓰기 작업의 초당 I/O 작업 수입니다. 이 지표는 개별 노드에도 사용할 수 있습니다. 마이크로 버스팅은 EBS 볼륨이 상당히 짧은 시간(1분 미만) 동안 높은 IOPS 또는 처리량을 버스팅할 때 발생합니다. 관련 통계: 최소, 최대, 평균

지표	설명
BurstBalance	EBS 볼륨에 대해 버스트 버킷에 남아 있는 입력 및 출력(I/O) 크레딧의 비율입니다. 값이 100이면 볼륨에 최대 크레딧 수가 누적되었음을 의미합니다. 이 비율이 70% 미만으로 떨어지면 the section called “낮은 EBS 버스트 밸런스” 섹션을 참조하세요. gp3 볼륨 유형이 있는 도메인과 볼륨 크기가 1000GiB를 초과하는 gp2 볼륨이 있는 도메인의 경우 버스트 균형은 0으로 유지됩니다. 관련 통계: 최소, 최대, 평균
VolumeStalledIOcheck	손상된 시점을 결정하는 EBS 볼륨의 상태입니다. 지표는 EBS 볼륨이 입력 및 출력 작업을 완료할 수 있는지 여부에 따라 0(통과) 또는 1(실패) 상태를 반환하는 이진 값입니다. VolumeStalledIOcheck 는 개별 노드에도 사용할 수 있습니다. 관련 통계: 최소, 최대, 평균

인스턴스 지표

Amazon OpenSearch Service는 도메인의 각 인스턴스에 대해 다음 지표를 제공합니다. OpenSearch Service는 이러한 인스턴스 지표를 집계하여 전체 클러스터 상태에 대한 이해를 돕습니다. 콘솔에서 Sample Count(샘플 수) 통계를 이용하여 이 동작을 확인할 수 있습니다. 다음 표의 각 지표는 노드 및 클러스터 관련 통계를 포함합니다.

> **Important**
>
> 다양한 버전의 Elasticsearch는 서로 다른 스레드 풀을 사용하여 _index API에 대한 호출을 처리합니다. Elasticsearch 1.5 및 2.3은 인덱스 스레드 풀을 사용합니다. Elasticsearch 5.x, 6.0, 6.2는 벌크 스레드 풀을 사용합니다. OpenSearch 및 Elasticsearch 6.3 이상은 쓰기 스레드 풀을 사용합니다. 현재 OpenSearch Service 콘솔에는 벌크 스레드 풀에 대한 그래프가 포함되어 있지 않습니다.
> GET _cluster/settings?include_defaults=true를 사용하여 클러스터의 스레드 풀과 대기열 크기를 확인합니다.

지표	설명
FetchLatency	노드의 모든 샤드 가져오기 작업에서 분당 N과 분(N - 1) 사이의 총 시간 차이를 밀리초 단위로 표시합니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum
FetchRate	데이터 노드의 모든 샤드에 대한 분당 총 샤드 가져오기 작업 수입니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum, Sum
ScrollTotal	데이터 노드의 모든 샤드에 대한 분당 총 샤드 스크롤 작업 수입니다. 관련 노드 통계: Average, Maximum 관련 클러스터 통계: Average, Maximum, Sum
ScrollCurrent	현재 실행 중인 샤드 스크롤 작업 수입니다. 관련 노드 통계: Average, Maximum 관련 클러스터 통계: Average, Maximum, Sum
OpenContexts	열린 검색 컨텍스트의 수입니다. 관련 노드 통계: Average, Maximum 관련 클러스터 통계: Average, Maximum, Sum
ThreadCount	OpenSearch 프로세스에서 현재 사용 중인 총 스레드 수입니다. 관련 노드 통계: Average, Maximum 관련 클러스터 통계: Average, Maximum, Sum

지표	설명
ShardReactivateCount	모든 샤드가 유휴 상태에서 활성화된 총 횟수입니다. 관련 노드 통계: Sum, Maximum 관련 클러스터 통계: Sum, Maximum
ConcurrentSearchRate	한 데이터 노드의 모든 샤드에 대한 분당 동시 세그먼트 검색을 사용한 총 검색 요청 수. _search API에 대한 단일 호출은 많은 샤드로부터 결과를 반환할 수 있습니다. 이러한 샤드 중 5개가 한 노드에 있는 경우, 클라이언트가 단 한 개만 요청했더라도 노드는 이 지표에 대해 5를 보고할 것입니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum, Sum
ConcurrentSearchLatency	한 노드에서 동시 세그먼트 검색을 사용한 모든 검색에 소요된 N분과 (N-1)분 사이의 총 시간 차이(밀리초). 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum
IndexingLatency	한 노드의 모든 인덱싱 작업에 소요된 총 시간 차이(밀리초)로, 이 차이는 분 N에서 (N-1)분입니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum

지표	설명
IndexingRate	분당 인덱싱 작업 수입니다. 2개의 문서를 추가하고 2개를 4개 작업으로 업데이트하는 _bulk API에 대한 하나의 호출입니다. 이것은 하나 이상의 노드에 분산될 수 있습니다. 인덱스에 하나 이상의 복제본이 있고 최적화된 인스턴스 없이 OpenSearch 도메인에 있는 경우 클러스터의 다른 노드 역시 총 4개의 인덱싱 작업을 기록합니다. 최적화된 인스턴스를 포함하는 OpenSearch 도메인의 경우 복제본이 있는 다른 노드는 작업을 기록하지 않습니다. 문서 삭제는 이 지표에 포함되지 않습니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum, Sum
SearchLatency	한 노드의 모든 검색 작업에 소요된 총 시간 차이(밀리초)로, 이 차이는 분 N에서 (N-1)분입니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum
SearchRate	한 데이터 노드의 모든 샤드에 대한 분당 검색 요청의 총 수입니다. _search API에 대한 단일 호출은 많은 샤드로부터 결과를 반환할 수 있습니다. 이러한 샤드 중 5개가 한 노드에 있는 경우, 클라이언트가 단 한 개만 요청했더라도 노드는 이 지표에 대해 5를 보고할 것입니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum, Sum
SegmentCount	데이터 노드의 세그먼트 수입니다. 세그먼트가 많을수록 각 검색 시간이 길어집니다. OpenSearch는 때때로 작은 세그먼트를 더 큰 세그먼트로 병합합니다. 관련 노드 통계: Maximum, Average 관련 클러스터 통계: Sum, Maximum, Average

지표	설명
SysMemoryUtilization	사용 중인 인스턴스 메모리의 비율(%)입니다. 이 지표의 값이 큰 것은 정상이며 일반적으로 클러스터에 문제가 있음을 나타내지 않습니다. 잠재적인 성능 및 안정성 문제에 대한 더 나은 지표는 JVMMemoryPressure 지표를 참조하세요. 관련 노드 통계: Minimum, Maximum, Average 관련 클러스터 통계: Minimum, Maximum, Average
JVMGCYoungCollectionCount	"신세대" 가비지 수집이 실행된 횟수입니다. 클러스터 작업은 일반적으로 실행 수가 계속 증가하여 커집니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
JVMGCYoungCollectionTime	클러스터가 "신세대" 가비지 수집을 수행하는 데 소비 한 시간(밀리초)입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
JVMGCOldCollectionCount	"구세대" 가비지 수집이 실행된 횟수입니다. 리소스가 충분한 클러스터에서는 이 수가 적게 유지되고 자주 증가하지 않습니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
JVMGCOldCollectionTime	클러스터가 "구세대" 가비지 수집을 수행하는 데 소비 한 시간 (밀리초)입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average

지표	설명
OpenSearchDashboardsConcurrentConnections	OpenSearch 대시보드에 대한 활성 동시 연결 수입니다. 이 수가 계속 증가하면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
OpenSearchDashboardsHealthyNode	개별 OpenSearch 대시보드 노드에 대한 상태 확인입니다. 값이 1이면 정상적인 동작 상태를 나타냅니다. 값이 0이면 Dashboards에 액세스할 수 없다는 것을 나타냅니다. 관련 노드 통계: Minimum 관련 클러스터 통계: Minimum, Maximum, Average
OpenSearchDashboardsHeapTotal	OpenSearch 대시보드에 할당된 힙 메모리 양(MiB)입니다. 다른 EC2 인스턴스 유형은 정확한 메모리 할당에 영향을 줄 수 있습니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
OpenSearchDashboardsHeapUsed	OpenSearch 대시보드에서 사용하는 힙 메모리의 절대 양(MiB)입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
OpenSearchDashboardsHeapUtilization	OpenSearch 대시보드에서 사용하는 사용 가능한 힙 메모리의 최대 백분율입니다. 이 값이 80% 이상으로 증가하면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Minimum, Maximum, Average

지표	설명
OpenSearchDashboardsOS1MinuteLoad	OpenSearch 대시보드에 대한 1분 CPU 로드 평균입니다. CPU 로드는 이상적으로 1.00 미만으로 유지되어야 합니다. 일시적인 급증은 정상이지만 이 지표가 지속해서 1.00을 초과할 경우 인스턴스 유형의 크기를 늘리는 것이 좋습니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum
OpenSearchDashboardsRequestTotal	OpenSearch 대시보드에 대한 총 HTTP 요청 수입니다. 시스템 속도가 느리거나 Dashboards 요청 수가 많으면 인스턴스 유형의 크기를 늘리는 것을 고려합니다. 관련 노드 통계: Sum 관련 클러스터 통계: Sum
OpenSearchDashboardsResponseTimesMaxInMillis	OpenSearch 대시보드가 요청에 응답하는 데 걸리는 최대 시간(밀리초)입니다. 요청 결과가 반환되는 데 시간이 지속해서 오래 걸리는 경우 인스턴스 유형의 크기를 늘리는 것을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Maximum, Average
SearchTaskCancelled	코디네이터 노드 취소 횟수. 관련 노드 통계: Sum 관련 클러스터 통계: Sum
SearchShardTaskCancelled	데이터 노드 취소 횟수. 관련 노드 통계: Sum 관련 클러스터 통계: Sum,

지표	설명
ThreadpoolForce_mergeQueue	강제 병합 스레드 풀에서 대기 중인 작업의 수입니다. 대기열 크기가 지속해서 높으면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
ThreadpoolForce_mergeRejected	강제 병합 스레드 풀에서 거부된 작업의 수입니다. 이 수가 계속 증가하면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum
ThreadpoolForce_mergeThreads	강제 병합 스레드 풀의 크기입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum
ThreadpoolIndexQueue	인덱스 스레드 풀에서 대기 중인 작업의 수입니다. 대기열 크기가 지속해서 높으면 클러스터 확장을 고려합니다. 인덱스 대기열의 최대 크기는 200입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
ThreadpoolIndexRejected	인덱스 스레드 풀에서 거부된 작업의 수입니다. 이 수가 계속 증가하면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum
ThreadpoolIndexThreads	인덱스 스레드 풀의 크기입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum

지표	설명
ThreadpoolSearchQueue	검색 스레드 풀에서 대기 중인 작업의 수입니다. 대기열 크기가 지속해서 높으면 클러스터 확장을 고려합니다. 검색 대기열의 최대 크기는 1,000입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
ThreadpoolSearchRejected	검색 스레드 풀에서 거부된 작업의 수입니다. 이 수가 계속 증가하면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum
ThreadpoolSearchThreads	검색 스레드 풀의 크기입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum
Threadpoolsql-workerQueue	SQL 검색 스레드 풀에서 대기 중인 작업의 수입니다. 대기열 크기가 지속해서 높으면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
Threadpoolsql-workerRejected	SQL 검색 스레드 풀에서 거부된 작업의 수입니다. 이 수가 계속 증가하면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum
Threadpoolsql-workerThreads	SQL 검색 스레드 풀의 크기입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum

지표	설명
ThreadpoolBulkQueue	벌크 스레드 풀에서 대기 중인 작업의 수입니다. 대기열 크기가 지속해서 높으면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
ThreadpoolBulkRejected	벌크 스레드 풀에서 거부된 작업의 수입니다. 이 수가 계속 증가하면 클러스터 확장을 고려합니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum
ThreadpoolBulkThreads	벌크 스레드 풀의 크기입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum
ThreadpoolIndexSearcherQueue	인덱스 검색기 스레드 풀에서 대기 중인 작업의 수. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
ThreadpoolIndexSearcherRejected	인덱스 검색기 스레드 풀에서 거부된 작업의 수. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum
ThreadpoolIndexSearcherThreads	인덱스 검색기 스레드 풀의 크기. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum

지표	설명
ThreadpoolWriteThreads	쓰기 스레드 풀의 크기입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum
ThreadpoolWriteQueue	쓰기 스레드 풀에서 대기 중인 작업의 수입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum
ThreadpoolWriteRejected	쓰기 스레드 풀에서 거부된 작업의 수입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum **Note** 버전 7.1에서는 기본 쓰기 대기열 크기가 200에서 10000으로 증가했기 때문에 이 지표는 더 이상 OpenSearch Service에서 거부하는 유일한 지표가 아닙니다. CoordinatingWriteRejected , PrimaryWriteRejected , ReplicaWriteRejected 지표를 사용하여 7.1 및 이후 버전에서 거부를 모니터링합니다.
CoordinatingWriteRejected	마지막 OpenSearch Service 프로세스 시작 이후 인덱싱 압력으로 인해 조정 노드에서 발생한 총 거부 횟수입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum 이 지표는 버전 7.1 및 이후 버전에서 사용할 수 있습니다.

지표	설명
PrimaryWriteRejected	마지막 OpenSearch Service 프로세스 시작 이후 인덱싱 압력으로 인해 기본 샤드에서 발생한 총 거부 횟수입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum 이 지표는 버전 7.1 및 이후 버전에서 사용할 수 있습니다.
ReplicaWriteRejected	마지막 OpenSearch Service 프로세스 시작 이후 인덱싱 압력으로 인해 복제본 샤드에서 발생한 총 거부 횟수입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum 이 지표는 버전 7.1 및 이후 버전에서 사용할 수 있습니다.
WorkloadManagementEnabled	워크로드 관리 기능이 활성화되었는지 여부를 나타냅니다. 값이 1이면 활성화됨, 값이 0이면 비활성화monitor_only 됨입니다. 관련 노드 통계: Maximum, Minimum 관련 클러스터 통계: Average, Sum 이 지표는 버전 7.1 및 이후 버전에서 사용할 수 있습니다.
SoftQueryGroupCount	도메인의 소프트 모드에 있는 쿼리 그룹 수입니다. 관련 노드 통계: Average, Maximum 관련 클러스터 통계: Average, Maximum, Sum 이 지표는 버전 7.1 및 이후 버전에서 사용할 수 있습니다.

지표	설명
EnforcedQueryGroupCount	도메인에서 적용 모드에 있는 쿼리 그룹 수입니다. 관련 노드 통계: Average, Maximum 관련 클러스터 통계: Average, Maximum, Sum 이 지표는 버전 7.1 및 이후 버전에서 사용할 수 있습니다.

UltraWarm 지표

Amazon OpenSearch Service는 UltraWarm 노드에 대한 다음 지표를 제공합니다.

지표	설명
WarmCPUUtilization	클러스터의 UltraWarm 노드에 대한 CPU 사용량 백분율입니다. 최대는 CPU 사용량이 가장 높은 노드를 나타냅니다. 평균은 클러스터의 모든 UltraWarm 노드를 나타냅니다. 이 지표는 개별 UltraWarm 노드에도 사용할 수 있습니다. 관련 통계: Maximum, Average
WarmFreeStorageSpace	사용 가능한 웜 스토리지 공간(MiB)입니다. UltraWarm은 연결된 디스크 대신 Amazon S3를 사용하기 때문에 Sum이 유일한 관련 통계입니다. 정확한 값을 얻으려면 이 기간을 1분으로 두어야 합니다. 관련 통계: 합계
WarmSearchableDocuments	클러스터의 모든 웜 인덱스에서 검색 가능한 총 문서 수입니다. 정확한 값을 얻으려면 이 기간을 1분으로 두어야 합니다. 관련 통계: 합계
WarmSearchLatency	UltraWarm에서 모든 검색에 소요된 N분과 (N-1)분 사이의 총 시간 차이(밀리초)입니다. 관련 노드 통계: Average

지표	설명
	관련 클러스터 통계: Average, Maximum
WarmSearchRate	한 UltraWarm 노드의 모든 샤드에 대한 분당 검색 요청의 총 수입니다. _search API에 대한 단일 호출은 많은 샤드로부터 결과를 반환할 수 있습니다. 이러한 샤드 중 5개가 한 노드에 있는 경우, 클라이언트가 단 한 개만 요청했더라도 노드는 이 지표에 대해 5를 보고할 것입니다. 관련 노드 통계: Average 관련 클러스터 통계: Average, Maximum, Sum
WarmStorageSpaceUtilization	클러스터가 사용 중인 총 웜 스토리지 공간 크기(MiB)입니다. 관련 통계: Maximum
HotStorageSpaceUtilization	클러스터를 사용 중인 총 핫 스토리지 공간 크기입니다. 관련 통계: Maximum
WarmSysMemoryUtilization	사용 중인 웜 노드 메모리의 비율입니다. 관련 통계: Maximum
HotToWarmMigrationQueueSize	현재 핫 스토리지에서 웜 스토리지로의 마이그레이션을 기다리는 인덱스 수입니다. 관련 통계: Maximum
WarmToHotMigrationQueueSize	현재 웜 스토리지에서 핫 스토리지로의 마이그레이션을 기다리는 인덱스 수입니다. 관련 통계: Maximum
HotToWarmMigrationFailureCount	실패한 핫-웜 마이그레이션의 총 수입니다. 관련 통계: 합계

지표	설명
HotToWarmMigrationForceMergeLatency	마이그레이션 프로세스의 강제 병합 단계의 평균 대기 시간입니다. 이 단계가 일관되게 너무 오래 걸리면 index.ultrawarm.migration.force_merge.max_num_segments 를 늘리는 것을 고려합니다. 관련 통계: Average
HotToWarmMigrationSnapshotLatency	마이그레이션 프로세스 중 스냅샷 단계의 평균 대기 시간입니다. 이 단계가 일관되게 너무 오래 걸리면 샤드의 크기가 적절하게 조정되고 클러스터 전체에 분산되어 있는지 확인합니다. 관련 통계: Average
HotToWarmMigrationProcessingLatency	성공한 핫-웜 마이그레이션의 평균 대기 시간으로, 대기열에서 소요된 시간을 포함하지 않습니다. 이 값은 마이그레이션 프로세스의 강제 병합, 스냅샷 및 샤드 재배치 단계를 완료하는 데 걸리는 시간의 합계입니다. 관련 통계: Average
HotToWarmMigrationSuccessCount	성공한 핫-웜 마이그레이션의 총 수입니다. 관련 통계: 합계
HotToWarmMigrationSuccessLatency	성공한 핫-웜 마이그레이션의 평균 대기 시간으로, 대기열에서 소요된 시간을 포함합니다. 관련 통계: Average
WarmThreadpoolSearchThreads	UltraWarm 검색 스레드 풀의 크기입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Average, Sum
WarmThreadpoolSearchRejected	UltraWarm 검색 스레드 풀에서 거부된 작업의 수입니다. 이 수가 계속 증가하면 UltraWarm 노드를 추가하는 것이 좋습니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum

지표	설명
WarmThreadpoolSearchQueue	UltraWarm 검색 스레드 풀에서 대기 중인 작업의 수입니다. 대기열 크기가 지속해서 높으면 UltraWarm 노드를 추가하는 것이 좋습니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
WarmJVMMemoryPressure	UltraWarm 노드에 사용되는 Java 힙의 최대 비율입니다. 관련 통계: Maximum **Note** 서비스 소프트웨어 R20220323에서 이 지표에 대한 로직이 변경되었습니다. 자세한 내용은 릴리스 정보를 참조하세요.
WarmOldGenJVMMemoryPressure	UltraWarm 노드당 '구세대'에 사용된 Java 힙의 최대 비율입니다. 관련 통계: Maximum
WarmJVMGCYoungCollectionCount	UltraWarm 노드에서 "신세대" 가비지 수집이 실행된 횟수입니다. 클러스터 작업은 일반적으로 실행 수가 계속 증가하여 커집니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
WarmJVMGCYoungCollectionTime	클러스터가 UltraWarm 노드에서 "신세대"가비지 수집을 수행하는 데 소비한 시간(밀리초)입니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average

지표	설명
WarmJVMGCOldCollectionCount	UltraWarm 노드에서 "구세대" 가비지 수집이 실행된 횟수입니다. 리소스가 충분한 클러스터에서는 이 수가 적게 유지되고 자주 증가하지 않습니다. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
WarmConcurrentSearchRate	한 UltraWarm 노드의 모든 샤드에 대한 분당 동시 세그먼트 검색 요청을 사용한 총 검색 요청 수. _search API에 대한 단일 호출은 많은 샤드로부터 결과를 반환할 수 있습니다. 이러한 샤드 중 5개가 한 노드에 있는 경우, 클라이언트가 단 한 개만 요청했더라도 노드는 이 지표에 대해 5를 보고할 것입니다. 관련 노드 통계: Average 관련 클러스터 통계: Sum, Maximum, Average
WarmConcurrentSearchLatency	UltraWarm 노드에서 동시 세그먼트 검색을 사용한 모든 검색에 소요된 N분과 (N-1)분 사이의 총 시간 차이(밀리초). 관련 노드 통계: Average 관련 클러스터 통계: Maximum, Average
WarmThreadpoolIndexSearcherQueue	UltraWarm 인덱스 검색기 스레드 풀에서 대기 중인 작업의 수. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum, Maximum, Average
WarmThreadpoolIndexSearcherRejected	UltraWarm 인덱스 검색기 스레드 풀에서 거부된 작업의 수. 관련 노드 통계: Maximum 관련 클러스터 통계: Sum

지표	설명
WarmThreadpoolIndexSearcherThreads	UltraWarm 인덱스 검색기 스레드 풀의 크기. 관련 노드 통계: Maximum 관련 클러스터 통계: 합계, 평균

콜드 스토리지 지표

Amazon OpenSearch Service는 콜드 스토리지에 대한 다음 지표를 제공합니다.

지표	설명
ColdStorageSpaceUtilization	클러스터를 사용 중인 총 콜드 스토리지 공간 크기(MiB)입니다. 관련 통계: 최대
ColdToWarmMigrationFailureCount	실패한 콜드-웜 마이그레이션의 총 수입니다. 관련 통계: 합계
ColdToWarmMigrationLatency	콜드-웜 마이그레이션을 성공적으로 완료하는 데 걸리는 시간입니다. 관련 통계: Average
ColdToWarmMigrationQueueSize	현재 콜드 스토리지에서 웜 스토리지로의 마이그레이션을 기다리는 인덱스 수입니다. 관련 통계: Maximum
ColdToWarmMigrationSuccessCount	성공한 콜드-웜 마이그레이션의 총 수입니다. 관련 통계: 합계
WarmToColdMigrationFailureCount	실패한 웜-콜드 마이그레이션의 총 수입니다. 관련 통계: 합계

지표	설명
WarmToColdMigrationLatency	웜-콜드 마이그레이션을 성공적으로 완료하는 데 걸리는 시간입니다. 관련 통계: Average
WarmToColdMigrationQueueSize	현재 웜 스토리지에서 콜드 스토리지로의 마이그레이션을 기다리는 인덱스 수입니다. 관련 통계: Maximum
WarmToColdMigrationSuccessCount	성공한 웜-콜드 마이그레이션의 총 수입니다. 관련 통계: 합계

OR1 지표

Amazon OpenSearch Service는 OR1 인스턴스에 대한 다음 지표를 제공합니다.

지표	설명
RemoteStorageUsedSpace	클러스터가 사용 중인 총 Amazon S3 공간 크기(MiB)입니다. 관련 통계: 합계
RemoteStorageWriteRejected	원격 스토리지 및 복제 압력으로 인해 기본 샤드에서 거부된 총 요청 수입니다. 이 값은 마지막 OpenSearch Service 프로세스 스타트업 시점부터 계산됩니다. 관련 통계: 합계
ReplicationLagMaxTime	복제본 샤드가 기본 샤드 뒤에 있는 밀리초 단위의 시간. 관련 통계: Maximum

알림 지표

Amazon OpenSearch Service는 알림에 대한 다음 지표를 제공합니다.

지표	설명
AlertingDegraded	값이 1이면 알림 인덱스가 빨간색이거나 하나 이상의 노드가 일정에 따라 실행되지 않음을 의미하고, 값이 0이면 정상적인 동작 상태를 나타냅니다. 관련 통계: Maximum
AlertingIndexExists	값이 1이면 .opensearch-alerting-config 인덱스가 존재함을 의미하고, 값이 0이면 존재하지 않음을 의미합니다. 알림 기능을 처음 사용할 때까지 이 값은 0으로 유지됩니다. 관련 통계: Maximum
AlertingIndexStatus.green	인덱스의 상태입니다. 값이 1이면 녹색을 의미하고, 값이 0이면 인덱스가 존재하지 않거나 녹색이 아님을 의미합니다. 관련 통계: Maximum
AlertingIndexStatus.red	인덱스의 상태입니다. 값이 1이면 빨간색을 의미하고, 값이 0이면 인덱스가 존재하지 않거나 빨간색이 아님을 의미합니다. 관련 통계: Maximum
AlertingIndexStatus.yellow	인덱스의 상태입니다. 값이 1이면 노란색을 의미하고, 값이 0이면 인덱스가 존재하지 않거나 노란색이 아님을 의미합니다. 관련 통계: Maximum
AlertingNodesNotOnSchedule	값이 1이면 일부 작업이 일정에 따라 실행되고 있지 않음을 의미하고, 값이 0이면 모든 알림 작업이 일정에 따라 실행 중이거나 알림 작업이 없음을 의미합니다. OpenSearch Service 콘솔을 점검하거나 _nodes/stats 요청을 실행하여 리소스 사용량이 높은 노드가 있는지 확인합니다. 관련 통계: Maximum
AlertingNodesOnSchedule	값이 1이면 모든 알림 작업이 일정에 따라 실행 중이거나 알림 작업이 없음을 의미하고, 값이 0이면 일부 작업이 일정에 따라 실행되고 있지 않음을 의미합니다. 관련 통계: Maximum

지표	설명
AlertingScheduledJobEnabled	값이 1이면 opensearch.scheduled_jobs.enabled 클러스터 설정이 true임을 의미하고, 값이 0이면 false이며 예약된 작업이 비활성화되었음을 의미합니다. 관련 통계: Maximum

이상 탐지 지표

Amazon OpenSearch Service는 이상 탐지에 대한 다음 지표를 제공합니다.

지표	설명
ADPluginUnhealthy	값이 1이면 실패 횟수가 많거나 사용하는 인덱스 중 하나가 빨간색이기 때문에 이상 탐지 플러그 인이 제대로 작동하지 않음을 의미합니다. 값이 0이면 플러그인이 예상대로 작동하고 있음을 나타냅니다. 관련 통계: Maximum
ADExecuteRequestCount	이상을 탐지하기 위한 요청 수입니다. 관련 통계: 합계
ADExecuteFailureCount	이상을 탐지하기 위한 실패한 요청 수입니다. 관련 통계: 합계
ADHCExecuteFailureCount	높은 카디널리티 탐지기를 위한 이상 탐지 요청 중 실패한 요청 수입니다. 관련 통계: 합계
ADHCExecuteRequestCount	높은 카디널리티 탐지기를 위한 이상 탐지 요청 수입니다. 관련 통계: 합계
ADAnomalyResultsIndexStatusIndexExists	값이 1이면 .opensearch-anomaly-results 별칭이 가리키는 인덱스가 존재함을 의미합니다. 이상 탐지를 처음 사용할 때까지 이 값은 0으로 유지됩니다.

지표	설명
	관련 통계: Maximum
ADAnomaly ResultsIn dexStatus.red	값이 1이면 .opensearch-anomaly-results 별칭이 가리키는 인덱스가 빨간색임을 의미합니다. 값이 0이면 그렇지 않음을 의미합니다. 이상 탐지를 처음 사용할 때까지 이 값은 0으로 유지됩니다. 관련 통계: Maximum
ADAnomaly Detectors IndexStat usIndexExists	값이 1이면 .opensearch-anomaly-detectors 인덱스가 존재함을 의미하고, 값이 0이면 존재하지 않음을 의미합니다. 이상 탐지를 처음 사용할 때까지 이 값은 0으로 유지됩니다. 관련 통계: Maximum
ADAnomaly Detectors IndexStat us.red	값이 1이면 .opensearch-anomaly-detectors 인덱스가 빨간색임을 의미합니다. 값이 0이면 그렇지 않음을 의미합니다. 이상 탐지를 처음 사용할 때까지 이 값은 0으로 유지됩니다. 관련 통계: Maximum
ADModelsC heckpoint IndexStat usIndexExists	값이 1이면 .opensearch-anomaly-checkpoints 인덱스가 존재함을 의미하고, 값이 0이면 존재하지 않음을 의미합니다. 이상 탐지를 처음 사용할 때까지 이 값은 0으로 유지됩니다. 관련 통계: Maximum
ADModelsC heckpoint IndexStat us.red	값이 1이면 .opensearch-anomaly-checkpoints 인덱스가 빨간색임을 의미합니다. 값이 0이면 그렇지 않음을 의미합니다. 이상 탐지를 처음 사용할 때까지 이 값은 0으로 유지됩니다. 관련 통계: Maximum

비동기 검색 지표

Amazon OpenSearch Service는 비동기 검색에 대한 다음 지표를 제공합니다.

비동기 검색 코디네이터 노드 통계(코디네이터 노드당)

지표	설명
AsynchronousSearchSubmissionRate	지난 1분 동안 제출된 비동기 검색 수입니다.
AsynchronousSearchInitializedRate	지난 1분 동안 제출된 비동기 검색 수입니다.
AsynchronousSearchRunningCurrent	현재 실행 중인 비동기 검색 수입니다.
AsynchronousSearchCompletionRate	지난 1분 동안 성공적으로 완료한 비동기 검색 수입니다.
AsynchronousSearchFailureRate	지난 1분 동안 완료 및 실패한 비동기 검색 수입니다.
AsynchronousSearchPersistRate	지난 1분 동안 지속된 비동기 검색 수입니다.
AsynchronousSearchPersistFailedRate	지난 1분 동안 지속되지 못한 비동기 검색 수입니다.
AsynchronousSearchRejected	노드 작동 시간 이후 거부된 총 비동기 검색 수입니다.

지표	설명
AsynchronousSearchCancelled	노드 작동 시간 이후 취소된 총 비동기 검색 수입니다.
AsynchronousSearchMaxRunningTime	지난 1분 동안 노드에서 가장 오래 실행되는 비동기 검색의 지속 시간입니다.

비동기 검색 클러스터 통계

지표	설명
AsynchronousSearchStoreHealth	지난 1분 동안 지속된 인덱스(빨간색/비 빨간색)에 있는 스토어의 상태입니다.
AsynchronousSearchStoreSize	지난 1분 동안 모든 샤드에 있는 시스템 인덱스의 크기입니다.
AsynchronousSearchStoredResponseCount	지난 1분 동안 시스템 인덱스에 저장된 응답 수입니다.

지표 자동 조정

Amazon OpenSearch Service는 자동 조정에 대한 다음 지표를 제공합니다.

지표	설명
AutoTuneChangesHistoryHeapSize	힙 크기 조정 값에 대한 MiB 변경 기록.

지표	설명
AutoTuneChangesHistoryJVMYoungGenArgs	JVM YongGen 인수 변경 기록.
AutoTuneFailed	자동 조정 변경에 실패했는지 여부를 나타내는 부울입니다.
AutoTuneSucceeded	자동 조정 변경에 성공했는지 여부를 나타내는 부울입니다.
AutoTuneValue	무중단 변경에 대한 대기열 변경 기록(개수) 및 캐시 조정 변경 기록(MiB 단위).

Multi-AZ with Standby 지표

Amazon OpenSearch Service는 Multi-AZ with Standby에 대한 다음 지표를 제공합니다.

활성 가용 영역의 데이터 노드에 대한 노드 수준 지표

지표	설명
CPUUtilization	클러스터의 데이터 노드에 대한 CPU 사용량 백분율입니다. 최대는 CPU 사용량이 가장 높은 노드를 나타냅니다. 평균은 클러스터의 모든 노드를 나타냅니다. 이 지표는 개별 노드에도 사용할 수 있습니다.
FreeStorageSpace	클러스터에서 사용할 수 있는 데이터 노드 공간입니다. Sum은 클러스터의 사용 가능한 전체 공간을 표시하지만, 정확한 값을 얻으려면 이 기간을 1분으로 두어야 합니다. Minimum, Maximum은 사용 가능한 공간이 가장 작은 노드와 가장 큰 노드를 각각 표시합니다. 이 지표는 개별 노드에도 사용할 수 있습니다. OpenSearch Service는 이 지표가 0에 도달하는 경우 ClusterBlockException 를 발생시킵니다. 복구하려면 인덱스를 삭제하거나, 더 큰 인스턴스를 추가하거나 기존 인스턴스에 EBS 기반 스토리지를 추가해야 합니다. 자세한 내용은 the section called “사용 가능한 스토리지 공간 부족” 섹션을 참조하세요.

지표	설명
	OpenSearch Service 콘솔은 이 값을 GiB로 표시합니다. Amazon CloudWatch 콘솔은 이 값을 MiB로 표시합니다.
JVMMemory Pressure	클러스터의 모든 데이터 노드에 사용된 Java 힙의 최대 비율입니다. OpenSearch Service는 Java 힙에 인스턴스 RAM의 절반을 사용합니다(최대 힙 크기 32GiB). 인스턴스를 최대 64GiB의 RAM까지 수직 확장할 수 있으며 인스턴스를 추가하면 수평 확장도 가능합니다. the section called “권장되는 CloudWatch 경보” 섹션을 참조하세요.
SysMemory Utilization	사용 중인 인스턴스 메모리의 비율(%)입니다. 이 지표의 값이 큰 것은 정상이며 일반적으로 클러스터에 문제가 있음을 나타내지 않습니다. 잠재적인 성능 및 안정성 문제에 대한 더 나은 지표는 JVMMemoryPressure 지표를 참조하세요.
IndexingLatency	한 노드의 모든 인덱싱 작업에 소요된 총 시간 차이(밀리초)로, 이 차이는 분 N에서 (N-1)분입니다.
IndexingRate	분당 인덱싱 작업 수입니다.
SearchLatency	한 노드의 모든 검색 작업에 소요된 총 시간 차이(밀리초)로, 이 차이는 분 N에서 (N-1)분입니다.
SearchRate	한 데이터 노드의 모든 샤드에 대한 분당 검색 요청의 총 수입니다.
ThreadpoolSearchQueue	검색 스레드 풀에서 대기 중인 작업의 수입니다. 대기열 크기가 지속해서 높으면 클러스터 확장을 고려합니다. 검색 대기열의 최대 크기는 1,000입니다.
ThreadpoolWriteQueue	쓰기 스레드 풀에서 대기 중인 작업의 수입니다.
ThreadpoolSearchRejected	검색 스레드 풀에서 거부된 작업의 수입니다. 이 수가 계속 증가하면 클러스터 확장을 고려합니다.
ThreadpoolWriteRejected	쓰기 스레드 풀에서 거부된 작업의 수입니다.

활성 가용 영역의 클러스터에 대한 클러스터 수준 지표

지표	설명
DataNodes	활성 및 대기 샤드의 총 수입니다.
DataNodes Shards.active	활성 기본 및 복제본 샤드의 총 수입니다.
DataNodes Shards.unassigned	클러스터의 노드에 할당되지 않은 샤드 수입니다.
DataNodes Shards.initializing	초기화 중인 샤드 수입니다.
DataNodes Shards.relocating	재배치 중인 샤드 수입니다.

가용 영역 회전 지표

ActiveReads.*Availability-Zone* = 1인 경우 영역이 활성 상태입니다.
ActiveReads.*Availability-Zone* = 0인 경우 영역이 대기 상태입니다.

특정 시점 지표

Amazon OpenSearch Service는 특정 시점(PIT) 검색에 대한 다음 지표를 제공합니다.

PIT 코디네이터 노드 통계(코디네이터 노드당)

지표	설명
CurrentPointInTime	노드의 활성 PIT 검색 컨텍스트 수입니다.
TotalPointInTime	노드 작동 시간 이후 만료된 PIT 검색 컨텍스트 수입니다.

지표	설명
AvgPointInTimeAliveTime	노드 작동 시간 이후 적용된 평균 PIT 검색 컨텍스트입니다.
HasActivePointInTime	값이 1이면 노드 가동 시간 이후 노드에 활성 PIT 컨텍스트가 있음을 나타냅니다. 값이 0이면 없는 것입니다.
HasUsedPointInTime	값이 1이면 노드 가동 시간 이후 노드에 활성 PIT 컨텍스트가 있음을 나타냅니다. 값이 0이면 없는 것입니다.

SQL 지표

Amazon OpenSearch Service는 SQL 지원에 대한 다음 지표를 제공합니다.

지표	설명
SQLFailedRequestCountByCusErr	클라이언트 문제로 인해 실패한 _sql API에 대한 요청 수입니다. 예를 들어 IndexNotFoundException 으로 인해 요청이 HTTP 상태 코드 400을 반환할 수 있습니다. 관련 통계: 합계
SQLFailedRequestCountBySysErr	서버 문제 또는 기능 제한으로 인해 실패한, _sql API에 대한 요청 수입니다. 예를 들어 VerificationException 으로 인해 요청이 HTTP 상태 코드 503을 반환할 수 있습니다. 관련 통계: 합계
SQLRequestCount	_sql API 요청 수입니다. 관련 통계: 합계
SQLDefaultCursorRequestCount	SQLRequestCount 와 유사하지만 페이지 매김 요청만 계산합니다. 관련 통계: 합계
SQLUnhealthy	값이 1이면 특정 요청에 대한 응답으로 SQL 플러그인이 5xx 응답 코드를 반환하거나 잘못된 쿼리 DSL을 OpenSearch에 전달함을 나타냅니

지표	설명
	다. 다른 요청은 계속 성공합니다. 값이 0이면 최근 실패가 없음을 나타냅니다. 지속해서 값이 1이면 클라이언트가 플러그인에 수행하는 요청 문제를 해결합니다. 관련 통계: Maximum

k-NN 지표

Amazon OpenSearch Service에는 k-nearest neighbor(k-NN) 플러그인에 대한 다음 지표가 포함됩니다.

지표	설명
KNNCacheCapacityReached	캐시 용량에 도달했는지에 대한 노드별 지표입니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: Maximum
KNNCircuitBreakerTriggered	회로 차단기가 트리거되는지 여부에 대한 클러스터별 지표입니다. 어떤 노드가 KNNCacheCapacityReached 에 대한 1의 값을 반환하는 경우 이 값도 1을 반환합니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: Maximum
KNNEvictionCount	메모리 제약 조건 또는 유휴 시간으로 인해 캐시에서 제거된 그래프 수에 대한 노드별 지표입니다. 인덱스 삭제로 인해 발생하는 명시적 제거는 계산되지 않습니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: 합계
KNNGraphIndexErrors	문서의 knn_vector 필드를 오류를 생성한 그래프에 추가하려는 요청 수에 대한 노드별 지표입니다. 관련 통계: 합계

지표	설명
KNNGraphIndexRequests	문서의 knn_vector 필드를 그래프에 추가하려는 요청 수에 대한 노드별 지표입니다. 관련 통계: 합계
KNNGraphMemoryUsage	현재 캐시 크기(메모리에 있는 모든 그래프의 총 크기)에 대한 노드별 지표입니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: Average
KNNGraphQueryErrors	오류를 생성한 그래프 쿼리 수에 대한 노드별 지표입니다. 관련 통계: 합계
KNNGraphQueryRequests	그래프 쿼리 수에 대한 노드별 지표입니다. 관련 통계: 합계
KNNHitCount	캐시 적중 수에 대한 노드별 지표입니다. 캐시 적중은 사용자가 이미 메모리에 로드된 그래프를 쿼리할 때 발생합니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: 합계
KNNLoadExceptionCount	그래프를 캐시로 로드하려고 시도하는 동안 예외가 발생한 횟수에 대한 노드별 지표입니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: 합계
KNNLoadSuccessCount	플러그인이 그래프를 캐시에 성공적으로 로드한 횟수에 대한 노드별 지표입니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: 합계

지표	설명
KNNMissCount	캐시 누락 수에 대한 노드별 지표입니다. 캐시 누락은 사용자가 아직 메모리에 로드되지 않은 그래프를 쿼리할 때 발생합니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: 합계
KNNQueryRequests	k-NN 플러그인이 받은 쿼리 요청 수에 대한 노드별 지표입니다. 관련 통계: 합계
KNNScriptCompilationErrors	스크립트 컴파일 중 오류 수에 대한 노드별 지표입니다. 이 통계는 k-NN 점수 스크립트 검색에만 관련이 있습니다. 관련 통계: 합계
KNNScriptCompilations	k-NN 스크립트가 컴파일된 횟수에 대한 노드별 지표입니다. 이 값은 일반적으로 1 또는 0이어야 하지만 컴파일된 스크립트가 포함된 캐시가 채워지면 k-NN 스크립트가 다시 컴파일될 수 있습니다. 이 통계는 k-NN 점수 스크립트 검색에만 관련이 있습니다. 관련 통계: 합계
KNNScriptQueryErrors	스크립트 쿼리 중 오류 수에 대한 노드별 지표입니다. 이 통계는 k-NN 점수 스크립트 검색에만 관련이 있습니다. 관련 통계: 합계
KNNScriptQueryRequests	총 스크립트 쿼리 수에 대한 노드별 지표입니다. 이 통계는 k-NN 점수 스크립트 검색에만 관련이 있습니다. 관련 통계: 합계
KNNTotalLoadTime	k-NN이 그래프를 캐시로 로드하는 데 소요된 시간(나노초)입니다. 이 지표는 대략적인 k-NN 검색에만 관련됩니다. 관련 통계: 합계

클러스터 간 검색 지표

Amazon OpenSearch Service는 클러스터 간 검색에 대한 다음 지표를 제공합니다.

소스 도메인 지표

지표	차원	설명
CrossClusterOutboundConnections	ConnectionId	연결된 노드 수입니다. 응답에 하나 이상의 건너뛴 도메인이 포함된 경우 이 지표를 사용하여 비정상 연결을 추적합니다. 이 숫자가 0으로 떨어지면 연결이 비정상입니다.
CrossClusterOutboundRequests	ConnectionId	대상 도메인으로 전송된 검색 요청 수입니다. 클러스터 간 검색 요청의 부하가 도메인에 너무 부담되는지 확인하고 이 지표의 스파이크와 JVM/CPU 스파이크의 상관관계를 분석하는 데 사용합니다.

대상 도메인 지표

지표	차원	설명
CrossClusterInboundRequests	ConnectionId	소스 도메인에서 받은 수신 연결 요청 수입니다.

예기치 않게 연결이 끊어지는 경우 CloudWatch 경보를 추가합니다. 경보를 생성하는 단계는 정적 임계값을 기반으로 CloudWatch 경보 생성을 참조하세요.

클러스터 간 복제 지표

Amazon OpenSearch Service는 클러스터 간 복제에 대한 다음 지표를 제공합니다.

지표	설명
ReplicationRate	초당 평균 복제 작업 속도. 이 지표는 IndexingRate 지표와 유사합니다.

지표	설명
LeaderCheckPoint	특정 연결에 대한 모든 복제 인덱스에 걸친 리더 체크포인트의 합계입니다. 이 지표를 사용하여 복제 대기 시간을 측정할 수 있습니다.
FollowerCheckPoint	특정 연결에 대한 모든 복제 인덱스에 걸친 팔로워 체크포인트의 합계입니다. 이 지표를 사용하여 복제 대기 시간을 측정할 수 있습니다.
ReplicationNumSyncingIndices	복제 상태가 SYNCING인 인덱스의 수입니다.
ReplicationNumBootstrappingIndices	복제 상태가 BOOTSTRAPPING 인 인덱스의 수입니다.
ReplicationNumPausedIndices	복제 상태가 PAUSED인 인덱스의 수입니다.
ReplicationNumFailedIndices	복제 상태가 FAILED인 인덱스의 수입니다.
CrossClusterOutboundReplicationRequests	팔로워 도메인의 복제 전송 요청 수입니다. 전송 요청은 내부적이며 복제 API 작업이 호출될 때마다 발생합니다. 팔로워 도메인 폴이 리더 도메인에서 변경될 때도 발생합니다.
CrossClusterInboundReplicationRequests	리더 도메인의 복제 전송 요청 수입니다. 전송 요청은 내부적이며 복제 API 작업이 호출될 때마다 발생합니다.

지표	설명
AutoFollowNumSuccessStartReplication	특정 연결에 대한 복제 규칙에 의해 성공적으로 생성된 팔로워 인덱스의 수입니다.
AutoFollowNumFailedStartReplication	일치하는 패턴이 있을 때 복제 규칙에 의해 생성되지 못한 팔로워 인덱스의 수입니다. 이 문제는 원격 클러스터의 네트워크 문제 또는 보안 문제(즉, 연결된 역할에 복제를 시작할 권한이 없음)로 인해 발생할 수 있습니다.
AutoFollowLeaderCallFailure	새 데이터를 가져오기 위해 팔로워 인덱스에서 리더 인덱스로의 쿼리가 실패했는지 여부입니다. 값 1은 최근 1분 동안 1회 이상의 실패한 호출이 있음을 의미합니다.

순위 학습 지표

Amazon OpenSearch Service는 순위 학습에 대한 다음 지표를 제공합니다.

지표	설명
LTRRequestTotalCount	순위 요청의 총 수입니다.
LTRRequestErrorCount	실패한 요청의 총 수입니다.
LTRStatus.red	플러그 인을 실행하는 데 필요한 인덱스 중 하나가 빨간색인지 추적합니다.
LTRMemoryUsage	플러그인이 사용하는 총 메모리입니다.
LTRFeatureMemoryUsageInBytes	순위 학습 기능 필드에서 사용되는 메모리의 양(바이트)입니다.

지표	설명
LTRFeaturesetMemoryUsageInBytes	모든 순위 학습 기능 집합에서 사용되는 메모리의 양(바이트)입니다.
LTRModelMemoryUsageInBytes	모든 순위 학습 모델에서 사용되는 메모리의 양(바이트)입니다.

파이프 처리 언어 지표

Amazon OpenSearch Service는 파이프 처리 언어에 대한 다음 지표를 제공합니다.

지표	설명
PPLFailedRequestCountByCusErr	클라이언트 문제로 인해 실패한 _ppl API에 대한 요청 수입니다. 예를 들어 IndexNotFoundException 으로 인해 요청이 HTTP 상태 코드 400을 반환할 수 있습니다.
PPLFailedRequestCountBySysErr	서버 문제 또는 기능 제한으로 인해 실패한, _ppl API에 대한 요청 수입니다. 예를 들어 VerificationException 으로 인해 요청이 HTTP 상태 코드 503을 반환할 수 있습니다.
PPLRequestCount	_ppl API 요청 수입니다.

Amazon CloudWatch Logs를 사용하여 OpenSearch 로그 모니터링

Amazon OpenSearch Service는 Amazon CloudWatch Logs를 통해 다음과 같은 OpenSearch 로그를 노출합니다.

- 오류 로그
- 느린 검색 요청 로그
- 느린 샤드 로그
- 감사 로그

느린 검색 샤드 로그, 느린 인덱싱 느린 로그 및 오류 로그는 성능 및 안정성 문제 해결에 유용합니다. 감사 로그는 규정 준수를 위해 사용자 활동을 추적합니다. 모든 로그는 기본적으로 비활성화되어 있습니다. 활성화되면, 표준 CloudWatch 요금이 적용됩니다.

> **Note**
>
> 오류 로그는 OpenSearch 및 Elasticsearch 버전 5.1 이상에서만 사용할 수 있습니다. 느린 로그는 모든 OpenSearch 및 Elasticsearch 버전에서 사용할 수 있습니다.

OpenSearch는 로그에 대해 Apache Log4j 2 및 `TRACE`, `DEBUG`, `INFO`, `WARN`, `ERROR`, `FATAL`의 내장형 로그 수준(최저~최고 수준의 심각도)을 사용합니다.

오류 로그를 활성화하면 OpenSearch Service에서는 `WARN`,`ERROR` 및 `FATAL`의 로그 줄을 CloudWatch에 게시합니다. 또한 OpenSearch Service는 `DEBUG` 수준에서 다음을 비롯한 여러 제외 항목을 게시합니다.

- `org.opensearch.index.mapper.MapperParsingException`
- `org.opensearch.index.query.QueryShardException`
- `org.opensearch.action.search.SearchPhaseExecutionException`
- `org.opensearch.common.util.concurrent.OpenSearchRejectedExecutionException`
- `java.lang.IllegalArgumentException`

오류 로그는 다음을 포함하여 많은 상황에서 문제를 해결하는 데 도움이 될 수 있습니다.

- Painless 스크립트 컴파일 문제
- 잘못된 쿼리
- 인덱싱 문제
- 스냅샷 실패
- 인덱스 상태 관리 마이그레이션 실패

> **Note**
>
> 모든 오류가 오류 로그에 보고되지는 않습니다.

> **Note**
>
> OpenSearch Service는 발생하는 모든 오류를 로깅하지 않습니다.

주제

- 로그 게시 활성화(콘솔)
- 로그 게시 활성화(AWS CLI)
- 로그 게시 활성화(AWS SDK)
- 로그 게시 활성화(CloudFormation)
- 느린 검색 요청 로그 임계치 설정
- 느린 샤드 로그 임계치 설정
- 느린 로그 테스트
- 로그 보기

로그 게시 활성화(콘솔)

OpenSearch Service 콘솔은 CloudWatch에 대한 로그 게시를 활성화하는 가장 간편한 방법입니다.

CloudWatch에 대한 로그 게시를 활성화하려면(콘솔)

1. aws.amazon.com 이동한 다음 로그인을 선택하고 자격 증명을 제공합니다.
2. 분석(Analytics)에서 Amazon OpenSearch Service를 선택합니다.
3. 업데이트할 도메인을 선택합니다.
4. [로그(Logs)] 탭에서 로그 유형을 선택하고 [사용(Enable)]을 선택합니다.
5. 새 CloudWatch 로그 그룹을 생성하거나 기존 로그 그룹을 선택합니다.

   > **Note**
   >
   > 여러 로그를 활성화하려는 경우 자체 로그 그룹에 각각 게시하는 것이 좋습니다. 이렇게 분리하면 로그를 더 쉽게 검사할 수 있습니다.

6. 적절한 사용 권한이 포함된 액세스 정책을 선택하거나 콘솔에서 제공하는 JSON을 사용하여 정책을 만듭니다.

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:*"
    }
  ]
}
```

혼동된 대리자 문제로부터 자신을 보호하기 위하여 aws:SourceAccount 및 aws:SourceArn 조건 키를 정책에 추가할 것을 권장합니다. 소스 계정은 도메인의 소유자이고 소스 ARN은 도메인의 ARN입니다. 이러한 조건 키를 추가하려면 도메인에 서비스 소프트웨어 R20211203 이상을 사용해야 합니다.

예를 들어 정책에 다음 조건 블록을 추가할 수 있습니다.

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

> **Important**
>
> CloudWatch Logs는 리전당 10개의 리소스 정책을 지원합니다. 여러 OpenSearch Service 도메인에 대해 로그를 사용하도록 설정하려는 경우, 이 제한에 도달하지 않도록 하려면 여러 로그 그룹을 포함하는 더 광범위한 정책을 생성하여 재사용해야 합니다. 정책 업데이트 단계는 the section called "로그 게시 활성화(AWS CLI)" 섹션을 참조하세요.

7. 활성화(Enable)를 선택합니다.

   도메인 상태가 활성(Active)에서 처리 중(Processing)으로 바뀝니다. 상태가 다시 활성(Active)으로 돌아온 다음에 로그 게시를 활성화해야 합니다. 이 변경은 일반적으로 30분이 소요되지만 도메인 구성에 따라 시간이 더 오래 걸릴 수 있습니다.

느린 샤드 로그 중 하나를 활성화한 경우 the section called "느린 샤드 로그 임계치 설정" 섹션을 참조하세요. 감사 로그를 활성화한 경우 the section called "2단계: OpenSearch Dashboards에서 감사 로그 켜기" 섹션을 참조하세요. 오류 로그만 활성화한 경우 추가 구성 단계를 수행할 필요가 없습니다.

로그 게시 활성화(AWS CLI)

로그 게시를 활성화하려면 CloudWatch 로그 그룹이 필요합니다. 아직 없는 경우 다음 명령을 사용하여 생성할 수 있습니다.

```
aws logs create-log-group --log-group-name my-log-group
```

다음 명령을 입력하여 로그 그룹의 ARN을 찾은 다음 이를 기록해 둡니다.

```
aws logs describe-log-groups --log-group-name my-log-group
```

이제 로그 그룹에 작성할 수 있는 권한을 OpenSearch Service에 부여할 수 있습니다. 명령의 끝 부분에 로그 그룹의 ARN을 제공해야 합니다.

```
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 Logs는 리전당 10개의 리소스 정책을 지원합니다. 여러 OpenSearch Service 도메인에 대해 느린 샤드 로그를 사용하도록 설정하려는 경우, 이 제한에 도달하지 않도록 하려면 여러 로그 그룹을 포함하는 더 광범위한 정책을 생성하여 재사용해야 합니다.

나중에 이 정책을 검토해야 하는 경우 `aws logs describe-resource-policies` 명령을 사용합니다. 정책을 업데이트하려면 새 정책 문서에 동일한 `aws logs put-resource-policy` 명령을 실행합니다.

마지막으로, `--log-publishing-options` 옵션을 사용하여 게시를 활성화할 수 있습니다. 옵션에 대한 구문은 `create-domain` 및 `update-domain-config` 명령 둘 다에서 동일합니다.

파라미터	유효한 값
`--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}`
	`AUDIT_LOGS={CloudWatchLogsLogGroupArn=` *`cw_log_group_arn`* `,Enabled=true|false}`

> **Note**
>
> 여러 로그를 활성화하려는 경우 자체 로그 그룹에 각각 게시하는 것이 좋습니다. 이렇게 분리하면 로그를 더 쉽게 검사할 수 있습니다.

예제

다음 예제는 지정된 도메인에 대한 느린 샤드 로그 검색 및 인덱싱의 게시를 활성화합니다.

```
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}"
```

CloudWatch에 대한 게시를 비활성화하려면 `Enabled=false`를 통해 동일한 명령을 실행합니다.

느린 샤드 로그 중 하나를 활성화한 경우 the section called “느린 샤드 로그 임계치 설정” 섹션을 참조하세요. 감사 로그를 활성화한 경우 the section called “2단계: OpenSearch Dashboards에서 감사 로그 켜기” 섹션을 참조하세요. 오류 로그만 활성화한 경우 추가 구성 단계를 수행할 필요가 없습니다.

로그 게시 활성화(AWS SDK)

로그 게시를 활성화하려면 먼저 CloudWatch 로그 그룹을 생성하고, ARN을 얻고, OpenSearch Service에 작성할 수 있는 권한을 부여해야 합니다. 관련 작업은 Amazon CloudWatch Logs API 참조에 문서화되어 있습니다.

- `CreateLogGroup`
- `DescribeLogGroup`
- `PutResourcePolicy`

AWS SDK를 사용하여 이 작업에 액세스할 수 있습니다.

AWS SDKs(Android 및 iOS SDKs 제외)는 `CreateDomain` 및에 대한 `--log-publishing-options` 옵션을 포함하여 Amazon OpenSearch Service API 참조에 정의된 모든 작업을 지원합니다`UpdateDomainConfig`.

느린 샤드 로그 중 하나를 활성화한 경우 the section called “느린 샤드 로그 임계치 설정” 섹션을 참조하세요. 오류 로그만 활성화한 경우 추가 구성 단계를 수행할 필요가 없습니다.

로그 게시 활성화(CloudFormation)

이 예제에서는 CloudFormation을 사용하여 `opensearch-logs`라는 로그 그룹을 생성하고 적절한 권한을 할당한 다음, 애플리케이션 로그, 느린 검색 샤드 로그 및 느린 인덱싱 로그에 대한 로그 게시가 활성화된 도메인을 생성합니다.

로그 게시를 활성화하려면 CloudWatch 로그 그룹을 생성해야 합니다.

```
Resources:
  OpenSearchLogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      LogGroupName: opensearch-logs
Outputs:
  Arn:
    Value:
      'Fn::GetAtt':
        - OpenSearchLogGroup
        - Arn
```

템플릿은 로그 그룹의 ARN을 출력합니다. 이 경우 ARN은 `arn:aws:logs:us-east-1:123456789012:log-group:opensearch-logs`입니다.

ARN을 사용하여 로그 그룹에 작성할 수 있는 권한을 OpenSearch Service를 부여하는 리소스 정책을 만듭니다.

```
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:*\"}]}"
```

마지막으로 로그 게시를 사용하여 OpenSearch 서비스 도메인을 생성하는 다음 CloudFormation 스택을 생성합니다. 액세스 정책은의 사용자가 도메인에 대한 모든 HTTP 요청을 할 수 AWS 계정 있도록 허용합니다.

```
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
```

자세한 구문 정보는 AWS CloudFormation 사용 설명서의 로그 게시 옵션을 참조하세요.

느린 검색 요청 로그 임계치 설정

느린 검색 요청 로그는 버전 2.13 이상에서 실행되는 OpenSearch Service 도메인에서 검색할 수 있습니다. 느린 검색 요청 로그 임계치는 총 요청 소요 시간에 대해 구성됩니다. 이는 개별 샤드 소요 시간에 대해 구성된 느린 샤드 요청 로그와 다릅니다.

클러스터 설정을 사용하여 느린 검색 요청 로그를 지정할 수 있습니다. 이는 인덱스 설정으로 활성화하는 느린 샤드 로그와 다릅니다. 예를 들어 OpenSearch REST API를 통해 다음 설정을 지정할 수 있습니다.

```
PUT domain-endpoint/_cluster/settings
{
  "transient": {
    "cluster.search.request.slowlog.threshold.warn": "5s",
    "cluster.search.request.slowlog.threshold.info": "2s"
  }
}
```

느린 샤드 로그 임계치 설정

OpenSearch는 기본적으로 느린 샤드 로그를 비활성화합니다. CloudWatch에 느린 샤드 로그 게시를 활성화한 후 각 OpenSearch 인덱스에 대한 로깅 임계치를 지정해야 합니다. 이러한 임계값은 정확하게 기록할 내용과 로그 수준을 정의합니다.

예를 들어 OpenSearch REST API를 통해 이러한 설정을 지정할 수 있습니다.

```
PUT domain-endpoint/index/_settings
{
  "index.search.slowlog.threshold.query.warn": "5s",
  "index.search.slowlog.threshold.query.info": "2s"
}
```

느린 로그 테스트

느린 검색 요청 및 샤드 로그가 성공적으로 게시되고 있는지 테스트하려면 매우 낮은 값으로 시작하여 로그가 CloudWatch에 나타나는지 확인한 다음, 임계치를 더 유용한 수준으로 늘립니다.

로그가 나타나지 않는 경우 다음 정보를 확인합니다.

- CloudWatch 로그 그룹이 있습니까? CloudWatch 콘솔을 확인합니다.
- 로그 그룹에 작성할 수 있는 권한이 OpenSearch Service에 있습니까? OpenSearch Service 콘솔을 확인합니다.
- OpenSearch Service 도메인이 로그 그룹에 게시되도록 구성되었습니까? OpenSearch Service 콘솔을 확인하거나, 옵션을 사용하거나 AWS CLI `describe-domain-config`, SDK 중 하나를 `DescribeDomainConfig` 사용하여를 호출합니다. SDKs
- 요청이 해당 값을 초과할 만큼 OpenSearch 로깅 임계값이 낮습니까?

  도메인에 대한 느린 검색 요청 로그 임계치를 검토하려면 다음 명령을 사용합니다.

```
GET domain-endpoint/_cluster/settings?flat_settings
```

인덱스에 대한 느린 샤드 로그 임계치를 검토하려면 다음 명령을 사용합니다.

```
GET domain-endpoint/index/_settings?pretty
```

인덱스에 대해 느린 로그를 사용하지 않으려면 변경한 임계값을 -1의 기본값으로 되돌립니다.

OpenSearch Service 콘솔을 사용하여 CloudWatch에 게시를 비활성화하거나 OpenSearch의 로그 생성을 중지하지 AWS CLI 않고 해당 로그의 게시만 중지합니다. 느린 샤드 로그가 더 이상 필요하지 않은 경우 인덱스 설정을 확인하고 느린 검색 요청 로그가 더 이상 필요하지 않은 경우 도메인 설정을 확인해야 합니다.

로그 보기

CloudWatch에서 애플리케이션 및 느린 로그를 보는 것은 다른 CloudWatch 로그를 보는 것과 같습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 로그 데이터 보기를 참조하세요.

다음은 로그를 볼 때 고려해야 할 몇 가지 사항입니다.

- OpenSearch Service는 각 줄에 있는 처음 255,000개 문자만 CloudWatch에 게시합니다. 남아 있는 모든 콘텐츠는 잘리게 됩니다. 감사 로그의 경우 메시지당 10,000자입니다.
- CloudWatch에서 로그 스트림 이름에는 콘텐츠를 식별하는 데 도움이 되는 `-index-slow-logs`, `-search-slow-logs`, `-application-logs` 및 `-audit-logs` 접미사가 있습니다.

Amazon OpenSearch Service의 감사 로그 모니터링

Amazon OpenSearch Service 도메인에서 세분화된 액세스 제어를 사용하는 경우 데이터에 대한 감사 로그를 활성화할 수 있습니다. 감사 로그는 고도로 사용자 정의할 수 있으며 인증 성공 및 실패, OpenSearch에 요청, 인덱스 변경, 수신 검색 쿼리 등 OpenSearch 클러스터에서의 사용자 활동을 추적할 수 있습니다. 기본 구성은 자주 사용되는 사용자 작업 집합을 추적하지만 정확한 요구 사항에 맞게 설정을 조정하는 것이 좋습니다.

OpenSearch 애플리케이션 로그 및 느린 로그와 마찬가지로 OpenSearch Service는 CloudWatch Logs를 통해 감사 로그를 게시합니다. 활성화되면, 표준 CloudWatch 요금이 적용됩니다.

> **Note**
>
> 감사 로그를 활성화하려면 사용자 역할이 `security_manager` 역할에 매핑되어OpenSearch `plugins/_security` REST API에 액세스할 수 있어야 합니다. 자세한 내용은 the section called “마스터 사용자 수정” 섹션을 참조하세요.

주제

- 제한 사항
- 감사 로그 활성화
- 를 사용하여 감사 로깅 활성화 AWS CLI
- 구성 API를 사용하여 감사 로깅 활성화
- 감사 로그 계층 및 범주
- 감사 로그 설정
- 감사 로그 예제
- REST API를 사용하여 감사 로그 구성

제한 사항

감사 로그에는 다음과 같은 제한 사항이 있습니다.

- 감사 로그에는 대상의 도메인 액세스 정책에 의해 거부된 클러스터 간 검색 요청이 포함되지 않습니다.
- 각 감사 로그 메시지의 최대 크기는 10,000자입니다. 이 제한을 초과하면 감사 로그 메시지가 잘립니다.

감사 로그 활성화

감사 로그를 활성화하는 절차는 두 단계로 이루어져 있습니다. 먼저 감사 로그를 CloudWatch Logs에 게시하도록 도메인을 구성합니다. 그런 다음 OpenSearch Dashboards에서 감사 로그를 활성화하고 필요에 맞게 구성합니다.

> **Important**
>
> 이 단계를 수행하는 동안 오류가 발생하면 the section called “감사 로그를 활성화할 수 없음”에서 문제 해결 정보를 참조하세요.

1단계: 감사 로그 활성화 및 액세스 정책 구성

다음 단계에서는 콘솔을 사용하여 감사 로그를 활성화하는 방법을 설명합니다. 또는 OpenSearch Service API를 사용하여 활성화 AWS CLI할 수도 있습니다.

OpenSearch Service 도메인(콘솔)에 대한 감사 로그를 활성화하려면

1. 도메인을 선택하여 구성을 열고 로그(Logs) 탭으로 이동합니다.
2. 감사 로그(Audit logs)를 선택한 후 사용 설정(Enable)을 선택합니다.
3. CloudWatch 로그 그룹을 생성하거나 기존 로그 그룹을 선택합니다.
4. 적절한 사용 권한이 포함된 액세스 정책을 선택하거나 콘솔에서 제공하는 JSON을 사용하여 정책을 만듭니다.

   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/*"
    }
  ]
}
```

혼동된 대리자 문제로부터 자신을 보호하기 위하여 `aws:SourceAccount` 및 `aws:SourceArn` 조건 키를 정책에 추가할 것을 권장합니다. 소스 계정은 도메인의 소유자이고 소스 ARN은 도메인의 ARN입니다. 이러한 조건 키를 추가하려면 도메인에 서비스 소프트웨어 R20211203 이상을 사용해야 합니다.

예를 들어 정책에 다음 조건 블록을 추가할 수 있습니다.

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

5. 활성화를 선택합니다.

2단계: OpenSearch Dashboards에서 감사 로그 켜기

OpenSearch Service 콘솔에서 감사 로그를 활성화한 후 OpenSearch Dashboards에서도 활성화하고 필요에 맞게 구성해야 합니다.

1. OpenSearch Dashboards를 열고 왼쪽 메뉴에서 보안(Security)을 선택합니다.
2. 감사 로그(Audit logs)를 선택합니다.
3. 감사 로깅 활성화(Enable audit logging)를 선택합니다.

Dashboards UI에서는 일반 설정(General settings) 및 규정 준수 설정(Compliance settings)에서 감사 로그 설정을 완전히 제어할 수 있습니다. 모든 구성 옵션에 대한 설명은 감사 로그 설정을 참조하세요.

를 사용하여 감사 로깅 활성화 AWS CLI

다음 AWS CLI 명령은 기존 도메인에 대한 감사 로그를 활성화합니다.

```
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}"
```

도메인을 생성할 때 감사 로그를 활성화할 수도 있습니다. 자세한 내용은 AWS CLI 명령 참조를 참조하세요.

구성 API를 사용하여 감사 로깅 활성화

구성 API에 다음을 요청하면 기존 도메인에서 감사 로그를 활성화할 수 있습니다.

```
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
    }
  }
}
```

자세한 내용은 Amazon OpenSearch Service API 참조를 확인하세요.

감사 로그 계층 및 범주

클러스터 통신은 두 개의 별도 계층인 REST 계층과 전송 계층을 통해 이루어집니다.

- REST 계층은 curl, Logstash, OpenSearch 대시 보드, Java 상위 수준 REST 클라이언트, Python Requests 라이브러리와 같은 HTTP 클라이언트와의 통신을 다룹니다. 이들은 모두 클러스터에 도착하는 모든 HTTP 요청입니다.
- 전송 계층은 노드 간 통신을 다룹니다. 예를 들어, 검색 요청이 REST 계층을 통해 클러스터에 도착한 후 요청을 제공하는 조정 노드는 쿼리를 다른 노드로 보내고 응답을 수신하고 필요한 문서를 수집하여 최종 응답으로 수집합니다. 샤드 할당 및 재조정과 같은 작업도 전송 계층을 통해 이루어집니다.

계층에 대한 개별 감사 범주뿐만 아니라 전체 계층에 대한 감사 로그를 활성화하거나 비활성화할 수 있습니다. 다음 표에는 감사 범주 및 감사 범주가 사용할 수 있는 계층에 대한 요약이 나와 있습니다.

범주	설명	REST 사용 가능	전송 사용 가능
FAILED_LOGIN	요청에 잘못된 자격 증명이 포함되어 있으며 인증에 실패했습니다.	예	예
MISSING_PRIVILEGES	사용자에게 요청을 할 수 있는 권한이 없습니다.	예	예
GRANTED_PRIVILEGES	사용자에게 요청을 할 수 있는 권한이 있었습니다.	예	예
OPENSEARCH_SECURITY_INDEX_ATTEMPT	요청에서 `.opendistro_security` 인덱스를 수정하려고 시도했습니다.	아니요	예
AUTHENTICATED	요청에 유효한 자격 증명이 포함되어 있으며 인증에 성공했습니다.	예	예
INDEX_EVENT	요청에서 인덱스에 대한 관리 작업(예: 인덱스 생성, 별칭 설정 또는 강제 병합 수행)을 수행했습니다. 범주에 포함되는 `indices:admin/` 작업의 전체 목록은 OpenSearch 설명서에서 확인할 수 있습니다.	아니요	예

이러한 표준 범주 외에도 세분화된 액세스 제어는 데이터 규정 준수 요구 사항을 충족하도록 설계된 몇 가지 추가 범주를 제공합니다.

범주	설명
COMPLIANCE_DOC_READ	요청에서 인덱스의 문서에 대해 읽기 이벤트를 수행했습니다.
COMPLIANCE_DOC_WRITE	요청에서 인덱스의 문서에 대해 쓰기 이벤트를 수행했습니다.
COMPLIANCE_INTERNAL_CONFIG_READ	요청에서 `.opendistro_security` 인덱스에 대해 읽기 이벤트를 수행했습니다.
COMPLIANCE_INTERNAL_CONFIG_WRITE	요청에서 `.opendistro_security` 인덱스에 대해 쓰기 이벤트를 수행했습니다.

범주와 메시지 속성을 자유롭게 조합할 수 있습니다. 예를 들어 문서를 인덱싱하기 위해 REST 요청을 보내는 경우 감사 로그에 다음 줄이 표시될 수 있습니다.

- AUTHENTICATED on REST layer (authentication)
- GRANTED_PRIVILEGE on transport layer (authorization)
- COMPLIANCE_DOC_WRITE (document written to an index)

감사 로그 설정

감사 로그에는 다양한 구성 옵션이 있습니다.

일반 설정

일반 설정을 사용하면 개별 범주 또는 전체 계층을 활성화하거나 비활성화할 수 있습니다. GRANTED_PRIVILEGES 및 AUTHENTICATED를 제외된 범주로 남겨 두는 것이 좋습니다. 그렇지 않으면 클러스터에 대한 모든 유효한 요청에 대해 이러한 범주가 기록됩니다.

명칭	백엔드 설정	설명
REST 계층	enable_rest	REST 계층에서 발생하는 이벤트를 활성화하거나 비활성화합니다.

명칭	백엔드 설정	설명
REST 비활성화 범주	disabled_rest_categories	REST 계층에서 무시할 감사 범주를 지정합니다. 이러한 범주를 수정하면 감사 로그의 크기가 많이 늘어날 수 있습니다.
전송 계층	enable_transport	전송 계층에서 발생하는 이벤트를 활성화하거나 비활성화합니다.
전송 비활성화 범주	disabled_transport_categories	전송 계층에서 무시해야 하는 감사 범주를 지정합니다. 이러한 범주를 수정하면 감사 로그의 크기가 많이 늘어날 수 있습니다.

속성 설정을 사용하여 각 로그 행의 세부 정보 양을 사용자 지정할 수 있습니다.

명칭	백엔드 설정	설명
대량 요청	resolve_bulk_requests	이 설정을 활성화하면 대량 요청하는 각 문서에 대한 로그가 생성되므로 감사 로그의 크기가 많이 늘어날 수 있습니다.
요청 본문	log_request_body	요청의 요청 본문을 포함합니다.
인덱스 해석	resolve_indices	별칭을 인덱스로 해석합니다.

무시 설정을 사용하여 사용자 또는 API 경로 집합을 제외합니다.

명칭	백엔드 설정	설명
무시된 사용자	ignore_users	제외할 사용자를 지정합니다.
무시된 요청	ignore_requests	제외할 요청 패턴을 지정합니다.

규정 준수 설정

규정 준수 설정을 사용하면 색인, 문서 또는 필드 수준 액세스를 조정할 수 있습니다.

명칭	백엔드 설정	설명
규정 준수 로깅	enable_compliance	규정 준수 로깅을 활성화하거나 비활성화합니다.

읽기 및 쓰기 이벤트 로깅에 대한 다음 설정을 지정할 수 있습니다.

명칭	백엔드 설정	설명
내부 구성 로깅	internal_config	`.opendistro_security` 인덱스에서 이벤트 로깅을 활성화하거나 비활성화합니다.

읽기 이벤트에 대한 다음 설정을 지정할 수 있습니다.

명칭	백엔드 설정	설명
메타데이터 읽기	read_metadata_only	읽기 이벤트에 대한 메타데이터만 포함합니다. 문서 필드는 포함하지 않습니다.
무시된 사용자	read_ignore_users	읽기 이벤트에 특정 사용자를 포함하지 않습니다.
감시된 필드	read_watched_fields	읽기 이벤트를 감시할 인덱스와 필드를 지정합니다. 감시된 필드를 추가하면 문서 액세스당 하나의 로그가 생성되므로 감사 로그의 크기가 많이 늘어날 수 있습니다. 감시된 필드는 인덱스 패턴 및 필드 패턴을 지원합니다. `{ "index-name-pattern": [ "field-name-pattern" ], "logs*": [ "message" ], "twitter": [ "id", "user*"`

명칭	백엔드 설정	설명
		<pre>] }</pre>

쓰기 이벤트에 대한 다음 설정을 지정할 수 있습니다.

명칭	백엔드 설정	설명
메타데이터 쓰기	write_metadata_only	쓰기 이벤트에 대한 메타데이터만 포함합니다. 문서 필드는 포함하지 않습니다.
로그 차이	write_log_diffs	write_metadata_only가 거짓인 경우 쓰기 이벤트 간의 차이만 포함합니다.
무시된 사용자	write_ignore_users	쓰기 이벤트에 특정 사용자를 포함하지 않습니다.
인덱스 감시	write_watched_indices	쓰기 이벤트를 감시할 인덱스 또는 인덱스 패턴을 지정합니다. 감시된 필드를 추가하면 문서 액세스당 하나의 로그가 생성되므로 감사 로그의 크기가 많이 늘어날 수 있습니다.

감사 로그 예제

이 섹션에는 인덱스의 모든 읽기 및 쓰기 이벤트에 대한 예제 구성, 검색 요청 및 결과 감사 로그가 포함되어 있습니다.

1단계: 감사 로그 구성

CloudWatch Logs 로그 그룹에 감사 로그 게시를 활성화한 후 OpenSearch Dashboards 감사 로깅 페이지로 이동하여 감사 로깅 활성화(Enable audit logging)를 선택합니다.

1. 일반 설정(General Settings)에서 구성(Configure)을 선택하고 REST 계층(REST layer)이 활성화되었는지 확인합니다.
2. 규정 준수 설정(Compliance Settings)에서 구성(Configure)을 선택합니다.
3. 감시된 필드(Watched Fields)의 쓰기(Write)에서 모든 쓰기 이벤트에 대한 `accounts`을 이 인덱스에 추가합니다.

4. 감시된 필드(Watched Fields)의 읽기(Read)에서 `ssn` 인덱스의 `id-` 필드 및 `accounts`를 추가합니다.

```
{
  "accounts-": [
    "ssn",
    "id-"
  ]
}
```

2단계: 읽기 및 쓰기 이벤트 수행

1. OpenSearch 대시보드로 이동하여 개발자 도구(Dev Tools)를 선택하고 샘플 문서를 인덱싱합니다.

```
PUT accounts/_doc/0
{
  "ssn": "123",
  "id-": "456"
}
```

2. 읽기 이벤트를 테스트하려면 다음 요청을 보냅니다.

```
GET accounts/_search
{
  "query": {
    "match_all": {}
  }
}
```

3단계: 로그 관찰

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹(Log groups)을 선택합니다.
3. 감사 로그를 활성화하는 동안 지정한 로그 그룹을 선택합니다. 로그 그룹 내에서 OpenSearch Service가 도메인의 각 노드에 대한 로그 스트림을 생성합니다.
4. 로그 스트림(Log streams)에서 모두 검색(Search all)을 선택합니다.
5. 읽기 및 쓰기 이벤트는 해당 로그를 참조하세요. 로그가 나타나기 전 예상 지연 시간은 5초입니다.

샘플 쓰기 감사 로그


```
{
  "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"
  ]
}
```


샘플 읽기 감사 로그


```
{
  "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"
  ]
}
```

요청 본문을 포함하려면 OpenSearch Dashboards의 규정 준수 설정(Compliance settings)으로 돌아가 메타데이터 쓰기(Write metadata)를 비활성화합니다. 특정 사용자별로 이벤트를 제외하려면 해당 사용자를 무시된 사용자(Ignored Users)에 추가합니다.

각 감사 로그 필드에 대한 설명은 감사 로그 필드 참조를 참조하세요. 감사 로그 데이터 검색 및 분석에 대한 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs Insights를 사용한 로그 데이터 분석을 참조하세요.

REST API를 사용하여 감사 로그 구성

OpenSearch Dashboards를 사용하여 감사 로그를 구성하는 것이 좋지만 세분화된 액세스 제어 REST API를 사용할 수도 있습니다. 이 섹션에는 샘플 요청이 포함되어 있습니다. REST API에 대한 전체 설명서는 OpenSearch 설명서에서 확인할 수 있습니다.

```
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"
    ]
  }
}
```

Amazon EventBridge를 사용하여 OpenSearch Service 이벤트 모니터링

Amazon OpenSearch Service는 Amazon EventBridge와 통합하여 도메인에 영향을 주는 특정 이벤트를 사용자에게 알립니다. AWS 서비스의 이벤트는 거의 실시간으로 EventBridge로 전달됩니다. 동일한 이벤트가 Amazon EventBridge 이전 버전인 Amazon CloudWatch Events에도 전송됩니다. 원하는

이벤트만 표시하도록 간단한 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 자동으로 트리거할 수 있는 태스크는 다음과 같습니다.

- AWS Lambda 함수 호출
- Amazon EC2 Run Command 호출
- Amazon Kinesis Data Streams로 이벤트 릴레이
- AWS Step Functions 상태 시스템 활성화
- SNS 주제 또는 Amazon SQS 대기열 알림

자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 시작하기를 참조하세요.

주제

- 서비스 소프트웨어 업데이트 이벤트
- 이벤트 자동 조정
- 클러스터 상태 이벤트
- VPC 엔드포인트 이벤트
- 노드 만료 이벤트
- 성능 저하된 노드 사용 중지 이벤트
- 도메인 오류 이벤트
- 자습서: Amazon OpenSearch Service EventBridge 이벤트 수신
- 자습서: 사용 가능한 소프트웨어 업데이트에 대한 Amazon SNS 알림 보내기

서비스 소프트웨어 업데이트 이벤트

OpenSearch Service는 다음 서비스 소프트웨어 업데이트 이벤트가 발생할 때 EventBridge에 이벤트를 보냅니다.

서비스 소프트웨어 업데이트 사용 가능

서비스 소프트웨어 업데이트를 사용할 수 있을 때 OpenSearch Service가 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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"
  }
}
```

서비스 소프트웨어 업데이트 예약 완료

서비스 소프트웨어 업데이트가 예약 완료되면 OpenSearch Service가 이 이벤트를 보냅니다. 선택적 업데이트의 경우 예약된 날짜에 알림을 받게 되며 언제든지 일정을 변경할 수 있습니다. 필수 업데이트의 경우 예정된 날짜보다 3일 전에 알림을 받게 되며 필수 기간 내에서 일정을 조정할 수 있습니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

서비스 소프트웨어 업데이트 일정 변경

선택적 서비스 소프트웨어 업데이트가 일정이 변경되면 OpenSearch Service가 이 이벤트를 보냅니다. 자세한 내용은 the section called “선택적 업데이트와 필수 업데이트 비교” 단원을 참조하십시오.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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."
  }
```

```
}
```

서비스 소프트웨어 업데이트 시작

서비스 소프트웨어 업데이트가 시작되면 OpenSearch Service가 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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.
  }
}
```

서비스 소프트웨어 업데이트 완료

서비스 소프트웨어 업데이트가 완료되면 OpenSearch Service가 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

서비스 소프트웨어 업데이트 취소

서비스 소프트웨어 업데이트가 예약 취소되면 OpenSearch Service가 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

예약된 서비스 소프트웨어 업데이트 취소

OpenSearch Service는 도메인에 대해 이전에 예약된 서비스 소프트웨어 업데이트가 취소되면 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

서비스 소프트웨어 업데이트 미실행

서비스 소프트웨어 업데이트가 시작할 수 없으면 OpenSearch Service가 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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]"
```

```
  }
}
```

서비스 소프트웨어 업데이트 실패

서비스 소프트웨어 업데이트가 실패하면 OpenSearch Service가 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

서비스 소프트웨어 업데이트 필요

서비스 소프트웨어 업데이트가 필요한 경우 OpenSearch Service가 이 이벤트를 보냅니다. 자세한 내용은 the section called “선택적 업데이트와 필수 업데이트 비교” 단원을 참조하십시오.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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"
  }
}
```

이벤트 자동 조정

OpenSearch Service는 다음 자동 조정 이벤트가 발생할 때 EventBridge에 이벤트를 보냅니다.

자동 조정 보류 중

OpenSearch Service는 자동 조정으로 클러스터 성능 및 가용성 향상을 위한 조정 권장 사항이 확인된 경우 이 이벤트를 보냅니다. 자동 조정이 비활성화된 도메인에 대해서만 이 이벤트가 표시됩니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

자동 조정 시작

자동 조정으로 도메인에 새 설정이 적용되기 시작하면 OpenSearch Service가 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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%}."
  }
}
```

자동 조정에 예약된 블루/그린(Blue/Green) 배포 필요

OpenSearch Service는 자동 조정으로 예약된 블루/그린(Blue/Green) 배포가 필요한 조정 권장 사항이 확인된 경우 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

자동 조정 취소

OpenSearch Service는 보류 중인 조정 권장 사항이 없기 때문에 자동 조정 일정이 취소된 경우 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

자동 조정 완료

OpenSearch Service는 자동 조정이 블루/그린(Blue/Green) 배포를 완료하고 클러스터가 새 JVM 설정으로 작동할 때 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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%}."
  }
}
```

자동 조정 비활성화 및 변경 사항 철회

OpenSearch Service는 자동 조정이 비활성화되고 적용된 변경 사항이 롤백된 경우 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

자동 조정 비활성화 및 변경 사항 유지

OpenSearch Service는 자동 조정이 비활성화되고 적용된 변경 사항이 유지된 경우 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

클러스터 상태 이벤트

OpenSearch Service는 클러스터의 상태가 손상되면 특정 이벤트를 EventBridge로 전송합니다.

빨간색 클러스터 복구 시작됨

OpenSearch Service는 클러스터 상태가 1시간 이상 지속적으로 빨간색으로 표시된 후 이 이벤트를 전송합니다. 클러스터 상태를 수정하기 위해 스냅샷에서 하나 이상의 빨간색 인덱스를 자동으로 복원하려고 시도합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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]"
   }
```

```
}
```

빨간색 클러스터 복구가 부분적으로 완료됨

OpenSearch Service는 빨간색 클러스터 상태를 수정하려고 시도하는 동안 스냅샷에서 빨간색 인덱스의 하위 집합만 복원할 수 있을 때 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

빨간색 클러스터 복구 실패함

OpenSearch Service는 빨간색 클러스터 상태를 수정하려고 시도하는 동안 인덱스를 복원하지 못할 때 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

삭제할 샤드

OpenSearch Service는 14일 이상 지속적으로 빨간색으로 표시된 빨간색 클러스터 상태를 자동으로 수정하려고 시도했지만 하나 이상의 인덱스가 빨간색으로 유지될 때 이 이벤트를 전송합니다. 7일 더 지난 후(총 21일 연속 빨간색으로 표시됨), OpenSearch Service는 모든 빨간색 인덱스에서 할당되지 않은 샤드 삭제를 진행합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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"
   }
}
```

샤드 삭제됨

OpenSearch Service는 클러스터 상태가 21일 이상 지속적으로 빨간색으로 표시된 후 이 이벤트를 전송합니다. 모든 빨간색 인덱스에서 할당되지 않은 샤드(스토리지 및 컴퓨팅)가 삭제됩니다. 자세한 내용은 the section called “빨간색 클러스터의 자동 수정” 섹션을 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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"
   }
}
```

샤드 수 높음 경고

OpenSearch Service는 핫 데이터 노드의 평균 샤드 수가 권장 기본 제한인 1,000개의 90%를 초과하면 이 이벤트를 전송합니다. 최신 버전의 Elasticsearch 및 OpenSearch는 노드 제한당 구성 가능한 최대 샤드 수를 지원하지만, 노드당 샤드 수를 1,000개 이하로 두는 것이 좋습니다. 샤드 수 선택을 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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."
```

```
  }
}
```

샤드 수 제한 초과됨

OpenSearch Service는 핫 데이터 노드의 평균 샤드 수가 권장 기본 제한인 1,000개를 초과하면 이 이벤트를 전송합니다. 최신 버전의 Elasticsearch 및 OpenSearch는 노드 제한당 구성 가능한 최대 샤드 수를 지원하지만, 노드당 샤드 수를 1,000개 이하로 두는 것이 좋습니다. 샤드 수 선택을 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

디스크 공간 부족

OpenSearch Service는 클러스터에 있는 하나 이상의 노드가 사용 가능한 스토리지 공간의 25% 미만 또는 25GB 미만인 경우 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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"
  }
}
```

낮은 디스크 워터마크 위반

OpenSearch Service는 클러스터의 모든 노드가 사용 가능한 스토리지 공간의 10% 미만 또는 10GB 미만일 때 이 이벤트를 전송합니다. 모든 노드가 로우 디스크 워터마크를 위반할 경우 새 인덱스는 노란색 클러스터로 표시되고 모든 노드가 하이 디스크 워터마크 아래로 떨어지면 빨간색 클러스터로 이어집니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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 버스트 밸런스 70% 미만

OpenSearch Service는 하나 이상의 데이터 노드에서 EBS 버스트 밸런스가 70% 미만으로 떨어질 때 이 이벤트를 전송합니다. EBS 버스트 밸런스 고갈로 인해 광범위한 클러스터 가용성 및 I/O 요청 제한이 발생할 수 있으며, 이로 인해 인덱싱 및 검색 요청에 대한 지연과 시간 초과가 발생할 수 있습니다. 이 문제를 해결하는 단계는 the section called “낮은 EBS 버스트 밸런스” 섹션을 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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 버스트 밸런스 20% 미만

OpenSearch Service는 하나 이상의 데이터 노드에서 EBS 버스트 밸런스가 20% 미만으로 떨어질 때 이 이벤트를 전송합니다. EBS 버스트 밸런스 고갈로 인해 광범위한 클러스터 가용성 및 I/O 요청 제한이 발생할 수 있으며, 이로 인해 인덱싱 및 검색 요청에 대한 지연과 시간 초과가 발생할 수 있습니다. 이 문제를 해결하는 단계는 the section called “낮은 EBS 버스트 밸런스” 섹션을 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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.
  }
}
```

디스크 처리량(Throughput) 제한

OpenSearch Service는 EBS 볼륨 또는 EC2 인스턴스의 처리량 제한으로 인해 도메인에 대한 읽기 및 쓰기 요청이 제한될 때 이 이벤트를 전송합니다. 이 알림을 받는 경우 AWS 권장 모범 사례에 따라 볼륨 또는 인스턴스를 확장하는 것이 좋습니다. 볼륨 유형이 gp2(와)과 같으면 볼륨 크기를 늘리세요. 볼륨 유형이 gp3(와)과 같으면 처리량을 더 많이 프로비저닝하세요. 또한 인스턴스 기본 및 최대 EBS 처리량이 프로비저닝된 볼륨 처리량보다 크거나 같은지 확인하고 그에 따라 확장할 수 있습니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

대형 샤드 크기

OpenSearch Service는 클러스터에 있는 하나 이상의 샤드가 50GiB 또는 65GiB를 초과하면 이 이벤트를 전송합니다. 최적의 클러스터 성능과 안정성을 보장하려면 샤드 크기를 줄입니다.

자세한 내용은 샤딩 모범 사례를 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

높은 JVM 사용량

OpenSearch Service는 도메인에 대한 JVMMemoryPressure 지표가 80%를 초과하면 이 이벤트를 전송합니다. 30분 동안 92%를 초과하면 클러스터에 대한 모든 쓰기 작업이 차단됩니다. 클러스터 안정성을 최적화하려면 클러스터로 향하는 트래픽을 줄이거나 도메인을 확장하여 워크로드에 충분한 메모리를 제공하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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 부족

OpenSearch Service는 최대 JVM이 70%를 초과하고 최대값과 최소값 간의 차이가 30% 미만일 때 이 이벤트를 전송합니다. 이는 워크로드의 가비지 수집 주기 동안 JVM이 충분한 메모리를 확보하지 못했음을 의미할 수 있습니다. 이로 인해 응답 속도가 점점 느려지고 지연 시간이 길어질 수 있으며, 상태 확인 시간 초과로 인해 노드가 중단되는 경우도 있습니다. 클러스터 안정성을 최적화하려면 클러스터로 향하는 트래픽을 줄이거나 도메인을 확장하여 워크로드에 충분한 메모리를 제공하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

사용자 지정 인덱스 라우팅 경고

OpenSearch Service는 도메인이 처리 중이고 블루/그린 배포가 중단될 수 있는 사용자 지정 index.routing.allocation 설정이 있는 인덱스가 포함되어 있을 때 이 이벤트를 전송합니다. 설정이 제대로 적용되었는지 확인하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

샤드 잠금 실패

OpenSearch Service는 [ShardLockObtainFailedException]로 샤드가 할당되지 않아 도메인이 비정상인 경우 이 이벤트를 전송합니다. 자세한 내용은 Amazon OpenSearch Service의 인메모리 샤드 잠금 예외를 해결하려면 어떻게 해야 합니까?를 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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 엔드포인트 이벤트

OpenSearch Service는 AWS PrivateLink 인터페이스 엔드포인트와 관련된 특정 이벤트를 EventBridge로 전송합니다.

VPC 엔드포인트 생성 실패

OpenSearch Service는 요청된 VPC 엔드포인트를 생성할 수 없을 때 이 이벤트를 전송합니다. 이 오류는 리전 내에서 허용되는 VPC 엔드포인트 수 제한에 도달했기 때문에 발생했을 수 있습니다. 지정된 서브넷이나 보안 그룹이 없는 경우에도 이 오류가 표시됩니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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 엔드포인트 업데이트 실패

OpenSearch Service는 요청된 VPC 엔드포인트를 삭제할 수 없을 때 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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 엔드포인트 삭제 실패

OpenSearch Service는 요청된 VPC 엔드포인트를 삭제할 수 없을 때 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

노드 만료 이벤트

OpenSearch Service는 다음 노드 만료 이벤트 중 하나가 발생할 때 EventBridge에 이벤트를 보냅니다.

노드 만료 예정

OpenSearch Service는 노드 만료 날짜가 예약되면 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

노드 만료 완료

OpenSearch Service는 노드 만료가 완료된 경우 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

노드 만료 실패

OpenSearch Service는 노드 만료가 실패할 경우 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

성능 저하된 노드 사용 중지 이벤트

OpenSearch Service는 노드의 하드웨어 성능 저하로 인해 노드 교체가 필요한 경우 이러한 이벤트를 전송합니다.

성능 저하된 노드 사용 중지 알림

OpenSearch Service는 도메인에 성능이 저하된 노드를 사용 중지하고 교체하는 자동 작업이 예약된 경우 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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"
   }
}
```

성능 저하된 노드 사용 중지 완료

OpenSearch Service는 성능이 저하된 노드가 사용 중지되고 새 노드로 교체된 경우 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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"
   }
}
```

성능 저하된 노드 사용 중지 실패

OpenSearch Service는 노드 만료가 실패할 경우 이 이벤트를 전송합니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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"
   }
}
```

도메인 오류 이벤트

OpenSearch Service는 다음 도메인 오류가 발생할 때 EventBridge에 이벤트를 보냅니다.

도메인 업데이트 검증 실패

OpenSearch Service는 도메인에 대한 구성 변경을 업데이트하거나 수행하려고 할 때 하나 이상의 검증 실패가 발생하면 이 이벤트를 전송합니다. 이러한 실패를 해결하는 단계는 the section called “Troubleshooting validation errors(검증 오류 문제 해결 중)” 섹션을 참조하세요.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
   "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"
   }
}
```

KMS 키에 액세스할 수 없음

OpenSearch Service는 AWS KMS 키에 액세스할 수 없을 때 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

```
{
  "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."
```

```
  }
}
```

도메인 격리

OpenSearch Service는 도메인이 격리되어 네트워크에서 연결할 수 없어 요청을 받거나 읽거나 쓸 수 없을 때 이 이벤트를 보냅니다.

예제

다음은 이러한 유형의 이벤트 예입니다.

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

자습서: Amazon OpenSearch Service EventBridge 이벤트 수신

이 자습서에서는 Amazon OpenSearch Service 이벤트를 수신 대기하고 CloudWatch Logs 로그 스트림에 쓰는 간단한 AWS Lambda 함수를 설정합니다.

사전 조건

이 자습서에서는 사용자가 기존 OpenSearch Service 도메인을 가지고 있다고 가정합니다. 도메인을 생성하지 않았으면 도메인 생성 및 관리에 있는 단계에 따라 도메인을 생성합니다.

1단계: Lambda 함수 생성

이 절차에서는 OpenSearch Service 이벤트 메시지의 대상으로 사용할 간단한 Lambda 함수를 생성합니다.

대상 Lambda 함수를 생성하려면

1. https://console.aws.amazon.com/lambda/ AWS Lambda 콘솔을 엽니다.
2. 함수 생성(Create function)과 새로 작성(Author from scratch)을 차례로 선택합니다.
3. 함수 이름(Function name)에 event-handler를 입력합니다.
4. 런타임에서 Python 3.8을 선택합니다.
5. 함수 생성(Create function)을 선택합니다.
6. 함수 코드(Function code) 섹션에서 다음 예제와 일치하도록 샘플 코드를 수정합니다.

```
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))
```

   다음은 OpenSearch Service에서 전송하는 이벤트를 인쇄하는 간단한 Python 3.8 함수입니다. 모든 설정이 올바르게 구성되면 이 자습서가 끝날 때 이 Lambda 함수와 연결된 CloudWatch Logs 로그 스트림에 이벤트 세부 정보가 표시됩니다.
7. 배포(Deploy)를 선택합니다.

2단계: 이벤트 규칙 등록

이 단계에서는 OpenSearch Service 도메인에서 이벤트를 캡처하는 EventBridge 규칙을 생성합니다. 이 규칙은 규칙이 정의된 계정 내의 모든 이벤트를 캡처합니다. 이벤트 메시지 자체에 작업이 시작된 도메인을 포함하여 이벤트 소스에 대한 정보가 포함됩니다. 이 정보를 사용하여 프로그래밍 방식으로 이벤트를 필터링하고 정렬할 수 있습니다.

EventBridge 규칙을 생성하려면

1. https://console.aws.amazon.com/events/에서 EventBridge 콘솔을 엽니다.

2. 규칙 생성(Create rule)을 선택합니다.
3. 규칙 이름을 event-rule로 지정합니다.
4. 다음(Next)을 선택합니다.
5. 이벤트 패턴에서 AWS services, Amazon OpenSearch Service, All Events(모든 이벤트)를 선택합니다. 이 패턴은 모든 OpenSearch Service 도메인과 모든 OpenSearch Service 이벤트에 적용됩니다. 또는 더 한정적인 패턴을 만들어 일부 결과를 필터링할 수 있습니다.
6. 다음(Next)을 누릅니다.
7. 대상에서 Lambda 함수(Lambda function)를 선택합니다. 함수 드롭다운에서 event-handler를 선택합니다.
8. 다음(Next)을 누릅니다.
9. 태그를 건너뛰고 다음(Next)을 다시 누릅니다.
10. 구성을 살펴본 후 규칙 생성(Create rule)을 선택합니다.

3단계: 구성 테스트

다음에 OpenSearch Service 콘솔의 알림 섹션에서 알림을 받을 때, 모두 제대로 구성된 경우 Lambda 함수가 트리거되고 해당 함수에 대한 CloudWatch Logs 로그 스트림에 이벤트 데이터를 기록합니다.

구성을 테스트하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그(Logs)를 선택하고 Lambda 함수의 로그 그룹을 선택합니다(예: /aws/lambda/event-handler).
3. 이벤트 데이터를 보려면 로그 스트림을 선택합니다.

자습서: 사용 가능한 소프트웨어 업데이트에 대한 Amazon SNS 알림 보내기

이 자습서에서는 Amazon OpenSearch Service에서 사용 가능한 서비스 소프트웨어 업데이트에 대한 알림을 캡처하고 Amazon Simple Notification Service (Amazon SNS)를 통해 이메일 알림을 보내는 Amazon EventBridge 이벤트 규칙을 구성합니다.

사전 조건

이 자습서에서는 사용자가 기존 OpenSearch Service 도메인을 가지고 있다고 가정합니다. 도메인을 생성하지 않았으면 도메인 생성 및 관리에 있는 단계에 따라 도메인을 생성합니다.

1단계: Amazon SNS 주제 생성 및 구독

새 이벤트 규칙의 이벤트 대상으로 사용할 Amazon SNS 주제를 구성합니다.

Amazon SNS 대상을 생성하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 주제(Topics), 주제 생성(Create topic)을 차례로 선택합니다.
3. 작업 유형에 대해 표준(Standard)을 선택하고 작업 이름을software-update로 지정합니다.
4. 주제 생성(Create topic)을 선택합니다.
5. 주제를 생성한 후 구독 생성(Create subscription)을 선택합니다.
6. 프로토콜(Protocol)에서 이메일(Email)을 선택합니다. 엔드포인트(Endpoint)에 현재 액세스 권한이 있는 이메일 주소를 입력하고 구독 생성(Create subscription)을 선택합니다.
7. 이메일 계정을 확인하고 구독 확인 이메일 메시지를 기다립니다. 메시지를 수신하면 구독 확인(Confirm subscription)을 선택합니다.

2단계: 이벤트 규칙 등록

다음으로 서비스 소프트웨어 업데이트 이벤트만 캡처하는 이벤트 규칙을 등록합니다.

이벤트 규칙 생성

1. https://console.aws.amazon.com/events/에서 EventBridge 콘솔을 엽니다.
2. 규칙 생성(Create rule)을 선택합니다.
3. 규칙 이름을 softwareupdate-rule로 지정합니다.
4. 다음을 선택합니다.
5. 이벤트 패턴에서 AWS 서비스, Amazon OpenSearch Service(관리형 클러스터) 및 Amazon OpenSearch Service 소프트웨어 업데이트 알림을 선택합니다. 이 패턴은 OpenSearch Service의 모든 서비스 소프트웨어 업데이트 이벤트와 일치합니다. 이벤트 패턴에 대한 자세한 내용은 Amazon EventBridge 사용 설명서에서 Amazon EventBridge 이벤트 패턴을 참조하세요.
6. 또는 특정 심각도로만 필터링할 수 있습니다. 각 이벤트의 심각도는 the section called “서비스 소프트웨어 업데이트 이벤트” 섹션을 참조하세요.
7. 다음을 선택합니다.
8. 대상에 SNS 주제(SNS topic)를 선택하고 software-update를 선택합니다.
9. 다음을 선택합니다.

10. 태그를 건너뛰고 다음(Next)을 선택합니다.
11. 규칙 구성을 살펴본 후 규칙 생성(Create rule)을 선택합니다.

다음에 사용 가능한 서비스 소프트웨어 업데이트에 대한 OpenSearch Service에서 알림을 받을 때 모든 것이 제대로 구성되어 있으면 Amazon SNS에서 업데이트에 대한 이메일 알림을 보내야 합니다.

AWS CloudTrail을 사용한 Amazon OpenSearch Service API 호출 모니터링

Amazon OpenSearch Service는 OpenSearch Service에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail 은 OpenSearch Service에 대한 모든 구성 API 호출을 이벤트로 캡처합니다.

> **Note**
>
> CloudTrail은 구성 API(예: `CreateDomain` 및 `GetUpgradeStatus`) 호출만을 캡처합니다. CloudTrail은 OpenSearch API(예: `_search` 및 `_bulk`) 호출을 캡처하지 않습니다. 이러한 호출에 대한 내용은 the section called “감사 로그 모니터링” 섹션을 참조하세요.

캡처된 호출에는 OpenSearch Service 콘솔 AWS CLI또는 AWS SDK의 호출이 포함됩니다. 추적을 생성하면 OpenSearch Service에 대한 이벤트를 포함한 CloudTrail 이벤트를 지속해서 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록(Event history)에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 OpenSearch Service에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail 의 Amazon OpenSearch Service 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. OpenSearch Service에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

OpenSearch Service에 대한 이벤트를 포함하여 AWS 계정 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습

니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 에 대한 추적 생성 AWS 계정
- AWS CloudTrail Logs와의 서비스 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 수신 및 여러 계정에서 CloudTrail 로그 파일 수신

모든 OpenSearch Service 구성 API 작업은 CloudTrail에서 로깅되며 Amazon OpenSearch Service API 참조에 문서화되어 있습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부
- 역할 또는 페더레이션 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부

자세한 설명은 CloudTrail userIdentity 요소를 참조하세요.

Amazon OpenSearch Service 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 `CreateDomain` 작업을 보여주는 CloudTrail 로그 항목입니다.

```
{
  "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"
}
```

Amazon OpenSearch Service의 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 클라우드에서 AWS AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사원은 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다. Amazon OpenSearch Service에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 AWS 규정 준수 프로그램 제공 범위 내 서비스를 참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 OpenSearch Service 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 OpenSearch Service를 구성하는 방법을 보여줍니다. 또한 OpenSearch Service 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- Amazon OpenSearch Service의 데이터 보호
- Amazon OpenSearch Service의 Identity and Access Management
- 교차 서비스 혼동된 대리자 방지
- Amazon OpenSearch Service에서 세분화된 액세스 제어
- Amazon OpenSearch Service에 대한 규정 준수 확인
- Amazon OpenSearch Service의 복원성
- Amazon OpenSearch Service에 대한 JWT 인증 및 권한 부여
- Amazon OpenSearch Service의 인프라 보안
- OpenSearch 대시보드에 대한 SAML 인증
- OpenSearch에 대한 IAM Identity Center 신뢰할 수 있는 자격 증명 전파 지원

- OpenSearch Dashboards에 대한 Amazon Cognito 인증 구성
- Amazon OpenSearch Service에 대한 서비스 연결 역할 사용

Amazon OpenSearch Service의 데이터 보호

AWS 공동 책임 모델 Amazon OpenSearch Service의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 OpenSearch Service 또는 기타 AWS 서비스 에서 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

Amazon OpenSearch Service의 저장된 데이터 암호화

OpenSearch Service 도메인은 저장된 데이터 암호화, 데이터 무단 액세스를 방지하는 보안 기능을 제공합니다. 이 기능은 AWS Key Management Service (AWS KMS)를 사용하여 암호화 키를 저장하고 관리하며 고급 암호화 표준 알고리즘을 256비트 키(AES-256)와 함께 사용하여 암호화를 수행합니다. 활성화된 경우 이 기능은 다음과 같은 도메인 측면을 암호화합니다.

- 모든 인덱스(UltraWarm 스토리지의 인덱스 포함)
- OpenSearch 로그
- 전환 파일
- 애플리케이션 디렉터리의 모든 기타 데이터
- 자동 스냅샷

저장된 데이터 암호화를 활성화할 때 다음은 암호화되지 않지만 추가 단계를 수행하여 보호할 수 있습니다.

- 수동 스냅샷: 현재 AWS KMS 키를 사용하여 수동 스냅샷을 암호화할 수 없습니다. 그러나 스냅샷 리포지토리로 사용하는 버킷을 암호화하기 위해 S3 관리형 키로 서버 측 암호화를 사용할 수 있습니다. 지침은 the section called “수동 스냅샷 리포지토리 등록” 섹션을 참조하세요.
- 느린 로그 및 오류 로그: 로그를 게시하고 암호화하려는 경우 OpenSearch Service 도메인과 동일한 AWS KMS 키를 사용하여 CloudWatch Logs 로그 그룹을 암호화할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 AWS Key Management Service를 사용하여 CloudWatch Logs에서 로그 데이터 암호화를 참조하세요.

> **Note**
>
> 도메인에서 UltraWarm 또는 콜드 스토리지를 활성화한 경우 기존 도메인에서 저장된 암호화를 활성화할 수 없습니다. 먼저 UltraWarm 또는 콜드 스토리지를 비활성화하고 저장된 암호화를 활성화한 다음 UltraWarm 또는 콜드 스토리지를 다시 활성화해야 합니다. UltraWarm 또는 콜드 스토리지에 인덱스를 보존하려면 UltraWarm 또는 콜드 스토리지를 비활성화하기 전에 핫 스토리지로 인덱스를 이동해야 합니다.

OpenSearch Service는 비대칭이 아닌 대칭 암호화 KMS 키만 지원합니다. 대칭 키를 생성하는 방법을 알아보려면 AWS Key Management Service 개발자 안내서의 KMS 키 생성을 참조하세요.

유휴 상태의 암호화가 사용되는지와 관계없이 AES-256 및 OpenSearch Service 관리형 키를 사용하여 모든 도메인이 자동으로 사용자 지정 패키지를 암호화합니다.

권한

OpenSearch Service 콘솔을 사용하여 저장 데이터 암호화를 구성하려면 다음 자격 증명 기반 정책과 AWS KMS같은에 대한 읽기 권한이 있어야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS 소유 키 이외의 키를 사용하려면 키에 대한 권한 부여를 생성할 권한도 있어야 합니다. 이러한 권한은 보통 키를 만들 때 지정하는 리소스 기반 정책의 형식입니다.

키를 OpenSearch Service에만 적용하려면 해당 키 정책에 kms:ViaService 조건을 추가할 수 있습니다.

```
"Condition": {
  "StringEquals": {
    "kms:ViaService": "es.us-west-1.amazonaws.com"
  },
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
```

자세한 내용은AWS Key Management Service 개발자 안내서의 AWS KMS의 키 정책을 참조하세요.

저장된 데이터 암호화 활성화

새 도메인의 저장된 데이터 암호화에는 OpenSearch 또는 Elasticsearch 5.1 이상이 필요합니다. 기존 도메인에서 이 기능을 활성화하려면 OpenSearch 또는 Elasticsearch 6.7 이상이 필요합니다.

저장된 데이터의 암호화를 활성화하려면(콘솔)

1. AWS 콘솔에서 도메인을 열고 작업 및 보안 구성 편집을 선택합니다.
2. 암호화 아래에서 저장된 데이터 암호화 활성화를 선택하세요.
3. 사용할 AWS KMS 키를 선택한 다음 변경 사항 저장을 선택합니다.

구성 API를 통해 암호화를 활성화할 수도 있습니다. 다음 요청은 기존 도메인에 저장된 데이터의 암호화를 활성화합니다.

```
{
   "ClusterConfig":{
      "EncryptionAtRestOptions":{
         "Enabled": true,
         "KmsKeyId":"arn:aws:kms:us-east-1:123456789012:alias/my-key"
      }
   }
}
```

사용 중지 또는 삭제된 KMS 키

도메인을 암호화하는 데 사용한 키를 사용 중지하거나 삭제하면 도메인에 액세스할 수 없게 됩니다. OpenSearch Service에서 KMS 키에 액세스할 수 없다는 알림을 보냅니다. 도메인에 액세스하려면 즉시 키를 다시 사용 설정하세요.

OpenSearch Service 팀은 키가 삭제된 경우 데이터 복구를 지원할 수 없습니다.는 최소 7일의 대기 기간이 지난 후에만 키를 AWS KMS 삭제합니다. 키가 삭제 보류 중인 경우 삭제를 취소하거나 데이터 손실을 방지하기 위해 도메인의 수동 스냅샷을 생성합니다.

저장된 데이터 암호화 비활성화

저장된 데이터를 암호화하기 위해 도메인을 구성한 후 설정을 비활성화할 수 없습니다. 대신 기존 도메인의 수동 스냅샷을 가져와 다른 도메인을 생성하고, 데이터를 마이그레이션하며, 이전 도메인을 삭제할 수 있습니다.

저장된 데이터를 암호화하는 도메인 모니터링

저장된 데이터를 암호화하는 도메인에는 2개의 추가 지표 `KMSKeyError` 및 `KMSKeyInaccessible`이 있습니다. 이러한 지표는 도메인에 암호화 키 문제가 있을 때만 나타납니다. 이러한 지표에 대한 자세한 설명은 the section called “클러스터 지표” 섹션을 참조하세요. OpenSearch Service 콘솔 또는 Amazon CloudWatch 콘솔을 사용하여 볼 수 있습니다.

> **Tip**
>
> 각 지표는 도메인과 관련된 중요한 문제를 나타내므로 모두를 위한 CloudWatch 경보를 생성하는 것이 좋습니다. 자세한 내용은 the section called “권장되는 CloudWatch 경보” 섹션을 참조하세요.

기타 고려 사항

- 자동 키 교체는 AWS KMS 키의 속성을 보존하므로 교체는 OpenSearch 데이터에 액세스하는 기능에 영향을 주지 않습니다. 암호화된 OpenSearch Service 도메인은 새로운 키를 생성하거나 이전 키에 대한 모든 참조를 업데이트하는 수동 키 교체를 지원하지 않습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS 키 교체를 참조하세요.
- 특정 인스턴스 유형은 저장된 데이터의 암호화를 지원하지 않습니다. 자세한 내용은 the section called “지원되는 인스턴스 유형” 섹션을 참조하세요.
- 저장된 데이터를 암호화하는 도메인의 경우 자동 스냅샷을 위해 다른 리포지토리 이름을 사용합니다. 자세한 내용은 the section called “스냅샷 복원” 단원을 참조하십시오.
- 저장 시 암호화를 활성화하는 것이 좋지만 CPU 오버헤드와 지연 시간이 몇 밀리초만큼 추가될 수 있습니다. 그러나 대부분의 사용 사례는 이러한 차이에 민감하지 않으며, 클러스터, 클라이언트, 사용 프로필 구성에 따라 영향을 미치는 정도는 달라집니다.

Amazon OpenSearch Service를 위한 노드 간 암호화

노드 간 암호화는 Amazon OpenSearch Service의 기본 기능에 추가적인 보안 계층을 제공합니다.

도메인에서 VPC 액세스를 사용하는지와 관계없이 각 OpenSearch Service 도메인은 자체 전용 VPC 내에 있습니다. 이 아키텍처는 OpenSearch 노드 간 트래픽을 가로채는 잠재적 공격자를 방지하고 클러스터를 안전하게 유지합니다. 하지만 기본적으로 VPC 내에서는 암호화되지 않습니다. 노드 간 암호화를 사용하면 VPC 내 모든 통신에 대해 TLS 1.2 암호화를 사용할 수 있습니다.

HTTPS를 통해 OpenSearch Service로 데이터를 전송하는 경우 노드 간 암호화는 OpenSearch Service가 클러스터 전체에 데이터를 분배(및 재분배)할 때 데이터를 암호화된 상태로 유지하는 데 도움이 됩니다. 데이터가 암호화되지 않은 상태로 HTTP를 통해 도달할 경우, OpenSearch Service가 클러스터에 도달한 데이터를 암호화합니다. 콘솔 AWS CLI또는 구성 API를 사용하여 도메인에 대한 모든 트래픽이 HTTPS를 통해 도착하도록 요구할 수 있습니다.

세분화된 액세스 제어를 활성화하려면 노드 간 암호화가 필수입니다.

노드 간 암호화 활성화

새 도메인의 노드 간 암호화에는 OpenSearch의 모든 버전 또는 Elasticsearch 6.0 이상이 필요합니다. 기존 도메인의 노드 간 암호화를 활성화하려면 OpenSearch의 모든 버전 또는 Elasticsearch 6.7 이상이 필요합니다. AWS 콘솔에서 기존 도메인을 선택하고 [작업(Actions)], [보안 구성 편집(Edit security configuration)]을 선택합니다.

또는 AWS CLI 또는 구성 API를 사용할 수 있습니다. 자세한 내용은 AWS CLI 명령 참조 및 OpenSearch Service API 참조를 확인하세요.

노드 간 암호화 비활성화

노드 간 암호화를 사용하도록 도메인을 구성한 후 이 설정을 비활성화할 수 없습니다. 대신 암호화된 도메인의 수동 스냅샷을 가져와 다른 도메인을 생성하고, 데이터를 마이그레이션한 후 이전 도메인을 삭제할 수 있습니다.

Amazon OpenSearch Service의 Identity and Access Management

Amazon OpenSearch Service는 도메인에 대한 액세스를 제어하는 여러 가지 방법을 제공합니다. 이 주제에서는 다양한 정책 유형과 정책 유형 사이의 상호 작용 방식, 그리고 사용자 지정 정책을 생성하는 방법에 대해서 살펴보겠습니다.

> **Important**
>
> VPC 지원으로 OpenSearch Service 액세스 제어에 대해 몇 가지 고려해야 할 사항이 추가되었습니다. 자세한 내용은 the section called “VPC 도메인 액세스 정책에 대하여” 섹션을 참조하세요.

정책 유형

OpenSearch Service는 다음 세 가지 유형의 액세스 정책을 지원합니다.

- the section called “리소스 기반 정책”
- the section called “ID 기반 정책”
- the section called “IP 기반 정책”

리소스 기반 정책

도메인을 생성할 때 도메인 액세스 정책이라고도 하는 리소스 기반 정책을 추가합니다. 이 정책은 보안 주체가 도메인의 하위 리소스에서 실행할 수 있는 작업을 지정합니다(클러스터 간 검색 제외). 하위 리소스에는 OpenSearch 인덱스 및 API가 있습니다. IAM의 보안 주체 JSON 정책 요소는 액세스가 허용된 계정, 사용자 또는 역할을 지정합니다. 리소스 JSON 정책 요소는 이러한 보안 주체가 액세스할 수 있는 하위 리소스를 지정합니다.

예를 들어 다음 리소스 기반 정책은 `test-user`에게 `test-domain`의 하위 리소스에 대한 모든 액세스 권한(`es:*`)을 부여합니다.

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/*"
    }
  ]
}
```

이 정책에는 다음과 같이 두 가지 중요한 고려 사항이 있습니다.

- 이 권한은 해당 도메인에만 적용됩니다. 다른 도메인에서 유사한 정책을 만들지 않는 한 `test-user`만 `test-domain`에 액세스할 수 있습니다.

- Resource에서 /* 요소의 추적은 중요하며 리소스 기반 정책은 도메인 자체가 아닌 도메인의 하위 리소스에만 적용됨을 나타냅니다. 리소스 기반 정책에서 es:* 작업은 es:ESHttp*와 동일합니다.

  예를 들어 test-user는 인덱스(GET https://search-test-domain.us-west-1.es.amazonaws.com/test-index)에 대해 요청할 수 있지만 도메인의 구성(POST https://es.us-west-1.amazonaws.com/2021-01-01/opensearch/domain/test-domain/config)을 업데이트하지는 못합니다. 두 엔드포인트의 차이점을 참고하세요. 구성 API에 액세스하기 위해서는 자격 증명 기반 정책이 필요합니다.

와일드카드를 추가하여 부분 인덱스 이름을 지정할 수 있습니다. 이 예시는 commerce(으)로 시작하는 모든 인덱스를 식별합니다.

```
arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce*
```

이 경우 이 와일드카드를 사용하여 지정하면 test-user이(가) commerce(으)로 이름이 시작되는 test-domain 내의 인덱스에 요청할 수 있음을 의미합니다.

test-user의 액세스 권한을 추가로 제한할 때는 다음과 같이 정책을 적용할 수 있습니다.

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"
    }
  ]
}
```

이제 test-user은(는) 한 가지 작업, 즉 commerce-data 인덱스에 대한 비교 검색만 실행할 수 있습니다. 그 밖에 도메인에 속한 모든 인덱스에는 액세스할 수 없으며, es:ESHttpPut 또는 es:ESHttpPost 작업 사용 권한이 없기 때문에 test-user이(가) 문서를 추가하거나 수정하는 것은 제한됩니다.

다음으로 고급 사용자 역할을 구성할 수도 있습니다. 이 정책은 인덱스에 있는 모든 URI의 HTTP GET 및 PUT 메서드에 대한 액세스 권한을 power-user-role에 부여합니다.

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/*"
    }
  ]
}
```

도메인이 VPC에 있거나 세분화된 액세스 제어를 사용하는 경우 개방형 도메인 액세스 정책을 사용할 수 있습니다. 그렇지 않으면 도메인 액세스 정책에 보안 주체 또는 IP 주소별로 몇 가지 제한 사항이 포함되어야 합니다.

가능한 작업에 대한 자세한 내용은 the section called “정책 요소 참조” 섹션을 참조하세요. 데이터를 훨씬 더 세밀하게 제어하려면 세분화된 액세스 제어와 함께 개방형 도메인 액세스 정책을 사용합니다.

ID 기반 정책

각 OpenSearch Service 도메인의 일부인 리소스 기반 정책과 달리 AWS Identity and Access Management (IAM) 서비스를 사용하여 사용자 또는 역할에 자격 증명 기반 정책을 연결합니다. 자격 증명 기반 정책 역시 리소스 기반 정책과 마찬가지로 서비스에 대한 액세스 주체와 실행 가능한 작업, 그리고 해당되는 경우에 한해 작업을 실행할 수 있는 리소스까지 지정합니다.

반드시 그래야 하는 것은 아니지만 자격 증명 기반 정책은 더욱 포괄적으로 적용되는 경우가 많습니다. 대부분의 경우 사용자가 수행할 수 있는 구성 API 동작만 관리합니다. 또한 이 정책을 할당한 후에도 OpenSearch Service의 리소스 기반 정책(또는 세분화된 액세스 제어)을 사용하여 사용자에게 OpenSearch 인덱스 및 API에 대한 액세스 권한을 부여할 수 있습니다.

> **Note**
>
> AWS 관리형 `AmazonOpenSearchServiceReadOnlyAccess` 정책이 있는 사용자는 콘솔에서 클러스터 상태를 볼 수 없습니다. 이러한 사용자가 클러스터 상태 정보(및 기타 OpenSearch 데이터)를 볼 수 있게 하려면 액세스 정책에 `es:ESHttpGet` 작업을 추가하고 이를 해당하는 계정 또는 역할에 연결합니다.

자격 증명 기반 정책은 사용자 또는 역할(보안 주체)에 연결되기 때문에 JSON이 보안 주체를 따로 지정하지 않습니다. 다음 정책은 `Describe` 및 `List`로 시작하는 작업에 대한 액세스 권한을 부여합니다. 이러한 조합의 작업은 도메인 구성에 대한 읽기 전용 액세스 권한을 제공하지만, 도메인 자체에 저장된 데이터에 대한 액세스 권한은 제공하지 않습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:Describe*",
        "es:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

관리자는 OpenSearch Service 및 모든 도메인에 저장된 모든 데이터에 대한 전체 액세스 권한을 가지고 있을 수 있습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

자격 증명 기반 정책을 사용하면 태그를 사용하여 구성 API에 대한 액세스를 제어할 수 있습니다. 예를 들어 다음 정책은 도메인에 `team:devops` 태그가 있는 경우 연결된 보안 주체가 도메인 구성을 보고 업데이트할 수 있도록 허용합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:UpdateDomainConfig",
      "es:DescribeDomain",
      "es:DescribeDomainConfig"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/team": [
```

```
          "devops"
        ]
      }
    }
  }]
}
```

태그를 사용하여 OpenSearch API에 대한 액세스를 제어할 수도 있습니다. OpenSearch API에 대한 태그 기반 정책은 HTTP 메서드에만 적용됩니다. 예를 들어, 도메인에 `environment:production` 태그가 있을 경우 다음 정책은 연결된 보안 주체가 OpenSearch API에 GET 및 PUT 요청을 보낼 수 있도록 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:ESHttpGet",
      "es:ESHttpPut"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

OpenSearch API를 보다 세부적으로 제어하려면 세분화된 액세스 제어 사용을 고려하세요.

> **Note**
>
> 태그 기반 정책에 하나 이상의 OpenSearch API를 추가한 후에는 도메인에 변경 사항을 적용하기 위해 단일 태그 작업(예: 태그 추가, 제거 또는 수정)을 수행해야 합니다. 태그 기반 정책에 OpenSearch API 작업을 포함하려면 서비스 소프트웨어 R20211203 이상이어야 합니다.

OpenSearch Service는 OpenSearch API가 아닌 구성 API에 대한 `RequestTag` 및 `TagKeys` 전역 조건 키를 지원합니다. 이러한 조건은 `CreateDomain`, `AddTags`, `RemoveTags`와 같은 요청 내에 태그를 포함하는 API 직접 호출에만 적용됩니다. 다음 정책을 사용하면 연결된 보안 주체가 도메인을 생성할 수 있지만 요청에 `team:it` 태그를 포함하는 경우에만 해당합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "es:CreateDomain",
      "es:AddTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/team": [
          "it"
        ]
      }
    }
  }
}
```

액세스 제어를 위한 태그 사용 및 리소스 기반 정책과 자격 증명 기반 정책 간의 차이점에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통해 속성을 기반으로 권한 정의를 참조하세요.

IP 기반 정책

IP 기반 정책은 도메인에 대한 액세스를 하나 이상의 IP 주소 또는 CIDR 블록으로 제한합니다. 기술적으로 보았을 때 IP 기반 정책은 별개의 정책이 아닙니다. 대신 익명 보안 주체를 지정하고 특수 조건을 포함하는 리소스 기반 정책일 뿐입니다. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.

IP 기반 정책의 기본적인 이점은 OpenSearch Service 도메인에 대한 무서명 요청이 가능하기 때문에 curl 및 OpenSearch Dashboards 같은 클라이언트를 사용하거나 프록시 서버를 통해 도메인에 액세스할 수 있다는 점입니다. 자세한 내용은 the section called “프록시를 사용하여 Dashboards에서 OpenSearch Service에 액세스” 섹션을 참조하세요.

> **Note**
>
> 도메인에서 VPC 액세스를 활성화한 경우 IP 기반 정책은 구성할 수 없습니다. 대신 `security groups`를 사용하여 도메인에 액세스할 수 있는 IP 주소를 제어할 수 있습니다. 자세한 정보는 다음의 주제를 참조하세요.
>
> - the section called “VPC 도메인 액세스 정책에 대하여”
> - Amazon VPC 사용 설명서의 보안 그룹을 사용하여 AWS 리소스에 대한 트래픽 제어

다음은 지정된 IP 범위에서 시작되는 모든 HTTP 요청에 `test-domain`에 대한 액세스 권한을 부여하는 정책입니다.

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/*"
    }
  ]
}
```

도메인에 퍼블릭 엔드포인트가 있고 세분화된 액세스 제어가 사용되지 않는 경우 IAM 보안 주체와 IP 주소를 결합하는 것이 좋습니다. 이 정책은 요청이 지정된 IP 범위에서 시작된 경우에만 `test-user`에 HTTP 액세스 권한을 부여합니다.

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/*"
  }]
}
```

OpenSearch Service 요청 작성 및 서명

완전 개방형 리소스 기반 액세스 정책을 구성하는 경우에도 OpenSearch Service 구성 API에 대한 모든 요청에 서명해야 합니다. 정책에서 IAM 역할 또는 사용자를 지정하는 경우 OpenSearch APIs에 대한 요청도 AWS 서명 버전 4를 사용하여 서명해야 합니다. 서명 메서드는 API에 따라 다음과 같이 다릅니다.

- OpenSearch Service 구성 API를 호출할 때는 AWS SDK 중에서 한 가지를 사용하는 것이 바람직합니다. SDK는 프로세스를 대폭 간소화할 뿐만 아니라 사용자 자신의 요청을 생성한 후 서명을 추가할 때와 비교하여 시간을 크게 절감할 수 있습니다. 구성 API 엔드포인트는 다음 형식을 사용합니다.

  ```
  es.region.amazonaws.com/2021-01-01/
  ```

  예를 들어 다음 요청은 `movies` 도메인의 구성을 변경하지만, 사용자가 직접 서명해야 합니다(권장하지 않음).

  ```
  POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/movies/config
  {
    "ClusterConfig": {
      "InstanceType": "c5.xlarge.search"
    }
  }
  ```

  SDK 중 하나를 사용하면(예: Boto 3) SDK가 서명하여 자동으로 요청을 처리합니다.

  ```
  import boto3

  client = boto3.client(es)
  response = client.update_domain_config(
    DomainName='movies',
    ClusterConfig={
      'InstanceType': 'c5.xlarge.search'
    }
  )
  ```

  Java 코드 샘플의 경우 the section called “ AWS SDKs 사용” 섹션을 참조하세요.

- OpenSearch API를 호출할 때는 사용자 자신의 요청에 서명이 필요합니다. OpenSearch API는 다음 형식을 사용합니다.

```
domain-id.region.es.amazonaws.com
```

예를 들어 다음 요청은 movies 인덱스에서 thor를 검색합니다.

```
GET https://my-domain.us-east-1.es.amazonaws.com/movies/_search?q=thor
```

> **Note**
>
> 서명 버전 4로 서명된 HTTP POST 요청을 위해 URL로 전달된 파라미터는 무시됩니다.

정책 충돌 시

정책이 서로 일치하지 않거나 사용자를 명시적으로 지정하지 않으면 복잡한 문제가 발생합니다. IAM 사용 설명서의 IAM 작동 방식은 정책 평가 로직의 간결한 요약을 제공합니다.

- 기본적으로 모든 요청을 거부합니다.
- 명시적 허용은 이러한 기본 설정을 무시합니다.
- 명시적 거부는 모든 허용을 무시합니다.

예를 들어 리소스 기반 정책이 도메인 하위 리소스(OpenSearch 인덱스 또는 API)에 대한 액세스 권한을 부여하지만 자격 증명 기반 정책이 액세스를 거부할 경우에는 액세스가 거부됩니다. 자격 증명 기반 정책이 액세스를 권한을 부여하고 리소스 기반 정책이 액세스 필요 여부를 지정하지 않을 경우에는 액세스가 허용됩니다. 정책이 서로 엇갈렸을 때 도메인 하위 리소스에 대한 결과는 아래 요약 표를 참조하세요.

	리소스 기반 정책에서 허용됨	리소스 기반 정책에서 거부됨	리소스 기반 정책에서 허용 및 거부되지 않음
Allowed in identity-based policy	허용	Deny	Allow
Denied in identity-based policy	Deny	Deny	Deny

	리소스 기반 정책에서 허용됨	리소스 기반 정책에서 거부됨	리소스 기반 정책에서 허용 및 거부되지 않음
Neither allowed nor denied in identity-based policy	Allow	Deny	Deny

정책 요소 참조

OpenSearch Service는 `NotPrincipal`을 제외하고 IAM 정책 요소 참조에 포함된 대부분의 정책 요소를 지원합니다. 다음 표에는 가장 일반적인 요소가 나와 있습니다.

JSON 정책 요소	요약
`Version`	정책 언어의 현재 버전은 `2012-10-17` 입니다. 모든 액세스 정책이 이 값을 지정해야 합니다.
`Effect`	이 요소는 명령문이 지정한 작업에 대한 액세스를 허용 또는 거부 여부를 지정합니다. 유효한 값은 `Allow` 또는 `Deny`입니다.
`Principal`	이 요소는 리소스에 대한 액세스가 허용되거나 거부되고 여러 형식을 취할 수 있는 AWS 계정 또는 IAM 역할 또는 사용자를 지정합니다. • AWS 계정: `"Principal":{"AWS": ["123456789012"]}` 또는 `"Principal":{"AWS": ["arn:aws:iam::123456789012:root"]}` • IAM 사용자: `"Principal":{"AWS": ["arn:aws:iam::123456789012:user/test-user"]}` • IAM 역할: `"Principal":{"AWS": ["arn:aws:iam::123456789012:role/test-role"]}` **Important** * 와일드카드를 지정하면 도메인에 대한 익명 액세스가 가능하므로 IP 기반 조건을 추가하거나, VPC 지원을 사용하거나, 세분화된 액세스 제어를 활성화하지 않는 한 권장되지 않습니다. 또한 다음

JSON 정책 요소	요약
	정책을 주의 깊게 검토하여 광범위한 액세스를 허용하지 않는지 확인합니다. • 관련 AWS 보안 주체(예: IAM 역할)에 연결된 자격 증명 기반 정책 • 연결된 AWS 리소스에 연결된 리소스 기반 정책(예: AWS Key Management Service KMS 키)

JSON 정책 요소	요약
`Action`	OpenSearch Service는 OpenSearch HTTP 메서드에 대해 `ESHttp*` 작업을 사용합니다. 나머지 작업은 구성 API에 적용됩니다. 일부 `es:` 작업은 리소스 수준 권한을 지원합니다. 예를 들어 모든 도메인이 아닌 특정 도메인 1개만 삭제할 수 있는 권한을 사용자 1명에게 부여할 수 있습니다. 그 밖의 작업은 서비스 자체에만 적용됩니다. 예를 들어 `es:ListDomainNames` 는 단일 도메인으로 이해하지 않기 때문에 와일드카드가 필요합니다. 사용 가능한 모든 작업의 목록과 해당 작업이 도메인 하위 리소스(`test-domain/*`), 도메인 구성(`test-domain`) 또는 서비스(`*`)에만 적용되는지 여부는 서비스 인증 참조에서 Amazon OpenSearch Service의 작업, 리소스 및 조건 키를 참조하세요. 리소스 기반 정책은 리소스 수준 권한과 다릅니다. 리소스 기반 정책은 도메인에 연결되는 모든 JSON 정책입니다. 따라서 리소스 수준 권한에서는 작업을 특정 도메인이나 하위 리소스로 제한할 수 있습니다. 실제로 리소스 수준 권한을 리소스 또는 자격 증명 기반 정책의 옵션으로 볼 수도 있습니다. 무엇보다 이미 존재하는 도메인에 대한 생성 권한을 사용자에게 부여하는 이유를 알 수 없다는 점에서 `es:CreateDomain` 에 대한 리소스 수준 권한이 직관적이지 않은 것으로 보일 수 있지만, `"Resource": "arn:aws:es:us-west-1:987654321098:domain/my-team-name-*"` 같이 와일드카드를 사용하여 간단한 도메인 명명 체계를 적용할 수 있습니다. 물론 다음과 같이 리소스 요소의 제한을 줄여서 작업을 추가하는 것도 가능합니다. JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow",`

JSON 정책 요소	요약

	<pre>"Action": [
	  "es:ESHttpGet",
	  "es:DescribeDomain"
	],
	"Resource": "*"
	}
	]
	}</pre>
	작업과 리소스의 페어링에 대한 자세한 내용은 여기 표에서 Resource 요소를 참조하세요.

Condition	OpenSearch Service는 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키에서 설명하는 대부분의 조건을 지원합니다. 주목할 만한 예외에는 OpenSearch Service가 지원하지 않는 aws:PrincipalTag 키가 포함됩니다.
	
	IP 기반 정책을 구성할 때는 다음과 같이 IP 주소 또는 CIDR 블록을 조건으로 지정합니다.
	
	<pre>"Condition": {
	  "IpAddress": {
	  "aws:SourceIp": [
	  "192.0.2.0/32"
	  ]
	  }
	}</pre>
	the section called “ID 기반 정책”에서 언급 한 바와 같이 aws:ResourceTag , aws:RequestTag , 및 aws:TagKeys 조건 키는 OpenSearch API와 더불어 구성 API에 적용됩니다.

JSON 정책 요소	요약

`Resource`	OpenSearch Service는 다음과 같은 세 가지 기본적인 방법으로 `Resource` 요소를 사용합니다.
	
	• OpenSearch Service 자체에 적용되는 작업(`es:ListDomainNames` 등)이거나 모든 액세스를 허용할 때는 다음 구문을 사용합니다.
	
	`"Resource": "*"`
	
	• 도메인 구성과 관련된 작업(`es:DescribeDomain` 등)일 때는 다음 구문을 사용합니다.
	
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name "`
	
	• 도메인 하위 리소스에 적용되는 작업(`es:ESHttpGet` 등)일 때는 다음 구문을 사용합니다.
	
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name /*"`
	
	와일드카드를 사용할 필요는 없습니다. OpenSearch Service에서는 각 OpenSearch 인덱스 또는 API에 대해 다른 액세스 정책을 정의할 수 있습니다. 예를 들어 다음과 같이 사용자의 권한을 `test-index` 인덱스로 제한할 수도 있습니다.
	
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name /test-index"`
	
	`test-index` 에 대한 모든 액세스 권한 대신 검색 API만 사용하도록 정책을 제한할 수 있습니다.
	
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name /test-index/_search"`
	
	다음과 같이 개별 문서에 대한 액세스를 제어할 수도 있습니다.

JSON 정책 요소	요약
	`"Resource": "arn:aws:es:  region:aws-account-id:domain/domain-name /test-index/test-type/1"` 기본적으로 OpenSearch가 하위 리소스를 URI로 표현하는 경우에는 액세스 정책을 사용하여 액세스를 제어할 수 있습니다. 사용자가 액세스할 수 있는 리소스에 대한 한층 세부적인 제어가 필요한 경우 the section called “세분화된 액세스 제어” 섹션을 참조하세요. 리소스 수준 권한을 지원하는 작업에 대한 자세한 내용은 여기 표에서 Action 요소를 참조하세요.

고급 옵션 및 API 고려 사항

OpenSearch Service에는 몇 가지 고급 옵션이 있으며, 그 중 하나인 `rest.action.multi.allow_explicit_index` 옵션이 액세스 제어에 영향을 미칩니다. 기본 설정 값인 true일 때는 사용자가 일부 상황에서 하위 리소스 권한을 우회할 수 있습니다.

예를 들어 다음과 같은 리소스 기반 정책이 있다고 가정하겠습니다.

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/*"
    }
  ]
}
```

위 정책은 `test-user`에게 `test-index` 및 OpenSearch 대량 API에 대한 모든 액세스 권한을 부여합니다. 또한 `restricted-index`에 대한 `GET` 요청도 허용합니다.

예상할 수 있겠지만 다음 인덱싱 요청은 권한 오류로 인해 실패할 수 밖에 없습니다.

```
PUT https://search-test-domain.us-west-1.es.amazonaws.com/restricted-index/movie/1
{
  "title": "Your Name",
  "director": "Makoto Shinkai",
  "year": "2016"
}
```

인덱스 API와 달리 대량 API에서는 호출 한 번으로 다수의 문서를 생성하거나, 업데이트하거나, 삭제할 수 있습니다. 하지만 이러한 작업을 요청 URL이 아닌 요청 본문에 지정하는 경우가 많습니다. OpenSearch Service는 URL을 사용하여 도메인 하위 리소스에 대한 액세스를 제어하기 때문에 실제로 `test-user`는 대량 API를 사용하여 `restricted-index`를 변경할 수 있습니다. 사용자에게 인덱스에 대한 `POST` 권한이 없더라도 다음 요청은 성공합니다.

```
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" }
```

이러한 상황에서는 액세스 정책이 목적을 이루지 못합니다. 따라서 사용자가 액세스 제한을 우회하지 못하도록 `rest.action.multi.allow_explicit_index`를 false로 변경할 수 있습니다. 이 값이 false일 경우에는 요청 본문에서 인덱스 이름을 지정하는 대량, mget 및 msearch API 호출이 모두 중단됩니다. 다시 말해서 `_bulk` 호출은 더 이상 유효하지 않지만 `test-index/_bulk` 호출은 유효합니다. 이 두 번째 엔드포인트에 인덱스 이름이 포함되기 때문에 요청 본문에 이름을 지정할 필요가 없습니다.

OpenSearch Dashboards는 mget 및 msearch에 대한 의존도가 크기 때문에 위와 같은 변경 이후 정상적으로 실행될 가능성이 거의 없습니다. 이러한 문제를 부분적으로 해결하고 싶다면 `rest.action.multi.allow_explicit_index`를 true로 남겨두고 하나 이상의 mget 및 msearch API에 대한 일부 사용자 액세스를 거부하는 방법도 있습니다.

이 설정의 변경에 대한 자세한 내용은 the section called “고급 클러스터 설정” 섹션을 참조하세요.

마찬가지로 다음 리소스 기반 정책 역시 두 가지 미묘한 문제가 있습니다.

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/*"
    }
  ]
}
```

- 명시적인 거부에도 불구하고 `test-user`가 여전히 `GET https://search-test-domain.us-west-1.es.amazonaws.com/_all/_search`나 `GET https://search-test-domain.us-west-1.es.amazonaws.com/*/_search` 같은 호출을 통해 `restricted-index`의 문서에 액세스할 수 있습니다.
- `Resource` 요소가 `restricted-index/*`를 참조하기 때문에 `test-user`는 인덱스의 문서에 직접 액세스할 수 있는 권한이 없습니다. 하지만 사용자에게 전체 인덱스를 삭제할 권한은 있습니다. 이러한 액세스 및 삭제 문제를 방지하려면 정책에 `restricted-index*`를 지정해야 합니다.

광범위한 액세스 허용과 집약적 액세스 거부를 혼합하기보다 최소 권한 원칙을 따라 태스크에 필요한 권한만 부여하는 것이 가장 안전한 방법입니다. 개별 인덱스 또는 OpenSearch 작업에 대한 액세스 제어와 관련된 자세한 내용은 the section called “세분화된 액세스 제어”을(를) 참조하세요.

> **Important**
>
> 와일드카드(*)를 지정하면 도메인에 익명으로 액세스할 수 있습니다. 와일드카드를 사용하지 않는 것이 좋습니다. 또한 다음 정책을 주의 깊게 검토하여 광범위한 액세스를 허용하지 않는지 확인합니다.
>
> - 연결된 AWS 보안 주체에 연결된 자격 증명 기반 정책(예: IAM 역할)
> - 연결된 리소스에 연결된 AWS 리소스 기반 정책(예: AWS Key Management Service KMS 키)

액세스 정책 구성

- OpenSearch Service에서 리소스 및 IP 기반 정책을 생성하거나 수정하는 방법에 대한 자세한 내용은 the section called “액세스 정책 구성” 섹션을 참조하세요.
- IAM에서 자격 증명 기반 정책을 생성하거나 수정하는 방법에 대한 지침은 IAM 사용 설명서의 고객 관리형 정책을 사용하여 사용자 지정 IAM 권한 정의를 참조하세요.

추가 샘플 정책

이 장에는 많은 샘플 정책이 포함되어 있지만 AWS 액세스 제어는 예제를 통해 가장 잘 이해되는 복잡한 주제입니다. 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 기반 정책의 예제를 참조하세요.

Amazon OpenSearch Service에 대한 API 참조

액세스 제어를 설정하면 IAM 자격 증명에 연결할 수 있는 권한 정책(자격 증명 기반 정책)을 작성할 수 있습니다. 자세한 참조 정보는 서비스 권한 부여 참조의 다음 주제를 참조하세요.

- Amazon OpenSearch Service의 작업, 리소스 및 조건 키.
- Amazon OpenSearch Ingestion의 작업, 리소스 및 조건 키.

이 참조에는 IAM 정책에서 사용할 수 있는 API 작업에 대한 정보가 포함되어 있습니다. 여기에는 권한을 부여할 수 있는 AWS 리소스와 세분화된 액세스 제어를 위해 포함할 수 있는 조건 키도 포함됩니다.

정책의 `Action` 필드에서 작업을 지정하고, `Resource` 필드에서 리소스 값을 지정하고, `Condition` 필드에서 조건을 지정합니다. OpenSearch Service 작업을 지정하려면 `es:` 접두사 다음에 API 작업 이름을 사용합니다(예: `es:CreateDomain`). OpenSearch Ingestion 작업을 지정하려면 `osis:` 접두사 다음에 API 작업을 사용합니다(예: `osis:CreatePipeline`).

AWS Amazon OpenSearch Service에 대한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 줍니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하세요.

AmazonOpenSearchDirectQueryGlueCreateAccess

Amazon OpenSearch Service Direct Query Service에 `CreateDatabase`, `CreatePartitionCreateTable`및에 대한 액세스 권한을 부여합니다`BatchCreatePartition` AWS Glue API.

IAM 콘솔에서 AmazonOpenSearchIngestionReadOnlyAccess 정책을 찾을 수 있습니다.

AmazonOpenSearchServiceFullAccess

AWS 계정의 OpenSearch Service 구성 API 작업 및 리소스에 대한 전체 액세스 권한을 부여합니다.

IAM 콘솔에서 AmazonOpenSearchServiceFullAccess 정책을 찾을 수 있습니다.

AmazonOpenSearchServiceReadOnlyAccess

에 대한 모든 OpenSearch Service 리소스에 대한 읽기 전용 액세스 권한을 부여합니다 AWS 계정.

IAM 콘솔에서 AmazonOpenSearchServiceReadOnlyAccess 정책을 찾을 수 있습니다.

AmazonOpenSearchServiceRolePolicy

`AmazonOpenSearchServiceRolePolicy`를 IAM 엔티티에 연결할 수 없습니다. 이 정책은 OpenSearch Service가 계정 리소스에 액세스할 수 있도록 허용하는 서비스 연결 역할에 연결됩니다. 자세한 내용은 the section called “권한” 단원을 참조하십시오.

IAM 콘솔에서 AmazonOpenSearchServiceRolePolicy 정책을 찾을 수 있습니다.

AmazonOpenSearchServiceCognitoAccess

Cognito 인증 활성화에 필요한 최소의 Amazon Cognito 권한을 제공합니다.

IAM 콘솔에서 AmazonOpenSearchServiceCognitoAccess 정책을 찾을 수 있습니다.

AmazonOpenSearchIngestionServiceRolePolicy

`AmazonOpenSearchIngestionServiceRolePolicy`를 IAM 엔티티에 연결할 수 없습니다. 이 정책은 OpenSearch Ingestion에서 수집 파이프라인에 대한 VPC 액세스 권한을 부여하고, 태그를 생성하며, 수집 관련 CloudWatch 지표를 사용자 계정에 게시할 수 있도록 허용하는 서비스 연결 역할에 연결됩니다. 자세한 내용은 the section called “서비스 연결 역할 사용” 단원을 참조하십시오.

IAM 콘솔에서 AmazonOpenSearchIngestionServiceRolePolicy 정책을 찾을 수 있습니다.

OpenSearchIngestionSelfManagedVpcePolicy

`OpenSearchIngestionSelfManagedVpcePolicy`를 IAM 엔티티에 연결할 수 없습니다. 이 정책은 OpenSearch Ingestion에서 수집 파이프라인에 대한 자체 관리형 VPC 액세스 권한을 부여하고, 태그를 생성하며, 수집 관련 CloudWatch 지표를 사용자 계정에 게시할 수 있도록 허용하는 서비스 연결 역할에 연결됩니다. 자세한 내용은 the section called “서비스 연결 역할 사용” 단원을 참조하십시오.

IAM 콘솔에서 OpenSearchIngestionSelfManagedVpcePolicy 정책을 찾을 수 있습니다.

AmazonOpenSearchIngestionFullAccess

AWS 계정의 OpenSearch Ingestion API 작업 및 리소스에 대한 전체 액세스 권한을 부여합니다.

IAM 콘솔에서 AmazonOpenSearchIngestionFullAccess 정책을 찾을 수 있습니다.

AmazonOpenSearchIngestionReadOnlyAccess

AWS 계정의 모든 OpenSearch Ingestion 리소스에 대한 읽기 전용 액세스 권한을 부여합니다.

IAM 콘솔에서 AmazonOpenSearchIngestionReadOnlyAccess 정책을 찾을 수 있습니다.

AmazonOpenSearchServerlessServiceRolePolicy

OpenSearch Serverless 지표 데이터를 CloudWatch로 전송하는 데 필요한 최소 Amazon CloudWatch 권한을 제공합니다.

IAM 콘솔에서 AmazonOpenSearchServiceRolePolicy 정책을 찾을 수 있습니다.

AWS 관리형 정책에 대한 OpenSearch Service 업데이트

이 서비스가 변경 내용 추적을 시작한 이후부터 OpenSearch Service의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다.

변경 사항	설명	날짜
를 업데이트했습니다. `AmazonOpenSearchServiceRolePolicy`	정책에 다음 문을 추가했습니다. Amazon OpenSearch Service가 `AWSServiceRoleForAmazonOpenSearchService` 서비스 연결 역할을 맡으면 정책의이 새 문을 통해 OpenSearch는 OpenSearch에서만 관리하는 애플리케이션의 AWS IAM Identity Center 액세스 범위를 업데이트할 수 OpenSearch. `{`	2025년 3월 31일

변경 사항	설명	날짜
	"Effect": "Allow", "Action": "sso:PutApplicationAccessScope", "Resource": "arn:aws:sso::*:application/*/*", "Condition": { "StringEquals": { "aws:ResourceOrgID": "${aws:PrincipalOrgID}" } } }	
AmazonOpenSearchServerlessServiceRolePolicy 업데이트됨	정책 AmazonOpenSearchServerlessServiceRolePolicy 에 Sid AllowAOSSCloudwatchMetrics 를 추가했습니다. Sid는 정책 문에 입력되는 식별자(선택 사항)로 명령문 ID입니다.	2024년 7월 12일
OpenSearchIngestionSelfManagedVpcePolicy 추가됨	OpenSearch Ingestion에서 수집 파이프라인에 대한 자체 관리형 VPC 액세스 권한을 부여하고, 태그를 생성하며, 수집 관련 CloudWatch 지표를 사용자 계정에 게시할 수 있도록 허용하는 새 정책. 정책 JSON에 대한 자세한 내용은 IAM 콘솔을 참조하세요.	2024년 6월 12일

변경 사항	설명	날짜
`AmazonOpenSearchDirectQueryGlueCreateAccess` 추가됨	Amazon OpenSearch Service Direct Query Service에 `CreateDatabase` , `CreatePartition` `CreateTable` 및 에 대한 액세스 권한을 부여합니다`BatchCreatePartition` AWS Glue API.	2024년 5월 6일
`AmazonOpenSearchServiceRolePolicy` 및 `AmazonElasticsearchServiceRolePolicy` 업데이트됨	서비스 연결 역할 정책 에서 IPv6 주소를 할당하고 할당 취소하는 데 필요한 권한을 추가합니다. 더 이상 사용되지 않는 Elasticsearch 정책 도 이전 버전과의 호환성을 보장하기 위해 업데이트되었습니다.	2023년 10월 18일
`AmazonOpenSearchIngestionServiceRolePolicy` 추가됨	이 정책은 OpenSearch Ingestion에서 수집 파이프라인에 대한 VPC 액세스 권한을 부여하고, 태그를 생성하며, 수집 관련 CloudWatch 지표를 사용자 계정에 게시할 수 있도록 하는 합니다. 정책 JSON에 대한 자세한 내용은 IAM 콘솔을 참조하세요.	2023년 4월 26일

변경 사항	설명	날짜
`AmazonOpenSearchIngestionFullAccess` 추가됨	AWS 계정의 OpenSearch Ingestion API 작업 및 리소스에 대한 전체 액세스 권한을 부여하는 새 정책입니다. 정책 JSON에 대한 자세한 내용은 IAM 콘솔을 참조하세요.	2023년 4월 26일
`AmazonOpenSearchIngestionReadOnlyAccess` 추가됨	AWS 계정의 전체 OpenSearch Ingestion 리소스에 대한 읽기 전용 액세스 권한을 부여하는 새 정책입니다. 정책 JSON에 대한 자세한 내용은 IAM 콘솔을 참조하세요.	2023년 4월 26일
`AmazonOpenSearchServerlessServiceRolePolicy` 추가됨	OpenSearch Serverless 지표 데이터를 Amazon CloudWatch에 보내는 데 필요한 최소 권한을 제공하는 새 정책입니다. 정책 JSON에 대한 자세한 내용은 IAM 콘솔을 참조하세요.	2022년 11월 29일

변경 사항	설명	날짜
`AmazonOpenSearchServiceRolePolicy` 및 `AmazonElasticsearchServiceRolePolicy` 업데이트됨	OpenSearch Service 관리형 VPC 엔드포인트를 생성하기 위해 서비스 연결 역할에 필요한 권한을 추가했습니다. 일부 작업은 요청에 `OpenSearchManaged=true` 태그가 포함된 경우에만 수행할 수 있습니다. 더 이상 사용되지 않는 Elasticsearch 정책 도 이전 버전과의 호환성을 보장하기 위해 업데이트되었습니다.	2022년 11월 7일
`AmazonOpenSearchServiceRolePolicy` 및 `AmazonElasticsearchServiceRolePolicy` 업데이트됨	Amazon CloudWatch에 OpenSearch 클러스터 지표를 게시하는 데 필요한 `PutMetricData` 작업에 대한 지원이 추가되었습니다. 더 이상 사용되지 않는 Elasticsearch 정책 도 이전 버전과의 호환성을 보장하기 위해 업데이트되었습니다. 정책 JSON에 대한 자세한 내용은 IAM 콘솔을 참조하세요.	2022년 9월 12일

변경 사항	설명	날짜
`AmazonOpenSearchServiceRolePolicy` 및 `AmazonElasticsearchServiceRolePolicy` 업데이트됨	`acm` 리소스 유형에 대한 지원이 추가되었습니다. 이 정책은 사용자 지정 엔드포인트 지원 도메인을 생성하고 업데이트하기 위해 서비스 연결 역할이 ACM 리소스를 확인하고 검증하는 데 필요한 최소 AWS Certificate Manager (ACM) 읽기 전용 권한을 제공합니다. 더 이상 사용되지 않는 Elasticsearch 정책도 이전 버전과의 호환성을 보장하기 위해 업데이트되었습니다.	2022년 7월 28일
`AmazonOpenSearchServiceCognitoAccess` 및 `AmazonESCognitoAccess` 업데이트됨	Elasticsearch에서 OpenSearch로 업그레이드하는 중 Cognito 사용자 풀 구성을 설정하는 데 필요한 `UpdateUserPoolClient` 작업에 대한 지원이 추가됐습니다. 모든 리소스에 대한 액세스를 허용하기 위해 `SetIdentityPoolRoles` 작업에 대한 권한을 수정했습니다. 더 이상 사용되지 않는 Elasticsearch 정책도 이전 버전과의 호환성을 보장하기 위해 업데이트되었습니다.	2021년 12월 20일

변경 사항	설명	날짜
AmazonOpenSearchServiceRolePolicy 업데이트됨	security-group 리소스 유형에 대한 지원이 추가되었습니다. 이 정책은 VPC 액세스를 활성화하기 위해 서비스 연결 역할에 필요한 최소 Amazon EC2 및 Elastic Load Balancing 권한을 제공합니다.	2021년 9월 9일
• AmazonOpenSearchServiceFullAccess 추가됨 • AmazonESFullAccess 사용되지 않음	이 새 정책은 이전 정책을 대체하기 위한 것입니다. 두 정책 모두 OpenSearch Service 구성 API와 OpenSearch API용 모든 HTTP 메서드에 대한 전체 액세스 권한을 제공합니다. 세분화된 액세스 제어 및 리소스 기반 정책은 여전히 액세스를 제한할 수 있습니다.	2021년 9월 7일
• AmazonOpenSearchServiceReadOnlyAccess 추가됨 • AmazonESReadOnlyAccess 사용되지 않음	이 새 정책은 이전 정책을 대체하기 위한 것입니다. 두 정책 모두 OpenSearch Service 구성 API(es:Describe* , es:List*, es:Get*)에 대한 읽기 전용 액세스 권한을 제공하며, OpenSearch API용 HTTP 메서드에 대한 액세스 권한은 제공하지 않습니다.	2021년 9월 7일
• AmazonOpenSearchServiceCognitoAccess 추가됨 • AmazonESCognitoAccess 사용되지 않음	이 새 정책은 이전 정책을 대체하기 위한 것입니다. 두 정책 모두 Cognito 인증 활성화에 필요한 최소의 Amazon Cognito 권한을 제공합니다.	2021년 9월 7일

변경 사항	설명	날짜
• AmazonOpenSearchServiceRolePolicy 추가됨 • AmazonElasticsearchServiceRolePolicy 사용되지 않음	이 새 정책은 이전 정책을 대체하기 위한 것입니다. 두 정책 모두 VPC 액세스를 활성화하기 위해 서비스 연결 역할에 필요한 최소의 Amazon EC2 및 Elastic Load Balancing 권한을 제공합니다.	2021년 9월 7일
변경 내용 추적 시작	Amazon OpenSearch Service는 이제 AWS관리형 정책에 대한 변경 사항을 추적합니다.	2021년 9월 7일

교차 서비스 혼동된 대리자 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS교차 서비스 가장은 혼동된 대리자 문제를 초래할 수 있습니다. 교차 서비스 가장은 한 서비스(호출하는 서비스)가 다른 서비스(호출되는 서비스)를 직접적으로 호출할 때 발생할 수 있습니다. 직접적으로 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS 에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

Amazon OpenSearch Service가 리소스에 다른 서비스를 제공하는 권한을 제한하려면 리소스 정책에서 `aws:SourceArn` 및 `aws:SourceAccount` 글로벌 조건 컨텍스트 키를 사용하는 것이 좋습니다. 만약 `aws:SourceArn` 값에 Amazon S3 버킷 ARN과 같은 계정 ID가 포함되어 있지 않은 경우, 권한을 제한하려면 두 전역 조건 컨텍스트 키를 모두 사용해야 합니다. 두 전역 조건 컨텍스트 키와 계정을 포함한 `aws:SourceArn` 값을 모두 사용하는 경우, `aws:SourceAccount` 값 및 `aws:SourceArn` 값의 계정은 동일한 정책 명령문에서 사용할 경우 반드시 동일한 계정 ID를 사용해야 합니다. 하나의 리소스만 교차 서비스 액세스와 연결되도록 허용하려는 경우 `aws:SourceArn`를 사용하세요. 해당 계정의 모든 리소스가 교차 서비스 사용과 연결되도록 허용하려는 경우 `aws:SourceAccount`를 사용하세요.

`aws:SourceArn`의 값은 OpenSearch Service 도메인의 ARN이어야 합니다.

혼동된 대리인 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn` 글로벌 조건 컨텍스트 키를 사용하는 것입니다. 리소스의 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드(*)를 포함한 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용합니다. 예제: `arn:aws:es:*:123456789012:*`.

다음 예는 OpenSearch Service에서 `aws:SourceArn` 및 `aws:SourceAccount` 전역 조건 컨텍스트 키를 사용하여 혼동된 대리자 문제를 방지하는 방법을 보여줍니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":{
      "Sid":"ConfusedDeputyPreventionExamplePolicy",
      "Effect":"Allow",
      "Principal":{
         "Service":"es.amazonaws.com"
      },
      "Action":"sts:AssumeRole",
      "Condition":{
         "StringEquals":{
            "aws:SourceAccount":""
         },
         "ArnLike":{
            "aws:SourceArn":"arn:aws:es:us-east-1::domain/my-domain"
         }
      }
   }
}
```

Amazon OpenSearch Service에서 세분화된 액세스 제어

세분화된 액세스 제어를 사용하면 추가적인 방법으로 Amazon OpenSearch Service에서 데이터에 대한 액세스를 제어할 수 있습니다. 예를 들어, 요청하는 사용자에 따라 하나의 인덱스에서만 검색 결과를 반환하도록 할 수 있습니다. 또는 문서의 특정 필드를 숨기거나 특정 문서를 모두 제외할 수 있습니다.

세분화된 액세스 제어는 다음과 같은 이점을 제공합니다.

- 역할 기반 액세스 제어

- 인덱스, 문서 및 필드 수준의 보안
- OpenSearch 대시보드 멀티테넌시
- OpenSearch 및 OpenSearch 대시보드를 위한 HTTP 기본 인증

주제

- 개요: 세분화된 액세스 제어와 OpenSearch Service 보안
- 주요 개념
- 마스터 사용자 정보
- 세분화된 액세스 제어 활성화
- 마스터 사용자로 OpenSearch 대시보드에 액세스
- 권한 관리
- 권장 구성
- 제한 사항
- 마스터 사용자 수정
- 추가 마스터 사용자
- 수동 스냅샷 수
- 통합
- REST API 차이점
- 자습서: IAM 마스터 사용자 및 Amazon Cognito 인증을 사용하여 도메인 구성
- 자습서: 내부 사용자 데이터베이스와 HTTP 기본 인증을 사용하여 도메인 구성

개요: 세분화된 액세스 제어와 OpenSearch Service 보안

Amazon OpenSearch Service 보안은 크게 3가지 계층으로 구성됩니다.

네트워크

첫 번째 보안 계층은 네트워크로, OpenSearch Service 도메인에 요청이 도달하는지를 결정합니다. 도메인을 생성할 때 퍼블릭 액세스(Public access)를 선택하는 경우 인터넷에 연결된 클라이언트의 요청이 도메인 엔드포인트에 도달할 수 있습니다. VPC 액세스(VPC access)를 선택하는 경우 요청이 엔드포인트에 도달하려면 클라이언트를 VPC에 연결해야 합니다(연결된 보안 그룹에서 이를 허용해야 함). 자세한 내용은 the section called “VPC 지원” 섹션을 참조하세요.

도메인 액세스 정책

두 번째 보안 계층은 도메인 액세스 정책입니다. 도메인 엔드포인트에 요청이 도달하면 리소스 기반 액세스 정책에서 지정된 URI에 대한 요청 액세스를 허용하거나 거부합니다. 액세스 정책은 요청이 OpenSearch 자체에 도달하기 전에 도메인 "경계"에서 요청을 승인하거나 거부합니다.

세분화된 액세스 제어

마지막 세 번째 보안 계층은 세분화된 액세스 제어입니다. 리소스 기반 액세스 정책에서 도메인 엔드포인트에 요청이 도달하도록 허용한 이후 세분화된 액세스 제어에서 사용자 자격 증명을 평가하고 사용자를 인증하거나 요청을 거부합니다. 세분화된 액세스 제어에서 사용자를 인증하는 경우 해당 사용자에게 매핑된 모든 역할을 가져오고 전체 권한 세트를 사용하여 요청을 처리하는 방법을 결정합니다.

> **Note**
>
> 리소스 기반 액세스 정책에 IAM 역할 또는 사용자가 포함된 경우 클라이언트는 AWS 서명 버전 4를 사용하여 서명된 요청을 보내야 합니다. 따라서 액세스 정책이 세분화된 액세스 제어와 충돌할 수 있으며, 내부 사용자 데이터베이스와 HTTP 기본 인증을 사용하는 경우 특히 그렇습니다. 사용자 이름 및 암호와 IAM 보안 인증 정보를 함께 사용하여 요청에 서명할 수는 없습니다. 일반적으로 세분화된 액세스 제어를 활성화하는 경우 서명된 요청이 필요 없는 도메인 액세스 정책을 사용하는 것이 좋습니다.

다음 다이어그램은 세분화된 액세스 제어가 활성화된 VPC 액세스 도메인, IAM 기반 액세스 정책, IAM 마스터 사용자를 포함한 일반적인 구성을 보여줍니다.

다음 다이어그램은 세분화된 액세스 제어가 활성화된 퍼블릭 액세스 도메인, IAM 보안 주체를 사용하지 않는 액세스 정책, 내부 사용자 데이터베이스의 마스터 사용자를 포함한 또 다른 일반적인 구성을 보여줍니다.

예시

`movies/_search?q=thor`에 대한 `GET` 요청을 예로 들어 보겠습니다. 사용자가 `movies` 인덱스를 검색할 수 있는 권한을 갖습니까? 그렇다면 사용자에게 그 안에 있는 모든 문서를 볼 수 있는 권한이 있

나요? 응답에서 일부 필드를 생략하거나 익명화해야 합니까? 마스터 사용자의 경우 응답은 다음과 같을 수 있습니다.

```
{
  "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
        }
      },
      ...
    ]
  }
}
```

보다 제한된 권한을 가진 사용자가 동일한 요청을 실행하는 경우 응답은 다음과 같을 수 있습니다.

```
{
```

```
  "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"
        }
      },
      ...
    ]
  }
}
```

응답의 결과 수와 각 결과의 필드 수가 더 적습니다. 또한 `release_date` 필드는 익명화됩니다. 권한이 없는 사용자가 동일한 요청을 하는 경우 클러스터에서 오류가 반환됩니다.

```
{
  "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
}
```

사용자가 유효하지 않은 자격 증명을 제공하는 경우 클러스터에서 `Unauthorized` 예외가 반환됩니다.

주요 개념

세분화된 액세스 제어를 시작할 때 다음 개념을 고려합니다.

- 역할 - 세분화된 액세스 제어를 사용하는 핵심 방법. 이 경우 역할은 IAM 역할과 구별됩니다. 역할에는 클러스터 전체, 인덱스별, 문서 수준, 필드 수준 등 다양한 권한 조합이 포함됩니다.
- 매핑 - 역할을 구성한 후에는 한 명 이상의 사용자에게 매핑합니다. 예를 들어, Dashboards에 대한 액세스 권한을 제공하는 역할, `index1`에 대한 읽기 전용 액세스 권한을 제공하는 역할, `index2`에 대한 쓰기 액세스 권한을 제공하는 역할이라는 3가지 역할을 단일 사용자에게 매핑할 수 있습니다. 또는 이러한 모든 권한을 단일 역할에 포함할 수 있습니다.
- 사용자 - OpenSearch 클러스터에 요청을 수행하는 사람 또는 애플리케이션. 사용자에게는 요청할 때 지정하는 보안 인증 정보(IAM 액세스 키 또는 사용자 이름과 암호)가 있습니다.

마스터 사용자 정보

OpenSearch Service의 마스터 사용자는 기본 OpenSearch 클러스터에 대한 전체 권한이 있는 사용자 이름과 암호 조합 또는 IAM 위탁자입니다. 사용자는 OpenSearch 대시보드 내에서 내부 사용자, 역할 및 역할 매핑을 생성할 수 있는 기능과 함께 OpenSearch 클러스터에 대한 모든 액세스 권한이 있는 경우 마스터 사용자로 간주됩니다.

OpenSearch Service 콘솔 또는 CLI를 통해 생성된 마스터 사용자는 두 개의 사전 정의된 역할에 자동으로 매핑됩니다.

- `all_access` - 모든 클러스터 전체 작업에 대한 전체 액세스 권한, 모든 클러스터 인덱스에 대한 쓰기 권한 및 모든 테넌트에 대한 쓰기 권한을 제공합니다.
- `security_manager` - 보안 플러그인에 대한 액세스와 사용자 및 권한 관리를 제공합니다.

이 두 역할을 사용하면 사용자는 OpenSearch 대시보드의 보안 탭에 액세스하여 사용자와 권한을 관리할 수 있습니다. 다른 내부 사용자를 생성하고 `all_access` 역할에만 매핑하는 경우 사용자는 보안 탭에 액세스할 수 없습니다. `all_access` 및 `security_manager` 역할 모두에 명시적으로 매핑하여 추가 마스터 사용자를 생성할 수 있습니다. 지침은 the section called “추가 마스터 사용자” 섹션을 참조하세요.

도메인의 마스터 사용자를 생성할 때 기존 IAM 위탁자를 지정하거나 내부 사용자 데이터베이스 내에서 마스터 사용자를 생성할 수 있습니다. 사용할 항목을 결정할 때 다음 사항을 고려합니다.

- IAM 보안 주체 - 마스터 사용자의 IAM 보안 주체를 선택하는 경우 클러스터에 대한 모든 요청은 AWS 서명 버전 4를 사용하여 서명해야 합니다.

  OpenSearch Service는 IAM 위탁자의 권한을 고려하지 않습니다. IAM 사용자 또는 역할은 인증을 위해서만 지원됩니다. 해당 사용자 또는 역할에 대한 정책은 마스터 사용자의 권한 부여와 관련이 없습니다. 권한 부여는 OpenSearch Security 플러그인의 다양한 권한을 통해 처리됩니다.

  예를 들어 IAM 위탁자에 대해 0개의 IAM 권한을 할당할 수 있으며, 기계 또는 사람이 해당 사용자 또는 역할에 인증할 수 있는 한 OpenSearch Service에서 마스터 사용자의 권한을 갖습니다.

  여러 클러스터에서 동일한 사용자를 사용하려는 경우, Amazon Cognito를 사용하여 Dashboards에 액세스하려는 경우 또는 서명 버전 4를 사용한 서명을 지원하는 OpenSearch 클라이언트가 있는 경우 IAM이 권장됩니다.
- 내부 사용자 데이터베이스 - 내부 사용자 데이터베이스에서 마스터를 생성하는 경우(사용자 이름과 암호 조합 사용) HTTP 기본 인증(및 IAM 자격 증명)을 사용하여 클러스터에 대한 요청를 수행할 수 있습니다. 대부분의 클라이언트는 curl을 포함한 기본 인증을 지원하며, --aws-sigv4 옵션을 통해 AWS 서명 버전 4도 지원합니다. 내부 사용자 데이터베이스는 OpenSearch 인덱스에 저장되므로 다른 클러스터와 공유할 수 없습니다.

  여러 클러스터에서 사용자를 재사용할 필요가 없는 경우, Amazon Cognito가 아닌 HTTP 기본 인증을 사용하여 Dashboards에 액세스하려는 경우 또는 기본 인증만 지원하는 클라이언트가 있는 경우 내부 사용자 데이터베이스가 권장됩니다. 내부 사용자 데이터베이스는 OpenSearch Service를 시작하는 가장 간단한 방법입니다.

세분화된 액세스 제어 활성화

콘솔 AWS CLI또는 구성 API를 사용하여 세분화된 액세스 제어를 활성화합니다. 단계는 도메인 생성 및 관리를 참조하세요.

세분화된 액세스 제어는 OpenSearch 또는 Elasticsearch 6.7 이상이 필요합니다. 저장된 데이터 암호화 및 노드 간 암호화 등 도메인에 대한 모든 트래픽에 HTTPS가 필요합니다. 세분화된 액세스 제어의 고급 기능을 구성하는 방법에 따라 요청을 추가로 처리하려면 개별 데이터 노드의 컴퓨팅 및 메모리 리소스가 필요할 수 있습니다. 세분화된 액세스 제어를 활성화한 후에는 비활성화할 수 없습니다.

기존 도메인에서의 세분화된 액세스 제어 사용 설정

OpenSearch 또는 Elasticsearch 6.7 이상을 실행하는 기존 도메인에서 세분화된 액세스 제어를 사용 설정할 수 있습니다.

기존 도메인(콘솔)에서 세분화된 액세스 제어 사용 설정

1. 도메인을 선택하고 작업(Actions), 보안 구성 편집(Edit security configuration)을 선택합니다.
2. 세분화된 액세스 제어 사용 설정(Enable fine-grained access control)을 선택합니다.
3. 마스터 사용자를 생성하는 방법을 선택합니다.
   - 사용자 관리에 IAM을 사용하려면 IAM ARN을 마스터 사용자로 설정(Set IAM ARN as master user)을 선택하고 IAM 역할의 ARN을 지정합니다.
   - 내부 사용자 데이터베이스를 사용하려면 [기본 사용자 생성]을 선택하고 사용자 이름과 암호를 지정합니다.
4. (선택 사항) Open/IP 기반 액세스 정책에 대한 마이그레이션 기간 사용 설정(Enable migration period for open/IP-based access policy)을 선택합니다. 이 설정을 사용하면 기존 사용자가 중단 없이 도메인에 계속 액세스할 수 있는 30일의 전환 기간을 사용할 수 있으며, 기존의 개방형 및 IP 기반 액세스 정책이 도메인에서 계속 사용됩니다. 이 마이그레이션 기간 동안 관리자는 도메인에 대해 필요한 역할을 생성하고 사용자에게 매핑하는 것이 좋습니다. 개방형 또는 IP 기반 액세스 정책 대신 자격 증명 기반 정책을 사용하는 경우 이 설정을 사용 중지할 수 있습니다.

   또한 마이그레이션 기간 동안 세분화된 액세스 제어 기능을 사용할 수 있도록 클라이언트를 업데이트해야 합니다. 예를 들어, 세분화된 액세스 제어로 IAM 역할을 매핑하는 경우 AWS 서명 버전 4로 요청에 서명하도록 클라이언트를 업데이트해야 합니다. 세분화된 액세스 제어를 사용하여 HTTP 기본 인증을 구성하는 경우 요청에서 적절한 기본 인증 자격 증명을 제공하도록 클라이언트를 업데이트해야 합니다.

   마이그레이션 기간 동안 도메인의 OpenSearch 대시보드 엔드포인트에 액세스하는 사용자에게는 로그인 페이지 대신 검색 페이지가 바로 표시됩니다. 관리자 및 마스터 사용자는 로그인을 선택하여 관리자 자격 증명으로 로그인하고 역할 매핑을 구성할 수 있습니다.

   > **Important**
   >
   > OpenSearch Service는 30일 후에 마이그레이션 기간을 자동으로 사용 중지합니다. 필요한 역할을 만들고 사용자에게 매핑하는 즉시 종료하는 것이 좋습니다. 마이그레이션 기간이 끝난 후에는 다시 사용 설정할 수 없습니다.
5. 변경 사항 저장을 선택합니다.

변경 사항은 클러스터 상태가 빨간색으로 바뀐 동안 블루/그린 배포를 트리거하지만, 모든 클러스터 작업은 영향을 받지 않습니다.

기존 도메인(CLI) 에 대한 세분화된 액세스 제어 사용 설정

AnonymousAuthEnabled를 true로 설정하여 세분화된 액세스 제어를 통해 마이그레이션 기간을 사용 설정합니다.

```
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}'
```

default_role 정보

세분화된 액세스 제어에는 역할 매핑이 필요합니다. 도메인이 ID 기반 액세스 정책을 사용하는 경우, OpenSearch Service는 사용자를 default_role이라는 새로운 역할에 자동으로 매핑하여 기존 사용자를 올바르게 마이그레이션하도록 합니다. 이 임시 매핑을 사용하면 사용자가 고유한 역할 매핑을 생성할 때까지 IAM 서명이 된 GET 및 PUT 요청을 성공적으로 보낼 수 있습니다.

이 역할은 OpenSearch Service 도메인에 보안 취약점이나 결함을 추가하지 않습니다. 자신의 역할을 설정하고 적절히 매핑하는 즉시 기본 역할을 삭제하는 것이 좋습니다.

마이그레이션 시나리오

다음 표에서는 기존 도메인에 대한 세분화된 액세스 제어를 사용 설정하기 전후의 각 인증 방법에 대한 동작과 관리자가 사용자를 역할에 올바르게 매핑하기 위해 수행해야 하는 단계에 대해 설명합니다.

인증 방법	세분화된 액세스 제어 사용 설정 전	세분화된 액세스 제어 사용 설정 후	관리자 작업
ID 기반 정책	IAM 정책을 충족하는 모든 I사용자가 도메인에 액세스할 수 있습니다.	마이그레이션 기간을 사용 설정할 필요는 없습니다. OpenSearch Service가 IAM 정책을 만족하는 모든 사용자를 default_role에 자동으로 매핑하여 도메인에	1. 도메인에서 사용자 지정 역할 매핑을 생성합니다. 2. default_role을 삭제합니다.

인증 방법	세분화된 액세스 제어 사용 설정 전	세분화된 액세스 제어 사용 설정 후	관리자 작업
		계속 액세스할 수 있도록 합니다.	
IP 기반 정책	허용된 IP 주소 또는 CIDR 블록의 모든 사용자가 도메인에 액세스할 수 있습니다.	30일의 마이그레이션 기간 동안 허용된 IP 주소 또는 CIDR 블록의 모든 사용자가 도메인에 계속 액세스할 수 있습니다.	1. 도메인에서 사용자 지정 역할 매핑을 생성합니다. 2. 역할 매핑 구성에 따라 기본 인증 자격 증명 또는 IAM 자격 증명을 제공하도록 클라이언트를 업데이트합니다. 3. 마이그레이션 기간을 사용 중지합니다. 허용된 IP 주소 또는 CIDR 블록의 사용자가 기본 인증 또는 IAM 자격 증명 없이 요청을 보내면 도메인에 대한 액세스 권한을 잃게 됩니다.
개방형 액세스 정책	인터넷을 통해 모든 사용자가 도메인에 액세스할 수 있습니다.	30일의 마이그레이션 기간 동안 인터넷을 통해 모든 사용자가 도메인에 계속 액세스할 수 있습니다.	1. 도메인에서 역할 매핑을 생성합니다. 2. 역할 매핑 구성에 따라 기본 인증 자격 증명 또는 IAM 자격 증명을 제공하도록 클라이언트를 업데이트합니다. 3. 마이그레이션 기간을 사용 중지합니다. 기본 인증 또는 IAM 자격 증명 없이 요청을 보내는 사용자는 도메인에 대한 액세스 권한을 잃게 됩니다.

마스터 사용자로 OpenSearch 대시보드에 액세스

세분화된 액세스 제어 기능에는 관리 작업을 간소화하는 OpenSearch 대시보드 플러그인이 포함되어 있습니다. Dashboards를 사용하여 사용자, 역할, 매핑, 작업 그룹 및 테넌트를 관리할 수 있습니다. 그러나 OpenSearch 대시보드 로그인 페이지와 기본 인증 방법은 사용자를 관리하고 도메인을 구성한 방법에 따라 다릅니다.

- 사용자 관리를 위해 IAM을 사용하려면 the section called “OpenSearch Dashboards에 대한 Amazon Cognito 인증”을 클릭하여 Dashboards에 액세스합니다. 그렇지 않으면 Dashboards에서 작동하지 않는 로그인 페이지가 표시됩니다. the section called “제한 사항” 섹션을 참조하세요.

  Amazon Cognito 인증과 함께 자격 증명 풀의 수임된 역할 중 하나가 마스터 사용자에 지정한 IAM 역할과 일치해야 합니다. 이 구성에 대한 자세한 내용은 the section called “(선택 사항) 세분화된 액세스 구성” 및 the section called “자습서: Cognito 인증을 사용한 세분화된 액세스 제어” 섹션을 참조하세요.

- 내부 사용자 데이터베이스를 사용하도록 선택한 경우 마스터 사용자 이름과 암호를 사용하여 대시보드에 로그인할 수 있습니다. HTTPS를 통해 Dashboards에 액세스해야 합니다. Dashboards에 대한 Amazon Cognito 및 SAML 인증은 모두 이 로그인 화면을 대체합니다.

  이 구성에 대한 자세한 내용은 the section called “자습서: 기본 인증을 사용하는 내부 사용자 데이터베이스” 섹션을 참조하세요.

- SAML 인증을 사용하도록 선택한 경우 외부 자격 증명 공급자의 자격 증명을 사용하여 로그인할 수 있습니다. 자세한 내용은 the section called “OpenSearch 대시보드에 대한 SAML 인증” 섹션을 참조하세요.

권한 관리

the section called “주요 개념”에 설명된 대로 세분화된 액세스 제어 권한은 역할, 사용자 및 매핑을 사용하여 관리합니다. 이 단원에서는 이러한 리소스를 생성하고 적용하는 방법을 설명합니다. 이러한 작업을 수행하려면 마스터 사용자로 Dashboards에 로그인하는 것이 좋습니다.

> **Note**
>
> 사용자에게 부여하기로 선택하는 권한은 사용 사례에 따라 크게 다릅니다. 모든 시나리오를 이 설명서에서 실행 가능할 만큼 다룰 수는 없습니다. 사용자에게 부여할 권한을 결정할 때 다음 섹션에서 언급한 OpenSearch 클러스터 및 인덱스 권한을 참조하고 항상 최소 권한의 원칙을 따르십시오.

역할 생성

OpenSearch 대시보드 또는 REST API의 `_plugins/_security` 작업을 사용하여 세분화된 액세스 제어를 위한 새 역할을 생성할 수 있습니다. 자세한 정보는 역할 생성 섹션을 참조하세요.

세분화된 액세스 제어에는 여러 가지 미리 정의된 역할도 포함됩니다. OpenSearch 대시보드 및 Logstash와 같은 클라이언트는 OpenSearch에 다양한 요청을 수행하므로 최소한의 권한 세트를 사용하여 역할을 수동으로 생성하기가 어려울 수 있습니다. 예를 들어 `opensearch_dashboards_user` 역할에는 사용자가 인덱스 패턴, 시각화, 대시보드 및 테넌트를 사용하는 데 필요한 권한이 포함됩니다. 다른 인덱스에 대한 액세스를 허용하는 추가 역할과 함께 Dashboards에 액세스하는 모든 사용자 또는 백엔드 역할에 이를 매핑하는 것이 좋습니다.

Amazon OpenSearch Service는 다음과 같은 OpenSearch 역할을 제공하지 않습니다.

- `observability_full_access`
- `observability_read_access`
- `reports_read_access`
- `reports_full_access`

Amazon OpenSearch Service는 OpenSearch에서는 사용할 수 없는 여러 역할을 제공합니다.

- `ultrawarm_manager`
- `ml_full_access`
- `cold_manager`
- `notifications_full_access`
- `notifications_read_access`

클러스터 수준 보안

클러스터 수준 권한에는 `_mget`, `_msearch` 및 `_bulk`와 같은 다양한 요청을 실행하고, 상태를 모니터링하고, 스냅샷을 생성하는 것 등이 포함됩니다. 역할을 생성할 때 클러스터 권한(Cluster Permissions) 섹션을 사용하여 이러한 권한을 관리합니다. 클러스터 수준 권한의 전체 목록은 클러스터 권한 섹션을 참조하세요.

개별 권한보다는 기본 작업 그룹을 조합하여 원하는 보안 태세를 유지할 수 있는 경우가 많습니다. 클러스터 수준 작업 그룹의 목록은 클러스터 수준 섹션을 참조하세요.

인덱스 수준 보안

인덱스 수준 권한에는 새 인덱스를 생성하고, 인덱스를 검색하고, 문서를 읽고 쓰고, 문서를 삭제하고, 별칭을 관리하는 것 등이 포함됩니다. 역할을 생성할 때 인덱스 권한(Index Permissions) 섹션을 사용하여 이러한 권한을 관리합니다. 인덱스 수준 권한의 전체 목록은 인덱스 권한 부여 섹션을 참조하세요.

개별 권한보다는 기본 작업 그룹을 조합하여 원하는 보안 태세를 유지할 수 있는 경우가 많습니다. 인덱스 수준 작업 그룹의 목록은 인덱스 수준 섹션을 참조하세요.

문서 수준 보안

문서 수준 보안을 사용하면 인덱스 내에서 사용자가 볼 수 있는 문서를 제한할 수 있습니다. 역할을 생성할 때 인덱스 패턴과 OpenSearch 쿼리를 지정합니다. 해당 역할에 매핑하는 모든 사용자는 쿼리와 일치하는 문서만 볼 수 있습니다. 문서 수준 보안은 검색할 때 반환되는 결과 수에 영향을 미칩니다.

자세한 내용은 문서 수준 보안 섹션을 참조하세요.

필드 수준 보안

필드 수준 보안을 사용하면 사용자가 볼 수 있는 문서 필드를 제어할 수 있습니다. 역할을 생성할 때 포함하거나 제외할 필드 목록을 추가합니다. 필드를 포함하면 해당 역할에 매핑되는 모든 사용자가 해당 필드만 볼 수 있습니다. 필드를 제외하면 제외된 필드 이외의 모든 필드를 볼 수 있습니다. 필드 수준 보안은 검색할 때 결과에 포함되는 필드 수에 영향을 미칩니다.

자세한 내용은 필드 수준 보안 섹션을 참조하세요.

필드 마스킹

필드 마스킹은 필드 수준 보안의 대안으로, 필드를 제거하는 대신 필드의 데이터를 익명화합니다. 역할을 생성할 때 마스킹할 필드 목록을 추가합니다. 필드 마스킹은 검색할 때 필드의 내용을 볼 수 있는지에 영향을 미칩니다.

> **Tip**
>
> 표준 마스킹을 필드에 적용하는 경우 OpenSearch Service는 부정확한 집계 결과를 초래할 수 있는 안전한 무작위 해시를 사용합니다. 마스킹 처리된 필드에서 집계를 수행하려면 패턴 기반 마스킹을 대신 사용합니다.

사용자 생성

내부 사용자 데이터베이스를 활성화한 경우 OpenSearch 대시보드 또는 REST API의 `_plugins/_security` 작업을 사용하여 사용자를 생성할 수 있습니다. 자세한 내용은 사용자 생성 섹션을 참조하세요.

마스터 사용자에 IAM을 선택한 경우 이 Dashboards 부분은 무시하고 대신 IAM 역할을 생성합니다. 자세한 내용은 IAM 사용 설명서를 참조하십시오.

사용자에 역할 매핑

역할 매핑은 세분화된 액세스 제어의 가장 중요한 부분입니다. 세분화된 액세스 제어에는 시작하는 데 도움이 되는 몇 가지 미리 정의된 역할이 있지만, 사용자에게 역할을 매핑하지 않으면 클러스터에 대한 모든 요청이 권한 오류로 끝납니다.

백엔드 역할은 역할 매핑 프로세스를 단순화하는 데 도움이 될 수 있습니다. 동일한 역할을 100명의 개별 사용자에게 매핑하는 대신 100명의 사용자 모두가 공유하는 단일 백엔드 역할에 역할을 매핑할 수 있습니다. 백엔드 역할은 IAM 역할 또는 임의 문자열일 수 있습니다.

- Users(사용자) 섹션에서 사용자, 사용자 ARN 및 Amazon Cognito 사용자 문자열을 지정합니다. Cognito 사용자 문자열은 `Cognito/user-pool-id/username` 형식을 사용합니다.
- 백엔드 역할(Backend roles) 섹션에서 백엔드 역할 및 IAM 역할 ARN을 지정합니다.

OpenSearch 대시보드 또는 REST API의 `_plugins/_security` 작업을 사용하여 사용자에게 역할을 매핑할 수 있습니다. 자세한 내용은 사용자를 역할에 매핑 섹션을 참조하세요.

작업 그룹 생성

작업 그룹은 여러 리소스에서 재사용할 수 있는 권한 세트입니다. 대부분의 사용 사례에서 기본 작업 그룹으로 충분하지만 OpenSearch 대시보드 또는 REST API의 `_plugins/_security` 작업을 사용하여 새 작업 그룹을 생성할 수 있습니다. 기본 작업 그룹에 대한 자세한 내용은 기본 작업 그룹 섹션을 참조하세요.

OpenSearch 대시보드 멀티테넌시

테넌트는 인덱스 패턴, 시각화, 대시보드 및 기타 Dashboards 객체를 저장하는 공간입니다. Dashboards 멀티-테넌시를 사용하면 작업을 다른 Dashboards 사용자와 안전하게 공유하거나 프라이

빗 상태로 유지하고 테넌트를 역동적으로 구성할 수 있습니다. 테넌트에 액세스할 수 있는 역할과 해당 역할에 읽기 또는 쓰기 액세스 권한이 있는지를 제어할 수 있습니다. 글로벌 테넌트가 기본값입니다. 자세한 내용은 OpenSearch 대시보드 멀티테넌시를 참조하세요.

현재 테넌트를 보거나 테넌트를 변경하려면

1. OpenSearch 대시보드로 이동하여 로그인합니다.
2. 오른쪽 상단에서 사용자 아이콘을 선택하고 테넌트 전환(Switch tenants)을 선택합니다.
3. 시각화 또는 대시보드를 생성하기 전에 테넌트를 확인합니다. 다른 모든 Dashboards 사용자와 작업을 공유하려면 글로벌(Global)을 선택합니다. 일부 Dashboards 사용자와 하위 작업을 공유하려면 다른 공유 테넌트를 선택합니다. 그렇지 않으면 프라이빗(Private)을 선택합니다.

> **Note**
>
> OpenSearch 대시보드는 각 테넌트에 별도의 인덱스를 유지하고 `tenant_template` 인덱스 템플릿을 만듭니다. `tenant_template` 인덱스를 삭제하거나 수정하지 마십시오. 테넌트 인덱스 매핑이 잘못 구성되면 OpenSearch 대시보드가 오작동할 수 있기 때문입니다.

권장 구성

Amazon은 세분화된 액세스 제어가 다른 보안 기능과 상호 작용하는 방식을 고려하여 대부분의 사용 사례에서 원활하게 작동하는 세분화된 액세스 제어 구성 몇 가지를 권장합니다.

설명	마스터 사용자	도메인 액세스 정책
OpenSearch API 호출에 IAM 자격 증명을 사용하고, SAML 인증을 사용하여 Dashboards에 액세스합니다. Dashboards 또는 REST API를 사용하여 세분화된 액세스 제어 역할을 관리합니다.	IAM 역할 또는 사용자	JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": "es:ESHttp*",`

설명	마스터 사용자	도메인 액세스 정책
		"Resource": "arn:aws:es: us-east-1 :111122223333 :domain/domain-name /*"
OpenSearch API 호출에 IAM 자격 증명 또는 기본 인증을 사용합니다. Dashboards 또는 REST API를 사용하여 세분화된 액세스 제어 역할을 관리합니다. 이 구성은 특히 기본 인증만 지원하는 OpenSearch 클라이언트가 있는 경우 많은 유연성을 제공합니다. 기존 자격 증명 공급자가 있는 경우 SAML 인증을 사용하여 Dashboards에 액세스합니다. 그렇지 않으면 내부 사용자 데이터베이스에서 Dashboards 사용자를 관리합니다.	사용자 이름 및 암호	JSON { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": "es:ESHttp*", "Resource": "arn:aws:es: us-east-1 :111122223333 :domain/domain-name /*" }] }

설명	마스터 사용자	도메인 액세스 정책
OpenSearch API 호출에 IAM 자격 증명을 사용하고, Amazon Cognito를 사용하여 Dashboards에 액세스합니다. Dashboards 또는 REST API를 사용하여 세분화된 액세스 제어 역할을 관리합니다.	IAM 역할 또는 사용자	JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": "es:ESHttp*", "Resource": "arn:aws:es: us-east-1 :111122223333 :domain/domain-name /*" } ] }`

설명	마스터 사용자	도메인 액세스 정책

OpenSearch API 호출에 IAM 자격 증명을 사용하고, Dashboards에 대한 대부분의 액세스를 차단합니다. REST API를 사용하여 세분화된 액세스 제어 역할을 관리합니다.	IAM 역할 또는 사용자	JSON
		{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": "arn:aws:es: *us-east-1* ::domain/*domain-name* /*"
		},
		{
		"Effect": "Deny",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": "arn:aws:es: *us-east-1* ::domain/*domain-name* /_dashboards*"
		}
		]
		}

제한 사항

세분화된 액세스 제어에는 몇 가지 중요한 제한 사항이 있습니다.

- 역할을 호스트 이름 또는 IP 주소에 매핑하는 `hosts` 측면의 역할 매핑은 도메인이 VPC 내에 있는 경우 작동하지 않습니다. 사용자 및 백엔드 역할에는 여전히 역할을 매핑할 수 있습니다.
- 마스터 사용자에 IAM을 선택하고 Amazon Cognito 또는 SAML 인증을 활성화하지 않으면 Dashboards에 작동하지 않는 로그인 페이지가 표시됩니다.

- 마스터 사용자에 IAM을 선택하는 경우에도 내부 사용자 데이터베이스에 사용자를 생성할 수 있습니다. 그러나 이 구성에서는 HTTP 기본 인증이 활성화되지 않으므로 이러한 사용자 자격 증명으로 서명된 모든 요청이 거부됩니다.
- SQL을 사용하여 액세스 권한이 없는 인덱스를 쿼리하는 경우 “no permissions(권한 없음)” 오류가 발생합니다. 인덱스가 없으면 “no such index(해당 인덱스 없음)” 오류가 발생합니다. 오류 메시지의 이러한 차이는 이름을 추측할 경우 인덱스의 존재를 확인할 수 있음을 의미합니다.

  문제를 최소화하려면 인덱스 이름에 민감한 정보를 포함하지 마세요. SQL에 대한 모든 액세스를 거부하려면 도메인 액세스 정책에 다음 요소를 추가합니다.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": [
      "*"
    ]
  },
  "Action": [
    "es:*"
  ],
  "Resource": "arn:aws:es:us-east-1:123456789012:domain/my-domain/_plugins/_sql"
}
```

- 도메인 버전이 2.3 이상이고 세분화된 액세스 제어를 활성화한 경우 `max_clause_count`(을)를 1로 설정하면 도메인에 문제가 발생할 수 있습니다. 이 계정을 더 높은 숫자로 설정하는 것이 좋습니다.
- 세분화된 액세스 제어가 설정되지 않은 도메인에서 세분화된 액세스 제어를 활성화하면 직접 쿼리를 위해 생성된 데이터 소스의 경우 세분화된 액세스 제어 역할을 직접 설정해야 합니다. 세분화된 액세스 역할을 설정하는 방법에 대한 자세한 내용은 Amazon S3와 Amazon OpenSearch Service 데이터 소스 통합 생성을 참조하세요.

마스터 사용자 수정

마스터 사용자의 세부 정보를 잊어버린 경우 콘솔, AWS CLI또는 구성 API를 사용하여 다시 구성할 수 있습니다.

마스터 사용자를 수정하려면(콘솔)

1. https://console.aws.amazon.com/aos/home/에서 Amazon OpenSearch Service 콘솔로 이동합니다.

2. 도메인을 선택하고 Actions(작업), Edit security configuration(보안 구성 편집)을 선택합니다.
3. IAM ARN을 마스터 사용자로 설정(Set IAM ARN as master user) 또는 마스터 사용자 생성(Create master user)을 선택합니다.

   - 이전에 IAM 마스터 사용자를 사용한 경우 세분화된 액세스 제어가 `all_access` 역할을 지정한 새 IAM ARN에 다시 매핑합니다.
   - 이전에 내부 사용자 데이터베이스를 사용한 경우 세분화된 액세스 제어가 새 마스터 사용자를 생성합니다. 새 마스터 사용자를 사용하여 이전 마스터 사용자를 삭제할 수 있습니다.
   - 내부 사용자 데이터베이스에서 IAM 마스터 사용자로 전환하면 내부 사용자 데이터베이스에서 사용자가 삭제되지 않습니다. 대신 HTTP 기본 인증을 비활성화합니다. 내부 사용자 데이터베이스에서 사용자를 수동으로 삭제하거나 HTTP 기본 인증을 다시 활성화해야 할 경우를 대비하여 보관합니다.
4. 변경 사항 저장을 선택합니다.

추가 마스터 사용자

도메인을 생성할 때 마스터 사용자를 지정하지만 원하는 경우 이 마스터 사용자를 사용하여 추가 마스터 사용자를 생성할 수 있습니다. OpenSearch 대시보드 또는 REST API의 두 가지 옵션이 있습니다.

- Dashboards를 사용하는 경우 보안(Security), 역할(Roles)을 선택한 다음 새 마스터 사용자를 `all_access` 및 `security_manager` 역할에 매핑합니다.

- REST API를 사용하려면 다음 요청을 보냅니다.

```
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"
  ]
}
```

이러한 요청은 현재 역할 매핑을 대체하므로 PUT 요청에 현재 역할을 모두 포함할 수 있도록 GET 요청을 먼저 수행합니다. REST API는 Kibana에 액세스할 수 없고 IAM 역할을 Amazon Cognito에서 `all_access` 역할로 매핑하려는 경우에 특히 유용합니다.

수동 스냅샷 수

세분화된 액세스 제어를 사용하면 수동 스냅샷을 생성하는 데 따른 복잡성이 가중됩니다. 다른 모든 용도로 HTTP 기본 인증을 사용하더라도 스냅샷 리포지토리를 등록하려면 the section called “사전 조건”에 정의된 대로 `TheSnapshotRole`을 수임할 `iam:PassRole` 권한이 있는 IAM 역할에 `manage_snapshots` 역할을 매핑해야 합니다.

그런 다음 the section called “수동 스냅샷 리포지토리 등록”에 설명된 대로 해당 IAM 역할을 사용하여 서명된 요청을 도메인으로 보냅니다.

통합

OpenSearch Service에서 다른 AWS 서비스를 사용하는 경우 해당 서비스에 대한 IAM 역할에 적절한 권한을 제공해야 합니다. 예를 들어 Firehose 전송 스트림은 종종 `firehose_delivery_role`이라는 IAM 역할을 사용합니다. Dashboards에서 세분화된 액세스 제어를 위한 역할을 생성하고 이 역할에 IAM 역할을 매핑합니다. 이 경우 새 역할에는 다음 권한이 필요합니다.

```
{
  "cluster_permissions": [
    "cluster_composite_ops",
    "cluster_monitor"
  ],
  "index_permissions": [{
    "index_patterns": [
```

```
      "firehose-index*"
    ],
    "allowed_actions": [
      "create_index",
      "manage",
      "crud"
    ]
  }]
}
```

권한은 각 서비스가 수행하는 작업에 따라 다릅니다. 데이터를 인덱싱하는 AWS IoT 규칙 또는 AWS Lambda 함수에는 Firehose와 유사한 권한이 필요할 수 있지만 검색만 수행하는 Lambda 함수는 더 제한된 세트를 사용할 수 있습니다.

REST API 차이점

세분화된 액세스 제어 REST API는 OpenSearch/Elasticsearch 버전에 따라 약간 차이가 있습니다. PUT 요청을 수행하기 전에 GET 요청을 수행하여 예상 요청 본문을 확인합니다. 예를 들어, `_plugins/_security/api/user`에 대한 GET 요청은 모든 사용자를 반환하며, 이를 수정하여 유효한 PUT 요청을 생성하는 데 사용할 수 있습니다.

Elasticsearch 6.x에서 사용자를 생성하는 요청은 다음과 같습니다.

```
PUT _opendistro/_security/api/user/new-user
{
  "password": "some-password",
  "roles": ["new-backend-role"]
}
```

OpenSearch 또는 Elasticsearch 7.x에서 요청은 다음과 같습니다(Elasticsearch를 사용하는 경우 `_plugins`를 `_opendistro`로 바꾸기).

```
PUT _plugins/_security/api/user/new-user
{
  "password": "some-password",
  "backend_roles": ["new-backend-role"]
}
```

Elasticsearch 6.x에서 테넌트는 역할의 속성입니다.

```
GET _opendistro/_security/api/roles/all_access
```

```
{
  "all_access": {
    "cluster": ["UNLIMITED"],
    "tenants": {
      "admin_tenant": "RW"
    },
    "indices": {
      "*": {
        "*": ["UNLIMITED"]
      }
    },
    "readonly": "true"
  }
}
```

OpenSearch 및 Elasticsearch 7.x에서 이는 자체 URI가 있는 객체입니다(Elasticsearch를 사용하는 경우 `_plugins`를 `_opendistro`로 바꾸기).

```
GET _plugins/_security/api/tenants

{
  "global_tenant": {
    "reserved": true,
    "hidden": false,
    "description": "Global tenant",
    "static": false
  }
}
```

OpenSearch REST API에 대한 설명은 보안 플러그인 API 참조를 참조하세요.

Tip

내부 사용자 데이터베이스를 사용하는 경우 curl을 사용하여 요청을 수행하고 도메인을 테스트할 수 있습니다. 다음 샘플 명령을 시도해보세요.

```
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'
```

자습서: IAM 마스터 사용자 및 Amazon Cognito 인증을 사용하여 도메인 구성

이 자습서에서는 세분화된 액세스 제어를 위한 인기 있는 Amazon OpenSearch Service 사용 사례인 OpenSearch 대시보드에 대한 Amazon Cognito 인증을 사용하는 IAM 마스터 사용자를 다룹니다.

이 자습서에서는 마스터 IAM 역할과 제한된 IAM 역할을 구성한 다음 이를 Amazon Cognito의 사용자와 연결합니다. 그러면 마스터 사용자는 OpenSearch Dashboard에 로그인하여 제한된 사용자를 역할에 매핑하고 세분화된 액세스 제어를 사용하여 사용자의 권한을 제한할 수 있습니다.

이러한 단계는 Amazon Cognito 사용자 풀을 인증에 사용하지만, 동일한 기본 프로세스가 모든 Cognito 인증 공급자에 대해 작동하므로 다양한 사용자에게 다양한 IAM 역할을 할당할 수 있습니다.

이 자습서에서는 다음 단계를 완료합니다.

1. 마스터 및 제한된 IAM 역할 생성
2. Cognito 인증을 사용하여 도메인 생성
3. Cognito 사용자 풀 및 자격 증명 풀을 구성합니다
4. OpenSearch 대시보드에서 역할 매핑
5. 권한 테스트

1단계: 마스터 및 제한된 IAM 역할 생성

AWS Identity and Access Management (IAM) 콘솔로 이동하여 두 개의 개별 역할을 생성합니다.

- `MasterUserRole` – 마스터 사용자는 클러스터에 대한 전체 권한을 갖고 역할과 역할 매핑을 관리합니다.
- `LimitedUserRole` – 마스터 사용자로서 제한된 액세스 권한을 부여하는 좀 더 제한된 역할입니다.

역할을 생성하는 지침은 IAM 사용 설명서의 사용자 지정 신뢰 정책을 사용하여 역할 생성을 참조하세요.

두 역할 모두 Cognito 자격 증명 풀이 역할을 맡도록 허용하는 다음 신뢰 정책이 있어야 합니다.

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**
>
> `identity-pool-id`를 Amazon Cognito 자격 증명 풀의 고유 식별자로 교체하세요. 예를 들어 `us-east-1:0c6cdba7-3c3c-443b-a958-fb9feb207aa6`입니다.

2단계: Cognito 인증을 사용하여 도메인 생성

https://console.aws.amazon.com/aos/home/ Amazon OpenSearch Service 콘솔로 이동하여 다음 설정을 사용하여 도메인을 생성합니다.

- OpenSearch 1.0 이상 또는 Elasticsearch 7.8 이상
- 공개 액세스(Public access)
- 마스터 사용자(이전 단계에서 생성)로 `MasterUserRole`이(가) 활성화된 세분화된 액세스 제어
- OpenSearch 대시보드에 대한 Amazon Cognito 인증 활성화. Cognito 인증을 활성화하고 사용자 및 ID 풀을 선택하는 방법에 대한 지침은 the section called “Amazon Cognito 인증을 사용하도록 도메인 구성” 섹션을 참조하세요.

- 다음 도메인 액세스 정책:

  JSON

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

- 도메인에 대한 모든 트래픽에 HTTPS 필요
- 노드 간 암호화
- 저장 데이터의 암호화

3단계: Cognito 사용자 및 그룹 구성

도메인이 생성되는 동안 Amazon Cognito 개발자 안내서의 사용자 풀 생성에 따라 Amazon Cognito 내에서 마스터 및 제한된 사용자를 구성합니다. 마지막으로 Amazon Cognito에서 자격 증명 풀 생성의 단계에 따라 자격 증명 풀을 구성합니다. 사용자 풀과 자격 증명 풀은 동일한 AWS 리전에 있어야 합니다.

4단계: OpenSearch 대시보드에서 역할 매핑

이제 사용자가 구성되었으므로 OpenSearch Dashboards에 마스터 사용자로 로그인하여 사용자를 역할에 매핑할 수 있습니다.

1. OpenSearch Service 콘솔로 돌아가서 생성한 도메인의 OpenSearch 대시보드 URL로 이동합니다. URL은 *domain-endpoint*/_dashboards/ 형식입니다.
2. master-user 보안 인증으로 로그인합니다.
3. Add sample data(샘플 데이터 추가)를 선택하고 샘플 비행 데이터를 추가합니다.
4. 왼쪽 탐색 창에서 Security(보안), Roles(역할), Create role(역할 생성)을 선택합니다.

5. 역할 이름을 new-role로 지정합니다.
6. Index(인덱스)의 경우 opensearch_dashboards_sample_data_fli*(Elasticsearch 도메인의 경우 kibana_sample_data_fli*)를 지정합니다.
7. Index permissions(인덱스 권한)의 경우 read(읽기)를 선택합니다.
8. 문서 수준 보안 쿼리(Document Level Security Query)에 다음 쿼리를 지정합니다.

```
{
  "match": {
    "FlightDelay": true
  }
}
```

9. 필드 수준 보안의 경우 제외(Exclude)를 선택하고 FlightNum을 지정합니다.
10. 익명화(Anonymization)에 Dest를 지정합니다.
11. 생성(Create)을 선택합니다.
12. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다. LimitedUserRole에 대한 Amazon 리소스 이름(ARN)을 외부 자격 증명으로 추가하고 Map(매핑)을 선택합니다.
13. 역할 목록으로 돌아가서 opensearch_dashboards_user를 선택합니다. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다. LimitedUserRole에 대한 ARN을 백엔드 역할로 추가하고 맵(Map)을 선택합니다.

5단계: 권한 테스트

역할이 올바르게 매핑되면 제한된 사용자로 로그인하고 권한을 테스트할 수 있습니다.

1. 새로운 프라이빗 브라우저 창에서 도메인의 OpenSearch 대시보드 URL로 이동하고 limited-user 보안 인증을 사용하여 로그인한 다음 Explore on my own(직접 탐색)을 선택합니다.
2. 개발 도구(Dev Tools)로 이동하여 기본 검색을 실행합니다.

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

권한 오류를 확인합니다. `limited-user`에는 클러스터 전체 검색을 실행할 권한이 없습니다.

3. 또 다른 검색을 실행합니다.

```
GET opensearch_dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

일치하는 모든 문서에 `true` 값을 갖는 `FlightDelay` 필드와 익명화된 `Dest` 필드가 있으며 `FlightNum` 필드는 없는 것을 확인할 수 있습니다.

4. `master-user`로 로그인한 원래 브라우저 창에서 개발 도구(Dev Tools)를 선택한 다음 동일한 검색을 수행합니다. 권한, 결과 수, 일치하는 문서 및 포함된 필드의 차이를 확인합니다.

자습서: 내부 사용자 데이터베이스와 HTTP 기본 인증을 사용하여 도메인 구성

이 자습서에서는 내부 사용자 데이터베이스의 마스터 사용자와 OpenSearch 대시보드에 대한 HTTP 기본 인증이라는 또 다른 일반적인 세분화된 액세스 제어 사용 사례를 다룹니다. 그러면 마스터 사용자는 OpenSearch Dashboards에 로그인하여 내부 사용자를 생상하고 이 사용자를 역할에 매핑하고 세분화된 액세스 제어를 사용하여 사용자의 권한을 제한할 수 있습니다.

이 튜토리얼에서는 다음 단계를 완료합니다.

1. 마스터 사용자로 도메인 생성하기
2. OpenSearch Dashboards에서 내부 사용자 구성하기
3. OpenSearch 대시보드에서 역할 매핑
4. 권한 테스트

1단계: 도메인 생성

https://console.aws.amazon.com/aos/home/ Amazon OpenSearch Service 콘솔로 이동하여 다음 설정을 사용하여 도메인을 생성합니다.

- OpenSearch 1.0 이상 또는 Elasticsearch 7.9 이상

- 공개 액세스(Public access)
- 내부 사용자 데이터베이스의 마스터 사용자(이 자습서의 나머지 부분에서 `TheMasterUser`로 지칭)와 세분화된 액세스 제어
- Dashboards에 대한 Amazon Cognito 인증 비활성화
- 다음과 같은 액세스 정책:

  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}/*"
    }
  ]
}
```

- 도메인에 대한 모든 트래픽에 HTTPS 필요
- 노드 간 암호화
- 저장 데이터의 암호화

2다녜: OpenSearch Dashboards에서 내부 사용자 구성하기

이제 도메인이 생겼으므로 OpenSearch Dashboards에 로그인하여 내부 사용자를 생성할 수 있습니다.

1. OpenSearch Service 콘솔로 돌아가서 생성한 도메인의 OpenSearch 대시보드 URL로 이동합니다. URL은 *domain-endpoint*`/_dashboards/` 형식입니다.
2. `TheMasterUser`(으)로 로그인합니다.
3. Add sample data(샘플 데이터 추가)를 선택하고 샘플 비행 데이터를 추가합니다.
4. 왼쪽 탐색 창에서 보안, 내부 사용자, 내부 사용자 생성을 선택합니다.

5. 사용자의 이름을 new-user로 지정하고 암호를 지정합니다. 그런 다음 생성(Create)을 선택합니다.

3단계: OpenSearch Dashboards에서 역할 매핑

이제 사용자가 구성되었으므로 사용자를 역할에 매핑할 수 있습니다.

1. OpenSearch Dashboards의 보안 섹션에서 역할, 역할 생성을 선택하세요.
2. 역할 이름을 new-role로 지정합니다.
3. 인덱스 권한의 경우 인덱스 패턴에 opensearch_dashboards_sample_data_fli*(을)를 지정합니다(Elasticsearch 도메인의 kibana_sample_data_fli*인 경우).
4. 작업 그룹에 대해 읽기(read)를 선택합니다.
5. 문서 수준 보안 쿼리(Document Level Security Query)에 다음 쿼리를 지정합니다.

```
{
  "match": {
    "FlightDelay": true
  }
}
```

6. 필드 수준 보안의 경우 제외(Exclude)를 선택하고 FlightNum을 지정합니다.
7. 익명화(Anonymization)에 Dest를 지정합니다.
8. 생성(Create)을 선택합니다.
9. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다. 그런 다음 new-user를 사용자(Users)에 추가하고 맵(Map)을 선택합니다.
10. 역할 목록으로 돌아가서 opensearch_dashboards_user를 선택합니다. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다. 그런 다음 new-user를 사용자(Users)에 추가하고 맵(Map)을 선택합니다.

4단계: 권한 테스트

역할이 올바르게 매핑되면 제한된 사용자로 로그인하고 권한을 테스트할 수 있습니다.

1. 새로운 프라이빗 브라우저 창에서 도메인의 OpenSearch 대시보드 URL로 이동하고 new-user 보안 인증을 사용하여 로그인한 다음 Explore on my own(직접 탐색)을 선택합니다.
2. 개발 도구(Dev Tools)로 이동하여 기본 검색을 실행합니다.

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

권한 오류를 확인합니다. new-user에는 클러스터 전체 검색을 실행할 권한이 없습니다.

3. 또 다른 검색을 실행합니다.

```
GET dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

일치하는 모든 문서에 true 값을 갖는 FlightDelay 필드와 익명화된 Dest 필드가 있으며 FlightNum 필드는 없는 것을 확인할 수 있습니다.

4. TheMasterUser로 로그인한 원래 브라우저 창에서 개발 도구(Dev Tools)를 선택한 다음 동일한 검색을 수행합니다. 권한, 결과 수, 일치하는 문서 및 포함된 필드의 차이를 확인합니다.

Amazon OpenSearch Service에 대한 규정 준수 확인

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon OpenSearch Service의 보안 및 규정 준수를 평가합니다. 이 프로그램에는 SOC, PCI 및 HIPAA가 포함됩니다.

규정 준수 요구 사항이 있는 경우 OpenSearch의 모든 버전 또는 Elasticsearch 6.0 이상을 사용하는 것이 좋습니다. 이전 버전의 Elasticsearch는 저장된 데이터 암호화 및 노드 간 암호화 조합을 제공하지 않으며, 필요에 맞게 조정할 수 없습니다. 또한 세분화된 액세스 제어가 사용 사례에 중요하다면 OpenSearch의 모든 버전 또는 Elasticsearch 6.7 이상 버전의 사용을 고려할 수도 있습니다. 그러나 도메인을 만들 때 특정 OpenSearch 또는 Elasticsearch 버전을 선택한다고 해서 규정 준수가 보장되는 것은 아닙니다.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 제공 범위 섹션을 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 에서 보고서 다운로드 AWS Artifact에서 .

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다.는 규정 준수를 지원하기 위해 다음 리소스를 AWS 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- HIPAA 적격 서비스 참조 - HIPAA 적격 서비스가 나열되어 있습니다. 모두 HIPAA 자격이 AWS 서비스 있는 것은 아닙니다.
- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 산업 및 위치에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에는 여러 프레임워크(미국 국립표준기술연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제표준화기구(ISO) 포함)의 보안 제어에 대한 지침을 보호하고 AWS 서비스 매핑하는 모범 사례가 요약되어 있습니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 -이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub - 내 보안 상태에 대한 포괄적인 보기를 AWS 서비스 제공합니다 AWS. Security Hub는 보안 컨트롤을 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 의심스러운 활동 및 악의적인 활동이 있는지 환경을 모니터링하여 사용자, AWS 계정워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 따르는 데 도움을 줄 수 있습니다.
- AWS Audit Manager - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험과 규정 및 업계 표준 준수를 관리하는 방법을 간소화할 수 있습니다.

Amazon OpenSearch Service의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공하며,이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결됩니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

AWS 글로벌 인프라 외에도 OpenSearch Service는 데이터 복원력 및 백업 요구 사항을 지원하는 몇 가지 기능을 제공합니다.

- 다중 AZ 도메인 및 복제본 샤드
- 자동 및 수동 스냅샷

Amazon OpenSearch Service에 대한 JWT 인증 및 권한 부여

이제 Amazon OpenSearch Service에서 인증 및 권한 부여에 JSON 웹 토큰(JWT)을 사용할 수 있습니다. JWT는 Single Sign-On(SSO) 액세스 권한을 부여하는 데 사용되는 JSON 기반 액세스 토큰입니다. OpenSearch Service에서 JWT를 사용하여 Single Sign-On 토큰을 생성해 OpenSearch Service 도메인에 대한 요청을 검증할 수 있습니다. JWT를 사용하려면 세분화된 액세스 제어가 활성화되어 있어야 하며 유효한 RSA 또는 ECDSA PEM 형식의 퍼블릭 키를 제공해야 합니다. 세분화된 액세스 제어에 대한 자세한 내용은 Amazon OpenSearch Service에서 세분화된 액세스 제어를 참조하세요.

OpenSearch Service 콘솔, AWS Command Line Interface (AWS CLI) 또는 AWS SDKs.

고려 사항

Amazon OpenSearch Service에서 JWT를 사용하기 전에 다음을 고려해야 합니다.

- PEM 형식의 RSA 퍼블릭 키 크기로 인해 AWS 콘솔을 사용하여 JWT 인증 및 권한 부여를 구성하는 것이 좋습니다.
- JWT에 대한 제목 및 역할 필드를 지정할 때 유효한 사용자 및 역할을 제공해야 합니다. 그렇지 않으면 요청이 거부됩니다.
- OpenSearch 2.11은 JWT 인증에 사용할 수 있는 가장 빠른 호환 버전입니다.

도메인 액세스 정책 수정

JWT 인증 및 권한 부여를 사용하도록 도메인을 구성하기 전에 JWT 사용자가 도메인에 액세스할 수 있도록 도메인 액세스 정책을 업데이트해야 합니다. 그렇지 않으면 수신되는 모든 JWT 권한 부여된 요청이 거부됩니다. 하위 리소스(/*)에 대한 전체 액세스를 제공하기 위해 권장되는 도메인 액세스 정책은 다음과 같습니다.

JSON

```
{
```

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

JWT 인증 및 권한 부여 구성

도메인 생성 프로세스 중에 또는 기존 도메인을 업데이트하여 JWT 인증 및 권한 부여를 활성화할 수 있습니다. 설정 단계는 선택 사항에 따라 약간 다릅니다.

다음 단계에서는 OpenSearch Service 콘솔에서 JWT 인증 및 권한 부여를 위해 기존 도메인을 구성하는 방법을 설명합니다.

1. 도메인 구성에서 OpenSearch에 대한 JWT 인증 및 권한 부여로 이동하고 JWT 인증 및 권한 부여 활성화를 선택합니다.
2. 도메인에 대해 사용할 퍼블릭 키를 구성합니다. 이렇게 하려면 퍼블릭 키가 포함된 PEM 파일을 업로드하거나 수동으로 입력할 수 있습니다.

   > **Note**
   >
   > 업로드되거나 입력된 키가 유효하지 않으면 문제를 지정하는 경고가 텍스트 상자 위에 표시됩니다.

3. (선택 사항) 추가 설정에서 다음 선택적 필드를 구성할 수 있습니다.

   - 제목 키 - 이 필드를 비워 두고 JWT에 대한 기본 `sub` 키를 사용할 수 있습니다.
   - 역할 키 - 이 필드를 비워 두고 JWT에 대한 기본 `roles` 키를 사용할 수 있습니다.

   변경한 후에 도메인을 저장합니다.

JWT를 사용하여 테스트 요청 전송

지정된 주제 및 역할 페어로 새 JWT를 생성한 후 테스트 요청을 전송할 수 있습니다. 이를 수행하려면 프라이빗 키를 사용하여 JWT를 생성한 도구를 통해 요청에 서명합니다. OpenSearch Service는 이 서명을 확인하여 수신 요청을 검증할 수 있습니다.

> **Note**
>
> JWT에 사용자 지정 제목 키 또는 역할 키를 지정한 경우 JWT에 올바른 클레임 이름을 사용해야 합니다.

다음은 JWT 토큰을 사용하여 도메인의 검색 엔드포인트를 통해 OpenSearch Service에 액세스하는 방법의 예제입니다.

```
curl -XGET "$search_endpoint" -H "Authorization: Bearer <JWT>"
```

JWT 인증 및 권한 부여 구성(AWS CLI)

다음 AWS CLI 명령은 도메인이 있는 경우 OpenSearch에 대한 JWT 인증 및 권한 부여를 활성화합니다.

```
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>"}}'
```

JWT 인증 및 권한 부여 구성(API를 통한 구성)

구성 API에 대한 다음 요청은 기존 도메인에서 OpenSearch 대시보드에 대한 JWT 인증 및 권한 부여를 활성화합니다.

```
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"
```

```
    }
  }
}
```

키 페어 생성

OpenSearch 도메인에 대한 JWT를 구성하려면 개인정보 보호 강화 메일(PEM) 형식으로 퍼블릭 키를 제공해야 합니다. Amazon OpenSearch Service는 현재 JWT를 사용하는 두 가지 비대칭 암호화 알고리즘(RSA 및 ECDSA)을 지원합니다.

공통 openssl 라이브러리를 사용하여 RSA 키 페어를 생성하려면 다음 단계를 수행합니다.

1. `openssl genrsa -out privatekey.pem 2048`
2. `openssl rsa -in privatekey.pem -pubout -out publickey.pem`

이 예제에서는 `publickey.pem` 파일에 Amazon OpenSearch Service에서 사용할 퍼블릭 키가 포함된 반면, `privatekey.pem`에는 서비스로 전송된 JWT에 서명하는 프라이빗 키가 포함되어 있습니다. 또한 JWT를 생성하는 데 필요한 경우 프라이빗 키를 일반적으로 사용되는 `pkcs8` 형식으로 변환하는 옵션이 있습니다.

업로드 버튼을 사용하여 콘솔에 PEM 파일을 직접 추가하는 경우 파일 확장자는 `.pem`이어야 합니다. `.crt`, `.cert`, `.key`와 같은 다른 파일 확장자는 지원되지 않습니다.

Amazon OpenSearch Service의 인프라 보안

관리형 서비스인 Amazon OpenSearch Service는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및 가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 OpenSearch Service에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 위탁자와 관련된 보안 암호 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 자격 증명을 생성하여 요청에 서명할 수 있습니다.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 OpenSearch Service 구성 API에 액세스합니다. 허용할 수 있는 최소 필수 TLS 버전을 구성하려면 도메인 엔드포인트 옵션에서 `TLSSecurityPolicy` 값을 지정합니다.

```
aws opensearch update-domain-config --domain-name my-domain --domain-endpoint-options
 '{"TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"}'
```

자세한 내용은 AWS CLI 명령 참조를 확인하세요.

도메인 구성에 따라 OpenSearch API에 대한 요청에 서명해야 할 수도 있습니다. 자세한 내용은 the section called “OpenSearch Service 요청 작성 및 서명” 섹션을 참조하세요.

OpenSearch Service에서는 인터넷 연결 디바이스에서 요청을 수신할 수 있는 퍼블릭 액세스 도메인과 퍼블릭 인터넷에서 격리된 VPC 액세스 도메인을 지원합니다.

OpenSearch Service 관리형 VPC 엔드포인트를 사용하여 Amazon OpenSearch Service에 액세스(AWS PrivateLink)

OpenSearch Service 관리형 VPC 엔드포인트(제공)를 설정하여 Amazon OpenSearch Service 도메인에 액세스할 수 있습니다 AWS PrivateLink. 이러한 엔드포인트는 VPC와 Amazon OpenSearch Service 간의 프라이빗 연결을 생성합니다. 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결을 사용하지 않고 VPC에 있는 것처럼 OpenSearch Service VPC 도메인에 액세스할 수 있습니다. VPC의 인스턴스는 OpenSearch Service에 액세스하기 위해 퍼블릭 IP 주소가 필요하지 않습니다.

동일한 VPC, 다른 VPC 또는 다른 AWS 계정내의 퍼블릭 또는 프라이빗 서브넷에서 실행되는 추가 엔드포인트를 노출하도록 OpenSearch Service 도메인을 구성할 수 있습니다. 따라서 인프라를 관리할 필요가 없고 도메인이 실행되는 위치와도 관계없이 도메인에 액세스할 수 있도록 추가 보안 계층을 추가할 수 있습니다. 다음 다이어그램은 동일한 VPC 내의 OpenSearch Service 관리형 VPC 엔드포인트를 보여줍니다.

AWS PrivateLink기반 OpenSearch Service 관리형 인터페이스 VPC 엔드포인트를 생성하여 이 프라이빗 연결을 설정합니다. 인터페이스 VPC 엔드포인트에 대해 활성화하는 각 서브넷에 엔드포인트 네트

워크 인터페이스를 생성합니다. 이는 OpenSearch Service로 향하는 트래픽의 진입점 역할을 하는 서비스 관리형 네트워크 인터페이스입니다. 표준 AWS PrivateLink 인터페이스 엔드포인트 요금은에 따라 청구되는 OpenSearch Service 관리형 VPC 엔드포인트에 적용됩니다 AWS PrivateLink.

모든 버전의 OpenSearch 및 레거시 Elasticsearch를 실행하는 도메인에 대해 VPC 엔드포인트를 생성할 수 있습니다. 자세한 내용은AWS PrivateLink 가이드의 AWS PrivateLink를 통해 AWS 서비스 에 액세스를 참조하세요.

OpenSearch Service에 대한 고려 사항 및 제한

OpenSearch Service에 대한 인터페이스 VPC 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의 인터페이스 VPC 엔드포인트를 사용하여 AWS 서비스 액세스를 검토하세요.

OpenSearch Service 관리형 VPC 엔드포인트를 사용할 때 다음을 고려해야 합니다.

- 인터페이스 VPC 엔드포인트만 사용하여 VPC 도메인에 연결할 수 있습니다. 퍼블릭 도메인은 지원되지 않습니다.
- VPC 엔드포인트는 동일한 AWS 리전에 속한 도메인에만 연결할 수 있습니다.
- HTTPS는 VPC 엔드포인트를 지원하는 유일한 프로토콜입니다. HTTP는 허용되지 않습니다.
- OpenSearch Service는 인터페이스 VPC 엔드포인트를 통해 지원되는 OpenSearch API 작업 모두에 대한 호출을 지원합니다.
- 계정당 최대 50개의 엔드포인트와 도메인당 최대 10개의 엔드포인트를 구성할 수 있습니다. 단일 도메인은 최대 10개의 인증된 보안 주체를 보유할 수 있습니다.
- 현재를 사용하여 인터페이스 VPC 엔드포인트 AWS CloudFormation 를 생성할 수 없습니다.
- OpenSearch Service 콘솔이나 OpenSearch Service API를 통해서만 인터페이스 VPC 엔드포인트를 생성할 수 있습니다. Amazon VPC 콘솔을 사용하여 OpenSearch Service에 대한 인터페이스 VPC 엔드포인트를 생성할 수 없습니다.
- OpenSearch Service 관리형 VPC 엔드포인트는 인터넷에서 액세스할 수 없습니다. OpenSearch Service 관리형 VPC 엔드포인트는 엔드포인트가 프로비저닝된 VPC 또는 라우팅 테이블 및 보안 그룹에서 허용하는 대로 엔드포인트가 프로비저닝된 VPC와 피어링된 모든 VPC 내에서만 액세스할 수 있습니다.
- OpenSearch Service에는 VPC 엔드포인트 정책이 지원되지 않습니다. 보안 그룹을 엔드포인트 네트워크 인터페이스와 연결하여 인터페이스 VPC 엔드포인트를 통해 OpenSearch Service에 대한 트래픽을 제어할 수 있습니다.
- 서비스 연결 역할은 VPC 엔드포인트를 생성하는 데 사용하는 것과 동일한 AWS 계정에 있어야 합니다.

- OpenSearch Service VPC 엔드포인트를 생성, 업데이트 및 삭제하려면 Amazon OpenSearch Service 권한 외에도 다음과 같은 Amazon EC2 권한이 있어야 합니다.
  - `ec2:CreateVpcEndpoint`
  - `ec2:DescribeVpcEndpoints`
  - `ec2:ModifyVpcEndpoint`
  - `ec2:DeleteVpcEndpoints`
  - `ec2:CreateTags`
  - `ec2:DescribeTags`
  - `ec2:DescribeSubnets`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeVpcs`

> **Note**
>
> 현재는 VPC 엔드포인트 생성을 OpenSearch Service로 제한할 수 없습니다. 향후 업데이트에서 이를 가능하게 하기 위해 노력하고 있습니다.

도메인에 대한 액세스 제공

도메인에 액세스하려는 VPC가 다른에 있는 경우 인터페이스 VPC 엔드포인트를 생성하기 전에 소유자의 계정에서 권한을 부여 AWS 계정해야 합니다.

다른의 VPC가 도메인에 액세스 AWS 계정 하도록 허용하려면

1. https://console.aws.amazon.com/aos/home/ Amazon OpenSearch Service 콘솔을 엽니다.
2. 탐색 창에서 Domains(도메인)를 선택하고 액세스를 제공하려는 도메인을 엽니다.
3. 도메인에 액세스할 수 있는 계정 및 해당 VPC를 표시하는 VPC endpoints(VPC 엔드포인트) 탭으로 이동합니다.
4. Authorize principal(보안 주체 인증)을 선택합니다.
5. 도메인에 액세스할 계정의 AWS 계정 ID를 입력합니다. 이 단계는 도메인에 대해 VPC 엔드포인트를 생성하도록 지정된 계정에 권한을 부여합니다.
6. Authorize를 선택합니다.

VPC 도메인에 대한 인터페이스 VPC 엔드포인트 생성

OpenSearch Service 콘솔 또는 AWS Command Line Interface ()를 사용하여 OpenSearch Service용 인터페이스 VPC 엔드포인트를 생성할 수 있습니다AWS CLI.

OpenSearch Service 도메인에 대한 인터페이스 VPC 엔드포인트 생성하기

1. https://console.aws.amazon.com/aos/home/ Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 VPC endpoints(VPC 엔드포인트)를 선택합니다.
3. Create endpoint(엔드포인트 생성)을 선택합니다.
4. 현재의 도메인을 연결할지 AWS 계정 아니면 다른의 도메인을 연결할지 선택합니다 AWS 계정.
5. 이 엔드포인트로 연결할 도메인을 선택합니다. 도메인이 현재에 있는 경우 드롭다운을 AWS 계정 사용하여 도메인을 선택합니다. 도메인이 다른 계정에 있는 경우 연결할 도메인의 Amazon 리소스 이름(ARN)을 입력합니다. 다른 계정에서 도메인을 선택하려면 소유자가 도메인에 대한 액세스 권한을 제공해야 합니다.
6. VPC의 경우 OpenSearch Service에 액세스할 VPC를 선택합니다.
7. Subnets(서브넷)의 경우 OpenSearch Service에 액세스할 서브넷을 하나 이상 선택합니다.
8. Security group(보안 그룹)의 경우 엔드포인트 네트워크 인터페이스에 연결할 보안 그룹을 선택합니다. 이는 엔드포인트에 대해 권한을 부여하는 인바운드 트래픽의 포트, 프로토콜, 소스를 제한하는 중요한 단계입니다. 보안 그룹 규칙에서 VPC 엔드포인트를 사용하여 OpenSearch Service와 통신하는 리소스가 엔드포인트 네트워크 인터페이스와 통신할 수 있도록 허용해야 합니다.
9. Create endpoint(엔드포인트 생성)을 선택합니다. 엔드포인트는 2~5분 내에 활성화되어야 합니다.

구성 API를 사용하여 OpenSearch Service 관리형 VPC 엔드포인트 작업

다음 API 작업을 사용하여 OpenSearch Service 관리형 VPC 엔드포인트를 생성하고 관리합니다.

- CreateVpcEndpoint
- ListVpcEndpoints
- UpdateVpcEndpoint
- DeleteVpcEndpoint

다음 API 작업을 사용하여 VPC 도메인에 대한 엔드포인트 액세스를 관리합니다.

- AuthorizeVpcEndpointAccess

- ListVpcEndpointAccess
- ListVpcEndpointsForDomain
- RevokeVpcEndpointAccess

OpenSearch 대시보드에 대한 SAML 인증

OpenSearch 대시보드에 대한 SAML 인증을 사용하면 기존 자격 증명 공급자를 사용하여 OpenSearch 또는 Elasticsearch 6.7 이상을 실행하는 Amazon OpenSearch Service 도메인의 대시보드에 통합 인증(SSO)을 제공할 수 있습니다. SAML 인증을 사용하려면 세분화된 액세스 제어를 활성화해야 합니다.

Amazon Cognito 또는 내부 사용자 데이터베이스를 통해 인증하는 대신 OpenSearch 대시보드에 대한 SAML 인증을 사용하면 타사 자격 증명 공급자를 사용하여 대시보드에 로그인하고 세분화된 액세스 제어를 관리하고 데이터를 검색하고 시각화를 구축할 수 있습니다. OpenSearch Service는 Okta, Keycloak, Active Directory Federation Services(ADFS), Auth0, 등 SAML 2.0 표준을 사용하는 공급자를 지원합니다 AWS IAM Identity Center.

대시보드에 대한 SAML 인증은 웹 브라우저를 통해 OpenSearch 대시보드에 액세스하는 용도로만 사용됩니다. SAML 자격 증명을 사용하면 OpenSearch 또는 Dashboards API에 직접 HTTP 요청을 할 수 없습니다.

SAML 구성 개요

이 설명서에서는 기존 ID 제공업체가 있고 어느 정도 익숙하다고 가정합니다. OpenSearch Service 도메인의 경우에만 정확한 공급자에 대한 자세한 구성 단계를 제공할 수 없습니다.

OpenSearch 대시보드 로그인 흐름은 다음 두 가지 형식 중 하나를 취할 수 있습니다.

- 서비스 공급자(SP)가 시작됨: Dashboards로 이동하면(예: `https://my-domain.us-east-1.es.amazonaws.com/_dashboards`), 로그인 화면으로 리디렉션됩니다. 로그인하면 자격 증명 공급자가 사용자를 Dashboards로 리디렉션합니다.
- ID 제공업체(idP)가 시작됨: 자격 증명 공급자로 이동하여 로그인한 다음 애플리케이션 디렉터리에서 OpenSearch 대시보드를 선택합니다.

OpenSearch Service는 SP가 시작한 URL과 IdP가 시작한 두 개의 통합 인증(SSO) URL을 제공하지만 원하는 OpenSearch 대시보드 로그인 흐름과 일치하는 URL만 있으면 됩니다.

사용하는 인증 유형과 관계없이 자격 증명 공급자를 통해 로그인하고 사용자 이름(필수) 및 백엔드 역할(선택 사항이지만 권장함)을 포함한 SAML 어설션을 받는 것이 목적입니다. 이 정보를 통해 세분화된

액세스 제어로 SAML 사용자에게 권한을 할당할 수 있습니다. 외부 자격 증명 공급자의 백엔드 역할은 일반적으로 “역할” 또는 “그룹”이라고 합니다.

고려 사항

SAML 인증을 구성할 때 다음 사항을 고려하세요.

- IdP 메타데이터 파일의 크기로 인해 AWS 콘솔을 사용하여 SAML 인증을 구성하는 것이 좋습니다.
- 도메인은 한 번에 하나의 Dashboards 인증 방법만 지원합니다. OpenSearch 대시보드에 대한 Amazon Cognito 인증을 활성화한 경우 SAML 인증을 활성화하기 전에 이 기능을 먼저 비활성화해야 합니다.
- SAML과 함께 Network Load Balancer를 사용하는 경우 먼저 사용자 지정 엔드포인트를 생성해야 합니다. 자세한 내용은 ??? 단원을 참조하십시오.
- 비 IAM 자격 증명(예: Amazon OpenSearch Serverless 및 SAML의 SAML 및 Amazon OpenSearch Service의 기본 내부 사용자 권한 부여)의 경우 서비스 제어 정책(SCP)이 적용되거나 평가되지 않습니다.

VPC 도메인에 대한 SAML 인증

SAML은 ID 제공업체와 서비스 제공업체 간에 직접 통신이 필요하지 않습니다. 따라서 OpenSearch 도메인이 프라이빗 VPC 내에서 호스팅되는 경우에도 브라우저가 OpenSearch 클러스터 및 자격 증명 공급자 모두와 통신할 수 있는 한 SAML을 계속 사용할 수 있습니다. 브라우저는 본질적으로 자격 증명 공급자와 서비스 공급자 간의 중개자 역할을 수행합니다. SAML 인증 흐름을 설명하는 유용한 다이어그램은 Okta 설명서를 참조하세요.

도메인 액세스 정책 수정

SAML 인증을 구성하기 전에 SAML 사용자가 도메인에 액세스할 수 있도록 도메인 액세스 정책을 업데이트해야 합니다. 그렇지 않으면 액세스 거부 오류가 표시됩니다.

도메인의 하위 리소스(/*)에 대한 전체 액세스를 제공하는 다음 도메인 액세스 정책을 권장합니다.

JSON

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

```
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-east-1:111122223333:domain/domain-name/*"
    }
  ]
}
```

정책을 더 제한적으로 만들기 위해 정책에 IP 주소 조건을 추가할 수 있습니다. 이 조건은 지정된 IP 주소 범위 또는 서브넷으로만 액세스를 제한합니다. 예를 들어 다음 정책은 192.0.2.0/24 서브넷에서만 액세스를 허용합니다.

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**
>
> 개방형 도메인 액세스 정책을 사용하려면 도메인에서 세분화된 액세스 제어를 활성화해야 합니다. 그렇지 않으면 다음 오류가 표시됩니다.
>
> `To protect domains with public access, a restrictive policy or fine-grained access control is required.`
>
> 강력한 암호로 구성된 마스터 사용자 또는 내부 사용자가 있는 경우 세분화된 액세스 제어를 사용하는 동안 개방형 정책으로 유지해도 보안 관점에서 허용될 수 있습니다. 자세한 내용은 ??? 단원을 참조하십시오.

SP 및 IdP 시작 인증 구성

이 단계에서는 OpenSearch 대시보드에 대해 SP 시작 또는 IdP 시작 인증으로 SAML 인증을 사용 설정하는 방법을 설명합니다. 둘 다 활성화하는 데 필요한 추가 단계는 SP 및 IdP 시작 인증 모두 구성을 참조하세요.

1단계: SAML 인증 활성화

도메인 생성 중에 또는 기존 도메인에서 Actions(작업), Edit security configuration(보안 구성 편집)을 선택하여 SAML 인증을 활성화할 수 있습니다. 다음 단계는 선택 사항에 따라 약간 다릅니다.

도메인 구성 내의 SAML authentication for OpenSearch 대시보드/Kibana(OpenSearch 대시보드/Kibana에 대한 SAML 인증) 아래에서 Enable SAML authentication(SAML 인증 활성화)을 선택합니다.

2단계: ID 제공업체 구성

SAML 인증을 구성하는 시기에 따라 다음 단계를 수행하세요.

새 도메인을 생성하는 경우

새 도메인을 생성하는 중이라면 OpenSearch Service는 아직 서비스 제공업체 엔터티 ID 또는 SSO URL을 생성할 수 없습니다. ID 제공업체에서 SAML 인증을 제대로 활성화하려면 이러한 값이 필요하지만 도메인이 생성된 후에만 해당 값을 생성할 수 있습니다. 도메인을 생성하는 동안 이러한 상호 종속성을 해결하기 위해 IdP 구성에 임시 값을 제공하여 필요한 메타데이터를 생성한 다음 도메인이 활성화되면 업데이트할 수 있습니다.

사용자 지정 엔드포인트를 사용하는 경우 URL이 무엇인지 유추할 수 있습니다. 예를 들어 사용자 지정 엔드포인트가 `www.custom-endpoint.com`인 경우 서비스 제공업체 엔터티 ID

는 www.*custom-endpoint*.com, IdP 시작 SSO URL은 www.*custom-endpoint*.com/_dashboards/_opendistro/_security/saml/acs/idpinitiated, SP 시작 SSO URL은 www.*custom-endpoint*.com/_dashboards/_opendistro/_security/saml/acs이 됩니다. 도메인이 생성되기 전에 값을 사용하여 ID 제공업체를 구성할 수 있습니다. 예는 다음 단원을 참조하십시오.

> **Note**
>
> HTTP 요청의 FQDN이 SAML 요청의 FQDN과 다르므로 이중 스택 엔드포인트로 로그인할 수 없습니다. 이중 스택 엔드포인트를 사용하여 로그인하려는 경우 OpenSearch 관리자가 사용자 지정 엔드포인트를 설정하고 CNAME 값을 이중 스택 엔드포인트로 설정해야 합니다.

사용자 지정 엔드포인트를 사용하지 않는 경우 IdP에 임시 값을 입력하여 필요한 메타데이터를 생성한 다음 나중에 도메인이 활성화된 후 업데이트할 수 있습니다.

예를 들어 Okta 내에서 Single sign on URL(Single Sign-On URL) 필드와 Audience URI (SP Entity ID)(대상 URI(SP 엔터티 ID)) 필드에 https://*temp-endpoint*.amazonaws.com을 입력하면 메타데이터를 생성할 수 있습니다. 그런 다음 도메인이 활성화되면 올바른 값을 OpenSearch Service에서 검색하고 Okta에서 업데이트할 수 있습니다. 지침은 the section called “6단계: IdP URL 업데이트” 섹션을 참조하세요.

기존 도메인을 편집하는 경우

기존 도메인에서 SAML 인증을 활성화하는 경우 서비스 제공업체 엔터티 ID와 SSO URL 중 하나를 복사합니다. 사용할 URL에 대한 지침은 the section called “SAML 구성 개요”을 참조하세요.

값을 사용하여 ID 제공업체를 구성합니다. 이것이 프로세스의 가장 복잡한 부분이며 안타깝게도 용어와 단계는 공급자에 따라 크게 다릅니다. 공급자의 설명서를 참조하세요.

예를 들어 Okta에서는 SAML 2.0 웹 애플리케이션을 생성합니다. Single sign on URL(Single Sign-On URL)에 SSO URL을 지정합니다. 대상 Audience(SP 엔터티 ID)(Audience URI(SP Entity ID))에 SP 엔티티 ID를 지정합니다.

Okta에는 사용자 및 백엔드 역할이 아니라 사용자와 그룹이 있습니다. Group Attribute Statements(그룹 속성 문)의 경우 Name(이름) 필드에 role을 추가하고 Filter(필터) 필드에 정규식 .+를 추가하는 것이 좋습니다. 이 문은 Okta 자격 증명 공급자에 사용자가 인증한 후 SAML 어설션의 role 필드에서 모든 사용자 그룹을 포함하도록 지시합니다.

IAM ID 센터에서 SP 엔티티 ID를 Application SAML 대상으로 지정합니다. 또한 다음과 같은 속성 매핑 `Subject=${user:subject}:format=unspecified` 및 `Role=${user:groups}:format=uri`을 지정해야 합니다.

Auth0에서는 일반 웹 애플리케이션을 생성하고 SAML 2.0 추가 기능을 활성화합니다. Keycloak에서는 클라이언트를 생성합니다.

3단계: IdP 메타데이터 가져오기

자격 증명 공급자를 구성하면 IdP 메타데이터 파일이 생성됩니다. 이 XML 파일에는 TLS 인증서, 통합 인증 엔드포인트 및 자격 증명 공급자의 엔터티 ID와 같은 공급자에 대한 정보가 들어 있습니다.

IdP 메타데이터 파일의 내용을 복사하여 OpenSearch Service 콘솔의 IdP 메타데이터 필드에 붙여넣습니다. 또는 XML 파일에서 가져오기(Import from XML file)를 선택하고 파일을 업로드합니다. 메타데이터 파일은 다음과 같아야 합니다.

```
<?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>
```

4단계: SAML 필드 구성

IdP 메타데이터를 입력한 후 OpenSearch Service 콘솔 내에서 다음 추가 필드를 구성합니다.

- IdP entity ID(IdP 엔터티 ID) – 메타데이터 파일에서 `entityID` 속성 값을 복사하여 이 필드에 붙여 넣습니다. 또한 많은 자격 증명 공급자는 사후 구성 요약의 일부로 이 값을 표시합니다. 일부 공급자는 이를 "발행자"라고 부릅니다.
- SAML master username(SAML 마스터 사용자 이름) 및 SAML master backend role(SAML 마스터 백엔드 역할) – 지정한 사용자 및/또는 백엔드 역할은 새 마스터 사용자와 동등한 클러스터에 대한 전체 권한을 받지만 OpenSearch 대시보드 내에서만 해당 권한을 사용할 수 있습니다.

  예를 들어 Okta에는 `admins` 그룹에 속한 사용자 `jdoe`가 있을 수 있습니다. `jdoe`를 SAML 마스터 사용자 이름 필드에서 추가할 경우, 해당 사용자만 모든 권한을 받습니다. `admins`를 SAML 마스터 백엔드 역할 필드에 추가할 경우, `admins` 그룹에 속한 모든 사용자가 모든 권한을 받습니다.

> **Note**
>
> SAML 어설션의 내용은 SAML 마스터 사용자 이름 및 SAML 마스터 역할에 사용하는 문자열과 정확히 일치해야 합니다. 일부 자격 증명 공급자는 사용자 이름 앞에 접두사를 추가하여 진단하기 어려운 불일치가 발생할 수 있습니다. 자격 증명 공급자 사용자 인터페이스에 `jdoe`가 보일 수 있지만 SAML 어설션은 `auth0|jdoe`를 포함할 수 있습니다. SAML 어설션에서 항상 문자열을 사용합니다.

많은 자격 증명 공급자를 사용하면 구성 프로세스 중에 샘플 어설션을 볼 수 있으며 SAML 추적기와 같은 도구는 실제 어설션의 내용을 검사하고 문제를 해결하는 데 도움이 될 수 있습니다. 어설션은 다음과 같습니다.

```
<?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>
```

5단계: (선택 사항) 추가 설정 구성

Additional settings(추가 설정)에서 다음 선택적 필드를 구성합니다.

- Subject key(제목 키) - 사용자 이름에 대한 SAML 어설션의 `NameID` 요소를 사용하려면 이 필드를 비워 둡니다. 어설션에서 이 표준 요소를 사용하지 않고 사용자 이름을 사용자 지정 속성으로 포함하는 경우 여기에 해당 속성을 지정합니다.
- Roles key(역할 키) - 백엔드 역할(권장)을 사용하려는 경우 이 필드에 어설션의 속성을 지정합니다(예: `role` 또는 `group`). 이것은 SAML 추적기와 같은 도구가 도움이 될 수 있는 또 다른 상황입니다.
- Session time to live(세션 TTL(Time To Live)) - 기본적으로 OpenSearch 대시보드는 24시간 후에 사용자를 로그아웃합니다. 새 값을 지정하여 이 값을 60에서 1,440(24시간) 사이의 임의의 숫자로 구성할 수 있습니다.

구성에 만족하면 도메인을 저장합니다.

6단계: IdP URL 업데이트

도메인을 생성하는 동안 SAML 인증을 활성화한 경우 XML 메타데이터 파일을 생성하기 위해 IdP 내에서 임시 URL을 지정해야 했습니다. 도메인 상태가 `Active`로 변경되면 올바른 URL을 가져오고 IdP를 수정할 수 있습니다.

URL을 검색하려면 도메인을 선택하고 Actions(작업), Edit security configuration(보안 구성 편집)을 선택합니다. SAML authentication for OpenSearch 대시보드/Kibana(OpenSearch 대시보드/Kibana에 대한 SAML 인증)에서 올바른 서비스 제공업체 엔터티 ID와 SSO URL을 찾을 수 있습니다. 값을 복사하고 이를 사용하여 ID 제공업체를 구성하고 2단계에서 제공한 임시 URL을 바꿉니다.

7단계: 역할에 SAML 사용자 매핑

도메인 상태가 활성이고 IdP가 올바르게 구성되었으면 OpenSearch 대시보드로 이동하세요.

- SP가 시작한 URL을 선택한 경우 *domain-endpoint*`/_dashboards`로 이동합니다. 특정 테넌트에 직접 로그인하려면 URL에 `?security_tenant=`*tenant-name*을 추가합니다.
- IdP가 시작한 URL을 선택한 경우 자격 증명 공급자의 애플리케이션 디렉터리로 이동합니다.

두 경우 모두 SAML 마스터 사용자나 SAML 마스터 백엔드 역할에 속한 사용자로 로그인합니다. 7단계의 예제를 계속하려면 `jdoe` 또는 `admins` 그룹의 멤버로 로그인합니다.

OpenSearch 대시보드가 로드된 후 Security(보안), Roles(역할)를 선택합니다. 그런 다음 다른 사용자가 OpenSearch 대시보드에 액세스할 수 있도록 역할을 매핑합니다.

예를 들어 신뢰할 수 있는 동료 `jroe`를 `all_access` 및 `security_manager` 역할에 매핑할 수 있습니다. 백엔드 역할 `analysts`를 `readall` 및`opensearch_dashboards_user` 역할에 매핑할 수도 있습니다.

OpenSearch 대시보드 대신 API를 사용하려는 경우 다음 샘플 요청을 참조하세요.

```
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"] }
  }
]
```

SP 및 IdP 시작 인증 모두 구성

SP와 IdP가 시작한 인증을 모두 구성하려면 자격 증명 공급자를 통해 인증을 구성해야 합니다. 예를 들어 Okta에서 다음 단계를 수행할 수 있습니다.

1. SAML 애플리케이션 내에서 General(일반)의 SAML settings(SAML 설정)로 이동합니다.
2. 단일 인증 URL(Single sign on URL)에서 IdP 시작 SSO URL을 제공합니다. 예를 들어 `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs/idpinitiated`입니다.
3. 이 앱이 다른 SSO URL을 요청하도록 허용(Allow this app to request other SSO URLs)을 사용 설정합니다.
4. 요청 가능한 SSO URL(Requestable SSO URLs)에서 SP 시작 SSO URL을 하나 이상 추가합니다. 예를 들어 `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs`입니다.

SAML 인증 구성(AWS CLI)

다음 AWS CLI 명령은 기존 도메인에서 OpenSearch Dashboards에 대한 SAML 인증을 활성화합니다.

```
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"}}'
```

메타데이터 XML의 모든 따옴표와 줄 바꿈 문자를 이스케이프해야 합니다. 예를 들어, `<KeyDescriptor use="signing">`와 줄 바꿈 대신 `<KeyDescriptor use=\"signing\">\n`을 사용합니다. AWS CLI사용에 대한 자세한 내용은 AWS CLI 명령 참조를 참조하세요.

SAML 인증 구성(구성 API)

구성 API에 대한 다음 요청은 기존 도메인의 OpenSearch 대시보드에 대한 SAML 인증을 활성화합니다.

```
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"
    }
  }
}
```

메타데이터 XML의 모든 따옴표와 줄 바꿈 문자를 이스케이프해야 합니다. 예를 들어, `<KeyDescriptor use="signing">`와 줄 바꿈 대신 `<KeyDescriptor use=\"signing\">\n`을 사용합니다. 구성 API 사용에 대한 자세한 내용은 OpenSearch Service API 참조를 확인하세요.

SAML 문제 해결

오류	세부 사항
요청: '*/some/path* '는 허용되지 않습니다.	자격 증명 공급자에게 올바른 SSO URL(3단계)을 제공했는지 확인하세요.
SAML을 활성화하려면 유효한 자격 증명 제공자 메타데이터 문서를 제공하세요.	IdP 메타데이터 파일이 SAML 2.0 표준을 준수하지 않습니다. 유효성 검사 도구를 사용하여 오류를 확인하세요.
SAML 구성 옵션은 콘솔에 표시되지 않습니다.	최신 서비스 소프트웨어로 업데이트하세요.

오류	세부 사항
SAML 구성 오류: SAML 구성을 검색하는 동안 문제가 발생했습니다. 설정을 확인하세요.	이 일반 오류는 여러 가지 이유로 발생할 수 있습니다. • 자격 증명 공급자에게 올바른 SP 엔터티 ID 및 SSO URL을 제공했는지 확인하세요. • IdP 메타데이터 파일을 다시 생성하고 IdP 엔터티 ID를 확인합니다. AWS 콘솔에 업데이트 된 메타데이터를 추가합니다. • 도메인 액세스 정책이 OpenSearch 대시보드 및 `_plugins/_security/*` 에 대한 액세스를 허용하는지 확인합니다. 일반적으로 세분화된 액세스 제어를 사용하는 도메인에는 개방적인 액세스 정책을 사용하는 것이 좋습니다. • SAML 구성 단계는 자격 증명 공급자의 설명서를 참조하세요.
역할 누락: 이 사용자에게 사용할 수 있는 역할이 없습니다. 시스템 관리자에게 문의하세요.	성공적으로 인증되었지만 SAML 어설션의 사용자 이름 및 백엔드 역할은 어떤 역할에도 매핑되지 않으므로 권한이 없습니다. 이러한 매핑은 대/소문자를 구분합니다. SAML-tracer와 같은 도구를 사용하여 SAML 어설션의 콘텐츠를 확인하고 다음 요청을 사용하여 역할 매핑을 확인합니다. `GET _plugins/_security/api/rolesmapping`
브라우저가 OpenSearch 대시보드에 액세스하려고 할 때 HTTP 500 오류를 지속해서 리디렉션하거나 수신합니다.	SAML 어설션에 총 1,500자 정도의 많은 역할이 포함된 경우 이러한 오류가 발생할 수 있습니다. 예를 들어 평균 길이가 20자인 80개의 역할을 전달하면 웹 브라우저에서 쿠키의 크기 제한을 초과할 수 있습니다. OpenSearch 버전 2.7부터 SAML 어설션은 최대 5000자의 역할을 지원합니다.

오류	세부 사항
ADFS에서 로그아웃할 수 없습니다.	ADFS에서는 OpenSearch Service가 지원하지 않는 모든 로그아웃 요청에 서명해야 합니다. OpenSearch Service가 자체 내부 로그아웃 메커니즘을 사용하도록 하려면 IdP 메타데이터 파일에서 `<SingleLogoutService />` 를 제거합니다.
`Could not find entity descriptor for __PATH__.`	메타데이터 XML에서 OpenSearch Service에 제공된 IdP의 엔티티 ID가 SAML 응답의 엔티티 ID와 다릅니다. 이 문제를 해결하려면 두 항목이 일치하는지 확인하세요. 도메인에서 CW 애플리케이션 오류 로그를 활성화하여 SAML 수집 문제를 디버깅하는 데 필요한 오류 메시지를 찾으십시오.
`Signature validation failed. SAML response rejected.`	OpenSearch Service는 메타데이터 XML에 제공된 IdP의 인증서를 사용하여 SAML 응답의 서명을 확인할 수 없습니다. 수동 오류이거나 IdP가 인증서를 교체한 것일 수 있습니다. AWS Management Console를 통해 OpenSearch Service에 제공된 메타데이터 XML에서 IdP의 최신 인증서를 업데이트합니다.
`__PATH__ is not a valid audience for this response.`	SAML 응답의 대상 필드가 도메인 엔드포인트와 일치하지 않습니다. 이 오류를 해결하려면 SP 대상 필드를 도메인 엔드포인트와 일치하도록 업데이트하세요. 사용자 지정 엔드포인트를 활성화한 경우 대상 필드는 사용자 지정 엔드포인트와 일치해야 합니다. 도메인에서 CW 애플리케이션 오류 로그를 활성화하여 SAML 수집 문제를 디버깅하는 데 필요한 오류 메시지를 찾으십시오.
브라우저에 응답 내 `Invalid Request Id`과 함께 HTTP 400 오류가 수신됩니다.	이 오류는 일반적으로 IdP에서 시작하는 URL을 *`<DashboardsURL>`* `/_opendistro/_security/saml/acs` 형식으로 구성한 경우에 발생합니다. 대신 *`<DashboardsURL>`* `/_opendistro/_security/saml/acs/idpinitiated` 형식으로 URL을 구성하세요.

오류	세부 사항
__PATH__ 대신 __PATH__에서 응답을 받았습니다.	SAML 응답의 대상 필드가 다음 URL 형식 중 하나와 일치하지 않습니다. • *<DashboardsURL>* /_opendistro/_security/saml/acs • *<DashboardsURL>* /_opendistro/_security/saml/acs/idpinitiated . 사용하는 로그인 흐름(SP 시작 또는 IDP 시작)에 따라 OpenSearch URL 중 하나와 일치하는 대상 필드를 입력합니다.
응답에는 InResponseTo 속성이 있지만 InResponseTo 은 예상되지 않았습니다.	SP에서 시작한 로그인 흐름에 IdP에서 시작한 URL을 사용하고 있습니다. SP에서 시작한 URL을 대신 사용하세요.

SAML 인증 비활성화

OpenSearch 대시보드에 대한 SAML 인증을 비활성화하려면(콘솔)

1. 도메인을 선택하고 [작업(Actions)], [보안 구성 편집(Edit security configuration)]을 선택합니다.
2. SAML 인증 활성화(Enable SAML authentication)를 선택 취소합니다.
3. 변경 사항 저장을 선택합니다.
4. 도메인이 처리를 마친 후 다음 요청으로 세분화된 액세스 제어 역할 매핑을 확인합니다.

```
GET _plugins/_security/api/rolesmapping
```

Dashboards에 대한 SAML 인증을 비활성화하면 SAML 마스터 사용자 이름 및/또는 SAML 마스터 백엔드 역할에 대한 매핑을 제거하지 않습니다. 이러한 매핑을 제거하려면 내부 사용자 데이터베이스(활성화된 경우)를 사용하여 Dashboards에 로그인하거나 API를 사용하여 제거합니다.

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "users": [
```

```
    "master-user"
  ]
}
```

OpenSearch에 대한 IAM Identity Center 신뢰할 수 있는 자격 증명 전파 지원

이제 신뢰할 수 있는 ID 전파를 통해 중앙에서 구성된 AWS IAM Identity Center 보안 주체(사용자 및 그룹)를 사용하여 OpenSearch Service 애플리케이션를 통해 Amazon OpenSearch Service 도메인에 액세스할 수 있습니다. OpenSearch에 대한 IAM Identity Center 지원을 활성화하려면 IAM Identity Center 사용을 활성화해야 합니다.이 방법을 자세히 알아보려면 IAM Identity Center란 무엇입니까?를 참조하세요. 자세한 내용은 OpenSearch 도메인을 OpenSearch 애플리케이션의 데이터 소스로 연결하는 방법 단원을 참조하십시오.

OpenSearch Service 콘솔, AWS Command Line Interface (AWS CLI) 또는 SDK를 사용하여 IAM Identity Center를 AWS 구성할 수 있습니다. SDKs

> **Note**
>
> IAM Identity Center 보안 주체는 대시보드(클러스터와 함께 위치)를 통해 지원되지 않습니다. 중앙 집중식 OpenSearch 사용자 인터페이스(대시보드)를 통해서만 지원됩니다.

고려 사항

Amazon OpenSearch Service에서 IAM Identity Center를 사용하려면 먼저 다음 사항을 고려해야 합니다.

- IAM Identity Center는 계정에서 활성화됩니다.
- IAM Identity Center는 해당 리전에서 사용할 수 있습니다.
- OpenSearch 도메인 버전은 1.3 이상입니다.
- 도메인에서 세분화된 액세스 제어가 활성화됩니다.
- 도메인이 IAM Identity Center 인스턴스와 동일한 리전에 있습니다.
- 도메인과 OpenSearch 애플리케이션은 동일한 AWS 계정에 속합니다.

도메인 액세스 정책 수정

IAM Identity Center를 구성하기 전에 신뢰할 수 있는 자격 증명 전파를 위해 OpenSearch 애플리케이션에 구성된 IAM 역할의 권한 또는 도메인 액세스 정책을 업데이트해야 합니다.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "IAM Role configured in OpenSearch application"
            },
            "Action": "es:ESHttp*",
            "Resource": "domain-arn/*"
        },
        {
        ... // Any other permissions
        }
    ]
}
```

IAM Identity Center 인증 및 권한 부여 구성(콘솔)

도메인 생성 프로세스 중에 또는 기존 도메인을 업데이트하여 IAM Identity Center 인증 및 권한 부여를 활성화할 수 있습니다. 설정 단계는 선택 사항에 따라 약간 다릅니다.

다음 단계에서는 Amazon OpenSearch Service 콘솔에서 IAM Identity Center 인증 및 권한 부여를 위해 기존 도메인을 구성하는 방법을 설명합니다.

1. 도메인 구성에서 보안 구성으로 이동하여 편집을 선택하고 IAM Identity Center 인증 섹션으로 이동한 다음 IAM Identity Center로 인증된 API 액세스 활성화를 선택합니다.
2. 다음과 같이 SubjectKey 및 Roles 키를 선택합니다.

    - 제목 키 - 해당 속성을 도메인에 액세스하는 보안 주체로 사용하려면 UserId(기본값), UserName 및 Email 중 하나를 선택합니다.

- 역할 키 - GroupId IdC 보안 주체와 연결된 모든 그룹에 대한 fine-grained-access-control 위해 해당 속성 값을 백엔드 역할로 사용하려면 GroupId(기본값) 및 GroupName 중 하나를 선택합니다.

변경한 후에 도메인을 저장합니다.

세분화된 액세스 제어 구성

OpenSearch 도메인에서 IAM Identity Center 옵션을 활성화한 후에는 백엔드 역할에 대한 역할 매핑을 생성하여 IAM Identity Center 보안 주체에 대한 액세스를 구성할 수 있습니다. 보안 주체의 백엔드 역할 값은 IdC 보안 주체의 그룹 멤버십과 GroupId 또는 GroupName의 RolesKey 구성을 기반으로 합니다.

> **Note**
>
> Amazon OpenSearch Service는 단일 사용자에 대해 최대 100개의 그룹을 지원할 수 있습니다. 허용된 인스턴스 수를 초과하여 사용하려고 하면 fine-grained-access-control 권한 부여 처리에 불일치가 발생하여 403 오류 메시지가 표시됩니다.

IAM Identity Center 인증 및 권한 부여 구성(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"}'
```

도메인에서 IAM Identity Center 인증 비활성화

OpenSearch 도메인에서 IAM Identity Center를 비활성화하려면:

1. 도메인을 선택하고 [작업(Actions)], [보안 구성 편집(Edit security configuration)]을 선택합니다.
2. IAM Identity Center로 인증된 API 액세스 활성화를 선택 취소합니다.
3. 변경 사항 저장을 선택합니다.
4. 도메인 처리가 완료되면 IdC 보안 주체에 대해 추가된 역할 매핑 제거

CLI를 통해 IAM Identity Center를 비활성화하려면 다음을 사용할 수 있습니다.

```
aws opensearch update-domain-config \
    --domain-name my-domain \
    --identity-center-options '{"EnabledAPIAccess": false}'
```

OpenSearch Dashboards에 대한 Amazon Cognito 인증 구성

Amazon Cognito를 사용하여 OpenSearch Dashboards의 Amazon OpenSearch Service 기본 설치를 인증하고 보호할 수 있습니다. Amazon Cognito 인증은 선택 사항이며 OpenSearch 또는 Elasticsearch 5.1 이상을 사용하는 도메인에서만 사용할 수 있습니다. Amazon Cognito 인증을 구성하지 않은 경우에도 IP 기반 액세스 정책 및 프록시 서버, HTTP 기본 인증 또는 SAML을 사용하여 Dashboards를 보호할 수 있습니다.

인증 프로세스의 많은 부분은 Amazon Cognito에서 진행되지만 이 섹션에서 OpenSearch Service 도메인과 호환되도록 Amazon Cognito 리소스를 구성하는 지침과 요건을 알려드립니다. 표준 요금은 모든 Amazon Cognito 리소스에 적용됩니다.

> **Tip**
>
> OpenSearch Dashboards에 대해 Amazon Cognito 인증을 사용하도록 도메인을 처음 구성할 때는 콘솔을 사용하는 것이 좋습니다. Amazon Cognito 리소스는 사용자 지정 기능이 뛰어나며, 콘솔은 사용자에게 중요한 기능을 식별하고 이해하는 데 도움이 됩니다.

주제

- 사전 조건
- Amazon Cognito 인증을 사용하도록 도메인 구성
- 인증된 역할 허용
- 자격 증명 공급자 구성
- (선택 사항) 세분화된 액세스 구성
- (선택 사항) 로그인 페이지 사용자 지정
- (선택 사항) 고급 보안 구성
- 테스트

- 할당량
- 일반적인 구성 문제
- OpenSearch Dashboards에 대한 Amazon Cognito 인증 비활성화
- OpenSearch Dashboards에 대한 Amazon Cognito 인증을 사용하는 도메인 삭제

사전 조건

OpenSearch Dashboards에 대한 Amazon Cognito 인증을 구성하기 전에 충족해야 하는 사전 조건이 여럿 있습니다. OpenSearch Service 콘솔은 이런 리소스의 생성을 간소화해 주지만, 각 리소스의 목적을 이해해야 구성과 문제 해결에 도움이 됩니다. Dashboards에 대한 Amazon Cognito 인증에는 다음 리소스가 필요합니다.

- Amazon Cognito 사용자 풀
- Amazon Cognito 자격 증명 풀
- `AmazonOpenSearchServiceCognitoAccess` 정책이 연결된 IAM 역할 (`CognitoAccessForAmazonOpenSearch`)

> **Note**
>
> 사용자 풀과 자격 증명 풀은 동일한 AWS 리전에 있어야 합니다. 동일한 사용자 풀과 자격 증명 풀 및 IAM 역할을 이용해 여러 OpenSearch Service 도메인에 Dashboards에 대한 Amazon Cognito 인증을 추가할 수 있습니다. 자세한 내용은 the section called “할당량” 섹션을 참조하세요.

사용자 풀 소개

사용자 풀의 주된 기능은 사용자 디렉터리를 만들고 관리하는 것과 사용자가 가입하고 로그인하게 하는 것 두 가지입니다. 사용자 풀을 생성하는 지침은 Amazon Cognito 개발자 안내서의 사용자 풀 시작하기를 참조하세요.

OpenSearch Service에 사용할 사용자 풀을 생성할 때는 다음 사항을 고려하세요.

- Amazon Cognito 사용자 풀에 도메인 이름이 있어야 합니다. OpenSearch Service는 이 도메인 이름을 사용해 사용자를 Dashboards에 액세스하는 로그인 페이지로 리디렉션합니다. 사용자 풀은 도메인 이름 외에 다른 기본값이 아닌 구성은 필요하지 않습니다.

- 풀의 필수 표준 속성(이름, 생년월일, 이메일 주소 및 전화번호 등)을 지정해야 합니다. 사용자 풀을 만든 후에는 이런 속성을 변경할 수 없으므로 이때 중요한 속성을 선택해야 합니다.
- 사용자 풀을 만드는 동안 사용자가 자기 계정을 만들 수 있는지 여부, 계정 암호의 최소 강도, 멀티 팩터 인증 활성화 여부 등을 선택하십시오. 외부 자격 증명 공급자를 이용할 계획이라면 이런 설정은 중요하지 않습니다. 기술적으로는 사용자 풀을 자격 증명 공급자로도 사용할 수 있고 동시에 외부 자격 증명 공급자로도 사용할 수 있지만, 대부분은 어느 한쪽을 선호합니다.

사용자 풀 ID는 *region_ID*의 형식입니다. AWS CLI 또는 AWS SDK를 사용하여 OpenSearch Service를 구성하려는 경우 ID를 기록해 둡니다.

자격 증명 풀 소개

자격 증명 풀을 사용하면 로그인 후 제한적 권한의 임시 역할을 사용자에게 할당할 수 있습니다. 자격 증명 풀 생성에 대한 지침은 Amazon Cognito 개발자 안내서의 자격 증명 풀 콘솔 개요를 참조하세요. OpenSearch Service에 사용할 자격 증명 풀을 생성할 때는 다음 사항을 고려하세요.

- Amazon Cognito 콘솔을 사용할 경우 인증되지 않은 자격 증명에 대한 액세스 활성화(Enable access to unauthenticated identities) 확인란을 선택해야 자격 증명 풀을 만들 수 있습니다. 자격 증명 풀을 만들고 OpenSearch Service 도메인을 구성하고 나면 Amazon Cognito에서 이 설정을 비활성화합니다.
- 자격 증명 풀에 외부 자격 증명 공급자를 추가할 필요는 없습니다. Amazon Cognito 인증을 사용하려고 OpenSearch Service를 구성할 때는 방금 만든 사용자 풀을 사용할 자격 증명 풀을 구성합니다.
- 자격 증명 풀을 만든 다음에는 인증 및 인증되지 않은 IAM 역할을 선택해야 합니다. 이러한 역할은 로그인 전후 사용자의 액세스 정책을 지정합니다. Amazon Cognito 콘솔을 사용하는 경우 이런 역할이 자동으로 생성됩니다. 인증 역할을 만든 후에는 ARN을 기록해 둡니다. `arn:aws:iam::`*123456789012*`:role/Cognito_`*identitypoolname*`Auth_Role`의 형식입니다.

자격 증명 풀 ID는 *region*`:`*ID*`-`*ID*`-`*ID*`-`*ID*`-`*ID*의 형식입니다. AWS CLI 또는 AWS SDK를 사용하여 OpenSearch Service를 구성하려는 경우 ID를 기록해 둡니다.

CognitoAccessForAmazonOpenSearch 역할 정보

OpenSearch Service는 권한이 있어야 Amazon Cognito 사용자와 자격 증명 풀을 구성하고 이것들을 인증에 사용할 수 있습니다. 이를 위해 AWS 관리형 정책`AmazonOpenSearchServiceCognitoAccess`인를 사용할 수 있습니다. `AmazonESCognitoAccess`는 서비스의 이름이 Amazon OpenSearch Service로 변경되었을

AmazonOpenSearchServiceCognitoAccess 때 로 대체된 레거시 정책입니다. 두 정책 모두 Amazon Cognito 인증을 활성화하는 데 필요한 최소 Amazon Cognito 권한을 제공합니다. 정책 세부 정보는 AWS 관리형 정책 참조 안내서의 AmazonOpenSearchServiceCognitoAccess를 참조하세요.

콘솔을 사용해 OpenSearch Service 도메인을 생성하거나 구성할 경우 IAM 역할이 자동으로 생성되며 역할에 AmazonOpenSearchServiceCognitoAccess 정책(Elasticsearch 도메인인 경우 AmazonESCognitoAccess 정책)이 연결됩니다. 이 역할의 기본 이름은 CognitoAccessForAmazonOpenSearch입니다.

역할 허가 정책 AmazonOpenSearchServiceCognitoAccess 및 AmazonESCognitoAccess는 모든 자격 증명 및 사용자 풀에서 다음 작업을 완료할 수 있도록 OpenSearch Service를 지원합니다.

- 작업: cognito-idp:DescribeUserPool
- 작업: cognito-idp:CreateUserPoolClient
- 작업: cognito-idp:DeleteUserPoolClient
- 작업: cognito-idp:UpdateUserPoolClient
- 작업: cognito-idp:DescribeUserPoolClient
- 작업: cognito-idp:AdminInitiateAuth
- 작업: cognito-idp:AdminUserGlobalSignOut
- 작업: cognito-idp:ListUserPoolClients
- 작업: cognito-identity:DescribeIdentityPool
- 작업: cognito-identity:SetIdentityPoolRoles
- 작업: cognito-identity:GetIdentityPoolRoles

AWS CLI 또는 AWS SDKs 중 하나를 사용하는 경우 OpenSearch Service 도메인을 구성할 때 자체 역할을 생성하고, 정책을 연결하고,이 역할에 대한 ARN을 지정해야 합니다. 역할은 다음과 같은 신뢰 관계를 맺고 있어야 합니다.

JSON

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

```
        "Service": "opensearchservice.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

지침은 IAM 사용 설명서의 AWS 서비스에 권한을 위임할 역할 생성 및 IAM 자격 증명 권한 추가 및 제거를 참조하세요. https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-attach-detach.html

Amazon Cognito 인증을 사용하도록 도메인 구성

사전 조건을 마친 후 Dashboards에 Amazon Cognito를 사용하기 위해 OpenSearch Service 도메인을 구성할 수 있습니다.

> **Note**
>
> Amazon Cognito를 전혀 사용할 수 없습니다 AWS 리전. 지원되는 리전 목록은 Amazon Cognito용 서비스 엔드포인트를 참조하세요. OpenSearch Service에 사용한 것과 똑같은 Amazon Cognito 리전을 사용할 필요는 없습니다.

Amazon Cognito 인증 구성(콘솔)

역할을 생성하므로 `CognitoAccessForAmazonOpenSearch` 콘솔은 가장 간단한 구성 환경을 제공합니다. 콘솔로 OpenSearch Dashboards에 대한 Amazon Cognito 인증을 사용하는 도메인을 만들려면 표준 OpenSearch Service 권한 외에도 다음과 같은 권한 모음이 있어야 합니다.

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"
    }
  ]
}
```

자격 증명 (사용자, 사용자 그룹 또는 역할) 에 권한을 추가하는 방법은 IAM 자격 증명 권한 추가(콘솔)를 참조하세요.

CognitoAccessForAmazonOpenSearch가 이미 존재한다면 다음과 같이 더 적은 권한만 필요합니다.

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"
    }
  ]
}
```

Dashboards에 대한 Amazon Cognito 인증을 구성하려면(콘솔)

1. https://console.aws.amazon.com/aos/home/ Amazon OpenSearch Service 콘솔을 엽니다.
2. [도메인(Domains)]에서 구성할 도메인을 선택합니다.
3. [작업(Actions)], [보안 구성 편집(Edit security configuration)]을 선택합니다.
4. Amazon Cognito 인증 활성화(Enable Amazon Cognito authentication)를 선택합니다.
5. 리전에서 Amazon Cognito 사용자 풀 및 자격 증명 풀 AWS 리전 이 포함된를 선택합니다.
6. [Cognito 사용자 풀(Cognito user pool)]에서 사용자 풀을 하나 선택하거나 만듭니다. 자세한 내용은 the section called “사용자 풀 소개” 단원을 참조하십시오.
7. [Cognito 자격 증명 풀(Cognito identity pool)]에서 자격 증명 풀을 하나 선택하거나 만듭니다. 자세한 내용은 the section called “자격 증명 풀 소개” 단원을 참조하십시오.

   > **Note**
   >
   > [사용자 풀 생성(Create user pool)] 및 [자격 증명 풀 생성(Create identity pool)] 링크는 Amazon Cognito 콘솔로 연결되며, 이런 리소스를 수동으로 만들어야 합니다. 이 프로세스는 자동이 아닙니다. 자세한 내용은 the section called “사전 조건” 단원을 참조하십시오.

8. [IAM 역할 이름(IAM role name)]의 경우 기본값 `CognitoAccessForAmazonOpenSearch`를 사용(권장)하거나 새 이름을 입력합니다. 자세한 내용은 the section called “CognitoAccessForAmazonOpenSearch 역할 정보” 단원을 참조하십시오.
9. 변경 사항 저장을 선택합니다.

도메인이 처리를 마친 후 추가 구성 단계는 the section called “인증된 역할 허용” 및 the section called “자격 증명 공급자 구성” 섹션을 참조하세요.

Amazon Cognito 인증 구성(AWS CLI)

OpenSearch Service 도메인을 구성하려면 `--cognito-options` 파라미터를 사용합니다. 다음 구문은 `create-domain` 및 `update-domain-config` 명령 둘 다에서 사용됩니다.

```
--cognito-options Enabled=true,UserPoolId="user-pool-id",IdentityPoolId="identity-pool-id",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonOpenSearch"
```

예제

다음은 `CognitoAccessForAmazonOpenSearch` 역할을 이용하여 Dashboards에 Amazon Cognito 인증을 사용하는 도메인을 `us-east-1` 리전에 만들어 `Cognito_Auth_Role`에 도메인 액세스를 제공하는 예입니다.

```
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"
```

도메인이 처리를 마친 후 추가 구성 단계는 the section called “인증된 역할 허용” 및 the section called “자격 증명 공급자 구성” 섹션을 참조하세요.

Amazon Cognito 인증 구성(AWS SDK)

AWS SDKs(Android 및 iOS SDKs 제외)는 `CreateDomain` 및 작업에 대한 `CognitoOptions` 파라미터를 포함하여 Amazon OpenSearch Service API 참조에 정의된 모든 `UpdateDomainConfig` 작업을 지원합니다. AWS SDKs 설치 및 사용에 대한 자세한 내용은 AWS 소프트웨어 개발 키트를 참조하세요.

도메인이 처리를 마친 후 추가 구성 단계는 the section called “인증된 역할 허용” 및 the section called “자격 증명 공급자 구성” 섹션을 참조하세요.

인증된 역할 허용

기본적으로 the section called “자격 증명 풀 소개”의 지침에 따라 구성한 인증된 IAM 역할은 OpenSearch Dashboards 액세스에 필요한 권한이 없습니다. 역할에 추가 권한을 부여해야 합니다.

> **Note**
>
> 세분화된 액세스 제어를 구성하고 “개방형” 또는 IP 기반 액세스 정책을 사용하는 경우 이 단계를 건너뛸 수 있습니다.

자격 증명 기반 정책에 이런 권한을 포함할 수도 있지만 인증된 사용자가 모든 OpenSearch Service 도메인에 액세스할 권한을 원하는 경우가 아니라면 리소스 기반 정책을 도메인 하나에 연결하는 것이 더 나은 접근 방식입니다.

`Principal`의 경우 the section called “자격 증명 풀 소개”의 지침에 따라 구성한 Cognito 인증 역할의 ARN을 지정합니다.

JSON

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

OpenSearch Service 도메인에 리소스 기반 정책을 추가하는 방법에 대한 자세한 내용은 the section called “액세스 정책 구성” 섹션을 참조하세요.

자격 증명 공급자 구성

Dashboards에 Amazon Cognito 인증을 사용하려고 도메인을 구성할 때, OpenSearch Service는 사용자 풀에 앱 클라이언트를 추가하고 그 사용자 풀을 자격 증명 풀에 인증 공급자로 추가합니다.

> **Warning**
>
> 앱 클라이언트의 이름을 바꾸거나 삭제하지 마세요.

사용자 풀을 어떻게 구성했느냐에 따라 사용자 계정을 수동으로 만들어야 할 수도 있고, 사용자가 직접 자신의 계정을 만들게 할 수도 있습니다. 이러한 설정이 허용되는 경우 추가 조치가 필요 없습니다. 그러나 외부 자격 증명 공급자 사용을 선호하는 사람들이 많습니다.

SAML 2.0 자격 증명 공급자를 활성화하려면 SAML 메타데이터 문서를 제공해야 합니다. Login with Amazon, Facebook, Google 같은 소셜 자격 증명 공급자를 사용하려면 이런 공급자들에게서 받은 앱 ID와 앱 암호를 가지고 있어야 합니다. 자격 증명 공급자는 자유롭게 조합하여 사용할 수 있습니다.

사용자 풀을 구성하는 가장 쉬운 방법은 Amazon Cognito 콘솔을 사용하는 것입니다. 지침은 Amazon Cognito 개발자 안내서의 타사 자격 증명 공급자를 사용한 사용자 풀 로그인 및 앱 클라이언트를 사용한 애플리케이션별 설정을 참조하세요.

(선택 사항) 세분화된 액세스 구성

기본 자격 증명 풀 설정이 동일한 IAM 역할(`Cognito_`*`identitypool`*`Auth_Role`)에 로그인하는 모든 사용자를 할당한다는 것을 알았을 것입니다. 즉, 모든 사용자가 동일한 AWS 리소스에 액세스할 수 있습니다. Amazon Cognito에서 세분화된 액세스 제어를 사용하려는 경우, 예를 들어 조직의 분석가가 여러 인덱스에 대한 읽기 전용 액세스 권한을 갖도록 하고 개발자는 모든 인덱스에 대한 쓰기 권한을 갖도록 하려면 다음 두 가지 옵션이 있습니다.

- 사용자 그룹을 생성하고 자격 증명 공급자가 사용자 인증 토큰을 기반으로 IAM 역할을 선택하도록 구성합니다(권장).
- 자격 증명 공급자가 하나 이상의 규칙을 기반으로 IAM 역할을 선택하도록 구성합니다.

세분화된 액세스 제어가 포함된 시연은 the section called “자습서: Cognito 인증을 사용한 세분화된 액세스 제어” 섹션을 참조하세요.

> **Important**
>
> 기본 역할과 마찬가지로 Amazon Cognito는 각 추가 역할의 신뢰 관계에 속해야 합니다. 자세한 내용은 Amazon Cognito 개발자 안내서의 역할 매핑을 위한 역할 생성을 참조하세요.

사용자 그룹과 토큰

사용자 그룹을 생성할 때 그 그룹의 구성원에 대한 IAM 역할을 선택합니다. 그룹 생성에 대한 자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀에 그룹 추가를 참조하세요.

사용자 그룹을 하나 이상 만든 후 인증 공급자를 구성하여 사용자에게 자격 증명 풀의 기본 역할 대신 그룹 역할을 할당할 수 있습니다. 토큰으로부터 역할 선택(Choose role from token)을 선택한 다음, 기본 인증된 역할 사용(Use default Authenticated role) 또는 거부(DENY) 중 하나를 선택하여 자격 증명 풀이 그룹의 일부가 아닌 사용자를 취급하는 방식을 지정합니다.

규칙

규칙은 기본적으로 Amazon Cognito가 순차적으로 평가하는 일련의 `if`문입니다. 예를 들어 사용자의 이메일 주소에 `@corporate`이 포함된 경우, Amazon Cognito는 그 사용자를 `Role_A`로 할당합니다. 사용자의 이메일 주소에 `@subsidiary`가 포함된 경우에는 해당 사용자를 `Role_B`로 할당합니다. 그렇지 않은 경우 사용자에게 기본 인증된 역할을 할당합니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 규칙 기반 매핑을 사용하여 사용자에게 역할 할당을 참조하세요.

(선택 사항) 로그인 페이지 사용자 지정

Amazon Cognito 콘솔의 UI 사용자 지정 페이지에서 사용자 지정 로고를 업로드하고 로그인 페이지에 CSS를 변경할 수 있습니다. CSS 속성의 지침 및 전체 목록은 Amazon Cognito 개발자 안내서의 호스팅 UI(클래식) 브랜딩 사용자 지정을 참조하세요.

(선택 사항) 고급 보안 구성

Amazon Cognito 사용자 풀은 멀티 팩터 인증, 손상된 자격 증명 확인, 조정 인증과 같은 어드밴스 보안 기능을 지원합니다. 자세한 내용은 Amazon Cognito 개발자 안내서의 Amazon Cognito 사용자 풀 보안 기능 사용을 참조하세요. Amazon Cognito

테스트

구성에 만족하면 사용자 경험이 기대에 부합하는지 확인하세요.

OpenSearch Dashboards에 액세스하려면

1. 웹 브라우저에서 `https://`*`opensearch-domain`*`/_dashboards`로 이동합니다. 특정 테넌트에 직접 로그인하려면 URL에 `?security_tenant=`*`tenant-name`*을 추가하세요.
2. 선호하는 자격 증명을 사용하여 로그인합니다.
3. OpenSearch Dashboards가 로드되면 인덱스 패턴을 한 개 이상 구성합니다. Dashboards는 이러한 패턴을 사용하여 분석할 인덱스를 식별하기 때문입니다. *를 입력하고 다음 단계(Next step)를 선택한 후 인덱스 패턴 생성(Create index pattern)을 선택합니다.
4. 데이터를 검색하거나 탐색하려면 검색(Discover)을 선택합니다.

이 과정의 어느 단계든 실패하면 the section called “일반적인 구성 문제”에서 문제 해결 정보를 확인하세요.

할당량

Amazon Cognito에는 다양한 리소스에 대한 소프트웨어 제한이 있습니다. 많은 수의 OpenSearch Service 도메인에 Dashboards 인증을 사용하려면 Amazon Cognito의 할당량을 확인하고 필요에 따라 한도 증가를 요청합니다.

각 OpenSearch Service 도메인은 사용자 풀에 앱 클라이언트를 추가하여 자격 증명 풀에 인증 공급자를 추가합니다. 10개 이상의 도메인에 OpenSearch Dashboards 인증을 사용할 경우 "자격 인증 풀당 최대 Amazon Cognito 사용자 풀 공급자" 제한에 걸릴 수도 있습니다. 제한을 초과할 경우 Dashboards에 Amazon Cognito 인증을 사용하려고 구성을 시도하는 모든 OpenSearch Service 도메인은 처리 중(Processing) 구성 상태에 고착될 수 있습니다.

일반적인 구성 문제

다음 표는 일반적인 구성 문제와 해결책 목록입니다.

OpenSearch Service 구성

문제	Solution
`OpenSearch Service can't create the role`(콘솔)	올바른 IAM 권한이 없습니다. the section called “Amazon Cognito 인증 구성(콘솔)”에 지정된 권한을 추가하세요.

문제	Solution
`User is not authorized to perform: iam:PassRole on resource CognitoAccessForAmazonOpenSearch (콘솔)`	CognitoAccessForAmazonOpenSearch 역할에 대한 `iam:PassRole` 권한이 없습니다. 사용자 계정에 다음 정책을 연결합니다. JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iam:PassRole" ], "Resource": "arn:aws:iam::123456789012:role/service-role/CognitoAccessForAmazonOpenSearch" } ] }` 또는 `IAMFullAccess` 정책을 연결할 수 있습니다.
`User is not authorized to perform: cognito-identity:ListIdentityPools on resource`	Amazon Cognito에 대한 읽기 권한이 없습니다. 사용자 계정에 `AmazonCognitoReadOnly` 정책을 연결하세요.
`An error occurred (ValidationException) when calling the CreateDomain operation: OpenSearch Service must be allowed to use the passed role`	OpenSearch Service가 `CognitoAccessForAmazonOpenSearch` 역할의 신뢰 관계에 지정되지 않았습니다. 역할이 the section called “CognitoAccessForAmazonOpenSearch 역할 정보”에 지정된 신뢰 관계를 사용하는지 확인하세요. 또는 콘솔을 이용해 Amazon Cognito 인증을 구성하세요. 콘솔에서 자동으로 역할을 만들어줍니다.

문제	Solution
`An error occurred (ValidationException) when calling the CreateDomain operation: User is not authorized to perform: cognito-idp: action on resource: user pool`	`--cognito-options` 에 지정된 역할에 Amazon Cognito 액세스 권한이 없습니다. 역할에 AWS 관리형 `AmazonOpenSearchServiceCognitoAccess` 정책이 연결되어 있는지 확인합니다. 또는 콘솔을 이용해 Amazon Cognito 인증을 구성하세요. 콘솔에서 자동으로 역할을 만들어줍니다.
`An error occurred (ValidationException) when calling the CreateDomain operation: User pool does not exist`	OpenSearch Service에서 사용자 풀을 찾을 수 없습니다. 자격 증명 풀을 생성했고 ID가 올바른지 확인합니다. ID를 찾으려면 Amazon Cognito 콘솔 또는 다음 AWS CLI 명령을 사용할 수 있습니다. `aws cognito-idp list-user-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateDomain operation: IdentityPool not found`	OpenSearch Service에서 자격 증명 풀을 찾을 수 없습니다. 자격 증명 풀을 생성했고 ID가 올바른지 확인합니다. ID를 찾으려면 Amazon Cognito 콘솔 또는 다음 AWS CLI 명령을 사용할 수 있습니다. `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`	사용자 풀에 도메인 이름이 없습니다. Amazon Cognito 콘솔 또는 다음 AWS CLI 명령을 이용해 도메인 이름을 구성할 수 있습니다. `aws cognito-idp create-user-pool-domain --domain name --user-pool-id id`

OpenSearch Dashboards 액세스

문제	Solution
로그인 페이지에 선호하는 자격 증명 공급자가 보이지 않습니다.	the section called “자격 증명 공급자 구성”에 지정한 대로 OpenSearch Service 앱 클라이언트에 자격 증명 공급자를 활성화했는지 확인합니다.
로그인 페이지가 나의 조직과 연결된 것처럼 보이지 않습니다.	the section called “(선택 사항) 로그인 페이지 사용자 지정” 섹션을 참조하세요.
나의 로그인 자격 증명이 통하지 않습니다.	the section called “자격 증명 공급자 구성”에 지정한 대로 자격 증명 공급자를 구성했는지 확인합니다. 사용자 풀을 자격 증명 공급자로 사용하는 경우 해당 계정이 존재하고 Amazon Cognito 콘솔의 사용자 및 그룹 페이지에서 확인되는지 확인합니다.
OpenSearch Dashboards가 전혀 로드되지 않거나 제대로 작동하지 않습니다.	Amazon Cognito 인증된 역할이 Dashboards에 액세스하고 사용하려면 그 도메인(/*)에 대한 `es:ESHttp*` 권한이 필요합니다. the section called “인증된 역할 허용”에 지정된 대로 액세스 정책을 추가했는지 확인합니다.
한 탭에서 OpenSearch Dashboards에서 로그아웃하면 나머지 탭에 새로 고침 토큰이 취소되었다는 메시지가 표시됩니다.	Amazon Cognito 인증을 사용하는 동안 OpenSearch Dashboards 세션에서 로그아웃하면 OpenSearch Service는 AdminUserGlobalSignout 작업을 실행하여 모든 활성 OpenSearch Dashboards 세션에서 로그아웃합니다.
`Invalid identity pool configuration. Check assigned IAM roles for this pool.`	Amazon Cognito에는 인증된 사용자를 대신하여 IAM 역할을 수임할 권한이 없습니다. 다음을 포함하도록 역할에 대한 신뢰 관계를 수정합니다. JSON `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": {`

문제	Solution

	"Federated": "cognito-identity.amazonaws.com"
	},
	"Action": "sts:AssumeRoleWithWebIdentity",
	"Resource": "arn:aws:iam:: *111122223333* :role/*cognito-role* ",
	"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.`	이 일반적인 오류는 사용자 풀에서 앱 클라이언트를 삭제할 때 발생합니다. 새 브라우저 세션에서 Dashboards를 열어 보세요.

OpenSearch Dashboards에 대한 Amazon Cognito 인증 비활성화

다음 절차에 따라 Dashboards에 대한 Amazon Cognito 인증을 비활성화합니다.

Dashboards에 대한 Amazon Cognito 인증을 비활성화하려면(콘솔)

1. Amazon OpenSearch Service 콘솔을 엽니다.
2. Domains(도메인)에서 구성할 도메인을 선택합니다.
3. [작업(Actions)], [보안 구성 편집(Edit security configuration)]을 선택합니다.
4. Amazon Cognito 인증 활성화(Enable Amazon Cognito authentication)를 선택 취소합니다.
5. 변경 사항 저장을 선택합니다.

> **Important**
>
> Amazon Cognito 사용자 풀과 자격 증명 풀이 더 이상 필요하지 않으면 삭제합니다. 그러지 않으면 계속 요금이 부과됩니다.

OpenSearch Dashboards에 대한 Amazon Cognito 인증을 사용하는 도메인 삭제

Dashboards에 Amazon Cognito 인증을 사용하는 도메인이 Processing(처리 중) 구성 상태에서 멈춰있지 않도록 하려면, OpenSearch Service 도메인을 먼저 삭제한 다음 연결된 Amazon Cognito 사용자 및 자격 증명 풀을 삭제해야 합니다.

Amazon OpenSearch Service에 대한 서비스 연결 역할 사용

Amazon OpenSearch Service는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 OpenSearch Service에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 OpenSearch Service에서 사전 정의하며 서비스가 사용자를 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할은 OpenSearch Service를 더 쉽게 설정할 수 있습니다. OpenSearch Service에서 서비스 연결 역할의 권한을 정의하므로 다르게 정의되지 않은 한, OpenSearch Service만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔티티에 연결할 수 없습니다. 서비스 연결 역할 및 권한 정책에 대한 업데이트는 Amazon OpenSearch Service의 문서 기록을 참조하세요.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 AWS IAM으로 작업하는 서비스를 참조하고 서비스 연결 역할 열에서 예인 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

주제

- 서비스 연결 역할을 사용하여 VPC 도메인 및 직접 쿼리 데이터 소스 생성
- 서비스 연결 역할을 사용하여 OpenSearch Serverless 컬렉션 생성
- 서비스 연결 역할을 사용하여 OpenSearch Ingestion 파이프라인 생성

서비스 연결 역할을 사용하여 VPC 도메인 및 직접 쿼리 데이터 소스 생성

Amazon OpenSearch Service는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 OpenSearch Service에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 OpenSearch Service에서 사전 정의하며 사용자를 대신하여 서비스에서 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

OpenSearch Service는 AWSServiceRoleForAmazonOpenSearchService라는 서비스 연결 역할을 사용합니다.이 역할은 도메인 또는 직접 쿼리 데이터 소스에 대한 VPC 액세스를 활성화하는 데 필요한 최소 Amazon EC2 및 Elastic Load Balancing 권한을 제공합니다.

레거시 Elasticsearch 역할

Amazon OpenSearch Service는 `AWSServiceRoleForAmazonOpenSearchService`라는 서비스 연결 역할을 사용합니다. 또한 계정에는 더 이상 사용되지 않는 Elasticsearch API 엔드포인트와 함께 작동하는 `AWSServiceRoleForAmazonElasticsearchService`라는 레거시 서비스 연결 역할도 포함될 수 있습니다.

레거시 Elasticsearch 역할이 계정에 존재하지 않는 경우, OpenSearch 도메인을 처음 생성할 때 OpenSearch Service가 자동으로 새로운 OpenSearch 서비스 연결 역할을 생성합니다. 그렇지 않으면 계정에서 Elasticsearch 역할을 계속 사용합니다. 이 자동 생성이 성공하려면 사용자가 `iam:CreateServiceLinkedRole` 작업에 대한 권한을 보유해야 합니다.

권한

`AWSServiceRoleForAmazonOpenSearchService` 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `opensearchservice.amazonaws.com`

`AmazonOpenSearchServiceRolePolicy`라는 이름의 역할 권한 정책은 OpenSearch Service가 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업: *에 대한 `acm:DescribeCertificate`
- 작업: *에 대한 `cloudwatch:PutMetricData`
- 작업: *에 대한 `ec2:CreateNetworkInterface`
- 작업: *에 대한 `ec2:DeleteNetworkInterface`
- 작업: *에 대한 `ec2:DescribeNetworkInterfaces`

- 작업: *에 대한 ec2:ModifyNetworkInterfaceAttribute
- 작업: *에 대한 ec2:DescribeSecurityGroups
- 작업: *에 대한 ec2:DescribeSubnets
- 작업: *에 대한 ec2:DescribeVpcs
- 작업: 모든 네트워크 인터페이스 및 VPC 엔드포인트에 대한 ec2:CreateTags
- 작업: *에 대한 ec2:DescribeTags
- 규칙의 내용을 설명하는 이름입니다작업: 모든 VPC, 보안 그룹, 서브넷, 라우팅 테이블과 요청에 OpenSearchManaged=true 태그가 포함될 경우 모든 VPC 엔드포인트에 대한 ec2:CreateVpcEndpoint
- 작업: 모든 VPC, 보안 그룹, 서브넷, 라우팅 테이블과 요청에 OpenSearchManaged=true 태그가 포함될 경우 모든 VPC 엔드포인트에 대한 ec2:ModifyVpcEndpoint
- 작업: 요청에 OpenSearchManaged=true 태그가 포함될 경우 모든 엔드포인트에 대한 ec2:DeleteVpcEndpoints
- 작업: *에 대한 ec2:AssignIpv6Addresses
- 작업: *에 대한 ec2:UnAssignIpv6Addresses
- 작업: *에 대한 elasticloadbalancing:AddListenerCertificates
- 작업: *에 대한 elasticloadbalancing:RemoveListenerCertificates

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 링크 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

서비스 연결 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. 를 사용하여 VPC 지원 도메인 또는 직접 쿼리 데이터 소스를 생성하면 AWS Management Console OpenSearch Service가 서비스 연결 역할을 생성합니다. 이 자동 생성이 성공하려면 사용자가 iam:CreateServiceLinkedRole 작업에 대한 권한을 보유해야 합니다.

또한 IAM 콘솔, IAM CLI 또는 IAM API를 사용하여 서비스 연결 역할을 수동으로 생성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 생성 섹션을 참조하세요.

서비스 연결 역할 편집

OpenSearch Service를 사용하여 AWSServiceRoleForAmazonOpenSearchService 서비스 연결 역할을 편집할 수 없습니다. 서비스 연결 역할을 생성한 후에는 다양한 엔터티가 역할을 참조할 수 있

기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔터티가 없도록 합니다. 단, 서비스 연결 역할을 정리해야 수동으로 삭제할 수 있습니다.

서비스 연결 역할 정리

IAM을 사용하여 서비스 연결 역할을 삭제하기 전에 먼저 역할에 활성 세션이 없는지 확인하고 역할에서 사용되는 리소스를 모두 제거해야 합니다.

IAM 콘솔에서 서비스 연결 역할에 활성 세션이 있는지 확인하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택합니다. 그런 다음 `AWSServiceRoleForAmazonOpenSearchService` 역할의 이름(확인란 아님)을 선택합니다.
3. 선택한 역할의 요약(Summary) 페이지에서 액세스 관리자(Access Advisor) 탭을 선택합니다.
4. 액세스 관리자(Access Advisor) 탭에서 서비스 연결 역할의 최근 활동을 검토합니다.

> **Note**
>
> OpenSearch Service에서 `AWSServiceRoleForAmazonOpenSearchService` 역할을 사용하는지 잘 모를 경우에는 역할을 삭제할 수 있습니다. 서비스에서 역할을 사용하는 경우에는 삭제에 실패하고 역할을 사용하는 리소스를 볼 수 있습니다. 역할이 사용 중인 경우에는 세션이 종료될 때까지 기다렸다가 역할을 삭제하거나 역할을 사용하는 리소스를 삭제할 수 있습니다. 서비스 연결 역할에 대한 세션은 취소할 수 없습니다.

서비스 연결 역할 수동 삭제

IAM 콘솔, API 또는 AWS CLI에서 서비스 연결 역할을 삭제합니다. 지침은 IAM 사용 설명서의 서비스 연결 역할 삭제 섹션을 참조하세요.

서비스 연결 역할을 사용하여 OpenSearch Serverless 컬렉션 생성

OpenSearch Serverless는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 OpenSearch Service에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 OpenSearch Service에서 사전 정의하며 사용자를 대신하여 서비스에서 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

OpenSearch Serverless는 AmazonOpenSearchServerlessServiceRole이라는 서비스 연결 역할을 사용합니다. 이 역할은 서버리스 관련 CloudWatch 지표를 계정에 게시하는 데 필요한 권한을 제공합니다. AWSServiceRoleForAmazonOpenSearchServerless와 연결된 역할 권한 정책의 이름은 `AmazonOpenSearchServerlessServiceRolePolicy`입니다. 정책에 대한 자세한 내용은 AWS 관리형 정책 참조 안내서의 AmazonOpenSearchServerlessServiceRolePolicy를 참조하세요.

OpenSearch Serverless에 대한 서비스 연결 역할 권한

OpenSearch Serverless는 AmazonOpenSearchServerlessServiceRole이라는 서비스 연결 역할을 사용하여 OpenSearch Serverless가 사용자를 대신하여 AWS 서비스를 호출할 수 있도록 합니다.

AWSServiceRoleForAmazonOpenSearchServerless 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `observability.aoss.amazonaws.com`

`AmazonOpenSearchServerlessServiceRolePolicy`라는 이름의 역할 권한 정책은 OpenSearch Serverless가 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업: 모든 AWS 리소스에 대한 `cloudwatch:PutMetricData`

> **Note**
>
> 정책에는 조건 키 `{"StringEquals": {"cloudwatch:namespace": "AWS/AOSS"}}`가 포함되어 있습니다. 즉, 서비스 연결 역할은 지표 데이터를 `AWS/AOSS` CloudWatch 네임스페이스로만 보낼 수 있습니다.

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 링크 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

OpenSearch Serverless에 대한 서비스 연결 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 AWS API에서 OpenSearch Serverless 컬렉션을 생성하면 OpenSearch Serverless가 서비스 연결 역할을 생성합니다.

> **Note**
>
> 컬렉션을 처음 생성할 때 ID 기반 정책에서 `iam:CreateServiceLinkedRole`을 할당받아야 합니다.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. OpenSearch Serverless 컬렉션을 생성하면 OpenSearch Serverless가 서비스 연결 역할을 다시 생성합니다.

IAM 콘솔을 사용해 Amazon OpenSearch Serverless 사용 사례로 서비스 연결 역할을 생성할 수도 있습니다. AWS CLI 또는 AWS API에서 서비스 이름을 사용하여 `observability.aoss.amazonaws.com` 서비스 연결 역할을 생성합니다.

```
aws iam create-service-linked-role --aws-service-name
 "observability.aoss.amazonaws.com"
```

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 생성을 참조하세요. 이 서비스 연결 역할을 삭제하면 동일한 프로세스를 사용하여 역할을 다시 생성할 수 있습니다.

OpenSearch Serverless에 대한 서비스 연결 역할 편집

은 AWSServiceRoleFor 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 엔터티가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

OpenSearch Serverless에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않는 미사용 엔터티가 없게 됩니다. 단, 서비스 링크 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

AWSServiceRoleForAmazonOpenSearchServerless를 삭제하려면 먼저에서 모든 OpenSearch Serverless 컬렉션을 삭제해야 합니다 AWS 계정.

> **Note**
>
> 리소스를 삭제하려고 할 때 OpenSearch Serverless가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 AWSServiceRoleForAmazonOpenSearchServerless 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오.

OpenSearch Serverless 서비스 연결 역할에 대해 지원되는 리전

OpenSearch Serverless는 OpenSearch Serverless를 사용할 수 있는 모든 리전에서 AmazonOpenSearchServerlessServiceRole 서비스 연결 역할 사용을 지원합니다. 지원되는 리전 목록은 일반 참조의 Amazon OpenSearch Serverless 엔드포인트 및 할당량을 참조하세요AWS 일반 참조.

서비스 연결 역할을 사용하여 OpenSearch Ingestion 파이프라인 생성

Amazon OpenSearch Ingestion은 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 OpenSearch Ingestion에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 OpenSearch Ingestion에서 사전 정의하며 사용자를 대신하여 서비스에서 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

자체 관리형 VPC를 사용하는 경우 AWSServiceRoleForOpensearchIngestionSelfManagedVpce라는 서비스 연결 역할을 사용할 때를 제외하면 OpenSearch Ingestion은 AWSServiceRoleForAmazonOpenSearchIngestionService라는 서비스 연결 역할을 사용합니다. 첨부된 정책은 계정과 OpenSearch Ingestion 간에 Virtual Private Cloud(VPC)를 생성하고 CloudWatch 지표를 계정에 게시하는 데 필요한 권한을 제공합니다.

권한

`AWSServiceRoleForAmazonOpenSearchIngestionService` 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `osis.amazon.com`

AmazonOpenSearchIngestionServiceRolePolicy라는 이름의 역할 권한 정책은 OpenSearch Ingestion이 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업: *에 대한 ec2:DescribeSubnets
- 작업: *에 대한 ec2:DescribeSecurityGroups
- 작업: *에 대한 ec2:DeleteVpcEndpoints
- 작업: *에 대한 ec2:CreateVpcEndpoint
- 작업: *에 대한 ec2:DescribeVpcEndpoints
- 작업: arn:aws:ec2:*:*:network-interface/*에 대한 ec2:CreateTags
- 작업: cloudwatch:namespace": "AWS/OSIS"에 대한 cloudwatch:PutMetricData

AWSServiceRoleForOpensearchIngestionSelfManagedVpce 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- self-managed-vpce.osis.amazon.com

OpenSearchIngestionSelfManagedVpcePolicy라는 이름의 역할 권한 정책은 OpenSearch Ingestion이 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업: *에 대한 ec2:DescribeSubnets
- 작업: *에 대한 ec2:DescribeSecurityGroups
- 작업: *에 대한 ec2:DescribeVpcEndpoints
- 작업: cloudwatch:namespace": "AWS/OSIS"에 대한 cloudwatch:PutMetricData

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 링크 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

OpenSearch Ingestion에 대한 서비스 연결 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console, AWS CLI또는 AWS API에서 OpenSearch Ingestion 파이프라인을 생성하면 OpenSearch Ingestion이 서비스 연결 역할을 생성합니다.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. OpenSearch Ingestion 파이프라인을 생성할 때 OpenSearch Ingestion은 서비스 연결 역할을 다시 생성합니다.

OpenSearch Ingestion에 대한 서비스 연결 역할 편집

DAX에서는 AWSServiceRoleForAmazonOpenSearchIngestionService 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 링크 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

OpenSearch Ingestion에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않은 미사용 엔터티가 없습니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

서비스 연결 역할을 정리

IAM을 사용하여 서비스 연결 역할을 삭제하기 전에 먼저 역할에서 사용되는 리소스를 삭제해야 합니다.

> **Note**
>
> 리소스를 삭제하려고 할 때 OpenSearch Ingestion이 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

AWSServiceRoleForAmazonOpenSearchIngestionService 또는 **AWSServiceRoleForOpensearchIngestionSelfManagedVpce** 역할에서 사용하는 OpenSearch Ingestion 리소스를 삭제하는 방법

1. Amazon OpenSearch Service 콘솔로 이동하고 수집을 선택합니다.
2. 모든 파이프라인을 삭제합니다. 지침은 the section called “파이프라인 삭제” 섹션을 참조하세요.

OpenSearch Ingestion에 대한 서비스 연결 역할 삭제

콘솔을 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할을 삭제하는 방법(콘솔)

1. IAM 콘솔로 이동합니다.
2. 역할을 선택하고 AWSServiceRoleForAmazonOpenSearchIngestionService 또는 AWSServiceRoleForOpensearchIngestionSelfManagedVpce 역할을 검색합니다.

3. 역할을 선택하고 삭제를 선택합니다.

Amazon OpenSearch Service용 샘플 코드

이 장에는 Amazon OpenSearch Service로 작업하기 위한 일반적인 샘플 코드가 포함되어 있습니다. 다양한 프로그래밍 언어로 HTTP 요청 서명, HTTP 요청 본문 압축, AWS SDK를 사용하여 도메인 생성 등이 그 예입니다.

주제

- Elasticsearch 클라이언트 호환성
- Amazon OpenSearch Service에서 HTTP 요청 압축
- AWS SDKs를 사용하여 Amazon OpenSearch Service와 상호 작용

Elasticsearch 클라이언트 호환성

최신 버전의 Elasticsearch 클라이언트에는 인위적으로 호환성을 깨뜨리는 라이선스 또는 버전 검사가 포함될 수 있습니다. 다음 표에는 OpenSearch Service와의 호환성을 극대화하기 위해 사용할 클라이언트 버전에 대한 권장 사항이 포함되어 있습니다.

> **Important**
>
> 이러한 클라이언트 버전은 최신이며 Log4j를 비롯한 최신 종속성으로 업데이트되지 않습니다. 가능하면 OpenSearch 버전의 클라이언트를 사용하는 것이 좋습니다.

클라이언트	권장 버전
Java 하위 수준 REST 클라이언트	7.13.4
Java 상위 수준 REST 클라이언트	7.13.4
Python Elasticsearch 클라이언트	7.13.4
Ruby Elasticsearch 클라이언트	7.13.3
Node.js Elasticsearch 클라이언트	7.13.0

Amazon OpenSearch Service에서 HTTP 요청 압축

gzip 압축을 사용하여 Amazon OpenSearch Service 도메인에서 HTTP 요청 및 응답을 압축할 수 있습니다. gzip 압축을 사용하면 문서 크기를 줄이고 대역폭 사용률과 대기 시간을 줄여 전송 속도를 향상시킬 수 있습니다.

gzip 압축은 OpenSearch 또는 Elasticsearch 6.0 이상을 실행하는 모든 도메인에 대해 지원됩니다. 일부 OpenSearch 클라이언트는 gzip 압축을 기본적으로 지원하며 많은 프로그래밍 언어에는 프로세스를 단순화하는 라이브러리가 있습니다.

gzip 압축 활성화

유사한 OpenSearch 설정과 혼동하지 마세요. `http_compression.enabled`는 OpenSearch Service에 고유하며 도메인에서 gzip 압축을 활성화 또는 비활성화합니다. OpenSearch 또는 Elasticsearch 7.x를 실행하는 도메인은 기본적으로 gzip 압축을 사용하지만 Elasticsearch 6.x를 실행하는 도메인은 기본적으로 기능이 비활성화되어 있습니다.

gzip 압축을 활성화하려면 다음 요청을 전송합니다.

```
PUT _cluster/settings
{
  "persistent" : {
    "http_compression.enabled": true
  }
}
```

`_cluster/settings`에 대한 요청은 압축 해제되어야 하므로 별도의 클라이언트 또는 표준 HTTP 요청을 사용하여 클러스터 설정을 업데이트해야 할 수 있습니다.

gzip 압축을 성공적으로 활성화했는지 확인하려면 다음 요청을 전송합니다.

```
GET _cluster/settings?include_defaults=true
```

응답에 다음 설정이 표시되는지 확인합니다.

```
...
"http_compression": {
  "enabled": "true"
}
...
```

필수 헤더

gzip으로 압축된 요청 본문을 포함할 때 표준 `Content-Type: application/json` 헤더를 유지하고 `Content-Encoding: gzip` 헤더를 추가합니다. gzip으로 압축된 응답을 수락하려면 `Accept-Encoding: gzip` 헤더도 추가합니다. OpenSearch 클라이언트가 gzip 압축을 지원하는 경우 이러한 헤더를 자동으로 포함할 가능성이 큽니다.

샘플 코드(Python 3)

다음 샘플에서는 opensearch-py를 사용하여 압축을 수행하고 요청을 보냅니다. 이 코드는 IAM 자격 증명을 사용하여 요청에 서명합니다.

```
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))
```

또는 적절한 헤더를 지정하고 요청 본문을 직접 압축하고 요청과 같은 표준 HTTP 라이브러리를 사용할 수 있습니다. 이 코드는 HTTP 기본 자격 증명을 사용하여 요청에 서명합니다. 세분화된 액세스 제어를 사용하는 경우 도메인에서 이 기능을 지원할 수 있습니다.

```
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)
```

AWS SDKs를 사용하여 Amazon OpenSearch Service와 상호 작용

이 섹션에는 AWS SDKs를 사용하여 Amazon OpenSearch Service 구성 API와 상호 작용하는 방법의 예가 포함되어 있습니다. 이러한 코드 샘플은 OpenSearch Service 도메인을 생성, 업데이트, 삭제하는 방법을 보여줍니다.

Java

이 섹션에는의 버전 1 및 2에 대한 예제가 포함되어 있습니다 AWS SDK for Java.

Version 2

이 예시는 AWS SDK for Java 의 버전 2에서 OpenSearchClientBuilder 생성자를 사용하여 OpenSearch 도메인을 생성하고, 도메인 구성을 업데이트하고, 도메인을 삭제합니다. `waitForDomainProcessing`에 대한 호출을 제거하여 (그리고 `deleteDomain`에 대한 호출을 언급하여) 해당 도메인이 온라인에 연결되어 사용 가능한 상태가 되도록 허용합니다.

```
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

이 예시는 AWS SDK for Java 의 버전 1에서 AWSElasticsearchClientBuilder 생성자를 사용하여 레거시 Elasticsearch 도메인을 생성하고, 도메인 구성을 업데이트하고, 도메인을 삭제합니다. `waitForDomainProcessing`에 대한 호출을 제거하여 (그리고 `deleteDomain`에 대한 호출을 언급하여) 해당 도메인이 온라인에 연결되어 사용 가능한 상태가 되도록 허용합니다.

```
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

이 예제에서는의 OpenSearchService 하위 수준 Python 클라이언트 AWS SDK for Python (Boto) 를 사용하여 도메인을 생성하고, 구성을 업데이트하고, 삭제합니다.

```
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)
```

노드

이 예시는 Node.js OpenSearch client의 JavaScript용 SDK 버전 3을 사용하여 도메인을 생성하고, 도메인 구성을 업데이트하고, 도메인을 삭제합니다.

```
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.');
            }
    };
}
```

Amazon OpenSearch Service의 데이터 인덱싱

Amazon OpenSearch Service에서는 REST API를 사용하기 때문에 문서를 인덱싱하는 방법이 무수히 많습니다. curl 같은 표준 클라이언트를 사용해도 되고, HTTP 요청을 보낼 수 있는 프로그래밍 언어를 사용해도 됩니다. OpenSearch Service는 상호 작용 과정을 한층 더 간소화하기 위해 각종 프로그래밍 언어용 클라이언트도 갖추고 있습니다. 고급 사용자는 바로 the section called “OpenSearch Service로 스트리밍 데이터 로드” 단원으로 건너뛸 수 있습니다.

Amazon OpenSearch Ingestion을 사용하여 데이터를 수집하는 것이 좋으며, 이는 OpenSearch Service 내에 구축된 완전 관리형 데이터 수집기입니다. 자세한 내용은 Amazon OpenSearch Ingestion을 참조하세요.

인덱싱에 대한 소개는 OpenSearch 설명서를 참조하세요.

인덱스에 대한 이름 지정 제약 조건

OpenSearch Service 인덱스에는 다음과 같은 이름 지정 제약 조건이 있습니다.

- 모든 문자는 소문자여야 합니다.
- 인덱스 이름은 _ 또는 -로 시작할 수 없습니다.
- 인덱스 이름에는 공백, 쉼표, :, ", *, +, /, \, |, ?, #, > 또는 <가 포함될 수 없습니다.

인덱스, 유형 또는 문서 ID 이름에는 민감한 정보를 포함하지 않습니다. OpenSearch Service는 URI(Uniform Resource Identifier)에 이러한 이름을 사용합니다. 서버 및 애플리케이션에서 흔히 HTTP 요청을 로깅하는데, 그럴 경우 URI에 민감한 정보가 포함된다면 불필요한 데이터 노출이 발생할 수 있습니다.

```
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"
```

연결된 JSON 문서를 볼 수 있는 권한이 없는 경우에도 이 가짜 로그 줄을 통해, Doe 박사의 환자 중 전화번호가 202-555-0100인 환자가 2018년에 독감에 걸린 적이 있음을 유추할 수 있습니다.

OpenSearch Service가 인덱스 이름에서 실제 또는 특정 IP 주소를 감지하는 경우(예: `my-index-12.34.56.78.91`) IP 주소를 마스킹합니다. `_cat/indices` 호출 시 다음 응답을 산출합니다.

```
green open my-index-x.x.x.x.91    soY19tBERoKo71WcEScidw 5 1 0 0   2kb  1kb
```

불필요한 혼동을 방지하기 위해, 인덱스 이름에 IP 주소를 포함하지 마십시오.

응답 크기 감소

_index 및 _bulk API의 응답에는 많은 정보가 포함되어 있습니다. 이 정보는 요청을 해결하거나 재시도 로직을 구현하는 데 유용할 수 있지만 상당한 대역폭을 사용할 수 있습니다. 이 예제에서 32바이트 문서를 인덱싱하면 339바이트의 응답이 발생합니다(헤더 포함).

```
PUT opensearch-domain/more-movies/_doc/1
{"title": "Back to the Future"}
```

응답

```
{
  "_index": "more-movies",
  "_type": "_doc",
  "_id": "1",
  "_version": 4,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 2,
    "failed": 0
  },
  "_seq_no": 3,
  "_primary_term": 1
}
```

이 응답 크기는 최소한으로 보일 수 있지만 하루에 1,000,000개의 문서(초당 약 11.5개 문서)를 인덱싱하는 경우 응답당 339바이트는 매월 10.17GB의 다운로드 트래픽으로 작동합니다.

데이터 전송 비용에 대한 우려가 있는 경우 filter_path 파라미터를 사용하여 OpenSearch Service 응답의 크기를 줄입니다. 하지만 실패한 요청을 식별하거나 재시도하는 데 필요한 필드를 필터링하지 않도록 주의합니다. 이러한 필드는 클라이언트에 따라 다릅니다. filter_path 파라미터는 모든 OpenSearch Service REST API에 대해 작동하지만 _index 및 _bulk API와 같이 자주 호출하는 API에 특히 유용합니다.

```
PUT opensearch-domain/more-movies/_doc/1?filter_path=result,_shards.total
{"title": "Back to the Future"}
```

응답

```
{
  "result": "updated",
  "_shards": {
    "total": 2
  }
}
```

필드를 포함하는 대신 - 접두사가 있는 필드를 제외할 수 있습니다. `filter_path`는 와일드카드도 지원합니다.

```
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"}
```

응답

```
{
  "errors": false,
  "items": [
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    },
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    }
  ]
}
```

인덱스 코덱

인덱스 코덱은 인덱스에 저장된 필드를 압축하여 디스크에 저장하는 방법을 결정합니다. 인덱스 코덱은 압축 알고리즘을 지정하는 정적 `index.codec` 설정에 의해 제어됩니다. 이 설정은 인덱스 샤드 크기 및 작업 성능에 영향을 줍니다.

지원되는 코덱 목록과 성능 특성은 OpenSearch 설명서의 지원되는 코덱을 참조하세요.

인덱스 코덱을 선택할 때는 다음 사항을 고려하세요.

- 기존 색인의 코덱 설정을 변경하는 문제를 피하려면 새 코덱 설정을 사용하기 전에 비프로덕션 환경에서 대표적인 워크로드를 테스트하세요. 자세한 내용은 인덱스 코덱 변경을 참조하세요.
- k-NN 또는 보안 분석 인덱스에 대해 Zstandard 압축 코덱(`"index.codec": "zstd"` 또는 `"index.codec": "zstd_no_dict"`)을 사용할 수 없습니다.

Amazon OpenSearch Service로 스트리밍 데이터 로드

타사 솔루션을 사용할 필요 없이 OpenSearch Ingestion을 사용하여 스트리밍 데이터를 Amazon OpenSearch Service 도메인으로 직접 로드할 수 있습니다. OpenSearch Ingestion으로 데이터를 보내기 위해서는 데이터 생산자를 구성하면 사용자가 지정한 도메인 또는 컬렉션에 서비스가 데이터를 자동으로 전송합니다. OpenSearch Ingestion을 시작하려면 the section called “튜토리얼: 컬렉션에 데이터 수집”(을)를 참조하세요.

OpenSearch Service를 기본으로 지원하는 Amazon Data Firehose 및 Amazon CloudWatch Logs와 같은 다른 소스를 사용하여 스트리밍 데이터 로드할 수 있습니다. Amazon S3, Amazon Kinesis Data Streams 및 Amazon DynamoDB와 같은 다른 소스는 AWS Lambda 함수를 이벤트 핸들러로 사용합니다. Lambda 함수는 새 데이터를 처리한 다음 도메인으로 스트리밍하여 응답합니다.

> **Note**
>
> Lambda는 다양한 주요 프로그래밍 언어를 지원하며, 대부분의 AWS 리전에서 사용할 수 있습니다. 자세한 내용은 개발자 안내서의 Lambda 시작하기 및 AWS Lambda 일반 참조의 AWS 서비스 엔드포인트를 참조하세요AWS 일반 참조.

OpenSearch Ingestion에서 스트리밍 데이터 로드

Amazon OpenSearch Ingestion를 사용하여 데이터를 OpenSearch Service 도메인으로 로드할 수 있습니다. 데이터 생산자를 구성하면 데이터를 OpenSearch Ingestion으로 전송하고, 이를 통해 사용자가 지정한 컬렉션에 데이터를 자동으로 전송합니다. 전송 전에 데이터를 변환하도록 OpenSearch Ingestion을 구성할 수도 있습니다. 자세한 내용은 Amazon OpenSearch Ingestion 단원을 참조하십시오.

Amazon S3에서 스트리밍 데이터 로드

Lambda를 사용하여 Amazon S3에서 OpenSearch Service 도메인으로 데이터를 전송할 수 있습니다. S3 버킷에 도착한 새 데이터는 Lambda로 이벤트 알림을 트리거한 다음 사용자 지정 코드를 실행해 인덱싱합니다.

이러한 방식의 데이터 스트리밍은 대단히 유연합니다. 객체 메타데이터를 인덱싱할 수도 있고, 객체가 일반 텍스트라면 객체 본문의 일부 요소를 구문 분석하고 인덱싱할 수도 있습니다. 이 단원에는 정규식을 이용해 로그 파일을 구문 분석하고 매치를 인덱싱하는 단순한 Python 샘플 코드가 나와 있습니다.

사전 조건

계속하려면 먼저 다음 리소스를 확보해야 합니다.

전제 조건	설명
Amazon S3 버킷	자세한 내용은 Amazon Simple Storage Service 사용 설명서에서 첫 S3 버킷 생성을 참조하세요. 버킷은 OpenSearch Service 도메인과 같은 리전에 있어야 합니다.
OpenSearch Service 도메인	Lambda 함수로 처리한 후의 데이터 대상 주소입니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 섹션을 참조하세요.

Lambda 배포 패키지 생성

배포 패키지는 코드와 종속 프로그램이 포함된 ZIP 또는 JAR 파일로 구성됩니다. 이 단원에는 Python 샘플 코드가 나와 있습니다. 다른 프로그래밍 언어는 AWS Lambda 개발자 안내서의 Lambda 배포 패키지를 참조하세요.

1. 디렉터리를 생성합니다. 이 샘플에서는 `s3-to-opensearch` 이름을 사용합니다.
2. `sample.py`라는 디렉터리에서 파일을 생성합니다.

```
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)
```

region과 host의 변수를 편집합니다.

3. 아직 설치하지 않았다면 pip를 설치한 다음, 새 package 디렉터리에 종속 항목을 설치합니다.

```
cd s3-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

모든 Lambda 실행 환경에는 Boto3가 설치되어 있으므로 배포 패키지에 이를 포함할 필요가 없습니다.

4. 애플리케이션 코드와 종속 항목을 패키지화합니다.

```
cd package
zip -r ../lambda.zip .

cd ..
zip -g lambda.zip sample.py
```

Lambda 함수 생성

배포 패키지를 만든 뒤에는 Lambda 함수를 생성할 수 있습니다. 함수를 생성할 때는 이름, 런타임(예: Python 3.8)과 IAM 역할을 선택해야 합니다. IAM 역할은 함수에 대한 권한을 정의합니다. 자세한 지침은 AWS Lambda 개발자 안내서의 콘솔로 Lambda 함수 생성을 참조하세요.

이 예제에서는 콘솔을 사용하는 것으로 가정합니다. 다음 스크린샷처럼 Python 3.9와 S3 읽기 권한 및 OpenSearch Service 쓰기 권한이 있는 역할을 선택합니다.

함수를 생성했으면 이제 트리거를 추가해야 합니다. 이 예제에서는 로그 파일이 S3 버킷에 도착할 때마다 코드를 실행하려 합니다.

1. 트리거 추가(Add trigger)를 선택하고 S3를 선택합니다.
2. 버킷을 선택합니다.

3. 이벤트 유형(Event type)에서 PUT을 선택합니다.
4. 접두사(Prefix)에는 `logs/`를 입력합니다.
5. 접미사(Suffix)에는 `.log`를 입력합니다.
6. 재귀 호출 경고를 확인하고 추가(Add)를 선택합니다.

마지막으로, 배포 패키지를 업로드할 수 있습니다.

1. 업로드 원본(Upload from)과 .zip 파일(.zip file)을 선택한 다음, 지시에 따라 배포 패키지를 업로드합니다.
2. 업로드가 완료되면 런타임 설정(Runtime settings)을 변경하고 핸들러(Handler)를 `sample.handler`로 변경합니다. 이 설정은 트리거 후 실행해야 하는 파일(`sample.py`)과 메서드(`handler`)를 Lambda에게 알려 줍니다.

이제 사용자는 완벽한 리소스 모음, 즉 로그 파일용 버킷, 로그 파일이 버킷에 추가될 때마다 실행되는 함수, 구문 분석과 인덱싱을 수행하는 코드, 검색과 시각화를 위한 OpenSearch Service 도메인을 모두 확보하게 됩니다.

Lambda 함수 테스트

함수를 만들었으면 이제 Amazon S3 버킷에 파일을 업로드해 함수를 테스트할 수 있습니다. 다음 샘플 로그 행을 이용해 `sample.log`라는 파일을 만듭니다.

```
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"
```

파일을 S3 버킷의 `logs` 폴더에 업로드합니다. 지침을 보려면 Amazon Simple Storage Service 사용 설명서에서 버킷에 객체 업로드를 참조하세요.

그런 다음 OpenSearch Service 콘솔 또는 OpenSearch Dashboards를 사용하여 `lambda-s3-index` 인덱스에 두 개의 문서가 있음을 확인합니다. 표준 검색 요청을 할 수도 있습니다.

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

Amazon Kinesis Data Streams에서 스트리밍 데이터 로드

Kinesis Data Streams에서 OpenSearch Service로 스트리밍 데이터를 로드할 수 있습니다. 데이터 스트림에 도착한 새 데이터는 Lambda로 이벤트 알림을 트리거한 다음 사용자 지정 코드를 실행해 인덱싱합니다. 이 단원에는 단순한 Python 샘플 코드가 있습니다.

사전 조건

계속하려면 먼저 다음 리소스를 확보해야 합니다.

전제 조건	설명
Amazon Kinesis Data Stream	Lambda 함수의 이벤트 소스. 자세한 내용은 Kinesis Data Streams를 참조하세요.

전제 조건	설명
OpenSearch Service 도메인	Lambda 함수로 처리한 후의 데이터 대상 주소입니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 섹션을 참조하세요.

전제 조건	설명

IAM 역할	이 역할에는 다음과 같은 기본 OpenSearch Service, Kinesis 및 Lambda 권한이 있어야 합니다.
	
	JSON
	
	{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"es:ESHttpPost",
	"es:ESHttpPut",
	"logs:CreateLogGroup",
	"logs:CreateLogStream",
	"logs:PutLogEvents",
	"kinesis:GetShardIterator",
	"kinesis:GetRecords",
	"kinesis:DescribeStream",
	"kinesis:ListStreams"
	],
	"Resource": "*"
	}
	]
	}
	
	역할은 다음과 같은 신뢰 관계를 맺고 있어야 합니다.
	
	JSON
	
	{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Principal": {
	"Service": "lambda.amazonaws.com"
	},

전제 조건	설명
	"Action": "sts:AssumeRole" }] } 자세한 내용은 IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

Lambda 함수 생성

the section called “Lambda 배포 패키지 생성”의 지침을 따르되, `kinesis-to-opensearch`라는 디렉터리를 만들고 `sample.py`에는 다음과 같은 코드를 사용합니다.

```
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.'
```

region과 host의 변수를 편집합니다.

아직 설치하지 않았다면 pip를 설치한 다음, 다음 명령을 사용하여 종속 항목을 설치합니다.

```
cd kinesis-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

이제 the section called “Lambda 함수 생성” 지침을 따르되, the section called “사전 조건”에서 IAM 역할을 지정하고 트리거에는 다음 설정을 지정합니다.

- Kinesis 스트림: 사용자의 Kinesis 스트림
- 배치 크기: 100
- 시작 위치: 수평 트리밍

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams란 무엇입니까?를 참조하세요.

이제 사용자는 완벽한 리소스 모음, 즉 Kinesis 데이터 스트림, 스트림에 새 데이터가 들어오면 실행되어 해당 데이터를 인덱싱하는 함수, 검색과 시각화를 위한 OpenSearch Service 도메인을 모두 확보하게 됩니다.

Lambda 함수 테스트

함수를 만든 뒤에는 AWS CLI에서 데이터 스트림에 새 레코드를 추가해 함수를 테스트할 수 있습니다.

```
aws kinesis put-record --stream-name test --data "My test data." --partition-key
 partitionKey1 --region us-west-1
```

그런 다음 OpenSearch Service 콘솔 또는 OpenSearch Dashboards를 사용하여 `lambda-kine-index`에 한 개의 문서가 있음을 확인합니다. 다음 요청을 사용할 수도 있습니다.

```
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"
      }
    }
  ]
}
```

Amazon DynamoDB에서 스트리밍 데이터 로드

AWS Lambda 를 사용하여 Amazon DynamoDB에서 OpenSearch Service 도메인으로 데이터를 전송할 수 있습니다. 데이터베이스 테이블에 도착한 새 데이터는 Lambda로 이벤트 알림을 트리거한 다음 사용자 지정 코드를 실행해 인덱싱합니다.

사전 조건

계속하려면 먼저 다음 리소스를 확보해야 합니다.

전제 조건	설명
DynamoDB 테이블	이 테이블에는 소스 데이터가 있습니다. 자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 테이블에 대한 기본 작업을 참조하세요. 테이블은 OpenSearch Service 도메인과 같은 리전에 위치하고 새 이미지로 설정된 스트림이 있어야 합니다. 자세한 내용은 스트림 활성화를 참조하세요.

전제 조건	설명
OpenSearch Service 도메인	Lambda 함수로 처리한 후의 데이터 대상 주소입니다. 자세한 내용은 the section called “ OpenSearch Service 도메인 생성” 섹션을 참조하세요.

전제 조건	설명
IAM 역할	이 역할에는 다음과 같은 기본 OpenSearch Service, DynamoDB 및 Lambda 실행 권한이 있어야 합니다. JSON `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["es:ESHttpPost", "es:ESHttpPut", "dynamodb:DescribeStream", "dynamodb:GetRecords", "dynamodb:GetShardIterator", "dynamodb:ListStreams", "logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents"], "Resource": "*" }] }`  역할은 다음과 같은 신뢰 관계를 맺고 있어야 합니다.  JSON  `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" },`

전제 조건	설명

	<pre>"Action": "sts:AssumeRole"
	}
	]
	}</pre>
	자세한 내용은 IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

Lambda 함수 생성

the section called “Lambda 배포 패키지 생성”의 지침을 따르되, ddb-to-opensearch라는 디렉터리를 만들고 sample.py에는 다음과 같은 코드를 사용합니다.

```
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.'
```

`region`과 `host`의 변수를 편집합니다.

아직 설치하지 않았다면 pip를 설치한 다음, 다음 명령을 사용하여 종속 항목을 설치합니다.

```
cd ddb-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

이제 the section called “Lambda 함수 생성” 지침을 따르되, the section called “사전 조건”에서 IAM 역할을 지정하고 트리거에는 다음 설정을 지정합니다.

- 테이블: 사용자의 DynamoDB 테이블
- 배치 크기: 100
- 시작 위치: 수평 트리밍

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB Streams 및 Lambda를 사용하여 새 항목 처리를 참조하세요.

이제 사용자는 완벽한 리소스 모음, 즉 소스 데이터에 대한 DynamoDB 테이블, 테이블 변경 사항의 DynamoDB 스트림, 소스 데이터가 변경되면 실행되어 이러한 변경 사항을 인덱싱하는 함수, 검색과 시각화를 위한 OpenSearch Service 도메인을 모두 확보하게 됩니다.

Lambda 함수 테스트

함수를 만들었으면 이제 AWS CLI를 사용해 DynamoDB 테이블에 새 항목을 추가해 함수를 테스트할 수 있습니다.

```
aws dynamodb put-item --table-name test --item '{"director": {"S": "Kevin
 Costner"},"id": {"S": "00001"},"title": {"S": "The Postman"}}' --region us-west-1
```

그런 다음 OpenSearch Service 콘솔 또는 OpenSearch Dashboards를 사용하여 `lambda-index`에 한 개의 문서가 있음을 확인합니다. 다음 요청을 사용할 수도 있습니다.

```
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"
        }
    }
}
```

Amazon Kinesis Data Firehose에서 스트리밍 데이터 로드

Firehose는 전송 대상으로 OpenSearch Service를 지원합니다. OpenSearch Service로 스트리밍 데이터를 로드하는 방법에 관한 지침은 Amazon Data Firehose 개발자 안내서의 Kinesis Data Firehose 전송 스트림 생성 및 OpenSearch Service를 대상으로 선택을 참조하세요.

OpenSearch Service에 데이터를 로드하기 전에, 먼저 데이터 변환을 실행해야 할 수도 있습니다. Lambda 함수로 이 태스크를 수행하는 방법에 대한 자세한 내용은 동일한 안내서의 Amazon Kinesis Data Firehose Data 데이터 변환을 참조하세요.

전송 스트림을 구성할 때 Firehose에서는 OpenSearch Service로 데이터를 보내고, Amazon S3에서 데이터를 백업하며, Lambda로 데이터를 변환할 때 필요한 리소스 액세스 권한을 가진 '원클릭' IAM 역할을 사용합니다. 이러한 역할을 수동으로 생성하려면 복잡하기 때문에, 제공된 역할을 사용하는 것이 좋습니다.

Amazon CloudWatch에서 스트리밍 데이터 로드

CloudWatch Logs 구독을 사용하면 CloudWatch Logs에서 OpenSearch Service 도메인으로 스트리밍 데이터를 로드할 수 있습니다. Amazon CloudWatch 구독에 대한 자세한 내용은 구독을 통한 로그 데이터 실시간 처리를 참조하세요. 구성에 관한 정보는 Amazon CloudWatch 개발자 안내서의 Amazon OpenSearch Service로 CloudWatch Logs 데이터 스트리밍을 참조하세요.

에서 스트리밍 데이터 로드 AWS IoT

규칙을 AWS IoT 사용하여에서 데이터를 전송할 수 있습니다. 자세한 내용은 AWS IoT 개발자 안내서의 OpenSearch 작업을 참조하세요.

Logstash를 사용하여 Amazon OpenSearch Service로 데이터 로드

Logstash의 오픈 소스 버전(Logstash OSS)에서는 대량 API를 사용하여 Amazon OpenSearch Service 도메인에 데이터를 업로드할 수 있는 편리한 방법을 제공합니다. 이 서비스는 Amazon S3 입력 플러그인을 포함하여 모든 표준 Logstash 입력 플러그인을 지원합니다. OpenSearch Service는 기본 인증 및 IAM 자격 증명을 모두 지원하는 logstash-output-opensearch 출력 플러그 인을 지원합니다. 플러그 인은 Logstash OSS 8.1 이하 버전에서 작동합니다.

구성

Logstash 구성은 도메인이 사용하는 인증 유형에 따라 다릅니다.

사용하는 인증 방법에 상관없이 구성 파일의 출력 섹션에서 `ecs_compatibility`를 `disabled`로 설정해야 합니다. Logstash 8.0은 모든 플러그인이 기본적으로 ECS 호환성 모드에서 실행되는 구분 변경을 도입했습니다. 레거시 동작을 유지하려면 기본값을 재정의해야 합니다.

세분화된 액세스 제어 구성

OpenSearch Service 도메인에서 HTTP 기본 인증을 사용하여 세분화된 액세스 제어를 사용하는 경우, 구성은 다른 OpenSearch 클러스터와 유사합니다. 이 예제 구성 파일은 Filebeat(Filebeat OSS)의 오픈 소스 버전에서 입력을 가져옵니다.

```
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
  }
}
```

구성은 Beats 애플리케이션 및 사용 사례에 따라 다르지만 Filebeat OSS 구성은 다음과 같습니다.

```
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 구성

도메인에서 IAM 기반 도메인 액세스 정책 또는 마스터 사용자의 세분화된 액세스 제어를 사용하는 경우, IAM 보안 인증 정보를 사용하여 OpenSearch Service에 대한 모든 요청에 서명해야 합니다. 다음 자격 증명 기반 정책에서는 도메인의 하위 리소스에 대한 모든 HTTP 요청을 부여합니다.

JSON

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

Logstash 구성을 설정하려면 출력을 위해 플러그인을 사용하도록 구성 파일을 변경합니다. 이 예제 구성 파일은 S3 버킷의 파일에서 입력을 가져옵니다.

```
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
  }
}
```

구성 파일 내에 IAM 자격 증명을 제공하지 않으려면 해당 자격 증명을 내보낼 수 있습니다(또는 `aws configure`를 실행).

```
export AWS_ACCESS_KEY_ID="your-access-key"
export AWS_SECRET_ACCESS_KEY="your-secret-key"
export AWS_SESSION_TOKEN="your-session-token"
```

OpenSearch Service 도메인이 VPC에 있는 경우, Logstash OSS 머신이 VPC에 연결되고 VPC 보안 그룹을 통해 도메인에 액세스할 수 있어야 합니다. 자세한 내용은 the section called “VPC 도메인 액세스 정책에 대하여” 섹션을 참조하세요.

Amazon OpenSearch Service의 데이터 검색

URI 검색 및 요청 본문 검색을 포함하여 Amazon OpenSearch Service에서 문서를 검색하는 몇 가지 일반적인 방법이 있습니다. OpenSearch Service는 사용자 지정 패키지, SQL 지원 및 비동기 검색과 같은 검색 환경을 개선하는 추가 기능을 제공합니다. 포괄적인 OpenSearch 검색 API 참조는 OpenSearch 설명서를 참조하세요.

> **Note**
>
> 다음 샘플 요청은 OpenSearch API에서 작동합니다. 일부 요청은 이전 버전의 Elasticsearch에서 작동하지 않을 수 있습니다.

주제

- URI 검색
- 요청 본문 검색
- 검색 결과 페이지 매김
- Dashboards Query Language
- Amazon OpenSearch Service에서 패키지 가져오기 및 관리
- SQL을 사용하여 Amazon OpenSearch Service 데이터 쿼리
- Amazon OpenSearch Service의 교차 클러스터 검색
- Amazon OpenSearch Service용 순위 학습
- Amazon OpenSearch Service의 비동기 검색
- Amazon OpenSearch Service의 특정 시점 검색
- Amazon OpenSearch Service의 의미 검색
- Amazon OpenSearch Service에서 동시 세그먼트 검색
- Amazon OpenSearch Service에서 자연어 쿼리 생성

URI 검색

URI(Universal Resource Identifier) 검색은 가장 단순한 형태의 검색입니다. URI 검색에서는 쿼리를 HTTP 요청 파라미터로 지정합니다.

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/_search?q=house
```

샘플 응답은 다음과 같습니다.


```
{
  "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"
        }
      },
      ...
    ]
  }
}
```

기본적으로 이 쿼리는 모든 색인의 모든 필드에서 검색어 house를 검색합니다. 검색 범위를 좁히려면 URI에서 색인(movies) 및 문서 필드(title)를 지정합니다.

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?q=title:house
```

이 요청에 추가 파라미터를 포함할 수 있지만, 지원되는 파라미터는 OpenSearch 검색 옵션을 일부만 제공합니다. 다음 요청은 20개 결과(기본 개수 10개가 아님)를 반환하고 연도 기준으로 정렬합니다(_score 기준이 아님).

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?
q=title:house&size=20&sort=year:desc
```

요청 본문 검색

더욱 복잡한 검색을 수행하려면 쿼리에 HTTP 요청 본문 및 OpenSearch DSL(Domain-Specific Language)을 사용합니다. 쿼리 DSL을 사용하면 전체 범위의 OpenSearch 검색 옵션을 지정할 수 있습니다.

> **Note**
>
> 텍스트 필드 값에 유니코드 특수 문자를 포함할 수 없습니다. 포함하면 값이 특수 문자로 구분된 여러 값으로 구문 분석됩니다. 이렇게 잘못된 구문 분석으로 인해 의도하지 않은 문서 필터링이 발생하여 문서 액세스에 대한 제어가 손상될 수 있습니다. 자세한 내용은 OpenSearch 설명서의 텍스트 필드의 유니코드 특수 문자에 대한 참고 사항을 참조하세요.

다음 match 쿼리는 마지막 URI 검색 예제와 유사합니다.

```
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 API는 요청 본문 검색에 HTTP GET 및 POST를 허용하지만, 모든 HTTP 클라이언트가 GET 요청에 요청 본문을 추가하는 것을 지원하지는 않습니다. POST가 더욱 범용적 선택입니다.

많은 경우에 전체 필드는 아니지만 여러 필드를 검색해야 합니다. multi_match 쿼리를 사용합니다.

```
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"]
    }
  }
}
```

필드 부스팅

특정 필드를 "부스팅"하여 검색 관련성을 개선할 수 있습니다. 부스트는 한 필드의 일치 항목을 다른 필드의 일치 항목보다 가중하는 배수입니다. 다음 예제에서 `title` 필드의 john에 대한 일치 항목은 `plot` 필드의 일치 항목보다 두 배, `actors` 또는 `directors` 필드의 일치 항목보다 네 배 많이 `_score`에 영향을 미칩니다. 그러면 결과에서 John Wick, John Carter 같은 영화는 검색 결과의 거의 맨 위에 있고, John Travolta가 주인공인 영화는 거의 맨 아래에 있습니다.

```
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"]
    }
  }
}
```

검색 결과 강조 표시

`highlight` 옵션은 쿼리가 하나 이상의 필드와 일치할 경우 `hits` 배열 내에 추가 객체를 반환하도록 OpenSearch에 지시합니다.

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

쿼리가 `plot`의 내용과 일치할 경우 히트는 다음과 같이 표시됩니다.

```
{
  "_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>."
    ]
  }
}
```

기본적으로 OpenSearch는 일치 문자열을 태그로 묶고, 일치 항목 주위로 최대 100자의 컨텍스트를 제공하고, 마침표, 공백, 줄바꿈을 식별하여 내용을 분할합니다. 이러한 설정은 모두 사용자 지정이 가능합니다.

```
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": ".,!? "
  }
}
```

Count API

문서 내용에는 관심이 없고 일치 항목 수만 알고 싶은 경우 _search API 대신 _count API를 사용할 수 있습니다. 다음 요청에서는 query_string 쿼리를 사용하여 로맨틱 코미디를 식별합니다.

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_count
{
  "query": {
    "query_string": {
      "default_field": "genres",
      "query": "romance AND comedy"
    }
  }
}
```

샘플 응답은 다음과 같습니다.

```
{
  "count": 564,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
```

```
  }
}
```

검색 결과 페이지 매김

많은 수의 검색 결과를 표시해야 하는 경우 파라미터를 사용하여 페이지 매김을 구현할 수 있습니다.

특정 시점

PIT(특정 시점) 기능은 고정된 데이터 세트에 대해 다양한 쿼리를 실행할 수 있는 검색 유형입니다. 이는 OpenSearch에서 선호되는 페이지 매김 메서드이며, 특히 심층 페이지 매김의 경우 더욱 그렇습니다. PIT는 OpenSearch Service 버전 2.5 이상에서 사용할 수 있습니다. ACL에 대한 자세한 내용은 ??? 단원을 참조하세요.

`from` 파라미터를 `size` 추가합니다.

페이지를 매기는 가장 간단한 방법은 `from` 및 `size` 파라미터를 사용하는 것입니다. 다음 요청은 검색 결과의 0 기반 인덱스 목록에서 20~39개 결과를 반환합니다.

```
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"]
    }
  }
}
```

검색 페이지 매김에 대한 자세한 내용은 OpenSearch 설명서의 결과 페이지 매김을 참조하세요.

Dashboards Query Language

Dashboards Query Language(DQL)를 사용하여 OpenSearch Dashboards에서 데이터와 시각화를 검색할 수 있습니다. DQL은 용어, 부울, 날짜 및 범위, 중첩 필드의 4가지 기본 쿼리 유형을 사용합니다.

용어 쿼리

용어 쿼리를 사용하려면 검색하려는 용어를 지정해야 합니다.

용어 쿼리를 수행하려면 다음을 입력합니다.

```
host:www.example.com
```

부울 쿼리

부울 연산자 AND, or 및 not을 사용하여 여러 쿼리를 결합할 수 있습니다.

부울 쿼리를 수행하려면 다음을 붙여 넣습니다.

```
host.keyword:www.example.com and response.keyword:200
```

날짜 및 범위 쿼리

날짜 및 범위 쿼리를 사용하여 쿼리 전후의 날짜를 찾을 수 있습니다.

- >는 지정한 날짜 후의 날짜 검색을 나타냅니다.
- <는 지정한 날짜 전의 날짜 검색을 나타냅니다.

```
@timestamp > "2020-12-14T09:35:33"
```

중첩 필드 쿼리

문서에 중첩 필드가 있는 경우 검색할 문서 부분을 지정해야 합니다. 다음은 중첩 필드가 있는 샘플 문서입니다.

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

DQL을 사용하여 특정 필드를 검색하려면 다음을 붙여 넣습니다.

```
NBA players: {player-name: Lebron James}
```

중첩 문서에서 여러 객체를 검색하려면 다음을 붙여 넣습니다.

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo}
```

범위 내에서 검색하려면 다음을 붙여 넣습니다.

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo and < 30}
```

문서에 다른 객체 내에 중첩된 객체가 있는 경우에도 모든 수준을 지정하여 데이터를 검색할 수 있습니다. 이렇게 하려면 다음을 붙여 넣습니다.

```
Top-Power-forwards.NBA players: {player-name:Lebron James}
```

Amazon OpenSearch Service에서 패키지 가져오기 및 관리

Amazon OpenSearch Service를 사용하면 중지 단어 및 동의어와 같은 사용자 지정 사전 파일을 업로드하고 플러그인을 도메인과 연결할 수 있습니다. 이러한 플러그인은 사전 패키징된 플러그인, 사용자 지정 플러그인 또는 타사 플러그인일 수 있으므로 도메인의 기능을 확장할 수 있는 유연성을 제공합니다. 이러한 모든 유형의 파일에 대한 일반 용어는 패키지입니다.

- 사전 파일은 OpenSearch에 일반적인 고주파 단어를 무시하거나 "Frozen custard", "gelato", "ice cream"과 같은 유사한 용어를 동등하게 처리하도록 지시하여 검색 결과를 구체화하는 데 도움이 됩니다. 또한 일본어(쿠로모지) 분석 플러그인에서 볼 수 있듯이 어간 추출을 개선할 수 있습니다.
- 사전 패키징된 플러그인은 맞춤형 검색 결과를 위한 Amazon Personalize 플러그인과 같은 기본 기능을 제공합니다. 이러한 플러그인은 `ZIP-PLUGIN` 패키지 유형을 사용합니다. 자세한 내용은 the section called “엔진 버전별 플러그인” 단원을 참조하십시오.

- 사용자 지정 및 타사 플러그인을 사용하면 맞춤형 기능을 추가하거나 외부 시스템과 통합할 수 있으므로 도메인에 훨씬 더 많은 유연성을 제공할 수 있습니다. 사전 패키징된 플러그인과 마찬가지로 사용자 지정 플러그인을 `ZIP-PLUGIN` 패키지로 업로드합니다. 타사 플러그인의 경우 플러그인 라이선스 및 구성 파일도 별도의 패키지로 가져온 다음 모두 도메인과 연결해야 합니다.

  자세한 정보는 다음의 주제를 참조하세요.

  - the section called “사용자 지정 플러그인”
  - the section called “타사 플러그인”

> **Note**
>
> 최대 20개의 플러그인을 단일 도메인에 연결할 수 있습니다. 이 제한에는 선택 사항, 타사 및 사용자 지정 플러그인과 같은 모든 플러그인 유형이 포함됩니다.

주제

- 필수 권한
- Amazon S3에 패키지 업로드
- 패키지 가져오기 및 연결
- OpenSearch에서 사용자 정의 패키지 사용
- 사용자 지정 패키지 업데이트
- 새 사전으로 인덱스 수동 업데이트
- 패키지 분리 및 제거
- Amazon OpenSearch Service에서 사용자 지정 플러그인 관리
- Amazon OpenSearch Service에 타사 플러그인 설치

필수 권한

관리자 액세스 권한이 없는 사용자는 패키지를 관리하려면 특정 AWS Identity and Access Management (IAM) 작업이 필요합니다.

- `es:CreatePackage` - 패키지 생성
- `es:DeletePackage` - 패키지 삭제
- `es:AssociatePackage` - 패키지를 도메인에 연결

- `es:DissociatePackage` - 도메인에서 패키지 연결 해제

또한 사용자 지정 패키지가 상주하는 Amazon S3 버킷 경로 또는 객체에 대한 권한도 필요합니다.

도메인 액세스 정책이 아닌 IAM 내에서 모든 권한을 부여합니다. 자세한 내용은 the section called “ID 및 액세스 관리” 섹션을 참조하세요.

Amazon S3에 패키지 업로드

이 섹션에서는 사전 패키징된 플러그인 패키지가 이미 설치되어 있으므로 사용자 지정 사전 패키지를 업로드하는 방법을 다룹니다. 패키지를 도메인에 연결하려면 먼저 Amazon S3 버킷에 패키지를 업로드해야 합니다. 지침은 Amazon Simple Storage Service 사용 설명서에서 객체 업로드를 참조하세요. 지원되는 플러그인을 업로드할 필요가 없습니다.

패키지에 민감한 정보가 포함되어 있는 경우 업로드할 때 S3 관리형 키를 사용한 서버 측 암호화를 지정합니다. OpenSearch Service는 AWS KMS 키를 사용하여 보호하는 S3의 파일에 액세스할 수 없습니다.

파일을 업로드한 후 S3 경로를 기록해 둡니다. 경로 형식은 `s3://`*`amzn-s3-demo-bucket`*`/`*`file-path`*`/`*`file-name`*입니다.

테스트 용도로 다음 동의어 파일을 사용할 수 있습니다. `synonyms.txt`로 파일을 저장합니다.

```
danish, croissant, pastry
ice cream, gelato, frozen custard
sneaker, tennis shoe, running shoe
basketball shoe, hightop
```

Hunspell 사전과 같은 특정 사전은 여러 파일을 사용하며 파일 시스템에 자체 디렉터리가 필요합니다. 현재 OpenSearch Service에서는 단일 파일 사전만 지원합니다.

패키지 가져오기 및 연결

사용자 지정 사전을 OpenSearch Service로 가져올 수 있는 가장 간단한 방법은 콘솔을 사용하는 것입니다. Amazon S3에서 패키지를 가져오면 OpenSearch Service가 패키지의 자체 복사본을 저장하고 OpenSearch Service 관리형 키와 AES-256을 사용하여 해당 복사본을 자동으로 암호화합니다.

선택적 플러그인은 이미 OpenSearch Service에 사전 설치되어 있으므로 직접 업로드할 필요는 없지만 플러그인을 도메인에 연결해야 합니다. 사용 가능한 플러그인은 콘솔의 패키지 화면에 나열되어 있습니다.

패키지를 가져와 도메인에 연결

1. Amazon OpenSearch Service 콘솔에서 패키지(Packages)를 선택합니다.
2. [패키지 가져오기(Import package)]를 선택합니다.
3. 패키지에 설명이 포함된 이름을 지정합니다.
4. 파일에 대한 S3 경로를 지정한 다음 가져오기(Import)를 선택합니다.
5. 패키지(Packages) 화면으로 돌아갑니다.
6. 패키지 상태가 사용 가능(Available)인 경우 패키지를 선택합니다.
7. 그런 다음 도메인에 연결을 선택합니다.
8. 도메인을 선택한 후 다음을 선택합니다. 패키지를 검토하고 연결을 선택합니다.
9. 탐색 창에서 해당하는 도메인을 선택하고 패키지(Packages) 탭으로 이동합니다.
10. 패키지가 사용자 지정 사전인 경우 패키지가 사용 가능한 상태가 되면 ID를 기록해 두세요. OpenSearch에 대한 요청에서 `analyzers/id`를 파일 경로로 사용합니다.

OpenSearch에서 사용자 정의 패키지 사용

이 섹션에서는 사용자 지정 사전과 사전 패키징된 플러그인 등 두 가지 유형의 패키지를 모두 사용하는 방법을 다룹니다.

사용자 지정 사전

파일을 도메인에 연결한 후 토큰화기 및 토큰 필터를 생성할 `user_dictionary` 때 `synonyms_pathstopwords_path`, 및와 같은 파라미터에 사용할 수 있습니다. 정확한 파라미터는 객체에 따라 다릅니다. 몇 가지 객체는 `synonyms_path` 및 `stopwords_path`를 지원하지만 `user_dictionary`는 kuromoji 플러그인에만 사용됩니다.

IK(중국어) 분석 플러그인의 경우 사용자 지정 사전 파일을 사용자 지정 패키지로 업로드하고 도메인에 연결할 수 있으며 `user_dictionary` 파라미터를 요구하지 않고 플러그인이 파일을 자동으로 선택합니다. 파일이 동의어 파일인 경우 `synonyms_path` 파라미터를 사용합니다.

다음 예제는 새 인덱스에 동의어 파일을 추가합니다.

```
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"
      }
    }
  }
}
```

이 요청은 표준 토크나이저 및 동의어 토큰 필터를 사용하는 인덱스에 대한 사용자 지정 분석기를 생성합니다.

- 토크나이저는 일련의 규칙에 따라 문자 스트림을 토큰(일반적으로 단어)으로 나눕니다. 가장 간단한 예는 공백 문자를 만날 때마다 앞의 문자를 토큰으로 나누는 공백 토크나이저입니다. 더욱 복잡한 예는 여러 언어에 걸쳐 일련의 문법 기반 규칙을 사용하는 표준 토크나이저입니다.
- 토큰 필터는 토큰을 추가, 수정 또는 삭제합니다. 예를 들어 동의어 토큰 필터는 동의어 목록에서 단어를 찾으면 토큰을 추가합니다. 중단 토큰 필터는 불용어 목록의 단어를 찾으면 토큰을 제거합니다.

또한 이 요청은 텍스트 필드(`description`)를 매핑에 추가하고 OpenSearch에서 검색 분석기로 새 분석기를 사용하도록 지정합니다. 여전히 표준 분석기를 인덱스 분석기로 사용할 수 있습니다.

마지막으로 토큰 필터에 `"updateable": true`줄을 기록해 둡니다. 이 필드는 인덱스 분석기가 아닌 검색 분석기에만 적용되며 추후 자동으로 검색 분석기를 업데이트하고자 할 때 중요합니다.

테스트를 위해 인덱스에 몇 가지 문서를 추가합니다.

```
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" }
```

그런 다음 동의어를 사용하여 문서를 검색하세요.

```
GET my-index/_search
{
  "query": {
    "match": {
      "description": "gelato"
    }
  }
}
```

이 경우 OpenSearch는 다음과 같은 응답을 반환합니다.

```
{
  "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**
>
> 사전 파일은 크기에 비례하여 Java 힙 공간을 사용합니다. 예를 들어, 2GiB 사전 파일은 노드에서 2GiB의 힙 공간을 사용할 수 있습니다. 큰 파일을 사용하는 경우 노드에 해당 파일을 수용할 수 있는 충분한 힙 공간이 있는지 확인합니다. `JVMMemoryPressure` 지표를 모니터링하고 필요에 따라 클러스터를 확장합니다.

사전 패키징된 플러그인 사용

OpenSearch Service를 사용하면 사전 설치된 선택적 OpenSearch 플러그인을 도메인과 연결하여 사용할 수 있습니다. 사전 패키징된 플러그인 패키지는 특정 OpenSearch 버전과 호환되며 해당 버전의 도메인에만 연결할 수 있습니다. 도메인에 사용할 수 있는 패키지 목록에는 도메인 버전과 호환되는 모든 지원 플러그인이 포함됩니다. 플러그인을 도메인에 연결한 후에 도메인에서의 설치 프로세스가 시작됩니다. 그러면 OpenSearch Service에 요청할 때 플러그인을 참조하고 사용할 수 있습니다.

플러그인을 연결하고 분리하려면 블루/그린 배포가 필요합니다. 자세한 내용은 the section called “블루/그린 배포의 원인이 되는 변경 사항” 단원을 참조하십시오.

선택적 플러그인에는 언어 분석기 및 사용자 지정 검색 결과가 포함됩니다. 예를 들어 Amazon Personalize Search Ranking 플러그인은 기계 학습을 사용하여 고객을 위한 검색 결과를 개인화합니다. 이 플러그인에 대한 자세한 내용은 OpenSearch의 검색 결과 개인 설정을 참조하세요. 지원되는 인스턴스 전체 목록은 the section called “엔진 버전별 플러그인” 섹션을 참조하세요.

Sudachi 플러그인

Sudachi 플러그인의 경우 사전 파일을 다시 연결해도 도메인에 즉시 반영되지 않습니다. 구성 변경 또는 기타 업데이트의 일환으로 도메인에서 다음 블루/그린 배포가 실행되면 사전이 새로 고쳐집니다. 또는 업데이트된 데이터로 새 패키지를 생성하고 이 새 패키지를 사용하여 새 인덱스를 생성하며 기존 인덱스를 새 인덱스로 다시 인덱싱한 후 이전 인덱스를 삭제할 수 있습니다. 재인덱싱 방식을 사용하려는 경우 트래픽이 중단되지 않도록 인덱스 별칭을 사용하세요.

또한 Sudachi 플러그인은 CreatePackage API 작업을 통해 업로드할 수 있는 바이너리 Sudachi 사전만 지원합니다. 사전 빌드된 시스템 사전 및 사용자 사전 컴파일 프로세스에 대한 자세한 내용은 Sudachi 설명서를 참조하세요.

다음 예제는 Sudachi 토큰화에서 시스템 및 사용자 사전을 사용하는 방법을 보여줍니다. 이러한 사전을 TXT-DICTIONARY 유형의 사용자 지정 패키지로 업로드하고 추가 설정에서 해당 패키지 ID를 제공해야 합니다.

```
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"
          }
        }
      }
    }
  }
}
```

사용자 지정 패키지 업데이트

이 섹션에서는 사전 패키징된 플러그인 패키지가 이미 업데이트되어 있으므로 사용자 지정 사전 패키지를 업데이트하는 방법만 다룹니다. Amazon S3에 새 버전의 패키지를 업로드해도 Amazon OpenSearch Services에 패키지가 자동으로 업데이트되지 않습니다. OpenSearch Service는 파일의 자체 복사본을 저장하므로 새 버전을 S3에 업로드하는 경우 수동으로 업데이트해야 합니다.

연결된 각 도메인은 해당 파일의 자체 복사본도 저장합니다. 검색 동작을 예측할 수 있게 유지하기 위해 도메인은 명시적으로 업데이트할 때까지 현재 패키지 버전을 계속 사용합니다. 사용자 지정 패키지를 업데이트하려면에서 파일을 수정하고 OpenSearch Service에서 패키지를 Amazon S3 Control업데이트한 다음 업데이트를 적용합니다.

콘솔

1. OpenSearch Service 콘솔에서 패키지(Packages)를 선택합니다.
2. 패키지를 선택하고 [업데이트(Update)]를 선택합니다.
3. 파일에 대한 새 S3 경로를 제공한 다음 패키지 업데이트를 선택합니다.
4. 패키지(Packages) 화면으로 돌아갑니다.
5. 패키지 상태가 사용 가능(Available)으로 변경되면 패키지를 선택합니다. 그런 다음 하나 이상의 연결된 도메인을 선택하고 업데이트 적용(Apply update)을 선택한 다음, 확인합니다. 연결 상태가 활성(Active)으로 변경될 때까지 기다립니다.
6. 다음 단계는 인덱스를 구성한 방법에 따라 달라집니다.

   - 도메인에서 OpenSearch 또는 Elasticsearch 7.8 이상을 실행하고 업데이트 가능 필드가 true로 설정된 검색 분석기만을 사용하는 경우 추가 작업은 필요하지 않습니다. OpenSearch Service는 _plugins/_refresh_search_analyzers API를 사용하여 인덱스를 자동으로 업데이트합니다.
   - 도메인에서 Elasticsearch 7.7 이하를 실행하거나 인덱스 분석기를 사용하거나 `updateable` 필드를 사용하지 않는 경우 the section called “새 사전으로 인덱스 수동 업데이트” 섹션을 참조하세요.

콘솔이 가장 간단한 방법이지만 AWS CLI, , SDKs 또는 구성 API를 사용하여 OpenSearch Service 패키지를 업데이트할 수도 있습니다. 자세한 내용은 AWS CLI 명령 참조 및 Amazon OpenSearch Service API 참조를 참조하세요.

AWS SDK

콘솔에서 패키지를 수동으로 업데이트하는 대신 SDK를 사용하여 업데이트 프로세스를 자동화할 수 있습니다. 다음 샘플 Python 스크립트는 Amazon S3 새 패키지 파일을 업로드하고 OpenSearch Service에서 패키지를 업데이트한 다음 새 패키지를 지정된 도메인에 적용합니다. 업데이트가 성공했음을 확인한 후 OpenSearch에 대한 샘플 호출을 통해 새 동의어가 적용되었음을 보여줍니다.

`host`, `region`, `file_name`, `bucket_name`, `s3_key`, `package_id`, `domain_name`, `query`의 값을 입력해야 합니다.

```
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**
>
> 를 사용하여 스크립트를 실행할 때 "패키지 찾을 수 없음" 오류가 발생하면 Boto3가 S3 버킷이 있는 리전이 아닌 ~/.aws/config에 지정된 리전을 사용하고 있음을 AWS CLI의미할 수 있습니다. `aws configure`를 실행하고 올바른 리전을 지정하거나 다음 클라이언트에 리전을 명시적으로 추가하세요.

```
client = boto3.client('opensearch', region_name='us-east-1')
```

새 사전으로 인덱스 수동 업데이트

수동 인덱스 업데이트는 사전 패키징된 플러그인이 아닌 사용자 지정 사전에만 적용됩니다. 업데이트된 패키지를 사용하려면 다음 조건 중 하나를 충족하는 경우 인덱스를 수동으로 업데이트해야 합니다.

- 도메인에서 Elasticsearch 7.7 이전 버전이 실행됩니다.
- 사용자 지정 패키지를 인덱스 분석기로 사용합니다.
- 사용자 지정 패키지를 검색 분석기로 사용하지만 업데이트 가능 필드를 포함하지 않습니다.

새 패키지 파일로 분석기를 업데이트하기 위해 다음 두 가지 옵션을 사용할 수 있습니다.

- 업데이트하려는 인덱스를 모두 닫고 엽니다.

```
POST my-index/_close
POST my-index/_open
```

- 인덱스를 다시 생성합니다. 먼저 업데이트된 동의어 파일(또는 전혀 새로운 파일)을 사용하는 인덱스를 생성합니다. UTF-8 버전만 지원됩니다.

```
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"
      }
    }
  }
}
```

그런 다음 이전 인덱스를 새 인덱스로 다시 생성합니다.

```
POST _reindex
{
  "source": {
    "index": "my-index"
  },
  "dest": {
    "index": "my-new-index"
  }
}
```

인덱스 분석기를 자주 업데이트하는 경우 인덱스 별칭을 사용하여 최신 인덱스에 대한 일관된 경로를 유지합니다.

```
POST _aliases
{
  "actions": [
    {
      "remove": {
        "index": "my-index",
        "alias": "latest-index"
      }
    },
    {
      "add": {
        "index": "my-new-index",
        "alias": "latest-index"
```

```
      }
    }
  ]
}
```

이전 인덱스가 필요하지 않으면 삭제합니다.

```
DELETE my-index
```

패키지 분리 및 제거

사용자 지정 사전 플러그인이든 사전 패키지 플러그인이든 상관없이 패키지를 도메인에서 연결 해제하면 새 인덱스를 생성할 때 해당 패키지를 더 이상 사용할 수 없습니다. 패키지가 분리되면 패키지를 사용하던 기존 인덱스는 더 이상 사용할 수 없습니다. 패키지를 분리하려면 먼저 인덱스에서 패키지를 제거해야 합니다. 그렇지 않으면 분리되지 않습니다.

도메인에서 패키지를 분리하고 OpenSearch Service에서 제거하는 가장 간단한 방법은 콘솔을 사용하는 것입니다. OpenSearch Service에서 패키지를 제거하더라도 Amazon S3 원래 위치에서 패키지가 제거되지 않습니다.

도메인에서 패키지 연결 해제

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 탐색 창에서 도메인을 선택합니다.
3. 도메인을 선택한 다음 패키지 탭으로 이동합니다.
4. 패키지, 작업(Actions), 분리(Dissociate)를 차례로 선택합니다. 선택 내용을 확인합니다.
5. 패키지가 목록에서 사라질 때까지 기다립니다. 브라우저를 새로 고쳐야 할 수 있습니다.
6. 패키지를 다른 도메인에 사용하려면 여기에서 작업을 중지합니다. 계속해서 패키지를 제거하려면 탐색 창에서 패키지를 선택합니다.
7. 패키지를 선택하고 삭제(Delete)를 선택합니다.

또는 AWS CLI, SDKs 또는 구성 API를 사용하여 패키지를 분리하고 제거합니다. 자세한 내용은 AWS CLI 명령 참조 및 Amazon OpenSearch Service API 참조를 참조하세요.

Amazon OpenSearch Service에서 사용자 지정 플러그인 관리

OpenSearch Service용 사용자 지정 플러그인을 사용하면 언어 분석, 사용자 지정 필터링, 순위 등의 영역에서 OpenSearch 기능을 확장하여 개인화된 검색 경험을 만들 수 있습니다. OpenSearch용 사용자 지정 플러그인은 `org.opensearch.plugins.Plugin` 클래스를 확장한 다음 `.zip` 파일로 패키징하여 개발할 수 있습니다.

다음 플러그인 확장은 현재 Amazon OpenSearch Service에서 지원됩니다.

- AnalysisPlugin - 텍스트 처리를 위한 사용자 지정 분석기, 캐릭터 토큰화기 또는 필터를 추가하여 분석 기능을 확장합니다.
- SearchPlugin - 사용자 지정 쿼리 유형, 유사성 알고리즘, 제안 옵션 및 집계를 사용하여 검색 기능을 개선합니다.
- MapperPlugin - OpenSearch에서 사용자 지정 필드 유형 및 매핑 구성을 생성할 수 있으므로 다양한 유형의 데이터를 저장하고 인덱싱하는 방법을 정의할 수 있습니다.
- ScriptPlugin – OpenSearch에 사용자 지정 스크립팅 기능을 추가할 수 있습니다. 예를 들어 검색 또는 인덱싱 중에 점수, 정렬 및 필드 값 변환과 같은 작업을 위한 사용자 지정 스크립트를 추가할 수 있습니다.

사용자 지정 패키지에 OpenSearch Service 콘솔 또는 기존 API 명령을 사용하여 플러그인을 업로드하고 Amazon OpenSearch Service 클러스터와 연결할 수 있습니다. DescribePackages 명령을 사용하여 계정의 모든 패키지를 설명하고 OpenSearch 버전 및 오류 세부 정보와 같은 세부 정보를 볼 수도 있습니다. OpenSearch Service는 플러그인 패키지의 버전 호환성, 보안 취약성 및 허용된 플러그인 작업을 검증합니다. 사용자 지정 패키지에 대한 자세한 내용은 섹션을 참조하세요the section called “패키지”.

OpenSearch 버전 및 AWS 리전 지원

사용자 지정 플러그인은 AWS 리전다음에서 OpenSearch 버전 2.15를 실행하는 OpenSearch Service 도메인에서 지원됩니다.

- 미국 동부(오하이오)(us-east-2)
- 미국 동부(버지니아 북부)(us-east-1)
- 미국 서부(오레곤)(us-west-2)
- 아시아 태평양(뭄바이)(ap-south-1)
- 아시아 태평양(서울)(ap-northeast-2)
- 아시아 태평양(싱가포르)(ap-southeast-1)

- 아시아 태평양(시드니)(ap-southeast-2)
- 아시아 태평양(도쿄)(ap-northeast-1)
- 캐나다(중부)(ca-central-1)
- 유럽(프랑크푸르트)(eu-central-1)
- 유럽(아일랜드)(eu-west-1)
- 유럽(런던) (eu-west-2)
- 유럽(파리)(eu-west-3)
- 남아메리카(상파울루)(sa-east-1)

> **Note**
>
> 사용자 지정 플러그인에는 사용자가 개발한 코드가 포함되어 있습니다. 사용자가 개발한 코드로 인한 SLA 위반을 포함한 모든 문제는 SLA 크레딧을 받을 수 없습니다. 자세한 내용은 Amazon OpenSearch Service - 서비스 수준 계약을 참조하세요.

주제

- 플러그인 할당량
- 사전 조건
- 문제 해결
- 콘솔을 사용하여 사용자 지정 플러그인 설치
- 를 사용하여 사용자 지정 플러그인 관리 AWS CLI
- Amazon OpenSearch Service 사용자 지정 패키지 AWS KMS 통합

플러그인 할당량

- 리전별로 계정당 최대 25개의 사용자 지정 플러그인을 생성할 수 있습니다.
- 플러그인의 최대 비압축 크기는 1GB입니다.
- 단일 도메인과 연결할 수 있는 최대 플러그인 수는 20개입니다. 이 할당량은 선택 사항, 타사 및 사용자 지정 등 결합된 모든 플러그인 유형에 적용됩니다.
- 사용자 지정 플러그인은 OpenSearch 버전 2.15 이상을 실행하는 도메인에서 지원됩니다.
- 플러그인의 `descriptor.properties` 파일은 패치 버전이 0으로 설정된 2.15.0 또는 2.x.x 버전과 유사한 엔진 버전을 지원해야 합니다.

사전 조건

사용자 지정 플러그인을 설치하고 도메인에 연결하기 전에 다음 요구 사항을 충족해야 합니다.

- `descriptor.properties` 파일의 플러그인에 대해 지원되는 엔진 버전은 `2.15.0` 또는와 유사해야 합니다`2.x.0`. 즉, 패치 버전은 `0`이어야 합니다.
- 도메인에서 다음 기능을 활성화해야 합니다.
  - 노드 간 암호화
  - 저장된 데이터 암호화
  - `EnforceHTTPS`가 'true'로 설정됨

    AWS Config 개발자 안내서의 opensearch-https-required도 참조하세요.
  - 클라이언트는 Policy-Min-TLS-1-2-PFS-2023-10을 지원해야 합니다. 다음 명령을 사용하여이 지원을 지정할 수 있습니다. *## ### ##* 자신의 정보로 바꿉니다.

    ```
    aws opensearch update-domain-config \
        --domain-name domain-name \
        --domain-endpoint-options '{"TLSSecurityPolicy":"Policy-Min-TLS-1-2-
    PFS-2023-10" }'
    ```

    자세한 내용은 Amazon OpenSearch Service API 참조의 DomainEndpointOptions를 참조하세요. OpenSearch

문제 해결

시스템에서 오류를 반환하는 경우 `PluginValidationFailureReason : The provided plugin could not be loaded`문제 해결 정보는 섹션을 참조the section called “버전 호환성으로 인해 사용자 지정 플러그인 설치 실패”하세요.

콘솔을 사용하여 사용자 지정 플러그인 설치

타사 플러그인을 도메인에 연결하려면 먼저 플러그인 라이선스 및 구성을 패키지로 가져옵니다.

사용자 지정 플러그인을 설치하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 패키지를 선택합니다.
3. [패키지 가져오기(Import package)]를 선택합니다.

4. 이름에 플러그인의 고유하고 쉽게 식별할 수 있는 이름을 입력합니다.
5. (선택 사항) 설명에서 패키지 또는 용도에 대한 유용한 세부 정보를 제공합니다.
6. 패키지 유형에서 플러그인을 선택합니다.
7. 패키지 소스에 경로를 입력하거나 Amazon S3에서 플러그인 ZIP 파일을 찾습니다.
8. OpenSearch 엔진 버전에서 플러그인이 지원하는 OpenSearch 버전을 선택합니다.
9. 패키지 암호화에서 패키지의 암호화 키를 사용자 지정할지 여부를 선택합니다. 기본적으로 OpenSearch Service는를 사용하여 플러그인 패키지를 암호화합니다 AWS 소유 키. 대신 고객 관리형 키를 사용할 수 있습니다.
10. 가져오기를 선택합니다.

플러그인 패키지를 가져온 후 도메인과 연결합니다. 지침은 the section called “패키지를 가져와 도메인에 연결” 섹션을 참조하세요.

를 사용하여 사용자 지정 플러그인 관리 AWS CLI

AWS CLI 를 사용하여 여러 사용자 지정 플러그인 작업을 관리할 수 있습니다.

업무

- 를 사용하여 사용자 지정 플러그인 설치 AWS CLI
- 를 사용하여 사용자 지정 플러그인 업데이트 AWS CLI
- AWS KMS 키 보안으로 사용자 지정 플러그인 생성 또는 업데이트
- 를 사용하여 사용자 지정 플러그인이 있는 OpenSearch Service 도메인을 최신 버전의 OpenSearch로 업그레이드 AWS CLI
- 사용자 지정 플러그인의 연결 해제 상태 제거 및 보기

를 사용하여 사용자 지정 플러그인 설치 AWS CLI

시작하기 전 준비 사항

사용자 지정 플러그인을 도메인과 연결하려면 먼저 Amazon Simple Storage Service(Amazon S3) 버킷에 업로드해야 합니다. 버킷은 플러그인을 사용하려는 AWS 리전 동일한에 있어야 합니다. S3 버킷에 객체를 추가하는 방법에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 객체 업로드를 참조하세요.

플러그인에 민감한 정보가 포함된 경우 업로드할 때 S3-managed 키를 사용한 서버 측 암호화를 지정합니다. 파일을 업로드한 후 S3 경로를 기록해 둡니다. 경로 형식은 `s3://amzn-s3-demo-bucket/file-path/file-name`입니다.

> **Note**
>
> 선택적으로 AWS Key Management Service (AWS KMS) 키를 지정하여 플러그인을 생성할 때 사용자 지정 플러그인을 보호할 수 있습니다. 자세한 내용은 the section called “AWS KMS 키 보안으로 사용자 지정 플러그인 생성 또는 업데이트” 단원을 참조하세요.

를 사용하여 사용자 지정 플러그인을 설치하려면 AWS CLI

1. 다음 create-package 명령을 실행하여 사용자 지정 플러그인에 대한 새 패키지를 생성하고 다음 요구 사항을 충족하는지 확인합니다.

   - 버킷과 키 위치는가 명령을 실행 중인 계정의 S3 버킷에 있는 플러그인 `.zip` 파일을 가리켜야 합니다.
   - S3 버킷은 패키지가 생성되는 리전과 동일한 리전에 있어야 합니다.
   - `ZIP-PLUGIN` 패키지에는 `.zip` 파일만 지원됩니다.
   - `.zip` 파일의 내용은 플러그인에서 예상한 디렉터리 구조를 따라야 합니다.
   - 의 값은 형식이어야 `--engine-version` 합니다`OpenSearch_{MAJOR}.{MINOR}`. 예를 들어 **`OpenSearch_2.17`**입니다.

   ## ### ## 자신의 정보로 바꿉니다.

   ```
   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. (선택 사항) describe-packages 명령을 사용하여 검증 및 보안 취약성 조사 결과를 포함한 `create-package` 작업 상태를 확인합니다. *## ### ##* 자신의 정보로 바꿉니다.

   ```
   aws opensearch describe-packages \
       --region region  \
   ```

```
    --filters '[{"Name": "PackageType","Value": ["ZIP-PLUGIN"]}, {"Name":
 "PackageName","Value": ["package-name"]}]'
```

이 명령은 다음과 유사한 정보를 반환합니다.

```
{
    "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**
>
> `create-package` 작업 중에 Amazon OpenSearch Service는 `ZIP-PLUGIN` 값에서 버전 호환성, 지원되는 플러그인 확장 및 보안 취약성을 확인합니다. 보안 취약성은 Amazon Inspector 서비스를 사용하여 스캔됩니다. 이러한 검사의 결과는 API 응답의 `ErrorDetails` 필드에 표시됩니다.

3. associate-package 명령을 사용하여 이전 단계에서 생성한 패키지의 패키지 ID를 사용하여 플러그인을 선택한 OpenSearch Service 도메인과 연결합니다.

> **Tip**
>
> 플러그인이 여러 개 있는 경우 associate-packages 명령을 사용하여 단일 작업으로 여러 패키지를 도메인에 연결할 수 있습니다.

자신의 정보로 바꿉니다.

```
aws opensearch associate-package \
    --domain-name domain-name \
    --region region \
    --package-id package-id
```

> **Note**
>
> 플러그인은 블루/그린 배포 프로세스를 사용하여 설치 및 제거됩니다.

4. (선택 사항) list-packages-for-domain 명령을 사용하여 연결 상태를 확인합니다. 워크플로가에서 ASSOCIATING로 진행됨에 따라 연결 상태가 변경됩니다ACTIVE. 플러그인 설치가 완료되고 플러그인을 사용할 준비가 되면 연결 상태가 ACTIVE로 변경됩니다.

   *## ###*를 자신의 정보로 바꿉니다.

```
aws opensearch list-packages-for-domain \
    --region region \
    --domain-name domain-name
```

를 사용하여 사용자 지정 플러그인 업데이트 AWS CLI

update-package 명령을 사용하여 플러그인을 변경합니다.

> **Note**
>
> AWS Key Management Service (AWS KMS) 키를 지정하여 플러그인을 업데이트할 때 선택적으로 사용자 지정 플러그인을 보호할 수 있습니다. 자세한 내용은 the section called “AWS KMS 키 보안으로 사용자 지정 플러그인 생성 또는 업데이트” 단원을 참조하세요.

를 사용하여 사용자 지정 플러그인을 업데이트하려면 AWS CLI

- 다음 명령을 실행합니다. *## ###*를 자신의 정보로 바꿉니다.

```
aws opensearch update-package \
```

```
    --region region \
    --package-id package-id \
    --package-source S3BucketName=amzn-s3-demo-bucket,S3Key=s3-key \
    --package-description description
```

패키지를 업데이트한 후 associate-package 또는 associate-packages 명령을 사용하여 도메인에 패키지 업데이트를 적용할 수 있습니다.

> **Note**
>
> 를 사용하여 플러그인에 대한 작업을 감사, 생성, 업데이트, 연결 및 연결 해제할 수 있습니다 AWS CloudTrail. 자세한 내용은 AWS CloudTrail을 사용한 Amazon OpenSearch Service API 호출 모니터링 단원을 참조하십시오.

AWS KMS 키 보안으로 사용자 지정 플러그인 생성 또는 업데이트

AWS KMS 키를 지정하여 플러그인을 생성하거나 업데이트할 때 사용자 지정 플러그인을 보호할 수 있습니다. 이렇게 하려면 다음 예제`true`와 같이를 `PackageEncryptionOptions` 로 설정하고 키의 Amazon 리소스 이름(ARN)을 지정합니다.

예: AWS KMS 키 보안으로 사용자 지정 플러그인 생성

```
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"
  }
```

예: AWS KMS 키 보안으로 사용자 지정 플러그인 업데이트

```
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

지정한 AWS KMS 키가 비활성화되거나 삭제되면 연결된 클러스터가 작동하지 않을 수 있습니다.

사용자 지정 패키지와의 AWS KMS 통합에 대한 자세한 내용은 단원을 참조하십시오the section called "Amazon OpenSearch Service 사용자 지정 패키지 AWS KMS 통합".

를 사용하여 사용자 지정 플러그인이 있는 OpenSearch Service 도메인을 최신 버전의 OpenSearch로 업그레이드 AWS CLI

사용자 지정 플러그인을 사용하는 OpenSearch Service 도메인을 최신 버전의 OpenSearch로 업그레이드해야 하는 경우 다음 프로세스를 완료하세요.

를 사용하여 사용자 지정 플러그인이 있는 OpenSearch Service 도메인을 최신 버전의 OpenSearch로 업그레이드하려면 AWS CLI

1. create-package 명령을 사용하여 새 OpenSearch 버전을 지정하는 플러그인에 대한 새 패키지를 생성합니다.

   모든 엔진 버전의 플러그인에 대해 패키지 이름이 동일한지 확인합니다. 패키지 이름을 변경하면 블루/그린 배포 중에 도메인 업그레이드 프로세스가 실패합니다.
2. 의 단계에 따라 도메인을 상위 버전으로 업그레이드합니다the section called "도메인 업그레이드".

   이 프로세스 중에 Amazon OpenSearch Service는 이전 버전의 플러그인 패키지 연결을 해제하고 블루/그린 배포를 사용하여 새 버전을 설치합니다.

사용자 지정 플러그인의 연결 해제 상태 제거 및 보기

도메인에서 플러그인을 제거하려면 dissociate-package 명령을 사용할 수 있습니다. 이 명령을 실행하면 관련 구성 또는 라이선스 패키지도 제거됩니다. 그런 다음 list-packages-for-domain 명령을 사용하여 연결 해제 상태를 볼 수 있습니다.

> **Tip**
>
> dissociate-packages 명령을 사용하여 단일 작업으로 도메인에서 여러 플러그인을 제거할 수도 있습니다.

사용자 지정 플러그인의 연결 해제 상태를 제거하고 보려면

1. 모든 인덱스에서 플러그인을 비활성화합니다. 플러그인 패키지를 연결 해제하기 전에이 작업을 수행해야 합니다.

   모든 인덱스에서 비활성화하기 전에 플러그인을 제거하려고 하면 블루/그린 배포 프로세스가 `Processing` 상태로 유지됩니다.
2. 다음 명령을 실행하여 플러그인을 제거합니다. *## ###*를 자신의 정보로 바꿉니다.

   ```
   aws opensearch dissociate-package \
       --region region \
       --package-id plugin-package-id \
       --domain-name domain name
   ```
3. (선택 사항) list-packages-for-domain 명령을 실행하여 연결 해제 상태를 확인합니다.

Amazon OpenSearch Service 사용자 지정 패키지 AWS KMS 통합

Amazon OpenSearch Service 사용자 지정 패키지는 기본적으로 암호화를 제공하여를 사용하여 저장된 `ZIP-PLUGIN` 패키지를 보호합니다 AWS 관리형 키.

- AWS 소유 키 - Amazon OpenSearch Service 사용자 지정 패키지는 기본적으로 이러한 키를 사용하여 `ZIP-PLUGIN` 패키지를 자동으로 암호화합니다. 사용을 확인, 관리 또는 사용하거나 AWS 소유 키 감사할 수 없습니다. 하지만 데이터를 암호화하는 키를 보호하기 위해 어떤 작업을 수행하거나 어떤 프로그램을 변경할 필요가 없습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS 소유 키 섹션을 참조하세요.

- 고객 관리형 키 - 사용자 `ZIP-PLUGIN` 지정 패키지를 생성할 때 고객 관리형 키를 AWS 소유 키 선택하여 기존에 두 번째 암호화 계층을 추가할 수 있습니다.

  Amazon OpenSearch Service 사용자 지정 패키지는 사용자가 생성, 소유 및 관리하는 대칭 고객 관리형 키를 사용하여 기존 AWS 소유 암호화에 두 번째 암호화 계층을 추가할 수 있도록 지원합니다. 이 암호화 계층을 완전히 제어할 수 있으므로 다음 작업을 수행할 수 있습니다.

  - 키 정책 수립 및 관리
  - AWS Identity and Access Management (IAM) 정책 및 권한 부여 설정 및 유지
  - 키 정책 활성화 및 비활성화
  - 키 암호화 자료 교체
  - 태그 추가
  - 키 별칭 생성
  - 키 삭제 예약

자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키를 참조하세요.

> **Note**
>
> Amazon OpenSearch Service 사용자 지정 패키지는 무료로 AWS 소유 키 를 사용하여 저장 시 암호화를 자동으로 활성화합니다. 그러나 고객 관리형 키를 사용하는 경우 AWS KMS 요금이 적용됩니다. 요금에 대한 자세한 내용은 AWS Key Management Service 요금을 참조하세요.

Amazon OpenSearch Service 사용자 지정 패키지 서비스가에서 권한 부여를 사용하는 방법 AWS KMS

OpenSearch Service 사용자 지정 패키지는 고객 관리형 키를 사용하려면 권한 부여가 필요합니다.

고객 관리형 키로 암호화된 `ZIP-PLUGIN` 패키지를 생성하면 Amazon OpenSearch Service 사용자 지정 패키지 서비스는 CreateGrant 요청을에 전송하여 사용자를 대신하여 권한을 생성합니다 AWS KMS. 의 권한 부여는 OpenSearch Service에 계정의 AWS KMS 키에 대한 액세스 권한을 AWS KMS 부여합니다. OpenSearch Service 사용자 지정 패키지에서 생성한 권한 부여에는 요청에 사용자 지정 패키지 ID가 있는 암호화 컨텍스트가 포함된 경우에만 작업을 허용하는 제약이 있습니다.

Amazon OpenSearch Service 사용자 지정 패키지는 다음 내부 작업에 고객 관리형 키를 사용하려면 권한 부여가 필요합니다.

Operation	설명
`DescribeKey`	플러그인 패키지를 생성할 때 입력한 대칭 고객 관리형 키 ID가 유효한지 AWS KMS 확인하는 `DescribeKey` 요청을에 보냅니다.
`GenerateDataKeyWithoutPlaintext`	고객 관리형 키로 암호화된 데이터 키를 생성하기 AWS KMS 위해에 `GenerateDataKeyWithoutPlaintext` 요청을 보냅니다.
`GenerateDataKey`	내부적으로 패키지를 복사할 때 패키지를 암호화하는 데이터 키를 생성 AWS KMS 하라는 `GenerateDataKey` 요청을에 보냅니다.
`Decrypt`	암호화된 데이터 키를 복호화 AWS KMS 하는 데 사용할 수 있도록에 `Decrypt` 요청을 보냅니다.

언제든지 권한 부여에 대한 액세스를 취소하거나 고객 관리형 키에 대한 서비스의 액세스를 제거할 수 있습니다. 이렇게 하면 OpenSearch Service는 고객 관리형 키로 암호화된 데이터에 액세스할 수 없으며, 이는 해당 데이터에 의존하는 작업에 영향을 미칩니다. 예를 들어 OpenSearch Service가 액세스할 수 없는 플러그인 패키지를 연결하려고 하면 작업이 `AccessDeniedException` 오류를 반환합니다.

고객 관리형 키 생성

AWS Management Console 또는 AWS KMS APIs.

대칭 고객 관리형 키를 만들려면

- AWS Key Management Service 개발자 안내서의 KMS 키 생성 단계를 따릅니다.

키 정책

키 정책에서는 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 만들 때 키 정책을 지정할 수 있습니다. 자세한 내용은AWS Key Management Service 개발자 안내서의 AWS KMS의 키 정책을 참조하세요.

고객 관리형 키를 플러그인 리소스와 함께 사용하려면 키 정책에서 다음 API 작업을 허용해야 합니다.

- `kms:CreateGrant` - 고객 관리형 키에 권한 부여를 추가합니다. 지정된 AWS KMS 키에 대한 제어 액세스 권한을 부여하여 OpenSearch Service 사용자 지정 패키지에 필요한 작업을 부여할 수 있는

액세스를 허용합니다. 권한 부여 사용에 대한 자세한 내용은 AWS KMS 개발자 안내서를 참조하세요.

이를 통해 OpenSearch Service는 다음을 수행할 수 있습니다.

- 를 호출GenerateDataKeyWithoutPlainText하여 암호화된 데이터 키를 생성하고 추가 검증을 위해 저장합니다.
- 를 호출GenerateDataKey하여 플러그인 패키지를 내부적으로 복사합니다.
- 를 호출Decrypt하여 플러그인 패키지에 내부적으로 액세스합니다.
- 서비스가 RetireGrant를 사용할 수 있도록 은퇴하는 보안 주체를 설정하세요.

- kms:DescribeKey - OpenSearch Service가 키를 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.
- kms:GenerateDataKey, kms:GenerateDataKeyWithoutPlaintext, kms:Decrypt- OpenSearch Service 사용자 지정 패키지에 권한 부여에서 이러한 작업을 사용할 수 있는 액세스 권한을 부여합니다.

다음은 OpenSearch Service 사용자 지정 패키지에 추가할 수 있는 정책 설명 예제입니다.

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

정책에서 권한을 지정하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 의 키 정책을 AWS KMS 참조하세요.

키 액세스 문제 해결에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS 권한 문제 해결을 참조하세요.

Amazon OpenSearch Service 사용자 지정 패키지에 대한 고객 관리형 키 지정

고객 관리형 키를 ZIP-PLUGIN 패키지의 두 번째 암호화 계층으로 지정할 수 있습니다.

플러그인 패키지를 생성할 때 OpenSearch Service 사용자 지정 패키지가 플러그인 패키지를 암호화하는 데 사용하는 AWS KMS 키 ID를 입력하여 데이터 키를 지정할 수 있습니다.

AWS KMS 키 ID - AWS KMS 고객 관리형 키의 키 식별자입니다. 키 ID, 키 ARN, 별칭 이름 또는 별칭 ARN을 입력합니다.

Amazon OpenSearch Service 사용자 지정 패키지 암호화 컨텍스트

암호화 컨텍스트는 데이터에 대한 추가 컨텍스트 정보를 포함하는 선택적 키-값 페어 세트입니다.

AWS KMS 는 암호화 컨텍스트를 추가 인증 데이터로 사용하여 인증된 암호화를 지원합니다. 데이터 암호화 요청에 암호화 컨텍스트를 포함하면는 암호화 컨텍스트를 암호화된 데이터에 AWS KMS 바인딩합니다. 요청에 동일한 암호화 컨텍스트를 포함해야 이 데이터를 해독할 수 있습니다.

Amazon OpenSearch Service 사용자 지정 패키지 암호화 컨텍스트

Amazon OpenSearch Service 사용자 지정 패키지는 모든 암호화 작업에서 동일한 암호화 컨텍스트를 사용합니다. 여기서 키는 AWS KMS `packageId`이고 값은 플러그인 패키지`package-id`의 입니다.

모니터링을 위한 암호화 컨텍스트 사용

대칭 고객 관리형 키를 사용하여 플러그인 패키지를 암호화하는 경우 감사 레코드 및 로그의 암호화 컨텍스트를 사용하여 고객 관리형 키가 사용되는 방식을 식별할 수 있습니다. 암호화 컨텍스트는 AWS CloudTrail 또는 Amazon CloudWatch Logs에서 생성된 로그에도 나타납니다.

암호화 컨텍스트를 사용하여 고객 관리형 키에 대한 액세스 제어

그러나 키 정책 및 IAM 정책에서 암호화 컨텍스트를 조건으로 사용하여 대칭형 고객 관리형 키에 대한 액세스를 제어할 수도 있습니다. 또한 권한 부여에서 암호화 컨텍스트 제약 조건을 사용할 수 있습니다.

OpenSearch Service 사용자 지정 패키지는 권한 부여에 암호화 컨텍스트 제약 조건을 사용하여 계정 또는 리전의 고객 관리형 키에 대한 액세스를 제어합니다. 권한 부여 제약 조건에 따라 권한 부여가 허용하는 작업은 지정된 암호화 컨텍스트를 사용해야 합니다.

다음은 특정 암호화 컨텍스트에서 고객 관리형 키에 대한 액세스 권한을 부여하는 키 정책 설명의 예입니다. 이 정책 설명의 조건에 따라 권한 부여에는 암호화 컨텍스트를 지정하는 암호화 컨텍스트 제약 조건이 있어야 합니다.

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

OpenSearch 사용자 지정 패키지 서비스에 대한 암호화 키 모니터링

OpenSearch Service 사용자 지정 패키지 서비스 리소스와 함께 AWS KMS 고객 관리형 키를 사용하는 경우 CloudTrail 또는 CloudWatch Logs를 사용하여 OpenSearch 사용자 지정 패키지가에 보내는 요청을 추적할 수 있습니다 AWS KMS.

자세히 알아보기

다음 리소스에서 키에 대한 추가 정보를 확인할 수 있습니다.

- AWS KMS 기본 개념에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서AWS KMS keys의 섹션을 참조하세요.
- 의 보안 모범 사례에 대한 자세한 내용은 AWS Key Management Service 모범 사례에 대한 AWS 권장 가이드 가이드를 AWS KMS참조하세요.

Amazon OpenSearch Service에 타사 플러그인 설치

Amazon OpenSearch Service는 선택한 파트너의 타사 플러그인을 지원합니다. 이러한 플러그인은 사용자 지정 분석기, 토큰화기 또는 암호화 기능과 같은 추가 기능으로 OpenSearch 설정을 개선할 수 있습니다. 타사 개발자가 제공하는 특정 설치 및 구성 지침에 따라 OpenSearch Service 도메인과 적절하게 통합되도록 합니다.

> **Note**
>
> 타사 개발자로부터 직접 유효한 라이선스를 획득하고 유지해야 합니다. 일부 공급자는 플러그인을 전혀 활성화하지 않을 수 AWS 리전있으므로 플러그인 공급자에게 가용성을 확인하세요.

OpenSearch Service에서 사용할 수 있는 타사 플러그인은 다음과 같습니다.

- Portal26 암호화 플러그인(Titanium-lockbox) - NIST FIPS 140-2 인증 암호화를 사용하여 인덱싱될 때 데이터를 암호화합니다. 여기에는 향상된 보안을 위해 암호화 키를 관리할 수 있는 BYOK(Bring Your Own Key) 지원이 포함되어 있습니다. 플러그인은 Portal26에서 제공하며 OpenSearch 버전 2.15 이상이 필요합니다.
- 이름 일치(RNI) - 24개 이상의 언어로 이름, 조직, 주소 및 날짜를 일치시켜 보안 및 규정 준수를 개선합니다. 플러그인은 Babel Street에서 제공하며 OpenSearch 버전 2.15 이상이 필요합니다.

주제

- 사전 조건
- 타사 플러그인 설치
- 다음 단계

사전 조건

타사 플러그인을 설치하기 전에 다음 단계를 수행합니다.

- 플러그인 구성 및 라이선스 파일을 가져와 Amazon S3 버킷에 업로드했습니다. 버킷은 도메인 AWS 리전 과 동일한에 있어야 합니다.
- 타사 플러그인은 사용자 지정 플러그인의 한 유형입니다. 도메인이 사용자 지정 플러그인의 사전 조건을 충족하는지 확인합니다.

타사 플러그인 설치

타사 플러그인을 OpenSearch Service 도메인과 연결하려면 먼저 라이선스 패키지, 구성 패키지 및 플러그인 패키지라는 세 개의 개별 패키지를 업로드해야 합니다.

- 라이선스 패키지에는 플러그인과 연결된 라이선스 정보 또는 메타데이터가 .json 또는 .xml 형식으로 포함되어 있습니다.

- 구성 패키지에는 플러그인 구성 파일과 지원되는 자산 및 설정이 포함되어 있습니다. 이러한 파일은 플러그인이 OpenSearch와 동작하거나 통합되는 방식을 정의합니다.
- 플러그인 패키지에는 OpenSearch가 실행하는 실행 코드인 컴파일된 플러그인 바이너리가 포함되어 있습니다. 플러그인 기능의 핵심입니다.

두 패키지를 모두 업로드한 후 플러그인과 라이선스를 호환되는 도메인과 연결할 수 있습니다.

콘솔

타사 플러그인을 도메인에 연결하려면 먼저 플러그인 라이선스 및 구성을 패키지로 가져옵니다.

타사 플러그인을 설치하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 패키지를 선택합니다.
3. 먼저 라이선스 패키지를 가져옵니다. [패키지 가져오기(Import package)]를 선택합니다.
4. 패키지 유형에서 라이선스를 선택합니다.
5. 패키지 소스에 Amazon S3의 라이선스 JSON 또는 XML 파일의 경로를 입력합니다.
6. 가져오기를 선택합니다. 패키지는 패키지 페이지의 라이선스 탭에 나타납니다.
7. 이제 플러그인 구성을 가져옵니다. 패키지 가져오기를 다시 선택합니다.
8. 패키지 유형에서 구성을 선택합니다.
9. 패키지 소스에 Amazon S3의 플러그인 구성 ZIP 파일의 경로를 입력합니다.
10. 가져오기를 선택합니다.
11. 마지막으로 플러그인 자체를 가져옵니다. [패키지 가져오기(Import package)]를 선택합니다.
12. 패키지 유형에서 플러그인을 선택합니다.
13. 패키지 소스에 Amazon S3의 플러그인 ZIP 파일 경로를 입력합니다.
14. 플러그인이 지원하는 OpenSearch 엔진 버전을 선택합니다.
15. 가져오기를 선택합니다.

타사 플러그인을 도메인에 연결하려면

1. 이제 플러그인 라이선스 및 구성을 도메인과 연결합니다. 왼쪽 탐색 창에서 도메인을 선택합니다.
2. 도메인 이름을 선택하여 클러스터 구성을 엽니다.
3. 플러그인 탭으로 이동합니다.

4. 패키지 연결을 선택하고 방금 가져온 플러그인, 라이선스 및 구성 패키지를 선택합니다.
5. 선택을 선택하세요.
6. 다음을 선택합니다. 연결할 패키지를 검토하고 연결을 선택합니다.

CLI

먼저 create-package 명령을 사용하여 플러그인 라이선스가 포함된 새 패키지를 생성합니다. 는 라이선스 텍스트 또는 메타데이터가 포함된 Amazon S3의 .json 또는 .xml 파일을 가리켜S3Key야 합니다.

```
aws opensearch create-package \
  --package-name plugin-license-package \
  --package-type PACKAGE-LICENSE \
  --package-source S3BucketName=my-bucket,S3Key=licenses/my-plugin-license.json
```

create-package 명령을 다시 사용하여 플러그인 구성이 포함된 패키지를 생성합니다. 는 플러그인에 필요한 디렉터리 구조를 준수하는 Amazon S3의 .zip 파일을 가리켜S3Key야 합니다.

```
aws opensearch create-package \
  --package-name plugin-config-package \
  --package-type PACKAGE-CONFIG \
  --package-source S3BucketName=my-bucket,S3Key=path/to/package.zip
```

create-package 명령을 다시 사용하여 플러그인 자체가 포함된 패키지를 생성합니다. 는 Amazon S3의 플러그인 .zip 파일을 가리켜S3Key야 합니다.

```
aws opensearch create-package \
  --package-name plugin-package \
  --package-type ZIP-PLUGIN \
  --package-source S3BucketName=my-bucket,S3Key=path/to/package.zip
```

마지막으로 associate-package 명령을 사용하여 각각에 대한 패키지 IDs를 지정하여 파트너 플러그인, 라이선스 및 구성을 호환 도메인에 연결합니다. 플러그인 ID를 다른 패키지의 사전 조건으로 지정합니다. 즉, 플러그인 ID를 다른 패키지보다 먼저 도메인과 연결해야 합니다.

```
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"]}]'
```

다음 단계

연결이 완료되면 특정 인덱스에서 플러그인을 활성화하거나 요구 사항에 따라 필요에 따라 구성할 수 있습니다. 타사 플러그인 기능을 특정 인덱스에 적용하려면 인덱스 생성 중에 인덱스 설정을 수정하거나 기존 인덱스를 업데이트합니다. 예를 들어 타사 플러그인에 사용자 지정 분석기가 포함된 경우 인덱스 설정에서 참조하세요.

플러그인 기능을 여러 인덱스에 일관되게 적용하려면 플러그인 구성이 포함된 인덱스 템플릿을 사용합니다. OpenSearch 설정에 맞게 기능을 구성하는 방법을 이해하려면 항상 플러그인 설명서를 참조하세요.

SQL을 사용하여 Amazon OpenSearch Service 데이터 쿼리

JSON 기반 OpenSearch 쿼리 DSL을 사용하는 대신 SQL을 사용하여 Amazon OpenSearch Service를 쿼리할 수 있습니다. SQL을 사용한 쿼리는 SQL에 이미 익숙하거나 도메인을 SQL을 사용하는 애플리케이션과 통합하려는 경우에 유용합니다. SQL 지원은 OpenSearch 또는 Elasticsearch 6.5 이상을 실행하는 도메인에서 사용할 수 있습니다.

> **Note**
>
> 이 설명서에서는 OpenSearch Service와 다양한 SQL 플러그인 버전 간 버전 호환성과 JDBC 및 ODBC 드라이버에 대해 설명합니다. 기본 및 복잡한 쿼리, 함수, 메타데이터 쿼리 및 집계 함수의 구문에 대한 자세한 내용은 오픈 소스 OpenSearch 설명서를 참조하세요.

다음 표를 사용하여 각 OpenSearch 및 Elasticsearch 버전에서 지원되는 SQL 플러그인 버전을 찾습니다.

OpenSearch

OpenSearch 버전	SQL 플러그인 버전	주목할 만한 기능
2.19.0	2.19.0.0	
2.18.0	2.18.0.0	
2.17.0	2.17.0.0	

OpenSearch 버전	SQL 플러그인 버전	주목할 만한 기능
2.15.0	2.15.0.0	
2.13.0	2.13.0.0	
2.11.0	2.11.0.0	PPL 언어 및 쿼리에 대한 지원 추가
2.9.0	2.9.0.0	Spark 커넥터 추가, 표 및 PromQL 함수 지원
2.7.0	2.7.0.0	`datasource` API 추가
2.5.0	2.5.0.0	
2.3.0	2.3.0.0	`maketime` 및 `makedate` 날짜/시간 함수 추가
1.3.0	1.3.0.0	기본 쿼리 제한 크기 및 값 목록 내에서 선택할 수 있는 IN 절 지원
1.2.0	1.2.0.0	시각화 응답 형식에 대한 새 프로토콜 추가
1.1.0	1.1.0.0	SQL 및 PPL에서 필터로 일치 함수 지원
1.0.0	1.0.0.0	데이터 스트림 쿼리 지원

Open Distro for Elasticsearch

Elasticsearch 버전	SQL 플러그인 버전	주목할 만한 기능
7.10	1.13.0	윈도 함수용 NULL FIRST 및 LAST, CAST() 함수, SHOW 및 DESCRIBE 명령
7.9	1.11.0	추가 날짜/시간 함수 추가, ORDER BY 키워드
7.8	1.9.0	
7.7	1.8.0	

Elasticsearch 버전	SQL 플러그인 버전	주목할 만한 기능
7.3	1.3.0	여러 문자열 및 숫자 연산자
7.1	1.1.0	

샘플 호출

SQL을 사용하여 데이터를 쿼리하려면 다음 형식을 사용하여 `_sql`에 HTTP 요청을 전송합니다.

```
POST domain-endpoint/_plugins/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

> **Note**
>
> 도메인이 OpenSearch 대신 Elasticsearch를 실행하는 경우 형식은 `_opendistro/_sql`입니다.

참고 사항 및 차이점

`_plugins/_sql`에 대한 호출은 인덱스 이름을 요청 본문에 포함하므로 대량, mget 및 msearch 작업과 동일한 액세스 정책 고려 사항을 갖습니다. 항상 그렇듯이, API 작업에 권한을 부여할 때는 최소 권한의 원칙을 따릅니다.

세분화된 액세스 제어와 함께 SQL을 사용하는 것과 관련된 보안 고려 사항은 the section called “세분화된 액세스 제어”를 참조하세요.

OpenSearch SQL 플러그 인은 많은 조정 가능한 설정을 포함합니다. OpenSearch Service에서는 플러그 인 설정 경로(`_plugins/_query/settings`)가 아닌 `_cluster/settings` 경로를 사용합니다.

```
PUT _cluster/settings
{
  "transient" : {
    "plugins.sql.enabled" : true
```

```
  }
}
```

레거시 Elasticsearch 도메인의 경우 `plugins`을(를) `opendistro`(으)로 대체합니다.

```
PUT _cluster/settings
{
  "transient" : {
    "opendistro.sql.enabled" : true
  }
}
```

SQL Workbench

SQL Workbench는 온디맨드 SQL 쿼리를 실행하고, SQL을 해당 REST로 변환하며, 결과를 텍스트, JSON, JDBC 또는 CSV로 보고 저장할 수 있는 OpenSearch 대시보드 사용자 인터페이스입니다. 자세한 내용은 쿼리 워크벤치를 참조하세요.

SQL CLI

SQL CLI는 `opensearchsql` 명령을 사용하여 시작할 수 있는 독립형 Python 애플리케이션입니다. 설치, 구성 및 사용 단계는 SQL CLI를 참조하세요.

JDBC 드라이버

JDBC(Java Database Connectivity) 드라이버를 사용하여 OpenSearch Service 도메인을 선호하는 비즈니스 인텔리전스(BI) 애플리케이션과 통합할 수 있습니다. 드라이버를 다운로드하려면 여기를 클릭하세요. 자세한 내용은 GitHub 리포지토리를 참조하세요.

ODBC 드라이버

오픈 데이터베이스 연결(ODBC) 드라이버는 Microsoft Excel과 같은 비즈니스 인텔리전스 및 데이터 시각화 애플리케이션을 SQL 플러그인에 연결할 수 있는 Windows 및 macOS용 읽기 전용 ODBC 드라이버입니다.

OpenSearch 아티팩트 페이지에서 예제 작동 드라이버 파일을 다운로드할 수 있습니다. 드라이버 설치에 대한 정보는 GitHub의 SQL 리포지토리를 확인하세요.

Amazon OpenSearch Service의 교차 클러스터 검색

Amazon OpenSearch Service의 클러스터 간 검색을 사용하면 연결된 여러 도메인에서 쿼리 및 집계를 수행할 수 있습니다. 특히 여러 유형의 워크로드를 실행하는 경우 큰 단일 도메인 대신 여러 개의 작은 도메인을 사용하는 것이 더 좋습니다.

워크로드별 도메인을 사용하면 다음 작업을 수행할 수 있습니다.

- 특정 워크로드에 대한 인스턴스 유형을 선택하여 각 도메인을 최적화합니다.
- 워크로드 전반에 걸쳐 결함 격리 경계를 설정합니다. 즉, 워크로드 중 하나가 실패하면 해당 특정 도메인 내에 결함이 포함되며 다른 워크로드에 영향을 주지 않습니다.
- 여러 도메인에서 더욱 쉽게 조정

클러스터 간 검색은 OpenSearch Dashboards를 지원하므로 모든 도메인에서 시각화 및 대시보드를 생성할 수 있습니다. 도메인 간에 전송된 검색 결과에 대해 표준 AWS 데이터 전송 요금을 지불합니다.

> **Note**
>
> 오픈 소스 OpenSearch에는 클러스터 간 검색을 위한 설명서도 제공합니다. 관리형 Amazon OpenSearch Service 도메인과 비교했을 때 오픈 소스 클러스터에 대한 설정은 크게 다릅니다. 특히 OpenSearch Service에서는 cURL을 통하지 AWS Management Console 않고를 사용하여 클러스터 간 연결을 구성합니다. 또한 관리형 서비스는 세분화된 액세스 제어 외에도 클러스터 간 인증에 AWS Identity and Access Management (IAM)을 사용합니다. 따라서 오픈 소스 OpenSearch 설명서 대신 이 설명서를 사용하여 도메인에 대한 교차 클러스터 검색을 구성하는 것이 좋습니다.

주제

- 제한 사항
- 클러스터 간 검색 전제 조건
- 클러스터 간 검색 요금
- 연결 설정
- 연결 제거
- 보안 설정 및 샘플 시연
- OpenSearch Dashboards

제한 사항

클러스터 간 검색에는 몇 가지 중요한 제한 사항이 있습니다.

- Elasticsearch 도메인을 OpenSearch 도메인과 연결할 수 없습니다.
- 자체 관리형 OpenSearch/Elasticsearch 클러스터에는 연결할 수 없습니다.
- 리전 간에 도메인을 연결하려면 두 도메인 모두 Elasticsearch 7.10 이상이거나 OpenSearch를 사용해야 합니다.
- 도메인에는 최대 20개의 발신 연결이 있을 수 있습니다. 마찬가지로 도메인에는 최대 20개의 수신 연결이 있을 수 있습니다. 즉, 한 도메인은 최대 20개의 다른 도메인에 연결할 수 있습니다.
- 원본 도메인은 대상 도메인과 같거나 상위 버전에 있어야 합니다. 두 도메인 간에 양방향 연결을 설정하고 둘 중 하나 또는 둘 다 업그레이드하려는 경우 먼저 연결 중 하나를 삭제해야 합니다.
- 클러스터 간 검색에는 사용자 지정 사전이나 SQL을 사용할 수 없습니다.
- AWS CloudFormation 를 사용하여 도메인을 연결할 수 없습니다.
- M3 또는 버스트 가능(T2 및 T3) 인스턴스에서는 클러스터 간 검색을 사용할 수 없습니다.

클러스터 간 검색 전제 조건

클러스터 간 검색을 설정하기 전에 도메인이 다음 요구 사항을 충족하는지 확인하십시오.

- 버전 6.7 이상의 OpenSearch 도메인 또는 Elasticsearch 도메인 2개
- 세분화된 액세스 제어를 사용하도록 설정됨
- 노드 간 암호화를 사용하도록 설정됨

클러스터 간 검색 요금

도메인 간 검색에는 추가 요금이 부과되지 않습니다.

연결 설정

“소스” 도메인은 클러스터 간 검색 요청이 시작된 도메인을 나타냅니다. 즉, 소스 도메인은 초기 검색 요청을 보내는 도메인입니다.

“대상” 도메인은 소스 도메인이 쿼리하는 도메인입니다.

클러스터 간 연결은 소스 도메인에서 대상 도메인으로 단방향입니다. 즉, 대상 도메인이 소스 도메인을 쿼리할 수 없습니다. 그러나 반대 방향으로 다른 연결을 설정할 수 있습니다.

소스 도메인은 대상 도메인에 대한 "아웃바운드" 연결을 생성합니다. 대상 도메인은 소스 도메인에서 “인바운드” 연결 요청을 받습니다.

연결을 설정하려면

1. 도메인 대시보드에서 도메인을 선택하고 연결(Connections) 탭을 선택합니다.
2. [아웃바운드 연결(Outbound connections)] 섹션에서 [요청(Request)]을 선택합니다.
3. [연결 별칭(Connection alias)]에 연결 이름을 입력합니다.
4. AWS 계정 및 리전 또는 다른 계정 또는 리전의 도메인에 연결할지 선택합니다.
   - AWS 계정 및 리전의 클러스터에 연결하려면 드롭다운 메뉴에서 도메인을 선택하고 요청을 선택합니다.
   - 다른 AWS 계정 또는 리전의 클러스터에 연결하려면 원격 도메인의 ARN을 선택하고 요청을 선택합니다. 리전 간에 도메인을 연결하려면 두 도메인 모두 Elasticsearch 버전 7.10 이상이거나 OpenSearch를 실행해야 합니다.
5. 클러스터 쿼리에 사용할 수 없는 클러스터를 건너뛰려면 사용할 수 없는 클러스터 건너뛰기를 선택합니다. 이 설정을 사용하면 하나 이상의 원격 클러스터에서 오류가 발생하더라도 클러스터 간 쿼리가 일부 결과를 반환할 수 있습니다.
6. 클러스터 간 검색은 먼저 연결 요청을 검증하여 전제 조건이 충족되는지 확인합니다. 도메인이 호환되지 않는 것으로 확인되면 연결 요청이 `Validation failed` 상태로 들어갑니다.
7. 연결 요청이 성공적으로 검증되면 대상 도메인으로 전송되어 승인을 받아야 합니다. 이 승인이 이루어질 때까지 연결은 `Pending acceptance` 상태로 유지됩니다. 대상 도메인에서 연결 요청이 수락되면 상태가 `Active`으로 변경되고 대상 도메인을 쿼리에 사용할 수 있게 됩니다.
   - 도메인 페이지에는 대상 도메인의 전체 도메인 상태 및 인스턴스 상태 세부 정보가 표시됩니다. 도메인 소유자만 도메인과의 연결을 유연하게 생성하고 보고 제거하고 모니터링할 수 있습니다.

연결이 설정되면 연결된 도메인의 노드 간에 흐르는 모든 트래픽이 암호화됩니다. VPC 도메인을 VPC가 아닌 도메인에 연결하고 VPC가 아닌 도메인이 인터넷에서 트래픽을 수신할 수 있는 퍼블릭 엔드포인트인 경우, 도메인 간의 클러스터 간 트래픽은 여전히 암호화되고 안전합니다.

연결 제거

연결을 제거하면 인덱스에 대한 교차 클러스터 작업이 중지됩니다.

1. 도메인 대시보드에서 [연결(Connections)] 탭으로 이동합니다.
2. 제거할 도메인 연결을 선택하고 삭제(Delete)를 선택한 다음 삭제를 확인합니다.

소스 도메인이나 대상 도메인에서 이러한 단계를 수행하여 연결을 제거할 수 있습니다. 연결을 제거한 후에도 15일 동안 `Deleted` 상태로 계속 표시됩니다.

활성 클러스터 간 연결이 있는 도메인은 삭제할 수 없습니다. 도메인을 삭제하려면 먼저 해당 도메인과의 수신 연결과 발신 연결을 모두 제거합니다. 그러면 도메인을 삭제하기 전에 클러스터 간 도메인 사용자를 고려할 수 있습니다.

보안 설정 및 샘플 시연

1. 소스 도메인에 클러스터 간 검색 요청을 보냅니다.
2. 소스 도메인은 해당 도메인 액세스 정책을 기준으로 해당 요청을 평가합니다. 클러스터 간 검색에는 세분화된 액세스 제어가 필요하므로 소스 도메인에서 오픈 액세스 정책을 사용하는 것이 좋습니다.

   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**
>
> 경로에 원격 인덱스를 포함하는 경우 도메인 ARN에서 URI를 URL로 인코딩해야 합니다. 예를 들어 `arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst:remote_index` 대신 `arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst%3Aremote_index`를 사용합니다.

세분화된 액세스 제어 외에 제한적인 액세스 정책을 사용하도록 선택하는 경우 정책에서 최소한 `es:ESHttpGet`에 대한 액세스를 허용해야 합니다.

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. 소스 도메인에 대한 세분화된 액세스 제어가 요청을 평가합니다.

   - 요청이 유효한 IAM 또는 HTTP 기본 자격 증명으로 서명되었습니까?
   - 그렇다면 사용자에게 검색을 수행하고 데이터에 액세스할 수 있는 권한이 있습니까?

   요청이 대상 도메인(예: `dest-alias:dest-index/_search`)의 데이터만 검색하는 경우 대상 도메인에 대한 사용 권한만 필요합니다.

요청이 두 도메인(예: source-index,dest-alias:dest-index/_search)에서 데이터를 검색하는 경우 두 도메인에 대한 사용 권한이 필요합니다.

세분화된 액세스 제어에서 사용자는 관련 인덱스에 대한 표준 read 또는 search 권한 외에 indices:admin/shards/search_shards 권한도 있어야 합니다.

4. 소스 도메인은 요청을 대상 도메인에 전달합니다. 대상 도메인은 해당 도메인 액세스 정책을 기준으로 이 요청을 평가합니다. 대상 도메인에 대한 es:ESCrossClusterGet 권한을 포함해야 합니다.

   JSON

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "AWS": "*"
         },
         "Action": "es:ESCrossClusterGet",
         "Resource": "arn:aws:es:us-east-1:111122223333:domain/dst-domain"
       }
     ]
   }
   ```

   es:ESCrossClusterGet 권한이 /dst-domain/*이 아닌 /dst-domain에 적용되었는지 확인합니다.

   그러나 이 최소 정책은 클러스터 간 검색만 허용합니다. 문서 인덱싱 및 표준 검색 수행과 같은 다른 작업을 수행하려면 추가 권한이 필요합니다. 대상 도메인에서 다음 정책을 사용하는 것이 좋습니다.

   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**
>
> 도메인 간의 모든 클러스터 간 검색 요청은 기본적으로 노드 간 암호화의 일부로 전송 중에 암호화됩니다.

5. 대상 도메인은 검색을 수행하고 결과를 소스 도메인에 반환합니다.
6. 소스 도메인은 자체 결과(있는 경우)를 대상 도메인의 결과와 결합하여 반환합니다.
7. 테스트 요청을 위해 Postman을 사용하는 것이 좋습니다.

   - 대상 도메인에서 문서를 인덱싱합니다.

   ```
   POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1

   {
     "Dracula": "Bram Stoker"
   }
   ```

   - 소스 도메인에서 이 인덱스를 쿼리하려면 쿼리 내에 대상 도메인의 연결 별칭을 포함합니다.

```
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"
      }
    }
  ]
}
```

도메인 대시보드의 연결(Connections) 탭에서 연결 별칭을 찾을 수 있습니다.

- 연결 별칭이 `cluster_b`인 `domain-a -> domain-b`와 연결 별칭이 `cluster_c`인 `domain-a -> domain-c` 간에 연결을 설정하는 경우, 다음과 같이 `domain-a`, `domain-b` 및 `domain-c`를 검색합니다.

```
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"
    }
  }
}
```

응답

```
{
  "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
```

```
        }
      }
    ]
  }
}
```

연결 설정에서 사용할 수 없는 클러스터를 건너뛰도록 선택하지 않은 경우 검색 요청이 성공적으로 실행되려면 검색하는 모든 대상 클러스터를 사용할 수 있어야 합니다. 그렇지 않으면 전체 요청이 실패합니다. 도메인 중 하나를 사용할 수 없더라도 검색 결과가 반환되지 않습니다.

OpenSearch Dashboards

`connection-alias:index`을(를) 사용하여 원격 인덱스에 액세스해야 한다는 점을 제외하면 연결된 여러 도메인의 데이터를 단일 도메인과 동일한 방식으로 시각화할 수 있습니다. 따라서 인덱스 패턴이 `connection-alias:index`와 일치해야 합니다.

Amazon OpenSearch Service용 순위 학습

OpenSearch는 BM-25라는 확률 순위 프레임워크를 사용하여 관련성 점수를 계산합니다. 문서에 고유 키워드가 더 자주 나타나는 경우 BM-25는 해당 문서에 더 높은 관련성 점수를 할당합니다. 그러나 이 프레임워크는 클릭 광고 데이터와 같은 사용자 동작을 고려하지 않으므로 관련성을 더욱 향상시킬 수 있습니다.

순위 학습은 기계 학습 및 행동 데이터를 사용하여 문서의 관련성을 조정할 수 있는 오픈 소스 플러그 인입니다. 이는 XGBoost 및 Ranklib 라이브러리의 모델을 사용하여 검색 결과를 다시 작성합니다. Elasticsearch LTR 플러그 인은 초기에 OpenSource Connections에 의해 개발되었으며, Wikimedia Foundation, Snagajob Engineering, Bonsai, Yelp Engineering에게서 중요한 기여를 받았습니다. 플러그 인의 OpenSearch 버전은 Elasticsearch LTR 플러그 인에서 파생됩니다.

순위 학습에는 OpenSearch 또는 Elasticsearch 7.7 이상이 필요합니다. 순위 학습 플러그인을 사용하려면 전체 관리자 권한이 있어야 합니다. 자세한 내용은 the section called “마스터 사용자 수정” 섹션을 참조하세요.

> **Note**
>
> 이 설명서는 순위 학습 플러그인에 대한 일반적인 개요를 제공하고 이를 시작하는 데 도움이 됩니다. 자세한 단계 및 API 설명을 포함한 전체 설명서는 순위 학습 설명서에서 확인할 수 있습니다.

주제

- 순위 학습 시작하기
- 순위 학습 API

순위 학습 시작하기

판단 목록을 제공하고, 교육 데이터 세트를 준비하며, Amazon OpenSearch Service 외부에서 모델을 교육해야 합니다. 파란색으로 표시된 부분은 OpenSearch Service 외부에서 발생합니다.

1단계: 플러그인 초기화

순위 학습 플러그인을 초기화하려면 OpenSearch Service 도메인으로 다음 요청을 보냅니다.

```
PUT _ltr
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : ".ltrstore"
}
```

이 명령은 기능 집합 및 모델과 같은 메타데이터 정보를 저장하는 숨겨진 `.ltrstore` 인덱스를 생성합니다.

2단계: 판단 목록 생성

> **Note**
>
> OpenSearch Service 외부에서 이 단계를 수행해야 합니다.

판단 목록은 기계 학습 모델이 학습하는 예제 모음입니다. 판단 목록에는 중요한 키워드와 각 키워드에 대한 등급 문서 세트가 포함되어야 합니다.

이 예제에서는 영화 데이터 집합에 대한 판단 목록이 있습니다. 4등급은 완벽한 일치를 나타냅니다. 0등급은 최악의 일치를 나타냅니다.

학년	키워드	문서 ID	영화 이름
4	rambo	7555	Rambo
3	rambo	1370	Rambo III
3	rambo	1369	Rambo: First Blood Part II
3	rambo	1368	First Blood

다음 형식으로 판단 목록을 준비합니다.

```
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"
```

판단 목록의 더욱 완벽한 예제는 영화 판단을 참조하세요.

인간 주석자의 도움을 받아 이 판단 목록을 수동으로 작성하거나 분석 데이터에서 프로그래밍 방식으로 추론할 수 있습니다.

3단계: 기능 집합 작성

기능은 문서의 관련성에 해당하는 필드입니다(예:`title`, `overview`, `popularity score`(뷰 수) 등).

각 기능에 대한 Mustache 템플릿을 사용하여 기능 집합을 작성합니다. 기능에 대한 자세한 내용은 기능 작업을 참조하세요.

이 예제에서는 `title` 및 `overview` 필드를 사용하여 `movie_features` 기능 집합을 작성합니다.

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

원본 `.ltrstore` 인덱스를 쿼리하는 경우 기능 집합을 가져옵니다.

```
GET _ltr/_featureset
```

4단계: 기능 값 로그

기능 값은 각 기능에 대해 BM-25에서 계산한 관련성 점수입니다.

기능 집합과 판단 목록을 결합하여 기능 값을 로그합니다. 로깅 기능에 대한 자세한 내용은 기능 점수 로깅을 참조하세요.

이 예제에서 `bool` 쿼리는 필터를 사용하여 등급이 매겨진 문서를 검색한 다음 `sltr` 쿼리로 기능 집합을 선택합니다. `ltr_log` 쿼리는 문서와 해당 기능 값을 로그하는 기능을 결합합니다.

```
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"
      }
    }
  }
}
```

샘플 응답은 다음과 같습니다.

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

앞의 예제에서는 ID가 1368인 문서의 제목 필드에 “rambo”라는 키워드가 나타나지 않기 때문에 첫 번째 기능에는 기능 값이 없습니다. 이 값은 교육 데이터에서 누락된 기능 값입니다.

5단계: 교육 데이터 세트 생성

> **Note**
>
> OpenSearch Service 외부에서 이 단계를 수행해야 합니다.

다음 단계는 판단 목록과 기능 값을 결합하여 교육 데이터 집합을 만드는 것입니다. 원래 판단 목록이 다음과 같은 경우:

```
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
```

다음과 같은 최종 교육 데이터 집합으로 변환합니다.

```
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
```

이 단계를 수동으로 수행하거나 프로그램을 작성하여 자동화할 수 있습니다.

6단계: 알고리즘 선택 및 모델 구축

> **Note**
>
> OpenSearch Service 외부에서 이 단계를 수행해야 합니다.

교육 데이터 집합을 마련한 다음 단계는 XGBoost 또는 Ranklib 라이브러리를 사용하여 모델을 구축하는 것입니다. XGBoost 및 Ranklib 라이브러리를 사용하면 LambdaMART, Random Forests 등과 같은 인기 모델을 구축할 수 있습니다.

XGBoost와 Ranklib를 사용하여 모델을 구축하는 단계는 XGBoost 및 RankLib 설명서를 각각 참조하세요. Amazon SageMaker를 사용하여 XGBoost 모델을 구축하려면 XGBoost 알고리즘을 참조하세요.

7단계: 모델 배포

모델을 구축한 후 순위 학습 플러그인에 배포합니다. 모델 배포에 대한 자세한 내용은 훈련된 모델 업로드를 참조하세요.

이 예제에서는 Ranklib 라이브러리를 사용하여 `my_ranklib_model` 모델을 구축합니다.

```
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>
"""
    }
  }
}
```

모델을 보려면 다음 요청을 보냅니다.

```
GET _ltr/_model/my_ranklib_model
```

8단계: 순위 학습으로 검색

모델을 배포하면 검색할 준비가 됩니다.

사용 중인 기능 및 실행하려는 모델의 이름으로 `sltr` 쿼리를 수행합니다.

```
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"
        }
      }
    }
  }
}
```

“Rambo”를 판단 목록에서 최고 등급으로 지정했기 때문에 순위 학습에서 “Rambo”를 첫 번째 결과로 볼 수 있습니다.

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

순위 학습 플러그인을 사용하지 않고 검색하는 경우 OpenSearch는 다른 결과를 반환합니다.

```
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"
        }
```

```
      }
    ]
  }
}
```

모델이 얼마나 잘 작동하는지에 대한 의견에 따라 판단 목록과 기능을 조정합니다. 그런 다음 2~8단계를 반복하여 시간에 따른 순위 결과를 개선합니다.

순위 학습 API

순위 학습 작업을 사용하여 기능 집합 및 모델을 프로그래밍 방식으로 작업할 수 있습니다.

스토어 생성

기능 집합 및 모델과 같은 메타데이터 정보를 저장하는 숨겨진 `.ltrstore` 인덱스를 생성합니다.

```
PUT _ltr
```

스토어 삭제

숨겨진 `.ltrstore` 인덱스를 삭제하고 플러그인을 재설정합니다.

```
DELETE _ltr
```

기능 집합 생성

기능 집합을 생성합니다.

```
POST _ltr/_featureset/<name_of_features>
```

기능 집합 삭제

기능 집합을 삭제합니다.

```
DELETE _ltr/_featureset/<name_of_feature_set>
```

기능 집합 가져오기

기능 집합을 검색합니다.

```
GET _ltr/_featureset/<name_of_feature_set>
```

모델 생성

모델을 생성합니다.

```
POST _ltr/_featureset/<name_of_feature_set>/_createmodel
```

모델 삭제

모델을 삭제합니다.

```
DELETE _ltr/_model/<name_of_model>
```

모델 가져오기

모델을 검색합니다.

```
GET _ltr/_model/<name_of_model>
```

통계 가져오기

플러그인이 작동하는 방법에 대한 정보를 제공합니다.

```
GET _ltr/_stats
```

필터를 사용하여 단일 통계를 검색할 수도 있습니다.

```
GET _ltr/_stats/<stat>
```

또한 정보를 클러스터의 단일 노드로 제한할 수 있습니다.

```
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
    }
  }
}
```

통계는 다음 표에 지정된 대로 노드 및 클러스터의 두 수준에서 제공됩니다.

노드 수준 통계

필드 이름	설명
request_total_count	순위 요청의 총 수입니다.
request_error_count	실패한 요청의 총 수입니다.
cache	모든 캐시(기능, 기능 집합, 모델)에 대한 통계입니다. 캐시 적중은 사용자가 플러그인을 쿼리하고 모델이 이미 메모리에 로드되었을 때 발생합니다.
cache.eviction_count	캐시 제거 횟수입니다.
cache.hit_count	캐시 적중 횟수입니다.
cache.miss_count	캐시 누락 횟수입니다. 캐시 누락은 사용자가 플러그인을 쿼리하고 모델이 아직 메모리에 로드되지 않았을 때 발생합니다.
cache.entry_count	캐시의 항목 수입니다.
cache.memory_usage_in_bytes	사용된 총 메모리(바이트)입니다.
cache.cache_capacity_reached	캐시 제한에 도달했는지를 나타냅니다.

클러스터 수준 통계

필드 이름	설명
스토어	기능 집합과 모델 메타데이터가 저장되는 위치를 나타냅니다. (기본값은 “.ltrstore”입니다. 그렇지 않으면 접두사가 “.ltrstore_”이고 사용자가 제공한 이름이 붙습니다.)
stores.status	인덱스 상태입니다.
stores.feature_sets	기능 세트 수입니다.

필드 이름	설명
stores.features_count	기능 수입니다.
stores.model_count	모델 수입니다.
상태	특성 저장소 인덱스(빨간색, 노란색 또는 녹색) 및 회로 차단기 상태(열림 또는 닫힘)를 기반으로 하는 플러그인 상태입니다.
cache.cache_capacity_reached	캐시 제한에 도달했는지를 나타냅니다.

캐시 통계 가져오기

캐시 및 메모리 사용에 대한 통계를 반환합니다.

```
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": {
                ...
            }
        }
    }
}
```

캐시 지우기

플러그인 캐시를 지웁니다. 이 옵션을 사용하여 모델을 새로 고칩니다.

```
POST _ltr/_clearcache
```

Amazon OpenSearch Service의 비동기 검색

Amazon OpenSearch Service에 대한 비동기 검색을 사용하면 백그라운드에서 실행되는 검색 쿼리를 제출하고 요청 진행 상황을 모니터링하며 이후 단계에서 결과를 검색할 수 있습니다. 검색이 완료되기 전에 사용할 수 있게 되므로 부분 결과를 검색할 수 있습니다. 검색이 완료된 후 나중에 검색 및 분석할 수 있도록 결과를 저장합니다.

비동기 검색에는 OpenSearch 1.0 이상 또는 Elasticsearch 7.10 이상이 필요합니다.

이 설명서에서는 비동기 검색에 대한 간략한 개요를 제공합니다. 또한 오픈 소스 OpenSearch 클러스터가 아닌 관리형 Amazon OpenSearch Service 도메인에서 비동기 검색 사용 시 제한 사항에 대해서도 설명합니다. 사용 가능한 설정, 권한 및 전체 API 참조를 포함하여 비동기 검색에 대한 전체 설명서는 OpenSearch 설명서의 Asynchronous search를 참조하세요.

샘플 검색 호출

비동기 검색을 수행하려면 다음 형식을 사용하여 HTTP 요청을 `_plugins/_asynchronous_search`로 전송합니다.

```
POST opensearch-domain/_plugins/_asynchronous_search
```

> **Note**
>
> OpenSearch 버전 대신 Elasticsearch 7.10을 사용하고 있다면, 모든 비동기 검색 요청에서 `_plugins`를 `_opendistro`로 바꿉니다.

다음 비동기 검색 옵션을 지정할 수 있습니다.

옵션	설명	기본값	필수
`wait_for_completion_timeout`	결과를 기다릴 시간을 지정합니다. 일반 검색과 마찬가지로 이 시간 내에 얻은 결과를 확인할 수 있습니다. ID를 기반으로 나머지 결과를 폴링할 수 있습니다. 최댓값은 300초입니다.	1초	아니요
`keep_on_completion`	검색이 완료된 후 결과를 클러스터에 저장할지를 지정합니다. 나중에 저장된 결과를 검토할 수 있습니다.	`false`	아니요
`keep_alive`	결과가 클러스터에 저장되는 시간을 지정합니다. 예를 들어 2d는 결과가 48시간 동안 클러스터에 저장됨을 의미합니다. 이 기간 이후 또는 검색이 취소된 경우 저장된 검색 결과가 삭제됩니다. 여기에는 쿼리 런타임이 포함됩니다. 쿼리가 이 시간을 초과하면 프로세스가 이 쿼리를 자동으로 취소합니다.	12시간	아니요

샘플 요청

```
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**
>
> 표준 _search 쿼리에 적용되는 모든 요청 파라미터가 지원됩니다. OpenSearch 버전 대신 Elasticsearch 7.10을 사용하고 있다면, _plugins를 _opendistro로 바꿉니다.

비동기 검색 권한

비동기 검색은 세분화된 액세스 제어를 지원합니다. 사용 사례에 맞게 권한을 혼합하고 일치시키는 방법에 대한 자세한 내용은 비동기 검색 보안을 참조하세요.

세분화된 액세스 제어가 활성화된 도메인의 경우 역할에 대해 다음과 같은 최소 권한이 필요합니다.

```
# 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'
```

세분화된 액세스 제어가 비활성화된 도메인의 경우 IAM 액세스 및 보안 키를 사용하여 모든 요청에 서명합니다. 비동기 검색 ID를 사용하여 결과에 액세스할 수 있습니다.

비동기 검색 설정

OpenSearch를 사용하면 _cluster/settings API를 사용하여 사용 가능한 모든 비동기 검색 설정을 변경할 수 있습니다. OpenSearch Service에서는 다음 설정만 변경할 수 있습니다.

- plugins.asynchronous_search.node_concurrent_running_searches
- plugins.asynchronous_search.persist_search_failures

클러스터 간 검색

다음과 같은 사소한 제한 사항과 함께 클러스터 전체에서 비동기 검색을 수행할 수 있습니다.

- 소스 도메인에서만 비동기 검색을 실행할 수 있습니다.
- 클러스터 간 검색 쿼리의 일부로 네트워크 왕복을 최소화할 수 없습니다.

연결 별칭이 cluster_b인 domain-a -> domain-b와 연결 별칭이 cluster_c인 domain-a -> domain-c 간에 연결을 설정하는 경우, 다음과 같이 domain-a, domain-b 및 domain-c를 비동기 검색합니다.

```
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": []
    }
  }
}
```

응답

```
{
  "id" :
 "Fm9pYzJyVG91U19xb0hIQUJnMHJfRFEAAAAAAAknghQ1OWVBczNZQjVEa2dMYTBXaTdEagAAAAAAAAAB",
  "state" : "RUNNING",
  "start_time_in_millis" : 1609329314796,
  "expiration_time_in_millis" : 1609761314796
}
```

자세한 내용은 the section called “클러스터 간 검색” 단원을 참조하십시오.

UltraWarm

UltraWarm 인덱스를 사용한 비동기 검색은 계속 작동합니다. 자세한 내용은 the section called “UltraWarm 스토리지” 단원을 참조하십시오.

> **Note**
>
> CloudWatch에서 비동기 검색 통계를 모니터링할 수 있습니다. 전체 지표 목록은 the section called “비동기 검색 지표” 섹션을 참조하세요.

Amazon OpenSearch Service의 특정 시점 검색

특정 시점(PIT) 기능은 고정된 데이터세트에 대해 다양한 쿼리를 실행할 수 있는 검색 유형입니다. 문서가 계속해서 인덱싱, 업데이트 및 삭제되기 때문에 서로 다른 시점에 동일한 인덱스에서 동일한 쿼리를 실행하면 다른 결과가 나타나는 경우가 일반적입니다. PIT를 사용하면 데이터 세트의 상수 상태를 기준으로 쿼리할 수 있습니다.

PIT 검색의 주요 용도는 `search_after` 기능과 결합하는 것입니다. 이는 OpenSearch에서 선호되는 페이지 매김 메서드이며, 시간 제한이 있는 데이터 집합에서 작동하고 쿼리에 바인딩되지 않으며 앞뒤로 일관된 페이지 매김을 지원하기 때문에 특히 딥 페이지 매김의 경우에 선호됩니다. OpenSearch 버전 2.5를 실행하는 도메인에서 PIT를 사용할 수 있습니다.

> **Note**
>
> 이 주제에서는 PIT 개요 그리고 자체 관리형 OpenSearch 클러스터가 아닌 관리형 Amazon OpenSearch Service 도메인에서 PIT를 사용할 때 고려해야 할 몇 가지 사항을 제공합니다. 포괄적인 API 참조를 포함한 PIT의 전체 설명서는 오픈 소스 OpenSearch 설명서의 특정 시점을 참조하세요.

고려 사항

PIT 검색을 구성할 때 다음 사항을 고려하세요.

- OpenSearch 버전 2.3을 실행하는 도메인에서 업그레이드하고 PIT 작업에 대한 세분화된 액세스 제어가 필요한 경우 해당 작업과 역할을 수동으로 추가해야 합니다.
- PIT에 대한 복원성이 없습니다. 노드 재부팅, 노드 종료, 블루/그린 배포, EOpenSearch 프로세스 재시작으로 인해 모든 PIT 데이터가 손실됩니다.
- 블루/그린 배포 중에 샤드가 재배치되는 경우 라이브 데이터 세그먼트만 새 노드로 전송됩니다. PIT가 보유한 샤드 세그먼트(단독 및 라이브 데이터와 공유된 샤드 세그먼트 모두)는 이전 노드에 그대로 남아 있습니다.

- PIT 검색은 현재 비동기 검색에서는 작동하지 않습니다.

PIT 생성

PIT 쿼리를 실행하려면 다음 형식을 사용하여 _search/point_in_time에 HTTP 요청을 전송합니다.

```
POST opensearch-domain/my-index/_search/point_in_time?keep_alive=time
```

다음 PIT 옵션을 지정할 수 있습니다.

옵션	설명	기본값	필수
keep_alive	PIT를 보존하는 시간입니다. 검색 요청으로 PIT에 액세스할 때마다 PIT 수명이 keep_alive 파라미터와 동일한 시간만큼 연장됩니다. 이 쿼리 파라미터는 PIT를 생성할 때는 필수이지만 검색 요청에서는 선택 사항입니다.		예
preference	검색을 수행하는 데 사용되는 노드 또는 샤드를 지정하는 문자열입니다.	무작위	아니요
routing	검색 요청을 특정 샤드로 라우팅하도록 지정하는 문자열입니다.	문서의 _id	아니요
expand_wildcards	와일드카드 패턴과 일치할 수 있는 인덱스 유형을 지정하는 문자열입니다. 쉼표로 분리된 값을 지원합니다. 유효한 값은 다음과 같습니다. • all: 숨겨진 인덱스 또는 데이터 스트림을 포함하여 모든 인덱스 또는 데이터 스트림을 일치시킵니다. • open: 열려 있거나 숨겨지지 않은 인덱스 또는 숨겨지지 않은 데이터 스트림을 일치시킵니다.	open	아니요

옵션	설명	기본값	필수
	• closed: 닫혀 있고 숨겨지지 않은 인덱스 또는 숨겨지지 않은 데이터 스트림을 일치시킵니다. • hidden: 숨겨진 인덱스 또는 데이터 스트림을 일치시킵니다. 개방형, 폐쇄형 또는 개방형 및 폐쇄형 모두와 결합해야 합니다. • none: 와일드카드 패턴은 허용되지 않습니다.		
allow_partial_pit_creation	부분 오류가 있는 PIT를 생성할지 여부를 지정하는 부울입니다.	true	아니요

샘플 응답

```
{
    "pit_id":
 "o463QQEPbXktaW5kZXgtMDAwMDAxFnNOWU43ckt3U3IyaFVpbGE1UWEtMncAFjFyeXBsRGJmVFM2RTB6eVg1aVVqQncAA
    "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
    },
    "creation_time": 1658146050064
}
```

PIT를 생성하면 응답으로 PIT ID를 받게 됩니다. PIT로 검색을 수행하는 데 사용하는 ID입니다.

특정 시점 권한

PIT는 세분화된 액세스 제어를 지원합니다. OpenSearch 버전 2.5 도메인으로 업그레이드하고 세분화된 액세스 제어가 필요한 경우 다음 권한이 있는 역할을 수동으로 생성해야 합니다.

```
# 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"
```

OpenSearch 버전 2.5 이상인 도메인의 경우 기본 제공 `point_in_time_full_access` 역할을 사용할 수 있습니다. 자세한 내용은 OpenSearch 설명서의 보안 모델을 참조하세요.

PIT 설정

OpenSearch를 사용하면 `_cluster/settings` API를 사용하여 사용 가능한 모든 PIT 설정을 변경할 수 있습니다. OpenSearch Service에서는 현재 설정을 수정할 수 없습니다.

클러스터 간 검색

다음과 같은 사소한 제한 사항을 제외하고 PIT를 생성하고, PIT ID로 검색하고, PIT를 나열하고, 클러스터 전체에서 PIT를 삭제할 수 있습니다.

- 소스 도메인에서만 PIT를 모두 나열하고 삭제할 수 있습니다.
- 클러스터 간 검색 쿼리의 일부로 네트워크 왕복을 최소화할 수 없습니다.

자세한 내용은 the section called “클러스터 간 검색” 단원을 참조하십시오.

UltraWarm

UltraWarm 인덱스를 사용한 PIT 검색은 계속 작동합니다. 자세한 내용은 the section called “UltraWarm 스토리지” 단원을 참조하십시오.

> **Note**
>
> CloudWatch에서 PIT 검색 통계를 모니터링할 수 있습니다. 전체 지표 목록은 the section called “특정 시점 지표” 섹션을 참조하세요.

Amazon OpenSearch Service의 의미 검색

OpenSearch 버전 2.9부터 시맨틱 검색을 사용하여 검색 쿼리를 이해하고 검색 관련성을 개선할 수 있습니다. 시맨틱 검색은 신경망 검색 및 k-Nearest Neighbor(k-NN)의 두 가지 방법 중 하나로 사용할 수 있습니다.

OpenSearch Service를 사용하면 AWS 서비스에 대한 AI 커넥터 및 외부 서비스를 설정할 수 있습니다. 콘솔을 사용하여 AWS CloudFormation 템플릿을 사용하여 ML 모델을 생성할 수도 있습니다. 자세한 내용은 the section called “CloudFormation 템플릿 통합” 단원을 참조하십시오.

시맨틱 검색을 사용하는 단계별 가이드를 포함한 시맨틱 검색에 대한 전체 설명서는 오픈 소스 OpenSearch 설명서의 Semantic search를 참조하세요.

Amazon OpenSearch Service에서 동시 세그먼트 검색

OpenSearch 버전 2.17부터 동시 세그먼트 검색은 새 설정을 사용하여 동시 검색 동작을 제어합니다.

- 버전 2.17로 생성된 새 도메인에는 2xl 이상의 노드에서 기본적으로 자동 모드로 기본 동시 세그먼트 검색이 설정되어 있습니다.
- 2.17로 업그레이드하는 기존 도메인에는 2xl 이상의 모든 노드에 대한 인스턴스 유형과 지난 1주 동안 클러스터의 전체 CPU 사용률이 45% 미만인 경우에 따라 기본 동시 세그먼트 검색이 자동으로 설정됩니다.
- 자세한 내용은 동시 세그먼트 검색 버전 2.17을 참조하세요.

OpenSearch 버전 2.13부터 동시 세그먼트 검색을 사용하여 쿼리 단계에서 세그먼트를 병렬로 검색할 수 있습니다. 동시 세그먼트 검색에 대한 전체 설명서는 오픈 소스 OpenSearch 설명서의 Concurrent segment search를 참조하세요. 동시 세그먼트 검색과 관련된 Amazon CloudWatch 지표에 대한 자세한 내용은 인스턴스 지표 및 UltraWarm 지표를 참조하세요.

Amazon OpenSearch Service에서 현재 세그먼트 검색을 사용할 때 적용되는 몇 가지 추가 제한 사항이 있습니다.

- OpenSearch Service에서는 동시 세그먼트 검색을 인덱스 수준에서 활성화할 수 없습니다.
- 기본적으로 OpenSearch Service는 최대 슬라이스 수 메커니즘과 함께 2개의 슬라이스 수를 사용합니다.

Amazon OpenSearch Service에서 자연어 쿼리 생성

Amazon OpenSearch Service의 자연어 쿼리 생성 기능을 사용하여 자연어를 통해 운영 및 보안 로그 데이터를 쿼리할 수 있습니다. OpenSearch는 확장성과 성능이 뛰어난 로그 분석 및 검색 엔진이므로 로그 데이터를 탐색하는 데 이상적인 옵션입니다. 이제 자연어를 사용하여 이러한 로그를 탐색할 수 있습니다. 이 기능을 사용하면 OpenSearch Piped Processing Language(PPL)에 의존하거나 쿼리를 빌드할 때 데이터 정의를 조회하지 않고도 문제를 식별할 수 있습니다. 버전 2.13 이상의 OpenSearch Service 도메인에서 자연어 쿼리 생성 기능을 사용할 수 있습니다. 세분화된 액세스 제어를 활성화해야 합니다.

이 기능은 OpenSearch Assistant Toolkit을 사용하여 빌드되었습니다. 대규모 언어 모델에 연결하는 유사한 기능을 생성하려면 툴킷을 사용하여 자체 에이전트와 도구를 구성할 수 있습니다.

사전 조건

자연어 쿼리 생성 기능을 사용하려면 먼저 도메인에 다음 조건이 갖추어져 있어야 합니다.

- 버전 2.13 이상.
- 서비스 소프트웨어 R20240520-P4 이상.
- 세분화된 액세스 제어가 활성화된 상태입니다. 자세한 내용은 the section called “세분화된 액세스 제어 활성화” 단원을 참조하십시오.

시작

자연어 쿼리 생성은 세분화된 액세스 제어가 활성화된 버전 2.13 이상으로 생성된 모든 도메인에서 기본적으로 활성화됩니다.

다른 도메인의 경우 자연어 쿼리 생성 및 Amazon Q Developer 기능 활성화를 선택하여 활성화합니다.

활성화한 후 OpenSearch 대시보드의 로그 페이지로 이동합니다. 이벤트 탐색기를 선택하고 쿼리 도우미에 질문합니다.

권한 구성

기존 OpenSearch Service 도메인에서 자연어 쿼리 생성을 활성화하면 도메인에 query_assistant_access 역할이 정의되지 않았을 수 있습니다. 관리자가 아닌 사용자는 이 역할에 매핑되어 세분화된 액세스 제어를 사용하는 도메인의 원 인덱스를 관리해야 합니다. 수동으로 query_assistant_access 역할을 생성하려면 다음 단계를 수행합니다.

1. OpenSearch 대시보드에서 보안으로 이동하고 역할을 선택합니다.
2. 역할 생성을 선택하고 다음 클러스터 권한을 구성합니다.
   - `cluster:admin/opensearch/ml/config/get`
   - `cluster:admin/opensearch/ml/execute`
   - `cluster:admin/opensearch/ml/predict`
   - `cluster:admin/opensearch/ppl`
3. 역할 이름을 query_assistant_access로 지정합니다.
4. 역할 생성을 선택합니다. 이제 query_assistant_access 역할을 사용할 수 있습니다.

> **Note**
>
> 자연어 질문을 사용하려는 인덱스에 대한 `indices:admin/mappings/get` 및 `read` 인덱스 권한도 있어야 합니다.

구성 자동화

Flow Framework는 쿼리 생성 및 대화형 채팅과 같은 사용 사례에 대한 OpenSearch 구성을 자동화하는 방법을 제공하는 OpenSearch 플러그인입니다. 플러그인은 자연어 쿼리 생성 기능을 활성화하는 리소스를 추적하므로 흐름 프레임워크 인덱스는 쿼리 지원을 사용하는 각 도메인에 대한 템플릿을 저장합니다.

흐름 프레임워크를 사용하면 사전 정의된 템플릿 세트에서 선택하거나 생성형 모델의 백엔드로 OpenSearch를 준비하는 기계 학습 커넥터, 도구, 에이전트 및 기타 구성 요소에 대한 자체 자동화를 생성할 수 있습니다.

벡터 검색

Amazon OpenSearch Service의 벡터 검색을 사용하면 기존 키워드 일치 대신 기계 학습 임베딩을 사용하여 의미상 유사한 콘텐츠를 검색할 수 있습니다. 벡터 검색은 데이터(텍스트, 이미지, 오디오 등)를 콘텐츠의 의미론적 의미를 캡처하는 고차원 숫자 벡터(임베딩)로 변환합니다. 검색을 수행할 때 OpenSearch는 쿼리의 벡터 표현을 저장된 벡터와 비교하여 가장 유사한 항목을 찾습니다.

벡터 검색에는 다음과 같은 주요 구성 요소가 포함됩니다.

벡터 필드

OpenSearch는 `knn_vector` 필드 유형을 지원하여 구성 가능한 차원(최대 16,000개)이 있는 고밀도 벡터를 저장합니다.

검색 방법

- k-NN(k-nearest neighbors): 가장 유사한 k 벡터를 찾습니다.
- 대략적인 k-NN: 대규모 데이터 세트에서 더 빠른 검색을 위해 HNSW(Hierarchical Navigable Small World)와 같은 알고리즘 사용

거리 지표

다음을 포함한 다양한 유사성 계산을 지원합니다.

- 유클리드 거리
- 코사인 유사성
- 점 제품

일반 사용 사례

벡터 검색은 다음과 같은 일반적인 사용 사례를 지원합니다.

- 의미 체계 검색: 키워드 일치뿐만 아니라 유사한 의미의 문서 찾기
- 권장 시스템: 유사한 제품, 콘텐츠 또는 사용자 제안
- 이미지 검색: 시각적으로 유사한 이미지 찾기
- 이상 탐지: 데이터 패턴의 이상치 식별
- RAG(Retrieval Augmented Generation): 관련 컨텍스트를 사용하여 LLM 응답 개선

기계 학습과 통합

OpenSearch는 다음 기계 학습 서비스 및 모델과 통합됩니다.

- Amazon Bedrock: 파운데이션 모델을 사용하여 임베딩 생성
- Amazon SageMaker AI: 사용자 지정 ML 모델 배포용
- Hugging Face 모델: 사전 훈련된 임베딩 모델
- 사용자 지정 모델: 자체 훈련된 임베딩 모델

벡터 검색을 사용하면 기존의 텍스트 일치 기능을 훨씬 넘어서 컨텍스트와 의미를 이해하는 정교한 AI 기반 애플리케이션을 구축할 수 있습니다.

(미리 보기) Amazon S3 벡터와 OpenSearch Service 통합

Amazon OpenSearch Service는 다음과 같은 방식으로 Amazon S3 벡터와 통합됩니다.

- (미리 보기) Amazon S3 벡터에서 OpenSearch Serverless로 가져오기
- (미리 보기) Amazon S3 벡터 엔진을 사용한 고급 검색 기능

> **Important**
>
> Amazon S3 Vectors와 OpenSearch Service의 통합은 평가판 릴리스이며 변경될 수 있습니다.

(미리 보기) Amazon S3 벡터에서 OpenSearch Serverless로 가져오기

> **Important**
>
> Amazon S3 Vectors와 OpenSearch Service의 통합은 평가판 릴리스이며 변경될 수 있습니다.

Amazon S3 Vectors는 벡터 저장 및 쿼리를 기본적으로 지원하는 첫 번째 클라우드 객체 스토어를 제공합니다. S3 벡터는 의미와 유사성을 기반으로 쿼리할 수 있는 비용 효율적이고 탄력적이며 내구성이 뛰어난 벡터 스토리지를 제공합니다. 1초 미만의 쿼리 응답 시간과 벡터 업로드, 저장 및 쿼리 비용을 최대 90% 절감합니다.

Amazon S3 Vectors에는 인프라를 프로비저닝하지 않고도 벡터 데이터를 저장, 액세스 및 쿼리하는 데 사용할 수 있는 S3 벡터 버킷이 도입되었습니다. 벡터 버킷 내에서 벡터 인덱스 내에서 벡터 데이터를 구성할 수 있습니다. 벡터 버킷에는 여러 벡터 인덱스가 있을 수 있으며 각 벡터 인덱스에는 수백만 개

의 벡터가 있을 수 있습니다. 자세한 내용은 Amazon S3 사용 설명서의 Amazon S3 벡터 및 벡터 버킷 작업을 참조하세요. Amazon S3

각 벡터는 다음으로 구성됩니다.

- 고유 키
- 벡터 데이터
- JSON 형식의 선택적 메타데이터

벡터 인덱스는 유사성 검색 작업을 위해 유클리드 및 코사인 거리 함수를 지원합니다.

> **Note**
>
> 벡터 버킷의 주요 이점은 벡터 작업에 대한 직접 API 액세스를 제공하면서 매우 저렴한 비용으로 대량 데이터 세트를 저장하는 기능입니다.

생성 방법을 포함하여 Amazon S3 벡터 버킷에 대한 자세한 내용은 Amazon S3 사용 설명서의 Amazon S3 벡터 및 벡터 버킷 작업을 참조하세요. Amazon S3 이 주제에 설명된 것 이상의 OpenSearch Service와의 통합에 대한 자세한 내용은 OpenSearch Service에서 S3 벡터 사용을 참조하세요.

Amazon OpenSearch Service와 함께 S3 Vectors를 사용하면 쿼리 빈도가 낮을 때 벡터 스토리지 비용을 절감한 다음 수요 증가에 따라 해당 데이터 세트를 OpenSearch로 빠르게 이동하거나 검색 기능을 개선할 수 있습니다.

OpenSearch Service는 Amazon S3 벡터와 통합되어 Amazon S3 벡터 버킷이 자체적으로 제공하는 것 이상의 향상된 성능과 기능을 제공합니다. 필요한 경우이 통합을 고려하세요.

- 쿼리 처리량 증가
- 1초 미만의 검색 지연 시간
- 집계와 같은 고급 분석 기능
- 텍스트 및 벡터 데이터를 결합한 하이브리드 검색

이 통합은 여러 애플리케이션이 성능 요구 사항이 다른 동일한 벡터 데이터를 사용할 때 특히 유용합니다. 일부 애플리케이션은 비용에 민감한 사용 사례를 위해 Amazon S3 벡터 버킷과 직접 상호 작용하는 반면, 성능에 중요한 작업을 위해 OpenSearch 통합을 활용하는 애플리케이션도 있습니다.

통합 아키텍처

통합은 Amazon OpenSearch Ingestion(OSI)을 Amazon S3 벡터 인덱스와 Amazon OpenSearch Serverless 벡터 컬렉션 간의 데이터 파이프라인으로 사용합니다. OpenSearch Ingestion은 지정된 벡터 인덱스에서 벡터 데이터를 자동으로 내보내고 고성능 검색 작업을 위해 이를 OpenSearch Serverless 벡터 컬렉션으로 수집합니다.

> **Note**
>
> 내보내기 후에도 데이터는 S3 벡터 인덱스에 계속 존재합니다. 데이터 사본이 두 개 있습니다.

각 벡터 인덱스는 OpenSearch Service 컬렉션의 해당 인덱스에 매핑됩니다. 통합:

- 벡터 차원 보존
- 메타데이터를 유지합니다.
- OpenSearch의 벡터 검색 기능을 위한 데이터 구조 최적화

구성 후 OpenSearch Ingestion은 Amazon S3 ListVectors API를 사용하여 지정된 벡터 인덱스의 벡터를 사용하여 데이터 내보내기 프로세스를 시작합니다. 이 서비스는 벡터를 병렬로 처리하여 OpenSearch Ingestion과 Amazon OpenSearch Serverless 모두의 조정 제한을 준수하면서 수집 속도를 최적화합니다.

수집 중에 서비스는 다음을 수행합니다.

- OpenSearch Service의 예상 형식과 일치하도록 벡터 데이터를 변환합니다.
- 벡터 값, 메타데이터 및 거리 지표를 포함한 필수 정보를 보존합니다.
- 지능형 재시도 메커니즘을 통해 장애 시나리오 처리
- 이후 분석을 위해 배달 못한 편지 대기열로 사용되는 Amazon S3 버킷에 문제가 있는 레코드를 배치합니다.

통합은 벡터 차원, 데이터 세트 크기 및 구성된 조정 제한에 따라 성능을 발휘하여 대규모 데이터 세트를 효율적으로 처리합니다. OSI는 파이프라인당 최대 16명의 작업자를 확장할 수 있으며, OpenSearch Serverless는 수집 수요에 따라 용량을 자동으로 조정합니다. 기본적으로 OpenSearch는 `maxSearch` OpenSearch Serverless 측의 OpenSearch 전산 단위(OCU)를 100으로 늘립니다.

> **Note**
>
> 통합은 다음을 통해 비용 효율성을 우선시합니다.
>
> - 내보내기 완료 후 자동 파이프라인 종료
> - OpenSearch Serverless 컬렉션 조정
> - Pay-per-use 리소스 모델

필수 IAM 권한

통합을 위해서는 서비스 간의 보안 통신을 활성화하기 위해 IAM 권한을 신중하게 구성해야 합니다. OpenSearch Ingestion에는 Amazon S3 벡터 인덱스에서 읽고, OpenSearch Service 벡터 컬렉션에 쓰고, 관련 보안 정책을 관리할 수 있는 권한이 필요합니다.

이 주제의 뒷부분에서 절차를 사용하여 통합을 활성화하면 권한 관리를 위해 다음 옵션 중 하나를 선택할 수 있습니다.

- 시스템이 필요한 권한을 가진 서비스 역할을 자동으로 생성하도록 허용
- 요구 사항을 충족하는 기존 역할 제공

자동으로 생성된 역할에는 다음에 대한 정책이 포함됩니다.

- Amazon S3 벡터 인덱스 APIs 액세스
- OpenSearch Service 컬렉션 작업 관리
- 실패한 수집 시도에 대한 배달 못한 편지 대기열 작업 처리

기존 역할을 지정하기로 선택한 경우 역할에 다음 IAM 권한이 있는지 확인합니다.

(필수): OpenSearch Ingestion과 OpenSearch Serverless 간의 데이터 파이프라인 권한

JSON

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

```
            "Sid": "allowAPIs",
            "Effect": "Allow",
            "Action": [ "aoss:APIAccessAll", "aoss:BatchGetCollection" ],
            "Resource": [ "arn:aws:aoss:*:account-id:collection/collection-id" ]
        },
        {
            "Sid": "allowSecurityPolicy",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:UpdateSecurityPolicy",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*",
            "Condition":{
               "StringLike":{
                  "aoss:collection": [ "collection-name" ]
                },
               "StringEquals": {
                  "aws:ResourceAccount": [ "account-id" ]
               }
            }
        }
    ]
}
```

(필수): OpenSearch Ingestion과 Amazon S3 배달 못한 편지 대기열 간의 데이터 수집 권한

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "s3Access",
            "Effect": "Allow",
            "Action": [
              "s3:PutObject"
            ],
            "Resource": [ "arn:aws:s3:::bucket/*" ]
        }
    ]
```

```
}
```

(필수): OpenSearch Ingestion과 Amazon S3 Vectors 간의 데이터 수집 권한

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3VectorIndexAccess",
            "Effect": "Allow",
            "Action": [
               "s3vectors:ListVectors",
               "s3vectors:GetVectors"
            ],
            "Resource": [
                "arn:aws:s3vectors:region:account-id:bucket/bucket-name/
index/index-name"
            ]
        }
    ]
}
```

(암 AWS KMS 호화가 활성화된 경우 필수): OpenSearch Ingestion과 Amazon S3 Vectors 간의 통신을 위한 복호화 권한

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowS3VectorDecryptionOfCustomManagedKey",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [ "arn:aws:kms:region:account-id:key/key-id" ],
```

```
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3vectors.region.amazonaws.com",
                    "kms:EncryptionContext:aws:s3vectors:arn":
 "arn:aws:s3vectors:region:account-id:bucket/bucket-name"
                }
            }
        }
    ]
}
```

OpenSearch와 Amazon S3 Vectors 통합 구성

다음 절차에 따라 OpenSearch Serverless와의 Amazon S3 Vectors 통합을 구성합니다.

> **Note**
>
> 벡터 버킷 페이지에서 OpenSearch로 내보내기 옵션을 선택하여 Amazon S3 콘솔에서 통합을 구성하는 프로세스를 시작한 경우 절차에 설명된 대로 다음 절차의 일부 단계는 해당되지 않습니다.

OpenSearch Serverless와 Amazon S3 Vectors 통합을 구성하려면

1. Amazon OpenSearch Service 콘솔에서 OpenSearch 벡터 엔진으로 S3 벡터 인덱스 가져오기 페이지를 엽니다. OpenSearch Amazon S3 콘솔에서 OpenSearch로 내보내기를 클릭하면 페이지가 자동으로 표시됩니다. OpenSearch 콘솔에서 시작하는 경우 왼쪽 탐색 창에서 통합을 선택한 다음 S3 벡터 인덱스 가져오기를 선택합니다.
2. 소스 섹션에서 Amazon S3 콘솔에서 시작한 경우 벡터 인덱스의 이름과 해당 Amazon 리소스 이름(ARN)이 이미 지정되어 있는지 확인합니다. OpenSearch 콘솔에서 시작한 경우 S3 벡터 인덱스 ARN 필드에 인덱스 ARN을 입력합니다.
3. 서비스 액세스 섹션에서 옵션을 선택합니다. 기존 역할을 선택하는 경우에 설명된 대로 통합에 필요한 모든 권한이 있는지 확인합니다필수 IAM 권한.
4. (선택 사항) 추가 설정(Additional settings)을 확장합니다. 중복성 활성화(활성 복제본)의 경우 프로덕션 환경에 대해이 옵션을 선택한 상태로 두는 것이 좋습니다. 첫 번째 컬렉션을 생성할 때 OpenSearch Serverless는 두 개의 OCU(하나는 인덱싱용, 다른 하나는 검색용)를 인스턴스화합니다. 또한 고가용성을 보장하기 위해 다른 가용 영역에서 예비 노드 세트를 시작합니다. 개발 및 테스트를 위해 컬렉션에 대한 중복 활성화 설정을 비활성화할 수 있습니다. 그러면 두 개의 대기 복

제본이 제거되고 두 개의 OCU만 인스턴스화됩니다. 기본적으로 중복 활성 복제본이 활성화됩니다. 즉, 계정의 첫 번째 컬렉션에 대해 총 4개의 OCU가 인스턴스화됩니다.

Amazon OpenSearch Serverless 벡터에 대한 고객 관리형 AWS KMS 키 추가에서 고객 관리형 키를 사용하여 벡터 컬렉션의 데이터를 암호화하려면이 옵션을 선택합니다. 기본적으로 OpenSearch는를 사용합니다 AWS 관리형 키.

5. Amazon S3 콘솔에서 OpenSearch로 내보내기 옵션을 클릭하여이 프로세스를 시작한 경우 세부 정보 내보내기 섹션에 OpenSearch가 다음에 수행할 단계가 나열됩니다. 준비가 되면 내보내기를 선택합니다.

   OpenSearch Service 콘솔에서이 프로세스를 시작한 경우 세부 정보 가져오기 섹션에 OpenSearch가 다음에 수행할 단계가 나열됩니다. 준비가 되면 가져오기를 선택합니다.

   OpenSearch는 기록 페이지를 열어 Amazon S3 벡터 인덱스의 모든 내보내기/OpenSearch Serverless 인덱스로 가져오기를 표시합니다.

수집에 성공하면 OSI는 파이프라인을 자동으로 중지하여 OpenSearch에서 내보낸 데이터를 유지하면서 불필요한 비용을 방지합니다. CloudWatch 지표를 통해 통합 진행 상황을 모니터링하고 문제 해결을 위해 세부 로그에 액세스할 수 있습니다.

OpenSearch 컬렉션은 활성 상태로 유지되며 초기 수집이 완료된 후 쿼리에 사용할 수 있습니다. 다음을 수행할 수 있습니다.

- 유사성 검색
- 집계
- 분석 작업

(미리 보기) Amazon S3 벡터 엔진을 사용한 고급 검색 기능

> **Important**
>
> Amazon S3 Vectors와 OpenSearch Service의 통합은 평가판 릴리스이며 변경될 수 있습니다.

Amazon OpenSearch Service는 Amazon S3를 벡터 인덱스의 벡터 엔진으로 사용할 수 있는 기능을 제공합니다. 이 기능을 사용하면 1초 미만의 벡터 검색 기능을 저렴한 비용으로 유지하면서 벡터 데이터를 Amazon S3로 오프로드할 수 있습니다.

이 기능을 사용하면 OpenSearch는 다른 문서 필드를 OpenSearch 클러스터의 스토리지에 유지하면서 벡터 임베딩을 Amazon S3 벡터 인덱스에 저장합니다. 이 아키텍처는 다음과 같은 이점을 제공합니다.

- 내구성: S3 벡터에 기록된 데이터는 11 9초의 데이터 내구성을 위해 설계된 S3에 저장됩니다.
- 확장성: 클러스터 스토리지를 사용하지 않고 대용량 벡터 데이터 세트를 S3로 오프로드합니다.
- 비용 효율성: 벡터가 많은 워크로드의 스토리지 비용을 최적화합니다.

OpenSearch에는 S3 벡터 인덱스를 사용하기 위한 다음과 같은 요구 사항이 있습니다.

- OpenSearch 버전 2.19 이상
- OpenSearch 최적화 인스턴스
- OpenSearch 릴리스의 최신 패치 버전

S3 벡터 활성화

새 도메인을 생성하거나 기존 도메인을 업데이트할 때 고급 기능 섹션에서 엔진으로 S3 벡터 활성화 옵션을 선택할 수 있습니다. 이 설정을 사용하면 S3 벡터를 엔진으로 활용할 때 OpenSearch가 S3 벡터 버킷을 생성할 수 있습니다. 이 옵션을 활성화하면 OpenSearch는 다음을 통해 도메인에 대한 S3 벡터를 구성합니다.

1. 도메인으로 구성된 AWS KMS 키에 대해 두 개의 새 권한 부여 생성:
   - 복호화 권한이 있는 S3 벡터 백그라운드 인덱싱 작업에 대한 권한 부여
   - `GenerateDataKey` 권한이 있는 S3 벡터 버킷을 생성하기 위한 OpenSearch 권한 부여
2. OpenSearch 도메인에서 사용하는 KMS 키를 모든 벡터 인덱스 데이터의 나머지 부분에서 암호화를 위한 CMK로 구성합니다.

S3 벡터 엔진을 사용하여 인덱스 생성

도메인을 구성한 후 인덱스 매핑에서를 백엔드 벡터 엔진`s3vector`으로 사용하여 필드를 사용하여 하나 이상의 k-NN 인덱스를 생성할 수 있습니다. 사용 사례에 따라 다양한 엔진 유형으로 다양한 벡터 필드를 구성할 수 있습니다.

> **Important**
>
> 인덱스 생성 중에 필드 정의를 매핑할 때만 `s3vector` 엔진을 사용할 수 있습니다. 인덱스 생성 후에는 `s3vector` 엔진과의 매핑을 추가하거나 업데이트할 수 없습니다.

다음은 S3 벡터 엔진 인덱스를 생성하는 몇 가지 예입니다.

예: S3 벡터 엔진을 사용하여 k-NN 인덱스 생성

```
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"
        }
    }
  }
}
```

예: S3 벡터와 FAISS 엔진을 모두 사용하여 k-NN 인덱스 생성

이 예제에서는 동일한 인덱스 내에서 여러 벡터 엔진을 사용할 수 있다는 사실을 강조합니다.

```
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
                }
            }
        }
    }
  }
}
```

지원되지 않는 예제: 인덱스 생성 후 S3 벡터 엔진 추가

다음 접근 방식은 지원되지 않으며 실패합니다.

```
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"
        }
    }
}
```

기능 제한 사항

인덱스에서 s3vector 엔진을 사용하기 전에 다음 제한 사항을 고려하세요.

s3vector 엔진에서 지원되지 않는 기능 및 동작

Feature	동작
Split/Shrink/Clone 인덱스	knn_vector 필드의 s3vector 엔진으로 구성된 인덱스와 함께 사용하면 이러한 APIs 실패합니다.
스냅샷	s3vector 엔진을 사용하는 인덱스는 스냅샷을 지원하지 않습니다. 관리형 도메인의 경우: • 자동 스냅샷에는 s3vector 엔진을 사용하지 않는 인덱스만 포함됩니다. • s3vector 인덱스에 대한 수동 스냅샷 요청이 실패합니다.

Feature	동작
	Note 스냅샷은 point-in-time으로 복구할 수 없지만 `s3vector` 엔진은 OpenSearch Optimized 인스턴스와 함께 119개의 내구성을 제공합니다.
UltraWarm 티어	`s3vector` 엔진으로 구성된 인덱스는 UltraWarm 계층으로 마이그레이션할 수 없습니다.
클러스터 간 복제	`s3vector` 엔진으로 구성된 인덱스는 클러스터 간 복제를 지원하지 않습니다.
우발적 삭제 방지	`s3vector` 엔진을 사용하는 인덱스에는 스냅샷이 지원되지 않으므로 실수로 인한 삭제 방지 기능을 사용할 수 없습니다. 도메인에서 다른 인덱스를 복원할 수 있습니다.
방사형 검색	방사형 검색이 있는 쿼리는 `s3vector` 엔진을 사용하는 필드에서 지원되지 않습니다.

문서 인덱싱

S3 벡터 엔진으로 인덱스를 생성한 후 표준 _bulk API를 사용하여 문서를 수집할 수 있습니다. OpenSearch는 `s3vector` 엔진을 사용하여 `knn_vector` 필드의 벡터 데이터를 S3 벡터 인덱스로 실시간으로 자동으로 오프로드합니다. 다른 엔진을 사용하는 다른 필드 또는 `knn_vector` 필드에 속한 데이터는 OpenSearch가 자체 스토리지 계층에서 유지합니다.

승인된 모든 대량 요청에 대해 OpenSearch는 모든 데이터(벡터 및 비벡터)의 내구성을 보장합니다. 요청이 부정 승인을 받는 경우 해당 대량 요청의 문서 내구성에 대한 보장은 없습니다. 이러한 요청을 다시 시도해야 합니다.

대량 인덱싱 예제

```
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 }
```

문서 검색

표준 _search API를 사용하여 인덱스를 검색하여 텍스트, k-NN 또는 하이브리드 쿼리를 실행할 수 있습니다. s3vector 엔진으로 구성된 knn_vector 필드에 대한 쿼리의 경우 OpenSearch는 쿼리를 해당 S3 벡터 인덱스로 자동으로 오프로드합니다.

> **Note**
>
> s3vector 엔진의 경우 새로 고침 시맨틱은 s3vector 엔진을 사용하지 않는 필드에만 적용됩니다. 그러나 로 오프로드된 벡터 데이터는 문서가 성공적으로 s3vector 인덱싱된 직후 표시됩니다.

검색 쿼리 예

```
GET my-first-s3vector-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector_1": {
        "vector": [2.5, 3.5],
        "k": 2
      }
    }
  }
}
```

지원되는 매핑 파라미터

s3vector 엔진의 경우 knn_vector 필드는 매핑에서 다음 파라미터를 지원합니다.

벡터 필드 파라미터

파라미터	필수	설명	지원되는 값
type	예	문서에 있는 필드의 유형입니다.	knn_vector
dimension	예	인덱스에 수집될 각 벡터의 차원입니다.	>0, <=4096
space_type	아니요	벡터 간 거리를 계산하는 데 사용되는 벡터 공간입니다.	l2, cosinesimil
method.engine	예	인덱싱 및 검색에 사용할 대략적인 k-NN 엔진입니다.	s3vector
method.name	아니요	가장 가까운 이웃 방법	""

Important

중첩 knn_vector 필드 유형은 s3vector 엔진을 사용하여 지원되지 않습니다.

측정 및 결제

측정이 발표될 때까지는이 기능에 요금이 청구되지 않습니다.

s3vector 엔진 비활성화

s3vector 엔진을 비활성화하기 전에 현재 사용 중인 모든 인덱스를 삭제합니다. 그렇지 않으면 엔진 비활성화 시도가 실패합니다.

또한 s3vector 엔진을 활성화하거나 비활성화하면 도메인에 블루/그린 배포가 트리거됩니다.

s3vector 엔진을 비활성화하려면 도메인 구성을 편집하고를 설정합니다S3VectorsEngine.Enabled: false.

Amazon OpenSearch Service에서 k-Nearest Neighbor(k-NN) 검색

관련 k-nearest neighbors 알고리즘의 약자인 Amazon OpenSearch Service용 k-NN을 사용하면 벡터 공간에서 포인트를 검색하고 해당 포인트에 대한 "가장 가까운 아웃"을 Euclidean 거리 또는 코사인 유사도로 찾을 수 있습니다. 사용 사례에는 권장 사항(예: 음악 애플리케이션의 "좋아하는 다른 노래" 기능), 이미지 인식 및 사기 탐지가 포함됩니다.

> **Note**
>
> 이 설명서에서는 k-NN 플러그인에 대한 간략한 개요와 관리형 OpenSearch Service에서 플러그인을 사용할 때의 제한 사항을 제공합니다. 간단하고 복잡한 예제, 파라미터 참조 및 전체 API 참조를 포함하여 k-NN 플러그인에 대한 포괄적인 설명서는 오픈 소스 OpenSearch 설명서를 참조하세요. 오픈 소스 설명서에서는 성능 조정 및 k-NN별 클러스터 설정도 다룹니다.

k-NN 시작하기

k-NN을 사용하려면 `index.knn` 설정으로 인덱스를 만들고 `knn_vector` 데이터 유형의 필드를 하나 이상 추가해야 합니다.

```
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 데이터 유형은 필수 dimension 파라미터로 정의된 부동 소수점 수를 사용하여 최대 10,000개의 부동 소수점으로 구성된 단일 목록을 지원합니다. 인덱스를 생성한 후 일부 데이터를 추가합니다.

```
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 }
```

그런 다음 knn 쿼리 유형을 사용하여 데이터를 검색할 수 있습니다.

```
GET my-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  }
}
```

이 경우 k는 쿼리를 반환하려는 이웃 수입니다. 하지만 size 옵션도 포함해야 합니다. 그렇지 않으면 전체 쿼리에 대한 k 결과가 아닌 각 샤드(및 각 세그먼트)에 대한 k 결과를 얻습니다. k-NN은 최대 k 값인 10,000을 지원합니다.

knn 쿼리를 다른 절과 혼합하면 k 결과보다 적게 수신할 수 있습니다. 이 예제에서 post_filter 절은 결과 수를 2에서 1로 줄입니다.

```
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
      }
    }
  }
}
```

최적의 성능을 유지하면서 대량의 쿼리를 처리해야 하는 경우 _msearch API를 사용하여 JSON으로 대량 검색을 구성하고 단일 요청을 전송하여 여러 검색을 수행할 수 있습니다.

```
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 }} } }
```

다음 동영상은 K-NN 쿼리에 대한 대량 벡터 검색을 설정하는 방법을 보여줍니다.

k-NN의 차이점, 조정, 제한 사항

OpenSearch를 사용하면 `_cluster/settings` API를 사용해 모든 k-NN 설정을 수정합니다. OpenSearch Service에서는 `knn.memory.circuit_breaker.enabled` 및 `knn.circuit_breaker.triggered`를 제외한 모든 설정을 변경할 수 있습니다. k-NN 통계량은 Amazon CloudWatch 지표로 포함됩니다.

특히, 각 데이터 노드의 `KNNGraphMemoryUsage` 지표를 `knn.memory.circuit_breaker.limit` 통계 및 해당 인스턴스 유형에 사용 가능한 RAM과 비교해 확인합니다. OpenSearch Service는 Java 힙에 인스턴스 RAM의 절반을 사용합니다(최대 힙 크기 32GiB). 기본적으로 k-NN은 나머지 절반의 최대 50%를 사용하므로 RAM이 32GiB인 인스턴스 유형에서는 그래프 8GiB(32 * 0.5 * 0.5)를 수용할 수 있습니다. 그래프 메모리 사용량이 이 값을 초과하면 성능이 저하될 수 있습니다.

버전 2.x 이상에서 생성된 k-NN 인덱스를 버전 2.17 이상의 도메인에서 UltraWarm 또는 콜드 스토리지로 마이그레이션할 수 있습니다.

k-NN 인덱스에 대한 캐시 지우기 API 및 워밍업 API는 원 인덱스에 대해 차단됩니다. 인덱스에 대한 첫 번째 쿼리가 시작되면 Amazon S3에서 그래프 파일을 다운로드하고 그래프를 메모리에 로드합니다. 마찬가지로 그래프의 TTL이 만료되면 파일이 메모리에서 자동으로 제거됩니다.

Amazon OpenSearch Service와 함께 OpenSearch Dashboards 사용

OpenSearch Dashboards는 OpenSearch와 함께 작동하도록 제작된 오픈 소스 시각화 도구입니다. Amazon OpenSearch Service는 모든 OpenSearch Service 도메인에 Dashboards 설치를 제공합니다. 대시보드는 도메인의 핫 데이터 노드에서 실행됩니다.

OpenSearch Dashboards는 단일 OpenSearch 도메인 내의 데이터를 탐색하고 분석하기 위한 시각화 도구입니다. 반면 중앙 집중식 OpenSearch 사용자 인터페이스(OpenSearch 애플리케이션이라고도 함)는 여러 OpenSearch 도메인, OpenSearch Serverless 컬렉션 및 AWS 데이터 소스에 연결하는 클라우드 기반 UI입니다. OpenSearch 여기에는 관찰성 및 보안 분석과 같은 특정 사용 사례를 위한 워크스페이스가 포함되며 데이터 세트 전반에서 통합된 환경을 제공합니다. 대시보드는 개별 도메인에 연결되어 있지만 중앙 집중식 UI를 사용하면 도메인 간 데이터 통합 및 분석이 가능합니다. 자세한 내용은 OpenSearch UI 단원을 참조하십시오.

OpenSearch Service 콘솔의 도메인 대시보드에서 OpenSearch 대시보드 링크를 찾을 수 있습니다. OpenSearch 실행하는 도메인의 경우 URL은 *domain-endpoint*/_dashboards/입니다. 레거시 Elasticsearch를 실행하는 도메인의 경우 URL은 *domain-endpoint*/_plugin/kibana입니다.

이 기본 대시보드 설치를 사용하는 쿼리의 제한 시간은 300초입니다.

> **Note**
>
> 이 설명서에서는 다양한 연결 방법을 포함하여 Amazon OpenSearch Service의 컨텍스트에서 OpenSearch 대시보드를 설명합니다. 시작하기 가이드, 대시보드 생성 지침, 대시보드 관리 및 대시보드 쿼리 언어(DQL)를 포함한 포괄적인 설명서는 오픈 소스 OpenSearch 설명서의 OpenSearch Dashboards를 참조하세요.

대시보드에 대한 액세스 제어

Dashboards는 IAM 사용자와 역할을 기본으로 지원하지는 않지만, OpenSearch Service가 Dashboards 액세스 제어를 위한 다양한 솔루션을 제공합니다.

- Dashboards에 대한 SAML 인증을 활성화합니다.
- HTTP 기본 인증과 함께 세분화된 액세스 제어를 사용합니다.
- Dashboards에 대한 Cognito 인증을 구성합니다.

- 공용 액세스 도메인의 경우 프록시 서버를 사용하거나 사용하지 않는 IP 기반 액세스 정책을 구성합니다.
- VPC 액세스 도메인의 경우 프록시 서버를 사용하거나 사용하지 않는 오픈 액세스 정책 및 보안 그룹을 이용해 액세스를 제어합니다. 자세한 내용은 the section called “VPC 도메인 액세스 정책에 대하여” 섹션을 참조하세요.

프록시를 사용하여 Dashboards에서 OpenSearch Service에 액세스

> **Note**
>
> 이 프로세스는 도메인이 퍼블릭 액세스를 사용하며 Cognito 인증을 사용하지 않으려는 경우에만 적용됩니다. the section called “대시보드에 대한 액세스 제어 ” 섹션을 참조하세요.

Kibana는 JavaScript 애플리케이션이므로 요청은 사용자의 IP 주소에서 발생합니다. 각 사용자에게 대시보드 액세스 권한을 주기 위해 허용해야 하는 IP 주소 수가 늘어나기 때문에, IP 기반 액세스 제어는 실용적이지 않을 수 있습니다. 한 가지 해결 방법은 Dashboards와 OpenSearch Service 사이에 프록시 서버를 배치하는 것입니다. 그런 다음 하나의 IP 주소인 프록시의 요청만 허용하는 IP 기반 액세스 정책을 추가할 수 있습니다. 다음 다이어그램은 이 구성을 보여줍니다.

1. 이것이 OpenSearch Service 도메인입니다. IAM은 이 도메인에 대해 인증된 액세스 권한을 제공합니다. 추가 IP 기반 액세스 정책은 프록시 서버에 대한 액세스를 제공합니다.
2. 이 프록시 서버는 Amazon EC2 인스턴스에서 실행됩니다.
3. 다른 애플리케이션은 서명 버전 4 서명 프로세스를 사용하여 인증 요청을 OpenSearch Service로 보낼 수 있습니다.
4. 대시보드 클라이언트는 프록시를 통해 OpenSearch Service 도메인에 연결합니다.

이러한 유형의 구성을 활성화하려면 역할 및 IP 주소를 지정하는 리소스 기반 정책이 필요합니다. 다음은 샘플 정책입니다.

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/*"
      }
   ]
}
```

탄력적 IP 주소를 사용하여 프록시 서버를 실행하는 EC2 인스턴스를 구성하는 것이 좋습니다. 이러한 방식으로 필요한 경우 인스턴스를 대체하고 동일한 퍼블릭 IP 주소를 계속 연결할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 탄력적 IP 주소를 참조하세요.

프록시 서버 및 Cognito 인증을 사용한다면, Dashboards와 Amazon Cognito용 설정을 추가해 `redirect_mismatch` 오류를 방지해야 할 수도 있습니다. 다음 `nginx.conf` 예를 참조하세요.

```
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**
>
> (선택 사항) 전용 조정자 노드를 프로비저닝하려는 경우 자동으로 OpenSearch 대시보드 호스팅이 시작됩니다. 따라서 CPU 및 메모리와 같은 데이터 노드 리소스의 가용성이 증가합니다.

이렇게 증가된 데이터 노드 리소스 가용성은 도메인의 전반적인 복원력을 개선하는 데 도움이 될 수 있습니다.

WMS 맵 서버를 사용하도록 대시보드 구성

Dashboards for OpenSearch Service의 기본 설치에는 인도 및 중국 리전의 도메인을 제외한 맵 서비스가 포함됩니다. 맵 서비스는 최대 10개의 줌 레벨을 지원합니다.

리전과 관계없이 좌표 맵 시각화를 위해 다른 WMS(Web Map Service) 서버를 사용하도록 대시보드를 구성할 수 있습니다. 리전 맵 시각화는 기본 맵 서비스만 지원합니다.

WMS 맵 서버를 사용하도록 Dashboards를 구성하려면:

1. Dashboards를 엽니다.
2. 스택 관리(Stack Management)를 선택합니다.
3. 고급 설정(Advanced Settings)을 선택합니다.
4. visualization:tileMap:WMSdefaults를 찾습니다.
5. `enabled`를 `true`로 바꾸고 `url`은 유효한 WMS 맵 서버의 URL로 변경합니다.

```
{
  "enabled": true,
  "url": "wms-server-url",
  "options": {
    "format": "image/png",
    "transparent": true
  }
}
```

6. 변경 사항 저장(Save changes)을 선택합니다.

새 기본값을 시각화에 적용하려면 Dashboards를 다시 로드해야 할 수 있습니다. 시각화를 저장한 경우 시각화를 연 후 옵션(Options)을 선택합니다. WMS 맵 서버가 활성화되어 있고 WMS url에 기본 설정 맵 서버가 포함되어 있는지 확인한 다음 변경 사항 적용을 선택합니다.

> **Note**
>
> 맵 서비스에는 종종 라이선스 요금이 부과되거나 제한이 따릅니다. 어떤 맵 서버를 지정하든 간에 그러한 부분은 모두 사용자의 책임입니다. 미국 지질조사국의 맵 서비스로 테스트해 보면 유용합니다.

로컬 Dashboards 서버를 OpenSearch Service에 연결

이미 자체 대시보드 인스턴스를 구성하는 데 상당한 시간을 투자한 경우 OpenSearch Service에서 제공하는 기본 대시보드 인스턴스 대신(또는 이에 추가하여) 사용할 수 있습니다. 다음 절차는 오픈 액세스 정책과 함께 세분화된 액세스 제어를 사용하는 도메인에 적용됩니다.

로컬 Dashboards 서버를 OpenSearch Service에 연결하려면

1. OpenSearch Service 도메인에서 적절한 권한을 가진 사용자를 생성합니다.
   a. Dashboards에서 보안(Security),내부 사용자(Internal users)로 이동하여 내부 사용자 생성(Create internal user)을 선택합니다.
   b. 사용자 이름과 암호를 입력하고 생성(Create)을 선택합니다.
   c. 역할(Roles)로 이동하여 역할을 선택합니다.
   d. 매핑된 사용자(Mapped users)를 선택하고 매핑 관리(Manage mapping)를 선택합니다.
   e. 사용자(Users)에서 사용자 이름을 추가하고 맵(Map)을 선택합니다.
2. 자체 관리형 Dashboards OSS 설치에 적절한 버전의 OpenSearch 보안 플러그인을 다운로드하고 설치합니다.
3. 로컬 Dashboards 서버에서 `config/opensearch_dashboards.yml` 파일을 열고 이전에 만든 사용자 이름과 암호를 사용하여 OpenSearch Service 엔드포인트를 추가합니다.

```
opensearch.hosts: ['https://domain-endpoint']
opensearch.username: 'username'
opensearch.password: 'password'
```

다음 샘플 `opensearch_dashboards.yml` 파일을 사용할 수 있습니다.

```
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]
```

OpenSearch Service 인덱스를 보려면 로컬 대시보드 서버를 시작하고 Dev Tools로 이동하여 다음 명령을 실행합니다.

```
GET _cat/indices
```

대시보드에서 인덱스 관리

OpenSearch Service 도메인에 대시보드를 설치하면 도메인의 여러 스토리지 계층에서 인덱스를 관리하는 데 유용한 UI가 제공됩니다. 인덱스 상태 관리(ISM) 정책에 따라 관리되는 인덱스뿐만 아니라 핫, UltraWarm, 콜드 스토리지의 모든 인덱스를 보려면 Dashboards 주 메뉴에서 인덱스 관리(Index Management)를 선택합니다. 인덱스 관리를 사용하여 웜 스토리지와 콜드 스토리지 간에 인덱스를 이동하고 세 계층 간의 마이그레이션을 모니터링합니다.

UltraWarm 및/또는 콜드 스토리지를 활성화하지 않으면 핫, 웜 및 콜드 인덱스 옵션이 표시되지 않습니다.

기타 기능

각 OpenSearch Service 도메인의 기본 대시보드 설치에는 몇 가지 추가 기능이 있습니다.

- 다양한 OpenSearch 플러그인을 위한 사용자 인터페이스
- 테넌트
- 보고서

  보고(Reporting) 메뉴를 사용하여 검색 페이지에서 온디맨드 CSV 보고서 및 대시보드 또는 시각화의 PDF 또는 PNG 보고서를 생성할 수 있습니다. CSV 보고서의 행은 10,000개로 제한됩니다.
- Gantt 차트
- 노트북

Amazon OpenSearch Service에서 OpenSearch UI 사용

OpenSearch UI(사용자 인터페이스)는 Amazon OpenSearch Service를 위한 현대화된 운영 분석 환경으로, 여러 소스의 데이터와 상호 작용할 수 있는 통합 보기를 제공합니다. 호스팅하는 하나의 도메인 또는 컬렉션에서만 작동하는 OpenSearch Dashboards와 달리 OpenSearch UI는에서 호스팅됩니다 AWS 클라우드. 이를 통해 OpenSearch UI는 클러스터 업그레이드 중에 고가용성을 달성하고 기능을 유지하고 기본적으로 여러 데이터 소스와 연결할 수 있습니다. OpenSearch 대시보드에 대한 자세한 내용은 섹션을 참조하세요 OpenSearch 대시보드.

다음은 OpenSearch UI의 주요 기능입니다.

- 다중 데이터 소스 지원 - OpenSearch UI는 여러 데이터 소스와 연결하여 포괄적인 보기를 생성할 수 있습니다. 여기에는 OpenSearch 도메인 및 서버리스 컬렉션과 Amazon CloudWatch, Amazon Security Lake, Amazon Simple Storage Service(Amazon S3)와 같은 통합 AWS 데이터 소스가 포함됩니다.
- 업그레이드 중 가동 중지 시간 없음 - OpenSearch UI는에서 호스팅됩니다 AWS 클라우드. 즉, OpenSearch는 계속 작동하며 업그레이드 프로세스 중에 클러스터에서 데이터를 검색할 수 있습니다.
- Workspaces - 관찰성, 보안 분석 및 검색과 같은 다양한 워크플로에 대한 팀 협업을 위해 큐레이션된 공간입니다. 개인 정보 설정을 정의하고 워크스페이스에서 공동 작업자에 대한 권한을 관리할 수 있습니다.
- Single Sign-On - OpenSearch UI는 AWS IAM Identity Center 및 AWS Identity and Access Management (IAM) 페더레이션을 통한 SAML과 함께 작동하여 자격 증명 공급자와 통합하여 최종 사용자를 위한 Single Sign-On 환경을 생성합니다.
- GenAI 기반 분석 - OpenSearch UI는 자연어 쿼리 생성을 지원하여 분석에 적합한 쿼리를 생성하는 데 도움이 됩니다. 또한 OpenSearch UI는 Amazon Q Developer와 협력하여 Amazon Q 채팅을 제공하고 시각화, 알림 요약, 인사이트 및 권장 이상 탐지기를 생성하는 데 도움이 됩니다.
- 다중 쿼리 언어 지원 - OpenSearch UI는 파이프 처리 언어(PPL), SQL, Lucene 및 대시보드 쿼리 언어(DQL)를 지원합니다.
- 교차 리전 및 교차 계정 지원 - OpenSearch UI는 클러스터 간 검색 기능을 활용하여 집계된 분석 및 시각화를 위해 다른 계정 및 다른 리전의 OpenSearch 도메인에 연결할 수 있습니다.

첫 번째 OpenSearch UI를 시작하고 생성하려면의 지침을 따르세요the section called “시작”.

OpenSearch UI용으로 릴리스된 최신 기능에 대한 자세한 내용은 섹션을 참조하세요the section called “릴리스 이력”.

주제

- Amazon OpenSearch Service 사용자 인터페이스 릴리스 기록
- Amazon OpenSearch Service에서 OpenSearch 사용자 인터페이스 시작하기
- 를 사용하여 SAML 페더레이션 활성화 AWS Identity and Access Management
- 데이터 소스 연결 및 Virtual Private Cloud 액세스 권한 관리
- Amazon OpenSearch Service 워크스페이스 사용
- 클러스터 간 검색을 통한 리전 간 및 계정 간 데이터 액세스
- VPC 엔드포인트에서 OpenSearch UI에 대한 액세스 관리
- OpenSearch UI 엔드포인트 및 할당량

Amazon OpenSearch Service 사용자 인터페이스 릴리스 기록

다음 표에는 OpenSearch UI에 대한 Amazon OpenSearch Service 지원의 모든 릴리스와 각 릴리스에 포함된 기능 및 개선 사항이 나열되어 있습니다.

변경 사항	릴리스 날짜	설명
새로운 특성	2025-04-16	이제 OpenSearch UI가 클러스터 간 검색과 함께 작동합니다. 이를 통해 OpenSearch UI를 하나의 AWS 리전 에서 사용하여 다른 리전의 클러스터에 액세스할 수 있습니다. 이는 동일한 리전의 클러스터와 연결된 원격 클러스터로 구성하여 수행됩니다. 자세한 내용은 the section called “클러스터 간 검색을 통한 리전 간 및 계정 간 데이터 액세스” 단원을 참조하십시오.
새로운 특성	2025-03-31	이제 Amazon OpenSearch Service에서 Amazon Q Developer를 정식으로 사용할 수 있습니다. 자세한 내용은 Amazon Q 지원 단원을 참조하세요.
새로운 특성	2025-02-05	Security Assertion Markup Language 2.0)(SAML)~ AWS Identity and Access Management (IAM) 페더레이션은 이제 OpenSearch UI에서 작동합니다. 이를 통해 최종 사용자를 위한 자격 증명 공급자 시작 SSO(Single Sign-On) 환경을 생성할 수 있습니다. 자세한 내용은 the section called “IAM을 사용하여 SAML 페더레이션 활성화” 단원을 참조하십시오.

변경 사항	릴리스 날짜	설명
새로운 통합	2024-12-01	Amazon CloudWatch와의 제로 ETL 통합은 로그 데이터 분석 및 시각화를 간소화하여 기술 오버헤드와 운영 비용을 절감합니다. 자세한 내용은 AWS 뉴스 블로그의 New Amazon CloudWatch and Amazon OpenSearch Service launch an integrated analytics experience를 참조하세요.
새로운 통합	2024-12-01	Amazon Security Lake와의 제로 ETL 통합을 통해 조직은 보안 데이터에서 실행 가능한 인사이트를 효율적으로 검색, 분석 및 얻을 수 있습니다. 자세한 내용은 AWS 뉴스 블로그의 보안 분석을 간소화하기 위한 Amazon OpenSearch Service 및 Amazon Security Lake 통합 소개를 참조하세요.
초기 릴리스	2024-11-07	OpenSearch UI의 최초 공개 릴리스입니다. 자세한 내용은 AWS 빅 데이터 블로그의 Amazon OpenSearch Service에서 차세대 OpenSearch UI 시작을 참조하세요.

Amazon OpenSearch Service에서 OpenSearch 사용자 인터페이스 시작하기

Amazon OpenSearch Service에서 애플리케이션은 OpenSearch 사용자 인터페이스(OpenSearch UI)의 인스턴스입니다. 각 애플리케이션을 여러 데이터 소스와 연결할 수 있으며 단일 소스를 여러 애플리케이션과 연결할 수 있습니다. 지원되는 다양한 인증 옵션을 사용하여 다양한 관리자를 위한 여러 애플리케이션을 생성할 수 있습니다.

이 주제의 정보를 사용하여 AWS Management Console 또는를 사용하여 OpenSearch UI 애플리케이션을 생성하는 프로세스를 안내합니다 AWS CLI.

주제

- Amazon OpenSearch Service 애플리케이션을 생성하는 데 필요한 권한
- OpenSearch UI 애플리케이션 생성
- 애플리케이션 관리자 관리

Amazon OpenSearch Service 애플리케이션을 생성하는 데 필요한 권한

애플리케이션을 생성하기 전에 작업에 필요한 권한이 부여되었는지 확인합니다. 필요한 경우 계정 관리자에게 문의하여 도움을 받으세요.

일반 권한

OpenSearch Service에서 애플리케이션을 사용하려면 다음 정책에 표시된 권한이 필요합니다. 권한은 다음과 같은 목적으로 사용됩니다.

- 애플리케이션을 생성하고 관리하려면 5가지 `es:*Application` 권한이 필요합니다.
- 애플리케이션에서 태그를 추가, 나열 및 제거하려면 세 가지 `es:*Tags` 권한이 필요합니다.
- 데이터 소스를 연결하려면 `aoss:BatchGetCollection`, `es:DescribeDomain` 및 `es:GetDirectQueryDataSource` 권한이 필요합니다.
- 데이터 소스에 액세스하려면 `aoss:APIAccessAlles:ESHttp*`, 및 4 `opensearch:*DirectQuery*` 권한이 필요합니다.
- 는 Amazon OpenSearch Service에 계정에서 서비스 연결 역할(SLR)을 생성할 수 있는 권한을 `iam:CreateServiceLinkedRole` 제공합니다. 이 역할을 사용하면 OpenSearch UI 애플리케이션이 계정에 Amazon CloudWatch 지표를 게시할 수 있습니다. 자세한 내용은 the section called “권한” 주제에서 서비스 연결 역할을 사용하여 VPC 도메인 및 직접 쿼리 데이터 소스 생성 섹션을 참조하세요.

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"
        }
    ]
}
```

IAM Identity Center 인증을 사용하는 애플리케이션을 생성할 수 있는 권한(선택 사항)

기본적으로 대시보드 애플리케이션은 AWS Identity and Access Management (IAM)을 사용하여 인증되어 AWS 리소스 사용자의 권한을 관리합니다. 그러나 기존 ID 제공업체를 사용하여 OpenSearch UI 애플리케이션에 로그인할 수 있는 IAM Identity Center를 사용하여 Single Sign-On 환경을 제공하도록 선택할 수 있습니다. 이 경우이 주제의 뒷부분에 나오는 절차에서 IAM Identity Center를 사용한 인증 옵션을 선택한 다음 IAM Identity Center 사용자에게 OpenSearch UI 애플리케이션에 액세스하는 데 필요한 권한을 부여합니다.)

IAM Identity Center 인증을 사용하는 애플리케이션을 생성하려면 다음 권한이 필요합니다. ## ###를 자신의 정보로 바꿉니다. 필요한 경우 계정 관리자에게 문의하여 도움을 받으세요.

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"
        }
    ]
}
```

OpenSearch UI 애플리케이션 생성

다음 절차 중 하나를 사용하여 및 애플리케이션 이름, 인증 방법 및 관리자를 지정하는 애플리케이션을 생성합니다.

주제

- 콘솔에서 IAM 인증을 사용하는 OpenSearch UI 애플리케이션 생성
- 콘솔에서 AWS IAM Identity Center 인증을 사용하는 OpenSearch UI 애플리케이션 생성
- 를 사용하여 AWS IAM Identity Center 인증을 사용하는 OpenSearch UI 애플리케이션 생성 AWS CLI

콘솔에서 IAM 인증을 사용하는 OpenSearch UI 애플리케이션 생성

콘솔에서 IAM 인증을 사용하는 OpenSearch UI 애플리케이션을 생성하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 OpenSearch UI(대시보드)를 선택합니다.
3. 애플리케이션 생성을 선택합니다.
4. 애플리케이션 이름에 애플리케이션 이름을 입력합니다.

5. IAM Identity Center를 사용한 인증 확인란을 선택하지 마십시오. 를 통한 인증으로 애플리케이션을 생성하는 방법에 대한 자세한 내용은이 주제 the section called “콘솔에서 AWS IAM Identity Center 인증을 사용하는 OpenSearch UI 애플리케이션 생성”뒷부분의 섹션을 AWS IAM Identity Center참조하세요.
6. (선택 사항) 생성 중인 애플리케이션의 관리자로 자동으로 추가됩니다. OpenSearch 애플리케이션 관리자 관리 영역에서 다른 사용자에게 관리자 권한을 부여할 수 있습니다.

   > **Note**
   >
   > OpenSearch UI 애플리케이션 관리자 역할은 OpenSearch UI 애플리케이션을 편집하고 삭제할 수 있는 권한을 부여합니다. 애플리케이션 관리자는 OpenSearch UI 애플리케이션에서 워크스페이스를 생성, 편집 및 삭제할 수도 있습니다.

   다른 사용자에게 관리자 권한을 부여하려면 다음 중 하나를 선택합니다.

   - 특정 사용자(들)에게 관리자의 권한 부여 - OpenSearch 애플리케이션 관리자 필드의 속성 팝업 목록에서 IAM 사용자 또는

     AWS IAM Identity Center 사용자를 선택한 다음 관리자 권한을 부여할 개별 사용자를 선택합니다.
   - 모든 사용자에게 관리자 권한 부여 - 조직 또는 계정의 모든 사용자에게 관리자 권한이 부여됩니다.
7. (선택 사항) 태그 영역에서 애플리케이션에 하나 이상의 태그 키 이름/값 페어를 적용합니다.

   태그는 리소스에 할당하는 선택적 메타데이터입니다. 태그를 사용하면 용도, 소유자 또는 환경을 기준으로 하는 등 리소스를 다양한 방식으로 분류할 수 있습니다.
8. 생성(Create)을 선택합니다.

콘솔에서 AWS IAM Identity Center 인증을 사용하는 OpenSearch UI 애플리케이션 생성

인증을 사용하는 AWS IAM Identity Center OpenSearch UI 애플리케이션을 생성하려면의이 주제 앞부분에서 설명한 IAM 권한이 있어야 합니다the section called “IAM Identity Center 인증을 사용하는 애플리케이션을 생성할 수 있는 권한(선택 사항)”.

콘솔에서 인증을 사용하는 AWS IAM Identity Center OpenSearch UI 애플리케이션을 생성하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 OpenSearch UI(대시보드)를 선택합니다.
3. 애플리케이션 생성을 선택합니다.
4. 애플리케이션 이름에 애플리케이션 이름을 입력합니다.
5. (선택 사항) 조직 또는 계정에 대해 Single Sign-On을 활성화하려면 다음을 수행합니다.

   a. 다음 이미지와 같이 IAM Identity Center를 사용한 인증 확인란을 선택합니다.

   b. 다음 중 하나를 수행합니다.

   - Identity Center용 IAM 역할 애플리케이션 목록에서 IAM Identity Center가 OpenSearch UI 및 관련 데이터 소스에 액세스하는 데 필요한 권한을 제공하는 기존 IAM 역할을 선택합니다. 역할이 보유해야 하는 권한은 다음 글머리표의 정책을 참조하세요.
   - 필요한 권한이 있는 새 역할을 생성합니다. IAM 사용 설명서의 다음 절차를 지정된 옵션과 함께 사용하여 새 역할을 생성하고 필요한 권한 정책 및 신뢰 정책을 사용합니다.
     - 절차: IAM 정책 생성(콘솔)

       이 절차의 단계를 따르면 다음 정책을 정책 편집기 JSON 필드에 붙여넣습니다.

       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": "*"
        }
    ]
}
```

- 절차: 사용자 지정 신뢰 정책을 사용하여 역할 생성

  이 절차의 단계를 따르면 사용자 지정 신뢰 정책 상자의 자리 표시자 JSON을 다음과 같이 바꿉니다.

  > **Tip**
  >
  > 기존 역할에 신뢰 정책을 추가하는 경우 역할의 신뢰 관계 탭에서 정책을 추가합니다.

  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. 조직 또는 계정에 IAM Identity Center 인스턴스가 이미 생성된 경우 콘솔은 다음 이미지와 같이 Amazon OpenSearch Dashboards가 IAM Identity Center의 조직 인스턴스에 이미 연결되어 있음을 보고합니다.

조직 또는 계정에서 IAM Identity Center를 아직 사용할 수 없는 경우 필요한 권한이 있는 사용자 또는 관리자가 조직 인스턴스 또는 계정 인스턴스를 생성할 수 있습니다. Amazon OpenSearch Dashboards를 IAM Identity Center에 연결 영역은 다음 이미지와 같이 둘 다에 대한 옵션을 제공합니다.

이 경우 테스트를 위해 IAM Identity Center에서 계정 인스턴스를 생성하거나 관리자에게 IAM Identity Center에서 조직 인스턴스를 생성하도록 요청할 수 있습니다. 자세한 내용은AWS IAM Identity Center 사용 설명서에서 다음 주제를 참조하세요.

> **Note**
>
> 현재 OpenSearch UI 애플리케이션은 IAM Identity Center 조직 인스턴스 AWS 리전와 동일한 에서만 생성할 수 있습니다. 애플리케이션을 생성한 후 해당 리전의 데이터

소스에 액세스하는 방법에 대한 자세한 내용은 섹션을 참조하세요the section called "클러스터 간 검색을 통한 리전 간 및 계정 간 데이터 액세스".

- IAM Identity Center의 조직 인스턴스
- IAM Identity Center의 계정 인스턴스
- 활성화 AWS IAM Identity Center

6. (선택 사항) 생성 중인 애플리케이션의 관리자로 자동으로 추가됩니다. OpenSearch 애플리케이션 관리자 관리 영역에서 다음 이미지와 같이 다른 사용자에게 관리자 권한을 부여할 수 있습니다.

> **Note**
>
> OpenSearch UI 애플리케이션 관리자 역할은 OpenSearch UI 애플리케이션을 편집하고 삭제할 수 있는 권한을 부여합니다. 애플리케이션 관리자는 OpenSearch UI 애플리케이션에서 워크스페이스를 생성, 편집 및 삭제할 수도 있습니다.

다른 사용자에게 관리자 권한을 부여하려면 다음 중 하나를 선택합니다.

- 특정 사용자(들)에게 관리자 권한 부여 - OpenSearch 애플리케이션 관리자 필드의 속성 팝업 목록에서 IAM 사용자 또는

  AWS IAM Identity Center 사용자를 선택한 다음 관리자 권한을 부여할 개별 사용자를 선택합니다.
- 모든 사용자에게 관리자 권한 부여 - 조직 또는 계정의 모든 사용자에게 관리자 권한이 부여됩니다.

7. (선택 사항) 태그 영역에서 애플리케이션에 하나 이상의 태그 키 이름/값 페어를 적용합니다.

   태그는 리소스에 할당하는 선택적 메타데이터입니다. 태그를 사용하면 용도, 소유자 또는 환경을 기준으로 하는 등 리소스를 다양한 방식으로 분류할 수 있습니다.

8. 생성(Create)을 선택합니다.

를 사용하여 AWS IAM Identity Center 인증을 사용하는 OpenSearch UI 애플리케이션 생성 AWS CLI

를 사용하여 AWS IAM Identity Center 인증을 사용하는 OpenSearch UI 애플리케이션을 생성하려면 다음 옵션과 함께 create-application 명령을 AWS CLI사용합니다.

- `--name` - 애플리케이션의 이름입니다.
- `--iam-identity-center-options` – (선택 사항) OpenSearch가 인증 및 액세스 제어에 사용할 IAM Identity Center 인스턴스 및 IAM 역할입니다.

*## ###*를 자신의 정보로 바꿉니다.

```
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\"
          }
    "
```

애플리케이션 관리자 관리

OpenSearch UI 애플리케이션 관리자는 OpenSearch UI 애플리케이션을 편집하고 삭제할 수 있는 권한이 있는 정의된 역할입니다.

기본적으로 OpenSearch UI 애플리케이션의 생성자는 OpenSearch UI 애플리케이션의 첫 번째 관리자입니다.

콘솔을 사용하여 OpenSearch UI 관리자 관리

애플리케이션 생성 워크플로 중에 AWS Management Console또는 애플리케이션이 생성된 후 편집 페이지에서의 OpenSearch UI 애플리케이션에 관리자를 추가할 수 있습니다.

OpenSearch UI 애플리케이션 관리자 역할은 OpenSearch UI 애플리케이션을 편집하고 삭제할 수 있는 권한을 부여합니다. 애플리케이션 관리자는 OpenSearch UI 애플리케이션에서 워크스페이스를 생성, 편집 및 삭제할 수도 있습니다.

애플리케이션 세부 정보 페이지에서 IAM 보안 주체의 Amazon 리소스 이름(ARN)을 검색하거나 IAM Identity Center 사용자의 이름을 검색할 수 있습니다.

콘솔을 사용하여 OpenSearch UI 관리자를 관리하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 OpenSearch UI(대시보드)를 선택합니다.
3. OpenSearch 애플리케이션 영역에서 기존 애플리케이션의 이름을 선택합니다.
4. 편집을 선택합니다.
5. 다른 사용자에게 관리자 권한을 부여하려면 다음 중 하나를 선택합니다.
   - 특정 사용자(들)에게 관리자의 권한 부여 - OpenSearch 애플리케이션 관리자 필드의 속성 팝업 목록에서 IAM 사용자 또는

     AWS IAM Identity Center 사용자를 선택한 다음 관리자 권한을 부여할 개별 사용자를 선택합니다.
   - 모든 사용자에게 관리자 권한 부여 - 조직 또는 계정의 모든 사용자에게 관리자 권한이 부여됩니다.
6. 업데이트를 선택합니다.

추가 관리자를 제거할 수 있지만 각 OpenSearch UI 애플리케이션은 한 명 이상의 관리자를 유지해야 합니다.

를 사용하여 OpenSearch UI 관리자 관리 AWS CLI

를 사용하여 OpenSearch UI 애플리케이션 관리자를 생성하고 업데이트할 수 있습니다 AWS CLI.

를 사용하여 OpenSearch UI 관리자 생성 AWS CLI

다음은 OpenSearch UI 애플리케이션을 생성할 때 IAM 보안 주체 및 IAM Identity Center 사용자를 관리자로 추가하는 예제입니다.

예제 1: IAM 사용자를 관리자로 추가하는 OpenSearch UI 애플리케이션 생성

다음 명령을 실행하여 IAM 사용자를 관리자로 추가하는 OpenSearch UI 애플리케이션을 생성합니다. *## ###*를 자신의 정보로 바꿉니다.

```
aws opensearch create-application \
    --name application-name \
```

```
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"arn:aws:iam::account-id:user/user-id\"
        }
    "
```

예제 2: IAM Identity Center를 활성화하고 IAM Identity Center 사용자 ID를 OpenSearch UI 애플리케이션 관리자로 추가하는 OpenSearch UI 애플리케이션 생성

다음 명령을 실행하여 IAM Identity Center를 활성화하고 IAM Identity Center 사용자 ID를 OpenSearch UI 애플리케이션 관리자로 추가하는 OpenSearch UI 애플리케이션을 생성합니다. *## ###*를 자신의 정보로 바꿉니다.

key는 OpenSearch UI 애플리케이션의 관리자 역할과 같이 설정할 구성 항목을 지정합니다. 유효한 값에는 opensearchDashboards.dashboardAdmin.users 및 opensearchDashboards.dashboardAdmin.groups(이)가 있습니다.

*xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*는 IAM 사용자의 Amazon 리소스 이름(ARN)과 같이 키에 할당된 값을 나타냅니다.

```
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\"
        }
    "
```

를 사용하여 OpenSearch UI 관리자 업데이트 AWS CLI

다음은 기존 OpenSearch 애플리케이션의 관리자로 할당된 IAM 보안 주체 및 IAM Identity Center 사용자를 업데이트하는 예입니다.

예제 1: IAM 사용자를 기존 OpenSearch 애플리케이션의 관리자로 추가

다음 명령을 실행하여 OpenSearch UI 애플리케이션을 업데이트하여 IAM 사용자를 관리자로 추가합니다. *## ###*를 자신의 정보로 바꿉니다.

```
aws opensearch update-application \
    --id myapplication \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"arn:aws:iam::account-id:user/user-id\"
        }
    "
```

예제 2: OpenSearch UI 애플리케이션을 업데이트하여 IAM Identity Center 사용자 ID를 OpenSearch UI 애플리케이션 관리자로 추가

다음 명령을 실행하여 OpenSearch UI 애플리케이션을 업데이트하여 IAM Identity Center 사용자 ID를 OpenSearch UI 애플리케이션 관리자로 추가합니다. *## ###*를 자신의 정보로 바꿉니다.

key는 OpenSearch UI 애플리케이션의 관리자 역할과 같이 설정할 구성 항목을 지정합니다. 유효한 값에는 opensearchDashboards.dashboardAdmin.users 및 opensearchDashboards.dashboardAdmin.groups(이)가 있습니다.

*xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*는 IAM 사용자의 Amazon 리소스 이름(ARN)과 같이 키에 할당된 값을 나타냅니다.

```
aws opensearch update-application \
    --id myapplication \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"
        }
    "
```

를 사용하여 SAML 페더레이션 활성화 AWS Identity and Access Management

OpenSearch UI는 많은 자격 증명 공급자가 사용하는 개방형 표준인 Security Assertion Markup Language 2.0(SAML)을 지원합니다. 이렇게 하면 AWS Identity and Access Management (IAM)와의 ID 페더레이션이 활성화됩니다. 이 지원을 통해 계정 또는 조직의 사용자는 IAM 역할을 수임하여 OpenSearch UI에 직접 액세스할 수 있습니다. 최종 사용자를 위한 ID 제공업체 시작(IdP) Single Sign-On 환경을 생성하여 외부 ID 제공업체에서 인증하고 OpenSearch UI의 정의된 페이지로 직접 라우팅할 수 있습니다. OpenSearch UI 및 관련 데이터 소스에 액세스할 수 있는 권한이 서로 다른 다양한 IAM 역할을 수임하도록 최종 사용자 또는 그룹을 구성하여 세분화된 액세스 제어를 구현할 수도 있습니다.

이 주제에서는 OpenSearch UI에서 SAML 사용을 구성하기 위한 step-by-step 지침을 제공합니다. 이 절차에서는 Okta 자격 증명 및 액세스 관리 애플리케이션을 구성하는 단계를 예로 사용합니다. Azure Active Directory 및 Ping과 같은 다른 자격 증명 공급자의 구성 단계는 비슷합니다.

주제

- 1단계: 자격 증명 공급자 애플리케이션 설정(Okta)
- 2단계: Okta에 대한 AWS 구성 설정
- 3단계: IAM에서 Amazon OpenSearch Service 액세스 정책 생성
- 4단계: SAML을 사용하여 자격 증명 공급자가 시작한 Single Sign-On 환경 확인
- 5단계: SAML 속성 기반 세분화된 액세스 제어 구성

1단계: 자격 증명 공급자 애플리케이션 설정(Okta)

OpenSearch UI에서 SAML을 사용하려면 첫 번째 단계는 자격 증명 공급자를 설정하는 것입니다.

작업 1: Okta 사용자 생성

1. 관리 권한이 있는 사용자로 https://login.okta.com/ Okta 조직에 로그인합니다.
2. 관리자 콘솔의 탐색 창의 디렉터리에서 사람을 선택합니다.
3. 사람 추가(Add Person)를 선택합니다.
4. 이름에 사용자의 이름을 입력합니다.
5. 성에 사용자의 성을 입력합니다.

6. 사용자 이름에 이메일 형식으로 사용자의 사용자 이름을 입력합니다.
7. 암호를 설정하겠습니다를 선택하고 암호를 입력합니다를 선택합니다.
8. (선택 사항) 사용자가 처음 로그인할 때 암호를 변경하지 않도록 하려면 사용자가 처음 로그인할 때 암호를 변경해야 합니다 상자의 선택을 취소합니다.
9. 저장을 선택합니다.

작업 2: 그룹 생성 및 할당

1. 관리 권한이 있는 사용자로 https://login.okta.com/ Okta 조직에 로그인합니다.
2. 관리자 콘솔의 탐색 창의 디렉터리에서 그룹을 선택합니다.
3. 그룹 추가를 선택합니다.
4. 그룹 이름을 입력하고 저장을 선택합니다.
5. 새로 생성된 그룹을 선택한 다음 사용자 할당을 선택합니다.
6. 더하기 기호(+)를 선택한 다음 완료를 선택합니다.
7. (선택 사항) 1~6단계를 반복하여 그룹을 더 추가합니다.

작업 3: Okta 애플리케이션 생성

1. 관리 권한이 있는 사용자로 https://login.okta.com/ Okta 조직에 로그인합니다.
2. 관리자 콘솔의 탐색 창의 애플리케이션에서 애플리케이션을 선택합니다.
3. 앱 통합 생성(Create App Integration)을 선택합니다.
4. 로그인 방법으로 SAML 2.0을 선택한 후 다음을 선택합니다.
5. 앱 통합의 이름(예: **`OpenSearch_UI`**)을 입력한 후 다음을 선택합니다.
6. 앱에 다음 값을 입력합니다. 다른 값은 변경할 필요가 없습니다.

    a. 1. Single Sign On URL에 상용 AWS 리전 **`https://signin.aws.amazon.com/saml`** 또는 해당 리전에 해당하는 URL을 입력합니다.

    b. 2. 대상 URI(SP 개체 ID)에를 입력합니다**`urn:amazon:webservices`**.

    c. 3. 이름 ID 형식에를 입력합니다**`EmailAddress`**.
7. 다음을 선택합니다.
8. 내부 앱을 추가하는 Okta 고객임을 선택한 다음 생성한 내부 앱입니다를 선택합니다.
9. 마침을 클릭합니다.

10. 할당을 선택한 다음 할당을 선택합니다.
11. 그룹에 할당을 선택한 다음 추가하려는 그룹 옆에 있는 할당을 선택합니다.
12. 완료를 선택합니다.

작업 4: Okta 고급 구성 설정

사용자 지정 SAML 애플리케이션을 생성한 후 다음 단계를 완료합니다.

1. 관리 권한이 있는 사용자로 https://login.okta.com/ Okta 조직에 로그인합니다.

   관리자 콘솔의 일반 영역에서 SAML 설정에서 편집을 선택합니다.
2. 다음을 선택합니다.
3. 형식을 사용하여 기본 릴레이 상태를 OpenSearch UI 엔드포인트로 설정합니다.

   ```
   https://region.console.aws.amazon.com/aos/home?
   region=region#opensearch/applications/application-id/
   redirectToDashboardURL.
   ```

   다음은 예제입니다.

   ```
   https://us-east-2.console.aws.amazon.com/aos/home?region=us-
   east-2#opensearch/applications/abc123def4567EXAMPLE/
   redirectToDashboardURL
   ```
4. 속성 문(선택 사항)에서 다음 속성을 추가합니다.

   a. 역할 속성을 사용하여 IAM 역할 및 자격 증명 공급자를 쉼표로 구분된 형식으로 제공합니다. AWS 구성을 설정할 때 이후 단계에서 동일한 IAM 역할 및 자격 증명 공급자를 사용합니다.

   b. RoleSessionName에 대한 user.login을 설정합니다. 이는 역할을 수임할 때 발급되는 임시 자격 증명의 식별자로 사용됩니다.

   참조용:

명칭	이름 형식	형식	예시
`https://aws.amazon.com/SAML/`	지정되지 않음	`arn:aws:iam:: aws-account-id :role/`	`arn:aws:iam::111222333444:role/`

명칭	이름 형식	형식	예시
Attributes/Role		role-name,arn:aws:iam::*aws-account-id*:saml-provider/*provider-name*	oktarole,arn:aws:iam::112223334444:saml-provider/oktaidp
https://aws.amazon.com/SAML/Attributes/RoleSessionName	지정되지 않음	user.login	user.login

5. 속성 속성을 추가한 후 다음을 선택한 후 완료를 선택합니다.

속성은 다음 이미지에 표시된 것과 형식이 비슷해야 합니다. 기본 릴레이 상태 값은 Okta에서 Single Sign-On 검증을 완료한 후 계정 또는 조직의 최종 사용자에 대한 랜딩 페이지를 정의하는 URL입니다. OpenSearch UI의 모든 페이지로 설정한 다음 해당 URL을 의도한 최종 사용자에게 제공할 수 있습니다.

2단계: Okta에 대한 AWS 구성 설정

다음 작업을 완료하여 Okta에 대한 AWS 구성을 설정합니다.

작업 1: Okta 정보 수집

이 단계에서는 나중에 구성할 수 있도록 Okta 정보를 수집해야 합니다 AWS.

1. 관리 권한이 있는 사용자로 https://login.okta.com/ Okta 조직에 로그인합니다.
2. 로그인 탭의 페이지 오른쪽 하단에서 SAML 설정 지침 보기를 선택합니다.
3. 자격 증명 공급자 Single Sign-On URL의 값을 기록해 둡니다. SQL Workbench/J와 같은 타사 SQL 클라이언트에 연결할 때이 URL을 사용할 수 있습니다.
4. 블록 4에서 자격 증명 공급자 메타데이터를 사용한 다음 메타데이터 파일을 .xml 형식(예: metadata.xml)으로 저장합니다.

작업 2: IAM 공급자 생성

IAM 공급자를 생성하려면 다음 단계를 완료합니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. 탐색 창의 액세스 관리에서 자격 증명 공급자를 선택합니다.
3. 공급자 추가를 선택합니다.
4. 공급자 유형에서 SAML을 선택합니다.
5. 공급자 이름에 이름을 입력합니다.
6. 메타데이터 문서에서 파일 선택을 선택하고 이전에 다운로드한 메타데이터 파일(.xml)을 업로드합니다.
7. 공급자 추가를 선택합니다.

작업 3: IAM 역할 생성

AWS Identity and Access Management 역할을 생성하려면 다음 단계를 완료합니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. 탐색 창의 액세스 관리에서 역할을 선택합니다.
3. 역할 생성을 선택합니다.
4. 신뢰할 수 있는 엔터티 유형에서 SAML 2.0 페더레이션을 선택합니다.
5. SAML 2.0 기반 공급자에서 이전에 생성한 자격 증명 공급자를 선택합니다.
6. 프로그래밍 방식 및 AWS Management Console 액세스 허용을 선택합니다.
7. 다음을 선택합니다.
8. 권한 정책 목록에서 이전에 생성한 정책의 확인란과 OpenSearchFullAccess의 확인란을 선택합니다.
9. 다음을 선택합니다.
10. 검토 영역의 역할 이름에 역할 이름을 입력합니다. 예: **`oktarole`**.
11. (선택 사항) 설명에 역할의 용도에 대한 간략한 설명을 입력합니다.
12. 역할 생성을 선택합니다.
13. 방금 생성한 역할로 이동하여 신뢰 관계 탭을 선택한 다음 신뢰 정책 편집을 선택합니다.
14. 문 편집 창의 STS에 대한 작업 추가에서 TagSession에 대한 상자를 선택합니다.

15. 정책 업데이트를 선택합니다.

3단계: IAM에서 Amazon OpenSearch Service 액세스 정책 생성

OpenSearch 액세스 제어를 위해 IAM 역할을 구성하는 방법을 알아봅니다. IAM 역할을 사용하면 Okta 사용자 그룹이 OpenSearch 리소스에 액세스할 수 있도록 세분화된 액세스 제어를 구현할 수 있습니다. 이 주제에서는 두 개의 예제 그룹을 사용하는 IAM 역할 기반 구성을 보여줍니다.

Sample group: Alice

요청:

```
GET _plugins/_security/api/roles/alice-group
```

결과:

```
{
  "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

요청:

```
GET _plugins/_security/api/roles/bob-group
```

결과:

```
{
  "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
  }
}
```

다음 예제와 같이 백엔드 역할 매핑을 사용하여 Amazon OpenSearch Service 도메인 역할을 IAM 역할에 매핑할 수 있습니다.

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

4단계: SAML을 사용하여 자격 증명 공급자가 시작한 Single Sign-On 환경 확인

기본 릴레이 상태의 URL을 열어 Okta 인증 페이지를 엽니다. 최종 사용자의 자격 증명을 입력합니다. 자동으로 OpenSearch UI로 리디렉션됩니다.

다음 이미지와 같이 탐색 패널 하단의 사용자 아이콘을 선택하여 현재 자격 증명을 확인할 수 있습니다.

탐색 패널 하단의 개발자 도구에 액세스하고 콘솔에서 쿼리를 실행하여 사용자의 세분화된 액세스 제어 권한을 확인할 수도 있습니다. 다음은 샘플 쿼리입니다.

Example 1: Displays information about the current user

요청:

```
GET _plugins/_security/api/account
```

결과:

```
{
  "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

요청:

```
GET bob-test/_search
```

결과:

```
{
  "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

요청:

```
GET alice-test
```

결과:

```
{
  "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
}
```

5단계: SAML 속성 기반 세분화된 액세스 제어 구성

Amazon OpenSearch Service를 사용하면 SAML과 함께 세분화된 액세스 제어를 사용하여 ID 제공업체의 사용자 및 그룹을 OpenSearch 세분화된 액세스 제어 사용자 및 역할에 동적으로 매핑할 수 있습니다. 이러한 역할의 범위를 특정 OpenSearch 도메인 및 서버리스 컬렉션으로 지정하고 인덱스 수준 권한 및 문서 수준 보안을 정의할 수 있습니다.

> **Note**
>
> 세분화된 액세스 제어에 대한 자세한 내용은 섹션을 참조하세요the section called “세분화된 액세스 제어”.

주제

- 세분화된 액세스 제어를 위한 SAML 속성
- 작업 1: 세분화된 액세스 제어를 위한 Okta 구성
- 작업 2: OpenSearch 도메인에서 SAML 구성
- 작업 3: OpenSearch Serverless 컬렉션에서 SAML 구성

세분화된 액세스 제어를 위한 SAML 속성

subjectKey

인증할 사용자를 식별하는 이메일 또는 사용자 이름과 같은 고유한 사용자 속성에 매핑됩니다.

rolesKey

권한 부여를 위한 역할 또는 권한을 결정하는 IdP의 그룹 또는 역할 속성에 매핑됩니다.

작업 1: 세분화된 액세스 제어를 위한 Okta 구성

세분화된 액세스 제어를 위해 Okta를 구성하려면

1. 속성 문 섹션에서 OpenSearch 사용자 보안 주체에 대한 새 속성을 추가합니다.

   - 이름: `UserName`
   - 값: `${user-email}`

   이 속성은 인증을 위한 OpenSearch 세분화된 액세스 제어 구성에서 주체 키로 사용됩니다.
2. 그룹 속성 문 섹션에서 역할에 대한 그룹 속성을 추가합니다.

   - 이름: `groups`
   - 필터: `OpenSearch_xxx`

   이 속성은 권한 부여를 위해 그룹을 OpenSearch 세분화된 액세스 제어 역할에 매핑하기 위한 역할 키로 사용됩니다.

작업 2: OpenSearch 도메인에서 SAML 구성

OpenSearch 도메인에서 SAML을 구성하려면

1. AWS 관리 콘솔에서 OpenSearch UI 사용자에 대해 세분화된 액세스 제어를 활성화하려는 OpenSearch Service 도메인을 식별합니다.
2. 특정 도메인의 세부 정보 페이지로 이동합니다.
3. 보안 구성 탭을 선택하고 편집을 클릭합니다.
4. IAM Federate를 통해 SAML을 확장합니다.
5. Okta에서 정의`roleKey`한 `subjectKey` 및를 입력합니다.
6. 변경 사항 저장을 선택합니다.

를 사용하여 세분화된 액세스 제어를 구성할 수도 있습니다 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
```

기존 도메인을 업데이트하려면:

```
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
```

작업 3: OpenSearch Serverless 컬렉션에서 SAML 구성

OpenSearch Serverless에서 SAML 기반 세분화된 액세스 제어를 구성하려면

1. 를 열고 Amazon OpenSearch Service로 AWS Management Console 이동합니다.
2. 탐색 창의 서버리스에서 보안을 선택한 다음 인증을 선택합니다.
3. IAM 연동 섹션에서 편집을 선택합니다.

   이 구성을 사용하여 SAML 속성 기반 세분화된 액세스 제어를 제어할 수 있습니다. IAM 연동은 기본적으로 비활성화되어 있습니다.

4. IAM 연동 활성화를 선택합니다.
5. Okta에서 정의한 `subjectKey` 및 `roleKey` 값을 입력합니다.

   자세한 내용은 the section called “세분화된 액세스 제어를 위한 SAML 속성 ” 단원을 참조하십시오.
6. 저장을 선택합니다.
7. 탐색 창의 서버리스에서 데이터 액세스 정책을 선택합니다.
8. 기존 정책을 업데이트하거나 새 정책을 생성합니다.
9. 규칙을 확장하고 보안 주체 추가를 선택한 다음 IAM 페더레이션 사용자 및 그룹을 선택합니다.
10. 필요한 보안 주체를 추가하고 저장을 선택합니다.
11. 권한 부여를 선택합니다.
12. 이 규칙에서 다음을 수행합니다.

    - 선택한 보안 주체에 대해 정의할 권한을 선택합니다.
    - 권한을 적용할 컬렉션을 지정합니다.
    - 선택적으로 인덱스 수준 권한을 정의합니다.

    > **Note**
    >
    > 여러 규칙을 생성하여 여러 보안 주체 그룹에 서로 다른 권한을 할당할 수 있습니다.
13. 완료하였으면 저장을 선택합니다.
14. 생성(Create)을 선택합니다.

또는 CLI를 사용하여 아래와 같이 컬렉션에 대한 보안 구성을 생성할 수 있습니다.

```
aws opensearchserverless create-security-config --region "region"  --type iamfederation
 --name "configuration_name" --description "description" --iam-federation-options
 '{"groupAttribute":"GroupKey","userAttribute":"UserKey"}'
```

데이터 소스 연결 및 Virtual Private Cloud 액세스 권한 관리

이 섹션의 절차를 사용하여 데이터 소스 연결을 관리하고 Virtual Private Cloud(VPC)에 필요한 액세스 권한을 구성합니다.

주제

- OpenSearch UI 애플리케이션과 데이터 소스 연결
- VPC의 도메인에 대한 액세스 관리
- VPC에서 OpenSearch Serverless 컬렉션에 대한 액세스 구성

OpenSearch UI 애플리케이션과 데이터 소스 연결

OpenSearch UI 애플리케이션을 생성한 후 콘솔 또는를 사용하여 하나 이상의 데이터 소스와 AWS CLI 연결할 수 있습니다. 그런 다음 최종 사용자는 검색, 대시보드 작업 등을 위해 이러한 데이터 소스에서 데이터를 검색할 수 있습니다.

데이터 소스를 OpenSearch UI 애플리케이션과 연결(콘솔)

콘솔을 사용하여 데이터 소스를 OpenSearch UI 애플리케이션과 연결하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. OpenSearch UI(대시보드)를 선택한 다음 OpenSearch UI 애플리케이션의 이름을 선택합니다.
3. 연결된 데이터 소스 영역에서 데이터 소스 관리를 선택합니다.
4. 애플리케이션과 연결할 OpenSearch 도메인 및 컬렉션 중에서 선택합니다.

> **Tip**
>
> 찾고 있는 데이터 소스를 찾을 수 없는 경우 관리자에게 문의하여 필요한 권한을 부여하세요. 자세한 내용은 the section called “IAM Identity Center 인증을 사용하는 애플리케이션을 생성할 수 있는 권한(선택 사항)” 단원을 참조하십시오.

5. 다음을 선택한 후 저장을 선택합니다.

데이터 소스를 애플리케이션에 연결하면 애플리케이션 세부 정보 페이지에서 애플리케이션 시작 버튼이 활성화됩니다. 애플리케이션 시작을 선택하여 워크스페이스를 생성하고 관리할 수 있는 OpenSearch 시작 페이지를 열 수 있습니다.

워크스페이스 작업에 대한 자세한 내용은 섹션을 참조하세요the section called “Amazon OpenSearch Service 워크스페이스 사용”.

VPC의 도메인에 대한 액세스 관리

VPC의 OpenSearch 도메인이 애플리케이션과 연결된 경우 VPC 관리자는 콘솔 또는를 사용하여 OpenSearch UI와 VPC 간의 액세스를 승인해야 합니다 AWS CLI.

VPC의 도메인에 대한 액세스 관리(콘솔)

를 사용하여 VPC 도메인에 대한 액세스를 구성하려면 AWS Management Console:

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 도메인을 선택하고 VPC 도메인의 이름을 선택합니다.

   -또는-

   도메인 생성을 선택한 다음 도메인에 대한 세부 정보를 구성합니다.
3. VPC 엔드포인트 탭을 선택한 다음 보안 주체 권한 부여를 선택합니다.
4. 보안 주체 권한 부여 대화 상자의 다른 AWS 서비스에서 보안 주체 권한 부여를 선택한 다음 목록에서 OpenSearch 애플리케이션(대시보드)을 선택합니다.
5. Authorize를 선택합니다.

VPC의 도메인에 대한 액세스 관리(AWS CLI)

를 사용하여 VPC 도메인에 권한을 부여하려면 AWS CLI

를 사용하여 VPC 도메인에 권한을 부여하려면 다음 명령을 AWS CLI실행합니다. *## ###*를 자신의 정보로 바꿉니다.

```
aws opensearch authorize-vpc-endpoint-access \
 --domain-name domain-name \
 --service application.opensearchservice.amazonaws.com \
 --region region-id
```

콘솔을 사용하여 VPC 도메인 연결을 취소하려면

연결이 더 이상 필요하지 않은 경우 VPC 도메인 소유자는 다음 절차에 따라 액세스를 취소할 수 있습니다.

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 도메인을 선택하고 VPC 도메인의 이름을 선택합니다.

3. VPC 엔드포인트 탭을 선택한 다음 OpenSearch 애플리케이션(대시보드) 행의 버튼을 선택합니다.
4. 액세스 취소를 선택합니다.

를 사용하여 VPC 도메인 연결을 취소하려면 AWS CLI

OpenSearch UI 애플리케이션과의 VPC 도메인 연결을 취소하려면 다음 명령을 실행합니다. ## ## #를 자신의 정보로 바꿉니다.

```
aws opensearch revoke-vpc-endpoint-access \
    --domain-name domain-name \
    --service application.opensearchservice.amazonaws.com \
    --region region-id
```

VPC에서 OpenSearch Serverless 컬렉션에 대한 액세스 구성

VPC의 Amazon OpenSearch Serverless 컬렉션이 애플리케이션과 연결된 경우 VPC 관리자는 새 네트워크 정책을 생성하고 컬렉션에 연결하여 액세스를 승인할 수 있습니다.

VPC에서 OpenSearch Serverless 컬렉션에 대한 액세스 구성(콘솔)

콘솔을 사용하여 VPC의 OpenSearch Serverless 컬렉션에 대한 액세스를 구성하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 네트워크 정책을 선택하고 네트워크 정책의 이름을 선택한 다음 편집을 선택합니다.

   -또는-

   네트워크 정책 생성을 선택한 다음 정책에 대한 세부 정보를 구성합니다.
3. 액세스 유형 영역에서 프라이빗(권장)을 선택한 다음 AWS 서비스 프라이빗 액세스를 선택합니다.
4. 검색 필드에서 서비스를 선택한 다음를 선택합니다`application.opensearchservice.amazonaws.com`.
5. 리소스 유형 영역에서 OpenSearch 엔드포인트에 대한 액세스 활성화 상자를 선택합니다.
6. 검색 컬렉션(들) 또는 특정 접두사 용어 입력의 검색 필드에서 컬렉션 이름을 선택한 다음 네트워크 정책과 연결할 컬렉션의 이름을 입력하거나 선택합니다.
7. 새 네트워크 정책에 대해 생성 또는 기존 네트워크 정책에 대해 업데이트를 선택합니다.

VPC에서 OpenSearch Serverless 컬렉션에 대한 액세스 구성(AWS CLI)

를 사용하여 VPC의 OpenSearch Serverless 컬렉션에 대한 액세스를 구성하려면 AWS CLI

1. 다음과 유사한 .json 파일을 생성합니다. *## ###*를 자신의 정보로 바꿉니다.

```
{
    "Description" : "policy-description",
    "Rules": [{
       "ResourceType" : "collection",
        "Resource" : ["collection/collection_name"]
     }],
    "SourceServices" : [
          "application.opensearchservice.amazonaws.com"
      ],
      "AllowFromPublic" : false
}
```

2. OpenSearch UI 애플리케이션에서 작동하도록 VPC의 컬렉션에 대한 네트워크 정책을 생성하거나 업데이트합니다.

   Create a network policy

   다음 명령을 실행합니다. *## ###*를 자신의 정보로 바꿉니다.

```
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
```

   이 명령은 다음과 유사한 정보를 반환합니다.

```
{
    "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

다음 명령을 실행합니다. *## ###*를 자신의 정보로 바꿉니다.

```
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
```

이 명령은 다음과 유사한 정보를 반환합니다.

```
{
    "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"
    }
}
```

Amazon OpenSearch Service 워크스페이스 사용

Amazon OpenSearch Service는 여러 사용 사례별 워크스페이스 생성을 지원합니다. 각 워크스페이스는 관찰성, 보안 분석 및 검색과 같은 인기 있는 사용 사례에 대해 선별된 환경을 제공합니다. 또한 Workspace는 공동 작업자 관리를 지원하므로 의도한 공동 작업자와만 워크스페이스를 공유하고 각 공동 작업자에 대한 권한을 관리할 수 있습니다.

OpenSearch UI 애플리케이션 워크스페이스 생성

OpenSearch UI 애플리케이션을 생성하여 데이터 소스와 연결하고 애플리케이션에 대한 사용자 권한을 구성한 후 OpenSearch UI 애플리케이션을 시작하여 워크스페이스를 생성할 수 있습니다.

워크스페이스 생성을 시작하려면 애플리케이션 세부 정보 페이지에서 애플리케이션 시작 버튼을 선택하거나 OpenSearch UI 애플리케이션 URL을 사용하여 새 브라우저 창에서 OpenSearch UI 애플리케이션 홈페이지를 열 수 있습니다.

OpenSearch UI 애플리케이션은 워크스페이스를 생성하는 옵션을 제공하고 사용 사례별로 분류된 홈페이지의 모든 기존 워크스페이스를 나열합니다.

지원되는 워크스페이스 유형에 대한 자세한 내용은 섹션을 참조하세요the section called “Workspace 유형”.

Workspace 개인 정보 보호 및 공동 작업자

워크스페이스의 개인 정보 설정을 모든 사용자의 기본 권한 수준으로 정의할 수 있습니다. 워크스페이스를 생성하는 동안이 작업을 수행하거나 기존 워크스페이스를 수정할 수 있습니다(워크스페이스 공동 작업자 탭에서). 다음 세 가지 프라이버시 옵션 중에서 선택할 수 있습니다.

- 공동 작업자에 대한 비공개 - 워크스페이스에 명시적으로 추가하는 공동 작업자만 워크스페이스에 액세스할 수 있습니다. 각 공동 작업자에 대한 권한 수준을 정의할 수 있습니다.
- 누구나 볼 수 있음 - OpenSearch UI 애플리케이션에 액세스할 수 있는 사람은 누구나 워크스페이스에 액세스하여 자산을 볼 수 있지만 워크스페이스를 변경할 수는 없습니다.
- 누구나 편집할 수 있음 - OpenSearch UI 애플리케이션에 액세스할 수 있는 사람은 누구나 워크스페이스에 액세스하고, 워크스페이스의 자산을 보고, 워크스페이스의 자산을 변경할 수 있습니다.

워크스페이스 공동 작업자 탭에서 IAM 사용자 또는 역할 및 AWS IAM Identity Center 사용자를 워크스페이스의 공동 작업자로 추가할 수 있습니다. 공동 작업자가 선택할 수 있는 권한에는 세 가지 수준이 있습니다.

- 읽기 전용 - 공동 작업자는 워크스페이스의 자산만 볼 수 있습니다. 워크스페이스가 누구나 개인 정보 보호 설정을 편집할 수 있도록 구성된 경우이 설정이 재정의됩니다.
- 읽기 및 쓰기 - 공동 작업자가 워크스페이스에서 자산을 보고 편집할 수 있습니다. 누구나 개인 정보 설정을 사용하도록 구성된 워크스페이스가 개인 정보 설정을 볼 수 있는 경우에도 공동 작업자는 계속 편집할 수 있습니다.
- 관리자 - 공동 작업자가 설정을 업데이트하고 워크스페이스를 삭제할 수 있습니다. 공동 작업자는 워크스페이스 프라이버시 설정을 변경하고 공동 작업자를 관리할 수도 있습니다. 워크스페이스를 생성하는 사용자는 워크스페이스 관리자로 자동 할당됩니다.

Workspace 유형

Amazon OpenSearch Service는 다섯 가지 워크스페이스 유형을 제공하며, 각 워크스페이스 유형에는 다양한 사용 사례에 맞는 다양한 기능이 있습니다.

- 관찰성 워크스페이스는 로그, 지표 및 추적 모니터링을 통해 시스템 상태, 성능 및 신뢰성에 대한 가시성을 확보하도록 설계되었습니다.

- Security Analytics 워크스페이스는 시스템 및 데이터 전반의 잠재적 보안 위협 및 취약성을 탐지하고 조사하도록 설계되었습니다.
- 검색 워크스페이스는 조직의 데이터 소스에서 관련 정보를 빠르게 찾고 탐색하도록 설계되었습니다.
- Essentials 워크스페이스는 OpenSearch Serverless를 위해 데이터 소스로 설계되었으며, 데이터를 분석하여 인사이트를 도출하고, 패턴과 추세를 식별하고, 데이터 기반 결정을 빠르게 내릴 수 있습니다. Essentials 워크스페이스에서 조직의 데이터 소스 전반에 걸쳐 관련 정보를 찾고 탐색할 수 있습니다.
- 분석(모든 기능) 워크스페이스는 다목적 사용 사례를 위해 설계되었으며 OpenSearch Service UI(대시보드)에서 사용할 수 있는 모든 기능을 지원합니다.

클러스터 간 검색을 통한 리전 간 및 계정 간 데이터 액세스

Amazon OpenSearch Serverless에서 클러스터 간 검색을 사용하면 연결된 여러 도메인에서 쿼리 및 집계를 수행할 수 있습니다.

Amazon OpenSearch Serverless의 클러스터 간 검색은 소스 도메인 및 대상 도메인의 개념을 사용합니다. 클러스터 간 검색 요청은 소스 도메인에서 시작됩니다. 대상 도메인은 소스 도메인이 쿼리할 다른 AWS 계정 또는 AWS 리전 (또는 둘 다)에 있을 수 있습니다. 클러스터 간 검색을 사용하여 동일한 계정의 OpenSearch UI와 연결하도록 소스 도메인을 구성한 다음 대상 도메인에 대한 연결을 생성할 수 있습니다. 따라서 다른 계정 또는 리전에 있더라도 대상 도메인의 데이터와 함께 OpenSearch UI를 사용할 수 있습니다.

Amazon OpenSearch Service에서 송수신되는 AWS 데이터에 대해 표준 데이터 전송 요금을 지불합니다. OpenSearch Service 도메인 내의 노드 간에 전송된 데이터에 대해서는 요금이 부과되지 않습니다. "in" 및 "out" 요금에 대한 자세한 내용은 Amazon EC2 온디맨드 요금 페이지의 데이터 전송을 참조하세요.

클러스터 간 검색을 OpenSearch UI를 다른 계정 또는 다른 리전의 클러스터와 연결하는 메커니즘으로 사용할 수 있습니다. 도메인 간 요청은 기본적으로 node-to-node 암호화의 일부로 전송 중에 암호화됩니다.

> **Note**
>
> 오픈 소스 OpenSearch 도구는 클러스터 간 검색도 문서화합니다. 오픈 소스 도구에 대한 설정은 관리형 Amazon OpenSearch Serverless 도메인과 비교하여 오픈 소스 클러스터에서 크게 다릅니다.

특히 Amazon OpenSearch Serverless에서는 cURL 요청을 사용하는 AWS Management Console 대신를 사용하여 클러스터 간 연결을 구성합니다. 관리형 서비스는 세분화된 액세스 제어 외에도 클러스터 간 인증에 AWS Identity and Access Management (IAM)를 사용합니다. 따라서 오픈 소스 OpenSearch 설명서 대신이 주제의 콘텐츠를 사용하여 도메인에 대한 클러스터 간 검색을 구성하는 것이 좋습니다.

클러스터 간 검색 사용 시 기능적 차이

클러스터 간 검색을 사용하여 생성된 대상 도메인은 일반 도메인과 비교하여 다음과 같은 기능적 차이점과 요구 사항이 있습니다.

- 원격 클러스터에는에 쓰거나 PUT 명령을 실행할 수 없습니다. 원격 클러스터에 대한 액세스는 읽기 전용입니다.
- 소스 도메인과 대상 도메인 모두 OpenSearch 도메인이어야 합니다. OpenSearch UI용 Elasticsearch 도메인 또는 자체 관리형 OpenSearch/Elasticsearch 클러스터는 연결할 수 없습니다. OpenSearch
- 도메인은 다른 도메인에 최대 20개의 연결을 가질 수 있습니다. 여기에는 발신 연결과 수신 연결이 모두 포함됩니다.
- 소스 도메인은 대상 도메인과 동일하거나 더 높은 버전의 OpenSearch에 있어야 합니다. 두 도메인 간에 양방향 연결을 설정하려면 두 도메인이 동일한 버전이어야 합니다. 연결하기 전에 두 도메인을 모두 최신 버전으로 업그레이드하는 것이 좋습니다. 양방향 연결을 설정한 후 도메인을 업데이트해야 하는 경우 먼저 연결을 삭제한 다음 나중에 다시 생성해야 합니다.
- 원격 클러스터에는 사용자 지정 사전 또는 SQL을 사용할 수 없습니다.
- AWS CloudFormation 를 사용하여 도메인을 연결할 수 없습니다.
- M3 또는 버스트 가능(T2 및 T3) 인스턴스에서는 클러스터 간 검색을 사용할 수 없습니다.
- 클러스터 간 검색은 Amazon OpenSearch Serverless 컬렉션에서 작동하지 않습니다.

OpenSearch UI에 대한 클러스터 간 검색 사전 조건

두 개의 OpenSearch 도메인으로 클러스터 간 검색을 설정하기 전에 도메인이 다음 요구 사항을 충족하는지 확인합니다.

- 두 도메인 모두에 대해 세분화된 액세스 제어가 활성화됨
- Node-to-node 암호화가 두 도메인 모두에 대해 활성화됨

주제

- 클러스터 간 검색을 사용하여 리전 간 및 계정 간 데이터 액세스에 대한 액세스 권한 설정
- 도메인 간 연결 생성
- 교차 클러스터 검색을 사용하여 교차 리전 및 교차 계정 데이터 액세스를 위한 보안 설정 테스트
- 연결 삭제

클러스터 간 검색을 사용하여 리전 간 및 계정 간 데이터 액세스에 대한 액세스 권한 설정

클러스터 간 검색 요청을 소스 도메인으로 보내면 도메인이 해당 요청을 도메인 액세스 정책과 비교하여 평가합니다. 클러스터 간 검색에는 세분화된 액세스 제어가 필요합니다. 다음은 소스 도메인에 대한 오픈 액세스 정책이 있는 예제입니다.

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**
>
> 경로에 원격 인덱스를 포함하는 경우 도메인 ARN에서 URI를 URL로 인코딩해야 합니다. 예를 들어 다음 ARN 형식을 사용합니다.

:arn:aws:es:us-east-1:111222333444:domain/my-domain/local_index,dst%3Aremote_index
다음 ARN 형식을 사용하지 마십시오.
arn:aws:es:us-east-1:111222333444:domain/my-domain/local_index,dst:remote_index.

세분화된 액세스 제어 외에도 제한적인 액세스 정책을 사용하기로 선택한 경우 정책은 최소한에 대한 액세스를 허용해야 합니다es:ESHttpGet. 다음은 예제입니다.

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/*"
    }
  ]
}
```

소스 도메인에 대한 세분화된 액세스 제어는 요청을 평가하여 유효한 IAM 또는 HTTP 기본 자격 증명으로 서명되었는지 확인합니다. 그렇다면 세분화된 액세스 제어는 다음에 사용자에게 검색을 수행하고 데이터에 액세스할 수 있는 권한이 있는지 여부를 평가합니다.

다음은 검색에 대한 권한 요구 사항입니다.

- 요청이 대상 도메인의 데이터만 검색하는 경우(예: dest-alias:dest-index/_search), 대상 도메인에 대한 권한만 필요합니다.
- 요청이 두 도메인의 데이터를 검색하는 경우(예: source-index,dest-alias:dest-index/_search), 두 도메인에 대한 권한이 필요합니다.

- 세분화된 액세스 제어를 사용하려면 관련 인덱스에 대한 표준 읽기 또는 검색 권한 외에도 권한이 `indices:admin/shards/search_shards` 필요합니다.

소스 도메인은 요청을 대상 도메인에 전달합니다. 대상 도메인은 해당 도메인 액세스 정책을 기준으로 이 요청을 평가합니다. 문서 인덱싱 및 표준 검색 수행과 같은 OpenSearch UI의 모든 기능을 지원하려면 전체 권한을 설정해야 합니다. 다음은 대상 도메인에 대한 권장 정책의 예입니다.

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/"
    }
  ]
}
```

기본 검색만 수행하려는 경우 최소 정책 요구 사항은 와일드카드 지원 없이 대상 도메인에 `es:ESCrossClusterGet` 권한을 적용하는 것입니다. 예를 들어 이전 정책에서 도메인 이름을 */my-destination-domain/*# ## /my-destination-domain*으로 지정합니다.

이 경우 대상 도메인은 검색을 수행하고 결과를 소스 도메인에 반환합니다. 소스 도메인은 자체 결과(있는 경우)를 대상 도메인의 결과와 결합하여 반환합니다.

도메인 간 연결 생성

클러스터 간 검색 연결은 소스 도메인에서 대상 도메인으로 단방향입니다. 즉, 대상 도메인(다른 계정 또는 리전)은 OpenSearch UI에 로컬인 소스 도메인을 쿼리할 수 없습니다. 소스 도메인은 대상 도메인에 대한 아웃바운드 연결을 생성합니다. 대상 도메인은 소스 도메인으로부터 인바운드 연결 요청을 받습니다.

도메인 간 연결을 생성하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 도메인을 선택합니다.
3. 소스 도메인으로 사용할 도메인의 이름을 선택한 다음 연결 탭을 선택합니다.
4. 아웃바운드 연결 영역에서 요청을 선택합니다.
5. [연결 별칭(Connection alias)]에 연결 이름을 입력합니다. 연결 별칭은 OpenSearch UI에서 대상 도메인을 선택하는 데 사용됩니다.
6. 연결 모드에서 클러스터 간 검색 또는 복제를 위해 직접을 선택합니다.
7. 검색 중에 연결이 사용할 수 없는 클러스터를 건너뛰도록 지정하려면 사용할 수 없는 클러스터 건너뛰기 상자를 선택합니다. 이 옵션을 선택하면 클러스터 간 쿼리가 하나 이상의 원격 클러스터의 실패에 관계없이 부분적인 결과를 반환합니다.
8. 대상 클러스터에서이의 클러스터에 연결 AWS 계정과 다른의 클러스터에 연결 AWS 계정 중에서 선택합니다.
9. 원격 도메인 ARN에 클러스터의 Amazon 리소스 이름(ARN)을 입력합니다. 도메인 ARN은 도메인 세부 정보 페이지의 일반 정보 영역에 위치할 수 있습니다.

   도메인은 다음 요구 사항을 충족해야 합니다.

   - ARN은 형식이어야 합니다`arn:`*`partition`*`:es:`*`regionaccount-id`*`:`*`type`*`/`*`domain-id`*. 예:

     ```
     arn:aws:es:us-east-2:111222333444:domain/my-domain
     ```
   - 도메인은 OpenSearch 버전 1.0(이상) 또는 Elasticsearch 버전 6.7(이상)을 사용하도록 구성해야 합니다.
   - 도메인에서 세분화된 액세스 제어를 활성화해야 합니다.

- 도메인이 OpenSearch를 실행 중이어야 합니다.

10. 요청을 선택합니다.

클러스터 간 검색은 먼저 연결 요청을 검증하여 전제 조건이 충족되는지 확인합니다. 도메인이 호환되지 않으면 연결 요청이 `Validation failed` 상태로 전환됩니다.

연결 요청이 성공적으로 검증되면 대상 도메인으로 전송되어 승인되어야 합니다. 이 승인을 받을 때까지 연결은 `Pending acceptance` 상태로 유지됩니다. 대상 도메인에서 연결 요청이 수락되면 상태가 `Active`으로 변경되고 대상 도메인을 쿼리에 사용할 수 있게 됩니다.

도메인 페이지에는 대상 도메인의 전체 도메인 상태 및 인스턴스 상태 세부 정보가 표시됩니다. 도메인 소유자만 도메인과의 연결을 유연하게 생성하고 보고 제거하고 모니터링할 수 있습니다.

연결이 설정되면 연결된 도메인의 노드 간에 흐르는 모든 트래픽이 암호화됩니다. VPC 도메인을 비 VPC 도메인에 연결하고 비 VPC 도메인이 인터넷에서 트래픽을 수신할 수 있는 퍼블릭 엔드포인트인 경우 도메인 간의 클러스터 간 트래픽은 여전히 암호화되고 안전합니다.

교차 클러스터 검색을 사용하여 교차 리전 및 교차 계정 데이터 액세스를 위한 보안 설정 테스트

교차 클러스터 검색을 사용하여 교차 리전 및 교차 계정 데이터 액세스에 대한 액세스 권한을 설정한 후에는 협업 API 개발을 위한 타사 플랫폼Postman인를 사용하여 설정을 테스트하는 것이 좋습니다.

를 사용하여 보안 설정을 설정하려면 Postman

1. 대상 도메인에서 문서를 인덱싱합니다. 다음은 샘플 요청입니다.

```
POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1
{
"Dracula": "Bram Stoker"
}
```

2. 소스 도메인에서 이 인덱스를 쿼리하려면 쿼리 내에 대상 도메인의 연결 별칭을 포함합니다. 도메인 대시보드의 연결(Connections) 탭에서 연결 별칭을 찾을 수 있습니다. 다음은 샘플 요청 및 잘린 응답입니다.

```
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. (선택 사항) 단일 검색에서 여러 도메인을 포함하는 구성을 생성할 수 있습니다. 예를 들어 다음을 설정했다고 가정해 보겠습니다.

   연결 별칭 이름이 domain-a domain-b인 간 연결 cluster_b

   연결 별칭이 domain-a 인 간 domain-c연결 cluster_c

   이 경우 검색에는 콘텐츠 , domain-a domain-b및가 포함됩니다domain-c. 다음은 샘플 요청 및 응답입니다.

   요청

```
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"
    }
  }
}
```

   응답:

```
{
"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
```

```
        }
      }
    ]
  }
}
```

연결 설정에서 사용할 수 없는 클러스터를 건너뛰도록 선택하지 않은 경우 검색 요청이 성공적으로 실행되려면 검색하는 모든 대상 클러스터를 사용할 수 있어야 합니다. 그렇지 않으면 전체 요청이 실패합니다. 도메인 중 하나를 사용할 수 없더라도 검색 결과가 반환되지 않습니다.

연결 삭제

연결을 삭제하면 대상 도메인에서 클러스터 간 검색 작업이 중지됩니다.

소스 또는 대상 도메인에서 다음 절차를 수행하여 연결을 제거할 수 있습니다. 연결을 제거한 후에도 15일 `Deleted` 동안 상태로 표시됩니다.

활성 클러스터 간 연결이 있는 도메인은 삭제할 수 없습니다. 도메인을 삭제하려면 먼저 해당 도메인과의 수신 연결과 발신 연결을 모두 제거합니다. 그러면 도메인을 삭제하기 전에 클러스터 간 도메인 사용자를 고려할 수 있습니다.

연결 삭제

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색 창에서 도메인을 선택합니다.
3. 삭제할 도메인의 이름을 선택한 다음 연결 탭을 선택합니다.
4. 삭제할 연결의 이름을 선택합니다.
5. 삭제를 선택한 다음 삭제를 확인합니다.

VPC 엔드포인트에서 OpenSearch UI에 대한 액세스 관리

를 사용하여 VPC와 OpenSearch UI 간에 프라이빗 연결을 생성할 수 있습니다 AWS PrivateLink. 이 연결을 사용하면 동일한 VPC에 있는 것처럼 OpenSearch UI 애플리케이션에 액세스할 수 있습니다. 이렇게 하면 인터넷 게이트웨이, NAT 디바이스, VPN 연결을 구성하거나 연결을 AWS Direct Connect 설정할 필요가 없습니다. VPC의 인스턴스는 OpenSearch UI에 액세스하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

이 프라이빗 연결을 설정하려면 먼저 기반 인터페이스 엔드포인트를 생성합니다 AWS PrivateLink. 엔드포인트 네트워크 인터페이스는 인터페이스 엔드포인트에 지정한 각 서브넷에 자동으로 생성됩니다. 이는 OpenSearch UI 애플리케이션으로 향하는 트래픽의 진입점 역할을 하는 요청자 관리형 네트워크 인터페이스입니다.

VPC와 OpenSearch UI 간에 프라이빗 연결 생성

AWS Management Console 또는를 사용하여 VPC에서 OpenSearch UI에 액세스하기 위한 프라이빗 연결을 생성할 수 있습니다 AWS CLI.

VPC와 OpenSearch UI 간의 프라이빗 연결 생성(콘솔)

콘솔을 사용하여 VPC와 OpenSearch UI 간에 프라이빗 연결을 생성하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 왼쪽 탐색의 서버리스에서 VPC 엔드포인트를 선택합니다.
3. Create VPC endpoint(VPC 엔드포인트 생성)를 선택합니다.
4. 이름에 엔드포인트의 이름을 입력합니다.
5. VPC에서 OpenSearch UI 애플리케이션에 액세스할 VPC를 선택합니다.
6. 서브넷에서 OpenSearch UI 애플리케이션에 액세스할 서브넷을 하나 선택합니다.

> **Note**
>
> 엔드포인트의 IP 주소 및 DNS 유형은 서브넷 유형을 기반으로 합니다.
>
> - 듀얼 스택: 모든 서브넷에 IPv4 및 IPv6 주소 범위가 모두 있는 경우.
> - IPv6: 모든 서브넷이 IPv6 전용 서브넷인 경우.
> - IPv4: 모든 서브넷에 IPv4 주소 범위가 있는 경우.

7. 보안 그룹에서 엔드포인트 네트워크 인터페이스와 연결할 보안 그룹을 하나 이상 선택합니다.

> **Note**
>
> 이 단계에서는 엔드포인트에 권한을 부여하는 인바운드 트래픽의 포트, 프로토콜 및 소스를 제한합니다. 보안 그룹 규칙이 VPC 엔드포인트를 사용할 리소스가 OpenSearch UI 애플리케이션과 통신하여 엔드포인트 네트워크 인터페이스와도 통신할 수 있도록 허용하는지 확인합니다.

8. 8. Create endpoint(엔드포인트 생성)을 선택합니다.

VPC와 OpenSearch UI 간의 프라이빗 연결 생성(AWS CLI)

를 사용하여 VPC와 OpenSearch UI 간에 프라이빗 연결을 생성하려면 AWS CLI

다음 명령을 실행합니다. *## ###*를 자신의 정보로 바꿉니다.

```
aws opensearchserverless create-vpc-endpoint \
    --region region \
    --endpoint endpoint \
    --name vpc_endpoint_name \
    --vpc-id vpc_id \
    --subnet-ids subnet_ids
```

OpenSearch UI 애플리케이션에 대한 액세스를 허용하도록 VPC 엔드포인트 정책 업데이트

프라이빗 연결을 생성한 후 애플리케이션 ID를 지정하여 VPC 엔드포인트 정책에서 OpenSearch UI 애플리케이션에 대한 액세스를 허용하도록 VPC 엔드포인트 정책을 업데이트합니다.

VPC 엔드포인트 정책 업데이트에 대한 자세한 내용은 AWS PrivateLink 가이드의 VPC 엔드포인트 정책 업데이트를 참조하세요.

VPC 엔드포인트 정책에 다음 문이 포함되어 있는지 확인합니다. *## ### ##* 자신의 정보로 바꿉니다.

```
{
    "Statement": [{
        "Action": ["opensearch:*"],
        "Effect": "Allow",
        "Principal": "*",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "opensearch:ApplicationId": ["opensearch-ui-application-id"]
            }
        }
    }]
}
```

VPC 엔드포인트 정책에서 OpenSearch UI에 대한 액세스 취소

OpenSearch UI는 사용자가 VPC에서 애플리케이션에 액세스할 수 있도록 VPC 엔드포인트 정책에 명시적 권한이 필요합니다. 사용자가 더 이상 VPC에서 OpenSearch UI에 액세스하지 않도록 하려면 엔드포인트 정책에서 권한을 제거할 수 있습니다. 그런 다음 OpenSearch UI에 액세스하려고 하면 사용자에게 `403 forbidden` 오류 메시지가 표시됩니다.

VPC 엔드포인트 정책 업데이트에 대한 자세한 내용은 AWS PrivateLink 가이드의 VPC 엔드포인트 정책 업데이트를 참조하세요.

다음은 VPC에서 UI 애플리케이션에 대한 액세스를 거부하는 VPC 엔드포인트 정책의 예입니다.

```
{
    "Statement": [{
        "Action": ["opensearch:*"],
        "Effect": "Allow",
        "Principal": "*",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "opensearch:ApplicationId": [""]
            }
        }
    }]
}
```

OpenSearch UI 엔드포인트 및 할당량

AWS 서비스에 프로그래밍 방식으로 연결하려면 엔드포인트를 사용합니다. Service Quotas는 AWS 계정의 최대 서비스 리소스 또는 작업 수입니다.

Amazon OpenSearch UI는 OpenSearch Dashboards용 차세대 OpenSearch 사용자 인터페이스입니다. OpenSearch 대시보드에 액세스하기 위한 엔드포인트를 제공합니다. 이 주제를 사용하여 OpenSearch UI에 대한 서비스 엔드포인트 및 서비스 할당량을 찾습니다.

다른 OpenSearch 서비스에 대한 자세한 내용은 서비스 엔드포인트 및 할당량을 참조하세요.

OpenSearch UI 엔드포인트

OpenSearch UI는 다음 리전에서 사용할 수 있습니다.

지역명	지역	엔드포인트	프로토콜
아시아 태평양(뭄바이)	ap-south-1	opensearch.ap-south-1.amazonaws.com es.ap-south-1.amazonaws.com	HTTPS HTTPS
유럽(파리)	eu-west-3	opensearch.eu-west-3.amazonaws.com es.eu-west-3.amazonaws.com	HTTPS HTTPS
미국 동부(오하이오)	us-east-2	opensearch.us-east-2.amazonaws.com es.us-east-2.amazonaws.com	HTTPS HTTPS
유럽(아일랜드)	eu-west-1	opensearch.eu-west-1.amazonaws.com es.eu-west-1.amazonaws.com	HTTPS HTTPS
유럽(프랑크푸르트)	eu-central-1	opensearch.eu-central-1.amazonaws.com es.eu-central-1.amazonaws.com	HTTPS HTTPS
남아메리카(상파울루)	sa-east-1	opensearch.sa-east-1.amazonaws.com es.sa-east-1.amazonaws.com	HTTPS HTTPS
미국 동부(버지니아 북부)	us-east-1	opensearch.us-east-1.amazonaws.com es.us-east-1.amazonaws.com	HTTPS HTTPS
유럽(런던)	eu-west-2	opensearch.eu-west-2.amazonaws.com es.eu-west-2.amazonaws.com	HTTPS HTTPS

지역명	지역	엔드포인트	프로토콜
아시아 태평양(도쿄)	ap-northeast-1	opensearch.ap-northeast-1.amazonaws.com es.ap-northeast-1.amazonaws.com	HTTPS HTTPS
미국 서부(오레곤)	us-west-2	opensearch.us-west-2.amazonaws.com es.us-west-2.amazonaws.com	HTTPS HTTPS
아시아 태평양(싱가포르)	ap-southeast-1	opensearch.ap-southeast-1.amazonaws.com es.ap-southeast-1.amazonaws.com	HTTPS HTTPS
아시아 태평양(시드니)	ap-southeast-2	opensearch.ap-southeast-2.amazonaws.com es.ap-southeast-2.amazonaws.com	HTTPS HTTPS
캐나다(중부)	ca-central-1	opensearch.ca-central-1.amazonaws.com es.ca-central-1.amazonaws.com	HTTPS HTTPS
유럽(스톡홀름)	eu-north-1	opensearch.eu-north-1.amazonaws.com es.eu-north-1.amazonaws.com	HTTPS HTTPS
아시아 태평양(홍콩)	ap-east-1	opensearch.ap-east-1.amazonaws.com es.ap-east-1.amazonaws.com	HTTPS HTTPS
아시아 태평양(서울)	ap-northeast-2	opensearch.ap-northeast-2.amazonaws.com es.ap-northeast-2.amazonaws.com	HTTPS HTTPS

지역명	지역	엔드포인트	프로토콜
아시아 태평양(오사카)	ap-northeast-3	opensearch.ap-northeast-3.amazonaws.com es.ap-northeast-3.amazonaws.com	HTTPS HTTPS
아시아 태평양(하이데라바드)	ap-south-2	opensearch.ap-south-2.amazonaws.com es.ap-south-2.amazonaws.com	HTTPS HTTPS
유럽(스페인)	eu-south-2	opensearch.eu-south-2.amazonaws.com es.eu-south-2.amazonaws.com	HTTPS HTTPS
미국 서부(캘리포니아 북부)	us-west-1	opensearch.us-west-1.amazonaws.com es.us-west-1.amazonaws.com	HTTPS HTTPS
유럽(취리히)	eu-central-2	opensearch.eu-central-2.amazonaws.com es.eu-central-2.amazonaws.com	HTTPS HTTPS
유럽(밀라노)	eu-south-1	opensearch.eu-south-1.amazonaws.com es.eu-south-1.amazonaws.com	HTTPS HTTPS

OpenSearch UI 서비스 할당량

AWS 계정에는 OpenSearch UI 리소스와 관련된 다음과 같은 할당량이 있습니다.

명칭	기본값	조정 가능	Notes
리전별 계정당 OpenSearch UI 애플리케이션	30	예	리전별로 계정당 생성할 수 있는 최대 OpenSearch UI 애플리케이션 수입니다. 서비스 할당량을 사용하여 한도를 50으로 늘리고 자동으로 승인받을 수 있습니다. 더 높은 한도를 요청하려면 지원 티켓을 제출하세요.

Amazon OpenSearch Service에서 인덱스 관리

Amazon OpenSearch Service에 데이터를 추가한 후에는 해당 데이터를 다시 인덱싱하거나, 인덱스 별칭을 사용하여 작업하거나, 인덱스를 보다 비용 효율적인 스토리지로 이동하거나, 모두 삭제해야 하는 경우가 많습니다. 이 장에서는 UltraWarm 스토리지, 콜드 스토리지 및 인덱스 상태 관리에 대해 설명합니다. OpenSearch 인덱스 API에 대한 자세한 내용은 OpenSearch 설명서를 참조하세요.

주제

- Amazon OpenSearch Service를 위한 UltraWarm 스토리지
- Amazon OpenSearch Service용 콜드 스토리지
- Amazon OpenSearch Service용 OpenSearch 최적화 스토리지
- Amazon OpenSearch Service의 인덱스 상태 관리
- 인덱스 롤업을 사용하여 Amazon OpenSearch Service의 인덱스 요약
- Amazon OpenSearch Service에서 인덱스 변환
- Amazon OpenSearch Service의 클러스터 간 복제
- 원격 재인덱스를 사용하여 Amazon OpenSearch Service 인덱스 마이그레이션
- 데이터 스트림을 사용하여 Amazon OpenSearch Service에서 시계열 데이터 관리

Amazon OpenSearch Service를 위한 UltraWarm 스토리지

UltraWarm은 대량의 읽기 전용 데이터를 Amazon OpenSearch Service에 저장하는 비용 효율적인 방법을 제공합니다. 표준 데이터 노드는 각 노드에 연결된 Amazon EBS 볼륨 또는 인스턴스 스토어의 형태를 취하는 "핫" 스토리지를 사용합니다. 핫 스토리지의 새로운 데이터 인덱싱 및 검색 성능이 가장 빠릅니다.

UltraWarm 노드는 연결된 스토리지가 아닌 Amazon S3와 정교한 캐싱 솔루션을 사용하여 성능을 향상시킵니다. 활발하게 사용하지 않고 쿼리 빈도가 낮으며 동일한 성능이 필요하지 않은 인덱스의 경우, UltraWarm은 GiB당 데이터 비용이 아주 저렴합니다. 웜 인덱스는 핫 스토리지로 되돌리지 않는 한 읽기 전용이므로 UltraWarm은 로그와 같은 변경 불가능한 데이터에 가장 적합합니다.

OpenSearch에서, 웜 인덱스는 다른 인덱스와 마찬가지로 작동합니다. 동일한 API를 사용하여 쿼리하거나, OpenSearch Dashboards에 시각화를 만드는 데 사용할 수 있습니다.

주제

- 사전 조건
- UltraWarm 스토리지 요구 사항 및 성능 고려 사항
- UltraWarm 가격
- UltraWarm 활성화
- 인덱스를 UltraWarm 스토리지로 마이그레이션
- 마이그레이션 자동화
- 마이그레이션 조정
- 마이그레이션 취소
- 핫 인덱스 및 웜 인덱스 나열
- 핫 스토리지로 웜 인덱스 되돌리기
- 스냅샷에서 웜 인덱스 복원
- 웜 인덱스의 수동 스냅샷
- 콜드 스토리지로 웜 인덱스 마이그레이션
- KNN 인덱스 모범 사례
- UltraWarm 비활성화

사전 조건

UltraWarm에는 몇 가지 중요한 사전 조건이 있습니다.

- UltraWarm을 사용하려면 OpenSearch 또는 Elasticsearch 6.8 이상이 필요합니다.
- 웜 스토리지를 사용하려면 도메인에 전용 프라이머리 노드가 있어야 합니다.
- Multi-AZ with Standby 도메인을 사용하는 경우 웜 노드 수는 사용 중인 가용 영역 수의 배수여야 합니다.
- 도메인이 데이터 노드에 T2 또는 T3 인스턴스 유형을 사용하는 경우, 웜 스토리지를 사용할 수 없습니다.
- 인덱스가 대략적인 k-NN(`"index.knn":true`)을 사용하는 경우 버전 2.17 이상에서 웜 스토리지로 이동할 수 있습니다. 2.17 이전 버전의 도메인은이 기능을 사용하기 위해 2.17로 업그레이드할 수 있지만 2.x 이전 버전에서 생성된 KNN 인덱스는 UltraWarm으로 마이그레이션할 수 없습니다.
- 도메인이 세분화된 액세스 제어를 사용하는 경우 사용자는 OpenSearch Dashboards에서 `ultrawarm_manager` 역할에 매핑하여 UltraWarm API를 직접 호출해야 합니다.

> **Note**
>
> `ultrawarm_manager` 역할이 일부 기존 OpenSearch Service 도메인에 정의되지 않을 수 있습니다. Dashboards에 역할이 보이지 않으면 수동으로 생성해야 합니다.

권한 구성

기존 OpenSearch Service 도메인에서 UltraWarm을 활성화한 경우 `ultrawarm_manager` 역할이 도메인에 정의되어 있지 않을 수 있습니다. 관리자가 아닌 사용자는 이 역할에 매핑되어 세분화된 액세스 제어를 사용하는 도메인의 웜 인덱스를 관리해야 합니다. 수동으로 `ultrawarm_manager` 역할을 생성하려면 다음 단계를 수행합니다.

1. OpenSearch 대시보드에서 보안(Security)으로 이동하여 권한(Permissions)을 선택합니다.
2. 작업 그룹 생성(Create action group)을 선택하고 다음 그룹을 구성합니다.

그룹 이름	권한
`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` • `indices:admin/ultrawarm/migration/hot` • `indices:monitor/stats` • `indices:admin/ultrawarm/migration/cancel`

3. 역할(Roles)과 역할 생성(Create role)을 차례로 선택합니다.
4. 역할 이름을 ultrawarm_manager로 지정합니다.
5. 클러스터 권한(Cluster permissions)에서 `ultrawarm_cluster` 및 `cluster_monitor`를 선택합니다.
6. 인덱스(Index)에 *를 입력합니다.
7. 인덱스 권한(Index permissions)에서 `ultrawarm_index_read`, `ultrawarm_index_write`, `indices_monitor`를 선택합니다.

8. 생성(Create)을 선택합니다.
9. 역할을 생성한 후, UltraWarm 인덱스를 관리할 사용자 또는 백엔드 역할에 매핑합니다.

UltraWarm 스토리지 요구 사항 및 성능 고려 사항

the section called “스토리지 요구 사항 계산”에서 언급했듯이, 핫 스토리지의 데이터는 복제본, Linux 예약 공간 및 OpenSearch Service 예약 공간 등 상당한 오버헤드를 발생시킵니다. 예를 들어, 복제본 샤드가 1개인 20GiB 기본 샤드에는 약 58GiB의 핫 스토리지가 필요합니다.

UltraWarm은 Amazon S3를 사용하기 때문에 이러한 오버헤드를 발생시키지 않습니다. UltraWarm 스토리지 요구 사항을 계산할 때는 기본 샤드의 크기만 고려합니다. S3의 데이터 내구성 덕분에 복제본이 필요하지 않으며, S3는 운영 체제 또는 서비스 고려 사항을 추상화합니다. 동일한 20GiB 샤드에는 20GiB의 원 스토리지가 필요합니다. `ultrawarm1.large.search` 인스턴스를 프로비저닝하는 경우, 기본 샤드에 최대 스토리지 20TiB를 모두 사용할 수 있습니다. 인스턴스 유형 요약과 각 인스턴스 유형이 사용할 수 있는 최대 스토리지 용량은 the section called “UltraWarm 스토리지 할당량” 섹션을 참조하세요.

UltraWarm의 경우, 권장하는 최대 샤드 크기는 50GiB입니다. 각 UltraWarm 인스턴스 유형에 할당된 CPU 코어 수 및 RAM 용량은 동시에 검색 할 수 있는 샤드 수에 대한 정보를 제공합니다. 기본 샤드만 S3의 UltraWarm 스토리지에 포함되지만 OpenSearch Dashboards 및 `_cat/indices`는 여전히 UltraWarm 인덱스 크기를 모든 기본 및 복제 샤드의 총합으로 보고합니다.

예를 들어, 각 `ultrawarm1.medium.search` 인스턴스에는 2개의 CPU 코어가 있으며 S3에서 최대 1.5TiB 스토리지를 처리할 수 있습니다. 이러한 인스턴스 중 두 개에는 결합된 3TiB 스토리지가 있으며, 각 샤드가 50GiB인 경우 약 62개의 샤드로 작동합니다. 클러스터에 대한 요청이 이러한 샤드 중 네 개만 검색하는 경우 성능이 우수할 수 있습니다. 요청이 광범위하고 62개를 모두 검색하면 4개의 CPU 코어가 작업을 수행하는 데 어려움을 겪을 수 있습니다. 인스턴스가 워크로드를 처리하는 방법을 이해하기 위해 `WarmCPUUtilization` 및 `WarmJVMMemoryPressure` UltraWarm 지표를 모니터링합니다.

검색 범위가 넓거나 빈번한 경우 인덱스를 핫 스토리지에 남겨 두는 것이 좋습니다. 다른 OpenSearch 워크로드와 마찬가지로 요구에 맞는 UltraWarm을 판단하는 가장 중요한 단계는 실질적인 데이터 집합을 사용하여 대표적인 클라이언트 테스트를 수행하는 것입니다.

UltraWarm 가격

핫 스토리지를 사용하면 프로비저닝하는 만큼 비용을 지불합니다. 연결된 Amazon EBS 볼륨이 필요한 인스턴스가 있는가 하면, 인스턴스 스토어가 포함되어 있는 인스턴스도 있습니다. 스토리지가 비어 있든 가득 차 있든, 동일한 가격을 지불합니다.

UltraWarm 스토리지를 사용하면 사용한 만큼만 비용을 지불합니다. `ultrawarm1.large.search` 인스턴스는 S3에서 최대 20TiB의 스토리지를 처리할 수 있지만, 1TiB의 데이터만 저장하는 경우 1TiB의 데이터에 해당하는 비용만 청구됩니다. 다른 모든 노드 유형과 마찬가지로, 각 UltraWarm 노드별 시간당 요금만 지불합니다. 자세한 내용은 the section called “요금” 섹션을 참조하세요.

UltraWarm 활성화

콘솔은 웜 스토리지를 사용하는 도메인을 생성하는 가장 간단한 방법입니다. 도메인을 생성하는 동안 웜 데이터 노드 활성화와 원하는 웜 노드 수를 선택합니다. 사전 조건을 충족하는 경우 기존 도메인에서도 동일한 기본 프로세스가 적용됩니다. 도메인 상태가 처리 중(Processing)에서 활성(Active)으로 변경된 후에도 UltraWarm을 몇 시간 동안 사용하지 못할 수 있습니다.

다중 AZ를 대기 도메인과 함께 사용하는 경우 웜 노드 수는 가용 영역 수의 배수여야 합니다. 자세한 내용은 the section called “Multi-AZ with Standby” 단원을 참조하십시오.

UltraWarm을 활성화하는 AWS CLI 또는 구성 API, 특히 `ClusterConfig`의 `WarmEnabled`, `WarmCount`, `WarmType` 옵션을 사용할 수도 있습니다.

> **Note**
>
> 도메인은 최대 수의 웜 노드를 지원합니다. 자세한 내용은 the section called “할당량” 섹션을 참조하세요.

CLI 명령 예

다음 AWS CLI 명령은 데이터 노드 3개, 전용 마스터 노드 3개, 웜 노드 6개, 세분화된 액세스 제어가 활성화된 도메인을 생성합니다.

```
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
```

자세한 내용은 AWS CLI 명령 참조를 참조하세요.

샘플 구성 API 요청

구성 API에 대한 다음 요청은 3개의 데이터 노드, 3개의 전용 프라이머리 노드, 세분화된 액세스 제어가 활성화되어 있고 제한적인 액세스 정책이 있는 6개의 웜 노드로 도메인을 생성합니다.

```
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/*\"}]}"
}
```

자세한 내용은 Amazon OpenSearch Service 참조를 참조하세요.

인덱스를 UltraWarm 스토리지로 마이그레이션

인덱스에 쓰기를 완료했으며 더 이상 가장 빠른 검색 성능이 필요하지 않은 경우 핫에서 UltraWarm으로 마이그레이션합니다.

```
POST _ultrawarm/migration/my-index/_warm
```

그런 다음 마이그레이션 상태를 확인합니다.

```
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
```

```
    }
  }
}
```

마이그레이션을 수행하려면 인덱스 상태가 녹색이어야 합니다. 여러 인덱스를 빠르게 연속해서 마이그레이션하는 경우 _cat API와 비슷한 일반 텍스트로 모든 마이그레이션에 대한 요약을 얻을 수 있습니다.

```
GET _ultrawarm/migration/_status?v

index    migration_type state
my-index HOT_TO_WARM    RUNNING_SHARD_RELOCATION
```

OpenSearch Service는 한 번에 하나의 인덱스를 UltraWarm으로 마이그레이션합니다. 대기열에 최대 200번의 마이그레이션이 있을 수 있습니다. 한도를 초과하는 요청은 거부됩니다. 현재 대기열의 마이그레이션 번호를 확인하려면 HotToWarmMigrationQueueSize 지표를 모니터링합니다. 인덱스는 마이그레이션 프로세스 전반에 걸쳐 계속 사용할 수 있으며 가동 중지 없이 사용할 수 있습니다.

마이그레이션 프로세스의 상태는 다음과 같습니다.

```
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
```

이러한 상태가 나타내듯이 스냅샷, 샤드 재배치 또는 강제 병합 중에 마이그레이션이 실패할 수 있습니다. 스냅샷 또는 샤드 재배치 중 실패는 일반적으로 노드 오류 또는 S3 연결 문제로 인해 발생합니다. 일반적으로 디스크 공간 부족이 강제 병합 실패의 근본 원인입니다.

마이그레이션이 완료되면 동일한 _status 요청이 오류를 반환합니다. 이때 인덱스를 확인하면 웜 인덱스만의 고유한 몇 가지 설정을 볼 수 있습니다.

```
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"
          }
        }
      }
    }
  }
}
```

- 이 경우 number_of_replicas는 디스크 공간을 소비하지 않는 수동 복제본의 수입니다.
- routing.allocation.require.box_type은 인덱스가 표준 데이터 노드가 아닌 웜 노드를 사용하도록 지정합니다.
- merge.policy.max_merge_at_once_explicit는 마이그레이션 중에 동시에 병합할 세그먼트 수를 지정합니다.

웜 스토리지의 인덱스는 핫 스토리지로 되돌리지 않는 한 읽기 전용이므로 UltraWarm은 로그와 같은 변경 불가능한 데이터에 가장 적합합니다. 인덱스를 쿼리하여 삭제할 수 있지만 개별 문서를 추가, 업데이트 또는 삭제할 수 없습니다. 시도하는 경우 오류가 발생할 수 있습니다.

```
{
  "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
}
```

마이그레이션 자동화

인덱스가 특정 기간에 도달하거나 다른 조건을 충족한 후에는 the section called “인덱스 상태 관리”을 사용하여 마이그레이션 프로세스를 자동화하는 것이 좋습니다. 이 워크플로를 보여주는 샘플 정책을 참조하세요.

마이그레이션 조정

UltraWarm 스토리지로의 인덱스 마이그레이션에는 강제 병합이 필요합니다. 각 OpenSearch 인덱스는 몇 개의 샤드로 구성되며 각 샤드는 몇 개의 Lucene 세그먼트로 구성됩니다. 강제 병합 작업은 삭제하도록 표시된 문서를 소거하고 디스크 공간을 절약합니다. 기본적으로 UltraWarm은 인덱스를 하나의 세그먼트로 병합합니다. 단, 기본값 20이 사용되는 kNN 인덱스는 예외입니다.

`index.ultrawarm.migration.force_merge.max_num_segments` 설정을 사용하여 이 값을 최대 1,000개의 세그먼트까지 변경할 수 있습니다. 값이 높을수록 마이그레이션 프로세스 속도가 빨라지지만 마이그레이션이 완료된 후 웜 인덱스에 대한 쿼리 대기 시간이 늘어납니다. 설정을 변경하려면 다음과 같이 요청합니다.

```
PUT my-index/_settings
{
```

```
  "index": {
    "ultrawarm": {
      "migration": {
        "force_merge": {
          "max_num_segments": 1
        }
      }
    }
  }
}
```

마이그레이션 프로세스의 이 단계에 걸리는 시간을 확인하려면HotToWarmMigrationForceMergeLatency 지표를 모니터링합니다.

마이그레이션 취소

UltraWarm은 대기열에서 마이그레이션을 순차적으로 처리합니다. 마이그레이션이 대기열에 있지만 아직 시작되지 않은 경우 다음 요청을 사용하여 대기열에서 제거할 수 있습니다.

```
POST _ultrawarm/migration/_cancel/my-index
```

도메인에서 세분화된 액세스 제어를 사용하는 경우 이 요청을 하기 위해 `indices:admin/ultrawarm/migration/cancel` 권한이 필요합니다.

핫 인덱스 및 웜 인덱스 나열

UltraWarm에는 핫 인덱스와 웜 인덱스를 관리하는 데 도움이 되는, `_all`과 비슷한 두 가지 옵션이 추가되었습니다. 모든 웜 또는 핫 인덱스 목록을 보려면 다음과 같이 요청합니다.

```
GET _warm
GET _hot
```

인덱스를 지정하는 다른 요청에서 이러한 옵션을 사용할 수 있습니다. 예를 들면 다음과 같습니다.

```
_cat/indices/_warm
_cluster/state/_all/_hot
```

핫 스토리지로 웜 인덱스 되돌리기

인덱스에 다시 기록해야 하는 경우 핫 스토리지로 다시 마이그레이션합니다.

```
POST _ultrawarm/migration/my-index/_hot
```

웜 스토리지에서 핫 스토리지로 최대 10번의 마이그레이션(대기열에 입력됨)을 수행할 수 있습니다. OpenSearch Service는 마이그레이션 요청을 대기열에 입력된 순서대로 한 번에 하나씩 처리합니다. 현재 번호를 확인하려면 `WarmToHotMigrationQueueSize` 지표를 모니터링합니다.

마이그레이션을 완료한 후 인덱스 설정을 검토하여 요구 사항을 충족하는지 확인합니다. 인덱스가 하나의 복제본이 있는 핫 스토리지로 돌아갑니다.

스냅샷에서 웜 인덱스 복원

UltraWarm에는 자동 스냅샷용 표준 리포지토리 외에 웜 인덱스용 보조 리포지토리인 `cs-ultrawarm`이 추가됩니다. 이 리포지토리의 각 스냅샷에는 하나의 인덱스만 포함됩니다. 웜 인덱스를 삭제하면 해당 스냅샷은 다른 자동 스냅샷과 마찬가지로 14일 동안 `cs-ultrawarm` 리포지토리에 남아 있습니다.

`cs-ultrawarm`에서 스냅샷을 복원하면 핫 스토리지가 아닌 웜 스토리지로 복원됩니다. `cs-automated` 및 `cs-automated-enc` 리포지토리의 스냅샷은 핫 스토리지로 복원됩니다.

UltraWarm 스냅샷을 웜 스토리지로 복원하려면

1. 복원할 인덱스가 포함된 최신 스냅샷을 식별합니다.

```
GET _snapshot/cs-ultrawarm/_all?verbose=false

{
  "snapshots": [{
    "snapshot": "snapshot-name",
    "version": "1.0",
    "indices": [
      "my-index"
    ]
  }]
}
```

> **Note**
>
> 기본적으로 `GET _snapshot/<repo>` 작업에는 리포지토리 내 각 스냅샷의 시작 시간, 종료 시간, 기간과 같은 자세한 데이터 정보가 표시됩니다. 이 `GET _snapshot/<repo>` 작업은 리포지토리에 포함된 각 스냅샷의 파일에서 정보를 검색합니다. 시작 시간, 종료

시간 및 기간이 필요하지 않고 스냅샷의 이름 및 인덱스 정보만 필요한 경우 스냅샷을 나열할 때 `verbose=false` 파라미터를 사용하여 처리 시간을 최소화하고 시간 초과를 방지하는 것이 좋습니다.

2. 인덱스가 이미 있는 경우 삭제합니다.

```
DELETE my-index
```

인덱스를 삭제하지 않으려면 핫 스토리지로 돌아가 재인덱스합니다.

3. 스냅샷을 복원합니다.

```
POST _snapshot/cs-ultrawarm/snapshot-name/_restore
```

UltraWarm은 이 복원 요청에서 지정한 인덱스 설정을 무시하지만 `rename_pattern` 및 `rename_replacement`와 같은 옵션을 지정할 수 있습니다. OpenSearch 스냅샷 복원 옵션에 대한 요약은 OpenSearch 설명서를 참조하세요.

웜 인덱스의 수동 스냅샷

웜 인덱스의 수동 스냅 샷을 생성할 수 있지만 권장하지 않습니다. 마이그레이션 중에 생성한 각 웜 인덱스에 대한 스냅샷이 추가 비용 없이 자동 `cs-ultrawarm` 리포지토리에 이미 포함되어 있습니다.

기본적으로 OpenSearch Service는 수동 스냅샷에 웜 인덱스를 포함하지 않습니다. 예를 들어, 다음 호출에는 핫 인덱스만 포함됩니다.

```
PUT _snapshot/my-repository/my-snapshot
```

웜 인덱스의 수동 스냅샷을 생성하도록 선택하면 몇 가지 중요한 고려 사항이 적용됩니다.

- 핫 인덱스와 웜 인덱스를 혼합할 수 없습니다. 예를 들어 다음 요청은 실패합니다.

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1,hot-index-1",
  "include_global_state": false
}
```

핫 인덱스와 웜 인덱스의 혼합을 포함하는 경우, 와일드카드(*) 문도 실패합니다.

- 스냅샷당 하나의 웜 인덱스만 포함할 수 있습니다. 예를 들어 다음 요청은 실패합니다.

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1,warm-index-2,other-warm-indices-*",
  "include_global_state": false
}
```

  이 요청이 성공한 경우:

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1",
  "include_global_state": false
}
```

- 수동 스냅샷은 원래 웜 인덱스가 포함된 경우에도 항상 핫 스토리지로 복원합니다.

콜드 스토리지로 웜 인덱스 마이그레이션

자주 쿼리하지 않는 UltraWarm에 데이터가 있는 경우 콜드 스토리지로 마이그레이션하는 것이 좋습니다. 콜드 스토리지는 가끔 액세스하거나 더 이상 사용하지 않는 데이터를 위한 것입니다. 콜드 인덱스에서 읽거나 쓸 수는 없지만 쿼리해야 할 때마다 무료로 웜 스토리지로 다시 마이그레이션 할 수 있습니다. 지침은 인덱스를 콜드 스토리지로 마이그레이션을 참조하세요.

KNN 인덱스 모범 사례

- Ultrawarm/Cold 티어는 모든 KNN 인덱스 엔진 유형에 사용할 수 있습니다. 그래프 데이터를 오프 힙 메모리에 완전히 로드할 필요가 없는 Lucene 엔진 및 디스크 최적화 벡터 검색을 사용하는 KNN 인덱스의 경우 권장됩니다. FAISS 및 NMSLIB와 같은 네이티브 인 메모리 엔진과 함께 사용하는 경우 적극적으로 검색할 샤드 그래프 크기를 고려하고 UltraWarm 인스턴스를 프로비저닝해야 하며, 이에 따라 가급적 `uw.large` 인스턴스 유형을 프로비저닝해야 합니다. 예를 들어 고객이 2개의 `uw.large` 인스턴스를 구성한 경우 각각 약 `knn.memory.circuit_breaker.limit * 61` GiB의 힙 외부 메모리를 사용할 수 있습니다. 모든 웜 쿼리가 누적 그래프 크기가 사용 가능한 오프 힙 메모리를 초과하지 않는 샤드를 대상으로 하는 경우 최적의 성능을 얻을 수 있습니다. 사용 가능한 메모리가 그래프를 로드하는 데 필요한 것보다 낮으면 제거되고 힙 외부 메모리를 사용할 수 있을 때까지 기다리기 때문에 지연 시간이 영향을 받습니다. 따라서 인 메모리 엔진이 사용되는 사용 사례나 엔진에 관계없이 더 높은 검색 처리량 사례에서는 `uw.medium` 인스턴스를 사용하지 않는 것이 좋습니다.

- UltraWarm으로 마이그레이션하는 KNN 인덱스는 단일 세그먼트로 강제 병합되지 않습니다. 이렇게 하면 그래프 크기가 인 메모리 엔진에 비해 너무 커지기 때문에 OOM 문제로 실행되는 핫 노드와 웜 노드에 영향을 주지 않습니다. 샤드당 세그먼트 수가 증가하여 로컬 캐시 공간을 더 많이 사용하고 웜 티어로 마이그레이션할 인덱스 수가 줄어들 수 있습니다. 기존 설정을 사용하고 인덱스를 웜 티어로 마이그레이션하기 전에 재정의하여 인덱스를 단일 세그먼트로 강제 병합하도록 선택할 수 있습니다. 자세한 내용은 the section called “마이그레이션 조정” 단원을 참조하십시오.
- 인덱스가 자주 검색되지 않고 지연 시간에 민감한 워크로드를 제공하지 않는 사용 사례가 있는 경우 해당 인덱스를 UltraWarm 계층으로 마이그레이션하도록 선택할 수 있습니다. 이렇게 하면 핫 티어 컴퓨팅 인스턴스를 스케일 다운하고 UltraWarm 티어 컴퓨팅이 이러한 낮은 우선 순위 인덱스에 대한 쿼리를 처리할 수 있습니다. 또한 우선 순위가 낮은 인덱스와 높은 인덱스의 쿼리 간에 소비되는 리소스를 격리하여 서로 영향을 미치지 않도록 하는 데 도움이 될 수 있습니다.

UltraWarm 비활성화

콘솔은 UltraWarm을 비활성화하는 가장 간단한 방법입니다. 도메인을 선택하고 [작업(Actions)], [클러스터 구성 편집(Edit cluster configuration)]을 선택합니다. 웜 데이터 노드 활성화를 선택 취소하고 변경 사항 저장을 선택합니다. AWS CLI 및 구성 API에서 `WarmEnabled` 옵션을 사용할 수도 있습니다.

UltraWarm을 사용 중지하기 전에 모든 웜 인덱스를 삭제하거나 핫 스토리지로 다시 마이그레이션해야 합니다. 웜 스토리지가 비어 있으면 5분 정도 기다린 후 UltraWarm을 사용 중지합니다.

Amazon OpenSearch Service용 콜드 스토리지

콜드 스토리지를 사용하면 Amazon OpenSearch Service 도메인에 액세스 빈도가 낮은 데이터 또는 기록 데이터를 저장하고 필요에 따라 다른 스토리지 계층보다 저렴한 비용으로 분석할 수 있습니다. 콜드 스토리지는 오래된 데이터에 대한 정기적인 연구 또는 포렌식 분석을 수행해야 하는 경우에 적합합니다. 콜드 스토리지에 적합한 데이터의 실용적인 예로는 액세스 빈도가 낮은 로그, 규정 준수 요구 사항을 충족하기 위해 보존해야 하는 데이터 또는 기록 가치가 있는 로그가 있습니다.

UltraWarm스토리지와 마찬가지로 콜드 스토리지는 Amazon S3 에서 지원됩니다. 콜드 데이터를 쿼리해야 하는 경우 기존 UltraWarm 노드에 선택적으로 연결할 수 있습니다. 수동으로 또는 인덱스 상태 관리 정책을 사용하여 콜드 데이터의 마이그레이션 및 수명 주기를 관리할 수 있습니다.

주제

- 사전 조건
- 콜드 스토리지 요구 사항 및 성능 고려 사항

- 콜드 스토리지 요금
- 콜드 스토리지 활성화
- OpenSearch Dashboards의 콜드 인덱스 관리
- 콜드 스토리지로 인덱스 마이그레이션
- 콜드 스토리지로 마이그레이션 자동화
- 콜드 스토리지로의 마이그레이션 취소
- 콜드 인덱스 목록 표시
- 웜 스토리지로 콜드 인덱스 마이그레이션
- 스냅샷에서 콜드 인덱스 복원
- 콜드 스토리지에서 웜 스토리지로의 마이그레이션 취소
- 콜드 인덱스 메타데이터 업데이트
- 콜드 인덱스 삭제
- 콜드 스토리지 비활성화

사전 조건

콜드 스토리지에는 다음과 같은 사전 요구 사항이 있습니다.

- 콜드 스토리지를 사용하려면 OpenSearch 또는 Elasticsearch 버전 7.9 이상이 필요합니다.
- OpenSearch Service 도메인에서 콜드 스토리지를 활성화하려면 동일한 도메인에서 웜 스토리지도 활성화해야 합니다.
- 콜드 스토리지를 사용하려면 도메인에 전용 프라이머리 노드가 있어야 합니다.
- 도메인에서 데이터 노드에 T2 또는 T3 인스턴스 유형을 사용하는 경우, 콜드 스토리지를 사용할 수 없습니다.
- 인덱스가 대략적인 k-NN(`"index.knn":true`)을 사용하는 경우 버전 2.17 이상에서 콜드 스토리지로 이동할 수 있습니다. 2.17 이전 버전의 도메인은이 기능을 사용하기 위해 2.17로 업그레이드할 수 있지만 2.x 이전 버전에서 생성된 KNN 인덱스는 콜드로 마이그레이션할 수 없습니다.
- 도메인에서 세분화된 액세스 제어를 사용하는 경우 관리자가 아닌 사용자는 OpenSearch Dashboards에서 `cold_manager` 역할에 매핑되어 콜드 인덱스를 관리해야 합니다.

> **Note**
>
> `cold_manager` 역할이 일부 기존 OpenSearch Service 도메인에 존재하지 않을 수 있습니다. Dashboards에 역할이 보이지 않으면 수동으로 생성해야 합니다.

권한 구성

기존 OpenSearch Service 도메인에서 콜드 스토리지를 활성화한 경우 `cold_manager` 역할이 도메인에 정의되어 있지 않을 수 있습니다. 도메인에서 세분화된 액세스 제어를 사용하는 경우 관리자가 아닌 사용자는 이 역할에 매핑되어 콜드 인덱스를 관리해야 합니다. 수동으로 `cold_manager` 역할을 생성하려면 다음 단계를 수행합니다.

1. OpenSearch 대시보드에서 보안(Security)으로 이동하여 권한(Permissions)을 선택합니다.
2. 작업 그룹 생성(Create action group)을 선택하고 다음 그룹을 구성합니다.

그룹 이름	권한
`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. 역할(Roles)과 역할 생성(Create role)을 차례로 선택합니다.
4. 역할 이름을 cold_manager로 지정합니다.
5. 클러스터 권한(Cluster permissions)의 경우 생성한 `cold_cluster` 그룹을 선택합니다.
6. 인덱스(Index)에 *를 입력합니다.
7. 인덱스 권한(Index permissions)의 경우 생성한 `cold_index` 그룹을 선택합니다.
8. 생성(Create)을 선택합니다.
9. 역할을 생성한 후, 콜드 인덱스를 관리하는 모든 사용자 또는 백엔드 역할에 매핑합니다.

콜드 스토리지 요구 사항 및 성능 고려 사항

콜드 스토리지는 Amazon S3를 사용하기 때문에 복제본, Linux 예약 공간 및 OpenSearch Service 예약 공간과 같은 핫 스토리지의 오버헤드가 발생하지 않습니다. 콜드 스토리지에는 컴퓨팅 용량이 연결되어 있지 않기 때문에 특정 인스턴스 유형이 없습니다. 콜드 스토리지에 원하는 양의 데이터를 저장할 수 있습니다. Amazon CloudWatch에서 `ColdStorageSpaceUtilization` 지표를 모니터링하여 사용 중인 콜드 스토리지 공간의 양을 확인할 수 있습니다.

콜드 스토리지 요금

UltraWarm 스토리지와 마찬가지로 콜드 스토리지를 사용하면 데이터 스토리지에 대해서만 비용을 지불합니다. 콜드 데이터에 대한 컴퓨팅 비용이 없으며 콜드 스토리지에 데이터가 없는 경우 요금이 청구되지 않습니다.

콜드 스토리지와 웜 스토리지 간에 데이터를 이동할 때 전송 요금이 발생하지 않습니다. 웜 스토리지와 콜드 스토리지 간에 인덱스를 마이그레이션하는 동안에는 인덱스 복사본 하나에 대해서만 비용을 계속 지불합니다. 마이그레이션이 완료되면 마이그레이션된 스토리지 계층에 따라 인덱스가 청구됩니다. 콜드 스토리지 요금에 대한 자세한 내용은 Amazon OpenSearch Service 요금을 참조하세요.

콜드 스토리지 활성화

콘솔은 콜드 스토리지를 사용하는 도메인을 생성하는 가장 간단한 방법입니다. 도메인을 생성하는 동안 먼저 웜 데이터 노드 활성화를 선택합니다. 동일한 도메인에서 웜 스토리지를 활성화해야 하기 때문입니다. 그런 다음 콜드 스토리지 활성화를 선택합니다.

사전 조건을 충족하는 경우 기존 도메인에서도 동일한 프로세스가 적용됩니다. 도메인 상태가 처리 중(Processing)에서 활성(Active)으로 변경된 후에도 콜드 스토리지를 몇 시간 동안 사용하지 못할 수 있습니다.

AWS CLI 또는 구성 API를 사용하여 콜드 스토리지를 활성화할 수도 있습니다.

CLI 명령 예

다음 AWS CLI 명령은 데이터 노드 3개, 전용 마스터 노드 3개, 콜드 스토리지 활성화 및 세분화된 액세스 제어가 활성화된 도메인을 생성합니다.

```
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
```

자세한 내용은 AWS CLI 명령 참조를 참조하세요.

샘플 구성 API 요청

구성 API에 대한 다음 요청은 3개의 데이터 노드, 3개의 전용 프라이머리 노드, 활성화된 콜드 스토리지 및 세분화된 액세스 제어가 활성화되어 있는 도메인을 생성합니다.

```
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"
}
```

자세한 내용은 Amazon OpenSearch Service 참조를 참조하세요.

OpenSearch Dashboards의 콜드 인덱스 관리

OpenSearch Service 도메인의 기존 Dashboards 인터페이스를 사용하여 핫 인덱스, 웜 인덱스 및 콜드 인덱스를 관리할 수 있습니다. Dashboards를 사용하면 CLI 또는 구성 API를 사용하지 않고도 웜 스토리지와 콜드 스토리지 간에 인덱스를 마이그레이션하고 인덱스 마이그레이션 상태를 모니터링할 수 있습니다. 자세한 내용은 OpenSearch Dashboards의 인덱스 관리를 참조하세요.

콜드 스토리지로 인덱스 마이그레이션

콜드 스토리지로 인덱스를 마이그레이션하는 경우 데이터를 더욱 쉽게 검색할 수 있도록 시간 범위를 제공합니다. 인덱스의 데이터를 기반으로 타임스탬프 필드를 선택하거나, 시작 및 종료 타임스탬프를 수동으로 제공하거나, 타임스탬프를 지정하지 않도록 선택할 수 있습니다.

파라미터	지원되는 값	설명
`timestamp_field`	인덱스 매핑의 날짜/시간 필드입니다.	제공된 필드의 최솟값과 최댓값이 계산되어 콜드 인덱

파라미터	지원되는 값	설명
		스에 대한 start_time 및 end_time 메타데이터로 저장됩니다.
start_time 및 end_time	다음 형식 중 하나: • strict_date_optional_time. 예: yyyy-MM-dd'T'HH:mm:ss.SSSZ 또는 yyyy-MM-dd • Epoch 시간(밀리초)	제공된 값은 콜드 인덱스에 대한 start_time 및 end_time 메타데이터로 저장됩니다.

타임 스탬프를 지정하지 않으려면 대신 ?ignore=timestamp를 요청에 추가합니다.

다음 요청은 웜 인덱스를 콜드 스토리지로 마이그레이션하고 해당 인덱스의 데이터에 대한 시작 및 종료 시간을 제공합니다.

```
POST _ultrawarm/migration/my-index/_cold
  {
    "start_time": "2020-03-09",
    "end_time": "2020-03-09T23:00:00Z"
  }
```

그런 다음 마이그레이션 상태를 확인합니다.

```
GET _ultrawarm/migration/my-index/_status

{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_METADATA_RELOCATION",
    "migration_type": "WARM_TO_COLD"
  }
}
```

OpenSearch Service는 한 번에 하나의 인덱스를 콜드 스토리지로 마이그레이션합니다. 대기열에 최대 100번의 마이그레이션이 있을 수 있습니다. 한도를 초과하는 요청은 거부됩니다. 현재 대기열의 마이

그레이션 번호를 확인하려면 WarmToColdMigrationQueueSize 지표를 모니터링합니다. 마이그레이션 프로세스의 상태는 다음과 같습니다.

```
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.
```

콜드 스토리지로 마이그레이션 자동화

인덱스가 특정 기간에 도달하거나 다른 조건을 충족한 후에는 인덱스 상태 관리를 사용하여 마이그레이션 프로세스를 자동화할 수 있습니다. 샘플 정책을 참조하세요. 핫 스토리지에서 UltraWarm, 콜드 스토리지로 인덱스를 자동 마이그레이션하는 방법을 보여줍니다.

> **Note**
>
> 명시적 timestamp_field은(는) 인덱스 상태 관리 정책을 사용하여 인덱스를 콜드 스토리지로 이동하는 데 필요합니다.

콜드 스토리지로의 마이그레이션 취소

콜드 스토리지로의 마이그레이션이 대기 중이거나 실패 상태인 경우 다음 요청을 사용하여 마이그레이션을 취소할 수 있습니다.

```
POST _ultrawarm/migration/_cancel/my-index

{
  "acknowledged" : true
}
```

도메인에서 세분화된 액세스 제어를 사용하는 경우 이 요청을 하기 위해 indices:admin/ultrawarm/migration/cancel 권한이 필요합니다.

콜드 인덱스 목록 표시

쿼리하기 전에 콜드 스토리지에 인덱스를 나열하여 추가 분석을 위해 UltraWarm으로 마이그레이션할 인덱스를 결정할 수 있습니다. 다음 요청에는 인덱스 이름별로 정렬된 모든 콜드 인덱스가 나열됩니다.

```
GET _cold/indices/_search
```

샘플 응답

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

필터링

접두사 기반 인덱스 패턴 및 시간 범위 오프셋을 기반으로 콜드 인덱스를 필터링할 수 있습니다.

다음 요청은 `event-*`의 접두사 패턴과 일치하는 인덱스를 나열합니다.

```
GET _cold/indices/_search
 {
   "filters":{
      "index_pattern": "event-*"
   }
 }
```

샘플 응답

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

다음 요청은 `2019-03-01~2020-03-01`의 `start_time` 및 `end_time` 메타데이터 필드를 사용하여 인덱스를 반환합니다.

```
GET _cold/indices/_search
{
  "filters": {
    "time_range": {
      "start_time": "2019-03-01",
      "end_time": "2020-03-01"
    }
  }
}
```

샘플 응답

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

정렬

인덱스 이름이나 크기와 같은 메타데이터 필드별로 콜드 인덱스를 정렬할 수 있습니다. 다음 요청은 크기별로 정렬된 모든 인덱스를 내림차순으로 나열합니다.

```
GET _cold/indices/_search
 {
 "sort_key": "size:desc"
 }
```

샘플 응답

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

다른 유효한 정렬 키는`start_time:asc/desc`, `end_time:asc/desc`, `index_name:asc/desc`입니다.

페이지 매김

콜드 인덱스 목록을 페이지 매김할 수 있습니다. `page_size` 파라미터를 사용해 페이지당 반환될 인덱스 수를 구성합니다(기본값은 10). 콜드 인덱스에 대한 모든 `_search` 요청은 후속 호출에 사용할 수 있는 `pagination_id`을(를) 반환합니다.

다음 요청은 콜드 인덱스의 `_search` 요청 결과를 페이지 매김하고 다음 100개의 결과를 표시합니다.

```
GET _cold/indices/_search?page_size=100
{
"pagination_id": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
}
```

웜 스토리지로 콜드 인덱스 마이그레이션

이전 섹션의 필터링 기준을 적용해 콜드 인덱스 목록 범위를 좁힌 후, 데이터를 쿼리하고 이를 사용해 시각화를 생성할 수 있는 UltraWarm으로 다시 마이그레이션합니다.

다음 요청은 두 콜드 인덱스를 다시 웜 스토리지로 마이그레이션합니다.

```
POST _cold/migration/_warm
```

```
 {
 "indices": "my-index1,my-index2"
 }


{
  "acknowledged" : true
}
```

마이그레이션 상태를 확인하고 마이그레이션 ID를 검색하려면 다음 요청을 보냅니다.

```
GET _cold/migration/_status
```

샘플 응답

```
{
  "cold_to_warm_migration_status" : [
    {
      "migration_id" : "tyLjXCA-S76zPQbPVHkOKA",
      "indices" : [
        "my-index1,my-index2"
      ],
      "state" : "RUNNING_INDEX_CREATION"
    }
  ]
}
```

인덱스 관련 마이그레이션 정보를 가져오려면 인덱스 이름을 포함하세요.

```
GET _cold/migration/my-index/_status
```

인덱스를 지정하는 대신 현재 마이그레이션 상태별로 인덱스를 나열할 수 있습니다. 유효한 값은 `_failed`, `_accepted`, `_all`입니다.

다음 명령은 단일 마이그레이션 요청에서 모든 인덱스의 상태를 가져옵니다.

```
GET _cold/migration/_status?migration_id=my-migration-id
```

상태 요청을 사용하여 마이그레이션 ID를 검색합니다. 자세한 마이그레이션 정보를 보려면 `&verbose=true`를 추가합니다.

최대 100개의 지표를 동시에 마이그레이션하여 콜드 스토리지에서 웜 스토리지로 인덱스를 10개 이하의 배치에 마이그레이션할 수 있습니다. 한도를 초과하는 요청은 거부됩니다. 현재 수행하고 있는 마이그레이션 번호를 확인하려면 `ColdToWarmMigrationQueueSize` 지표를 모니터링합니다. 마이그레이션 프로세스의 상태는 다음과 같습니다.

```
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.
```

스냅샷에서 콜드 인덱스 복원

삭제된 콜드 인덱스를 복원해야 하는 경우 the section called “스냅샷에서 웜 인덱스 복원”의 지침에 따라 다시 웜 티어로 복원한 다음 인덱스를 다시 콜드 티어로 마이그레이션하면 됩니다. 삭제된 콜드 인덱스를 콜드 티어에 직접 복원할 수는 없습니다. OpenSearch Service는 콜드 인덱스를 삭제한 후 14일 동안 유지합니다.

콜드 스토리지에서 웜 스토리지로의 마이그레이션 취소

콜드 스토리지에서 웜 스토리지로의 인덱스 마이그레이션이 대기 중이거나 실패 상태인 경우 다음 요청으로 취소할 수 있습니다.

```
POST _cold/migration/my-index/_cancel

{
  "acknowledged" : true
}
```

인덱스 배치에 대한 마이그레이션을 취소하려면(한 번에 최대 10개) 마이그레이션 ID를 지정합니다.

```
POST _cold/migration/_cancel?migration_id=my-migration-id

{
  "acknowledged" : true
}
```

상태 요청을 사용하여 마이그레이션 ID를 검색합니다.

콜드 인덱스 메타데이터 업데이트

콜드 인덱스에 대한 `start_time` 및 `end_time` 필드를 업데이트할 수 있습니다.

```
PATCH _cold/my-index
 {
 "start_time": "2020-01-01",
 "end_time": "2020-02-01"
 }
```

콜드 스토리지에 있는 인덱스의 `timestamp_field`를 업데이트할 수 없습니다.

> **Note**
>
> OpenSearch Dashboards는 PATCH 메서드를 지원하지 않습니다. curl, Postman 또는 다른 메서드를 사용하여 콜드 메타데이터를 업데이트합니다.

콜드 인덱스 삭제

ISM 정책을 사용하지 않는 경우 콜드 인덱스를 수동으로 삭제할 수 있습니다. 다음 요청은 콜드 인덱스를 삭제합니다.

```
DELETE _cold/my-index

{
  "acknowledged" : true
}
```

콜드 스토리지 비활성화

OpenSearch Service 콘솔은 콜드 스토리지를 비활성화하는 가장 간단한 방법입니다. 도메인을 선택하고 [작업(Actions)], [클러스터 구성 편집(Edit cluster configuration)]을 선택한 다음 [콜드 스토리지 사용(Enable cold storage)]을 선택 취소합니다.

AWS CLI 또는 구성 API를 사용하려면 아래에서를 `ColdStorageOptions`설정합니다`"Enabled"="false"`.

콜드 스토리지를 비활성화하기 전에 모든 콜드 인덱스를 삭제하거나 웜 스토리지로 다시 마이그레이션해야 합니다. 그렇지 않으면 비활성화 작업이 실패합니다.

Amazon OpenSearch Service용 OpenSearch 최적화 스토리지

Amazon OpenSearch Service용 OpenSearch 최적화 인스턴스 패밀리는 대용량 데이터를 저장하는 비용 효율적인 솔루션입니다. OR1 인스턴스가 있는 도메인은 Amazon Elastic Block Store(Amazon EBS) gp3 또는 io1 볼륨을 기본 스토리지로 사용하며, 데이터가 도착하면 Amazon S3에 동기적으로 복사됩니다. 이 스토리지 구조는 향상된 인덱싱 처리량과 높은 내구성을 제공합니다. OpenSearch 최적화 인스턴스 패밀리는 장애 발생 시 자동 데이터 복구도 지원합니다. OR1 인스턴스 유형 옵션에 대한 자세한 내용은 the section called “현재 세대 인스턴스 유형” 섹션을 참조하세요.

로그 분석, 관찰성 또는 보안 분석과 같은 운영 분석 워크로드의 인덱싱을 실행하는 경우 OR1 인스턴스의 향상된 성능과 컴퓨팅 효율성의 이점을 누릴 수 있습니다. 또한 OR1 인스턴스에서 제공하는 자동 데이터 복구는 도메인의 전반적인 신뢰성을 개선합니다.

OpenSearch Service는 Amazon CloudWatch에 스토리지 관련 OR1 지표를 전송합니다. 사용 가능한 지표 목록은 ??? 단원을 참조하십시오.

OR1 인스턴스는 온디맨드 또는 예약 인스턴스 요금으로 사용할 수 있으며, Amazon EBS 및 Amazon S3에서 프로비저닝된 인스턴스 및 스토리지에 대한 시간당 요금이 적용됩니다.

주제

- 제한 사항
- 더 나은 수집 처리량을 위한 조정
- OpenSearch 최적화 인스턴스와 다른 인스턴스의 차이점
- OR1과 UltraWarm 스토리지의 차이
- OR1 인스턴스를 사용하여 도메인 프로비저닝

제한 사항

도메인에 대해 OR1 인스턴스를 사용할 때 다음 제한 사항을 고려합니다.

- 새로 생성된 도메인은 OpenSearch 버전 2.11 이상을 실행해야 합니다.
- 기존 도메인은 OpenSearch 버전 2.15 이상을 실행해야 합니다.
- 도메인에서 저장 시 암호화가 활성화되어 있어야 합니다. 자세한 내용은 ??? 단원을 참조하십시오.

- 도메인이 전용 마스터 노드를 사용하는 경우 Graviton 인스턴스를 사용해야 합니다. 전용 마스터 노드에 대한 자세한 내용은 ??? 섹션을 참조하세요.
- OR1 인스턴스에서 인덱스의 새로 고침 간격은 10초 이상이어야 합니다. OR1 인스턴스의 기본 새로 고침 간격은 10초입니다.

더 나은 수집 처리량을 위한 조정

OR1 인스턴스에서 최적의 인덱싱 처리량을 얻으려면 다음을 수행하는 것이 좋습니다.

- 대용량을 사용하여 버퍼 사용률을 개선합니다. 권장 크기는 10MB입니다.
- 병렬 처리 성능을 개선하려면 여러 클라이언트를 사용합니다.
- 리소스 사용률을 극대화하기 위해 데이터 노드 수와 일치하도록 활성 기본 샤드 수를 설정합니다.

OpenSearch 최적화 인스턴스와 다른 인스턴스의 차이점

OpenSearch 최적화 인스턴스는 다음과 같은 점에서 최적화되지 않은 인스턴스와 다릅니다.

- OpenSearch의 최적화된 인스턴스에서는 기본 샤드에서만 인덱싱이 수행됩니다.
- OpenSearch의 최적화된 인스턴스가 복제본으로 구성된 경우 인덱싱 속도가 실제보다 낮게 나타날 수 있습니다. 예를 들어 기본 샤드 1개와 복제본 샤드 1개가 있는 경우 실제 인덱싱 속도가 2000일 때 인덱싱 속도에 1000의 속도가 표시될 수 있습니다.
- OpenSearch의 최적화된 인스턴스는 원격 소스로 전송하기 전에 버퍼 작업을 수행합니다. 이에 따라 수집 지연 시간이 길어집니다.

> **Note**
>
> `IndexingLatency` 지표에는 트랜스로그 동기화 시간이 포함되지 않으므로 영향을 받지 않습니다.

- 복제본 샤드는 기본 샤드보다 몇 초 지연될 수 있습니다. `ReplicationLagMaxTime` Amazon CloudWatch 지표를 사용하여 지연을 모니터링할 수 있습니다.

OR1과 UltraWarm 스토리지의 차이

OpenSearch Service는 대량의 읽기 전용 데이터를 저장하는 비용 효율적인 방법인 UltraWarm 인스턴스를 제공합니다. OR1 및 UltraWarm 인스턴스는 모두 Amazon EBS에 로컬로 데이터를 저장하고

Amazon S3에 원격으로 데이터를 저장합니다. 그러나 OR1 및 UltraWarm 인스턴스는 몇 가지 중요한 방식에서 차이가 납니다.

- OR1 인스턴스는 로컬 및 원격 스토어 모두에 데이터 사본을 보관합니다. UltraWarm 인스턴스에서 데이터는 스토리지 비용을 절감하기 위해 주로 원격 스토어에 보관됩니다. 사용량 패턴에 따라 데이터를 로컬 스토리지로 이동할 수 있습니다.
- OR1 인스턴스는 활성 상태이며 읽기 및 쓰기 작업을 수락할 수 있는 반면, UltraWarm 인스턴스의 데이터는 수동으로 핫 스토리지로 다시 이동할 때까지 읽기 전용입니다.
- UltraWarm은 데이터 내구성을 위해 인덱스 스냅샷을 사용합니다. 이에 비해 OR1 인스턴스는 백그라운드에서 복제 및 복구를 수행합니다. 빨간색 인덱스가 있는 경우 OR1 인스턴스는 Amazon S3의 원격 스토리지에서 누락된 샤드를 자동 복원합니다. 복구 시간은 복구할 데이터의 양에 따라 달라집니다.

UltraWarm 스토리지에 대한 자세한 내용은 ??? 섹션을 참조하세요.

OR1 인스턴스를 사용하여 도메인 프로비저닝

AWS Management Console 또는 AWS Command Line Interface ()를 사용하여 새 도메인을 생성할 때 데이터 노드에 대한 OR1 인스턴스를 선택할 수 있습니다AWS CLI. 기존 도구를 사용하여 데이터를 인덱싱하고 쿼리할 수 있습니다.

콘솔

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 Domains(도메인)를 선택합니다.
3. 도메인 생성(Create domain)을 선택합니다.
4. 데이터 노드 수 섹션에서 인스턴스 패밀리 메뉴를 확장하고 OpenSearch 최적화를 선택합니다.
5. 인스턴스 유형 및 기타 스토리지 설정을 선택합니다.
6. 암호화 섹션에서 저장 데이터 암호화 활성화가 선택되어 있는지 확인합니다.
7. 도메인의 나머지 부분을 구성하고 생성을 선택합니다.

AWS CLI

를 사용하여 OR1 스토리지를 사용하는 도메인을 프로비저닝하려면에 특정 OR1 인스턴스 유형 크기의 값을 제공해야 AWS CLI합니다`InstanceType`.

다음 예제에서는 크기의 OR1 인스턴스가 있는 도메인을 생성하고 저장 시 암호화를 2xlarge 활성화합니다.

```
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/*"}]}'
```

Amazon OpenSearch Service의 인덱스 상태 관리

Amazon OpenSearch Service에서 인덱스 상태 관리(ISM)를 사용하면 주기적으로 수행되는 태스크를 자동화하도록 고객 관리형 정책을 정의하고 해당 정책을 인덱스와 인덱스 패턴에 적용할 수 있습니다. 인덱스 작업을 실행하기 위해 더 이상 외부 프로세스를 설정하고 관리할 필요가 없습니다.

정책에는 기본 상태와 인덱스 전환에 사용할 수 있는 상태 목록이 포함되어 있습니다. 각 상태 내에서 수행할 작업 목록과 이러한 전환을 트리거할 조건을 정의할 수 있습니다. 일반적인 사용 사례는 일정 기간 후에 오래된 인덱스를 주기적으로 삭제하는 것입니다. 예를 들어 인덱스를 30일 후에 read_only 상태로 이동한 다음 90일 후에 삭제하는 정책을 정의할 수 있습니다.

정책을 인덱스에 연결하면 ISM은 5~8분(또는 1.3 이전 클러스터의 경우 30~48분)마다 실행되는 작업을 생성하여 정책 작업을 수행하고 조건을 확인하며 인덱스를 다른 상태로 전환합니다. 이 작업을 실행하는 기본 시간은 5분마다 임의의 0~60% 지터가 추가되어 모든 인덱스에서 동시에 활동이 급증하지 않도록 합니다. 클러스터 상태가 빨간색이면 ISM이 작업을 실행하지 않습니다.

ISM에는 OpenSearch 또는 Elasticsearch 6.8 이상이 필요합니다.

> **Note**
>
> 이 설명서에서는 ISM 및 여러 샘플 정책에 대한 간략한 개요를 제공합니다. 또한 Amazon OpenSearch Service 도메인의 ISM과 자체 관리형 OpenSearch 클러스터의 ISM의 차이를 설명합니다. 포괄적인 파라미터 참조, 각 설정에 대한 설명 및 API 참조를 포함한 ISM에 대한 전체 설명서는 OpenSearch 설명서의 Index State Management를 참조하세요.

> **Important**
>
> 더 이상 인덱스 템플릿을 사용하여 새로 생성된 인덱스에 ISM 정책을 적용할 수 없습니다. ISM 템플릿 필드에서 새로 생성된 인덱스를 계속해서 자동으로 관리할 수 있습니다. 이 업데이트에서는 이 설정을 사용하여 기존 CloudFormation 템플릿에 영향을 주는 주요 변경 사항을 소개합니다.

ISM 정책 생성

인덱스 상태 관리를 시작하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. ISM 정책을 생성하려는 도메인을 선택합니다.
3. 도메인의 대시보드에서 OpenSearch 대시보드 URL로 이동하여 마스터 사용자 이름과 암호로 로그인합니다. URL은 다음 형식을 따릅니다.

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

4. OpenSearch 대시보드에서 왼쪽 탐색 창을 열고 인덱스 관리(Index Management)를 선택한 다음 정책 생성(Create policy)을 선택합니다.
5. 시각적 편집기 또는 JSON 편집기를 사용하여 정책을 생성합니다. 시각적 편집기는 보다 체계적인 정책 정의 방법을 제공하므로 사용하는 것이 좋습니다. 정책 생성에 도움을 받으려면 아래 샘플 정책을 참조하세요.
6. 정책을 생성한 후 하나 이상의 인덱스에 연결합니다.

   ```
   POST _plugins/_ism/add/my-index
   {
     "policy_id": "my-policy-id"
   ```

```
}
```

> **Note**
>
> 도메인에서 레거시 Elasticsearch 버전을 실행 중인 경우, `_plugins` 대신 `_opendistro`를 사용하세요.

또는 OpenSearch Dashboards에서 인덱스를 선택하고 정책 적용(Apply policy)을 선택합니다.

샘플 정책

다음 샘플 정책은 일반 ISM 사용 사례를 자동화하는 방법을 보여줍니다.

핫 스토리지, 웜 스토리지, 콜드 스토리지

이 샘플 정책은 인덱스를 핫 스토리지에서 UltraWarm, 그리고 결국 콜드 스토리지로 이동합니다. 그런 다음 인덱스를 삭제합니다.

인덱스는 처음에 `hot` 상태입니다. 10일 후 ISM이 인덱스를 `warm` 상태로 전환하고 80일 후, 인덱스가 90일을 경과한 후에는 ISM이 인덱스를 `cold` 상태로 전환합니다. 1년 후, 서비스는 인덱스가 삭제 중이라는 알림을 Amazon Chime 공간에 보낸 다음 영구적으로 삭제합니다.

콜드 인덱스는 정상 `cold_delete` 작업이 아닌 `delete` 작업이 필요합니다. 또한 명시적 `timestamp_field`은(는) ISM으로 콜드 인덱스를 관리하기 위해 데이터에 필요합니다.

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

복제본 수 감소

이 샘플 정책은 7일 후에 복제본 수를 0으로 줄여 디스크 공간을 절약한 다음, 21일 후에 인덱스를 삭제합니다. 이 정책은 인덱스가 중요하지 않으며 더 이상 쓰기 요청을 수신하지 않는다고 가정합니다. 복제본 수가 0이면 데이터 손실의 위험이 있습니다.

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

인덱스 스냅샷 생성

이 샘플 정책은 snapshot 작업을 사용하여 하나 이상의 문서가 포함된 즉시 인덱스의 스냅샷을 생성할 수 있습니다. repository는 Amazon S3를 등록한 수동 스냅샷 리포지토리의 이름입니다. snapshot은 스냅샷의 이름입니다. 리포지토리를 등록하기 위한 스냅샷 사전 요구 사항 및 단계는 the section called “인덱스 스냅샷 생성” 섹션을 참조하세요.

```
{
  "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 템플릿

템플릿 패턴과 일치하는 인덱스를 생성할 때 정책이 해당 인덱스에 자동으로 연결되도록 정책에 `ism_template` 필드를 설정할 수 있습니다. 이 예제에서 "log"로 시작하는 이름으로 만든 인덱스는 ISM 정책 `my-policy-id`와 자동으로 일치됩니다.

```
PUT _plugins/_ism/policies/my-policy-id
{
  "policy": {
    "description": "Example policy.",
    "default_state": "...",
    "states": [...],
    "ism_template": {
      "index_patterns": ["log*"],
      "priority": 100
    }
  }
}
```

자세한 예제는 Sample policy with ISM template for auto rollover(자동 롤오버를 위한 ISM 템플릿을 사용한 샘플 정책)을 참조하세요.

차이

OpenSearch 및 Elasticsearch와 비교할 때 Amazon OpenSearch Service의 ISM은 몇 가지 차이점이 있습니다.

ISM 작업

- OpenSearch Service는 세 가지 고유한 ISM 작업인 `warm_migration`, `cold_migration`, `cold_delete`을(를) 지원합니다.
  - 도메인에 UltraWarm이 활성화된 경우 `warm_migration` 작업을 수행하면 인덱스가 웜 스토리지로 전환됩니다.
  - 도메인에 콜드 스토리지가 활성화된 경우, `cold_migration` 작업은 인덱스를 콜드 스토리지로 전환하고 `cold_delete` 작업은 콜드 스토리지에서 인덱스를 삭제합니다.

이러한 작업이 설정된 제한 시간 내에 완료되지 않더라도 인덱스 마이그레이션 또는 삭제는 여전히 계속됩니다. 위의 작업 중 하나에 대해 error_notification을 설정하면 시간 초과 기간 내에 완료되지 않은 경우 작업이 실패했음을 알립니다. 단, 알림은 참조용입니다. 실제 작업에는 고유한 제한 시간이 없으며 결국 성공 또는 실패할 때까지 계속 실행됩니다.

- 도메인에서 OpenSearch 또는 Elasticsearch 7.4 이상을 실행하는 경우, OpenSearch Service는 ISM `open` 및 `close` 작업을 지원합니다.

- 도메인에서 OpenSearch 또는 Elasticsearch 7.7 이상을 실행하는 경우, OpenSearch Service는 ISM `snapshot` 작업을 지원합니다.

콜드 스토리지 ISM 작업

콜드 인덱스의 경우 다음과 같은 ISM API를 사용할 때 `?type=_cold` 파라미터를 지정해야 합니다.

- 정책 추가
- 정책 제거
- 업데이트 정책
- 실패한 인덱스 재시도
- 인덱스 설명

콜드 인덱스에 대한 이러한 API에는 다음과 같은 추가 차이점이 있습니다.

- 와일드카드 연산자는 끝에서 사용할 때를 제외하고는 지원되지 않습니다. 예를 들어, `_plugins/_ism/<add, remove, change_policy, retry, explain>/logstash-*`는 지원되지만 `_plugins/_ism/<add, remove, change_policy, retry, explain>/iad-*-prod`는 지원되지 않습니다.
- 여러 인덱스 이름 및 패턴을 지원하지 않습니다. 예를 들어, `_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs`는 지원되지만 `_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs,sample-data`는 지원되지 않습니다.

ISM 설정

OpenSearch 및 Elasticsearch에서는 `_cluster/settings` API를 사용하여 이용 가능한 모든 ISM 설정을 변경할 수 있습니다. Amazon OpenSearch Service에서는 다음 ISM 설정만 변경할 수 있습니다.

- 클러스터 수준 설정:
  - `plugins.index_state_management.enabled`
  - `plugins.index_state_management.history.enabled`
- 인덱스 수준 설정:
  - `plugins.index_state_management.rollover_alias`

자습서: 인덱스 상태 관리 프로세스 자동화

이 자습서에서는 주기적으로 수행되는 인덱스 관리 태스크를 자동화하고 인덱스와 인덱스 패턴에 적용하는 ISM 정책을 구현하는 방법을 보여줍니다.

Amazon OpenSearch Service의 인덱스 상태 관리(ISM)를 사용하면 반복적인 인덱스 관리 활동을 자동화할 수 있으므로 인덱스 수명 주기를 관리하기 위해 추가 도구를 사용하지 않아도 됩니다. Amazon OpenSearch Service 도메인 내에서 인덱스 기간, 크기 및 기타 조건을 기반으로 이러한 작업을 자동화하는 정책을 생성할 수 있습니다.

OpenSearch Service는 활성 쓰기 및 지연 시간이 짧은 분석을 위한 기본 '핫' 상태, 최대 3페타바이트의 읽기 전용 데이터를 위한 UltraWarm, 무제한 장기 보관을 위한 콜드 스토리지의 세 가지 스토리지 계층을 지원합니다.

이 자습서에서는 일별 인덱스에서 시계열 데이터를 처리하는 샘플 사용 사례를 제공합니다. 이 자습서에서는 24시간 후에 연결된 각 인덱스의 자동 스냅샷을 생성하는 정책을 설정합니다. 그런 다음 2일 후에 기본 핫 상태에서 UltraWarm 스토리지로, 30일 후에 콜드 스토리지로 인덱스를 마이그레이션하고, 마지막으로 60일 후에 인덱스를 삭제합니다.

사전 조건

- OpenSearch Service 도메인은 Elasticsearch 버전 6.8 이상을 실행해야 합니다.
- 도메인에 UltraWarm과 콜드 스토리지가 활성화되어 있어야 합니다.
- 도메인에 대한 수동 스냅샷 리포지토리를 등록해야 합니다.
- 사용자 역할에는 OpenSearch Service 콘솔에 액세스할 수 있는 충분한 권한이 필요합니다. 필요한 경우 도메인에 대한 액세스를 구성하고 검증합니다.

1단계: ISM 정책 구성

먼저 OpenSearch Dashboards에서 ISM 정책을 구성합니다.

1. OpenSearch Service 콘솔의 도메인 대시보드에서 OpenSearch 대시보드 URL로 이동하여 마스터 사용자 이름과 암호로 로그인합니다. URL은 *domain-endpoint*/_dashboards/ 형식입니다.
2. OpenSearch Dashboards에서 Add sample data(샘플 데이터 추가)를 선택하고 하나 이상의 샘플 인덱스를 도메인에 추가합니다.
3. 왼쪽 탐색 패널을 열고 Index Management(인덱스 관리)를 선택한 다음 Create policy(정책 생성)를 선택합니다.
4. 정책 이름을 `ism-policy-example`로 지정합니다.
5. 기본 정책을 다음과 같은 정책으로 바꿉니다.

```
{
  "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` 필드는 지정된 `index_patterns` 중 하나와 일치하는 새로 생성된 인덱스에 정책을 자동으로 연결합니다. 이 경우 `index-`로 시작하는 모든 인덱스입니다. 사용

자 환경의 인덱스 형식과 일치하도록 이 필드를 수정할 수 있습니다. 자세한 내용은 ISM 템플릿을 참조하세요.

6. 정책의 snapshot 섹션에서 *snapshot-repo*를 도메인에 등록한 스냅샷 리포지토리의 이름으로 바꿉니다. 필요에 따라 *ism-snapshot*을 바꿀 수도 있습니다. 이는 스냅샷 생성 시 스냅샷의 이름이 됩니다.
7. 생성(Create)을 선택합니다. 정책이 이제 State management policies(상태 관리 정책) 페이지에 표시됩니다.

2단계: 하나 이상의 인덱스에 정책 연결

생성한 정책을 클러스터에 있는 하나 이상의 인덱스에 연결합니다.

1. Hot indicies(핫 인덱스) 탭으로 이동하고 opensearch_dashboards_sample을 검색합니다. 1단계에서 추가한 모든 샘플 인덱스가 나열됩니다.
2. 모든 인덱스를 선택하고 Apply policy(정책 적용)를 선택한 다음 방금 생성한 ism-policy-example 정책을 선택합니다.
3. 적용을 선택합니다.

Policy managed indices(정책 관리형 인덱스) 페이지에서 다양한 상태로 전환되는 인덱스를 모니터링할 수 있습니다.

인덱스 롤업을 사용하여 Amazon OpenSearch Service의 인덱스 요약

Amazon OpenSearch Service의 인덱스 롤업을 사용하면 오래된 데이터를 요약 인덱스로 주기적으로 롤업하여 스토리지 비용을 절감할 수 있습니다.

관심 있는 필드를 선택하고 인덱스 롤업을 사용하여 해당 필드만 대략적인 시간 버킷으로 집계된 새 인덱스를 생성합니다. 동일한 쿼리 성능으로 몇 달 또는 몇 년 동안의 기록 데이터를 훨씬 적은 비용으로 저장할 수 있습니다.

인덱스 롤업에는 OpenSearch 또는 Elasticsearch 7.9 이상이 필요합니다.

> **Note**
>
> 이 설명서는 Amazon OpenSearch Service에서 인덱스 롤업 작업 생성을 시작하는 데 도움이 됩니다. 사용 가능한 모든 설정 목록과 전체 API 참조를 포함한 포괄적인 설명서는 OpenSearch 설명서의 Index rollups를 참조하세요.

인덱스 롤업 작업 생성

시작하려면 OpenSearch Dashboards에서 인덱스 관리(Index Management)를 선택합니다. 롤업 작업(Rollup Jobs)을 선택하고 롤업 작업 생성(Create rollup job)을 선택합니다.

1단계: 인덱스 설정

소스 및 대상 인덱스를 설정합니다. 소스 인덱스는 롤업하려는 인덱스입니다. 대상 인덱스는 인덱스 롤업 결과가 저장되는 위치입니다.

인덱스 롤업 작업을 생성한 후에는 인덱스 선택을 변경할 수 없습니다.

2단계: 집계 및 지표 정의

롤업할 집계(용어 및 히스토그램) 및 지표(평균, 합계, 최대, 최소 및 값 개수)가 포함된 특성을 선택합니다. 많은 공간을 절약할 수 없으므로 매우 세분화된 속성을 많이 추가하지 않습니다.

3단계: 일정 지정

인덱스가 수집될 때 인덱스를 롤업할 일정을 지정합니다. 인덱스 롤업 작업은 기본적으로 활성화됩니다.

4단계: 검토 및 생성

구성을 검토하고 생성(Create)을 선택합니다.

5단계: 대상 인덱스 검색

표준 `_search` API를 사용하여 대상 인덱스를 검색할 수 있습니다. 플러그인이 백그라운드에서 대상 인덱스에 맞게 쿼리를 자동으로 다시 작성하므로 대상 인덱스 데이터의 내부 구조에 액세스할 수 없습니다. 이것은 소스 및 대상 인덱스에 대해 동일한 쿼리를 사용할 수 있도록 하기 위한 것입니다.

대상 인덱스를 쿼리하려면 `size`를 0으로 설정합니다.

```
GET target_index/_search
{
  "size": 0,
  "query": {
    "match_all": {}
  },
  "aggs": {
    "avg_cpu": {
      "avg": {
        "field": "cpu_usage"
      }
    }
  }
}
```

> **Note**
>
> OpenSearch 버전 2.2 및 이후 버전에서는 한 번의 요청으로 여러 롤업 인덱스를 검색할 수 있습니다. 2.2 이전의 OpenSearch 버전과 레거시 Elasticsearch OSS 버전은 검색당 하나의 롤업 인덱스만 지원합니다.

Amazon OpenSearch Service에서 인덱스 변환

인덱스 롤업 작업을 사용하면 이전 데이터를 압축된 인덱스로 롤업하여 데이터 세부 수준을 줄일 수 있으며 변환 작업을 통해 특정 필드를 중심으로 데이터의 다른 요약 보기를 만들 수 있으므로 데이터를 여러 가지 방법으로 시각화하거나 분석할 수 있습니다.

인덱스 변환에는 OpenSearch 대시보드 사용자 인터페이스와 REST API가 있습니다. 이 기능을 사용하려면 OpenSearch 1.0 이상이 필요합니다.

> **Note**
>
> 이 설명서에서는 Amazon OpenSearch Service 도메인에서 인덱스 변환을 시작하는 데 도움이 되는 인덱스 변환에 대한 간략한 개요를 제공합니다. 포괄적인 설명서 및 REST API 참조는 오픈 소스 OpenSearch 설명서의 Index transforms를 참조하세요.

인덱스 변환 작업 만들기

클러스터에 데이터가 없는 경우 OpenSearch Dashboards에서 샘플 비행 데이터를 사용하여 변환 작업을 시도합니다. 데이터를 추가한 후 OpenSearch Dashboards를 시작합니다. 그런 다음 인덱스 관리(Index Management), 변환 작업(Transform Jobs), 변환 작업 생성(Create Transform Job)을 차례로 선택합니다.

1단계: 인덱스 선택

인덱스(Indices) 섹션에서 소스 및 대상 인덱스를 선택합니다. 기존 대상 인덱스를 선택하거나 이름을 입력하여 새 대상 인덱스를 생성할 수 있습니다.

소스 인덱스의 하위 집합만 변환하려면 데이터 필터 추가(Add Data Filter)를 선택하고 OpenSearch 쿼리 DSL을 사용하여 소스 인덱스의 하위 집합을 지정합니다.

2단계: 필드 선택

인덱스를 선택한 후 변환 작업에 사용할 필드를 선택하고 그룹화 또는 집계 중 사용할 기능을 선택합니다.

- 그룹화를 사용하여 변환된 인덱스의 별도 버킷에 데이터를 배치할 수 있습니다. 예를 들어, 샘플 비행 데이터 내에서 모든 공항 목적지를 그룹화하려는 경우 `DestAirportID` 필드를 대상 필드인 `DestAirportID_terms` 필드로 그룹화하면 변환 작업이 완료된 후 변환된 인덱스에서 그룹화된 공항 ID를 확인할 수 있습니다.
- 반면에 집계를 사용하면 간단한 계산을 수행할 수 있습니다. 예를 들어 변환 작업에 집계를 포함해 모든 비행기 티켓의 합계를 계산하는 새 필드 `sum_of_total_ticket_price`를 정의할 수 있습니다. 그런 다음 변환된 인덱스의 새 데이터를 분석할 수 있습니다.

3단계: 일정 지정

변환 작업은 기본적으로 활성화되며 일정에 따라 실행됩니다. 변환 실행 간격에서 간격을 분, 시간 또는 일 단위로 지정합니다.

4단계: 검토 및 모니터링

구성을 검토하고 생성(Create)을 선택합니다. 그런 다음 변환 작업 상태(Transform job status) 열을 모니터링합니다.

5단계: 대상 인덱스 검색

작업이 완료되면 표준 `_search` API를 사용하여 대상 인덱스를 검색할 수 있습니다.

예를 들어, `DestAirportID` 필드를 기반으로 비행 데이터를 변환하는 변환 작업을 실행한 후 다음 요청을 실행하여 `SFO` 값이 있는 모든 필드를 반환할 수 있습니다.

```
GET target_index/_search
{
  "query": {
    "match": {
      "DestAirportID_terms" : "SFO"
    }
  }
}
```

Amazon OpenSearch Service의 클러스터 간 복제

Amazon OpenSearch Service의 클러스터 간 복제를 사용하면 특정 OpenSearch 서비스 도메인에서 다른 도메인으로 사용자 인덱스, 매핑 및 메타데이터를 복제할 수 있습니다. 클러스터 간 복제를 사용하면 중단 시 재해 복구를 보장하는 데 도움이 되며, 지리적으로 멀리 떨어진 데이터 센터 간에 데이터를 복제하여 대기 시간을 줄일 수 있습니다. 도메인 간에 전송되는 AWS 데이터에 대해 표준 데이터 전송 요금을 지불합니다.

클러스터 간 복제는 로컬 또는 팔로어 인덱스가 원격 또는 리더 인덱스에서 데이터를 가져오는 액티브-패시브 복제 모델을 따릅니다. 리더 인덱스는 데이터 원본 또는 데이터를 복제하려는 인덱스를 나타냅니다. 팔로워 인덱스는 데이터 대상 또는 데이터를 복제하려는 인덱스를 나타냅니다.

클러스터 간 복제는 Elasticsearch 7.10 또는 OpenSearch 1.1 이상을 실행하는 도메인에서 사용할 수 있습니다.

> **Note**
>
> 이 설명서에서는 Amazon OpenSearch Service 관점에서 교차 클러스터 복제를 설정하는 방법을 설명합니다. 여기에는 AWS Management Console 를 사용하여 자체 관리형 OpenSearch 클러스터에서 불가능한 교차 클러스터 연결을 설정하는 것이 포함됩니다. 설정 참조 및 포괄적인 API 참조를 포함한 전체 설명서는 OpenSearch 설명서의 Cross-cluster replication을 참조하세요.

주제

- 제한 사항
- 사전 조건
- 권한 요구 사항
- 클러스터 간 연결 설정
- 복제 시작
- 복제 확인
- 복제 일시 중지 및 다시 시작
- 복제 중지
- 자동 팔로우
- 연결된 도메인 업그레이드

제한 사항

클러스터 간 복제에는 다음 제한 사항이 적용됩니다.

- Amazon OpenSearch Service 도메인과 자체 관리형 OpenSearch 또는 Elasticsearch 클러스터 간에는 데이터를 복제할 수 없습니다.
- 팔로워 도메인의 인덱스를 다른 팔로워 도메인으로 복제할 수 없습니다. 인덱스를 여러 팔로워 도메인에 복제하려는 경우 단일 리더 도메인에서만 복제할 수 있습니다.
- 도메인은 인바운드 연결과 아웃바운드 연결의 조합을 통해 최대 20개의 다른 도메인에 연결할 수 있습니다.
- 클러스터 간 연결을 처음 설정할 때는 리더 도메인이 팔로워 도메인과 같거나 상위 버전에 있어야 합니다.
- AWS CloudFormation 를 사용하여 도메인을 연결할 수 없습니다.
- M3 또는 버스트 가능(T2 및 T3) 인스턴스에서는 클러스터 간 복제를 사용할 수 없습니다.
- UltraWarm 또는 콜드 인덱스 간에는 데이터를 복제할 수 없습니다. 두 인덱스 모두 핫 스토리지에 있어야 합니다.
- 리더 도메인에서 인덱스를 삭제해도 팔로워 도메인의 해당 인덱스는 자동으로 삭제되지 않습니다.

사전 조건

클러스터 간 복제를 설정하기 전에 도메인이 다음 요구 사항을 충족하는지 확인하세요.

- Elasticsearch 7.10 또는 OpenSearch 1.1 이상
- 세분화된 액세스 제어를 사용하도록 설정됨
- 노드 간 암호화를 사용하도록 설정됨

권한 요구 사항

복제를 시작하려면 원격(리더) 도메인에 대한 `es:ESCrossClusterGet` 권한을 포함해야 합니다. 원격 도메인에서 다음 IAM 정책을 사용하는 것이 좋습니다. 이 정책을 사용하면 문서 인덱싱 및 표준 검색 수행과 같은 다른 작업까지 수행할 수 있습니다.

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"
    }
  ]
}
```

es:ESCrossClusterGet 권한이 /leader-domain/*이 아닌 /leader-domain에 적용되었는지 확인합니다.

관리자가 아닌 사용자가 복제 작업을 수행하려면 해당 사용자도 적절한 권한에 매핑되어야 합니다. 대부분의 권한은 특정 REST API 작업에 해당합니다. 예를 들어 indices:admin/plugins/replication/index/_resume 권한을 사용하면 인덱스 복제를 재개할 수 있습니다. 전체 권한 목록은 OpenSearch 문서에서 복제 권한을 참조하세요.

> **Note**
>
> 복제를 시작하고 복제 규칙을 생성하는 명령은 특별한 경우입니다. 리더 도메인과 팔로워 도메인에서 백그라운드 프로세스를 호출하기 때문에 요청에서 leader_cluster_role 및 follower_cluster_role을(를) 통과해야 합니다. OpenSearch Service는 모든 백엔드 복제 작업에서 이러한 역할을 사용합니다. 이러한 역할을 매핑하고 사용하는 데 대한 자세한 내용은 OpenSearch 문서에서 리더 및 팔로워 클러스터 역할 매핑을 참조하세요.

클러스터 간 연결 설정

특정 도메인에서 다른 도메인으로 인덱스를 복제하려면 도메인 간에 클러스터 간 연결을 설정해야 합니다. 도메인을 연결하는 가장 쉬운 방법은 도메인 대시보드의 [연결(Connections)] 탭을 사용하는 것입니다. 구성 API 또는 AWS CLI를 사용할 수도 있습니다. 클러스터 간 복제는 '풀' 모델을 따르므로 팔로워 도메인에서 연결을 초기화합니다.

> **Note**
>
> 이전에 클러스터 간 검색을 수행하기 위해 2개의 도메인을 연결한 경우, 동일한 연결을 복제에 사용할 수 없습니다. 해당 연결은 콘솔에서 SEARCH_ONLY로 표시됩니다. 이전에 연결된 두 도메인 간에 복제를 수행하려면 연결을 삭제하고 다시 생성해야 합니다. 이렇게 하면 교차 클러스터 검색 및 교차 클러스터 복제 모두에 연결을 사용할 수 있습니다.

연결을 설정하려면

1. Amazon OpenSearch Service 콘솔에서 팔로워 도메인을 선택하고 [연결(Connections)] 탭으로 이동하여 [요청(Request)]을 선택합니다.
2. [연결 별칭(Connection alias)]에 연결 이름을 입력합니다.
3. AWS 계정 및 리전 또는 다른 계정 또는 리전의 도메인에 연결할지 선택합니다.

- AWS 계정 및 리전의 도메인에 연결하려면 도메인을 선택하고 요청을 선택합니다.
- 다른 AWS 계정 또는 리전의 도메인에 연결하려면 원격 도메인의 ARN을 지정하고 요청을 선택합니다.

OpenSearch 서비스는 연결 요청을 검증합니다. 도메인이 서로 호환되지 않으면 연결이 실패합니다. 검증에 성공하면 승인을 위해 대상 도메인으로 전송됩니다. 대상 도메인이 요청을 승인하면 복제를 시작할 수 있습니다.

클러스터 간 복제는 양방향 복제를 지원합니다. 즉, 도메인 A에서 도메인 B로의 아웃바운드 연결과 도메인 B에서 도메인 A로의 또 다른 아웃바운드 연결을 만들 수 있습니다. 그런 다음 도메인 A가 도메인 B의 인덱스를 따르고 도메인 B가 도메인 A의 인덱스를 따르도록 복제를 설정할 수 있습니다.

복제 시작

클러스터 간 연결을 설정하고 나면 데이터 복제를 시작할 수 있습니다. 먼저 복제할 리더 도메인에 인덱스를 생성합니다.

```
PUT leader-01
```

해당 인덱스를 복제하기 위해 다음 명령을 팔로워 도메인으로 보냅니다.

```
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"
   }
}
```

도메인 대시보드의 연결(Connections) 탭에서 연결 별칭을 찾을 수 있습니다.

이 예에서는 설명의 편의를 위해 관리자가 요청을 실행하고 `leader_cluster_role` 및 `follower_cluster_role`(으)로 `all_access`을(를) 사용하는 것으로 가정합니다. 하지만 프로덕션 환경에서는 리더 및 팔로워 인덱스 모두에 복제 사용자를 생성하고 그에 따라 매핑하는 것이 좋습니다. 사용자 이름은 동일해야 합니다. 이러한 역할과 그 매핑 방법에 대한 자세한 내용은 OpenSearch 문서에서 리더 및 팔로워 클러스터 역할 매핑을 참조하세요.

복제 확인

복제가 진행되고 있는지 확인하려면 복제 상태를 가져옵니다.

```
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
  }
}
```

리더 및 팔로워 체크포인트 값은 음의 정수로 시작하며, 보유한 샤드 수를 반영합니다(샤드가 하나인 경우 -1, 샤드가 5개인 경우 -5 등과 같은 식임). 변경할 때마다 값이 양의 정수로 증가합니다. 값이 동일하면 인덱스가 완전히 동기화되었음을 의미합니다. 이러한 체크포인트 값을 사용하여 도메인 전체의 복제 대기 시간을 측정할 수 있습니다.

복제를 추가로 검증하려면 리더 인덱스에 문서를 추가합니다.

```
PUT leader-01/_doc/1
{
   "Doctor Sleep":"Stephen King"
}
```

그리고 팔로워 인덱스에 표시되는지 확인합니다.

```
GET follower-01/_search

{
    ...
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "follower-01",
```

```
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "Doctor Sleep" : "Stephen King"
        }
      }
    ]
  }
}
```

복제 일시 중지 및 다시 시작

문제를 해결하거나 리더 도메인의 부하를 줄여야 하는 경우 복제를 일시적으로 중지할 수 있습니다. 이 요청을 팔로워 도메인에 보냅니다. 다음과 같이 빈 요청 본문을 포함해야 합니다.

```
POST _plugins/_replication/follower-01/_pause
{}
```

그런 다음 상태를 가져와 복제가 일시 중지되었는지 확인합니다.

```
GET _plugins/_replication/follower-01/_status

{
  "status" : "PAUSED",
  "reason" : "User initiated",
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01"
}
```

변경을 마치면 복제를 다시 시작합니다. 이 요청을 팔로워 도메인에 보냅니다. 다음과 같이 빈 요청 본문을 포함해야 합니다.

```
POST _plugins/_replication/follower-01/_resume
{}
```

12시간 이상 일시 중지된 후에는 복제를 재개할 수 없습니다. 복제를 중지하고 팔로워 인덱스를 삭제한 다음 리더의 복제를 다시 시작해야 합니다.

복제 중지

복제를 완전히 중지하면 팔로워 인덱스가 리더를 팔로우하지 않고 표준 인덱스가 됩니다. 복제를 중지한 후에는 다시 시작할 수 없습니다.

팔로워 도메인에서 복제를 중지합니다. 다음과 같이 빈 요청 본문을 포함해야 합니다.

```
POST _plugins/_replication/follower-01/_stop
{}
```

자동 팔로우

단일 리더 도메인에 대해 지정된 패턴과 일치하는 인덱스를 자동으로 복제하는 일련의 복제 규칙을 정의할 수 있습니다. 리더 도메인의 인덱스가 패턴 중 하나와 일치하는 경우(예: books*), 일치하는 팔로워 인덱스가 팔로워 도메인에 생성됩니다. OpenSearch Service에서는 패턴과 일치하는 기존 인덱스와 사용자가 생성하는 새 인덱스를 복제합니다. 팔로워 도메인에 이미 있는 인덱스는 복제하지 않습니다.

시스템에서 생성한 인덱스와 팔로워 도메인에 이미 있는 인덱스를 제외한 모든 인덱스를 복제하려면 와일드카드(*) 패턴을 사용합니다.

복제 규칙 생성

팔로워 도메인에 복제 규칙을 생성하고 클러스터 간 연결의 이름을 지정합니다.

```
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"
   }
}
```

도메인 대시보드의 연결(Connections) 탭에서 연결 별칭을 찾을 수 있습니다.

이 예에서는 설명의 편의를 위해 관리자가 요청을 실행하고 리더 및 팔로워 도메인 역할로 `all_access`을(를) 사용하는 것으로 가정합니다. 하지만 프로덕션 환경에서는 리더 및 팔로워 인덱스 모두에 복제 사용자를 생성하고 그에 따라 매핑하는 것이 좋습니다. 사용자 이름은 동일해야 합니다.

이러한 역할과 그 매핑 방법에 대한 자세한 내용은 OpenSearch 문서에서 리더 및 팔로워 클러스터 역할 매핑을 참조하세요.

도메인의 기존 복제 규칙 목록을 검색하려면 자동 팔로우 통계 API 작업을 사용합니다.

규칙을 테스트하려면 리더 도메인의 패턴과 일치하는 인덱스를 생성합니다.

```
PUT books-are-fun
```

그리고 해당 복제본이 팔로워 도메인에 표시되는지 확인합니다.

```
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
```

복제 규칙 삭제

복제 규칙을 삭제하면 OpenSearch Service가 패턴과 일치하는 새 인덱스의 복제를 중지하지만, 기존 복제 작업은 해당 인덱스의 복제를 중지할 때까지 계속합니다.

팔로워 도메인에서 복제 규칙을 삭제합니다.

```
DELETE _plugins/_replication/_autofollow
{
   "leader_alias" : "connection-alias",
   "name": "rule-name"
}
```

연결된 도메인 업그레이드

클러스터 간 연결이 있는 두 도메인의 엔진 버전을 업그레이드하려면 먼저 팔로워 도메인을 업그레이드한 다음 리더 도메인을 업그레이드하십시오. 두 도메인 간 연결을 삭제하면 복제가 일시 중지되고 다시 시작할 수 없으므로 연결을 삭제해서는 안 됩니다.

원격 재인덱스를 사용하여 Amazon OpenSearch Service 인덱스 마이그레이션

원격 재인덱스를 사용하면 한 Amazon OpenSearch Service 도메인에서 다른 도메인으로 인덱스를 복사할 수 있습니다. OpenSearch Service 도메인이나 자체 관리형 OpenSearch 및 Elasticsearch 클러스터에서 인덱스를 마이그레이션할 수 있습니다.

원격 도메인 및 인덱스는 데이터 원본 또는 데이터를 복사하려는 도메인과 인덱스를 나타냅니다. 로컬 도메인 및 인덱스는 데이터 대상 또는 데이터를 복사하려는 도메인과 인덱스를 나타냅니다.

원격 인덱싱을 사용하려면 로컬 도메인에서 OpenSearch 1.0 이상 또는 Elasticsearch 6.7 이상이 필요합니다. 원격 도메인의 메이저 버전은 로컬 도메인보다 더 낮거나 대상 도메인과 동일해야 합니다. Elasticsearch 버전은 OpenSearch 버전보다 낮은 것이 좋습니다. 즉, Elasticsearch 도메인에서 OpenSearch 도메인으로 데이터를 재인덱스할 수 있습니다. 동일한 메이저 버전 내에서 원격 도메인은 마이너 버전이 될 수 있습니다. 예를 들어 Elasticsearch 7.10.x에서 7.9로 원격 재인덱싱은 지원되지만 OpenSearch 1.0에서 Elasticsearch 7.10.x로의 원격 재인덱싱은 지원되지 않습니다.

> **Note**
>
> 이 설명서에서는 Amazon OpenSearch Service 도메인 사이에서 데이터를 다시 인덱싱하는 방법을 설명합니다. 자세한 단계 및 지원되는 옵션을 비롯한 `reindex` 작업에 대한 전체 설명서는 OpenSearch 설명서의 Reindex document를 참조하세요.

주제

- 사전 조건
- OpenSearch Service 인터넷 도메인 간의 데이터 재인덱스
- 원격이 VPC에 있을 때 OpenSearch Service 도메인 간의 데이터 재인덱스
- 비 OpenSearch Service 도메인 간 데이터 재인덱스
- 대용량 데이터 집합 재인덱스
- 원격 재인덱스 설정

사전 조건

원격 재인덱스의 요구 사항은 다음과 같습니다.

- 원격 도메인은 로컬 도메인에서 액세스할 수 있어야 합니다. VPC에 상주하는 원격 도메인의 경우, 로컬 도메인이 VPC에 액세스할 수 있어야 합니다. 이 프로세스는 네트워크 구성에 따라 다르지만, VPN 또는 관리형 네트워크 연결 또는 네이티브 VPC 엔드포인트 연결 사용이 필요할 수 있습니다. 자세한 내용은 the section called “VPC 지원”를 참조하세요.
- 요청은 다른 REST 요청과 마찬가지로 원격 도메인에서 승인해야 합니다. 원격 도메인에 세분화된 액세스 제어가 활성화된 경우, 원격 도메인에서 재인덱스를 수행하고 로컬 도메인의 인덱스를 읽을 수 있는 권한이 있어야 합니다. 보안 고려 사항에 대한 자세한 내용은 the section called “세분화된 액세스 제어” 단원을 고려하세요.
- 재인덱스 프로세스를 시작하기 전에 로컬 도메인에서 원하는 설정으로 인덱스를 생성하는 것이 좋습니다.
- 도메인에서 데이터 노드에 T2 또는 T3 인스턴스 유형을 사용하는 경우, 원격 재인덱스를 사용할 수 없습니다.

OpenSearch Service 인터넷 도메인 간의 데이터 재인덱스

가장 기본적인 시나리오는 원격 인덱스가 공개적으로 액세스할 수 있는 엔드포인트가 있는 로컬 도메인과 AWS 리전 동일하고 IAM 자격 증명에 서명했다는 것입니다.

원격 도메인에서 재인덱스해올 원격 인덱스와 재인덱스할 로컬 인덱스를 지정하세요.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

유효성 검사를 위해 원격 도메인 엔드포인트 끝에 443을 추가해야 합니다.

인덱스가 로컬 도메인으로 복사되었는지 확인하려면 다음 요청을 로컬 도메인에 보내세요.

```
GET local_index/_search
```

원격 인덱스가 로컬 도메인과 다른 리전에 있는 경우 다음 샘플 요청과 같이 해당 리전 이름을 전달하세요.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "region": "eu-west-1"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

AWS GovCloud (US) 또는 중국 리전과 같은 격리된 리전의 경우 IAM 사용자가 해당 리전에서 인식되지 않기 때문에 엔드포인트에 액세스하지 못할 수 있습니다.

원격 도메인이 기본 인증으로 보호되는 경우 사용자 이름과 암호를 지정합니다.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

원격이 VPC에 있을 때 OpenSearch Service 도메인 간의 데이터 재인덱스

모든 OpenSearch Service 도메인은 자체 내부 Virtual Private Cloud(VPC) 인프라로 구성됩니다. 기존 OpenSearch Service VPC에 새 도메인을 생성하면 VPC의 각 데이터 노드에 대해 탄력적인 네트워크 인터페이스가 생성됩니다.

원격 재인덱스 작업은 원격 OpenSearch Service 도메인에서 수행되므로 자체 프라이빗 VPC 내에서 수행되기 때문에 로컬 도메인의 VPC에 액세스할 방법이 필요합니다. 기본 제공 VPC 엔드포인트 연결 기능을 사용하여 연결을 설정 AWS PrivateLink하거나 프록시를 구성하여이 작업을 수행할 수 있습니다.

로컬 도메인에서 OpenSearch 버전 1.0 이상을 사용하는 경우 콘솔 또는를 사용하여 AWS PrivateLink 연결을 AWS CLI 생성할 수 있습니다. AWS PrivateLink 연결을 사용하면 로컬 VPC의 리소스가 동일한 내의 원격 VPC의 리소스에 비공개로 연결할 수 있습니다 AWS 리전.

VPC 엔드포인트 연결을 생성하려면 다시 인덱싱할 소스 도메인이 로컬 VPC에 있어야 하며 소스 도메인과 대상 도메인이 모두 동일한에 있어야 합니다 AWS 리전.

를 사용하여 데이터 재인덱싱 AWS Management Console

콘솔로 원격 재인덱싱을 사용하여 VPC 엔드포인트 연결을 공유하는 두 도메인 간에 인덱스를 복사할 수 있습니다.

1. Amazon OpenSearch Service 콘솔(https://console.aws.amazon.com/aos/)로 이동합니다.
2. 왼쪽 탐색 창에서 Domains(도메인)를 선택합니다.
3. 로컬 도메인 또는 데이터를 복사하려는 도메인을 선택합니다. 그러면 도메인 세부 정보 페이지가 열립니다. 일반 정보 아래에서 연결 탭을 선택하고 요청를 선택합니다.
4. 연결 요청 페이지에서 연결 모드로 VPC 엔드포인트 연결을 선택하고 기타 관련 세부 정보를 입력합니다. 이러한 세부 정보에는 데이터를 복사하려는 도메인인 원격 도메인이 포함됩니다. 그런 다음 Request(요청)를 선택합니다.
5. 원격 도메인의 세부 정보 페이지로 이동하고 연결 탭을 선택한 다음 인바운드 연결 테이블을 찾습니다. 방금 연결을 생성한 도메인(로컬 도메인)의 이름 옆에 있는 확인란을 선택합니다. Approve(승인)를 선택합니다.
6. 로컬 도메인으로 다시 이동하여 Connections(연결) 탭을 선택하고 Outbound connections(아웃바운드 연결) 테이블을 찾습니다. 두 도메인 간의 연결이 활성화되면 테이블의 Endpoint(엔드포인트) 열에서 엔드포인트를 사용할 수 있게 됩니다. 엔드포인트를 복사합니다.
7. 로컬 도메인의 대시보드를 열고 왼쪽 탐색에서 Dev Tools(개발 도구)를 선택합니다. 원격 도메인 인덱스가 아직 로컬 도메인에 존재하지 않는지 확인하려면 다음 GET 요청을 실행합니다. *remote-domain-index-name*을 자체 인덱스 이름으로 바꿉니다.

```
GET remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
```

```
   }
}
```

출력에 인덱스를 찾을 수 없다는 오류가 표시되어야 합니다.

8. 다음과 같이 GET 요청 아래에서 POST 요청을 생성하고 엔드포인트를 원격 호스트로 사용합니다.

```
POST _reindex
{
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

이 요청을 실행합니다.

9. GET 요청을 다시 실행합니다. 이제 출력에 로컬 인덱스가 존재한다는 내용이 표시되어야 합니다. 이 인덱스를 쿼리하여 OpenSearch가 원격 인덱스에서 모든 데이터를 복사했는지 확인할 수 있습니다.

OpenSearch Service API 작업을 사용한 데이터 재인덱싱

API로 원격 재인덱싱을 사용하여 VPC 엔드포인트 연결을 공유하는 두 도메인 간에 인덱스를 복사할 수 있습니다.

1. CreateOutboundConnection API 작업을 사용하여 로컬 도메인에서 원격 도메인으로의 새 연결을 요청하세요.

```
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"
      }
   }
}
```

응답에서 ConnectionId을 받게 됩니다. 다음 단계에서 사용할 수 있도록 이 ID를 저장합니다.

2. 연결 ID로 AcceptInboundConnection API 작업을 사용하여 로컬 도메인의 요청을 승인하세요.

```
PUT https://es.region.amazonaws.com/2021-01-01/opensearch/cc/
inboundConnection/ConnectionId/accept
```

3. DescribeOutboundConnections API 작업을 사용하여 원격 도메인의 엔드포인트를 검색하세요.

```
{
    "Connections": [
        {
            "ConnectionAlias": "remote-reindex-example",
            "ConnectionId": "connection-id",
            "ConnectionMode": "VPC_ENDPOINT",
            "ConnectionProperties": {
                "Endpoint": "connection-endpoint"
            },
            ...
        }
    ]
}
```

5단계에서 사용할 *connection-endpoint*를 저장합니다.

4. 원격 도메인 인덱스가 아직 로컬 도메인에 존재하지 않는지 확인하려면 다음 GET 요청을 실행합니다. *remote-domain-index-name*을 자체 인덱스 이름으로 바꿉니다.

```
GET local-domain-endpoint/remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

출력에 인덱스를 찾을 수 없다는 오류가 표시되어야 합니다.

5. 다음과 같이 POST 요청을 생성하고 엔드포인트를 원격 호스트로 사용합니다.

```
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"
   }
}
```

이 요청을 실행합니다.

6. GET 요청을 다시 실행합니다. 이제 출력에 로컬 인덱스가 존재한다는 내용이 표시되어야 합니다. 이 인덱스를 쿼리하여 OpenSearch가 원격 인덱스에서 모든 데이터를 복사했는지 확인할 수 있습니다.

원격 도메인이 VPC 내부에서 호스팅되고 VPC 엔드포인트 연결 기능을 사용하지 않으려는 경우, 공개적으로 액세스할 수 있는 엔드포인트로 프록시를 구성해야 합니다. 이 경우 OpenSearch Service에는 VPC로 트래픽을 전송할 수 있는 기능이 없으므로 퍼블릭 엔드포인트가 필요합니다.

VPC 모드에서 도메인을 실행하면 하나 이상의 엔드포인트가 VPC에 배치됩니다. 하지만 이러한 엔드포인트는 VPC 내에서 도메인으로 들어오는 트래픽만을 위한 것이며 VPC 자체로의 트래픽은 허용하지 않습니다.

원격 재인덱싱 명령은 로컬 도메인에서 실행되므로 원본 트래픽은 해당 엔드포인트를 사용하여 원격 도메인에 액세스할 수 없습니다. 이 사용 사례에서 프록시가 필요한 이유입니다. 프록시 도메인에는 공공 인증 기관(CA)에서 서명한 인증서가 있어야 합니다. 자체 서명 또는 개인 CA 서명 인증서는 지원되지 않습니다.

비 OpenSearch Service 도메인 간 데이터 재인덱스

원격 인덱스가 자체 관리형 EC2 인스턴스에서와 같이 OpenSearch 서비스 외부에서 호스팅되는 경우 `external` 파라미터를 `true`로 설정합니다.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password",
      "external": true
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

이 경우 사용자 이름과 암호를 사용한 기본 인증만 지원됩니다. 원격 도메인에는 공개적으로 액세스할 수 있는 엔드포인트(로컬 OpenSearch Service 도메인과 동일한 VPC에 있더라도)와 퍼블릭 CA에서 서명한 인증서가 있어야 합니다. 자체 서명 또는 개인 CA 서명 인증서는 지원되지 않습니다.

대용량 데이터 집합 재인덱스

원격 재인덱스는 다음 기본값을 사용하여 원격 도메인에 스크롤 요청을 보냅니다.

- 검색 컨텍스트 5분
- 소켓 제한 시간 30초
- 배치 크기 1,000

데이터를 수용하기 위해 이러한 파라미터를 조정하는 것이 좋습니다. 큰 문서의 경우 일괄 처리 크기를 줄이거나 제한 시간을 늘리는 것을 고려합니다. 자세한 내용은 결과 페이지 매김을 참조하세요.

```
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"
  }
}
```

또한 성능 향상을 위해 다음 설정을 로컬 인덱스에 추가하는 것이 좋습니다.

```
PUT local_index
{
  "settings": {
    "refresh_interval": -1,
    "number_of_replicas": 0
  }
}
```

재인덱스 프로세스가 완료되면 원하는 복제본 수를 설정하고 새로 고침 간격 설정을 제거할 수 있습니다.

쿼리를 통해 선택한 문서의 하위 집합만 재인덱스하려면 이 요청을 로컬 도메인으로 보내세요.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index",
    "query": {
      "match": {
        "field_name": "text"
      }
    }
```

```
  },
  "dest": {
    "index": "local_index"
  }
}
```

원격 재인덱스는 슬라이싱을 지원하지 않으므로 동일한 요청에 대해 여러 스크롤 작업을 병렬로 수행할 수 없습니다.

원격 재인덱스 설정

OpenSearch Service는 표준 재인덱싱 옵션 외에도 다음 옵션을 지원합니다.

옵션	유효값	설명	필수
외부	불	원격 도메인이 OpenSearch Service 도메인이 아니거나 2개의 VPC 도메인 사이에서 재인덱싱하는 경우 `true`로 지정하세요.	아니요
리전	String	원격 도메인이 다른 리전에 있는 경우 리전 이름을 지정하세요.	아니요

데이터 스트림을 사용하여 Amazon OpenSearch Service에서 시계열 데이터 관리

시계열 데이터를 관리하는 일반적인 워크플로에는 롤오버 인덱스 별칭 생성, 쓰기 인덱스 정의, 백업 인덱스에 대한 공통 매핑 및 설정 정의와 같은 여러 단계가 포함됩니다.

Amazon OpenSearch Service의 데이터 스트림은 이러한 초기 설정 프로세스를 간소화하는 데 도움이 됩니다. 보통 본질적으로 추가 전용인 애플리케이션 로그와 같은 시간 기반 데이터에 대해서는 데이터 스트림이 즉시 작동합니다.

데이터 스트림을 사용하려면 OpenSearch 버전 1.0 이상이 필요합니다.

> **Note**
>
> 이 설명서에서는 Amazon OpenSearch Service 도메인에서 데이터 스트림을 시작하는 데 도움이 되는 기본적인 단계를 제공합니다. 포괄적인 설명서를 보려면 OpenSearch 설명서의 Data streams를 참조하세요.

데이터 스트림 시작하기

데이터 스트림은 내부적으로 여러 백업 인덱스로 구성됩니다. 검색 요청은 모든 백업 인덱스로 라우팅되고 인덱싱 요청은 최신 쓰기 인덱스로 라우팅됩니다.

1단계: 인덱스 템플릿 생성

데이터 스트림을 생성하려면 먼저 인덱스 집합을 데이터 스트림으로 구성하는 인덱스 템플릿을 생성해야 합니다. `data_stream` 객체는 데이터 스트림이며 일반 인덱스 템플릿이 아니라는 것을 나타냅니다. 인덱스 패턴은 데이터 스트림의 이름과 일치합니다:

```
PUT _index_template/logs-template
{
  "index_patterns": [
    "my-data-stream",
    "logs-*"
  ],
  "data_stream": {},
  "priority": 100
}
```

이 경우 수집된 각 문서에는 `@timestamp` 필드가 있어야 합니다. 사용자 지정 타임스탬프 필드를 `data_stream` 객체의 속성으로 정의할 수도 있습니다.

```
PUT _index_template/logs-template
{
  "index_patterns": "my-data-stream",
  "data_stream": {
    "timestamp_field": {
      "name": "request_time"
    }
  }
}
```

2단계: 데이터 스트림 생성

인덱스 템플릿을 생성한 후에는 데이터 스트림을 생성하지 않고 직접 데이터 수집을 시작할 수 있습니다.

`data_stream` 객체와 일치하는 인덱스 템플릿이 있기 때문에 OpenSearch가 자동으로 데이터 스트림을 생성합니다.

```
POST logs-staging/_doc
{
  "message": "login attempt failed",
  "@timestamp": "2013-03-01T00:00:00"
}
```

3단계: 데이터 스트림에 데이터 수집

데이터를 데이터 스트림으로 수집하기 위해 일반 인덱싱 API를 사용할 수 있습니다. 인덱싱하는 모든 문서에 타임스탬프 필드가 있는지 확인합니다. 타임스탬프 필드가 없는 문서를 수집하려고 하면 오류 메시지가 표시됩니다.

```
POST logs-redis/_doc
{
  "message": "login attempt",
  "@timestamp": "2013-03-01T00:00:00"
}
```

4단계: 데이터 스트림 검색

일반 인덱스 또는 인덱스 별칭을 검색하는 것처럼 데이터 스트림을 검색할 수 있습니다. 검색 작업은 모든 백업 인덱스(스트림에 존재하는 모든 데이터) 에 적용됩니다.

```
GET logs-redis/_search
{
  "query": {
    "match": {
      "message": "login"
    }
  }
}
```

5단계: 데이터 스트림 롤오버

인덱스 상태 관리(ISM) 정책을 설정하여 데이터 스트림에 대한 롤오버 프로세스를 자동화할 수 있습니다. ISM 정책은 생성 시 백업 인덱스에 적용됩니다. 정책을 데이터 스트림에 연결하면 해당 데이터 스트림의 향후 백업 인덱스에만 영향을 줍니다. 또한 ISM 정책이 백업 인덱스에서 이 정보를 유추하기 때문에 `rollover_alias` 설정을 제공할 필요가 없습니다.

> **Note**
>
> 백업 인덱스를 콜드 스토리지로 롤오버하면 OpenSearch가 데이터 스트림에서 이 인덱스를 제거합니다. 인덱스를 UltraWarm으로 다시 이동하더라도 인덱스는 독립적으로 유지되며 원래 데이터 스트림의 일부가 아닙니다. 데이터 스트림에서 인덱스를 제거한 후 스트림을 검색해도 인덱스에서 데이터가 반환되지 않습니다.

> **Warning**
>
> 데이터 스트림의 쓰기 인덱스는 콜드 스토리지로 마이그레이션할 수 없습니다. 데이터 스트림의 데이터를 콜드 스토리지로 마이그레이션하려면 마이그레이션하기 전에 데이터 스트림을 롤오버해야 합니다.

6단계: OpenSearch Dashboards에서 데이터 스트림 관리

OpenSearch Dashboards에서 데이터 스트림을 관리하려면 OpenSearch Dashboards를 열고 인덱스 관리(Index Management)를 선택한 다음 인덱스(Indices) 또는 정책 관리형 인덱스(Policy managed indices)를 선택합니다.

7단계: 데이터 스트림 삭제

삭제 작업은 먼저 데이터 스트림의 백업 인덱스를 삭제한 다음 데이터 스트림 자체를 삭제합니다.

데이터 스트림과 숨겨진 모든 백업 인덱스를 삭제하려면 다음을 수행합니다.

```
DELETE _data_stream/name_of_data_stream
```

Amazon OpenSearch Service의 데이터 모니터링

알림 및 이상 탐지를 통해 Amazon OpenSearch Service에서 데이터를 사전 예방적으로 모니터링합니다. 데이터가 특정 임계값을 초과하면 알림을 수신하도록 알림을 설정합니다. 이상 탐지는 기계 학습을 사용하여 스트리밍 데이터의 이상치를 자동으로 탐지합니다. 이상 탐지와 알림 기능을 페어링하여 이상이 탐지되는 즉시 알림을 받을 수 있습니다.

주제

- Amazon OpenSearch Service의 알림 구성
- Amazon OpenSearch Service의 이상 탐지

Amazon OpenSearch Service의 알림 구성

Amazon OpenSearch Service에서 알림을 구성하여 하나 이상의 인덱스에 있는 데이터가 특정 조건을 충족하면 알림을 수신합니다. 예를 들어 애플리케이션이 1시간에 HTTP 503 오류를 6개 이상 기록하면 이메일을 받거나, 지난 20분간 인덱싱된 새 문서가 없으면 개발자에게 호출할 수 있습니다.

이 알림을 받으려면 OpenSearch 또는 Elasticsearch 6.2 이상이 필요합니다.

> **Note**
>
> 이 설명서에서는 알림에 대한 간략한 개요를 제공하고 Amazon OpenSearch Service 도메인의 알림과 오픈 소스 OpenSearch 클러스터의 알림 간 차이를 강조합니다. 포괄적인 API 참조, 복합 모니터에 대해 사용 가능한 요청 필드 목록, 사용 가능한 트리거 및 작업 변수에 대한 설명을 포함한 전체 알림 관련 설명서는 OpenSearch 설명서의 Alerting을 참조하세요.

주제

- 알림 권한
- 알림 시작하기
- 알림
- 차이

알림 권한

알림은 세분화된 액세스 제어를 지원합니다. 사용 사례에 맞게 권한을 혼합하고 일치시키는 방법에 대한 자세한 내용은 OpenSearch 설명서의 알림 보안을 참조하세요.

OpenSearch 대시보드의 알림 페이지에 액세스하려면 최소한 `alerting_read_access`의 사전 정의된 역할에 매핑되거나 이와 동등한 권한이 부여되어야 합니다. 이 역할은 알림, 대상 및 모니터를 볼 수 있는 권한을 부여하지만 알림을 승인하거나 대상 또는 모니터를 수정할 수는 없습니다.

알림 시작하기

알림을 생성하려면 정의된 일정에 따라 실행되고 OpenSearch 인덱스를 쿼리하는 작업인 모니터를 구성합니다. 또한 이벤트를 생성하는 조건을 정의하는 하나 이상의 트리거를 구성합니다. 마지막으로 알림이 트리거된 후 수행되는 작업을 구성합니다.

알림 시작하기

1. OpenSearch Dashboards 기본 메뉴에서 Alerting(알림)을 선택하고 Create monitor(모니터 생성)를 선택합니다.
2. 쿼리별, 버킷별, 클러스터별 지표 또는 문서별 모니터를 생성합니다. 지침은 모니터 생성을 참조하세요.
3. Triggers(트리거)의 경우 하나 이상의 트리거를 생성합니다. 지침은 트리거 생성을 참조하세요.
4. Actions(작업)의 경우 알림에 대한 notification channel(알림 채널)을 설정합니다. Slack, Amazon Chime, 사용자 지정 Webhook 또는 Amazon SNS 중에서 선택할 수 있습니다. 알림을 받으려면 채널에 연결되어야 합니다. 예를 들어 Slack 채널에 알리거나 사용자 지정 웹후크를 타사 서버로 전송하려면 OpenSearch Service 도메인에 적절한 네트워크 연결이 필요합니다. 사용자 지정 Webhook에 알림을 보내려면 OpenSearch Service 도메인에 퍼블릭 IP 주소가 있어야 합니다.

> **Tip**
>
> 작업이 메시지를 성공적으로 전송한 후 해당 메시지에 대한 액세스(예: Slack 채널에 대한 액세스)를 보호하는 것은 사용자의 책임입니다. 도메인에 민감한 데이터가 포함된 경우 작업 없이 트리거를 사용하고 정기적으로 Dashboards에서 알림을 확인하는 것이 좋습니다.

알림

알림 기능은 OpenSearch 알림을 위한 통합 시스템인 알림과 통합됩니다. 알림을 통해 사용하려는 통신 서비스를 구성하고 관련 통계 및 문제 해결 정보를 볼 수 있습니다. 포괄적인 설명서를 보려면 OpenSearch 설명서의 알림을 참조하세요.

알림을 사용하려면 도메인에서 OpenSearch 2.3 또는 이후 버전을 실행해야 합니다.

> **Note**
>
> OpenSearch 알림은 서비스 소프트웨어 업데이트, 자동 조정 기능 향상, 기타 중요한 도메인 수준 정보에 대한 세부 정보를 제공하는 OpenSearch Service 알림과는 별개입니다. OpenSearch 알림은 플러그인별로 다릅니다.

알림 채널은 OpenSearch 버전 2.0부터 알림 대상을 대체했습니다. 대상은 공식적으로 지원 중단되었으며, 모든 알림은 앞으로 채널을 통해 관리될 예정입니다.

(2.x에 대한 OpenSearch Service 지원은 2.3부터 시작되므로) 도메인을 2.3 또는 이후 버전으로 업그레이드하면 기존 대상이 자동으로 알림 채널로 마이그레이션됩니다. 대상이 마이그레이션에 실패하면 모니터가 알림 채널로 마이그레이션될 때까지 모니터는 해당 대상을 계속 사용합니다. 자세한 내용은 OpenSearch 설명서의 대상 관련 질문을 참조하세요.

알림을 시작하려면 OpenSearch 대시보드에 로그인하고 Notifications(알림), Channels(채널), Create channel(채널 생성)을 선택합니다.

Amazon Simple Notification Service (Amazon SNS)는 알림에 지원되는 채널 유형입니다. 사용자를 인증하려면 사용자에게 Amazon SNS에 대한 전체 액세스 권한을 제공하거나 Amazon SNS에 액세스할 권한이 있는 IAM 역할을 맡도록 해야 합니다. 지침은 채널 유형으로서의 Amazon SNS를 참조하세요.

차이

OpenSearch의 오픈 소스 버전과 비교했을 때, Amazon OpenSearch Service의 알림에는 몇 가지 주목할 만한 차이점이 있습니다.

알림 설정

OpenSearch Service를 사용하면 다음 알림 설정을 수정할 수 있습니다.

- `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

다른 모든 설정은 변경할 수 없는 기본값을 사용합니다.

알림을 비활성화하려면 다음 요청을 전송합니다.

```
PUT _cluster/settings
{
  "persistent" : {
    "plugins.scheduled_jobs.enabled" : false
  }
}
```

다음 요청은 기본 30일 대신 7일 후에 기록 인덱스를 자동으로 삭제하도록 알림을 구성합니다.

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.alerting.alert_history_retention_period": "7d"
  }
}
```

이전에 모니터를 생성한 경우 일일 알림 인덱스 생성을 중지하려면 알림 기록 인덱스를 모두 삭제합니다.

```
DELETE .plugins-alerting-alert-history-*
```

기록 인덱스의 샤드 수를 줄이려면 인덱스 템플릿을 생성합니다. 다음 요청은 기록 인덱스를 하나의 샤드와 하나의 복제본으로 설정합니다.

```
PUT _index_template/template-name
{
```

```
  "index_patterns": [".opendistro-alerting-alert-history-*"],
  "template": {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 1
    }
  }
}
```

데이터 손실에 대한 허용치에 따라 0 복제본을 사용하는 것을 고려할 수도 있습니다. 인덱스 템플릿 생성 및 관리에 대한 자세한 내용은 OpenSearch 설명서의 인덱스 템플릿을 참조하세요.

Amazon OpenSearch Service의 이상 탐지

Amazon OpenSearch Service의 이상 탐지 기능은 Random Cut Forest(RCF) 알고리즘을 사용하여 거의 실시간으로 OpenSearch 데이터에서 이상을 자동으로 탐지합니다. RCF는 수신 데이터 스트림의 스케치를 모델링하는 비지도 기계 학습 알고리즘입니다. 알고리즘은 수신 데이터 포인트마다 `anomaly grade` 및 `confidence score` 값을 계산합니다. 이상 탐지는 이러한 값을 사용하여 데이터의 정상적인 변이와 이상을 구분합니다.

알림 플러그인과 이상 탐지 플러그인과 페어링하여 이상이 탐지되는 즉시 알림을 받을 수 있습니다.

이상 탐지는 모든 OpenSearch 버전 또는 Elasticsearch 7.4 이상을 실행하는 도메인에서 사용할 수 있습니다. `t2.micro` 및 `t2.small`을 제외한 모든 인스턴스 유형이 이상 탐지를 지원합니다.

> **Note**
>
> 이 설명서에서는 Amazon OpenSearch Service의 컨텍스트에서 이상 탐지에 대한 간략한 개요를 제공합니다. 세부 단계, API 참조, 사용 가능한 모든 설정 참조, 시각화 및 대시보드를 생성하는 단계를 포함한 포괄적인 설명서는 오픈 소스 OpenSearch 설명서의 Anomaly detection을 참조하세요.

사전 조건

이상 탐지의 사전 조건은 다음과 같습니다.

- 이상 탐지에는 OpenSearch 또는 Elasticsearch 7.4 이상이 필요합니다.

- 이상 탐지는 Elasticsearch 버전 7.9 이상 및 OpenSearch의 모든 버전에서만 세분화된 액세스 제어를 지원합니다. Elasticsearch 7.9 이전 버전의 경우 관리자만 탐지기를 생성, 확인 및 관리할 수 있습니다.
- 도메인에서 세분화된 액세스 제어를 사용하는 경우 관리자가 아닌 사용자는 OpenSearch Dashboards에서 `anomaly_read_access` 역할에 매핑되어 탐지기를 보거나 `anomaly_full_access`에 매핑되어 탐지기를 생성하고 관리할 수 있습니다.

이상 탐지 시작하기

시작하려면 OpenSearch Dashboards에서 이상 탐지(Anomaly Detection)를 선택합니다.

1단계: 탐지기 생성

탐지기는 개별 이상 탐지 태스크입니다. 여러 탐지기를 생성할 수 있으며, 모든 탐지기가 서로 다른 소스의 각 분석 데이터에 대해 동시에 실행할 수 있습니다.

2단계: 탐지기에 기능 추가

기능은 이상이 있는지 확인하는 인덱스 필드입니다. 탐지기는 하나 이상의 기능에서 이상을 검색할 수 있습니다. 각 기능(`average()`, `sum()`, `count()`, `min()` 또는 `max()`)에 대해 다음 집계 중 하나를 선택해야 합니다.

> **Note**
>
> 이 `count()` 집계 방법은 OpenSearch 및 Elasticsearch 7.7 이상에서만 사용할 수 있습니다. Elasticsearch 7.4의 경우 다음과 같은 사용자 지정 표현식을 사용합니다.
>
> ```
> {
>   "aggregation_name": {
>      "value_count": {
>         "field": "field_name"
>      }
>   }
> }
> ```

집계 방법에 따라 이상을 구성하는 요소가 결정됩니다. 예를 들어, `min()`을 선택한 경우 탐지기는 기능의 최소값을 기준으로 이상을 찾는 데 초점을 맞춥니다. `average()`를 선택하면 탐지기가 기능의 평균값을 기준으로 이상을 찾습니다. 탐지기당 최대 5개의 기능을 추가할 수 있습니다.

다음과 같은 선택적 설정을 구성할 수 있습니다(Elasticsearch 7.7 이상에서 사용 가능).

- 범주 필드 - IP 주소, 제품 ID, 국가 코드 등과 같은 차원으로 데이터를 분류하거나 분할할 수 있습니다.
- 창 크기 - 검색 창에서 고려할 데이터 스트림의 집계 간격 수를 설정합니다.

기능을 설정한 후 샘플 이상을 미리 보고 필요한 경우 기능 설정을 조정합니다.

3단계: 결과 관찰

- 라이브 이상(Live anomalies) - 지난 60개 간격 동안의 라이브 이상 결과를 표시합니다. 예를 들어, 간격이 10으로 설정된 경우 지난 600분 동안의 결과를 표시합니다. 이 차트는 30초마다 새로 고쳐집니다.
- 이상 기록(Anomaly history) - 해당 신뢰도 척도와 함께 이상 등급을 플롯합니다.
- 기능 분석(Feature breakdown) - 집계 방법을 기준으로 기능을 플롯합니다. 탐지기의 날짜-시간 범위를 변경할 수 있습니다.
- 이상 발생(Anomaly occurrence) - 탐지된 각 이상에 대한 `Start time`, `End time`, `Data confidence` 및 `Anomaly grade`이 표시됩니다.

  범주 필드를 설정하면 추가 열 지도 차트에서 비정상적인 항목에 대한 결과의 상관관계를 분석합니다. 채워진 직사각형을 선택하면 이상에 대해 세부적으로 볼 수 있습니다.

4단계: 알림 설정

이상이 탐지될 때 알림을 보낼 모니터를 생성하려면 알림 설정(Set up alert)을 선택합니다. 플러그인은 알림을 구성할 수 있는 모니터 추가 페이지로 리디렉션합니다.

자습서: 이상 탐지로 높은 CPU 사용량 탐지

이 자습서에서는 Amazon OpenSearch Service에서 이상 탐지기를 생성하여 높은 CPU 사용량을 탐지하는 방법을 설명합니다. OpenSearch Dashboards로 탐지기를 구성하여 CPU 사용량을 모니터링하고 CPU 사용량이 지정된 임계값을 초과할 때 알림을 생성합니다.

> **Note**
>
> 이 단계는 최신 버전의 OpenSearch에 적용되며 이전 버전에서는 약간 다를 수 있습니다.

사전 조건

- Elasticsearch 7.4 이상 또는 OpenSearch(버전 무관)을 실행하는 OpenSearch 서비스 도메인이 있어야 합니다.
- CPU 사용량 데이터가 포함된 애플리케이션 로그 파일을 클러스터로 모아야 합니다.

1단계: 탐지기 생성

먼저 CPU 사용량 데이터에서 이상을 식별하는 탐지기를 생성합니다.

1. OpenSearch Dashboards에서 왼쪽 패널 메뉴를 열고 Anomaly Detection(이상 탐지)을 선택한 다음 Create detector(탐지기 생성)를 선택합니다.
2. 탐지기 이름을 **high-cpu-usage**로 지정합니다.
3. 이상을 식별하려는 CPU 사용량 로그 파일이 들어 있는 인덱스를 데이터 소스로 선택합니다.
4. 데이터에서 Timestamp field(타임스탬프 필드)를 선택합니다. 필요에 따라 데이터 필터를 추가할 수 있습니다. 이 데이터 필터는 데이터 소스의 하위 세트만 분석하고 관련이 없는 데이터에서 발생하는 노이즈를 줄입니다.
5. Detector interval(탐지기 간격)을 2분으로 설정합니다. 이 간격은 탐지기가 데이터를 수집하는 시간을 분 간격으로 정의합니다.
6. Window delay(기간 지연)에서 1-minute(1분) 지연을 추가합니다. 이 지연은 기간 내의 모든 데이터가 있는지 확인하기 위해 추가 처리 시간을 더합니다.
7. 다음을 선택합니다. 이상 탐지 대시보드의 탐지기 이름 아래에서 Configure model(모델 구성)을 선택합니다.
8. Feature name(기능 이름)에 **max_cpu_usage**를 입력합니다. Feature state(기능 상태)에서 Enable feature(기능 활성화)를 선택합니다.
9. Find anomalies based on(다음을 기준으로 이상 찾기)에서 Field value(필드 값)를 선택합니다.
10. Aggregation method(집계 방법)에서 **max()**를 선택합니다.
11. Field(필드)에서 이상이 있는지 확인할 데이터의 필드를 선택합니다. 예를 들어 `cpu_usage_percentage`라고 할 수 있습니다.
12. 다른 모든 설정을 기본값으로 유지하고 Next(다음)를 선택합니다.
13. 탐지기 작업 설정을 무시하고 Next(다음)를 선택합니다.
14. 팝업 창에서 탐지기를 시작할 시간(자동 또는 수동)을 선택한 다음 Confirm(확인)을 선택합니다.

탐지기가 구성되었으므로 초기화 후 탐지기 패널의 Real-time results(실시간 결과) 섹션에서 CPU 사용량의 실시간 결과를 볼 수 있습니다. Live anomalies(라이브 이상) 섹션에는 데이터를 실시간으로 모을 때 발생하는 이상이 표시됩니다.

2단계: 알림 구성

탐지기를 생성했으므로 탐지기 설정에 지정된 조건을 충족하는 CPU 사용량을 탐지할 때 Slack에 메시지를 전송하도록 알림을 호출하는 모니터를 생성합니다. 하나 이상의 인덱스의 데이터가 알림을 호출하는 조건을 충족하면 Slack 알림을 받게 됩니다.

1. OpenSearch Dashboards에서 왼쪽 패널 메뉴를 열고 Alerting(알림)을 선택한 다음 Create monitor(모니터 생성)를 선택합니다.
2. 모니터의 이름을 제공합니다.
3. Monitor type(모니터 유형)에서 Per-query monitor(쿼리별 모니터)를 선택합니다. 쿼리별 모니터는 지정된 쿼리를 실행하고 트리거를 정의합니다.
4. Monitor defining method(모니터 정의 방법)에서 Anomaly detector(이상 탐지기)를 선택한 다음 Detector(탐지기) 드롭다운 메뉴에서 이전 섹션에서 생성한 탐지기를 선택합니다.
5. Schedule(일정)에서 모니터가 데이터를 수집하는 빈도와 알림을 받는 빈도를 선택합니다. 본 자습서의 목적에 맞게 7분마다 실행되도록 일정을 설정합니다.
6. Triggers(트리거) 섹션에서 Add trigger(트리거 추가)를 선택합니다. Trigger name(트리거 이름)에 **`High CPU usage`**를 입력합니다. 본 자습서에서는 Severity level(심각도 수준)로 1(가장 높은 심각도 수준)을 선택합니다.
7. Anomaly grade threshold(이상 등급 임계값)에서 IS ABOVE(초과)를 선택합니다. 그 아래 메뉴에서 적용할 등급 임계값을 선택합니다. 본 자습서에서는 Anomaly grade(이상 등급)를 0.7로 설정합니다.
8. Anomaly confidence threshold(이상 신뢰도 임계값)에서 IS ABOVE(초과)를 선택합니다. 그 아래 메뉴에서 이상 등급과 동일한 숫자를 입력합니다. 본 자습서에서는 Anomaly confidence threshold(이상 신뢰도 임계값)를 0.7로 설정합니다.
9. Actions(작업) 섹션에서 Destination(대상)을 선택합니다. Name(이름) 필드에서 대상의 이름을 선택합니다. Type(유형) 메뉴에서 Slack을 선택합니다. Webhook URL(웹후크 URL) 필드에 알림을 수신할 웹후크 URL을 입력합니다. 자세한 내용은 Sending messages using incoming webhooks(수신 웹후크를 사용하여 메시지 전송)를 참조하세요.
10. 생성(Create)을 선택합니다.

관련 리소스

- the section called “알림”

- the section called “이상 탐지”
- Anomaly detection API(이상 탐지 API)

Amazon OpenSearch Service용 Amazon Q Developer

Amazon Q Developer는 개발자와 IT 전문가가 소프트웨어 개발 수명 주기 전반에 걸쳐 다양한 작업을 수행할 수 있도록 지원하는 생성형 AI 기반 어시스턴트입니다. Amazon Q는 AWS 고객이에서 실행되는 애플리케이션을 코딩, 테스트, 배포, 문제 해결 및 최적화할 수 있도록 지원합니다 AWS.

Amazon OpenSearch Service와의 Amazon Q 통합은 다음과 같은 생성 기능을 제공합니다.

- the section called “자연어를 사용하여 시각화 생성”
- the section called “알림 요약 및 인사이트 보기”
- the section called “검색 페이지에서 Amazon Q 생성 쿼리 결과 요약 보기”
- the section called “권장 이상 탐지기 보기”
- the section called “OpenSearch Service 질문을 위한 Amazon Q 채팅 액세스”

작업과 관련된 OpenSearch Service의 Amazon Q Developer 기능에 액세스하려면 알림, 검색 및 시각화 생성 페이지에서 다음 컨텍스트 내 아이콘을 찾습니다. 아이콘을 클릭하여 지원을 요청하고이 섹션에 설명된 기능을 사용합니다.

OpenSearch Service 콘솔의 오른쪽 상단에 있는 아이콘을 클릭하여 Amazon Q와 직접 채팅할 수도 있습니다. Amazon Q 채팅은 OpenSearch Service 기능에 대한 질문에 대한 답변을 지원합니다.

> **Note**
>
> Amazon Q 채팅은 데이터에 액세스할 수 없습니다. 따라서 챗봇을 데이터에 대한 대화에 참여시킬 수 없습니다.

Amazon Q에 대한 자세한 내용은 Amazon Q 사용 설명서의 Amazon Q Developer란 무엇입니까?를 참조하세요.

지원됨 AWS 리전

OpenSearch Service에 대한 Amazon Q 지원은 AWS 리전다음에서 사용할 수 있습니다.

- 미국 동부(버지니아 북부)
- 미국 서부(오리건)

- 아시아 태평양(뭄바이)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(런던)
- 유럽(파리)
- 남아메리카(상파울루)

OpenSearch Service용 Amazon Q 설정

다음 단계를 완료하여 OpenSearch Service용 Amazon Q를 설정합니다.

1. OpenSearch Service 세분화된 액세스 제어를 구성했는지 확인합니다. 자세한 내용은 Amazon OpenSearch Service에서 세분화된 액세스 제어 단원을 참조하십시오.
2. 데이터 소스가 OpenSearch 버전 2.17 이상인지 확인합니다.
3. 도메인 생성 중에 또는 클러스터 구성을 편집하여 인공 지능(AI) 및 Machine Learning(ML) 섹션에서 자연어 쿼리 생성 활성화 확인란을 선택했는지 확인합니다.

> **Note**
>
> Amazon Q for Developer OpenSearch 기능은 Q 프리 티어에서 사용할 수 있습니다. 자세한 내용은 Amazon Q 사용 설명서의 서비스 계층 이해를 참조하세요.

자연어를 사용하여 시각화 생성

운영 데이터에 대한 추가 인사이트를 얻을 수 있도록 Amazon Q Developer for OpenSearch Service는 자연어 프롬프트를 사용하여 시각화를 구축하도록 지원합니다. 시각화 페이지 또는 검색 페이지에서 자연어를 사용하여 다음 예제와 같은 시각화를 생성할 수 있습니다.

시각화는 다양한 차원에 따라 오류 분석을 분석하여 문제 해결을 가속화할 수 있습니다. 또한 시각화를 통해 패턴과 추세를 식별하고, 의사 결정 속도를 높이고, 관계를 파악하고, 복잡한 데이터를 간소화할 수 있습니다.

자연어를 사용하여 시각화를 생성하려면

1. OpenSearch Service용 Amazon Q를 설정했는지 확인합니다.
2. OpenSearch Dashboards 기본 메뉴에서 검색 페이지를 선택한 다음 데이터 소스를 선택합니다.
3. Amazon Q 메뉴에서 다음 스크린샷과 같이 시각화 생성을 선택합니다.

4. 검색 페이지에 자연어 쿼리를 입력한 경우 Amazon Q는 해당 쿼리를 기반으로 시각화를 생성할 때 컨텍스트를 복사합니다. 자연어 쿼리를 아직 입력하지 않은 경우 Amazon Q 텍스트 상자에 프롬프트를 입력한 다음 텍스트 상자 옆의 버튼을 클릭하여 쿼리를 실행합니다. Amazon Q에서 시각화를 생성하는 데 몇 초 정도 걸릴 수 있습니다.
5. 시각적 객체를 업데이트하려면 시각적 객체의 오른쪽 상단에 있는 시각적 객체 편집 버튼을 선택합니다. "줄 그래프로 변경"과 같은 새 프롬프트를 입력한 다음 적용을 선택합니다.

알림 요약 및 인사이트 보기

하나 이상의 인덱스의 데이터가 특정 조건을 충족할 때 알림 모니터를 생성하도록 OpenSearch Service를 구성할 수 있습니다. 알림을 빠르게 이해하고 문제를 해결하는 데 도움이 되도록 알림 옆에 있는 Amazon Q Developer 아이콘을 클릭하여 알림 요약을 볼 수 있습니다. 요약은 알림을 트리거한 기본 문제에 대한 세부 정보와 가능한 경우 문제의 근본 원인을 찾는 데 도움이 되는 추가 분석을 제공합니다. 다음 스크린샷은 Amazon Q에서 생성한 알림 요약의 예를 보여줍니다.

이 주제의 뒷부분에서 설명한 대로 지식 기반을 연결하여 환경에 대한 추가 컨텍스트를 제공하는 경우 Amazon Q는 알림에 대한 인사이트를 생성합니다. Insights는 알림의 근본 원인을 해결하는 데 도움이 되는 세부 정보 및 문제 해결 옵션을 제공합니다. 이전에 표시된 알림에 대해 Amazon Q는 다음과 같은 인사이트도 생성했습니다.

> **Note**
>
> 알림의 특성과 사용 가능한 정보에 따라 Amazon Q는 OpenSearch 대시보드의 검색 페이지에서 알림 데이터를 볼 수 있는 옵션을 제공할 수 있습니다. Amazon Q 알림 요약 하단에 검색에서 보기 버튼이 표시되면 버튼을 클릭하여 알림 데이터에 대한 활성 필터가 있는 검색에서 해당 데이터 세트를 엽니다.

주제

- 시작하기 전 준비 사항
- 알림 요약 및 인사이트 보기

시작하기 전 준비 사항

Amazon Q가 OpenSearch Service 알림에 대한 인사이트를 생성할 수 있도록 Amazon Bedrock 지식 기반을 구성하려면 다음 단계를 완료하세요.

1단계: LambdaInvokeOpenSearchMLCommonsRole IAM 역할 생성

AWS Identity and Access Management (IAM)`LambdaInvokeOpenSearchMLCommonsRole`에서 라는 새 역할을 생성합니다. OpenSearch Service는이 역할을 사용하여 OpenSearch에서 구성된 지식 기반 문서를 기반으로 인사이트를 생성하는 데 도움이 되는 AI 커넥터를 생성합니다. 2단계에 설명된 대로이 역할을 OpenSearch Service `ml_full_access` 역할에 매핑해야 합니다.

새 역할을 생성할 때 신뢰할 수 있는 엔터티 유형에서 AWS 계정을 선택합니다. 권한 정책을 지정할 필요가 없습니다. 권한 추가 페이지에서 다음을 선택합니다. 새 역할 생성에 대한 자세한 내용은 AWS 서비스에 대한 역할 생성(콘솔)을 참조하세요.

2단계: LambdaInvokeOpenSearchMLCommonsRole 역할을 OpenSearch Service ml_full_access 역할에 매핑

다음 절차에 따라 `LambdaInvokeOpenSearchMLCommonsRole` 역할을 OpenSearch Service `ml_full_access` 역할에 매핑합니다. 또한이 매핑은 OpenSearch Service가 AI 커넥터를 생성하는 데 도움이 됩니다.

필요한 IAM 역할을 OpenSearch Service ml_full_access 역할에 매핑하려면

1. OpenSearch Service Dashboard Data 관리 페이지를 엽니다.
2. 데이터 액세스 및 사용자에서 역할을 선택합니다.
3. 검색 상자를 사용하여 `ml_full_access` 역할을 찾습니다.
4. ml_full_access 페이지에서 매핑된 사용자 탭을 선택합니다.
5. 사용자 매핑을 선택합니다.
6. 백엔드 역할 필드에 `LambdaInvokeOpenSearchMLCommonsRole` 역할의 Amazon 리소스 이름(ARN)을 붙여넣은 다음 맵을 선택합니다.

3단계:를 사용하여 OpenSearch Service 지식 기반 구성 AWS CloudFormation

Amazon Q가 인사이트를 생성할 수 AWS CloudFormation 있도록 다음 절차에 따라를 사용하여 OpenSearch Service 지식 기반을 구성합니다.

인사이트를 위한 지식 기반을 구성하려면

1. 지원되는에서 Amazon OpenSearch Service 콘솔 https://console.aws.amazon.com/aos/home 로 그인합니다 AWS 리전. 자세한 내용은 지원됨 AWS 리전 단원을 참조하십시오.
2. 탐색 창에서 통합을 선택합니다.
3. 통합 템플릿 섹션에서 Amazon Bedrock 템플릿을 통해 지식 기반과 통합을 선택합니다. 이 템플릿이 표시되지 않으면 지원되는 리전에 있는지 확인합니다.
4. Amazon Bedrock을 통해 지식 기반과 통합 타일에서 도메인 구성을 선택한 다음 사용 가능한 옵션 중 하나를 선택합니다. OpenSearch Service는 필수 필드가 미리 채워진 AWS CloudFormation 스택 템플릿을 엽니다. AWS CloudFormation 스택은 퍼블릭 도메인과 VPC 도메인의 통합을 지원합니다.
5. 스택 생성을 선택합니다. 가 리소스를 AWS CloudFormation 생성하면 서비스에 Amazon Bedrock 에이전트 AgentId, ConnectorId 및 ModelId가 표시됩니다.

해당하는 경우 Amazon Q는 이제 OpenSearch Service 알림에 대한 인사이트를 생성합니다.

알림 요약 및 인사이트 보기

OpenSearch Service에서 알림 요약 및 인사이트를 보려면 다음 절차를 따르세요.

알림 요약 및 인사이트 보기

1. Amazon Q for OpenSearch Service를 설정했는지 확인합니다.
2. OpenSearch Service에 대한 알림을 설정했는지 확인합니다.
3. OpenSearch Dashboards 기본 메뉴에서 알림을 선택한 다음 알림을 선택합니다.
4. 알림 옆에 있는 Amazon Q 아이콘을 선택합니다. Amazon Q는 요약을 생성하는 데 최대 10초가 걸릴 수 있습니다.
5. 알림 요약에 있는 경우 인사이트 보기를 선택하여 구성된 지식 기반을 기반으로 알림에 대한 자세한 내용을 봅니다.
6. 알림 요약에 있는 경우 검색에서 보기를 선택하여 OpenSearch 대시보드의 검색 페이지에서 알림 데이터를 봅니다.

검색 페이지에서 Amazon Q 생성 쿼리 결과 요약 보기

OpenSearch Service를 사용하면 검색 페이지의 파이프 처리 언어(PPL) 쿼리 언어를 사용하여 자연어 프롬프트로 데이터를 쿼리할 수 있습니다. 예를 들어 다음과 같은 쿼리를 작성할 수 있습니다.

- 오류 로그에 오류가 있습니까?
- 주별 평균 요청 크기는 얼마입니까?
- 지난 주에 응답 코드별로 그룹이 있는 요청은 몇 개였나요?

이에 대한 응답으로 Amazon Q는 다음과 같이 처음 10개의 레코드를 기반으로 쿼리 결과의 자연어 요약을 생성합니다.

자연어 쿼리 생성과 쿼리 요약을 조합하면 복잡한 쿼리를 작성할 필요 없이 알림 문제를 해결할 때 또는 사용자 친화적인 데이터 이해 수단으로 추가 수준의 쿼리를 추가할 수 있습니다.

검색 페이지에서 Amazon Q 생성 쿼리 결과 요약을 보려면

1. Amazon Q for OpenSearch Service를 설정했는지 확인합니다.
2. OpenSearch Dashboards 기본 메뉴에서 검색을 선택합니다.
3. 쿼리 언어 드롭다운 목록에서 PPL을 선택합니다.
4. Amazon Q 텍스트 상자에 프롬프트를 입력한 다음 텍스트 상자 옆에 있는 버튼을 클릭하여 쿼리를 실행합니다. Amazon Q가 요약을 반환하는 데 최대 10초가 걸릴 수 있습니다. 초기 쿼리 후 후속 요약에 대한 요약 생성을 선택해야 합니다.

> **Note**
>
> 검색의 Amazon Q 드롭다운 목록에서 요약 생성을 끌 수 있습니다.

권장 이상 탐지기 보기

Amazon OpenSearch Service의 이상 탐지 기능은 Random Cut Forest(RCF) 알고리즘을 사용하여 거의 실시간으로 OpenSearch 데이터에서 이상을 자동으로 탐지합니다. RCF는 수신 데이터 스트림의 스케치를 모델링하는 비지도 기계 학습 알고리즘입니다. 알고리즘은 수신 데이터 포인트마다 `anomaly`

grade 및 confidence score 값을 계산합니다. 이상 탐지는 이러한 값을 사용하여 데이터의 정상적인 변이와 이상을 구분합니다.

이상 탐지기 생성 프로세스를 간소화하기 위해 Amazon Q는 검색 페이지에서 선택한 데이터 소스를 기반으로 제안된 탐지기를 생성할 수 있습니다. Amazon Q는 모든 언어에 대해 제안된 이상 탐지기를 지원합니다.

Amazon Q 권장 이상 탐지기를 보려면

1. Amazon Q for OpenSearch Service를 설정했는지 확인합니다.
2. OpenSearch Dashboards 기본 메뉴에서 검색 페이지를 선택한 다음 데이터 소스를 선택합니다.
3. Amazon Q 메뉴에서 다음 스크린샷과 같이 이상 탐지기 제안을 선택합니다.

   Amazon Q가 감지기에 대한 기능을 생성하는 데 몇 초 정도 걸릴 수 있습니다.
4. 감지기 생성을 선택합니다.

OpenSearch 관련 질문에 대한 Amazon Q 채팅 액세스

OpenSearch Service 기능과 관련된 개념적 또는 절차적 질문을 포함하여 Amazon OpenSearch Service에 대한 질문이 있는 경우 Amazon Q Developer에 정보를 요청할 수 있습니다. 챗봇은 토론의 컨텍스트를 유지하는 대화형 채팅을 지원합니다. 또한 나중에 참조할 수 있도록 대화 기록을 저장합니다. 챗봇에 액세스하려면 OpenSearch Service 페이지의 오른쪽 상단에 있는 Amazon Q 아이콘을 클릭합니다. 텍스트 상자에 질문을 입력하고 이동을 클릭합니다. Amazon Q에서 응답을 생성하는 데 최대 30초가 걸릴 수 있습니다.

다음은 Amazon Q의 응답 예제입니다.

> **Note**
>
> Amazon Q 채팅은 데이터에 액세스할 수 없습니다. 따라서 챗봇을 데이터에 대한 대화에 참여시킬 수 없습니다.

Amazon OpenSearch Service용 기계 학습

ML Commons는 전송 및 REST API 호출을 통해 공통 기계 학습(ML) 알고리즘 세트를 제공하는 OpenSearch 플러그인입니다. 이러한 직접적 호출은 각 ML 요청에 적합한 노드와 리소스를 선택하고 ML 작업을 모니터링하여 가동 시간을 보장합니다. 이를 통해 기존 오픈 소스 ML 알고리즘을 활용하고 새로운 ML 기능을 개발하는 데 필요한 노력을 줄일 수 있습니다. 플러그인에 대한 자세한 내용은 OpenSearch 설명서의 Machine learning을 참조하세요. 이 섹션에서는 Amazon OpenSearch Service에서 플러그인을 사용하는 방법을 다룹니다.

주제

- AWS 서비스에 대한 Amazon OpenSearch Service ML 커넥터
- 서드파티 플랫폼에 대한 Amazon OpenSearch Service ML 커넥터
- AWS CloudFormation 를 사용하여 의미 체계 검색을 위한 원격 추론 설정
- 지원되지 않는 ML Commons 설정
- OpenSearch Service 흐름 프레임워크 템플릿

AWS 서비스에 대한 Amazon OpenSearch Service ML 커넥터

Amazon OpenSearch Service 기계 학습(ML) 커넥터를 다른 커넥터와 함께 사용하는 경우 OpenSearch Service를 해당 서비스에 안전하게 연결하도록 IAM 역할을 설정해야 AWS 서비스합니다. AWS 서비스 Amazon SageMaker AI 및 Amazon Bedrock을 포함하도록 커넥터를 설정할 수 있습니다. 이 자습서에서는 OpenSearch Service에서 SageMaker 런타임으로 커넥터를 생성하는 방법을 다룹니다. 커넥터에 대한 자세한 내용은 지원되는 커넥터를 참조하세요.

주제

- 사전 조건
- OpenSearch Service 커넥터 생성

사전 조건

커넥터를 생성하려면 Amazon SageMaker AI 도메인 엔드포인트와 OpenSearch Service 액세스 권한을 부여하는 IAM 역할이 있어야 합니다.

Amazon SageMaker AI 도메인 설정

기계 학습 모델을 배포하려면 Amazon SageMaker AI 개발자 안내서의 Amazon SageMaker AI에서 모델 배포를 참조하세요. AI 커넥터를 생성하는 데 필요한 모델의 엔드포인트 URL을 기록하세요.

IAM 역할 생성

OpenSearch Service에 대한 SageMaker 런타임 권한을 위임할 IAM 역할을 설정합니다. 새 역할을 생성하려면 IAM 사용 설명서의 IAM 역할 생성(콘솔)을 참조하세요. 원하는 경우, 권한이 동일하다면 기존 역할을 사용할 수도 있습니다. AWS 관리형 역할을 사용하는 대신 새 역할을 생성하는 경우이 자습서`opensearch-sagemaker-role`의를 자신의 역할 이름으로 바꿉니다.

1. OpenSearch Service가 SageMaker AI 엔드포인트에 액세스할 수 있도록 새 역할에 다음 관리형 IAM 정책을 연결합니다. 역할에 정책을 연결하는 방법은 IAM 자격 증명 권한 추가를 참조하세요.

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
                   "sagemaker:InvokeEndpointAsync",
                   "sagemaker:InvokeEndpoint"
               ],
               "Effect": "Allow",
               "Resource": "*"
           }
       ]
   }
   ```

2. 역할 신뢰 정책 수정에 나와 있는 지침에 따라 역할의 신뢰 관계를 편집합니다. 다음 정책에서 *service-principle*을 OpenSearch Service 또는 OpenSearch Serverless에 대한 다음 서비스 원칙 중 하나로 바꿉니다.

   OpenSearch Service의 경우

   ```
   opensearchservice.amazonaws.com
   ```

   OpenSearch Serverless의 경우

   ```
   ml.opensearchservice.amazonaws.com
   ```

JSON

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

aws:SourceAccount 및 aws:SourceArn 조건 키를 사용하여 액세스 권한을 특정 도메인으로 제한하는 것이 좋습니다. SourceAccount는 도메인 소유자에 속하는 AWS 계정 ID이고 SourceArn는 도메인의 ARN입니다. 예를 들어 신뢰 정책에 다음 조건 블록을 추가할 수 있습니다.

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

권한 구성

커넥터를 생성하려면 IAM 역할을 OpenSearch Service에 전달할 수 있는 권한이 필요합니다. es:ESHttpPost 작업에도 액세스해야 합니다. 이러한 두 권한을 모두 부여하려면 요청에 서명하기 위해 자격 증명이 사용되는 IAM 역할에 다음 정책을 연결합니다.

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/*"
        }
    ]
}
```

사용자 또는 역할에 역할을 전달할 `iam:PassRole` 권한이 없는 경우 다음 단계에서 리포지토리를 등록하려고 할 때 권한 부여 오류가 발생할 수 있습니다.

세분화된 액세스 제어를 사용하는 경우 OpenSearch Dashboards에서 ML 역할을 매핑할 수 있습니다.

세분화된 액세스 제어를 사용하면 커넥터를 설정할 때 추가 단계가 있습니다. 다른 모든 목적으로 HTTP 기본 인증을 사용하더라도 `opensearch-sagemaker-role`을 전달할 `iam:PassRole` 권한이 있는 IAM 역할에 `ml_full_access` 역할을 매핑해야 합니다.

1. OpenSearch Service 도메인에 대한 OpenSearch 대시보드 플러그인으로 이동합니다. OpenSearch Service 콘솔의 도메인 대시보드에서 Dashboards 엔드포인트를 찾을 수 있습니다.
2. 주 메뉴에서 보안, 역할을 선택하고 ml_full_access 역할을 선택합니다.
3. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다.
4. 백엔드 역할에서 `opensearch-sagemaker-role`을 전달할 권한이 있는 역할의 ARN을 추가합니다.

```
arn:aws:iam::account-id:role/role-name
```

5. Map(맵)을 선택하고 Mapped users(매핑된 사용자)에 사용자 또는 역할이 나타나는지 확인합니다.

OpenSearch Service 커넥터 생성

커넥터를 생성하려면 OpenSearch Service 도메인 엔드포인트로 POST 요청을 보내세요. curl, 샘플 Python 클라이언트, Postman 또는 다른 메서드를 사용하여 서명된 요청을 보낼 수 있습니다. Kibana 콘솔에서는 POST 요청을 사용할 수 없습니다. 요청은 다음과 같은 형식을 사용합니다.

```
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}\" } }"
      }
   ]
}
```

도메인이 Virtual Private Cloud(VPC)에 상주하는 경우, 요청이 AI 커넥터를 생성하려면 컴퓨터가 VPC에 연결되어야 합니다. VPC 액세스는 네트워크 구성에 따라 다르지만, 대개는 VPN 또는 회사 네트워크 연결이 필요합니다. OpenSearch Service 도메인에 도달할 수 있는지 알아보려면 웹 브라우저에서

https://*your-vpc-domain*.*region*.es.amazonaws.com으로 이동하여 기본 JSON 응답을 받을 수 있는지 확인합니다.

샘플 Python 클라이언트

Python 클라이언트는 HTTP 요청보다 자동화가 간단하고 재사용성이 뛰어납니다. Python 클라이언트로 AI 커넥터를 만들려면 다음 샘플 코드를 Python 파일에 저장하세요. 클라이언트에는 AWS SDK for Python (Boto3), `requests`, `requests-aws4auth` 패키지가 필요합니다.

```
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)
```

서드파티 플랫폼에 대한 Amazon OpenSearch Service ML 커넥터

이 자습서에서는 OpenSearch Service에서 Cohere로 연결되는 커넥터를 생성하는 방법을 다룹니다. 커넥터에 대한 자세한 내용은 지원되는 커넥터를 참조하세요.

외부 원격 모델과 함께 Amazon OpenSearch Service 기계 학습(ML) 커넥터를 사용하는 경우 AWS Secrets Manager에 특정 권한 부여 자격 증명을 저장해야 합니다. 이는 API 키 또는 사용자 이름과 암호의 조합일 수 있습니다. 즉, OpenSearch Service가 Secrets Manager에서 읽을 수 있도록 허용하는 IAM 역할도 생성해야 합니다.

주제

- 사전 조건
- OpenSearch Service 커넥터 생성

사전 조건

Cohere 또는 OpenSearch Service를 사용하는 외부 공급자에 대한 커넥터를 생성하려면 자격 증명을 저장하는에 대한 OpenSearch Service 액세스 권한을 부여하는 IAM 역할 AWS Secrets Manager이 있어야 합니다. 또한 보안 인증 정보는 반드시 Secrets Manager에 저장해야 합니다.

IAM 역할 생성

OpenSearch Service에 대한 Secrets Manager 권한을 위임할 IAM 역할을 설정합니다. 기존의 `SecretManagerReadWrite` 역할을 사용해도 됩니다. 새 역할을 생성하려면 IAM 사용 설명서의 IAM 역할 생성(콘솔)을 참조하세요. AWS 관리형 역할을 사용하는 대신 새 역할을 생성하는 경우이 자습서`opensearch-secretmanager-role`의를 자신의 역할 이름으로 바꿉니다.

1. 다음과 같은 관리형 IAM 정책을 새 역할에 연결하여 OpenSearch Service가 Secrets Manager 값에 액세스할 수 있도록 허용합니다.. 역할에 정책을 연결하는 방법은 IAM 자격 증명 권한 추가를 참조하세요.

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
                   "secretsmanager:GetSecretValue"
               ],
               "Effect": "Allow",
               "Resource": "*"
           }
       ]
   }
   ```

2. 역할 신뢰 정책 수정에 나와 있는 지침에 따라 역할의 신뢰 관계를 편집합니다. 다음 정책에서 *service-principle*을 OpenSearch Service 또는 OpenSearch Serverless에 대한 다음 서비스 원칙 중 하나로 바꿉니다.

   OpenSearch Service의 경우

   ```
   opensearchservice.amazonaws.com
   ```

   OpenSearch Serverless의 경우

   ```
   ml.opensearchservice.amazonaws.com
   ```

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
                   "sts:AssumeRole"
               ],
               "Effect": "Allow",
               "Principal": {
                   "Service": [
                       "service-principle"
   ```

```
                ]
            }
        }
    ]
}
```

aws:SourceAccount 및 aws:SourceArn 조건 키를 사용하여 액세스 권한을 특정 도메인으로 제한하는 것이 좋습니다. SourceAccount는 도메인 소유자에 속하는 AWS 계정 ID이고 SourceArn는 도메인의 ARN입니다. 예를 들어 신뢰 정책에 다음 조건 블록을 추가할 수 있습니다.

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

권한 구성

커넥터를 생성하려면 IAM 역할을 OpenSearch Service에 전달할 수 있는 권한이 필요합니다. es:ESHttpPost 작업에도 액세스해야 합니다. 이러한 두 권한을 모두 부여하려면 요청에 서명하기 위해 자격 증명이 사용되는 IAM 역할에 다음 정책을 연결합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam:::role/opensearch-secretmanager-role"
    },
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPost",
      "Resource": "arn:aws:es:us-east-1::domain/domain-name/*"
    }
```

```
  ]
}
```

사용자 또는 역할에 역할을 전달할 `iam:PassRole` 권한이 없는 경우 다음 단계에서 리포지토리를 등록하려고 할 때 권한 부여 오류가 발생할 수 있습니다.

설정 AWS Secrets Manager

인증 자격 증명을 Secrets Manager에 저장하려면 AWS Secrets Manager 사용 설명서의 AWS Secrets Manager 암호 만들기를 참조하세요.

Secrets Manager가 키-값 쌍을 암호로 수락하면 `arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3`와 같은 형식의 ARN을 받게 됩니다. 사용할 때 이 ARN을 기록하고 다음 단계에서 커넥터를 만들 때 키를 기록해 두십시오.

세분화된 액세스 제어를 사용하는 경우 OpenSearch Dashboards에서 ML 역할을 매핑할 수 있습니다.

세분화된 액세스 제어를 사용하면 커넥터를 설정할 때 추가 단계가 있습니다. 다른 모든 목적으로 HTTP 기본 인증을 사용하더라도 `opensearch-sagemaker-role`을 전달할 `iam:PassRole` 권한이 있는 IAM 역할에 `ml_full_access` 역할을 매핑해야 합니다.

1. OpenSearch Service 도메인에 대한 OpenSearch 대시보드 플러그인으로 이동합니다. OpenSearch Service 콘솔의 도메인 대시보드에서 Dashboards 엔드포인트를 찾을 수 있습니다.
2. 주 메뉴에서 보안, 역할을 선택하고 ml_full_access 역할을 선택합니다.
3. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다.
4. 백엔드 역할에서 `opensearch-sagemaker-role`을 전달할 권한이 있는 역할의 ARN을 추가합니다.

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. Map(맵)을 선택하고 Mapped users(매핑된 사용자)에 사용자 또는 역할이 나타나는지 확인합니다.

OpenSearch Service 커넥터 생성

커넥터를 생성하려면 OpenSearch Service 도메인 엔드포인트로 POST 요청을 보내세요. curl, 샘플 Python 클라이언트, Postman 또는 다른 메서드를 사용하여 서명된 요청을 보낼 수 있습니다. Kibana 콘솔에서는 POST 요청을 사용할 수 없습니다. 요청은 다음과 같은 형식을 사용합니다.

```
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\" }"
        }
    ]
}
```

이 요청의 요청 본문은 오픈 소스 커넥터 요청의 요청 본문과 두 가지 측면에서 다릅니다. `credential` 필드 내에서 OpenSearch Service가 Secrets Manager에서 읽을 수 있도록 허용하는 IAM 역할에 대한 ARN과 어떤 암호에 대한 ARN을 함께 전달합니다. `headers` 필드에서는 암호 키를 사용하고 ARN에서 가져온 정보라는 사실을 사용하여 암호를 참조합니다.

도메인이 Virtual Private Cloud(VPC)에 상주하는 경우, 요청이 스냅샷 리포지토리를 등록하려면 컴퓨터가 VPC에 연결되어야 합니다. VPC 액세스는 네트워크 구성에 따라 다르지만, 대개는 VPN 또는 회사 네트워크 연결이 필요합니다. OpenSearch Service 도메인에 도달할 수 있는지 알아보려면 웹 브라우저에서 `https://your-vpc-domain.region.es.amazonaws.com`으로 이동하여 기본 JSON 응답을 받을 수 있는지 확인합니다.

샘플 Python 클라이언트

Python 클라이언트는 HTTP 요청보다 자동화가 간단하고 재사용성이 뛰어납니다. Python 클라이언트로 AI 커넥터를 만들려면 다음 샘플 코드를 Python 파일에 저장하세요. 클라이언트에는 AWS SDK for Python (Boto3), `requests`, `requests-aws4auth` 패키지가 필요합니다.

```
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)
```

AWS CloudFormation 를 사용하여 의미 체계 검색을 위한 원격 추론 설정

OpenSearch 버전 2.9부터 원격 추론과 시맨틱 검색을 사용하여 자체 기계 학습(ML) 모델을 호스팅할 수 있습니다. 원격 추론은 ML Commons 플러그인을 사용하여 Amazon SageMaker AI 및 Amazon BedRock과 같은 ML 서비스에서 모델 추론을 원격으로 호스팅하고 ML 커넥터를 사용하여 Amazon OpenSearch Service에 연결할 수 있습니다.

원격 추론을 쉽게 설정할 수 있도록 Amazon OpenSearch Service는 콘솔에서AWS CloudFormation 템플릿을 제공합니다. CloudFormation은 인프라를 코드로 취급하여 AWS 및 타사 리소스를 모델링, 프로비저닝 및 관리할 수 AWS 서비스 있는 입니다.

OpenSearch CloudFormation 템플릿은 모델 프로비저닝 프로세스를 자동화하므로 OpenSearch Service 도메인에서 모델을 쉽게 생성한 다음, 모델 ID를 사용하여 데이터를 수집하고 신경망 검색 쿼리를 실행할 수 있습니다.

OpenSearch Service 버전 2.12 이상에서 신경 희소 인코더를 사용하는 경우 원격으로 배포하는 대신 로컬에서 토큰화 도구 모델을 사용하는 것이 좋습니다. 자세한 내용은 OpenSearch 설명서의 Sparse encoding models를 참조하세요.

주제

- 사전 조건
- Amazon SageMaker AI 템플릿
- Amazon Bedrock 템플릿

사전 조건

CloudFormation Service에서 CloudFormation 템플릿을 사용하려면 다음 사전 조건을 완료하세요.

OpenSearch Service 도메인 설정

CloudFormation 템플릿을 사용하려면 먼저 버전 2.9 이상에서 세분화된 액세스 제어가 활성화된 Amazon OpenSearch Service 도메인을 설정해야 합니다. OpenSearch Service 백엔드 역할을 생성하여 ML Commons 플러그인에 커넥터를 생성할 수 있는 권한을 부여하세요.

CloudFormation 템플릿은 기본 이름 `LambdaInvokeOpenSearchMLCommonsRole`을 사용하여 Lambda IAM 역할을 생성하며, 다른 이름을 선택하려는 경우 기본 이름을 재정의할 수 있습니다. 템플릿이 이 IAM 역할을 생성한 후에는 Lambda 함수에 OpenSearch Service 도메인을 호출할 권한을 부여해야 합니다. 이렇게 하려면 다음 단계에 따라 이름이 `ml_full_access`으로 지정된 역할을 OpenSearch Service 백엔드 역할에 매핑하세요.

1. OpenSearch Service 도메인에 대한 OpenSearch 대시보드 플러그인으로 이동합니다. OpenSearch Service 콘솔의 도메인 대시보드에서 Dashboards 엔드포인트를 찾을 수 있습니다.
2. 주 메뉴에서 보안, 역할을 선택하고 ml_full_access 역할을 선택합니다.
3. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다.
4. 백엔드 역할에서 도메인 호출 권한이 필요한 Lambda 역할의 ARN을 추가합니다.

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. Map(맵)을 선택하고 Mapped users(매핑된 사용자)에 사용자 또는 역할이 나타나는지 확인합니다.

역할을 매핑한 후 도메인의 보안 구성으로 이동하여 Lambda IAM 역할을 OpenSearch Service 액세스 정책에 추가합니다.

AWS 계정에 대한 권한을 활성화합니다.

에는 템플릿에 AWS 서비스 대해 선택한 SageMaker 런타임 또는 Amazon BedRock과 함께 CloudFormation 및 Lambda에 액세스할 수 있는 권한이 AWS 계정 있어야 합니다.

Amazon Bedrock을 사용하는 경우 모델도 등록해야 합니다. 모델을 등록하려면 Amazon Bedrock 사용 설명서의 모델 액세스를 참조하세요.

자체 Amazon S3 버킷을 사용하여 모델 아티팩트를 제공하는 경우, CloudFormation IAM 역할을 S3 액세스 정책에 추가해야 합니다. 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거를 참조하세요.

Amazon SageMaker AI 템플릿

Amazon SageMaker AI CloudFormation 템플릿은 신경 플러그인 및 의미 체계 검색을 설정하기 위해 여러 AWS 리소스를 정의합니다.

먼저 Amazon SageMaker 템플릿을 통해 텍스트 임베딩 모델과 통합을 사용하여 SageMaker 런타임에 텍스트 임베딩 모델을 서버로 배포합니다. 모델 엔드포인트를 제공하지 않는 경우, CloudFormation은 SageMaker Runtime이 Amazon S3에서 모델 아티팩트를 다운로드하고 이를 서버에 배포할 수 있는 IAM 역할을 생성합니다. 엔드포인트를 제공하면 CloudFormation은 Lambda 함수가 OpenSearch Service 도메인에 액세스할 수 있도록 허용하는 IAM 역할을 생성하거나, 역할이 이미 존재하는 경우 역할을 업데이트하고 재사용합니다. 엔드포인트는 ML Commons 플러그인을 통해 ML 커넥터에 사용되는 원격 모델을 제공합니다.

그런 다음, Amazon Sagemaker를 통해 Sparse Encoder와 통합 템플릿을 사용하여 도메인에서 원격 추론 커넥터를 설정한 Lambda 함수를 생성합니다. OpenSearch Service에서 커넥터를 생성한 후 원격 추론은 SageMaker Runtime의 원격 모델을 사용하여 시맨틱 검색을 실행할 수 있습니다. 템플릿은 도메인의 모델 ID를 사용자에게 반환하므로 검색을 시작할 수 있습니다.

Amazon SageMaker AI CloudFormation 템플릿을 사용하려면

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 통합을 선택합니다.
3. 각 Amazon SageMaker AI 템플릿에서 도메인 구성, 퍼블릭 도메인 구성을 선택합니다.
4. CloudFormation 콘솔의 프롬프트에 따라 스택을 프로비저닝하고 모델을 설정합니다.

> **Note**
>
> OpenSearch Service는 VPC 도메인을 구성하기 위해 별도의 템플릿도 제공합니다. 이 템플릿을 사용하는 경우, Lambda 함수에 대한 VPC ID를 제공해야 합니다.

Amazon Bedrock 템플릿

Amazon SageMaker AI CloudFormation 템플릿과 마찬가지로 Amazon Bedrock CloudFormation 템플릿은 OpenSearch Service와 Amazon Bedrock 간에 커넥터를 생성하는 데 필요한 AWS 리소스를 프로비저닝합니다.

먼저, 템플릿은 향후 Lambda 함수가 OpenSearch Service 도메인에 액세스할 수 있도록 허용하는 IAM 역할을 생성합니다. 그런 다음, 템플릿은 Lambda 함수를 생성하며, 이 함수는 도메인이 ML Commons 플러그인을 사용하여 커넥터를 생성하도록 합니다. OpenSearch Service가 커넥터를 생성한 후에는 원격 추론 설정이 완료되고 Amazon Bedrock API 작업을 사용하여 시맨틱 검색을 실행할 수 있습니다.

Amazon Bedrock은 자체 ML 모델을 호스팅하므로 SageMaker Runtime에 모델을 배포할 필요가 없습니다. 대신 템플릿은 Amazon Bedrock의 미리 결정된 엔드포인트를 사용하며 엔드포인트 프로비저닝 단계를 건너뜁니다.

Amazon Bedrock CloudFormation 템플릿을 사용하는 방법

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 통합을 선택합니다.
3. Amazon Bedrock을 통해 Amazon Titan Text Embeddings 모델과 통합에서 도메인 구성, 퍼블릭 도메인 구성을 선택합니다.
4. 프롬프트에 따라 모델을 설정합니다.

> **Note**
>
> OpenSearch Service는 VPC 도메인을 구성하기 위해 별도의 템플릿도 제공합니다. 이 템플릿을 사용하는 경우, Lambda 함수에 대한 VPC ID를 제공해야 합니다.

또한 OpenSearch Service는 Cohere 모델 및 Amazon Titan Multimodal Embeddings 모델에 연결할 수 있는 다음과 같은 Amazon Bedrock 템플릿을 제공합니다.

- `Integration with Cohere Embed through Amazon Bedrock`
- `Integrate with Amazon Bedrock Titan Multi-modal`

지원되지 않는 ML Commons 설정

Amazon OpenSearch Service는 다음 ML Commons 설정 사용을 지원하지 않습니다.

- `plugins.ml_commons.allow_registering_model_via_url`
- `plugins.ml_commons.allow_registering_model_via_local_file`

> **Important**
>
> 프로덕션 클러스터에서는 클러스터 설정을 비활성화하지 마십시오`plugins.ml_commons.only_run_on_ml_node`(로 설정하지 마십시오`false`). 이 보호 기능을 비활성화하는 옵션은 개발을 촉진하기 위한 것이지만 프로덕션 클러스터는 커넥터를 사용해야 합니다. 자세한 내용은 the section called “용 커넥터 AWS 서비스” 단원을 참조하십시오.

ML Commons 설정에 대한 자세한 내용은 ML Commons cluster settings를 참조하세요.

OpenSearch Service 흐름 프레임워크 템플릿

Amazon OpenSearch Service 흐름 프레임워크 템플릿을 사용하면 공통 사용 사례에 대한 템플릿을 제공하여 복잡한 OpenSearch Service 설정 및 사전 처리 작업을 자동화할 수 있습니다. 예를 들어 흐름 프레임워크 템플릿을 사용하여 기계 학습 설정 작업을 자동화할 수 있습니다. Amazon OpenSearch Service 흐름 프레임워크 템플릿은 JSON 또는 YAML 문서의 설정 프로세스에 대한 간단한 설명을 제공합니다. 이러한 템플릿은 대화형 채팅 또는 쿼리 생성을 위한 자동화된 워크플로 구성, AI 커넥터, 도구, 에이전트 및 생성형 모델에 대한 백엔드 사용을 위해 OpenSearch Service를 준비하는 기타 구성 요소에 대해 설명합니다.

Amazon OpenSearch Service 흐름 프레임워크 템플릿은 특정 요구 사항에 맞게 사용자 지정할 수 있습니다. 사용자 지정 흐름 프레임워크 템플릿의 예제를 보려면 flow-framework를 참조하세요. OpenSearch Service 제공 템플릿은 workflow-templates를 참조하세요. 자세한 단계, API 참조 및 사용 가능한 모든 설정에 대한 참조를 포함한 포괄적인 설명서는 오픈 소스 OpenSearch 설명서의 automating configuration을 참조하세요.

> **Note**
>
> Flow-framework는 OpenSearch Service 2.17에 대한 백엔드 역할 필터링을 지원하지 않습니다.

OpenSearch Service에서 ML 커넥터 생성

Amazon OpenSearch Service 흐름 프레임워크 템플릿을 사용하면 ml-commons에서 제공되는 커넥터 생성 API를 활용하여 ML 커넥터를 구성하고 설치할 수 있습니다. ML 커넥터를 사용하여 OpenSearch Service를 다른 AWS 서비스 또는 타사 플랫폼에 연결할 수 있습니다. 이에 대한 자세한 내용은

Creating connectors for third-party ML platforms를 참조하세요. Amazon OpenSearch Service 흐름 프레임워크 API를 사용하면 OpenSearch Service 설정 및 사전 처리 작업을 자동화하고 ML 커넥터를 생성하는 데 사용할 수 있습니다.

OpenSearch Service에서 커넥터를 생성하려면 먼저 다음을 수행해야 합니다.

- Amazon SageMaker AI 도메인을 생성합니다.
- IAM 역할을 생성합니다.
- 역할 전달 권한을 구성합니다.
- OpenSearch 대시보드에서 flow-framework 및 ml-commons 역할을 매핑합니다.

AWS 서비스에 대한 ML 커넥터를 설정하는 방법에 대한 자세한 내용은 AWS 서비스에 대한 Amazon OpenSearch Service ML 커넥터를 참조하세요. 서드파티 플랫폼에서 OpenSearch Service ML 커넥터를 사용하는 방법에 대한 자세한 내용은 서드파티 플랫폼용 Amazon OpenSearch Service ML 커넥터를 참조하세요.

flow-framework 서비스를 통해 커넥터 생성

커넥터를 사용하여 flow-framework 템플릿을 생성하려면 OpenSearch Service 도메인 엔드포인트에 POST 요청을 전송해야 합니다. cURL, 샘플 Python 클라이언트, Postman 또는 다른 메서드를 사용하여 서명된 요청을 보낼 수 있습니다. POST 요청은 다음과 같은 형식을 사용합니다.

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

도메인이 가상 프라이빗 클라우드(Amazon VPC)에 상주하는 경우, 요청에서 AI 커넥터를 생성하려면 Amazon VPC에 연결되어야 합니다. Amazon VPC 액세스는 네트워크 구성에 따라 다르지만, 대개는 VPN 또는 회사 네트워크 연결이 필요합니다. OpenSearch Service 도메인에 도달할 수 있는지 알아보려면 웹 브라우저에서 `https://your-vpc-domain.region.es.amazonaws.com`으로 이동하여 기본 JSON 응답을 받을 수 있는지 확인합니다. (*## ### ####* 고유한 값으로 바꿉니다.

샘플 Python 클라이언트

Python 클라이언트는 HTTP 요청보다 자동화가 간단하고 재사용성이 뛰어납니다. Python 클라이언트로 AI 커넥터를 만들려면 다음 샘플 코드를 Python 파일에 저장하세요. 클라이언트에는 AWS SDK for Python(Boto3), Requests:HTTP for Humans 및 requests-aws4auth 1.2.3 패키지가 필요합니다.

```
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)
```

사전 정의된 워크플로 템플릿

Amazon OpenSearch Service는 몇 가지 공통 기계 학습(ML) 사용 사례에 대한 여러 워크플로 템플릿을 제공합니다. 템플릿을 사용하면 복잡한 설정이 간소화되고 시맨틱 또는 대화형 검색과 같은 사용 사례에 대한 많은 기본값이 제공됩니다. 워크플로 생성 API를 직접 호출할 때 워크플로 템플릿을 지정할 수 있습니다.

- OpenSearch Service 제공 워크플로 템플릿을 사용하려면 템플릿 사용 사례를 `use_case` 쿼리 파라미터로 지정합니다.
- 사용자 지정 워크플로 템플릿을 사용하려면 요청 본문에 전체 템플릿을 제공합니다. 사용자 지정 템플릿의 예제는 예제 JSON 템플릿 또는 예제 YAML 템플릿을 참조하세요.

템플릿 사용 사례

이 표에서는 사용 가능한 다양한 템플릿에 대한 개요, 템플릿에 대한 설명 및 필요한 파라미터를 제공합니다.

템플릿 사용 사례	설명	필요한 파라미터
`bedrock_titan_embedding_model_deploy`	Amazon Bedrock 임베딩 모델을 생성하고 배포합니다(기본적으로 `titan-embed-text-v1`).	`create_connector.credential.roleArn`
`bedrock_titan_embedding_model_deploy`	Amazon Bedrock 멀티모달 임베딩 모델을 생성하고 배포합니다(기본적으로 `titan-embed-text-v1`).	`create_connector.credential.roleArn`
`cohere_embedding_m`	Cohere 임베딩 모델(기본적으로 embed-english-v3.0)을 생성하고 배포합니다.	`create_connector.credential.roleArn` ,

템플릿 사용 사례	설명	필요한 파라미터
odel_deploy		create_connector.credential.secretArn
cohere_chat_model_deploy	Cohere 채팅 모델(기본적으로 Cohere Command)을 생성하고 배포합니다.	create_connector.credential.roleArn , create_connector.credential.secretArn
open_ai_embedding_model_deploy	OpenAI 임베딩 모델(기본적으로 text-embedding-ada-002)을 생성하고 배포합니다.	create_connector.credential.roleArn , create_connector.credential.secretArn
openai_chat_model_deploy	OpenAI 채팅 모델(기본적으로 gpt-3.5-turbo)을 생성하고 배포합니다.	create_connector.credential.roleArn , create_connector.credential.secretArn
semantic_search_with_cohere_embedding	시맨틱 검색을 구성하고 Cohere 임베딩 모델을 배포합니다. Cohere 모델의 API 키를 제공해야 합니다.	create_connector.credential.roleArn , create_connector.credential.secretArn
semantic_search_with_cohere_embedding_query_enricher	시맨틱 검색을 구성하고 Cohere 임베딩 모델을 배포합니다. 신경 쿼리의 기본 모델 ID를 설정하는 query_enricher 검색 프로세서를 추가합니다. Cohere 모델의 API 키를 제공해야 합니다.	create_connector.credential.roleArn , create_connector.credential.secretArn

템플릿 사용 사례	설명	필요한 파라미터
`multimodal_search_with_bedrock_titan`	Amazon Bedrock 멀티모달 모델을 배포하고 멀티모달 검색을 위한 text_image_embedding 프로세서 및 k-NN 인덱스를 사용하여 수집 파이프라인을 구성합니다. AWS 자격 증명을 제공해야 합니다.	`create_connector.credential.roleArn`

> **Note**
>
> 보안 암호 ARN이 필요한 모든 템플릿의 경우 기본값은 Secrets Manager에 키 이름이 "key"인 AWS 보안 암호를 저장하는 것입니다.

사전 훈련된 모델이 있는 기본 템플릿

Amazon OpenSearch Service는 오픈 소스 OpenSearch Service에서 사용할 수 없는 두 가지 추가 기본 워크플로 템플릿을 제공합니다.

템플릿 사용 사례	설명
`semantic_search_with_local_model`	시맨틱 검색을 구성하고 사전 훈련된 모델(`msmarco-distilbert-base-tas-b`)을 배포합니다. 신경 쿼리에 대한 기본 모델 ID를 설정하고 'my-nlp-index'라는 연결된 k-NN 인덱스를 생성하는 `neural_query_enricher` 검색 프로세서를 추가합니다.
`hybrid_search_with_local_model`	하이브리드 검색을 구성하고 사전 훈련된 모델(`msmarco-distilbert-base-tas-b`)을 배포합니다. 신경 쿼리에 대한 기본 모델 ID를 설정하고 'my-nlp-index'라는 연결된 k-NN 인덱스를 생성하는 `neural_query_enricher` 검색 프로세서를 추가합니다.

권한 구성

버전 2.13 이상에서 새 도메인을 생성하는 경우 권한이 이미 있습니다. 버전 2.11 이하의 기존 OpenSearch Service 도메인에서 흐름 프레임워크를 활성화한 다음, 버전 2.13 이상으로 업그레이드하는 경우 `flow_framework_manager` 역할을 정의해야 합니다. 관리자가 아닌 사용자는 이 역할에 매핑되어 세분화된 액세스 제어를 사용하는 도메인의 원 인덱스를 관리해야 합니다. 수동으로 `flow_framework_manager` 역할을 생성하려면 다음 단계를 수행합니다.

1. OpenSearch 대시보드에서 보안(Security)으로 이동하여 권한(Permissions)을 선택합니다.
2. 작업 그룹 생성(Create action group)을 선택하고 다음 그룹을 구성합니다.

그룹 이름	권한
`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. 역할(Roles)과 역할 생성(Create role)을 차례로 선택합니다.
4. 역할 이름을 flow_framework_manager로 지정합니다.
5. 클러스터 권한(Cluster permissions)에서 `flow_framework_full_access` 및 `flow_framework_read_access`를 선택합니다.
6. 인덱스(Index)에 *를 입력합니다.
7. 인덱스 권한(Index permissions)에서 `indices:admin/aliases/get`, `indices:admin/mappings/get`, `indices_monitor`를 선택합니다.
8. 생성(Create)을 선택합니다.
9. 역할을 생성한 후, 흐름 프레임워크 인덱스를 관리할 사용자 또는 백엔드 역할에 매핑합니다.

Amazon OpenSearch Service용 보안 분석

보안 분석은 조직의 인프라에 대한 가시성을 제공하고, 이상 활동을 모니터링하고, 잠재적 보안 위협을 실시간으로 탐지하고, 사전 구성된 대상에 경고를 트리거하는 OpenSearch 솔루션입니다. 보안 규칙을 지속적으로 평가하고 자동 생성된 보안 조사 결과를 검토하여 보안 이벤트 로그에서 악의적인 활동을 모니터링할 수 있습니다. 또한 보안 분석은 자동 경고를 생성하여 Slack 또는 이메일과 같은 지정된 알림 채널로 보낼 수 있습니다.

보안 분석 플러그인을 사용하면 일반적인 위협을 즉시 탐지하고 방화벽 로그, Windows 로그, 인증 감사 로그와 같은 기존 보안 이벤트 로그에서 중요한 보안 통찰력을 생성할 수 있습니다. 보안 알림을 사용하려면 도메인에서 OpenSearch 버전 2.5 이상을 실행해야 합니다.

> **Note**
>
> 이 설명서에서는 Amazon OpenSearch Service의 보안 분석에 대한 간략한 개요를 제공합니다. 여기에서는 주요 개념을 정의하고 권한을 구성하는 단계를 제공합니다. 설정 가이드, API 참조 및 사용 가능한 모든 설정에 대한 참조를 포함한 포괄적인 설명서는 OpenSearch 설명서의 Security Analytics를 참조하세요.

보안 분석 구성 요소 및 개념

다양한 도구와 기능이 보안 분석 운영의 토대를 제공합니다. 플러그인을 구성하는 주요 구성 요소에는 탐지기, 로그 유형, 규칙, 조사 결과 및 경고가 포함됩니다.

로그 유형

OpenSearch는 여러 유형의 로그를 지원하며 각 유형에 대한 기본 제공 매핑을 제공합니다. 탐지기를 생성할 때 로그 유형을 지정하고 시간 간격을 구성하면 보안 분석이 해당 간격으로 실행되는 관련 규칙 세트를 자동으로 활성화합니다.

탐지기

탐지기는 데이터 인덱스 전반의 로그 유형에 대한 다양한 사이버 보안 위협을 식별합니다. 시스템에서 발생하는 이벤트를 평가하는 사용자 지정 규칙과 사전 패키징된 Sigma 규칙을 모두 사용하도록 탐지

기를 구성합니다. 그런 다음 탐지기는 이러한 이벤트로부터 보안 결과를 생성합니다. 탐지기에 대한 자세한 내용은 OpenSearch 설명서의 탐지기 생성을 참조하세요.

규칙

위협 탐지 규칙은 탐지기가 보안 이벤트를 식별하기 위해 수집된 로그 데이터에 적용하는 조건을 정의합니다. 보안 분석은 요구 사항에 맞는 규칙 가져오기, 생성 및 사용자 지정을 지원하고, 로그에서 일반적인 위협을 탐지할 수 있도록 사전 패키징된 오픈 소스 Sigma 규칙도 제공합니다. 보안 분석은 MITRE ATT&CK 조직에서 유지 관리하는 적대적 전술 및 기법에 대해 계속 증가하는 지식 기반에 많은 규칙을 매핑합니다. OpenSearch Dashboards 또는 API를 모두 사용하여 규칙을 생성하고 사용할 수 있습니다. 규칙에 대한 자세한 내용은 OpenSearch 설명서의 규칙 작업을 참조하세요.

조사 결과

탐지기가 규칙을 로그 이벤트와 일치시키면 조사 결과가 생성됩니다. 각 조사 결과에는 선택 규칙, 로그 유형 및 규칙 심각도의 고유한 조합이 포함됩니다. 조사 결과가 반드시 시스템 내에 임박한 위협을 가리키는 것은 아니지만 항상 관심 있는 이벤트를 격리합니다. 조사 결과에 대한 자세한 내용은 OpenSearch 설명서의 조사 결과 작업을 참조하세요.

알림

탐지기를 생성할 때 알림을 트리거하는 하나 이상의 조건을 지정할 수 있습니다. 알림은 Slack 또는 이메일과 같은 선호 채널로 전송되는 알림입니다. 탐지기가 하나 이상의 규칙과 일치할 때 알림이 트리거되도록 설정하고 알림 메시지를 사용자 지정할 수 있습니다. 알림에 대한 자세한 내용은 OpenSearch 설명서의 알림 작업을 참조하세요.

보안 분석 살펴보기

OpenSearch 대시보드를 사용하여 보안 분석 플러그인을 시각화하고 이에 대한 인사이트를 얻을 수 있습니다. 개요 보기는 조사 결과 및 경고 수, 최근 조사 결과 및 경고, 빈번한 탐지 규칙, 탐지기 목록과 같은 정보를 제공합니다. 여러 시각화로 구성된 요약 보기를 볼 수 있습니다. 예를 들어, 다음 차트는 특정 기간 동안의 다양한 로그 유형에 대한 조사 결과 및 경고 추세를 보여줍니다.

페이지 하단에서 가장 최근의 조사 결과 및 경고를 검토할 수 있습니다.

또한 모든 활성 탐지기에서 가장 자주 트리거되는 규칙의 분포를 확인할 수 있습니다. 이를 통해 로그 유형별로 다양한 유형의 악성 활동을 탐지하고 조사할 수 있습니다.

마지막으로 구성된 감지기의 상태를 볼 수 있습니다. 이 패널에서 검출기 생성 워크플로로 이동할 수도 있습니다.

보안 분석 설정을 구성하려면 규칙 페이지에서 규칙을 생성하고 해당 규칙을 사용하여 탐지기 페이지에 탐지기를 작성합니다. 보안 분석 결과를 좀 더 집중적으로 보려면 조사 결과 및 경고 페이지를 사용할 수 있습니다.

권한 구성

기존 OpenSearch Service 도메인에서 보안 분석을 활성화한 경우 `security_analytics_manager` 역할이 도메인에 정의되어 있지 않을 수 있습니다. 관리자가 아닌 사용자는 이 역할에 매핑되어 세분화된 액세스 제어를 사용하는 도메인의 원 인덱스를 관리해야 합니다. 수동으로 `security_analytics_manager` 역할을 생성하려면 다음 단계를 수행합니다.

1. OpenSearch 대시보드에서 보안(Security)으로 이동하여 권한(Permissions)을 선택합니다.
2. 작업 그룹 생성(Create action group)을 선택하고 다음 그룹을 구성합니다.

그룹 이름	권한
`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`

그룹 이름	권한
	• 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. 역할(Roles)과 역할 생성(Create role)을 차례로 선택합니다.
4. 역할의 이름을 security_analytics_manager로 지정하세요.
5. 클러스터 권한(Cluster permissions)에서 `security_analytics_full_access` 및 `security_analytics_read_access`를 선택합니다.
6. 인덱스(Index)에 *를 입력합니다.
7. 인덱스 권한에서 `indices:admin/mapping/put`, `indices:admin/mappings/get`(을)를 선택합니다.
8. 생성(Create)을 선택합니다.
9. 역할을 생성한 후, 보안 분석 인덱스를 관리할 사용자 또는 백엔드 역할에 매핑합니다.

문제 해결

해당 인덱스 오류가 없습니다.

탐지기가 없는 상태에서 보안 분석 대시보드를 열면 오른쪽 하단에 `[index_not_found_exception] no such index [.opensearch-sap-detectors-config]`와 같은 알림이 표시될 수 있습니다. 이 알림은 무시해도 됩니다. 이 알림은 몇 초 내에 사라지고 탐지기를 만든 후에는 다시 표시되지 않습니다.

Amazon OpenSearch Service의 Observability

Amazon OpenSearch Service용 OpenSearch Dashboards의 기본 설치에는 Observability 플러그인이 포함되어 있습니다. 이 플러그인은 OpenSearch에 저장된 데이터를 탐색, 검색 및 쿼리하기 위해 파이프 처리 언어(PPL)를 사용하여 데이터 기반 이벤트를 시각화하는 데 사용할 수 있습니다. 이 기능을 사용하려면 OpenSearch 1.2 이상이 필요합니다.

Observability 플러그인은 공통 데이터 원본에서 지표, 로그 및 트레이스를 수집하고 모니터링할 수 있는 통합 환경을 제공합니다. 한 위치에서 데이터 수집 및 모니터링을 통해 전체 인프라의 전체 스택, 엔드-투-엔드 관찰이 가능합니다.

> **Note**
>
> 이 설명서에서는 OpenSearch Service의 관찰성에 대한 간략한 개요를 제공합니다. 권한을 포함한 관찰성 플러그인에 대한 포괄적인 설명서는 Observability를 참조하세요.

데이터 탐색 프로세스는 모두 다릅니다. 처음으로 데이터를 탐색하고 시각화를 생성하는 경우 다음과 같은 워크플로를 시도하는 것이 좋습니다.

이벤트 분석으로 데이터 탐색

우선, OpenSearch Service 도메인에서 항공 데이터를 수집하고 있으며 지난달 피츠버그 국제공항에 도착하는 항공편이 가장 많은 항공사를 찾고 싶다고 가정합니다. 다음 PPL 쿼리를 작성합니다.

```
source=opensearch_dashboards_sample_data_flights |
    stats count() by Dest, Carrier |
    where Dest = "Pittsburgh International Airport"
```

이 쿼리는 `opensearch_dashboards_sample_data_flights`라는 인덱스에서 데이터를 가져옵니다. 그런 다음 `stats` 명령을 사용하여 총항공편 수를 확보하고 목적지 공항 및 항공사에 따라 그룹화합니다. 마지막으로, `where` 절을 사용하여 피츠버그 국제 공항에 도착하는 항공편으로 결과를 필터링합니다.

지난달에 대해 표시되는 데이터는 다음과 같습니다.

쿼리 편집기의 PPL 버튼을 선택하여 각 PPL 명령에 대한 사용 정보 및 예제를 가져옵니다.

비행 지연에 대한 정보를 쿼리하는 좀 더 복잡한 예를 살펴보겠습니다.

```
source=opensearch_dashboards_sample_data_flights |
    where FlightDelayMin > 0 |
    stats sum(FlightDelayMin) as minimum_delay, count() as total_delayed by Carrier,
 Dest |
    eval avg_delay=minimum_delay / total_delayed |
    sort - avg_delay
```

쿼리의 각 명령은 최종 출력에 영향을 줍니다.

- `source=opensearch_dashboards_sample_data_flights` - 이전 예제와 동일한 인덱스에서 데이터를 가져옵니다.
- `where FlightDelayMin > 0` - 지연된 항공편으로 데이터를 필터링합니다.
- `stats sum(FlightDelayMin) as minimum_delay, count() as total_delayed by Carrier` - 각 항공사의 총 최소 지연 시간 및 지연된 총항공편 수를 가져옵니다.
- `eval avg_delay=minimum_delay / total_delayed` - 최소 지연 시간을 지연된 총항공편 수로 나누어 각 항공사의 평균 지연 시간을 계산합니다.
- `sort - avg_delay` - 평균 지연을 기준으로 결과를 내림차순으로 정렬합니다.

이 쿼리를 사용하면 OpenSearch Dashboards 항공사의 지연이 평균적으로 더 적은 것을 확인할 수 있습니다.

자세한 샘플 PPL 쿼리 샘플는 이벤트 분석 페이지의 쿼리 및 시각화에서 확인할 수 있습니다.

시각화 생성

관심 있는 데이터를 올바르게 쿼리하면 이러한 쿼리를 시각화로 저장할 수 있습니다.

그런 다음 해당 시각화를 작업 패널에 추가하여 서로 다른 데이터 조각을 비교합니다. 노트북을 활용하여 팀원들과 공유할 수 있는 다양한 시각화 및 코드 블록을 결합합니다.

Trace Analytics 자세히 살펴보기

Trace Analytics에서는 OpenSearch 데이터의 이벤트 흐름을 시각화하여 분산 애플리케이션의 성능 문제를 식별하고 해결할 방법을 제공합니다.

Amazon OpenSearch Service용 Trace Analytics

OpenSearch Observability 플러그인의 일부인 Trace Analytics를 사용하여 분산 애플리케이션의 추적 데이터를 분석할 수 있습니다. Trace Analytics에는 OpenSearch 또는 Elasticsearch 7.9 이상이 필요합니다.

분산 애플리케이션에서 사용자가 버튼을 클릭하는 것과 같은 단일 작업은 일련의 확장된 이벤트를 트리거할 수 있습니다. 예를 들어 애플리케이션 프런트 엔드는 다른 서비스를 호출하고 데이터베이스를 쿼리하며 쿼리를 처리하고 결과를 반환하는 백엔드 서비스를 호출할 수 있습니다. 그런 다음 첫 번째 백엔드 서비스가 UI를 업데이트하는 프런트 엔드에 확인을 보냅니다.

Trace Analytics를 사용하여 이러한 이벤트 흐름을 시각화하고 성능 문제를 식별할 수 있습니다.

> **Note**
>
> 이 설명서에서는 Trace Analytics에 대한 간략한 개요를 제공합니다. 포괄적인 설명서는 오픈 소스 OpenSearch 설명서의 Trace Analytics를 참조하세요.

사전 조건

Trace Analytics를 사용하려면 애플리케이션에 계측을 추가하고 Jaeger 또는 Zipkin과 같은 OpenTelemetry 지원 라이브러리를 사용하여 추적 데이터를 생성해야 합니다. 이 단계는 전적으로 OpenSearch Service 외부에서 발생합니다. AWS Distro for OpenTelemetry 설명서에는 Java, Python, Go 및 JavaScript를 비롯하여 시작하는 데 도움이 되는 다양한 프로그래밍 언어에 대한 예제 애플리케이션이 포함되어 있습니다.

애플리케이션에 계측을 추가한 후 OpenTelemetry Collector는 애플리케이션에서 데이터를 수신하고 OpenTelemetry 데이터로 포맷합니다. GitHub의 수신기 목록을 참조하세요. OpenTelemetry용 AWS 배포판에는 용 수신기가 포함되어 있습니다 AWS X-Ray.

마지막으로 Amazon OpenSearch Ingestion을 사용하여 OpenSearch 와 함께 사용할 OpenTelemetry 데이터를 포맷할 수 있습니다.

OpenTelemetry Collector 샘플 구성

Amazon OpenSearch Ingestion과 함께 OpenTelemetry Collector를 사용하려면 다음 샘플 구성을 시도합니다.

```
extensions:
  sigv4auth:
    region: "us-east-1"
    service: "osis"

receivers:
  jaeger:
    protocols:
      grpc:

exporters:
  otlphttp:
    traces_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/
opentelemetry.proto.collector.trace.v1.TraceService/Export"
    auth:
      authenticator: sigv4auth
    compression: none

service:
  extensions: [sigv4auth]
  pipelines:
    traces:
      receivers: [jaeger]
      exporters: [otlphttp]
```

OpenSearch Ingestion 샘플 구성

추적 데이터를 OpenSearch Service 도메인에 보내려면 다음 샘플 OpenSearch Ingestion 파이프라인 구성을 시도합니다. 파이프라인을 생성하는 방법에 대한 지침은 the section called “파이프라인 생성” 섹션을 참조하세요.

```
version: "2"
otel-trace-pipeline:
```

```
  source:
    otel_trace_source:
      "/${pipelineName}/ingest"
  processor:
    - trace_peer_forwarder:
  sink:
    - pipeline:
        name: "trace_pipeline"
    - pipeline:
        name: "service_map_pipeline"
trace-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - otel_traces:
  sink:
    - opensearch:
        hosts: ["https://domain-endpoint"]
        index_type: trace-analytics-raw
        aws:
          # IAM role that OpenSearch Ingestion assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"

service-map-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - service_map:
  sink:
    - opensearch:
        hosts: ["https://domain-endpoint"]
        index_type: trace-analytics-service-map
        aws:
          # IAM role that the pipeline assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
```

`sts_role_arn` 옵션에서 지정한 파이프라인 역할에는 싱크에 대해 쓰기 권한이 있어야 합니다. 파이프라인 역할에 대한 권한을 구성하는 지침은 the section called “역할 및 사용자 설정” 섹션을 참조하세요.

데이터 추적 탐색

대시보드 보기는 특정 작업과 관련된 평균 대기 시간, 오류율 및 추세를 볼 수 있도록 HTTP 메서드 및 경로별로 추적을 함께 그룹화합니다. 더욱 집중된 보기를 위해 추적 그룹 이름을 기준으로 필터링합니다.

추적 그룹을 구성하는 추적을 드릴다운하려면 오른쪽 열에서 추적 수를 선택합니다. 그런 다음 자세한 요약을 위해 개별 추적을 선택합니다.

서비스 보기는 애플리케이션의 모든 서비스와 다양한 서비스가 서로 연결되는 방법을 보여주는 대화형 맵을 나열합니다. 작업별로 문제를 식별하는 데 도움이 되는 대시보드와는 달리 서비스 맵은 서비스별로 문제를 식별하는 데 도움이 됩니다. 오류율 또는 대기 시간을 기준으로 정렬하여 애플리케이션의 잠재적 문제 영역을 파악합니다.

파이프 처리 언어를 사용하여 Amazon OpenSearch Service 데이터 쿼리

파이프 처리 언어(PPL) 는 파이프(|) 구문을 사용하여 Amazon OpenSearch Service에 저장된 데이터를 쿼리하도록 하는 쿼리 언어입니다. PPL에는 OpenSearch 또는 Elasticsearch 7.9 이상이 필요합니다.

> **Note**
>
> 이 설명서에서는 Amazon OpenSearch Service용 PPL에 대한 간략한 개요를 제공합니다. 자세한 단계와 전체 명령 참조는 오픈 소스 OpenSearch 설명서의 PPL을 참조하세요.

PPL 구문은 파이프 문자(|)로 구분된 명령으로 구성되며, 여기서 데이터가 각 파이프라인을 통해 왼쪽에서 오른쪽으로 흐릅니다. 예를 들어, HTTP 403 또는 503 오류가 있는 호스트의 수를 찾고 호스트별로 집계하고 영향 순서대로 정렬하는 PPL 구문은 다음과 같습니다.

```
source = dashboards_sample_data_logs | where response='403' or response='503' | stats
 count(request) as request_count by host, response | sort -request_count
```

시작하려면 OpenSearch Dashboards에서 쿼리 워크벤치(Query Workbench)를 열고 PPL을 선택합니다. `bulk` 작업을 사용하여 일부 샘플 데이터를 인덱싱합니다.

```
PUT accounts/_bulk?refresh
{"index":{"_id":"1"}}
{"account_number":1,"balance":39225,"firstname":"Amber","lastname":"Duke","age":32,"gender":"M"
 Holmes
 Lane","employer":"Pyrami","email":"amberduke@pyrami.com","city":"Brogan","state":"IL"}
{"index":{"_id":"6"}}
{"account_number":6,"balance":5686,"firstname":"Hattie","lastname":"Bond","age":36,"gender":"M"
 Bristol
 Street","employer":"Netagy","email":"hattiebond@netagy.com","city":"Dante","state":"TN"}
{"index":{"_id":"13"}}
{"account_number":13,"balance":32838,"firstname":"Nanette","lastname":"Bates","age":28,"gender"
 Mady Street","employer":"Quility","city":"Nogal","state":"VA"}
{"index":{"_id":"18"}}
{"account_number":18,"balance":4180,"firstname":"Dale","lastname":"Adams","age":33,"gender":"M"
 Hutchinson Court","email":"daleadams@boink.com","city":"Orick","state":"MD"}
```

다음 예제는 age가 18보다 큰 계정 인덱스에 있는 문서에 대해 firstname과 lastname 필드를 반환합니다.

```
search source=accounts | where age > 18 | fields firstname, lastname
```

샘플 응답

id	firstname	lastname
0	Amber	Duke
1	Hattie	Bond
2	Nanette	Bates
3	Dale	Adams

search, where, fields, rename, dedup, stats, sort, eval, head, top 및 rare과 같은 읽기 전용 명령의 전체 집합을 사용할 수 있습니다. PPL 플러그인은 수학, 삼각법, 날짜-시간, 문자열, 집계 및 고급 연산자와 표현식을 포함한 모든 SQL 함수를 지원합니다. 자세한 내용은 OpenSearch PPL 참조 설명서를 참조하세요.

Amazon OpenSearch Service의 운영 모범 사례

이 장에서는 Amazon OpenSearch Service 도메인 운영에 대한 몇 가지 모범 사례를 제공하며, 많은 사용 사례에 적용되는 일반 지침을 포함하고 있습니다. 각 워크로드는 고유한 특성을 가지고 있으므로 모든 사용 사례에 적합한 일반적인 권장 사항은 없습니다. 가장 중요한 모범 사례는 지속적인 주기로 도메인을 배포, 테스트 및 조정하여 워크로드에 대한 최적의 구성, 안정성 및 비용을 찾는 것입니다.

주제

- 모니터링 및 알림
- 샤드 전략
- 안정성
- 성능
- 보안
- 비용 최적화
- Amazon OpenSearch Service 도메인 크기 조정
- Amazon OpenSearch Service의 페타바이트 규모
- Amazon OpenSearch Service의 전용 조정자 노드
- Amazon OpenSearch Service의 전용 프라이머리 노드

모니터링 및 알림

다음 모범 사례는 OpenSearch Service 도메인을 모니터링하는 데 적용됩니다.

CloudWatch 경보 구성

OpenSearch Service는 Amazon CloudWatch에 성능 지표를 전송합니다. 클러스터 및 인스턴스 지표를 정기적으로 검토하고 워크로드 성능에 따라 권장되는 CloudWatch 경보를 구성합니다.

로그 게시 사용 설정

OpenSearch Service는 Amazon CloudWatch Logs에서 OpenSearch 오류 로그, 검색 느린 로그, 인덱싱 느린 로그 및 감사 로그를 노출합니다. 검색 느린 로그, 인덱싱 느린 로그 및 오류 로그는 성능 및 안정성 문제 해결에 유용합니다. 감사 로그는 세분화된 액세스 제어를 사용 설정한 경우에만 사용할 수 있으며, 사용자 활동을 추적합니다. 자세한 내용은 OpenSearch 설명서의 로그를 참조하세요.

검색 느린 로그 및 인덱싱 느린 로그는 검색 및 인덱싱 작업의 성능을 이해하고 문제를 해결하는 데 중요한 도구입니다. 모든 프로덕션 도메인에 대해 검색 및 인덱스 느린 로그 전달을 사용 설정합니다. 또한 로깅 임계값을 구성해야 하며, 그러지 않으면 CloudWatch가 로그를 캡처하지 않습니다.

샤드 전략

샤드는 OpenSearch Service 도메인의 데이터 노드 전체에 워크로드를 분산합니다. 인덱스를 올바르게 구성하면 전반적인 도메인 성능을 향상시킬 수 있습니다.

OpenSearch Service로 데이터를 보내면 해당 데이터를 인덱스로 보냅니다. 인덱스는 문서가 행으로, 필드가 열로 되어 있는 데이터베이스 테이블과 유사합니다. 인덱스를 만들 때 생성하려는 기본 샤드의 수를 OpenSearch에 지정합니다. 기본 샤드는 전체 데이터 세트의 독립 파티션입니다. OpenSearch Service는 인덱스의 기본 샤드에 데이터를 자동으로 배포합니다. 또한, 인덱스의 복제본을 구성할 수도 있습니다. 각 복제본 샤드 구성은 해당 인덱스에 대한 기본 샤드의 전체 복사본 집합으로 구성됩니다.

OpenSearch Service는 클러스터의 데이터 노드 전체에서 각 인덱스에 대한 샤드를 매핑합니다. 인덱스의 기본 및 복제본 샤드가 서로 다른 데이터 노드에 상주하도록 보장합니다. 첫 번째 복제본은 인덱스에 두 개의 데이터 복사본이 있는지 확인합니다. 항상 하나 이상의 복제본을 사용해야 합니다. 추가 복제본은 추가 중복성과 읽기 용량을 제공합니다.

OpenSearch는 인덱스에 속한 샤드를 포함하는 모든 데이터 노드에 인덱싱 요청을 보냅니다. 먼저 기본 샤드를 포함하는 데이터 노드로 인덱싱 요청을 보낸 다음 복제본 샤드를 포함하는 데이터 노드로 인덱싱 요청을 보냅니다. 코디네이터 노드는 검색 요청을 인덱스에 속한 모든 샤드의 기본 샤드 또는 복제본 샤드로 라우팅합니다.

예를 들어 5개의 기본 샤드와 1개의 복제본이 있는 인덱스의 경우 각 인덱싱 요청은 10개의 샤드를 접합니다. 반면에 검색 요청은 n개의 샤드로 전송됩니다. 여기서 n은 기본 샤드의 수입니다. 5개의 기본 샤드와 1개의 복제본이 있는 인덱스의 경우, 각 검색 쿼리는 해당 인덱스의 샤드 5개(기본 또는 복제본)를 접합니다.

샤드 및 데이터 노드 수 결정

다음 모범 사례를 사용하여 도메인의 샤드 및 데이터 노드 수를 결정합니다.

샤드 크기 - 디스크의 데이터 크기는 소스 데이터 크기의 직접적인 결과이며 더 많은 데이터를 인덱싱할 때 변경됩니다. 소스 대 인덱스 비율은 1:10에서 10:1 또는 그 이상까지 크게 다를 수 있지만, 일반적으로 약 1:1.10입니다. 이 비율을 사용하여 디스크의 인덱스 크기를 예측할 수 있습니다. 또한 일부 데이터를 인덱싱하고 실제 인덱스 크기를 검색하여 워크로드에 대한 비율을 결정할 수 있습니다. 인덱

스 크기를 예측했으면 각 샤드가 10~30GiB(검색 워크로드의 경우) 또는 30~50GiB(로그 워크로드의 경우)가 되도록 샤드 수를 설정합니다. 50GiB가 최대값이어야 하며, 성장에 대비한 계획을 세워야 합니다.

샤드 수 - 데이터 노드에 샤드를 배포하면 도메인 성능에 큰 영향을 미칩니다. 샤드가 여러 개인 인덱스가 있는 경우 샤드 수를 데이터 노드 수의 배수로 만듭니다. 이렇게 하면 샤드가 데이터 노드 간에 고르게 분산되며, 핫 노드를 방지할 수 있습니다. 예를 들어, 12개의 기본 샤드가 있는 경우 데이터 노드 수는 2, 3, 4, 6 또는 12여야 합니다. 단, 샤드 수는 샤드 크기에 부차적입니다. 5GiB의 데이터가 있는 경우에도 단일 샤드를 사용해야 합니다.

데이터 노드당 샤드 - 노드가 보유할 수 있는 총 샤드 수는 노드의 Java 가상 머신(JVM) 힙 메모리에 비례합니다. 힙 메모리 GiB당 25개 이하의 샤드를 목표로 합니다. 예를 들어, 32GiB의 힙 메모리가 있는 노드는 800개 이하의 샤드를 보유해야 합니다. 샤드 배포는 워크로드 패턴에 따라 다를 수 있지만 Elasticsearch 및 OpenSearch 1.1~2.15의 경우 노드당 1,000개의 샤드, OpenSearch 2.17 이상의 경우 4,000개의 샤드 제한이 있습니다. cat/allocation API는 데이터 노드의 샤드 수와 전체 샤드 스토리지에 대한 빠른 보기를 제공합니다.

샤드 대 CPU 비율 - 샤드가 인덱싱 또는 검색 요청에 관련된 경우 vCPU 사용하여 요청을 처리합니다. 샤드당 1.5 vCPU의 초기 확장 지점을 사용하는 것이 가장 좋습니다. 인스턴스 유형에 vCPU가 8개인 경우, 각 노드에 샤드가 6개 이하가 되도록 데이터 노드 수를 설정합니다. 이 값은 근사치입니다. 워크로드를 테스트하고 그에 따라 클러스터를 확장해야 합니다.

스토리지 볼륨, 샤드 크기, 인스턴스 유형 권장 사항은 다음 리소스를 참조하세요.

- the section called “도메인 크기 조정”
- the section called “페타바이트 규모”

스토리지 스큐 방지

스토리지 스큐는 클러스터 내의 하나 이상의 노드가 다른 노드보다 하나 이상의 인덱스에 대해 더 높은 비율의 스토리지를 보유할 때 발생합니다. 스토리지 스큐의 표시에는 불균등한 CPU 사용률, 간헐적이고 불균일한 대기 시간, 데이터 노드 전반의 불균등한 대기열이 포함됩니다. 스큐 문제가 있는지 확인하려면 다음 해결 방법 섹션을 참조하세요.

- the section called “노드 샤드 및 스토리지 스큐”
- the section called “인덱스 샤드 및 스토리지 스큐”

안정성

다음 모범 사례는 안정적이고 건강한 OpenSearch Service 도메인을 유지 관리하는 데 적용됩니다.

OpenSearch로 최신 정보 유지

서비스 소프트웨어 업데이트

OpenSearch Service는 기능을 추가하거나 도메인을 개선하는 소프트웨어 업데이트를 정기적으로 릴리스합니다. 업데이트는 OpenSearch 또는 Elasticsearch 엔진 버전을 변경하지 않습니다. 설명 도메인 API 작업을 실행하도록 반복 시간을 예약하고 `UpdateStatus`이(가) `ELIGIBLE`인 경우 서비스 소프트웨어 업데이트를 시작하는 것이 좋습니다. 특정 기간(일반적으로 2주) 내에 도메인을 업데이트하지 않으면 OpenSearch Service가 업데이트를 자동으로 수행합니다.

OpenSearch 버전 업그레이드

OpenSearch Service는 커뮤니티에서 관리하는 OpenSearch 버전에 대한 지원을 정기적으로 추가합니다. 최신 OpenSearch 버전이 출시되면 항상 최신 버전으로 업그레이드합니다.

OpenSearch Service는 OpenSearch와 OpenSearch Dashboards(또는 도메인이 레거시 엔진을 실행하는 경우 Elasticsearch와 Kibana)를 동시에 업그레이드합니다. 클러스터에 전용 프라이머리 노드가 있는 경우 가동 중지 없이 업그레이드가 완료됩니다. 그렇지 않으면 클러스터가 업그레이드 후 프라이머리 노드를 선택하는 몇 초 동안 응답하지 않을 수도 있습니다. 일부 또는 모든 업그레이드 도중 OpenSearch Dashboards를 사용하지 못할 수 있습니다.

도메인을 업그레이드하는 방법은 두 가지입니다.

- 인 플레이스(In-place) 업그레이드 - 동일한 클러스터를 유지하므로 이 옵션이 더 쉽습니다.
- 스냅샷/복원 업그레이드 - 이 옵션은 새 클러스터에서 새 버전을 테스트하거나 클러스터 간 마이그레이션에 적합합니다.

어떤 업그레이드 프로세스를 사용하든 개발 및 테스트 전용 도메인을 유지 관리하고 프로덕션 도메인을 업그레이드하기 전에 새 버전으로 업그레이드하는 것이 좋습니다. 테스트 도메인을 생성할 때, 배포 유형은 Development and testing(개발 및 테스트)를 선택합니다. 도메인 업그레이드 직후 모든 클라이언트를 호환되는 버전으로 업그레이드해야 합니다.

스냅샷 성능 개선

스냅샷이 처리 중에 중단되는 것을 방지하려면 전용 프라이머리 노드의 인스턴스 유형이 샤드 수와 일치해야 합니다. 자세한 내용은 the section called “전용 프라이머리 노드에 대한 인스턴스 유형 선택” 단

원을 참조하십시오. 또한 각 노드에는 Java 힙 메모리의 GiB당 25개 이상의 권장 샤드가 있어서는 안 됩니다. 자세한 내용은 the section called “샤드 수 선택” 단원을 참조하십시오.

전용 프라이머리 노드 사용 설정

전용 프라이머리 노드는 클러스터 안정성을 향상시킵니다. 전용 프라이머리 노드는 클러스터 관리 작업을 수행하지만 인덱스 데이터를 보유하거나 클라이언트 요청에 응답하지 않습니다. 클러스터 관리 작업을 오프로드하면 도메인의 안정성이 향상되고 일부 구성 변경이 다운타임 없이 일어날 수 있습니다.

3개의 전용 프라이머리 노드를 사용 설정하고 사용하여 3개의 가용 영역에서 도메인 안정성을 최적화합니다. Multi-AZ with Standby로 배포하면 전용 프라이머리 노드 3개가 구성됩니다. 인스턴스 유형 권장 사항은 the section called “전용 프라이머리 노드에 대한 인스턴스 유형 선택” 섹션을 참조하세요.

여러 가용 영역에 걸쳐 배포

서비스 중단 발생 시 데이터 손실을 방지하고 클러스터 가동 중지 시간을 최소화하기 위해 동일한 AWS 리전에 있는 두 개 또는 세 개의 가용 영역에 노드를 분산할 수 있습니다. 모범 사례는 Multi-AZ with Standby를 사용하여 배포하는 것으로, 이 배포는 3개의 가용 영역(활성 영역 2개와 대기 영역 1개, 인덱스당 복제 샤드 2개 포함)을 구성합니다. 이 구성을 통해 OpenSearch Service는 복제본 샤드를 해당 기본 샤드와 다른 AZ에 배포할 수 있습니다. 가용 영역 간 클러스터 통신에 대해서는 교차 AZ 데이터 전송 요금이 부과되지 않습니다.

가용 영역은 각 리전 내에 있는 격리된 위치입니다. 2개의 AZ 구성에서 하나의 가용 영역이 손실되면 전체 도메인 용량의 절반이 손실됩니다. 세 개의 가용 영역으로 이동하면 단일 가용 영역이 손실될 경우의 영향이 더욱 줄어듭니다.

수집 흐름 및 버퍼링 제어

_bulk API 작업을 사용하여 전체 요청 수를 제한하는 것이 좋습니다. 단일 문서가 포함된 5,000개의 요청을 보내는 것보다 5,000개의 문서가 포함된 하나의 `_bulk` 요청을 보내는 것이 더 효율적입니다.

최적의 운영 안정성을 위해 인덱싱 요청의 업스트림 흐름을 제한하거나 일시 중지해야 하는 경우가 있습니다. 인덱스 요청 비율을 제한하는 것은 클러스터를 압도할 수 있는 예기치 않은 또는 간헐적인 요청 급증을 처리하기 위한 중요한 메커니즘입니다. 업스트림 아키텍처에 흐름 제어 메커니즘을 구축하는 것이 좋습니다.

다음 다이어그램은 로그 수집 아키텍처의 여러 구성 요소 옵션을 보여줍니다. 갑작스러운 트래픽 급증 및 간단한 도메인 유지 관리를 위해 들어오는 데이터를 버퍼링할 수 있는 충분한 공간을 확보하도록 집계 계층을 구성합니다.

검색 워크로드에 대한 매핑 생성

검색 워크로드의 경우 OpenSearch가 문서와 해당 필드를 저장하고 인덱싱하는 방법을 정의하는 매핑을 만듭니다. 실수로 새 필드를 추가하지 않도록 `dynamic`을(를) `strict`(으)로 설정합니다.

```
PUT my-index
{
  "mappings": {
    "dynamic": "strict",
    "properties": {
      "title": { "type" : "text" },
      "author": { "type" : "integer" },
      "year": { "type" : "text" }
    }
  }
}
```

인덱스 템플릿 사용

인덱스가 생성될 때 인덱스를 구성하는 방법을 OpenSearch에 알려주는 방법으로 인덱스 템플릿을 사용할 수 있습니다. 인덱스를 만들기 전에 인덱스 템플릿을 구성합니다. 그런 다음 인덱스를 만들면 템플릿에서 설정 및 매핑을 상속합니다. 단일 인덱스에 둘 이상의 템플릿을 적용할 수 있으므로 한 템플릿에서 설정을 지정하고 다른 템플릿에서 매핑을 지정할 수 있습니다. 이 전략을 사용하면 여러 인덱스의 공통 설정을 위한 하나의 템플릿과 보다 구체적인 설정 및 매핑을 위한 별도의 템플릿을 사용할 수 있습니다.

다음 설정은 템플릿에서 구성할 때 유용합니다.

- 기본 및 복제본 샤드 수
- 새로 고침 간격(검색할 수 있도록 인덱스를 새로 고치고 최근 변경 사항을 적용하는 빈도)
- 동적 매핑 제어
- 명시적 필드 매핑

다음 예제 템플릿에는 이러한 각 설정이 포함되어 있습니다.

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

거의 변경되지 않더라도 OpenSearch에서 설정 및 매핑을 중앙에서 정의하면 여러 업스트림 클라이언트를 업데이트하는 것보다 관리가 간단합니다.

인덱스 상태 관리를 사용한 인덱스 관리

로그 또는 시계열 데이터를 관리하는 경우 인덱스 상태 관리(ISM)를 사용하는 것이 좋습니다. ISM을 사용하면 일반 인덱스 수명 주기 관리 작업을 자동화할 수 있습니다 ISM을 사용하면 인덱스 별칭 롤오버를 호출하고, 인덱스 스냅샷을 생성하며, 스토리지 계층 간에 인덱스를 이동하고, 이전 인덱스를 삭제하는 정책을 생성할 수 있습니다. 샤드 스큐를 방지하기 위한 대체 데이터 수명 주기 관리 전략으로 ISM 롤오버 작업을 사용할 수도 있습니다.

먼저 ISM 정책을 설정합니다. 예제는 the section called “샘플 정책”을 참조하세요. 그런 다음 정책을 하나 이상의 인덱스에 연결합니다. 정책에 ISM 템플릿 필드를 포함하면 OpenSearch Service는 지정된 패턴과 일치하는 모든 인덱스에 정책을 자동으로 적용합니다.

사용되지 않는 인덱스 삭제

클러스터의 인덱스를 정기적으로 검토하고 사용하지 않는 인덱스를 식별합니다. 이러한 인덱스의 스냅샷을 만들어 S3에 저장한 다음 삭제합니다. 사용되지 않는 인덱스를 제거하면 샤드 수가 줄어들고 노드 간에 보다 균형 잡힌 스토리지 배포 및 리소스 활용이 가능합니다. 유휴 상태에서도 인덱스는 내부 인덱스 유지 관리 작업 중에 일부 리소스를 소비합니다.

사용하지 않는 인덱스를 수동으로 삭제하는 대신 ISM을 사용하여 자동으로 스냅샷을 만들고 일정 시간 후 인덱스를 삭제할 수 있습니다.

고가용성을 위한 여러 도메인을 사용

여러 리전에서 99.9% 가동 시간 이상의 고가용성을 달성하려면 두 개의 도메인을 사용하는 것이 좋습니다. 작거나 느리게 변화하는 데이터 세트의 경우 클러스터 간 복제를 설정하여 액티브-패시브 모델을 유지할 수 있습니다. 이 모델에서는 리더 도메인만 기록되지만, 어느 도메인에서든 읽을 수 있습니다. 더 큰 데이터 세트와 빠르게 변경되는 데이터의 경우, 모든 데이터가 액티브-액티브 모델의 두 도메인에 독립적으로 기록되도록 수집 파이프라인에서 이중 전달을 구성합니다.

장애 조치를 염두에 두고 업스트림 및 다운스트림 애플리케이션을 설계합니다. 장애 조치 프로세스를 다른 재해 복구 프로세스와 함께 테스트해야 합니다.

성능

최적의 성능을 위해 도메인을 조정하는 데 다음 모범 사례가 적용됩니다.

대량 요청 크기 및 압축 최적화

대량 크기 조정은 데이터, 분석 및 클러스터 구성에 따라 다르지만, 좋은 시작점은 대량 요청당 3~5MiB입니다.

요청 및 응답 페이로드 크기를 줄이기 위해 gzip 압축을 사용하여 OpenSearch 도메인에서 요청을 보내고 응답을 받습니다. OpenSearch Python 클라이언트와 함께 또는 클라이언트 측에서 다음 헤더를 포함하여 gzip 압축을 사용할 수 있습니다.

- `'Accept-Encoding': 'gzip'`
- `'Content-Encoding': 'gzip'`

대량 요청 크기를 최적화하려면 3MiB의 대량 요청 크기로 시작합니다. 그런 다음 인덱싱 성능이 개선되지 않을 때까지 요청 크기를 서서히 늘립니다.

> **Note**
>
> Elasticsearch 6.x 버전을 실행하는 도메인에서 gzip 압축을 사용 설정하려면 클러스터 수준에서 `http_compression.enabled`를 설정해야 합니다. 이 설정은 Elasticsearch 7.x 버전 및 모든 버전의 OpenSearch에서 기본적으로 true입니다.

대량 요청 응답의 크기를 줄입니다.

OpenSearch 응답의 크기를 줄이려면 `filter_path` 파라미터를 사용하여 불필요한 필드를 제외합니다. 실패한 요청을 식별하거나 재시도하는 데 필요한 필드를 필터링하지 않도록 합니다. 자세한 정보와 지침은 the section called “응답 크기 감소” 섹션을 참조하세요.

새로 고침 주기 조정

OpenSearch 인덱스에는 최종 읽기 일관성이 있습니다. 새로 고침 작업을 수행하면 인덱스에 대해 수행된 모든 업데이트를 검색할 수 있습니다. 기본 새로 고침 간격은 1초입니다. 즉, OpenSearch는 인덱스가 기록되는 동안 1초마다 새로 고침을 수행합니다.

인덱스를 새로 고치는 빈도가 낮을수록(새로 고침 간격이 길수록) 전반적인 인덱싱 성능이 향상됩니다. 새로 고침 간격을 늘리면 인덱스 업데이트와 새 데이터를 검색할 수 있는 시간 사이의 지연 시간이 길어진다는 단점이 있습니다. 전체 성능을 향상시키려면 새로 고침 간격을 허용할 수 있는 한 높게 설정합니다.

모든 인덱스에 대한 `refresh_interval` 파라미터를 30초 이상으로 설정하는 것이 좋습니다.

자동 조정 사용 설정

자동 조정은 OpenSearch 클러스터의 성능 및 사용량 지표를 사용하여 노드의 대기열 및 캐시 크기 및 Java 가상 머신(JVM) 설정에 대한 변경을 제안합니다. 이러한 선택적 변경 사항은 클러스터 속도와 안정성을 향상시킵니다. 언제든지 기본 OpenSearch Service 설정으로 되돌릴 수 있습니다. 자동 조정은 명시적으로 사용 중지하지 않는 한 새 도메인에서 기본적으로 사용 설정됩니다.

모든 도메인에서 자동 조정을 사용하도록 설정하고 반복 유지 관리 기간을 설정하거나 권장 사항을 정기적으로 검토하는 것이 좋습니다.

보안

다음 모범 사례가 도메인 보안에 적용됩니다.

세분화된 액세스 제어 사용 설정

세분화된 액세스 제어를 사용하면 OpenSearch Service 도메인 내의 특정 데이터에 액세스할 수 있는 사용자를 제어할 수 있습니다. 일반화된 액세스 제어와 비교하여 세분화된 액세스 제어는 각 클러스터, 인덱스, 문서 및 필드에 액세스에 지정된 고유한 정책을 제공합니다. 액세스 기준은 액세스를 요청하는

사람의 역할 및 데이터에 대해 수행하려는 작업을 비롯한 여러 요소를 기반으로 할 수 있습니다. 예를 들어 한 사용자에게는 인덱스에 쓸 수 있는 액세스 권한을 부여하고 다른 사용자에게는 변경 없이 인덱스의 데이터를 읽을 수 있는 액세스 권한만 부여할 수 있습니다.

세분화된 액세스 제어를 통해 보안 또는 규정 준수 문제를 일으키지 않고 액세스 요구 사항이 서로 다른 데이터가 동일한 스토리지 공간에 존재할 수 있습니다.

도메인에서 세분화된 액세스 제어를 사용 설정하는 것을 권장합니다.

VPC 내에 도메인 배포

OpenSearch Service 도메인을 Virtual Private Cloud(VPC) 안에 배치하면 인터넷 게이트웨이, NAT 디바이스 또는 VPN 연결 없이 VPC 내부에서 OpenSearch Service와 다른 서비스 간에 보안 통신이 가능합니다. 모든 트래픽은 AWS 클라우드 내에서 안전하게 유지됩니다. 논리적 격리로 인해 퍼블릭 엔드포인트를 사용할 때에 비해, VPC에 상주하는 도메인에는 보안 계층이 하나 추가됩니다.

VPC 내에서 도메인을 생성하는 것이 좋습니다.

제한적 액세스 정책 적용

도메인이 VPC 내에 배포된 경우에도 계층으로 보안을 구현하는 것이 가장 좋습니다. 현재 액세스 정책의 구성을 확인합니다.

제한적인 리소스 기반 액세스 정책을 도메인에 적용하고 액세스 권한을 구성 API 및 OpenSearch API 작업에 부여할 때 최소 권한의 원칙을 따릅니다. 일반적인 액세스 정책에서 익명의 사용자 주체 `"Principal": {"AWS": "*" }`를 사용하지 않습니다.

단, 세분화된 액세스 제어를 사용하도록 설정하는 경우와 같이 오픈 액세스 정책을 사용하는 것이 허용되는 경우도 있습니다. 오픈 액세스 정책을 사용하면 특정 클라이언트 및 도구와 같이 요청 서명이 어렵거나 불가능한 경우 도메인에 액세스할 수 있습니다.

저장 시 암호화 사용 설정

OpenSearch Service 도메인은 데이터에 대한 무단 액세스를 방지하기 위해 저장 데이터 암호화를 제공합니다. 유휴 시 암호화는 AWS Key Management Service (AWS KMS)를 사용하여 암호화 키를 저장하고 관리하며, 256비트 키(AES-256)가 있는 고급 암호화 표준 알고리즘을 사용하여 암호화를 수행합니다.

도메인에 민감한 데이터가 저장되는 경우 저장 데이터 암호화를 사용 설정합니다.

노드 간 암호화 사용 설정

노드 간 암호화는 OpenSearch Service의 기본 보안 기능에 추가적인 보안 계층을 제공합니다. OpenSearch 내에서 프로비저닝된 노드 간의 모든 통신에 대해 전송 계층 보안(TLS)을 구현합니다. 노드 간 암호화를 사용하면 HTTPS를 통해 OpenSearch Service 도메인으로 전송된 모든 데이터가 노드 간에 배포 및 복제되는 동안 전송 중에 암호화된 상태로 유지됩니다.

도메인에 민감한 데이터가 저장되는 경우 노드 간 암호화를 사용 설정합니다.

를 사용하여 모니터링 AWS Security Hub

AWS Security Hub(을)를 사용하여 보안 모범 사례와 관련된 OpenSearch Service의 사용량을 모니터링하세요. Security Hub는 보안 제어를 사용하여 리소스 구성 및 보안 표준을 평가하여 다양한 규정 준수 프레임워크를 준수할 수 있도록 지원합니다. Security Hub를 사용하여 OpenSearch Service 리소스를 평가하는 방법에 대한 자세한 내용은 AWS Security Hub 사용 설명서의 Amazon OpenSearch Service 제어를 참조하세요.

비용 최적화

다음 모범 사례는 OpenSearch Service 비용을 최적화하고 절약하는 데 적용됩니다.

최신 세대 인스턴스 유형 사용

OpenSearch Service는 더 낮은 비용으로 더 나은 성능을 제공하는 새로운 Amazon EC2 인스턴스 유형을 항상 채택하고 있습니다. 항상 최신 세대 인스턴스를 사용하는 것이 좋습니다.

과부하가 지속되면 불안정해질 수 있으므로 프로덕션 도메인에는 T2 또는 `t3.small` 인스턴스를 사용하면 안 됩니다. `r6g.large` 인스턴스는 소규모 프로덕션 워크로드(데이터 노드 및 전용 프라이머리 노드 모두)를 위한 옵션입니다.

최신 Amazon EBS gp3 볼륨 사용

OpenSearch 데이터 노드에는 빠른 인덱싱 및 쿼리를 제공하기 위해 지연 시간이 짧고 처리량(throughput)이 높은 스토리지가 필요합니다. Amazon EBS gp3 볼륨을 사용하면 이전에 제공된 Amazon EBS gp2 볼륨 유형보다 9.6% 낮은 비용으로 더 높은 기준 성능(IOPS 및 처리량(throughput))을 얻을 수 있습니다. gp3를 사용하여 볼륨 크기와 관계없이 추가 IOPS와 처리량(throughput)을 프로비저닝할 수 있습니다. 이러한 볼륨은 또한 버스트 크레딧을 사용하지 않기 때문에 이전 세대 볼륨보다 더 안정적입니다. 또한 gp3 볼륨 유형은 gp2 볼륨 유형의 데이터 노드별 볼륨 크기 제한을 두 배로 늘

립니다. 이렇게 큰 볼륨을 사용하면 데이터 노드당 스토리지 양을 늘려 패시브 데이터의 비용을 줄일 수 있습니다.

시계열 로그 데이터에 UltraWarm 및 콜드 스토리지 사용

로그 분석에 OpenSearch를 사용하는 경우 데이터를 UltraWarm 또는 콜드 스토리지로 이동하여 비용을 절감합니다. 인덱스 상태 관리(ISM)를 사용하여 스토리지 계층 간에 데이터를 마이그레이션하고 데이터 보존을 관리할 수 있습니다.

UltraWarm은 대량의 읽기 전용 데이터를 OpenSearch Service에 저장하는 비용 효율적인 방법을 제공합니다. UltraWarm은 Amazon S3 스토리지로 사용합니다. 즉, 데이터를 변경할 수 없으며 하나의 사본만 있으면 됩니다. 인덱스의 기본 샤드 크기에 해당하는 스토리지에 대한 비용만 지불합니다. UltraWarm 쿼리의 지연 시간은 쿼리 서비스에 필요한 S3 데이터 양에 따라 증가합니다. 데이터가 노드에 캐시되면 UltraWarm 인덱스에 대한 쿼리는 핫 인덱스에 대한 쿼리와 유사하게 수행됩니다.

콜드 스토리지는 S3에서도 지원됩니다. 콜드 데이터를 쿼리해야 하는 경우 기존 UltraWarm 노드에 선택적으로 연결할 수 있습니다. 콜드 데이터는 UltraWarm과 동일한 관리 스토리지 비용이 발생하지만, 콜드 스토리지의 객체는 UltraWarm 노드 리소스를 사용하지 않습니다. 따라서 콜드 스토리지는 UltraWarm 노드 크기나 개수에 영향을 주지 않으면서 상당한 양의 스토리지 용량을 제공합니다.

UltraWarm은 핫 스토리지에서 마이그레이션할 약 2.5TiB의 데이터가 있는 경우 비용 효율적입니다. 채우기 비율을 모니터링하고 해당 데이터 볼륨에 도달하기 전에 인덱스를 UltraWarm으로 이동하도록 계획합니다.

예약 인스턴스 권장 사항 검토

성능 및 컴퓨팅 소비에 대한 기준이 양호하다면 예약 인스턴스(RI) 구매를 고려하세요. 할인은 선결제 없는 1년 예약의 경우 약 30%부터 모두 선결제한 3년 약정의 경우 최대 50%까지 증가할 수 있습니다.

최소 14일 동안 안정적인 작동이 관찰되면 AWS Cost Management 사용 설명서의 예약 권장 사항 액세스를 검토하세요. Amazon OpenSearch Service 제목에는 특정 RI 구매 권장 사항 및 예상 절감액이 표시됩니다.

Amazon OpenSearch Service 도메인 크기 조정

Amazon OpenSearch Service 도메인의 크기를 조정하는 완벽한 방법은 존재하지 않습니다. 하지만 스토리지 요구 사항, 서비스 및 OpenSearch 자체에 대해 이해한 다음 시작하면 하드웨어 요구 사항에 빈틈없이 대응하는 초기 예상치를 수립할 수 있습니다. 이러한 예상치는 도메인 크기 조정의 가장 중요한 측면인 주요 워크로드로 도메인 크기 조정 테스트와 해당 성능 모니터링을 위한 유용한 시작점을 제공할 수 있습니다.

주제

- 스토리지 요구 사항 계산
- 샤드 수 선택
- 인스턴스 유형 선택 및 테스트

스토리지 요구 사항 계산

대부분의 OpenSearch 워크로드는 두 가지 범주 중 하나로 분류됩니다.

- 장기 인덱스: 데이터를 하나 이상의 OpenSearch 인덱스로 처리하는 코드를 작성한 다음 소스 데이터가 변경됨에 따라 해당 인덱스를 주기적으로 업데이트합니다. 몇 가지 일반적인 예로 웹 사이트, 문서 및 전자 상거래 검색이 있습니다.
- 롤링 인덱스: 데이터가 인덱싱 기간과 보존 기간에 임시 인덱스 세트로 계속 유입됩니다(예: 2주 동안 보관되는 일일 인덱스 세트). 몇 가지 일반적인 예로 로그 분석, 시계열 처리 및 클릭스트림 분석이 있습니다.

장기 인덱스 워크로드의 경우 디스크에 있는 소스 데이터를 검사하여 스토리지 공간이 어느 정도 소비되었는지 쉽게 확인할 수 있습니다. 데이터를 여러 소스에서 가져온 경우 소스를 모두 추가하면 됩니다.

롤링 인덱스의 경우 주요 기간에 생성된 데이터의 양에 보존 기간을 곱할 수 있습니다. 예를 들어, 시간당 200MiB의 로그 데이터를 생성하면 하루에 4.7GiB가 생성되고 보존 기간이 2주면 이 기간에 66GiB의 데이터가 생성됩니다.

그러나 소스 데이터의 크기는 스토리지 요구 사항의 한 측면일 뿐입니다. 다음 사항도 고려해야 합니다.

- 복제본 수: 각 복제본은 기본 샤드의 전체 사본이며 인덱스의 저장 크기는 기본 및 복제본 샤드에서 가져온 크기를 보여줍니다. 기본적으로 각 OpenSearch 인덱스에는 한 개의 복제본이 포함됩니다. 데이터 손실을 방지하기 위해 하나 이상을 포함하는 것이 좋습니다. 또한 복제본은 검색 성능을 향상시켜 주므로 읽기 워크로드가 과중한 경우 여러 개를 사용할 수 있습니다. `PUT /my-index/_settings`을 사용하여 인덱스에 대한 `number_of_replicas` 설정을 업데이트합니다.
- OpenSearch 인덱싱 오버헤드: 인덱스의 디스크 크기는 다양합니다. 소스 데이터와 인덱스의 전체 크기는 대개 소스의 110%이고 인덱스는 소스 데이터의 최대 10%입니다. 데이터 인덱싱 후 `_cat/indices?v` API 및 `pri.store.size` 값을 사용하여 정확한 오버헤드를 계산할 수 있습니다. `_cat/allocation?v`에서도 유용한 요약을 제공합니다.

- 운영 체제 예약 공간: 기본적으로 Linux는 중요한 프로세스와 시스템 복구를 위해, 그리고 디스크 단편화 문제를 방지할 목적으로 `root` 사용자가 사용할 수 있도록 파일 시스템의 5%를 예약합니다.
- OpenSearch Service 오버헤드: OpenSearch Service는 세그먼트 병합, 로그 및 기타 내부 작업을 위해 각 인스턴스마다 스토리지 공간의 20%(최대 20GiB)를 예약해 둡니다.

  이 20GiB의 최댓값 때문에 예약된 총 공간은 도메인의 인스턴스 수에 따라 크게 달라질 수 있습니다. 예를 들어, 도메인에 3개의 `m6g.xlarge.search` 인스턴스가 포함될 수 있으며 각 스토리지 공간이 500GiB인 경우 총 공간은 1.46TiB입니다. 이 경우 예약된 총 공간은 60GiB에 불과합니다. 다른 도메인에 10개의 `m3.medium.search` 인스턴스가 포함될 수 있으며 각 스토리지 공간이 100GiB인 경우 총 공간은 0.98TiB입니다. 이 경우 첫 번째 도메인의 전체 스토리지 공간이 50% 더 크더라도, 두 번째 도메인의 예약된 총 공간은 200GiB입니다.

  다음 공식에서는 오버헤드에 대한 "최악의 경우" 추정치를 적용합니다. 이 추정치에는 노드 장애 및 가용 영역 중단의 영향을 최소화하는 데 도움이 되는 추가 여유 공간이 포함됩니다.

요약하면 지정된 기간에 66GiB의 데이터가 있고 한 개의 복제본이 필요한 경우 최소 스토리지 요구 사항은 66 * 2 * 1.1 / 0.95 / 0.8 = 191GiB에 근사합니다. 이 계산은 다음과 같이 일반화할 수 있습니다.

소스 데이터 * (1 + 복제본 수) * (1 + 인덱싱 오버헤드)/(1 - Linux 예약 공간)/(1 - OpenSearch Service 오버헤드) = 최소 스토리지 요구 사항

또는 아래와 같이 약식 버전을 사용할 수도 있습니다.

소스 데이터 * (1 + 복제본 수) * 1.45 = 최소 스토리지 요구 사항

스토리지 공간 부족은 클러스터 불안정성의 가장 일반적인 원인 중 하나입니다. 따라서 인스턴스 유형, 인스턴스 수, 스토리지 볼륨 선택 시 숫자를 교차 확인해야 합니다.

기타 스토리지 고려 사항은 다음과 같습니다.

- 최소 스토리지 요구 사항이 1PB를 초과하는 경우 the section called "페타바이트 규모" 섹션을 참조하세요.
- 롤링 인덱스가 있고 핫-웜 아키텍처를 사용하려면 the section called "UltraWarm 스토리지" 섹션을 참조하세요.

샤드 수 선택

스토리지 요구 사항을 이해했으면 인덱싱 전략을 조사할 수 있습니다. 기본적으로 OpenSearch Service에서 각 인덱스는 5개의 기본 샤드와 하나의 복제본(총 10개의 샤드)으로 나뉩니다. 이 동작은

기본 샤드와 하나의 복제본 샤드를 기본값으로 사용하는 오픈소스 OpenSearch와 다릅니다. 기존 인덱스에 대한 기본 샤드 수는 쉽게 변경할 수 없으므로, 첫 번째 문서를 인덱싱하기 전에 샤드 수를 결정해야 합니다.

여러 샤드를 선택하는 전반적인 목표는 클러스터의 모든 데이터 노드에서 인덱스를 균등하게 분산시키는 것입니다. 하지만 이러한 샤드는 너무 크거나 너무 많아서는 안 됩니다. 일반적인 지침은 검색 지연 시간이 핵심 성능 목표인 워크로드의 경우 샤드 크기를 10~30GiB로 유지하고, 로그 분석과 같은 쓰기 작업이 많은 워크로드의 경우 30~50GiB를 유지하는 것입니다.

크기가 큰 샤드는 OpenSearch에서 오류 발생 시 복구가 어렵기는 합니다. 하지만 각 샤드는 일정량의 CPU와 메모리를 사용하기 때문에 크기가 작은 샤드가 너무 많이 있으면 성능 문제와 메모리 부족 오류가 발생할 수 있습니다. 즉, 샤드는 기본 OpenSearch Service 인스턴스가 처리할 수 있을 정도로 작아야 하지만 너무 작아 하드웨어에 불필요한 부담을 주어서도 안 됩니다.

예를 들어, 66GiB의 데이터가 있다고 가정해 봅니다. 시간이 지남에 따라 그 수가 늘어날 것으로 예상하지 않으며 샤드를 각각 30GiB 정도로 유지하려고 합니다. 따라서 샤드 수는 약 66 * 1.1/30 = 3개가 되어야 합니다. 이 계산은 다음과 같이 일반화할 수 있습니다.

(소스 데이터 + 늘어날 공간) * (1 + 인덱싱 오버헤드)/원하는 샤드 크기 = 대략적인 기본 샤드 수

이 수식은 시간이 지남에 따라 데이터 성장 보정에 유용합니다. 동일한 66GiB의 데이터가 내년에 4배가 될 것으로 예상한다면 대략적인 샤드 수는 (66 + 198) * 1.1/30 = 10개가 됩니다. 하지만 아직 추가로 198GiB의 데이터가 필요하지는 않습니다. 향후 이 준비 작업을 통해 현재 엄청난 양의 CPU와 메모리를 소비하는 너무 작은 크기의 샤드를 생성하지 않는지 확인하세요. 이 경우 샤드당 66 * 1.1/10개 샤드 = 7.26GiB가 필요해 추가 리소스를 소비하지만 거의 권장 크기 범위에 미치지 못합니다. 샤드가 6개인 중간 정도의 접근 방식을 고려할 수 있으며, 이 경우 현재 12GiB 샤드, 향후 48GiB 샤드가 남게 됩니다. 그런 다음 다시 샤드 3개로 시작하여 샤드가 50GiB를 초과하면 데이터를 다시 인덱싱하는 것이 좋습니다.

훨씬 덜 일반적인 문제는 노드당 샤드 수 제한과 관련이 있습니다. 샤드의 크기를 적절하게 지정하면 일반적으로 디스크 공간이 먼저 소진되어 이 제한이 발생하는 경우가 거의 없습니다. 예를 들어 `m6g.large.search` 인스턴스의 최대 디스크 크기는 512GiB입니다. 디스크 사용량을 80% 미만으로 유지하고 샤드의 크기를 20GiB로 지정하면 약 20개의 샤드를 수용할 수 있습니다. Elasticsearch 7.x 이상 및 최대 2.15의 모든 OpenSearch 버전에는 노드당 1,000개의 샤드 제한이 있습니다. 노드당 최대 샤드를 조정하려면 `cluster.max_shards_per_node` 설정을 구성하세요. OpenSearch 2.17 이상의 경우 OpenSearch Service는 16GB의 JVM 힙 메모리마다 노드당 최대 4,000개의 샤드까지 1,000개의 샤드를 지원합니다. 관련 예제는 클러스터 설정을 참조하세요. 샤드 수에 대한 자세한 내용은 섹션을 참조하세요the section called “샤드 수 할당량”.

샤드의 크기를 적절하게 지정하면 이 제한을 초과하는 경우가 거의 없지만, Java 힙의 각 GiB에 대한 샤드 수를 고려해 볼 수도 있습니다. 주어진 노드에서 Java 힙의 GiB당 샤드 수는 25개 이하입니다. 예를 들어 `m5.large.search` 인스턴스의 힙은 4GiB이므로 각 노드의 샤드 수는 100개 이하여야 합니다. 샤드 수가 이와 같을 때 각 샤드의 크기는 대략 5GiB로 권장 사항보다 훨씬 작습니다.

인스턴스 유형 선택 및 테스트

스토리지 요구 사항을 계산하고 필요한 샤드 수를 선택한 후에는 하드웨어 결정을 시작할 수 있습니다. 하드웨어 요구 사항은 워크로드에 따라 크게 달라지기는 하지만 몇 가지 기본적인 권장 사항을 제공하겠습니다.

일반적으로 각 인스턴스 유형에 대한 스토리지 한도는 가벼운 워크로드에 필요한 CPU와 메모리 양에 매핑됩니다. 예를 들어, `m6g.large.search` 인스턴스는 최대 512GiB의 EBS 볼륨 크기, 2개의 vCPU 코어 및 8GiB의 메모리를 사용합니다. 클러스터에 샤드가 많이 있거나, 집계를 과도하게 수행하거나, 문서를 자주 업데이트하거나, 쿼리를 많이 처리하는 경우 해당 리소스가 충분하지 않을 수 있습니다. 클러스터가 이러한 범주 중 하나에 해당하는 경우 각 100GiB의 스토리지 요구 사항에 맞게 2개의 vCPU 코어와 8GiB의 메모리에 근접한 구성으로 시작해 보세요.

> **Tip**
>
> 각 인스턴스 유형에 할당되는 하드웨어 리소스 요약은 Amazon OpenSearch Service 요금을 참조하세요.

하지만 이러한 리소스도 부족할 수 있습니다. 일부 OpenSearch 사용자는 자신의 요구 사항을 충족시키기 위해 이와 같은 리소스가 여러 번 필요했다고 보고했습니다. 워크로드에 적합한 올바른 하드웨어를 찾으려면 초기 예상치를 치밀하게 작성하고, 주요 워크로드를 통해 테스트한 후 조정하고, 다시 테스트해야 합니다.

1단계: 초기 예상치 수립

시작할 때 브레인 분할 상태 등 잠재적인 OpenSearch 문제를 피하기 위해 최소 3개의 노드를 선택하는 것이 좋습니다(통신 경과로 클러스터에 두 개의 프라이머리 노드가 있는 경우). 3개의 전용 프라이머리 노드가 있는 경우 복제를 위해 최소 2개의 데이터 노드를 사용하는 것이 좋습니다.

2단계: 노드별 스토리지 요구 사항 계산

스토리지 요구 사항이 184GiB이고 권장되는 최소 노드 수가 3개인 경우 184/3 = 61GiB 수식을 사용하여 각 노드에 필요한 스토리지 양을 찾으세요. 이 예제에서는 3개의 `m6g.large.search` 인스턴스를 선택했고 각 인스턴스는 90GiB의 EBS 스토리지 볼륨을 사용하므로 시간이 지나면서 늘어나는 요구

사항에 대한 안전망과 공간을 확보할 수 있습니다. 이 구성은 6개의 vCPU 코어와 24GiB의 메모리를 제공하므로 더 가벼운 워크로드에 적합합니다.

더욱 실질적인 예로 14TiB(14,336GiB)의 스토리지 요구 사항과 과도한 워크로드를 고려해 보겠습니다. 이 경우 2 * 144 = 288개의 vCPU 코어 및 8 * 144 = 1,152GiB의 메모리로 시작하도록 선택할 수 있습니다. 이러한 수치는 약 18개의 `i3.4xlarge.search` 인스턴스에 해당합니다. 이렇게 빠른 로컬 스토리지가 필요 없는 경우에는 각각 1TiB의 EBS 스토리지 볼륨을 사용하여 `r6g.4xlarge.search` 인스턴스 18개로 테스트할 수도 있습니다.

귀하의 클러스터가 수백 테라바이트의 데이터를 포함한다면 the section called “페타바이트 규모” 섹션을 참조하세요.

3단계: 대표 테스트 수행

클러스터를 구성한 후에는 앞서 계산한 샤드 수를 사용하여 인덱스를 추가하고, 실제 데이터 세트를 사용하여 주요 클라이언트 테스트를 수행하고, CloudWatch 지표를 모니터링하여 클러스터가 워크로드를 처리하는 방식을 확인할 수 있습니다.

4단계: 성공 또는 반복

성능이 요구 사항을 충족하고 테스트에 성공했으며 CloudWatch 지표가 정상이면 클러스터 사용 준비를 마친 것입니다. 반드시 CloudWatch 경보를 설정하여 비정상적인 리소스가 사용되는지를 검사합니다.

성능이 기대 이하이고 테스트에 실패했거나 `CPUUtilization` 또는 `JVMMemoryPressure`가 높은 경우 다른 인스턴스 유형을 선택(또는 인스턴스 추가)하여 계속 테스트해야 할 수 있습니다. 인스턴스를 추가함에 따라 OpenSearch에서는 자동으로 클러스터 전체에 샤드 배포를 다시 조정합니다.

성능이 떨어진 클러스터에서 부족 용량을 측정하는 것보다 성능이 높은 클러스터에서 초과 용량을 측정하는 것이 더 쉬우므로 필요한 것보다 더 큰 클러스터로 시작하는 것이 좋습니다. 그런 다음, 추가 리소스가 있는 효율적인 클러스터를 테스트하고 축소하여 활동이 늘어난 기간에 안정적인 운영을 보장합니다.

프로덕션 클러스터나 상태가 복잡한 클러스터는 전용 프라이머리 노드의 이점을 활용하여 성능과 클러스터의 안정성을 향상시킵니다.

Amazon OpenSearch Service의 페타바이트 규모

Amazon OpenSearch Service 도메인은 최대 10PB의 연결된 스토리지를 제공합니다. 각각 36TB의 스토리지를 갖춘 1,000개의 `OR1.16xlarge.search` 인스턴스 유형으로 도메인을 구성할 수 있습니다.

현저한 규모 차이로 인해 이 크기의 도메인 권장 사항은 일반적인 권장 사항과 다릅니다. 이 단원에서는 도메인 생성, 비용, 스토리지, 샤드 크기에 대한 고려 사항을 설명합니다.

이 단원에서는 `i3.16xlarge.search` 인스턴스 유형을 자주 참조하지만 다른 여러 인스턴스 유형을 사용하여 총 도메인 스토리지의 10PB에 도달할 수 있습니다.

도메인 생성

이 크기의 도메인은 기본 한도(도메인당 80개의 인스턴스)를 초과합니다. 도메인당 최대 1,000개의 인스턴스에 대한 서비스 한도 증가를 요청하려면 AWS 지원 센터에서 사례를 엽니다.

요금

이 크기의 도메인을 생성하기 전에 Amazon OpenSearch Service 요금 페이지를 통해 연관된 비용이 예상치와 일치하는지 확인합니다. the section called “UltraWarm 스토리지” 검사로 핫-웜 아키텍처가 사용 사례에 적합한지 확인합니다.

스토리지

`i3` 인스턴스 유형은 빠른 로컬 NVMe(Non-Volatile Memory Express) 스토리지를 제공하기 위해 설계되었습니다. 이 로컬 스토리지는 Amazon Elastic Block Store와 비교하여 성능상 상당한 이점이 있기 때문에 OpenSearch Service에서 이러한 인스턴스 유형을 선택할 때 EBS 볼륨은 옵션이 아닙니다. EBS 스토리지를 선호한다면 `r6.12xlarge.search` 등 다른 인스턴스 유형을 사용하세요.

샤드 크기 및 개수

일반적인 OpenSearch 지침은 샤드당 50GB를 초과하지 말라는 것입니다. 대형 도메인 및 `i3.16xlarge.search` 인스턴스에 제공되는 리소스를 수용하는 데 필요한 샤드 수를 고려해 볼 때 100GB 크기의 샤드를 권장합니다.

예를 들어 450TB의 소스 데이터가 있고 한 개의 복제본이 필요한 경우 최소 스토리지 요구 사항은 450TB * 2 * 1.1 / 0.95 = 1.04PB에 가깝습니다. 이 계산에 대한 설명은 the section called “스토리지 요구 사항 계산” 섹션을 참조하세요. 1.04PB/15TB = 70개의 인스턴스가 있지만 스스로에게 스토리지 안전망을 제공하고 노드 실패를 처리하며 시간 경과에 따른 데이터 양 차이를 고려하기 위해 90개 이상의 `i3.16xlarge.search` 인스턴스를 선택할 수 있습니다. 각 인스턴스는 최소 스토리지 요구 사항에 20GiB를 더 추가하지만 이 크기의 디스크에서 20GiB는 거의 무시해도 될 정도입니다.

샤드 수를 제어하기는 까다롭습니다. OpenSearch 사용자는 매일 인덱스를 교체하고 1~2주 동안 데이터를 보관하는 경우가 많습니다. 이때 '활성'과 '비활성' 샤드를 구분하면 유용합니다. 활성화된 샤드는 아주 능동적으로 읽고 씁니다. 비활성화된 샤드는 몇몇 읽기 요청에 응하지만 대체로 유휴 상태입니다. 일반적으로 활성화된 샤드의 수는 몇천 이하여야 합니다. 활성화된 샤드의 수가 10,000을 넘어가면 성능 및 안정성에 치명적인 위협이 될 수 있습니다.

기본 샤드 수를 계산하려면 공식으로 450,000GB * 샤드당 1.1/100GB = 4,950개 샤드를 사용하세요. 복제본을 설명하기 위해서 그 수를 2배로 늘리면 9,900샤드가 되는데 모든 샤드가 활성화 상태라면 이는 주요한 문제가 됩니다. 만일 인덱스를 교체하여 어느 날이든 샤드의 1/7 또는 1/14(각각 1,414 또는 707샤드)만이 활성화 상태라면 클러스터는 정상적으로 작동합니다. 항상 그렇듯이 도메인 규모를 결정하고 구성하는 가장 중요한 단계는 실질적인 데이터 세트를 사용하여 대표적인 클라이언트 테스트를 수행하는 것입니다.

Amazon OpenSearch Service의 전용 조정자 노드

Amazon OpenSearch Service의 전용 조정자 노드는 데이터 노드에서 조정 작업을 오프로드하는 특수 노드입니다. 이러한 작업에는 검색 요청 관리 및 OpenSearch 대시보드 호스팅이 포함됩니다. 이러한 함수를 분리하면 전용 조정자 노드가 데이터 노드의 부하를 줄여 데이터 스토리지, 인덱싱 및 검색 작업에 집중할 수 있습니다. 이렇게 하면 전체 클러스터 성능과 리소스 사용률이 향상됩니다.

또한 전용 조정자 노드는 VPC 구성에 필요한 프라이빗 IP 주소 수를 줄여 네트워크 관리를 더 효율적으로 만듭니다. 이 설정을 통해 워크로드 특성에 따라 인덱싱 처리량이 최대 15% 향상되고 쿼리 성능이 20% 향상될 수 있습니다.

전용 조정자 노드를 사용해야 하는 경우

전용 조정자 노드는 다음 시나리오에서 가장 유용합니다.

- 대규모 클러스터 - 대량의 데이터 또는 복잡한 쿼리가 있는 환경에서 조정 작업을 전용 노드로 오프로드하면 클러스터 성능이 향상될 수 있습니다.
- 빈번한 쿼리 - 빈번한 검색 쿼리 또는 집계, 특히 복잡한 날짜 히스토그램 또는 여러 집계가 포함된 워크로드는 더 빠른 쿼리 처리의 이점을 누릴 수 있습니다.
- Heavy Dashboards 사용 - OpenSearch Dashboards는 리소스 집약적일 수 있습니다. 이 책임을 전용 조정자 노드로 오프로드하면 데이터 노드의 부담이 줄어듭니다.

아키텍처 및 동작

OpenSearch 클러스터에서 전용 조정자 노드는 두 가지 주요 책임을 처리합니다.

- 요청 처리 - 이러한 노드는 수신되는 검색 요청을 수신하여 관련 데이터를 저장하는 적절한 데이터 노드로 전달합니다. 그런 다음 여러 데이터 노드의 결과를 단일 글로벌 결과 세트로 통합하여 클라이언트에 반환합니다.

- 대시보드 호스팅 - 코디네이터 노드는 OpenSearch 대시보드를 관리하므로 OpenSearch 대시보드를 호스팅하고 관련 트래픽을 처리하는 데 따르는 추가 부담에서 데이터 노드를 줄일 수 있습니다.

VPC 도메인에서는 전용 조정자 노드에 데이터 노드가 아닌 탄력적 네트워크 인터페이스(ENIs 할당됩니다. 이 배열은 VPCs에 필요한 프라이빗 IP 주소 수를 줄이는 데 도움이 되므로 네트워크 효율성이 향상됩니다. 일반적으로 전용 조정자 노드는 총 데이터 노드의 약 10%를 차지합니다.

요구 사항 및 제한 사항

전용 조정자 노드에는 다음과 같은 요구 사항 및 제한 사항이 있습니다.

- 전용 조정자 노드는 모든 OpenSearch 버전 및 Elasticsearch 버전 6.8~7.10에서 지원됩니다.
- 전용 조정자 노드를 활성화하려면 도메인에 전용 마스터 노드가 활성화되어 있어야 합니다. 자세한 내용은 the section called “전용 프라이머리 노드” 단원을 참조하십시오.
- 전용 조정자 노드를 프로비저닝하면 추가 비용이 발생할 수 있습니다. 그러나 향상된 리소스 효율성과 향상된 성능은 특히 대규모 또는 복잡한 클러스터에 대한 투자를 정당화합니다.

전용 조정자 노드 프로비저닝

다음 단계를 수행하여 기존 도메인에서 전용 조정자 노드를 프로비저닝합니다. 조정자 노드를 프로비저닝하기 전에 도메인에 전용 마스터 노드가 활성화되어 있는지 확인합니다.

콘솔

에서 전용 조정자 노드를 프로비저닝하려면 AWS Management Console

1. https://console.aws.amazon.com/aos/home Amazon OpenSearch Service 콘솔에 로그인합니다.
2. 도메인을 선택한 다음 수정할 도메인을 선택합니다.
3. 클러스터 구성 섹션에서 편집을 선택합니다.
4. 전용 조정자 노드 활성화를 선택합니다.
5. 프로비저닝할 인스턴스 유형과 조정자 노드 수를 선택합니다.
6. 변경 사항 저장을 선택합니다. 도메인이 업데이트되는 데 몇 분 정도 걸릴 수 있습니다.

AWS CLI

를 사용하여 전용 조정자 노드를 프로비저닝하려면 update-domain-config 명령을 AWS CLI사용합니다. 다음 예제에서는 도메인에 세 개의 `r6g.large.search` 조정자 노드를 프로비저닝합니다.

```
aws opensearch update-domain-config \
  --domain-name my-opensearch-domain \
  --cluster-config
 InstanceCount=3,InstanceType=r6g.large.search,DedicatedCoordinatorCount=3,ZoneAwarenessEnabled
```

이 명령은 전용 조정자 노드를 활성화하고, 조정자 노드의 인스턴스 유형 및 수를 설정하고, 고가용성을 위한 영역 인식을 활성화합니다.

모범 사례

전용 조정자 노드를 사용할 때는 다음 모범 사례를 고려하세요.

- 대부분의 사용 사례에서는 범용 인스턴스를 사용합니다. 비용과 성능 간에 균형 잡힌 접근 방식을 제공합니다. 메모리 최적화 인스턴스는 복잡한 집계 또는 대규모 검색과 관련된 리소스와 같이 상당한 메모리 리소스가 필요한 워크로드에 적합합니다.
- 좋은 출발점은 데이터 노드의 5%~10%를 전용 조정자 노드로 프로비저닝하는 것입니다. 예를 들어 도메인에 특정 인스턴스 유형의 데이터 노드가 90개 있는 경우 동일한 인스턴스 유형의 조정자 노드를 5~9개 프로비저닝하는 것이 좋습니다.

  > **Note**
  >
  > 인스턴스 유형 가용성은 리전에 따라 다릅니다. 조정자 노드의 인스턴스 유형을 선택할 때 선택한 인스턴스 유형을 대상 리전에서 사용할 수 있는지 확인합니다. 도메인을 생성하거나 수정할 때 OpenSearch Service 콘솔에서 인스턴스 유형 가용성을 확인할 수 있습니다.

- 단일 장애 지점의 위험을 최소화하려면 두 개 이상의 전용 조정자 노드를 프로비저닝합니다. 이렇게 하면 노드 하나가 실패하더라도 클러스터가 계속 작동합니다.
- 리전 간 검색을 사용하는 경우 대상 도메인에 전용 조정자 노드를 프로비저닝합니다. 소스 도메인은 일반적으로 조정 작업을 처리하지 않습니다.
- 인덱싱이 많은 환경의 경우 최적의 성능을 위해 데이터 노드의 인스턴스 크기와 일치하는 CPU 최적화 인스턴스를 고려하세요.
- 메모리 집약적인 워크로드의 경우 전용 조정자 노드에 약간 더 큰 인스턴스 유형을 사용하여 증가하는 메모리 수요를 관리할 수 있습니다.
- `CoordinatorCPUUtilization` Amazon CloudWatch 지표를 추적합니다. 지속적으로 80%를 초과하는 경우 로드를 처리하기 위해 더 크거나 추가 조정자 노드가 필요할 수 있습니다.

클러스터 크기별 노드 권장 사항

클러스터 크기에 따라 전용 조정자 노드를 프로비저닝하기 위한 시작점으로 다음 지침을 사용합니다. 워크로드 특성 및 성능 지표에 따라 노드의 수와 유형을 조정합니다.

클러스터 크기	권장 조정자 노드	인스턴스 유형
소형(최대 50개의 노드)	노드 3~5개	범용
중간(50~100개 노드)	노드 5~9개	메모리 최적화
대형(100개 이상의 노드)	노드 10~15개	메모리 최적화

Amazon OpenSearch Service의 전용 프라이머리 노드

Amazon OpenSearch Service는 전용 프라이머리 노드를 사용하여 클러스터 안정성을 증가합니다. 전용 프라이머리 노드는 클러스터 관리 작업을 수행하지만 데이터를 보유하지 않거나 데이터 업로드 요청에 응답하지 않습니다. 클러스터 관리 작업을 오프로드하면 도메인의 안정성이 높아집니다. 다른 모든 노드 유형과 마찬가지로, 각 전용 프라이머리 노드에 대한 시간당 요금을 지불합니다.

전용 프라이머리 노드는 다음 클러스터 관리 작업을 수행합니다.

- 클러스터의 모든 노드를 추적합니다.
- 클러스터에 있는 인덱스 수를 추적합니다.
- 각 인덱스에 속한 샤드 수를 추적합니다.
- 클러스터에 있는 노드에 대한 라우팅 정보를 유지합니다.
- 인덱스 생성, 클러스터에서 노드 추가 또는 제거와 같은 상태 변경 후 클러스터 상태를 업데이트합니다.
- 클러스터의 모든 노드 간에 클러스터 상태 변경 사항을 복제합니다.
- 클러스터에서 데이터 노드의 가용성을 모니터링하는 주기적인 신호인 하트비트 신호를 전송하여 모든 클러스터 노드의 상태를 모니터링합니다.

다음 그림은 10개의 인스턴스가 있는 OpenSearch Service 도메인을 보여줍니다. 인스턴스 중 7개는 데이터 노드이고 3개는 전용 프라이머리 노드입니다. 전용 프라이머리 노드 중 하나만 활성화됩니다. 회색으로 표시된 전용 프라이머리 노드 2개는 활성 상태인 전용 프라이머리 노드에서 장애가 발생한 경우 백업으로 사용하기 위해 대기 중입니다. 모든 데이터 업로드 요청은 데이터 노드 7개에서 처리하고 모든 클러스터 관리 작업은 활성 상태인 전용 프라이머리 노드로 오프로드됩니다.

전용 프라이머리 노드 수 선택

Multi-AZ with Standby를 사용하여 각 프로덕션 OpenSearch Service 도메인에 3개의 전용 프라이머리 노드를 추가하는 것이 좋습니다. Multi-AZ without Standby 또는 단일 AZ로 배포하는 경우에도 전용 프라이머리 노드 3개를 사용하는 것이 좋습니다. 짝수의 전용 프라이머리 노드를 선택하지 않습니다. 전용 프라이머리 노드 수를 선택할 때 다음 사항을 고려합니다.

- 오류 발생 시 백업이 없기 때문에 OpenSearch Service에서 1개의 전용 프라이머리 노드를 명시적으로 금지합니다. 전용 프라이머리 노드가 하나만 있는 도메인을 생성하려고 시도하는 경우 유효성 검사 예외가 나타납니다.
- 2개의 전용 프라이머리 노드가 있는 경우, 오류 발생 시 클러스터에 새 프라이머리 노드를 선택하는 데 필요한 노드 쿼럼이 없습니다.

  쿼럼은 전용 프라이머리 노드 수/2 + 1(가장 가까운 정수로 반올림)입니다. 이 경우 2/2 + 1 = 2입니다. 전용 프라이머리 노드 1개에 오류가 발생했고 백업이 1개만 존재하기 때문에, 클러스터에 쿼럼이 없어 새 마스터를 선택할 수 없습니다.
- 권장되는 3개의 전용 프라이머리 노드는 프라이머리 노드에 오류가 발생했을 때 2개의 백업 노드와 새 마스터를 선택할 수 있는 필수 쿼럼(2)을 제공합니다.
- 네 개의 전용 프라이머리 노드는 세 개일 때보다 나은 점이 없으며 다중 가용 영역을 사용하는 경우 문제가 발생할 수 있습니다.
  - 1개의 프라이머리 노드에서 오류가 발생하면 쿼럼(3)이 있어 새 마스터를 선택합니다. 2개의 프라이머리 노드에서 오류가 발생하면 3개의 전용 프라이머리 노드와 마찬가지로 해당 쿼럼이 손실됩니다.
  - 세 개의 가용 영역 구성에서 2개의 AZ에는 하나의 전용 프라이머리 노드가 있고 한 AZ에는 두 개의 전용 프라이머리 노드가 있습니다. 해당 AZ가 중단되면 나머지 두 개의 AZ에는 새 마스터를 선택하는 데 필요한 쿼럼(3)이 없습니다.
- 5개의 전용 프라이머리 노드가 3개처럼 잘 작동하므로, 쿼럼을 유지 관리하는 동안 2개의 노드를 잃을 수 있습니다. 그러나 지정된 시간에 전용 프라이머리 노드 하나만 활성화되기 때문에 이 구성은 4개의 유휴 노드에 대한 비용을 지불하게 됩니다. 많은 사용자가 이 수준의 장애 조치 보호를 과하게 사용하고 있습니다.

클러스터에 마스터로 적합한 노드가 짝수 개 있는 경우 OpenSearch 및 Elasticsearch 버전 7.x 이상에서는 투표 구성이 항상 홀수가 되도록 하나의 노드를 무시합니다. 이 경우 4개의 전용 프라이머리 노드는 기본적으로 3개(2 대 1)와 동일합니다.

> **Note**
>
> 귀하의 클러스터에 새 프라이머리 노드를 선택하는 데 필요한 쿼럼이 없는 경우, 클러스터에 대한 읽기 및 쓰기 요청이 모두 실패로 끝납니다. 이 동작은 OpenSearch의 기본값과 다릅니다.

전용 프라이머리 노드에 대한 인스턴스 유형 선택

OpenSearch Service 도메인 및 인스턴스 할당량

전용 프라이머리 노드가 검색 및 쿼리 요청을 처리하지 않더라도 전용 프라이머리 노드의 크기는 자신이 관리할 수 있는 인스턴스, 인덱스 및 샤드의 수와 밀접한 관련이 있습니다. 프로덕션 클러스터의 경우 전용 프라이머리 노드에 대해 다음과 같은 인스턴스 유형이 권장됩니다.

다음 권장 사항은 일반적인 워크로드를 기반으로 한 것이며 필요에 따라 달라질 수 있습니다. 샤드나 필드 매핑이 여러 개인 클러스터는 대규모 인스턴스 유형을 활용할 수 있습니다. 자세한 내용은 Amazon OpenSearch Service에 권장되는 CloudWatch 경보를 참조하여 더 큰 인스턴스 유형을 사용해야 하는지 확인하세요.

RAM	Elasticsearch 및 OpenSearch Service 1.x~2.15에 대한 최대 노드 지원	Elasticsearch 및 OpenSearch Service 1.x~2.15에 대한 최대 샤드 지원	OpenSearch Service 2.17 이상에 대한 최대 노드 지원	OpenSearch Service 2.17 이상에 대한 최대 샤드 지원
2GB	해당 사항 없음	해당 사항 없음	10	1K
4GB	해당 사항 없음	해당 사항 없음	10	5K
8GB	10	1만	30	15K
16 GB	30	30K	60	30K
32GB	75	40K	120	60K

RAM	Elasticsearch 및 OpenSearch Service 1.x~2.15에 대한 최대 노드 지원	Elasticsearch 및 OpenSearch Service 1.x~2.15에 대한 최대 샤드 지원	OpenSearch Service 2.17 이상에 대한 최대 노드 지원	OpenSearch Service 2.17 이상에 대한 최대 샤드 지원
64GB	125	75K	240	120K
128GB	200	75K	480	240K
256GB	해당 사항 없음	해당 사항 없음	1002	50만

Amazon OpenSearch Service에 권장되는 CloudWatch 경보

CloudWatch 경보는 CloudWatch 지표가 일정 시간 동안 지정된 값을 초과하면 조치를 수행합니다. 예를 들어 클러스터 상태가 AWS 1분 이상 `red` 지속되면 이메일을 보낼 수 있습니다. 이 단원에는 Amazon OpenSearch Service에 권장되는 몇 가지 경보와 이에 대응하는 방법이 포함되어 있습니다.

를 사용하여 이러한 경보를 자동으로 배포할 수 있습니다 AWS CloudFormation. 샘플 스택은 관련 GitHub 리포지토리를 참조하세요.

> **Note**
>
> CloudFormation 스택을 배포하는 경우 `KMSKeyError` 및 `KMSKeyInaccessible` 경보는 `Insufficient Data` 상태로 존재하게 됩니다. 이러한 지표는 도메인에서 암호화 키에 문제가 발생한 경우에만 나타나기 때문입니다.

경보 구성에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 경보 생성을 참조하세요.

경보	문제
`ClusterStatus.red` 최댓값은 1분, 연속 횟수 1번 동안 >= 1임	하나 이상의 기본 샤드와 복제본이 노드에 할당되지 않았습니다. the section called “빨간색 클러스터 상태” 섹션을 참조하세요.

경보	문제
`ClusterStatus.yellow` 최댓값은 1분, 연속 횟수 5번 동안 >= 1임	하나 이상의 복제 샤드가 노드에 할당되지 않았습니다. the section called "노란색 클러스터 상태" 섹션을 참조하세요.
`FreeStorageSpace` 최소값은 1분, 연속 횟수 1번 동안 <= 20480임	클러스터 속 노드의 여유 스토리지 공간이 20GiB까지 떨어졌습니다. the section called "사용 가능한 스토리지 공간 부족" 섹션을 참조하세요. 이 값은 MiB 단위이므로 20480이 아닌 각 노드에 대한 총 스토리지 공간의 25%로 설정하는 것이 좋습니다.
`ClusterIndexWritesBlocked` 은 5분, 연속 1회 동안 >= 1임	클러스터가 쓰기 요청을 차단하고 있습니다. the section called "ClusterBlockException" 섹션을 참조하세요.
`Nodes` 최소값은 1일, 연속 횟수 1번 동안 < x임	x는 클러스터의 노드 수입니다. 이 경보는 클러스터에서 하나 이상의 노드가 하루 동안 연결되지 않았음을 나타냅니다. the section called "실패한 클러스터 노드" 섹션을 참조하세요.

`AutomatedSnapshotFailure` 최댓값은 1분, 연속 횟수 1번 동안 >= 1임	자동 스냅샷에 오류가 발생했습니다. 이런 오류는 red 클러스터 상태로 인해 자주 발생했습니다. the section called "빨간색 클러스터 상태" 섹션을 참조하세요.
	
	모든 자동 스냅샷과 오류에 대한 일부 정보 요약을 위해 다음 요청 중 하나를 시도합니다.
	
	`GET domain_endpoint /_snapshot/cs-automated/_all`
	`GET domain_endpoint /_snapshot/cs-automated-enc/_all`

`CPUUtilization` 또는 `WarmCPUUtilization` 최댓값은 15분, 연속 횟수 3번 동안 >= 80%임	때때로 100% CPU 사용률이 발생할 수 있지만 사용률이 높게 지속되는 것은 문제가 됩니다. 더 큰 인스턴스 유형을 사용하거나 인스턴스 추가를 고려하세요.

경보	문제
`JVMMemoryPressure` 최댓값은 1분, 연속 횟수 3번 동안 >= 95%임	사용량이 늘어나면 클러스터에서 메모리 부족 오류가 발생할 수 있습니다. 수직 확장을 고려하세요. OpenSearch Service는 Java 힙에 인스턴스 RAM의 절반을 사용합니다(최대 힙 크기 32GiB). 인스턴스를 최대 64GiB의 RAM까지 수직 확장할 수 있으며 인스턴스를 추가하면 수평 확장도 가능합니다.
`OldGenJVMMemoryPressure` 최댓값은 1분, 연속 횟수 3번 동안 >= 80%임	
`MasterCPUUtilization` 최댓값은 15분, 연속 횟수 3번 동안 >= 50%임	전용 프라이머리 노드에 더 큰 인스턴스 유형의 사용을 고려하세요. 클러스터 안정성 및 블루/그린(Blue/Green) 배포의 역할로 인해 전용 프라이머리 노드는 데이터 노드보다 CPU 사용량이 적어야 합니다.
`MasterJVMMemoryPressure` 최댓값은 1분, 연속 횟수 3번 동안 >= 95%임	
`MasterOldGenJVMMemoryPressure` 최댓값은 1분, 연속 횟수 3번 동안 >= 80%임	
`KMSKeyError` 은 1분, 연속 횟수 1번 동안 >= 1임	도메인에 저장된 데이터를 암호화하는 데 사용되는 암호화 AWS KMS 키는 비활성화됩니다. 정상 작동으로 복원하려면 다시 활성화해야 합니다. 자세한 내용은 the section called “저장 시 암호화” 섹션을 참조하세요.
`KMSKeyInaccessible` 은 1분, 연속 횟수 1번 동안 >= 1임	도메인의 저장 데이터를 암호화하는 데 사용되는 AWS KMS 암호화 키가 삭제되었거나 OpenSearch Service에 대한 권한 부여를 취소했습니다. 이 상태의 도메인은 복원할 수 없습니다. 하지만 수동 스냅샷이 있는 경우 새 도메인으로 마이그레이션하는 데 해당 스냅샷을 사용할 수 있습니다. 자세한 내용은 the section called “저장 시 암호화”를 참조하세요.

경보	문제
`shards.active` 는 1분, 연속 횟수 1번 동안 >= 30,000임	활성된 기본 및 복제본 샤드의 총 개수가 30,000개 이상입니다. 인덱스를 너무 자주 회전하고 있는 것일 수 있습니다. 특정 수명에 도달하면 ISM을 사용하여 인덱스를 제거하는 것이 좋습니다.
5xx 경보 >= `OpenSearchRequests` 의 10%	1개 이상의 데이터 노드가 오버로드됐거나 요청이 유휴 제한 시간 내에 완료하는 데 실패했습니다. 더 큰 인스턴스 유형으로 전환하거나 클러스터에 노드를 추가하는 것이 좋습니다. 샤드 및 클러스터 아키텍처 모범 사례를 준수하고 있는지 확인하세요.
`MasterReachableFromNode` 최댓값은 5분 동안 1 미만(연속 횟수 1회)	이 경보는 프라이머리 노드가 중지됐거나 도달할 수 없음을 나타냅니다. 이러한 장애는 일반적으로 네트워크 연결 문제 또는 AWS 종속성 문제의 결과입니다.
`ThreadpoolWriteQueue` 평균은 1분, 연속 횟수 1번 동안 >= 100임	클러스터의 인덱싱 동시성이 높습니다. 인덱싱 요청을 검토 및 제어하거나 클러스터 리소스를 늘리세요.
`ThreadpoolSearchQueue` 평균은 1분, 연속 횟수 1번 동안 >= 500임	클러스터의 검색 동시성이 높습니다. 클러스터 크기 조정을 고려하세요. 검색 대기열 크기를 늘릴 수도 있지만 지나치게 늘리면 메모리 부족 오류가 발생할 수 있습니다.
`ThreadpoolSearchQueue` 최댓값은 1분, 연속 횟수 1번 동안 >= 5,000임	
`ThreadpoolSearchRejected` 합계의 증량은 1분, 연속 횟수 1번 동안 >=1{ 수학식 DIFF ( )} 임	이러한 경보는 성능 및 안정성에 영향을 줄 수 있는 도메인 문제를 알려줍니다.

경보	문제
`Threadpool WriteRejected` 합계의 증량은 1분, 연속 횟수 1번 동안 >=1{ 수학식 DIFF ( )} 임	

> **Note**
>
> 지표만 확인하려면 the section called “클러스터 지표 모니터링” 섹션을 참조하세요.

고려할 만한 기타 경보

정기적으로 사용하는 OpenSearch Service 기능에 따라 다음 경보 구성을 고려하세요.

경보	문제
`WarmFreeStorageSpace` 는 10% 이상	사용 가능한 전체 웜 스토리지의 10%에 도달했습니다. `WarmFreeStorageSpace` 는 사용 가능한 웜 스토리지 공간의 합계(MiB)를 측정합니다. UltraWarm은 연결된 디스크 대신 Amazon S3를 사용합니다.
`HotToWarmMigrationQueueSize` 는 1분, 연속 횟수 3번 동안 >= 20임	많은 수의 인덱스가 동시에 핫 스토리지에서 UltraWarm 스토리지로 이동하고 있습니다. 클러스터 크기 조정을 고려하세요.
`HotToWarmMigrationSuccessLatency` 는 >= 1일, 연속 횟수 1번임	일일 인덱스를 회전하려고 할 때 `HotToWarmMigrationSuccessCount` x 대기 시간이 24시간 이상인 경우 알림을 받을 수 있도록 이 경보를 구성하세요.

경보	문제
`WarmJVMMemoryPressure` 최댓값은 1분, 연속 횟수 3번 동안 >= 95%임	사용량이 늘어나면 클러스터에서 메모리 부족 오류가 발생할 수 있습니다. 수직 확장을 고려하세요. OpenSearch Service는 Java 힙에 인스턴스 RAM의 절반을 사용합니다(최대 힙 크기 32GiB). 인스턴스를 최대 64GiB의 RAM까지 수직 확장할 수 있으며 인스턴스를 추가하면 수평 확장도 가능합니다.
`WarmOldGenJVMMemoryPressure` 최댓값은 1분, 연속 횟수 3번 동안 >= 80%임	
`WarmToColdMigrationQueueSize` 는 1분, 연속 횟수 3번 동안 >= 20임	많은 수의 인덱스가 동시에 UltraWarm에서 콜드 스토리지로 이동하고 있습니다. 클러스터 크기 조정을 고려하세요.
`HotToWarmMigrationFailureCount` 은 1분, 연속 횟수 1번 동안 >= 1임	스냅샷, 샤드 재배치 또는 강제 합병 중 마이그레이션이 실패할 수도 있습니다. 스냅샷 또는 샤드 재배치 중 실패는 일반적으로 노드 오류 또는 S3 연결 문제로 인해 발생합니다. 일반적으로 디스크 공간 부족이 강제 병합 실패의 근본 원인입니다.
`WarmToColdMigrationFailureCount` 은 1분, 연속 횟수 1번 동안 >= 1임	마이그레이션 실패는 인덱스 메타데이터를 콜드 스토리지로 마이그레이션하려는 시도가 실패할 때 주로 발생합니다. 웜 인덱스 클러스터 상태가 삭제될 때도 실패가 발생할 수 있습니다.
`WarmToColdMigrationLatency` 는 >= 1일, 연속 횟수 1번임	일일 인덱스를 회전하려고 할 때 `WarmToColdMigrationSuccessCount` x 대기 시간이 24시간 이상인 경우 알림을 받을 수 있도록 이 경보를 구성하세요.

경보	문제
`AlertingDegraded` 은 1분, 연속 횟수 1번 동안 >= 1임	알림 인덱스가 빨간색이거나 1개 이상의 노드가 스케줄을 따르지 않습니다.
`ADPluginUnhealthy` 은 1분, 연속 횟수 1번 동안 >= 1임	실패율이 높거나 사용되는 인덱스 중 1개 이상이 빨간색이기 때문에 이상 탐지 플러그인이 제대로 작동하지 않습니다.
`AsynchronousSearchFailureRate` 은 1분, 연속 횟수 1번 동안 >= 1임	마지막 순간에 1개 이상의 비동기 검색이 실패했으며. 이는 코디네이터 노드가 실패했을 가능성이 높음을 의미합니다. 비동기 검색 요청의 수명 주기는 코디네이터 노드에서만 관리되므로 코디네이터에 오류가 생기면 요청이 실패합니다.
`AsynchronousSearchStoreHealth` 은 1분, 연속 횟수 1번 동안 >= 1임	지속된 인덱스의 비동기 검색 응답 저장소 상태가 빨간색입니다. 클러스터를 불안정하게 만들 수 있는 큰 비동기 응답을 저장하고 있을 수도 있습니다. 비동기 검색 응답을 10MB 이하로 제한하세요.
`SQLUnhealthy` 는 1분, 연속 횟수 3번 동안 >= 1임	SQL 플러그인이 5xx 응답 코드를 반환하거나 유효하지 않은 쿼리 DSL을 OpenSearch로 넘기고 있습니다. 클라이언트가 플러그인에 하는 요청을 해결하세요.
`LTRStatus.red` 은 1분, 연속 횟수 1번 동안 >= 1임	Learning to Rank 플러그인을 실행하는 데 필요한 인덱스 중 1개 이상이 기본 샤드가 없으며 작동하지 않습니다.

Amazon OpenSearch Service에 대한 일반 참조

Amazon OpenSearch Service에서는 다양한 인스턴스, 작업, 플러그인 및 기타 리소스를 지원합니다.

주제

- Amazon OpenSearch Service에서 지원되는 인스턴스 유형
- Amazon OpenSearch Service의 엔진 버전별 기능
- Amazon OpenSearch Service의 엔진 버전별 플러그인
- Amazon OpenSearch Service의 지원 작업
- Amazon OpenSearch Service 할당량
- Amazon OpenSearch Service의 예약 인스턴스
- Amazon OpenSearch Service에서 지원되는 기타 리소스

Amazon OpenSearch Service에서 지원되는 인스턴스 유형

Amazon OpenSearch Service에서 지원되는 인스턴스 유형은 다음과 같습니다. 모든 리전에서 모든 인스턴스 유형이 지원되는 것은 아닙니다. 제공 여부에 대한 자세한 내용은 Amazon OpenSearch Service 가격을 참조하세요.

어떤 인스턴스 유형이 사용 사례에 적합한지에 대한 자세한 내용은 the section called “도메인 크기 조정”, the section called “EBS 볼륨 크기 할당량” 및 the section called “네트워크 할당량” 섹션을 참조하세요.

현재 세대 인스턴스 유형

최상의 성능을 위해서는 새 OpenSearch Service 도메인을 생성할 때 다음의 인스턴스 유형을 사용하는 것이 좋습니다.

인스턴스 유형	인스턴스	제한 사항
i4i	i4i.large.search i4i.xlarge.search	i4i 인스턴스 유형은 Elasticsearch 5.1 이상 또는 모든 버전의 OpenSearch가 필요하며 EBS 볼륨 스토리지를 지원하지 않습니다.

인스턴스 유형	인스턴스	제한 사항
	i4i.2xlarge.search	
	i4i.4xlarge.search	
	i4i.8xlarge.search	
	i4i.12xlarge.search	
	i4i.16xlarge.search	
	i4i.24xlarge.search	
	i4i.32xlarge.search	

인스턴스 유형	인스턴스	제한 사항
i4g	i4g.large.search i4g.xlarge.search i4g.2xlarge.search i4g.4xlarge.search i4g.8xlarge.search i4g.16xlarge.search	i4g 인스턴스 유형은 Elasticsearch 7.9 이상 또는 모든 버전의 OpenSearch가 필요하며 EBS 스토리지 볼륨을 지원하지 않습니다.

인스턴스 유형	인스턴스	제한 사항
Graviton3	C7g.large.search C7g.xlarge.search C7g.2xlarge.search C7g.4xlarge.search C7g.8xlarge.search C7g.12xlarge.search C7g.16xlarge.search M7g.large.search M7g.xlarge.search M7g.2xlarge.search M7g.4xlarge.search M7g.8xlarge.search	Graviton3는 GP3만 지원합니다.

인스턴스 유형	인스턴스	제한 사항
	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	
	R7gd.2xlarge.search	

인스턴스 유형	인스턴스	제한 사항
	R7gd.4xlarge.search R7gd.8xlarge.search R7gd.12xlarge.search R7gd.16xlarge.search	

인스턴스 유형	인스턴스	제한 사항
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 인스턴스 유형에는 OpenSearch 2.11 이상이 필요합니다. • OR1 인스턴스는 다른 Graviton 인스턴스 유형의 마스터 노드(C6g, M6g, R6g)와만 호환됩니다.

인스턴스 유형	인스턴스	제한 사항
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	

인스턴스 유형	인스턴스	제한 사항
OM2	om2.large.search om2.xlarge.search om2.2xlarge.search om2.4xlarge.search om2.8xlarge.search om2.12xlarge.search om2.16xlarge.search	

인스턴스 유형	인스턴스	제한 사항

Im4gn	`im4gn.large.search`
	`im4gn.xlarge.search`
	`im4gn.2xlarge.search`
	`im4gn.4xlarge.search`
	`im4gn.8xlarge.search`
	`im4gn.16xlarge.search`
		• Im4gn 인스턴스 유형은 Elasticsearch 7.9 이상 또는 모든 버전의 OpenSearch가 필요하며 EBS 스토리지 볼륨을 지원하지 않습니다.
		• Im4gn 인스턴스는 다른 Graviton 인스턴스 유형(C6g, M6g, R6g, R6gd)과만 호환됩니다. 동일한 클러스터에서 Graviton 인스턴스와 비 Graviton 인스턴스를 결합할 수 없습니다.

인스턴스 유형	인스턴스	제한 사항

C5	`c5.large.search`
	`c5.xlarge.search`
	`c5.2xlarge.search`
	`c5.4xlarge.search`
	`c5.9xlarge.search`
	`c5.18xlarge.search`
		C5 인스턴스 유형에는 Elasticsearch 5.1 이상 또는 모든 버전의 OpenSearch가 필요합니다.

인스턴스 유형	인스턴스	제한 사항
C6g	c6g.large.search c6g.xlarge.search c6g.2xlarge.search c6g.4xlarge.search c6g.8xlarge.search c6g.12xlarge.search	• C6g 인스턴스 유형에는 Elasticsearch 7.9 이상 또는 모든 버전의 OpenSearch가 필요합니다. • C6g 인스턴스는 다른 Graviton 인스턴스 유형(Im4gn, M6g, R6g, R6gd)과만 호환됩니다. 동일한 클러스터에서 Graviton 인스턴스와 비 Graviton 인스턴스를 결합할 수 없습니다.

인스턴스 유형	인스턴스	제한 사항
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 인스턴스 유형에는 Elasticsearch 5.1 이상 또는 모든 버전의 OpenSearch가 필요합니다.

인스턴스 유형	인스턴스	제한 사항
M6g	m6g.large.search m6g.xlarge.search m6g.2xlarge.search m6g.4xlarge.search m6g.8xlarge.search m6g.12xlarge.search	• M6g 인스턴스 유형에는 Elasticsearch 7.9 이상 또는 모든 버전의 OpenSearch가 필요합니다. • M6g 인스턴스는 다른 Graviton 인스턴스 유형(Im4gn, C6g, R6g, R6gd)과만 호환됩니다. 동일한 클러스터에서 Graviton 인스턴스와 비 Graviton 인스턴스를 결합할 수 없습니다.

인스턴스 유형	인스턴스	제한 사항
R5	r5.large.search r5.xlarge.search r5.2xlarge.search r5.4xlarge.search r5.12xlarge.search	R5 인스턴스 유형에는 Elasticsearch 5.1 이상 또는 모든 버전의 OpenSearch가 필요합니다.

인스턴스 유형	인스턴스	제한 사항
R6g	r6g.large.search r6g.xlarge.search r6g.2xlarge.search r6g.4xlarge.search r6g.8xlarge.search r6g.12xlarge.search	• R6g 인스턴스 유형에는 Elasticsearch 7.9 이상 또는 모든 버전의 OpenSearch가 필요합니다. • R6g 인스턴스는 다른 Graviton 인스턴스 유형(Im4gn, C6g, M6g, R6gd)과만 호환됩니다. 동일한 클러스터에서 Graviton 인스턴스와 비 Graviton 인스턴스를 결합할 수 없습니다.

인스턴스 유형	인스턴스	제한 사항
R6gd	r6gd.large.search r6gd.xlarge.search r6gd.2xlarge.search r6gd.4xlarge.search r6gd.8xlarge.search r6gd.12xlarge.search r6gd.16xlarge.search	• R6gd 인스턴스 유형은 Elasticsearch 7.9 이상 또는 모든 버전의 OpenSearch가 필요하며 EBS 스토리지 볼륨을 지원하지 않습니다. • R6gd 인스턴스는 다른 Graviton 인스턴스 유형(Im4gn, C6g, M6g, R6g)과만 호환됩니다. 동일한 클러스터에서 Graviton 인스턴스와 비 Graviton 인스턴스를 결합할 수 없습니다.

인스턴스 유형	인스턴스	제한 사항
T3	t3.small.search t3.medium.search	• T3 인스턴스 유형에는 Elasticsearch 5.6 이상 또는 모든 버전의 OpenSearch가 필요합니다. • 도메인이 대기 없이 프로비저닝된 경우에만 T3 인스턴스 유형을 사용할 수 있습니다. 자세한 내용은 the section called “Multi-AZ without Standby” 단원을 참조하십시오. • 도메인의 인스턴스 수가 10개 이하인 경우에만 T3 인스턴스 유형을 사용할 수 있습니다. • T3 인스턴스 유형은 UltraWarm 스토리지, 콜드 스토리지 또는 자동 조정을 지원하지 않습니다.
c7i	c7i.large.search c7i.xlarge.search c7i.2xlarge.search c7i.4xlarge.search c7i.8xlarge.search c7i.12xlarge.search c7i.16xlarge.search	• c7i 인스턴스에는 Elasticsearch 5.1 이상 또는 모든 버전의 OpenSearch가 필요하며 GP3 스토리지 볼륨만 지원합니다.

인스턴스 유형	인스턴스	제한 사항
m7i	m7i.large.search m7i.xlarge.search m7i.2xlarge.search m7i.4xlarge.search m7i.8xlarge.search m7i.12xlarge.search m7i.16xlarge.search	• m7i 인스턴스에는 Elasticsearch 5.1 이상 또는 모든 버전의 OpenSearch가 필요하며 GP3 스토리지 볼륨만 지원합니다.

인스턴스 유형	인스턴스	제한 사항
r7i	r7i.large.search r7i.xlarge.search r7i.2xlarge.search r7i.4xlarge.search r7i.8xlarge.search r7i.12xlarge.search r7i.16xlarge.search	• r7i 인스턴스에는 Elasticsearch 5.1 이상 또는 모든 버전의 OpenSearch가 필요하며 GP3 스토리지 볼륨만 지원합니다.

이전 세대 인스턴스 유형

OpenSearch Service에서는 이전 세대 인스턴스 유형을 기준으로 애플리케이션을 최적화했으며 아직 업그레이드하지 않은 사용자를 위해 이전 세대 인스턴스 유형을 제공합니다. 최상의 성능을 얻으려면 현재 세대 인스턴스 유형을 사용할 것을 권장합니다. 물론 다음과 같은 이전 세대 인스턴스 유형도 계속 지원됩니다.

인스턴스 유형	인스턴스	제한 사항
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 인스턴스 유형은 저장된 데이터 암호화, 세분화된 액세스 제어 또는 클러스터 간 검색을 지원하지 않습니다. • M3 인스턴스 유형에는 OpenSearch 버전에 따라 추가 제한 사항이 있습니다. 자세한 내용은 the section called “잘못된 M3 인스턴스 유형” 섹션을 참조하세요.
M4	m4.large.search	

인스턴스 유형	인스턴스	제한 사항
	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 인스턴스 유형은 저장된 데이터 암호화 또는 세분화된 액세스 제어를 지원하지 않습니다.

인스턴스 유형	인스턴스	제한 사항

R4	`r4.large.search`
	`r4.xlarge.search`
	`r4.2xlarge.search`
	`r4.4xlarge.search`
	`r4.8xlarge.search`
	`r4.16xlarge.search`	

T2	`t2.micro.search`
	`t2.small.search`
	`t2.medium.search`
		• 도메인의 인스턴스 수가 10개 이하인 경우에만 T2 인스턴스 유형을 사용할 수 있습니다.
		• `t2.micro.search` 인스턴스 유형은 Elasticsearch 1.5 및 2.3만 지원합니다.
		• T2 인스턴스 유형은 저장된 데이터 암호화, 세분화된 액세스 제어, UltraWarm 스토리지, 콜드 스토리지, 클러스터 간 검색 또는 자동 조정을 지원하지 않습니다.

Tip

전용 프라이머리 노드와 데이터 노드에 다른 인스턴스 유형을 사용하는 것이 좋습니다.

Amazon OpenSearch Service의 엔진 버전별 기능

많은 OpenSearch Service 기능에는 최소 OpenSearch 버전 요구 사항 또는 레거시 Elasticsearch OSS 버전 요구 사항이 있습니다. 기능의 최소 버전을 충족하지만 도메인에서 이 기능을 사용할 수 없는 경우 도메인의 서비스 소프트웨어를 업데이트하세요.

Feature	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
사용자 지정 플러그인	2.15	포함되지 않음
전용 조정자 노드	1.0	6.8
VPC 지원	1.0	1.0
도메인에 대한 모든 트래픽에 HTTPS 필요		
다중 AZ 지원		
전용 프라이머리 노드		
사용자 지정 패키지		
사용자 지정 엔드포인트		
느린 로그 게시		
오류 로그 게시	1.0	5.1

Feature	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
저장된 데이터 암호화		
OpenSearch 대시보드에 대한 Cognito 인증		
인 플레이스(in-place) 업그레이드		
Curator 지원	포함되지 않음	5.1
매 시간 자동화된 스냅샷	1.0	5.3
노드 간 암호화	1.0	6.0
Java 상위 수준 REST 클라이언트 지원		
HTTP 요청 및 응답 압축		
알림	1.0	6.2
SQL	1.0	6.5
클러스터 간 검색	1.0	6.7

Feature	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
세분화된 액세스 제어		
OpenSearch 대시보드에 대한 SAML 인증		
자동 조정		
원격 재인덱스		
UltraWarm	1.0	6.8
인덱스 상태 관리		
유클리드 거리별 k-NN	1.0	7.1
이상 탐지	1.0	7.4
k-NN 코사인 유사도	1.0	7.7
순위 학습		
파이프 처리 언어	1.0	7.9
OpenSearch 대시보드 보고서		

Feature	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
OpenSearch 대시보드 Trace Analytics		
ARM 기반 Graviton 인스턴스		
콜드 스토리지		
Hamming 거리, L1 표준 거리 및 k-NN에 대한 Painless 스크립팅	1.0	7.10
비동기 검색		
인덱스 변환	1.0	포함되지 않음
클러스터 간 복제	1.1	7.10
ML Commons	1.3	포함되지 않음
알림	2.3	포함되지 않음
특정 시점 검색	2.5	포함되지 않음
새로운 기계 학습 커넥터	2.9	포함되지 않음

Feature	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
시맨틱 검색	2.9	포함되지 않음
다중 모달 시맨틱 검색	2.11	포함되지 않음
직접 쿼리 데이터 소스	2.11	포함되지 않음
동시 세그먼트 검색	2.13	포함되지 않음
자연어 쿼리 생성	2.13	포함되지 않음
Amazon Q 지원	2.17	포함되지 않음

이러한 기능 중 일부와 추가 기능을 활성화하는 플러그인에 대한 자세한 내용은 the section called “엔진 버전별 플러그인” 섹션을 참조하세요. 각 버전의 OpenSearch API에 대한 자세한 내용은 the section called “지원되는 연산자” 섹션을 참조하세요.

Amazon OpenSearch Service의 엔진 버전별 플러그인

Amazon OpenSearch Service 도메인에는 OpenSearch 커뮤니티의 플러그인이 미리 포함되어 제공됩니다. 이 서비스는 자동으로 플러그인을 배포하고 관리하지만 도메인에 대해 선택된 OpenSearch 또는 레거시 Elasticsearch OSS 버전에 따라 다른 플러그인을 배포합니다.

다음 표에는 OpenSearch 버전별 플러그인과 호환 가능한 레거시 Elasticsearch OSS 버전이 나와 있습니다. 여기에는 상호 작용할 수 있는 플러그인만 포함되며, 종합적이지 않습니다. OpenSearch Service는 스냅샷용 S3 Repository 플러그 인과 최적화 및 모니터링용 OpenSearch Performance Analyzer 플러그 인 같은 추가 플러그 인을 사용하여 핵심 서비스 기능을 지원합니다. 도메인에서 실행 중인 모든 플러그인의 전체 목록을 보려면 다음을 요청하세요.

```
GET _cat/plugins?v
```

플러그인	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
HanLP	2.11	지원되지 않음
히브리어 분석	2.11	지원되지 않음
Amazon Personalize 검색 순위	2.9	지원되지 않음
신경망 검색	2.9	지원되지 않음
보안 분석	2.5	지원되지 않음
OpenSearch 알림	2.3	지원되지 않음
ML Commons	1.3	지원되지 않음
Sudachi 분석(일본어에 권장됨)	1.3	지원되지 않음
STConvert	1.3	지원되지 않음
Pinyin 분석	1.3	지원되지 않음
Nori 분석	1.3	지원되지 않음
OpenSearch 관찰	1.2	지원되지 않음
OpenSearch 클러스터 간 복제	1.1	7.10

플러그인	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
OpenSearch 비동기 검색	1.0	7.10
IK (Chinese) Analysis	1.0	7.7
Vietnamese Analysis		
Thai analysis		
순위 학습		
OpenSearch 이상 탐지	1.0	7.4
OpenSearch k-NN	1.0	7.1
OpenSearch 인덱스 상태 관리	1.0	6.8
OpenSearch 보안	1.0	6.7
OpenSearch SQL	1.0	6.5
OpenSearch 알림	1.0	6.2
Ukrainian Analysis	1.0	5.3

플러그인	필요한 최소 OpenSearch 버전	필요한 최소 Elasticsearch 버전
Mapper Size	1.0	5.3
Mapper Murmur3	1.0	5.1
Ingest User Agent Processor	1.0	5.1
Ingest Attachment Processor	1.0	5.1
Stempel Polish Analysis	1.0	5.1
Smart Chinese Analysis	1.0	5.1
Seunjeon 한국어 분석	1.0	5.1
Phonetic Analysis	1.0	2.3
Japanese (kuromoji) Analysis	1.0	모든 도메인에 포함됨
ICU Analysis	1.0	모든 도메인에 포함됨

옵션 플러그인

Amazon OpenSearch Service는 사전 설치된 기본 플러그인 외에도 여러 선택적 언어 분석기 플러그인을 지원합니다. AWS Management Console 및를 사용하여 플러그인을 도메인에 연결하고, 플러그인을 도메인에서 연결 해제하고, 모든 플러그인을 나열 AWS CLI 할 수 있습니다. 선택적 플러그인 패키지는 특정 OpenSearch 버전과 호환되며 해당 버전의 도메인에만 연결할 수 있습니다.

Sudachi 플러그인의 경우 사전 파일을 다시 연결해도 도메인에 즉시 반영되지 않습니다. 구성 변경 또는 기타 업데이트의 일환으로 도메인에서 다음 블루/그린 배포가 실행되면 사전이 새로 고쳐집니다. 또는 업데이트된 데이터로 새 패키지를 생성하고 이 새 패키지를 사용하여 새 인덱스를 생성하며 기존 인덱스를 새 인덱스로 다시 인덱싱한 후 이전 인덱스를 삭제할 수 있습니다. 재인덱싱 방식을 사용하려는 경우 트래픽이 중단되지 않도록 인덱스 별칭을 사용하세요.

선택적 플러그인은 `ZIP-PLUGIN` 패키지 유형을 사용합니다. 플러그인에 대한 자세한 내용은 the section called “패키지” 섹션을 참조하세요.

Amazon OpenSearch Service의 지원 작업

OpenSearch Service는 많은 버전의 OpenSearch 및 레거시 Elasticsearch OSS를 지원합니다. 이어지는 섹션에서는 각 버전별로 OpenSearch Service에서 지원되는 작업을 보여줍니다.

주제

- 주요 API 차이점

주요 API 차이점

새 목록 APIs

인덱스와 샤드가 많은 대규모 클러스터를 지원하기 위해 _list/indices 및 _list/샤드와 같은 페이지 매김 지원이 포함된 새로운 List APIs를 도입했습니다. 목록 API는 페이지 매김 형식으로 인덱스 및 샤드에 대한 통계를 검색합니다. 이렇게 하면 여러 인덱스가 포함된 응답을 처리하는 작업이 간소화됩니다.

- `_list/indices`: _list/indices
- `_list/shards`: _list/샤드

기존 APIs에 대한 변경 사항

대규모 클러스터를 지원하기 위해 `_cluster/stats/<metric>/nodes/<node-filters>` API에 및와 같은 관련 통계 응답만 검색할 수 있도록 지표 필터를 추가하는 지원을 `_cluster/stats` 추가했습니다`_cluster/stats/<metric>/<index_metric>/nodes/<node-filters>`. 자세한 내용은 _cluster/stats를 참조하세요.

`cancel_after_time_interval` 요청 파라미터를 지정하여 작업 취소에 대한 `_cat/shards` API 지원을 추가했습니다. 자세한 내용은 _cat/shards를 참조하세요.

_cat API의 응답 크기 제한

데이터 및 웜 노드에서 총 인스턴스 수가 200개를 초과하는 대규모 클러스터를 지원하기 위해에서 반환하는 인덱스 수에는 10K 제한이 있습니다`_cat/segments` API. 응답의 인덱스 수가이 제한을 초과하면 API는 429 오류를 반환합니다. 이를 방지하기 위해 쿼리에와 같은 인덱스 패턴 필터를 지정할 수 있습니다`_cat/segments/<index-pattern>`.

설정 및 통계

OpenSearch Service는 “플랫” 설정 양식을 사용하는 `_cluster/settings` API에 대한 PUT 요청만 허용합니다. 확장된 설정 양식을 사용하는 요청은 거부합니다.

```
// Accepted
PUT _cluster/settings
{
  "persistent" : {
    "action.auto_create_index" : false
  }
}

// Rejected
PUT _cluster/settings
{
  "persistent": {
    "action": {
      "auto_create_index": false
    }
  }
}
```

상위 수준 Java REST 클라이언트는 확장된 양식을 사용하므로, 설정 요청을 전송해야 하는 경우 하위 수준 클라이언트를 사용하세요.

Elasticsearch 5.3 이전까지는 OpenSearch Service 도메인의 `_cluster/settings` API가 GET 메서드가 아닌 HTTP PUT 메서드만 지원했습니다. OpenSearch 및 Elasticsearch 최신 버전은 다음 예제에서와 같이 GET 메서드를 지원합니다.

```
GET https://domain-name.region.es.amazonaws.com/_cluster/settings?pretty
```

다음은 반환 예제입니다.

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

오픈 소스 OpenSearch 클러스터 및 OpenSearch Service의 응답과 특정 설정 및 통계 API를 비교하면 필드가 누락된 것을 알 수 있습니다. OpenSearch Service는 `_nodes/stats`에서의 파일 시스템 데이터 경로 또는 `_nodes`에서의 운영 체제 이름 및 버전과 같이 서비스 내부를 노출하는 특정 정보를 수정합니다.

축소

_shrink API는 업그레이드, 구성 변경 및 도메인 삭제를 실패하게 만들 수 있습니다. Elasticsearch 버전 5.3 또는 5.1을 실행하는 도메인에서는 사용하지 않는 것이 좋습니다. 이들 버전에는 축소된 인덱스의 스냅샷 복원이 실패할 수 있는 버그가 있습니다.

다른 Elasticsearch 또는 OpenSearch 버전에서 _shrink API를 사용하는 경우 축소 작업을 시작하기 전에 다음 요청을 수행합니다.

```
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
  }
}
```

축소 작업을 완료한 후에 다음 요청을 수행합니다.

```
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
  }
}
```

새 목록 APIs

인덱스와 샤드 수가 많은 대규모 클러스터를 지원하기 위해 페이지 매김을 지원하는 _list/indices 및와 같은 새로운 목록 APIs 도입했습니다_list/shards. 목록 API는 페이지 매김 형식으로 인덱스 및 샤드에 대한 통계를 검색합니다. 이렇게 하면 여러 인덱스가 포함된 응답을 처리하는 작업이 간소화됩니다. 에 대한 자세한 내용은 인덱스 나열을 _list/indices참조하세요. 에 대한 자세한 내용

은 샤드 나열을 _list/shards참조하세요. https://opensearch.org/docs/latest/api-reference/list/list-shards/

기존 APIs에 대한 변경 사항

대규모 클러스터를 지원하기 위해 _cluster/stats/<metric>/nodes/<node-filters> 및 에 지원이 추가되었습니다_cluster/stats/<metric>/<index_metric>/nodes/<node-filters>. 에 대한 자세한 내용은 클러스터 통계를 _cluster/stats참조하세요. https://opensearch.org/docs/latest/api-reference/cluster-api/cluster-stats/

_cat APIs의 응답 크기 제한

데이터 및 웜 노드에서 총 인스턴스 수가 200개를 초과하는 대규모 클러스터를 지원하기 위해 _cat/segments API에서 반환되는 인덱스 수에는 10,000개의 제한이 있습니다. 응답의 인덱스 수가이 제한을 초과하면 API가 429 오류를 반환합니다. 이를 방지하기 위해 쿼리에 인덱스 패턴 필터(예: _cat/segments/<index-pattern>)를 지정할 수 있습니다.

또한 이제 API에서 cancel_after_time_interval 요청 파라미터를 지정하여 _cat/shards 태스크 취소를 지원할 수 있습니다. 이에 대한 자세한 내용은 CAT 샤드를 참조하세요.

전용 마스터 노드의 인스턴스 유형 선택

다음 표에는 전용 마스터 노드에 적합한 인스턴스 유형을 선택하기 위한 권장 사항이 나와 있습니다.

RAM	지원되는 최대 노드 수	지원되는 최대 샤드
2GB	10	1,000
4GB	10	5,000
8GB	30	15,000
16 GB	60	30,000개
32GB	120	60,000
64GB	240	120,000
128GB	480	240,000

RAM	지원되는 최대 노드 수	지원되는 최대 샤드	
256GB	1002	500,000	

OpenSearch 버전 2.18 및 2.19

OpenSearch 2.18 및 2.19 작업에 대한 자세한 내용은 OpenSearch REST API 참조 또는 특정 플러그인에 대한 API 참조를 참조하세요. 이러한 릴리스의 변경 사항에 대한 자세한 내용은 2.18 릴리스 정보 및 2.19 릴리스 정보를 참조하세요.

OpenSearch 버전 2.17

OpenSearch 2.17의 경우 OpenSearch Service는 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

> **Note**
>
> OpenSearch 2.17부터 `cluster.max_shards_per_node` 설정을 수정할 수 없습니다. OpenSearch 2.17 이상의 경우 OpenSearch Service는 노드당 최대 4,000개의 샤드까지 16GB의 JVM 힙 메모리마다 1,000개의 샤드를 지원합니다.

- 인덱스 경로의 모든 작업 (예:*index-name* /_forcemerge , *index-name* /update/*id* 및 *index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_list
- /_ltr
- /_mapping
- /_mget
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search [2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile

- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
  - cluster.search.request.slowlog.threshold.info
  - cluster.search.request.slowlog.threshold.debug
  - cluster.search.request.slowlog.threshold.trace
- /_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
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - search.phase_took_enabled
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 나타내며 이상 탐지, ISM 등을 위한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

> **Note**
>
> 현재 대기 상태의 다중 AZ(가용 영역)가 있는 고객은 cluster.max_shards_per_node 설정 기능 변경이 활성화되지 않습니다.

OpenSearch 버전 2.15

OpenSearch 2.15의 경우 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index

update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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.search.request.slowlog.level
  - cluster.search.request.slowlog.threshold.warn
- /_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
- /_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. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 2.13

OpenSearch 2.13의 경우 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge , /*index-name* /update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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.search.request.slowlog.level
- /_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
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_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.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

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 2.11

OpenSearch 2.11의 경우 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge , */index-name* /update/*id* 및 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 2.9

OpenSearch 2.9의 경우 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline

- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_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
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.

3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 2.7

OpenSearch 2.7의 경우 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_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/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 2.5

OpenSearch 2.5의 경우 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 `/_tasks` 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 `/_search/scroll`에 대한 DELETE 요청은 HTTP 헤더에 `"Content-Length"`를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. `scroll_id` 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 `scroll_id` 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 2.3

OpenSearch 2.3의 경우 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:*/index-name* `/_forcemerge` , */index-name* `/update/`*id* 및 */index-name* `/_close`)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`(`/_cat/nodeattrs` 제외)
- `/_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`

- 여러 가지 속성의 /_cluster/settings [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
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 1.3

OpenSearch 1.3에서는 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge , */index-name* /update/*id* 및 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 1.2

OpenSearch 1.2에서는 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile

- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.

3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 1.1

OpenSearch 1.1에서는 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_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/_security
- /_plugins/_sql
- /_plugins/_transforms
- /_percolate
- /_rank_eval

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.

OpenSearch 버전 1.0

OpenSearch 1.0에서는 OpenSearch Service가 다음 작업을 지원합니다. 대부분의 운영에 대한 정보는 OpenSearch REST API 참조 또는 특정 플러그인의 API 참조를 확인하세요.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 `/_tasks` 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 `/_search/scroll`에 대한 DELETE 요청은 HTTP 헤더에 `"Content-Length"`를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. `scroll_id` 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 `scroll_id` 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 7.10

Elasticsearch 7.10에서는 OpenSearch Service가 다음 작업을 지원합니다.

- 인덱스 경로의 모든 작업 (예:*/index-name* `/_forcemerge` , */index-name* `/update/`*id* 및 */index-name* `/_close`)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`(`/_cat/nodeattrs` 제외)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_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]

- 여러 가지 속성의 /_cluster/settings [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
- /_opendistro/_asynchronous_search
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_plugins/_replication
- /_rank_eval
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.
6. 레거시 인덱스 템플릿(_template)은 Elasticsearch 7.8부터 구성 가능한 템플릿(_index_template)으로 교체되었습니다. 구성 가능한 템플릿은 레거시 템플릿보다 우선합니다. 지정된 인덱스와 일치하는 구성 가능 템플릿이 없는 경우 레거시 템플릿은 여전히 일치할 수 있으며

이를 적용할 수 있습니다. 이 _template 작업은 OpenSearch 및 이후 버전의 Elasticsearch OSS에서도 계속 작동하지만 두 템플릿 유형에 대한 GET 호출은 다른 결과를 반환합니다.

Elasticsearch 버전 7.9

Elasticsearch 7.9에서는 OpenSearch Service가 다음 작업을 지원합니다.

- 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge ,/*index-name* /update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.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

  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 OpenSearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소”을(를) 참조하세요.
6. 레거시 인덱스 템플릿(_template)은 Elasticsearch 7.8부터 구성 가능한 템플릿(_index_template)으로 교체되었습니다. 구성 가능한 템플릿은 레거시 템플릿보다 우선합니다. 지정된 인덱스와 일치하는 구성 가능 템플릿이 없는 경우 레거시 템플릿은 여전히 일치할 수 있으며 이를 적용할 수 있습니다. 이 _template 작업은 OpenSearch 및 이후 버전의 Elasticsearch OSS에서도 계속 작동하지만 두 템플릿 유형에 대한 GET 호출은 다른 결과를 반환합니다.

Elasticsearch 버전 7.8

Elasticsearch 7.8에서는 OpenSearch Service가 다음 작업을 지원합니다.

- 인덱스 경로의 모든 작업(예:/*index-name* /_forcemerge ,/*index-name* /
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_refresh
- /_reindex [1]
- /_render
- /_rollover

update/*id* 및 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_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/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다.

scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.

3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.
6. 레거시 인덱스 템플릿(_template)은 Elasticsearch 7.8부터 구성 가능한 템플릿(_index_template)으로 교체되었습니다. 구성 가능한 템플릿은 레거시 템플릿보다 우선합니다. 지정된 인덱스와 일치하는 구성 가능 템플릿이 없는 경우 레거시 템플릿은 여전히 일치할 수 있으며 이를 적용할 수 있습니다. 이 _template 작업은 OpenSearch 및 이후 버전의 Elasticsearch OSS에서도 계속 작동하지만 두 템플릿 유형에 대한 GET 호출은 다른 결과를 반환합니다.

Elasticsearch 버전 7.7

Elasticsearch 7.7에서는 OpenSearch Service가 다음 작업을 지원합니다.

- 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge , */index-name* /update/*id* 및 */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template

- 여러 가지 속성의 `/_cluster/settings` [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`
- `/_opendistro/_alerting`
- `/_opendistro/_anomaly_detection`
- `/_opendistro/_ism`
- `/_opendistro/_security`
- `/_opendistro/_sql`
- `/_percolate`
- `/_plugin/kibana`
- `/_rank_eval`
- `/_update_by_query` [1]
- `/_validate`

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 `/_tasks` 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 `/_search/scroll`에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. `scroll_id` 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 `scroll_id` 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 7.4

Elasticsearch 7.4에서는 OpenSearch Service가 다음 작업을 지원합니다.

- 인덱스 경로의 모든 작업 (예:`/index-name /_forceme`
- `/_cluster/state`
- `/_cluster/stats`
- `/_refresh`
- `/_reindex` [1]

rge , */index-name* / update/*id* 및 */index-name* / _close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.

2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 7.1

Elasticsearch 7.1에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge 및 */index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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/_ism
- /_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
- cluster.max_shards_per_node

- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 6.8

Elasticsearch 6.8에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:/*index-name* /_forceme

- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]

- /_refresh
- /_reindex [1]
- /_render
- /_rollover

rge 및 /*index-name* / update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_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
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.

2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 6.7

Elasticsearch 6.7에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:/*index-name* /_forcemerge 및 /*index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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/_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

- 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

- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 6.5

Elasticsearch 6.5에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:/*index-name* /_forceme

- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]

- /_refresh
- /_reindex [1]
- /_render
- /_rollover

rge 및 /*index-name* / update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.

3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 6.4

Elasticsearch 6.4에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge 및 */index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_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

- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 6.3

Elasticsearch 6.3에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge 및 */index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]

- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 6.2

Elasticsearch 6.2에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업(예:*/index-name* /_forcemerge 및 */index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 `/_tasks` 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 `/_search/scroll`에 대한 DELETE 요청은 HTTP 헤더에 `"Content-Length"`를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. `scroll_id` 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 `scroll_id` 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 6.0

Elasticsearch 6.0에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* `/_close`를 제외한 인덱스 경로의 모든 작업 (예:*/index-name* `/_forcemerge` 및 */index-name* `/update/`*id*)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`(`/_cat/nodeattrs` 제외)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_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`
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search` [2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink` [5]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

- 여러 가지 속성의 `/_cluster/settings` [4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_refresh`
- `/_reindex` [1]

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 `/_tasks` 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 `/_search/scroll`에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. `scroll_id` 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 `scroll_id` 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 5.6

Elasticsearch 5.6에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* `/_close`를 제외한 인덱스 경로의 모든 작업(예:*/index-name* `/_forceme`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]

rge 및 /*index-name* / update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.

3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 5.5

Elasticsearch 5.5에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge 및 */index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [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
- /_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

- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. 스크립트 사용과 관련된 고려 사항은 the section called “지원되는 기타 리소스” 섹션을 참조하세요.
4. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
5. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 5.3

Elasticsearch 5.3에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업 (예:*/index-name* /_forcemerge 및 */index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_render
- /_rollover
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[4]
- /_snapshot
- /_stats
- /_status

- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings [3]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.
2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. PUT 메서드를 참조하세요. GET 메서드에 대한 자세한 내용은 the section called “주요 API 차이점” 섹션을 참조하세요. 이 목록은 OpenSearch Service가 지원하는 일반 Elasticsearch 작업만 참조하며 이상 탐지, ISM 등에 대한 플러그인별 지원 작업은 포함하지 않습니다.
4. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 5.1

Elasticsearch 5.1에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업(예:/*index-name* /_forcemerge 및 /*index-name* /update/*id*)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 여러 가지 속성의 /_cluster/settings (PUT만 해당):
  - 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. 클러스터 구성을 변경하면 완료 전 이러한 작업이 중단될 수 있습니다. 이러한 작업과 함께 /_tasks 작업을 사용하여 요청이 성공적으로 완료되었는지 확인하는 것이 좋습니다.

2. 메시지 본문이 있는 /_search/scroll에 대한 DELETE 요청은 HTTP 헤더에 "Content-Length"를 지정해야 합니다. 대부분의 클라이언트는 기본적으로 이 헤더를 추가합니다. scroll_id 값에서 = 문자로 인한 문제를 방지하려면 쿼리 문자열이 아닌 요청 본문을 사용하여 scroll_id 값을 OpenSearch Service에 전달합니다.
3. the section called “축소” 섹션을 참조하세요.

Elasticsearch 버전 2.3

Elasticsearch 2.3에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업(예:*/index-name* /_forcemerge 및 */index-name* /_recovery)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cache/clear (인덱스에만 해당)
- /_cat(/_cat/nodeattrs 제외)
- /_cluster/health
- 여러 가지 속성의 /_cluster/settings (PUT만 해당):
  - 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
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_render
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

- threadpool.search.queue_size
- threadpool.suggest.queue_size

Elasticsearch 버전 1.5

Elasticsearch 1.5에서는 OpenSearch Service가 다음 작업을 지원합니다.

- */index-name* /_close를 제외한 인덱스 경로의 모든 작업(예:*/index-name* /_optimize 및 */index-name* /_warmer)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat
- /_cluster/health
- 여러 가지 속성의 /_cluster/settings (PUT만 해당):
  - 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
  - threadpool.suggest.queue_size
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_plugin/kibana3
- /_plugin/migration
- /_refresh
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

Amazon OpenSearch Service 할당량

AWS 계정에는 각 AWS 서비스에 대해 이전에 제한이라고 하는 기본 할당량이 있습니다. 다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다.

OpenSearch Service 도메인 및 인스턴스, Amazon OpenSearch Serverless, Amazon OpenSearch Ingestion의 할당량을 보려면 AWS 일반 참조의 Amazon OpenSearch Service 할당량을 참조하세요.

에서 OpenSearch Service의 할당량을 보려면 Service Quotas 콘솔을 AWS Management Console엽니다. 탐색 창에서 AWS services를 선택하고 Amazon OpenSearch Service를 선택합니다. 할당량 증가를 요청하려면 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

UltraWarm 스토리지 할당량

다음 표에는 UltraWarm 인스턴스 유형과 각 유형에서 사용할 수 있는 최대 스토리지 용량이 나와 있습니다. UltraWarm에 대한 자세한 내용은 the section called “UltraWarm 스토리지” 섹션을 참조하세요.

인스턴스 유형	최대 스토리지
ultrawarm1.medium.search	1.5TiB
ultrawarm1.large.search	20TiB

AZ당 데이터 노드 수

다음 표에는 AZ 배포를 위한 총 데이터 노드 수가 아래에 나열되어 있습니다. 전체 제한은 핫 노드 수와 웜 노드 수를 모두 포함하여 제한당 데이터 노드 수를 나타냅니다. 각 유형에서 사용할 수 있는 스토리지입니다.

AZ 구성	핫 노드 수 제한	웜 노드 수 제한	전체 제한(핫 + 웜)
1 - AZ	334	250	334
2 - AZ	668	500	668
3 - AZ	1002	750	1002

인스턴스 패밀리별 총 노드 제한

다음 표에는 인스턴스 패밀리별 총 노드 제한이 나열되어 있습니다.

인스턴스 패밀리	ElasticSearch OpenSearch 최대 2.15	OpenSearch 2.17 이상	기본 한도
T2	10	10	10
T3	10	10	10
M3, C4, M4, R4, C5, M5, R5, I2, I3	10	200	80
Graviton 2, Gravtion 3	200	400	80
C7, R7i, M7i, i4i	200	400	80
OR1.medium.search OR1.large.search OR2.medium.search OR2.large.search OM2.large.search	200	400	80
OR1.xlarge.search 이상 OR2.xlarge.search 이상 OM2.xlarge.search 이상	200	1002	80
Ultrawarm1	150	750	150

EBS 볼륨 크기 할당량

다음 표에는 OpenSearch Service에서 지원하는 각 인스턴스 유형에 대한 최소 및 최대 EBS 볼륨 크기가 나와 있습니다. 인스턴스 스토리지 및 추가 하드웨어 세부 정보를 포함한 인스턴스 유형에 대한 정보는 Amazon OpenSearch Service 요금을 참조하세요.

- 도메인을 생성할 때 EBS 볼륨 유형에서 마그네틱 스토리지를 선택하면 t2.small 및 t2.medium을 제외한 모든 인스턴스 유형과 마그네틱 스토리지를 지원하지 않는 모든 Graviton 인스턴스(M6g, C6g, R6g 및 R6gd)에 대해 최대 볼륨 크기는 100GiB입니다. 아래 표에 나와 있는 최대 크기에 맞춰 SSD 옵션 하나를 선택합니다.
- 일부 이전 세대 인스턴스 유형은 인스턴스 스토리지를 포함할 뿐만 아니라 EBS 스토리지도 지원합니다. 이러한 인스턴스 유형 중 하나에 대해 EBS 스토리지를 선택할 경우, 스토리지 볼륨이 추가되지 않습니다. EBS 볼륨 또는 인스턴스 스토리지 중에서 하나를(둘 모두는 안됨) 선택할 수 있습니다.

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
t2.micro.search	10GiB	35GiB	N/A
t2.small.search	10GiB	35GiB	N/A
t2.medium.search	10GiB	35GiB	N/A
t3.small.search	10GiB	100GiB	100GiB
t3.medium.search	10GiB	200GiB	200GiB
m3.medium.search	10GiB	100GiB	N/A
m3.large.search	10GiB	512GiB	N/A
m3.xlarge.search	10GiB	512GiB	N/A
m3.2xlarge.search	10GiB	512GiB	N/A
m4.large.search	10GiB	512GiB	N/A
m4.xlarge.search	10GiB	1TiB	N/A

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
m4.2xlarge.search	10GiB	1.5TiB	N/A
m4.4xlarge.search	10GiB	1.5TiB	N/A
m4.10xlarge.search	10GiB	1.5TiB	N/A
m5.large.search	10GiB	512GiB	1TiB
m5.xlarge.search	10GiB	1TiB	2TiB
m5.2xlarge.search	10GiB	1.5TiB	3TiB
m5.4xlarge.search	10GiB	3TiB	6TiB
m5.12xlarge.search	10GiB	9TiB	18TiB
m6g.large.search	10GiB	512GiB	1TiB
m6g.xlarge.search	10GiB	1TiB	2TiB
m6g.2xlarge.search	10GiB	1.5TiB	3TiB
m6g.4xlarge.search	10GiB	3TiB	6TiB
m6g.8xlarge.search	10GiB	6TiB	12TiB
m6g.12xlarge.search	10GiB	9TiB	18TiB
c4.large.search	10GiB	100GiB	N/A
c4.xlarge.search	10GiB	512GiB	N/A
c4.2xlarge.search	10GiB	1TiB	N/A
c4.4xlarge.search	10GiB	1.5TiB	N/A
c4.8xlarge.search	10GiB	1.5TiB	N/A
c5.large.search	10GiB	256GiB	256GiB

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
c5.xlarge.search	10GiB	512GiB	512GiB
c5.2xlarge.search	10GiB	1TiB	1TiB
c5.4xlarge.search	10GiB	1.5TiB	1.5TiB
c5.9xlarge.search	10GiB	3.5TiB	3.5TiB
c5.18xlarge.search	10GiB	7TiB	7TiB
c6g.large.search	10GiB	256GiB	256GiB
c6g.xlarge.search	10GiB	512GiB	512GiB
c6g.2xlarge.search	10GiB	1TiB	1TiB
c6g.4xlarge.search	10GiB	1.5TiB	1.5TiB
c6g.8xlarge.search	10GiB	3TiB	3TiB
c6g.12xlarge.search	10GiB	4.5TiB	4.5TiB
r3.large.search	10GiB	512GiB	N/A
r3.xlarge.search	10GiB	512GiB	N/A
r3.2xlarge.search	10GiB	512GiB	N/A
r3.4xlarge.search	10GiB	512GiB	N/A
r3.8xlarge.search	10GiB	512GiB	N/A
r4.large.search	10GiB	1TiB	N/A
r4.xlarge.search	10GiB	1.5TiB	N/A
r4.2xlarge.search	10GiB	1.5TiB	N/A
r4.4xlarge.search	10GiB	1.5TiB	N/A

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
r4.8xlarge.search	10GiB	1.5TiB	N/A
r4.16xlarge.search	10GiB	1.5TiB	N/A
r5.large.search	10GiB	1TiB	2TiB
r5.xlarge.search	10GiB	1.5TiB	3TiB
r5.2xlarge.search	10GiB	3TiB	6TiB
r5.4xlarge.search	10GiB	6TiB	12TiB
r5.12xlarge.search	10GiB	12TiB	24TiB
r6g.large.search	10GiB	1TiB	2TiB
r6g.xlarge.search	10GiB	1.5TiB	3TiB
r6g.2xlarge.search	10GiB	3TiB	6TiB
r6g.4xlarge.search	10GiB	6TiB	12TiB
r6g.8xlarge.search	10GiB	8TiB	16TiB
r6g.12xlarge.search	10GiB	12TiB	24TiB
r6gd.large.search	N/A	해당 사항 없음	N/A
r6gd.xlarge.search	N/A	해당 사항 없음	N/A
r6gd.2xlarge.search	N/A	해당 사항 없음	N/A
r6gd.4xlarge.search	N/A	해당 사항 없음	N/A

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
r6gd.8xlarge.search	N/A	해당 사항 없음	N/A
r6gd.12xlarge.search	N/A	해당 사항 없음	N/A
r6gd.16xlarge.search	N/A	해당 사항 없음	N/A
i2.xlarge.search	10GiB	512GiB	N/A
i2.2xlarge.search	10GiB	512GiB	N/A
i3.large.search	N/A	해당 사항 없음	N/A
i3.xlarge.search	N/A	해당 사항 없음	N/A
i3.2xlarge.search	N/A	해당 사항 없음	N/A
i3.4xlarge.search	N/A	해당 사항 없음	N/A
i3.8xlarge.search	N/A	해당 사항 없음	N/A
i3.16xlarge.search	N/A	해당 사항 없음	N/A
또는1.medium.search	20GiB	N/A	768GiB
또는1.large.search	20GiB	N/A	1,532GiB
또는1.xlarge.search	20GiB	N/A	3TiB

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
또는1.2xlarge.search	20GiB	N/A	6TiB
또는1.4xlarge.search	20GiB	N/A	12TiB
또는1.8xlarge.search	20GiB	N/A	16TiB
또는1.12xlarge.search	20GiB	N/A	24TiB
또는1.16xlarge.search	20GiB	N/A	36TiB
또는2.medium.search	20GiB	N/A	768GiB
또는2.large.search	20GiB	N/A	1,532GiB
또는2.xlarge.search	20GiB	N/A	3TiB
또는2.2xlarge.search	20GiB	N/A	6TiB
또는2.4xlarge.search	20GiB	N/A	12TiB
또는2.8xlarge.search	20GiB	N/A	16TiB
또는2.12xlarge.search	20GiB	N/A	24TiB
또는2.16xlarge.search	20GiB	N/A	36TiB
om2.large.search	20GiB	N/A	768GiB
om2.xlarge.search	20GiB	N/A	2 Tib
om2.2xlarge.search	20GiB	N/A	3 Tib
om2.4xlarge.search	20GiB	N/A	6 Tib
om2.8xlarge.search	20GiB	N/A	12Tib
om2.12xlarge.search	20GiB	N/A	18 Tib
om2.16xlarge.search	20GiB	N/A	24Tib

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
im4gn.large.search	N/A	해당 사항 없음	N/A
im4gn.xlarge.search	N/A	해당 사항 없음	N/A
im4gn.2xlarge.search	N/A	해당 사항 없음	N/A
im4gn.4xlarge.search	N/A	해당 사항 없음	N/A
im4gn.8xlarge.search	N/A	해당 사항 없음	N/A
im4gn.16xlarge.search	N/A	해당 사항 없음	N/A
C7g.large.search	10GiB	N/A	256GiB
C7g.xlarge.search	10GiB	N/A	512GiB
C7g.2xlarge.search	10GiB	N/A	1TiB
C7g.4xlarge.search	10GiB	N/A	1.5TiB
C7g.8xlarge.search	10GiB	N/A	3TiB
C7g.12xlarge.search	10GiB	N/A	4.5TiB
C7g.16xlarge.search	10GiB	N/A	6TiB
M7g.medium.search	10GiB	N/A	512GiB
M7g.large.search	10GiB	N/A	768GiB
M7g.xlarge.search	10GiB	N/A	2TiB

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
M7g.2xlarge.search	10GiB	N/A	3TiB
M7g.4xlarge.search	10GiB	N/A	6TiB
M7g.8xlarge.search	10GiB	N/A	12TiB
M7g.12xlarge.search	10GiB	N/A	18TiB
M7g.16xlarge.search	10GiB	N/A	24TiB
R7g.medium.search	10GiB	N/A	768GiB
R7g.large.search	10GiB	N/A	1.5TiB
R7g.xlarge.search	10GiB	N/A	3TiB
R7g.2xlarge.search	10GiB	N/A	6TiB
R7g.4xlarge.search	10GiB	N/A	12TiB
R7g.8xlarge.search	10GiB	N/A	16TiB
R7g.12xlarge.search	10GiB	N/A	24TiB
R7g.16xlarge.search	10GiB	N/A	36TiB
R7gd.large.search	N/A	해당 사항 없음	N/A
R7gd.xlarge.search	N/A	해당 사항 없음	N/A
R7gd.2xlarge.search	N/A	해당 사항 없음	N/A
R7gd.4xlarge.search	N/A	해당 사항 없음	N/A

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
R7gd.8xlarge.search	N/A	해당 사항 없음	N/A
R7gd.12xlarge.search	N/A	해당 사항 없음	N/A
R7gd.16xlarge.search	N/A	해당 사항 없음	N/A
i4i.large.search	10GiB	N/A	N/A
i4i.xlarge.search	10GiB	N/A	N/A
i4i.2xlarge.search	10GiB	N/A	N/A
i4i.4xlarge.search	10GiB	N/A	N/A
i4i.8xlarge.search	10GiB	N/A	N/A
i4i.12xlarge.search	10GiB	N/A	N/A
i4i.16xlarge.search	10GiB	N/A	N/A
i4i.24xlarge.search	10GiB	N/A	N/A
i4i.32xlarge.search	10GiB	N/A	N/A
i4g.large.search	10GiB	N/A	N/A
i4g.xlarge.search	10GiB	N/A	N/A
i4g.2xlarge.search	10GiB	N/A	N/A
i4g.4xlarge.search	10GiB	N/A	N/A
i4g.8xlarge.search	10GiB	N/A	N/A
i4g.16xlarge.search	10GiB	N/A	N/A

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
c7i.large.search	10GiB	N/A	256GiB
c7i.xlarge.search	10GiB	N/A	512GiB
c7i.2xlarge.search	10GiB	N/A	1TiB
c7i.4xlarge.search	10GiB	N/A	1.5TiB
c7i.8xlarge.search	10GiB	N/A	3TiB
c7i.12xlarge.search	10GiB	N/A	4.5TiB
c7i.16xlarge.search	10GiB	N/A	6TiB
m7i.large.search	10GiB	N/A	768GiB
m7i.xlarge.search	10GiB	N/A	2TiB
m7i.2xlarge.search	10GiB	N/A	3TiB
m7i.4xlarge.search	10GiB	N/A	6TiB
m7i.8xlarge.search	10GiB	N/A	12TiB
m7i.12xlarge.search	10GiB	N/A	18TiB
m7i.16xlarge.search	10GiB	N/A	24TiB
r7i.large.search	10GiB	N/A	1.5TiB
r7i.xlarge.search	10GiB	N/A	3TiB
r7i.2xlarge.search	10GiB	N/A	6TiB
r7i.4xlarge.search	10GiB	N/A	12TiB
r7i.8xlarge.search	10GiB	N/A	16TiB
r7i.12xlarge.search	10GiB	N/A	24TiB

인스턴스 유형	최소 EBS 크기	최대 EBS 크기(gp2)	최대 EBS 크기(gp3)
r7i.12xlarge.search	10GiB	N/A	36TiB

네트워크 할당량

다음 표에는 HTTP 요청 페이로드의 최대 크기가 나와 있습니다.

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
t2.micro.search	10MiB
t2.small.search	10MiB
t2.medium.search	10MiB
t3.small.search	10MiB
t3.medium.search	10MiB
m3.medium.search	10MiB
m3.large.search	10MiB
m3.xlarge.search	100MiB
m3.2xlarge.search	100MiB
m4.large.search	10MiB
m4.xlarge.search	100MiB
m4.2xlarge.search	100MiB
m4.4xlarge.search	100MiB
m4.10xlarge.search	100MiB
m5.large.search	10MiB

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
m5.xlarge.search	100MiB
m5.2xlarge.search	100MiB
m5.4xlarge.search	100MiB
m5.12xlarge.search	100MiB
m6g.large.search	10MiB
m6g.xlarge.search	100MiB
m6g.2xlarge.search	100MiB
m6g.4xlarge.search	100MiB
m6g.8xlarge.search	100MiB
m6g.12xlarge.search	100MiB
c4.large.search	10MiB
c4.xlarge.search	100MiB
c4.2xlarge.search	100MiB
c4.4xlarge.search	100MiB
c4.8xlarge.search	100MiB
c5.large.search	10MiB
c5.xlarge.search	100MiB
c5.2xlarge.search	100MiB
c5.4xlarge.search	100MiB
c5.9xlarge.search	100MiB

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
c5.18xlarge.search	100MiB
c6g.large.search	10MiB
c6g.xlarge.search	100MiB
c6g.2xlarge.search	100MiB
c6g.4xlarge.search	100MiB
c6g.8xlarge.search	100MiB
c6g.12xlarge.search	100MiB
r3.large.search	10MiB
r3.xlarge.search	100MiB
r3.2xlarge.search	100MiB
r3.4xlarge.search	100MiB
r3.8xlarge.search	100MiB
r4.large.search	100MiB
r4.xlarge.search	100MiB
r4.2xlarge.search	100MiB
r4.4xlarge.search	100MiB
r4.8xlarge.search	100MiB
r4.16xlarge.search	100MiB
r5.large.search	100MiB
r5.xlarge.search	100MiB

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
r5.2xlarge.search	100MiB
r5.4xlarge.search	100MiB
r5.12xlarge.search	100MiB
r6g.large.search	100MiB
r6g.xlarge.search	100MiB
r6g.2xlarge.search	100MiB
r6g.4xlarge.search	100MiB
r6g.8xlarge.search	100MiB
r6g.12xlarge.search	100MiB
r6gd.large.search	100MiB
r6gd.xlarge.search	100MiB
r6gd.2xlarge.search	100MiB
r6gd.4xlarge.search	100MiB
r6gd.8xlarge.search	100MiB
r6gd.12xlarge.search	100MiB
r6gd.16xlarge.search	100MiB
i2.xlarge.search	100MiB
i2.2xlarge.search	100MiB
i3.large.search	100MiB
i3.xlarge.search	100MiB

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
i3.2xlarge.search	100MiB
i3.4xlarge.search	100MiB
i3.8xlarge.search	100MiB
i3.16xlarge.search	100MiB
또는1.medium.search	10MiB
또는1.large.search	100MiB
또는1.xlarge.search	100MiB
또는1.2xlarge.search	100MiB
또는1.4xlarge.search	100MiB
또는1.8xlarge.search	100MiB
또는1.12xlarge.search	100MiB
또는1.16xlarge.search	100MiB
또는2.medium.search	100MiB
또는2.large.search	100MiB
또는2.xlarge.search	100MiB
또는2.2xlarge.search	100MiB
또는2.4xlarge.search	100MiB
또는2.8xlarge.search	100MiB
또는2.12xlarge.search	100MiB
또는2.16xlarge.search	100MiB

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
om2.large.search	10MiB
om2.xlarge.search	100MiB
om2.2xlarge.search	100MiB
om2.4xlarge.search	100MiB
om2.8xlarge.search	100MiB
om2.12xlarge.search	100MiB
om2.16xlarge.search	100MiB
im4gn.large.search	100MiB
im4gn.xlarge.search	100MiB
im4gn.2xlarge.search	100MiB
im4gn.4xlarge.search	100MiB
im4gn.8xlarge.search	100MiB
im4gn.16xlarge.search	100MiB
i4i.large.search	100MiB
i4i.xlarge.search	100MiB
i4i.2xlarge.search	100MiB
i4i.4xlarge.search	100MiB
i4i.8xlarge.search	100MiB
i4i.12xlarge.search	100MiB
i4i.16xlarge.search	100MiB

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
i4i.24xlarge.search	100MiB
i4i.32xlarge.search	100MiB
i4g.large.search	100MiB
i4g.xlarge.search	100MiB
i4g.2xlarge.search	100MiB
i4g.4xlarge.search	100MiB
i4g.8xlarge.search	100MiB
i4g.16xlarge.search	100MiB
c7i.large.search	100MiB
c7i.xlarge.search	100MiB
c7i.2xlarge.search	100MiB
c7i.4xlarge.search	100MiB
c7i.8xlarge.search	100MiB
c7i.12xlarge.search	100MiB
c7i.16xlarge.search	100MiB
m7i.large.search	100MiB
m7i.xlarge.search	100MiB
m7i.2xlarge.search	100MiB
m7i.4xlarge.search	100MiB
m7i.8xlarge.search	100MiB

인스턴스 유형	HTTP 요청 페이로드의 최대 크기
m7i.12xlarge.search	100MiB
m7i.16xlarge.search	100MiB
r7i.large.search	100MiB
r7i.xlarge.search	100MiB
r7i.2xlarge.search	100MiB
r7i.4xlarge.search	100MiB
r7i.8xlarge.search	100MiB
r7i.12xlarge.search	100MiB
r7i.16xlarge.search	100MiB

샤드 크기 할당량

다음 섹션에서는 다양한 인스턴스 패밀리의 최대 샤드 크기를 살펴보겠습니다.

인스턴스 유형	Multi-AZ without Standby	Multi-AZ with Standby
R5, C5, M5	N/A	65GiB
I3	N/A	65GiB
R6g, C6g, M6g, R6gd	N/A	65GiB
OR1, OR2, OM2	100GiB	65GiB
Im4gn	N/A	65GiB

할당량 증가를 요청하려면 AWS Support에 문의하세요.

샤드 수 할당량

다음 섹션에서는 OpenSearch 버전의 최대 샤드 수를 나열합니다.

엔진 버전	Limit	Notes
Elasticsearch 1.5~6.x	기본 제한 없음	
Elasticsearch 7.x	1000	기본 제한은 클러스터를 통해 변경할 수 있습니다. max_shards_per_node 설정.
OpenSearch 1.x에서 2.15로	1000	기본 제한은 클러스터를 통해 변경할 수 있습니다. max_shards_per_node 설정.
OpenSearch 2.17 이상	힙 16GB당 1,000~최대 4,000	기본 제한은 변경할 수 없습니다.

Java 프로세스 할당량

OpenSearch Service에서 Java 프로세스는 힙 크기 32GiB로 제한됩니다. 고급 사용자는 필드 데이터에 사용할 힙 비율을 지정할 수 있습니다. 자세한 정보는 the section called “고급 클러스터 설정” 및 the section called “JVM OutOfMemoryError” 섹션을 참조하세요.

도메인 정책 할당량

OpenSearch Service는 도메인에 대한 액세스 정책을 100KiB로 제한합니다.

Amazon OpenSearch Service의 예약 인스턴스

Amazon OpenSearch Service의 예약 인스턴스(RI)는 표준 온디맨드 인스턴스에 비해 요금이 대폭 할인됩니다. 인스턴스 자체는 동일합니다. RI는 계정에서 온디맨드 인스턴스를 사용할 때 적용되는 결제 할인입니다. 사용량이 예측 가능하며 수명이 긴 애플리케이션의 경우, RI를 이용하면 시간이 지날수록 상당한 액수를 절감할 수 있습니다.

OpenSearch Service RI는 1년이나 3년 동안 이용해야 하며, 할인율이 달라지는 3가지 결제 방법을 제공합니다.

- 선결제 없음 – 선결제를 하지 않습니다. 사용 기간 내내 시간당 요금을 할인받습니다.
- 부분 선결제 – 일부 비용을 선결제하고, 사용 기간 내내 시간당 요금을 할인받습니다.
- 전체 선결제 – 모든 비용을 선결제합니다. 해당 기간 중 시간당 요금을 지불하지 않습니다.

일반적으로 선결제 금액이 많을수록 할인율이 증가합니다. 예약 인스턴스는 취소할 수 없습니다. 예약 인스턴스를 예약할 때 전체 기간에 대한 결제를 약정하고 선결제 금액은 환불되지 않습니다.

RI는 유연하지 않으며 사용자가 예약하는 정확한 인스턴스 유형에만 적용됩니다. 예를 들어, 8개의 `c5.2xlarge.search` 인스턴스에 대한 예약은 16개의 `c5.xlarge.search` 인스턴스 또는 4개의 `c5.4xlarge.search` 인스턴스에 적용되지 않습니다. 그러나에 있는 조직의 일부인 연결된 계정은 인스턴스 유형, 리전, 패밀리 및 크기가 일치하는 한 RI의 소유 계정에서 미사용 할인 애플리케이션을 활용할 AWS Organizations 수 있습니다. 자세한 내용은 Amazon OpenSearch Service 요금 및 FAQ를 참조하세요.

예약 인스턴스 구입(콘솔)

콘솔에서 기존 예약 인스턴스를 확인하고 새 예약 인스턴스를 구입할 수 있습니다.

예약 인스턴스 구입 방법

1. https://aws.amazon.com으로 이동하여 콘솔에 로그인(Sign In to the Console)을 선택합니다.
2. 분석(Analytics)에서 Amazon OpenSearch Service를 선택합니다.
3. 탐색 창에서 [예약 인스턴스 임대(Reserved Instance Leases)]를 선택하세요.

   이 페이지에서는 기존 예약을 확인할 수 있습니다. 예약이 여러 개라면, 필터를 적용해 특정 예약을 쉽게 찾아 확인할 수 있습니다.

   > **Tip**
   > 예약 인스턴스 임대 링크가 보이지 않으면 AWS 리전에서 도메인을 생성하세요.

4. [예약 인스턴스 주문(Order Reserved Instance)]을 선택합니다.
5. 고유한 서술식 이름을 입력합니다.
6. 인스턴스 유형과 인스턴스 수를 선택합니다. 자세한 지침은 the section called “도메인 크기 조정” 섹션을 참조하세요.
7. 사용 기간과 결제 옵션을 선택합니다. 결제 세부 정보를 자세히 검토합니다.

8. 다음을 선택합니다.
9. 구입 요약을 자세히 검토합니다. 구입한 예약 인스턴스는 환불할 수 없습니다.
10. 주문(Order)을 선택합니다.

예약 인스턴스 구매(AWS CLI)

AWS CLI 에는 상품 보기, 예약 구매, 예약 보기를 위한 명령이 있습니다. 다음 명령 및 샘플 응답은 지정된에 대한 오퍼링을 보여줍니다 AWS 리전.

```
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"
    }
  ]
}
```

각 반환 값에 대한 설명은 다음 표를 참조하세요.

필드	설명
FixedPrice	예약의 선결제 금액.
ReservedInstanceOfferingId	상품 ID입니다. 상품을 예약하고 싶다면 이 값을 기록해 두세요.
RecurringCharges	예약의 시간당 요금.

필드	설명
UsagePrice	레거시 필드. OpenSearch Services의 경우 이 값은 항상 0입니다.
PaymentOption	선결제 없음, 부분 선결제 또는 전체 선결제.
Duration	사용 기간(초): • 31,536,000초는 1년입니다. • 94,608,000초는 3년입니다.
InstanceType	예약의 인스턴스 유형. 각 인스턴스 유형에 할당되는 하드웨어 리소스 정보는 Amazon OpenSearch Service 요금을 참조하세요.
CurrencyCode	FixedPrice 와 RecurringChargeAmount 의 통화.

다음은 예약 구입 예제입니다.

```
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"
}
```

마지막으로 다음 예시를 활용해 해당 리전의 예약을 리스팅할 수 있습니다.

```
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`은 Unix epoch 시간으로, 1970년 1월 1일 자정 UTC 이후 경과 시간(초)을 의미합니다. 예를 들어 epoch 시간 1522872571은 UTC로 2018년 4월 4일 20:09:31입니다. 온라인 변환기를 이용할 수도 있습니다.

이전 예제에서 사용한 명령을 자세히 알아보려면 AWS CLI 명령 참조를 참조하세요.

예약 인스턴스(AWS SDKs) 구매

AWS SDKs(Android 및 iOS SDKs 제외)는 다음을 포함하여 Amazon OpenSearch Service API 참조에 정의된 모든 작업을 지원합니다.

- `DescribeReservedInstanceOfferings`
- `PurchaseReservedInstanceOffering`
- `DescribeReservedInstances`

이 샘플 스크립트는의 OpenSearchService 하위 수준 Python 클라이언트를 사용하여 예약 인스턴스 AWS SDK for Python (Boto3) 를 구매합니다. `instance_type`의 값을 제공해야 합니다.

```
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)
```

AWS SDKs 설치 및 사용에 대한 자세한 내용은 AWS 소프트웨어 개발 키트를 참조하세요.

비용 검사

Cost Explorer는 지난 13개월의 지출 데이터를 확인할 수 있는 무료 도구입니다. 이 데이터를 분석하면 지출 추세를 확인하고 RI가 사용 사례에 적합한지 확인할 수 있습니다. 이미 RI가 있으면 구매 옵션(Purchase Option)별로 그룹화하고 분할 상환 요금을 표시하여 온디맨드 인스턴스에 대한 지출과 해당 지출을 비교할 수 있습니다. 또한 예약을 최대한 활용하도록 사용 예산을 설정할 수 있습니다. 자세한 내용은 AWS Billing 사용 설명서의 Cost Explorer를 사용한 비용 분석을 참조하세요.

Amazon OpenSearch Service에서 지원되는 기타 리소스

이 주제에서는 Amazon OpenSearch Service에서 지원하는 추가 리소스에 대해 설명합니다.

bootstrap.memory_lock

OpenSearch Service는 `opensearch.yml`에서 `bootstrap.memory_lock`을 활성화하여 JVM 메모리를 잠그고 운영 체제가 디스크로 스와핑하지 못하도록 방지합니다. 이는 다음을 제외하고 지원되는 모든 인스턴스 유형에 적용됩니다.

- `t2.micro.search`
- `t2.small.search`
- `t2.medium.search`
- `t3.small.search`
- `t3.medium.search`

스크립팅 모듈

OpenSearch Service는 Elasticsearch 5.x 이상 도메인에 대한 스크립팅을 지원합니다. 1.5 또는 2.3에 대한 스크립팅은 지원되지 않습니다.

지원되는 스크립팅 옵션은 다음과 같습니다.

- Painless
- Lucene Expressions
- Mustache

Elasticsearch 5.5 이상 도메인과 모든 OpenSearch 도메인의 경우 OpenSearch Service는 `_scripts` 엔드포인트를 사용하는 저장된 스크립트를 지원합니다. Elasticsearch 5.3 및 5.1 도메인에서는 인라인 스크립트만 지원합니다.

TLS 전송

OpenSearch Service는 포트 80을 통한 HTTP와 포트 443을 통한 HTTPS는 지원하지만 TLS 전송은 지원하지 않습니다.

Amazon OpenSearch Service 자습서

이 장에는 서비스로 마이그레이션하고 간단한 검색 애플리케이션을 구축하고 OpenSearch Dashboards에서 시각화를 만드는 방법을 비롯하여 Amazon OpenSearch Service 작업을 위한 몇 가지 시작 후 완료 자습서가 포함되어 있습니다.

주제

- 자습서: Amazon OpenSearch Service에서 문서 생성 및 검색
- 튜토리얼: Amazon OpenSearch Service로 마이그레이션
- 자습서: Amazon OpenSearch Service를 사용하여 검색 애플리케이션 생성
- 자습서: OpenSearch Service 및 OpenSearch Dashboards를 사용하여 고객 지원 통화 시각화

자습서: Amazon OpenSearch Service에서 문서 생성 및 검색

이 자습서에서는 Amazon OpenSearch Service에서 문서를 생성하고 검색하는 방법을 알아봅니다. JSON 문서 형식으로 인덱스에 데이터를 추가합니다. OpenSearch Service는 사용자가 추가하는 첫 번째 문서 주위에 인덱스를 생성합니다.

이 자습서에서는 문서 생성을 위한 HTTP 요청, 문서 ID 자동 생성, 문서에 대한 기본 및 고급 검색 수행 방법을 설명합니다.

> **Note**
>
> 이 자습서에서는 개방 액세스가 가능한 도메인을 사용합니다. 최고 수준의 보안을 위해 도메인을 Virtual Private Cloud(VPC) 내부에 두는 것이 좋습니다.

사전 조건

이 자습서의 사전 요구 사항은 다음과 같습니다.

- 가 있어야 합니다 AWS 계정.
- 활성 OpenSearch Service 도메인이 있어야 합니다.

인덱스에 문서 추가

인덱스에 문서를 추가하려면 Postman, cURL 또는 OpenSearch Dashboards 콘솔과 같은 모든 HTTP 도구를 사용할 수 있습니다. 이 예제에서는 OpenSearch Dashboards에서 개발자 콘솔을 사용하고 있다고 가정합니다. 다른 도구를 사용하는 경우 필요에 따라 전체 URL과 자격 증명을 제공하여 적절히 조정합니다.

인덱스에 문서 추가

1. 도메인에 대한 OpenSearch Dashboards URL으로 이동합니다. OpenSearch Service 콘솔의 도메인 대시보드에서 URL을 찾을 수 있습니다. URL은 다음 형식을 따릅니다.

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

2. 기본 사용자 이름과 암호를 사용하여 로그인합니다.
3. 왼쪽 탐색 패널을 열고 Dev Tools(개발 도구)를 선택합니다.
4. 새 리소스를 생성하기 위한 HTTP 동사는 새 문서와 인덱스를 생성하는 데 사용하는 PUT입니다. 콘솔에서 다음 명령을 입력합니다.

   ```
   PUT fruit/_doc/1
   {
     "name":"strawberry",
     "color":"red"
   }
   ```

   PUT 요청은 이름이 fruit인 인덱스를 생성하고 ID가 1인 단일 문서를 인덱스에 추가합니다. 다음과 같은 응답이 생성됩니다.

   ```
   {
     "_index" : "fruit",
     "_type" : "_doc",
     "_id" : "1",
     "_version" : 1,
     "result" : "created",
     "_shards" : {
       "total" : 2,
       "successful" : 2,
       "failed" : 0
     },
     "_seq_no" : 0,
   ```

```
  "_primary_term" : 1
}
```

자동으로 생성되는 ID 만들기

OpenSearch Service는 문서에 대한 ID를 자동으로 생성할 수 있습니다. ID를 생성하는 명령은 PUT 요청 대신 POST 요청을 사용하며 문서 ID가 필요하지 않습니다(이전 요청과 비교).

개발자 콘솔에서 다음 요청을 입력합니다.

```
POST veggies/_doc
{
  "name":"beet",
  "color":"red",
  "classification":"root"
}
```

이 요청은 veggies라는 인덱스를 생성하고 인덱스에 문서를 추가합니다. 다음과 같은 응답이 생성됩니다.

```
{
  "_index" : "veggies",
  "_type" : "_doc",
  "_id" : "3WgyS4IB5DLqbRIvLxtF",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

응답의 추가 `_id` 필드는 ID가 자동으로 생성되었음을 나타냅니다.

> **Note**
>
> URL에서 _doc 다음에 아무 것도 제공하지 않습니다. 대개 이 자리에 ID가 들어갑니다. 생성된 ID로 문서를 만들고 있기 때문에 아직 ID를 제공하지 않습니다. 업데이트용으로 예약되어 있습니다.

POST 명령으로 문서 업데이트

문서를 업데이트하려면 ID 번호와 함께 HTTP POST 명령을 사용합니다.

먼저 ID가 42인 문서를 생성합니다.

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow"
}
```

그런 다음 해당 ID를 사용하여 문서를 업데이트합니다.

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow",
  "classification":"berries"
}
```

이 명령은 새 필드 `classification`으로 문서를 업데이트합니다. 다음과 같은 응답이 생성됩니다.

```
{
  "_index" : "fruits",
  "_type" : "_doc",
  "_id" : "42",
  "_version" : 2,
  "result" : "updated",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
```

```
  },
  "_seq_no" : 1,
  "_primary_term" : 1
}
```

> **Note**
>
> 존재하지 않는 문서를 업데이트하려고 하면 OpenSearch Service에서 문서를 생성합니다.

대량 작업 수행

POST _bulk API 작업을 사용하여 하나의 요청에서 하나 이상의 인덱스에 대해 여러 작업을 수행할 수 있습니다. 대량 작업 명령의 형식은 다음과 같습니다.

```
POST /_bulk
<action_meta>\n
<action_data>\n
<action_meta>\n
<action_data>\n
```

각 작업에는 두 줄의 JSON이 필요합니다. 먼저 작업 설명 또는 메타데이터를 제공합니다. 다음 줄에서 데이터를 제공합니다. 각 부분은 줄 바꿈(\n)으로 구분됩니다. 삽입에 대한 작업 설명은 다음과 같습니다.

```
{ "create" : { "_index" : "veggies", "_type" : "_doc", "_id" : "7" } }
```

데이터가 포함된 다음 줄은 다음과 같습니다.

```
{ "name":"kale", "color":"green", "classification":"leafy-green" }
```

종합하면 메타데이터와 데이터는 대량 작업의 단일 작업을 나타냅니다. 다음과 같이 하나의 요청으로 많은 작업을 수행할 수 있습니다.

```
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" } }
```

마지막 작업은 delete입니다. delete 작업 이후의 데이터가 없습니다.

문서 검색

이제 데이터가 클러스터에 있으므로 데이터를 검색할 수 있습니다. 예를 들어, 모든 뿌리 채소를 검색하거나, 잎이 많은 채소 수를 모두 구하거나, 시간당 기록된 오류 수를 찾을 수 있습니다.

기본 검색

기본 검색은 다음과 같습니다.

```
GET veggies/_search?q=name:l*
```

요청은 lettuce 문서를 포함하는 JSON 응답을 생성합니다.

고급 검색

요청 본문에 쿼리 옵션을 JSON으로 제공하여 고급 검색을 수행할 수 있습니다.

```
GET veggies/_search
{
  "query": {
    "term": {
      "name": "lettuce"
    }
  }
}
```

이 예제에서는 lettuce 문서가 포함된 JSON 응답도 생성합니다.

정렬

정렬을 사용하여 이러한 유형의 쿼리를 더 많이 수행할 수 있습니다. 먼저 자동 필드 매핑에서 기본적으로 정렬할 수 없는 유형을 선택했기 때문에인덱스를 다시 생성해야 합니다. 다음 요청을 전송하여 인덱스를 삭제했다가 다시 생성합니다.

```
DELETE /veggies

PUT /veggies
{
   "mappings":{
      "properties":{
         "name":{
            "type":"keyword"
         },
         "color":{
            "type":"keyword"
         },
         "classification":{
            "type":"keyword"
         }
      }
   }
}
```

그런 다음 인덱스를 데이터로 다시 채웁니다.

```
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" }
```

이제 정렬과 함께 검색할 수 있습니다. 다음 요청은 분류별 오름차순 정렬을 추가합니다.

```
GET veggies/_search
{
  "query" : {
```

```
    "term": { "color": "green" }
  },
  "sort" : [
      "classification"
  ]
}
```

관련 리소스

자세한 정보는 다음 자료를 참조하세요.

- 시작
- 데이터 인덱싱
- 데이터 검색

튜토리얼: Amazon OpenSearch Service로 마이그레이션

인덱스 스냅샷은 자체 관리형 OpenSearch 또는 레거시 Elasticsearch 클러스터에서 Amazon OpenSearch Service로 마이그레이션하는 데 널리 사용되는 방법입니다. 대체로 프로세스는 다음 단계로 구성됩니다.

1. 기존 클러스터의 스냅샷을 만들고 스냅샷을 Amazon S3 버킷에 업로드합니다.
2. OpenSearch Service 도메인 생성
3. 버킷에 액세스할 수 있는 권한을 OpenSearch Service에 부여하고 자신에게 스냅샷으로 작업할 수 있는 권한이 있는지 확인합니다.
4. OpenSearch Service 도메인에서 스냅샷을 복원합니다.

이 연습에서는 자세한 단계와 대체 옵션(해당되는 경우)을 다룹니다.

스냅샷 생성 및 업로드

리포지토리-s3 플러그인을 사용하여 S3에 직접 스냅샷을 생성할 수 있지만 모든 노드에 플러그인을 설치하고, 조정하고`opensearch.yml`(또는 Elasticsearch 클러스터를 사용하는 `elasticsearch.yml` 경우), 각 노드를 다시 시작하고, AWS 자격 증명을 추가하고, 마지막으로 스냅샷을 생성해야 합니다. 플러그인은 지속해서 사용하거나 더 큰 클러스터를 마이그레이션하기 위한 좋은 옵션입니다.

작은 클러스터의 경우 일회성 접근 방식은 공유 파일 시스템 스냅샷을 생성한 다음를 사용하여 S3 AWS CLI 에 업로드하는 것입니다. 이미 스냅샷이 있는 경우 4단계로 건너뜁니다.

스냅샷을 생성하여 Amazon S3에 업로드

1. 모든 노드에서 opensearch.yml(또는 Elasticsearch.yml)에 path.repo 설정을 추가한 다음 각 노드를 다시 시작합니다.

```
path.repo: ["/my/shared/directory/snapshots"]
```

2. 스냅샷을 찍기 전에 필요한 스냅샷 리포지토리를 등록합니다. 리포지토리는 공유 파일 시스템, Amazon S3, Hadoop 분산 파일 시스템(HDFS) 등, 단순한 스토리지 위치입니다. 이 경우 공유 파일 시스템("fs")을 사용합니다.

```
PUT _snapshot/my-snapshot-repo-name
{
  "type": "fs",
  "settings": {
    "location": "/my/shared/directory/snapshots"
  }
}
```

3. 스냅샷 생성:

```
PUT _snapshot/my-snapshot-repo-name/my-snapshot-name
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

4. AWS CLI를 설치하고 aws configure을 실행하여 자격 증명을 추가합니다.
5. 스냅샷 디렉터리로 이동합니다. 다음 명령을 실행하여 새 S3 버킷을 생성하고 스냅샷 디렉터리의 콘텐츠를 해당 버킷에 업로드합니다.

```
aws s3 mb s3://amzn-s3-demo-bucket --region us-west-2
aws s3 sync . s3://amzn-s3-demo-bucket --sse AES256
```

스냅샷의 크기와 인터넷 연결 속도에 따라 이 작업을 실행할 때 시간이 걸릴 수 있습니다.

도메인 생성

콘솔이 도메인을 생성하는 가장 쉬운 방법이지만 이 경우 이미 터미널이 열려 있고가 AWS CLI 설치되어 있습니다. 다음 명령을 수정하여 필요에 맞게 도메인을 만듭니다.

```
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":["*"]},"Action":
["es:ESHttp*"],"Resource":"arn:aws:es:us-west-2:123456789012:domain/migration-domain/
*"}]}' \
  --region us-west-2
```

마찬가지로 이 명령은 각각 100GiB의 스토리지가 있는 두 개의 데이터 노드를 갖춘 인터넷 액세스 가능 도메인을 만듭니다. 또한 HTTP 기본 인증 및 모든 암호화 설정으로 세분화된 액세스 제어가 가능합니다. VPC와 같은 고급 보안 구성이 필요한 경우, OpenSearch Service 콘솔을 사용합니다.

명령을 실행하기 전에 도메인 이름, 마스터 사용자 자격 증명 및 계정 번호를 변경합니다. 스냅샷과 호환되는 S3 버킷 및 OpenSearch/Elasticsearch 버전에 사용한 AWS 리전 것과 동일하게 지정합니다.

> **Important**
>
> 스냅샷은 하나의 주 버전에서만 호환됩니다. 예를 들어, Elasticsearch 7.x 클러스터에서는 OpenSearch 1.x 클러스터의 스냅샷을 복원할 수 없습니다. OpenSearch 1.x 또는 2.x 클러스터만 가능합니다. 마이너 버전도 중요합니다. 5.3.2 OpenSearch Service 도메인의 자체 관리형 5.3.3 클러스터에서는 스냅샷을 복원할 수 없습니다. 스냅샷에서 지원하는 OpenSearch 또는 Elasticsearch의 최신 버전을 선택하는 것이 좋습니다. 호환 가능한 버전 테이블은 the section called “스냅샷을 사용하여 데이터 마이그레이션” 섹션을 참조하세요.

S3 버킷에 권한 부여

AWS Identity and Access Management (IAM) 콘솔에서 다음과 같은 권한과 신뢰 관계를 가진 역할을 생성합니다. 역할을 생성할 때 AWS 서비스로 S3를 선택합니다. 쉽게 찾을 수 있도록 `OpenSearchSnapshotRole` 역할 이름을 지정합니다.

권한

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/*"
      ]
    }
  ]
}
```

신뢰 관계

JSON

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

```
    }
  ]
}
```

그런 다음 개인 IAM 역할에 `OpenSearchSnapshotRole`을 수임할 수 있는 권한을 부여합니다. 다음 정책을 만들어 자격 증명에 연결합니다.

권한

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
    }
  ]
}
```

세분화된 액세스 제어를 사용하는 경우 OpenSearch Dashboards에서 스냅샷 역할을 매핑할 수 있습니다.

세분화된 액세스 제어를 활성화한 경우 다른 모든 용도로 HTTP 기본 인증을 사용하더라도 `manage_snapshots` 역할을 IAM 역할에 할당하여 스냅샷으로 작업할 수 있도록 해야 합니다.

스냅샷으로 작업할 수 있는 자격 증명 권한을 부여하려면

1. OpenSearch Service 도메인을 생성할 때 지정한 마스터 사용자 자격 증명을 사용하여 Dashboards에 로그인합니다. Dashboards URL은 OpenSearch Service 콘솔에서 찾을 수 있습니다. `https://domain-endpoint/_dashboards/` 형식을 사용합니다.
2. 주 메뉴에서 보안(Security), 역할(Roles)을 선택하고 manage_snapshots 역할을 선택합니다.
3. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다.
4. 그런 다음 해당 필드에 개인 IAM 역할의 도메인 ARN을 추가합니다. ARN은 다음 형식 중 하나여야 합니다.

```
arn:aws:iam::123456789123:user/user-name
```

```
arn:aws:iam::123456789123:role/role-name
```

5. Map(맵)을 선택하고 Mapped users(매핑된 사용자)에 역할이 나타나는지 확인합니다.

스냅샷을 복원합니다.

이 시점에서 OpenSearch Service 도메인에 액세스하는 두 가지 방법, 즉 마스터 사용자 자격 증명을 사용한 HTTP 기본 인증 또는 IAM 자격 증명을 사용한 AWS 인증이 있습니다. 스냅샷에서는 마스터 사용자에 대한 개념이 없는 Amazon S3를 사용하므로, IAM 자격 증명을 사용하여 OpenSearch Service 도메인에 스냅샷 리포지토리를 등록해야 합니다.

대부분의 프로그래밍 언어에는 서명 요청에 도움이 되는 라이브러리가 있지만, 더 간단한 방법은 Postman과 같은 도구를 사용하여 IAM 보안 인증 정보를 권한 부여 섹션에 넣는 것입니다.

스냅샷을 복원하려면

1. 요청에 서명하는 방법과 관계없이 첫 번째 단계는 리포지토리를 등록하는 것입니다.

```
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. 그런 다음 리포지토리에 있는 스냅샷을 나열하고 복원할 스냅샷을 찾습니다. 이 시점에서 Postman을 계속 사용하거나 curl과 같은 도구로 전환할 수 있습니다.

   간편

```
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. 스냅샷을 복원합니다.

   간편

```
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. 마지막으로 인덱스가 예상대로 복원되었는지 확인합니다.

   간편

```
GET _cat/indices?v
```

   curl

```
curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/_cat/
indices?v
```

이 시점에서 마이그레이션이 완료됩니다. 새 OpenSearch 엔드포인트를 사용하도록 클라이언트를 구성하거나, 워크로드에 맞게 도메인 크기를 조정하거나, 인덱스의 샤드 수를 확인하거나, IAM 마스터 사용자로 전환하거나, OpenSearch Dashboards에서 시각화를 구축할 수 있습니다.

자습서: Amazon OpenSearch Service를 사용하여 검색 애플리케이션 생성

Amazon OpenSearch Service로 검색 애플리케이션을 생성하는 일반적인 방법은 웹 양식을 사용해 사용자 쿼리를 서버로 전송하는 것입니다. 그런 다음 서버가 OpenSearch API를 직접 호출하여 OpenSearch Service로 요청을 전송할 수 있도록 권한을 부여하면 됩니다. 하지만 서버에 의존하지 않는 클라이언트 측 코드를 작성하려면 보안 및 성능 위험을 상쇄해야 합니다. OpenSearch API에 대한 무서명 공개 액세스를 허용하는 것은 권장하지 않습니다. 사용자가 보호되지 않은 엔드포인트에 액세스하거나 너무 광범위한 쿼리(또는 너무 많은 쿼리)로 클러스터 성능에 악영향을 미칠 수 있습니다.

이 장에서는 Amazon API Gateway를 사용하여 사용자를 OpenSearch APIs의 하위 집합으로 제한하고 API Gateway에서 OpenSearch Service AWS Lambda 로의 요청에 서명하는 솔루션을 제공합니다.

> **Note**
>
> 표준 API Gateway 및 Lambda 요금 정책이 적용되지만, 이 자습서에서는 사용량이 제한적이므로 비용은 무시할만한 수준입니다.

사전 조건

이 자습서의 사전 조건은 OpenSearch Service 도메인입니다. 아직 도메인이 없는 경우 OpenSearch Service 도메인 생성 단계에 따라 도메인을 생성합니다.

1단계: 샘플 데이터 인덱싱

sample-movies.zip을 다운로드하여 압축을 해제한 다음 _bulk API 작업을 사용하여 5,000개 문서를 `movies` 인덱스에 추가합니다.

```
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"}
...
```

위는 사용 가능한 데이터의 하위 세트를 포함하는 예제 명령입니다. `_bulk` 작업을 수행하려면 `sample-movies` 파일의 전체 내용을 복사하여 붙여 넣어야 합니다. 자세한 지침은 단원을 참조하십시오the section called “옵션 2: 여러 문서 업로드”.

또한 다음 curl 명령을 사용하여 동일한 결과를 얻을 수도 있습니다.

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

2단계: Lambda 함수 생성 및 배포

API Gateway에서 API를 생성하기 전에 요청을 전달하는 Lambda 함수를 만듭니다.

Lambda 함수 생성

이 솔루션에서는 API Gateway가 요청을 다음 Lambda 함수로 전달합니다. 그러면 이 함수가 OpenSearch Service를 쿼리하고 결과를 반환합니다. 이 샘플 함수는 외부 라이브러리를 사용하므로 배포 패키지를 생성하고 Lambda에 업로드해야 합니다.

배포 패키지를 만드는 방법

1. 명령 프롬프트를 열고 `my-opensearch-function` 프로젝트 디렉터리를 만듭니다. 예를 들어, macOS에서는 다음을 수행합니다.

   ```
   mkdir my-opensearch-function
   ```

2. `my-sourcecode-function` 프로젝트 디렉터리로 이동합니다.

   ```
   cd my-opensearch-function
   ```

3. 다음과 같은 샘플 Python 코드의 콘텐츠를 복사하고 이름이 opensearch-lambda.py인 새 파일에 저장합니다. 리전 및 호스트 엔드포인트를 파일에 추가합니다.

```
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. 외부 라이브러리를 새 package 디렉터리에 설치합니다.

```
pip3 install --target ./package boto3
pip3 install --target ./package requests
pip3 install --target ./package requests_aws4auth
```

5. 루트에서 설치된 라이브러리를 포함하는 배포 패키지를 만듭니다. 다음 명령을 실행하면 프로젝트 디렉터리에 my-deployment-package.zip 파일이 생성됩니다.

```
cd package
zip -r ../my-deployment-package.zip .
```

6. zip 파일의 루트에 opensearch-lambda.py 파일을 추가합니다.

```
cd ..
zip my-deployment-package.zip opensearch-lambda.py
```

Lambda 함수 및 배포 패키지를 만드는 방법에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 zip 파일 아카이브를 사용하여 Python Lambda 함수 배포 및 본 가이드의 the section called “Lambda 배포 패키지 생성”를 참조하세요.

Lambda 콘솔을 사용하여 함수를 만들려면

1. https://console.aws.amazon.com/lambda/home에서 Lambda 콘솔로 이동합니다. 왼쪽 탐색 창에서 함수를 선택합니다.
2. 함수 생성을 선택합니다.
3. 다음 필드를 구성합니다.
   - 함수 이름: opensearch-function
   - 런타임 – Python 3.9

- 아키텍처: x86_64

다른 모든 기본 옵션은 그대로 두고 함수 생성을 선택합니다.

4. 함수 요약 페이지의 코드 소스 섹션에서 드롭다운에서 업로드를 선택하고 .zip 파일을 선택합니다. 생성한 `my-deployment-package.zip` 파일을 찾아 저장을 선택합니다.
5. 핸들러는 이벤트를 처리하는 함수 코드의 메서드입니다. 런타임 설정에서 편집을 선택하고 Lambda 함수가 있는 배포 패키지의 파일 이름에 따라 핸들러 이름을 변경합니다. 파일 이름이 `opensearch-lambda.py`이므로 핸들러 이름을 *`opensearch-lambda`*`.lambda_handler`로 변경합니다. 자세한 내용은 Python의 Lambda 함수 핸들러를 참조하세요.

3단계: API Gateway에서 API 생성

API Gateway를 사용하면 보다 제한된 API를 생성하고 OpenSearch `_search` API와의 상호 작용을 간소화할 수 있습니다. API Gateway를 사용하면 Amazon Cognito 인증 및 요청 조절 같은 보안 기능을 활성화할 수도 있습니다. API를 생성하고 배포하려면 다음 단계를 수행합니다.

API 생성 및 구성

API Gateway 콘솔을 사용하여 API를 생성하려면

1. https://console.aws.amazon.com/apigateway/home에서 API Gateway 콘솔로 이동합니다. 왼쪽 탐색 창에서 API를 선택합니다.
2. REST API(비공개 아님)를 찾고 빌드(Build)를 선택합니다.
3. 다음 페이지에서 새 API 생성 섹션을 찾아 새 API가 선택되어 있는지 확인합니다.
4. 다음 필드를 구성합니다.
   - API 이름: OpenSearch-api
   - 설명: Amazon OpenSearch Service 도메인을 검색하기 위한 퍼블릭 API
   - 엔드포인트 유형: 리전별
5. API 생성을 선택합니다.
6. 작업(Actions) 및 메서드 생성(Create Method)을 선택합니다.
7. 드롭다운에서 GET을 선택하고 확인 표시를 클릭하여 확인합니다.
8. 다음 설정을 구성한 다음 저장(Save)을 선택합니다.

설정	값
통합 유형	Lambda 함수
Lambda 프록시 통합 사용	예
Lambda 리전	*us-west-1*
Lambda 함수	opensearch-lambda
기본 제한 시간 사용	예

메서드 요청 구성

메서드 요청(Method Request)을 선택하고 다음 설정을 구성합니다.

설정	값
권한 부여	NONE
요청 검사기	쿼리 문자열 파라미터 및 헤더 검사
필수 API 키	false

URL 쿼리 문자열 파라미터에서 쿼리 문자열 추가를 선택하고 다음 파라미터를 구성합니다.

설정	값
명칭	q
필수	예

API 배포 및 단계 구성

API Gateway 콘솔에서 배포를 생성하고 새 단계 또는 기존 단계에 연결하여 API를 배포할 수 있습니다.

1. 작업(Actions) 및 API 배포(Deploy API)를 선택합니다.
2. 배포 단계(Deployment stage)에서 새 단계(New Stage)를 클릭하고 단계 이름을 `opensearch-api-test`로 지정합니다.
3. 배포(Deploy)를 선택합니다.
4. 단계 편집기에서 다음 설정을 구성한 다음 변경 내용 저장(Save Changes)을 선택합니다.

설정	값
조절 활성화	예
Rate	1000
버스트	500

이러한 설정은 엔드포인트 루트에 대한 GET 요청(`https://some-id.execute-api.us-west-1.amazonaws.com/search-es-api-test`) 메서드 하나뿐인 API를 구성합니다. 이 요청에는 파라미터 하나(q), 즉 검색할 쿼리 문자열이 필요합니다. 메서드를 호출하면 요청이 Lambda로 전송되어 `opensearch-lambda` 함수가 실행됩니다. 자세한 내용은 Amazon API Gateway에서 API 생성 및 Amazon API Gateway에서 REST API 배포를 참조하세요.

4단계: (선택 사항) 도메인 액세스 정책 수정

OpenSearch Service 도메인에서 Lambda 함수가 `movies` 인덱스에 GET 요청을 수행할 수 있도록 허용해야 합니다. 도메인에 세분화된 액세스 제어가 활성화된 오픈 액세스 정책이 있는 경우 그대로 둘 수 있습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
```

```
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

또는 도메인 액세스 정책을 보다 세분화하도록 선택할 수 있습니다. 예를 들어 다음 최소 정책은 `opensearch-lambda-role`(Lambda를 통해 생성됨)에 `movies` 인덱스에 대한 읽기 액세스를 제공합니다. Lambda가 자동으로 생성하는 역할의 정확한 이름을 가져오려면 AWS Identity and Access Management (IAM) 콘솔로 이동하여 역할을 선택하고 "lambda"를 검색합니다.

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**
>
> 도메인에 대해 세분화된 액세스 제어를 활성화한 경우 OpenSearch 대시보드에서 역할을 사용자에게 매핑해야 합니다. 그러지 않으면 권한 오류가 표시됩니다.

Lambda 실행 역할 권한 구성

도메인 액세스 정책을 구성하는 것 외에도 Lambda 실행 역할에 OpenSearch Service 도메인에 액세스하는 데 필요한 IAM 권한이 있는지도 확인해야 합니다. Lambda 함수에는 관리형 도메인을 사용하는지 아니면 OpenSearch Service Serverless 컬렉션을 사용하는지에 따라 특정 권한이 필요합니다.

관리형 OpenSearch Service 도메인의 경우:

다음 IAM 정책을 Lambda 실행 역할에 연결하여 OpenSearch Service 도메인에 요청할 수 있도록 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPost"
      ],
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

OpenSearch Service Serverless 컬렉션의 경우:

OpenSearch Service Serverless를 사용하는 경우 Lambda 실행 역할에 다음 IAM 정책을 연결합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aoss:*",
      "Resource": "arn:aws:aoss:us-west-1:123456789012:collection/collection-id"
```

```
    }
  ]
}
```

이러한 정책을 Lambda 실행 역할에 연결하려면:

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔로 이동합니다.
2. 역할을 선택하고 Lambda 실행 역할(일반적으로 이름)을 검색합니다`opensearch-lambda-role-xxxxxxxx`.
3. 권한 추가를 선택한 다음 인라인 정책 생성을 선택합니다.
4. JSON 탭을 선택하고 위에서 적절한 정책을 붙여넣어 자리 표시자 값을 실제 리소스 ARNs.
5. 정책 검토를 선택하고 `OpenSearchAccess`와 같은 이름을 입력한 다음 정책 생성을 선택합니다.

> **Note**
>
> 이러한 IAM 권한이 없으면 도메인 액세스 정책이 요청을 허용하더라도 OpenSearch Service 도메인 쿼리를 시도할 때 Lambda 함수에 "액세스 거부됨" 오류가 발생합니다.

액세스 정책에 대한 자세한 내용은 the section called “액세스 정책 구성” 섹션을 참조하세요.

Lambda 역할 매핑(세분화된 액세스 제어를 사용하는 경우)

세분화된 액세스 제어를 사용하면 애플리케이션을 테스트하기 전에 추가 단계가 안내됩니다. 다른 모든 목적으로 HTTP 기본 인증을 사용하더라도 Lambda 역할을 사용자에게 매핑해야 합니다. 그러지 않으면 권한 오류가 표시됩니다.

1. 도메인에 대한 OpenSearch 대시보드 URL로 이동합니다.
2. 기본 메뉴에서 보안, 역할을 선택한 후 Lambda 역할을 매핑해야 할 역할인 `all_access`에 대한 링크를 선택합니다.
3. 매핑된 사용자(Mapped users), 매핑 관리(Manage mapping)를 차례로 선택합니다.
4. Backend roles(백엔드 역할)에서 Lambda 역할의 Amazon 리소스 이름(ARN)을 추가합니다. ARN은 `arn:aws:iam::123456789123:role/service-role/opensearch-lambda-role-1abcdefg` 형식을 취해야 합니다.
5. Map(맵)을 선택하고 Mapped users(매핑된 사용자)에 사용자 또는 역할이 나타나는지 확인합니다.

5단계: 웹 애플리케이션 테스트

웹 애플리케이션을 테스트하려면

1. sample-site.zip을 다운로드하고 압축을 해제하여 자주 사용하는 텍스트 편집기에서 `scripts/search.js`를 엽니다.
2. `apigatewayendpoint` 변수를 업데이트하여 API Gateway 엔드포인트를 가리키도록 하고 지정된 경로의 끝에 백슬래시를 추가합니다. 단계(Stages)를 선택하고 API의 이름을 선택하여 API Gateway에서 엔드포인트를 빠르게 찾을 수 있습니다. `apigatewayendpoint` 변수는 `https://some-id.execute-api.us-west-1.amazonaws.com/opensearch-api-test/`의 형식을 취해야 합니다.
3. `index.html`을 열고 thor, house 등 몇 가지 단어를 검색해 봅니다.

CORS 오류 문제 해결

Lambda 함수가 CORS를 지원하기 위해 응답에 콘텐츠를 포함하더라도 다음과 같은 오류가 계속 표시될 수 있습니다.

```
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.
```

이러한 상황이 발생하면 다음 작업을 시도합니다.

1. GET 리소스에서 CORS를 활성화합니다. 고급(Advanced)에서 Access-Control-Allow-Credentials를 `'true'`로 설정합니다.
2. API Gateway에서 API를 재배포합니다(작업(Actions), API 배포(Deploy API)).
3. Lambda 함수 트리거를 삭제하고 다시 추가합니다. 다시 추가하려면 트리거 추가를 선택하고 함수를 호출하는 HTTP 엔드포인트를 생성합니다. 트리거 구성은 다음과 같아야 합니다.

트리거	API	배포 단계	보안
API Gateway	OpenSearch-api	OpenSearch-api-test	열기

다음 단계

이 장은 개념을 설명하기 위한 출발점에 불과합니다. 다음과 같은 수정을 고려할 수 있습니다.

- OpenSearch Service 도메인에 사용자의 데이터를 추가합니다.
- 사용자의 API에 메서드를 추가합니다.
- Lambda 함수에서 검색 쿼리를 수정하거나 다른 필드를 부스트합니다.
- 결과 스타일을 다르게 지정하거나 `search.js`를 수정하여 사용자에게 다른 필드를 표시합니다.

자습서: OpenSearch Service 및 OpenSearch Dashboards를 사용하여 고객 지원 통화 시각화

이 장에서는 몇 차례의 고객 지원 문의 전화를 받은 기업에서 이를 분석하려는 상황에 대해 자세히 알아봅니다. 각 통화의 주제는 무엇입니까? 긍정적인 내용은 몇 통이었습니까? 부정적인 내용은 몇 통이었습니까? 관리자가 이러한 통화의 녹취록을 검색하거나 검토하려면 어떻게 해야 합니까?

수작업 워크플로우에서는 직원들이 녹음된 내용을 듣고, 각 통화의 주제를 기록하고, 고객 상담 내용이 긍정적이었는지를 판단합니다.

따라서 이러한 프로세스는 대단히 노동 집약적입니다. 평균 통화 시간이 10분이라고 가정하면 직원 한 명이 하루에 48건의 통화밖에 들을 수 없습니다. 인간의 편견이 작용하지 않는다면 이들은 매우 정확한 데이터를 생산해 내겠지만, 그 데이터의 양은 최소한에 불과하여 통화의 주제와 고객이 만족했는지 여부에 대한 부울 값 정도를 얻을 수 있을 것입니다. 전체 녹취록 등 그 이상의 결과물이 필요한 경우에는 막대한 시간이 소요됩니다.

Amazon S3, Amazon Transcribe, Amazon Comprehend 및 Amazon OpenSearch Service를 사용하면 비슷한 프로세스를 굉장히 적은 코드로 자동화하여 훨씬 더 많은 데이터를 얻을 수 있습니다. 예를 들면 전체 통화 녹취록, 녹취록의 키워드, 그리고 통화의 전반적인 "감정"(긍정적, 부정적, 중립적, 혼합)을 파악할 수 있습니다. 그런 다음 OpenSearch 및 OpenSearch Dashboards를 사용하여 데이터를 검색하고 시각화할 수 있습니다.

이 연습 단계를 그대로 사용해도 되지만, JSON 문서를 OpenSearch Service에 인덱싱하기 전에 문서를 보강하는 방법에 관한 아이디어를 얻는 것이 이 과정의 목표입니다.

추정 비용

일반적으로, 이 연습 단계를 수행하는 데 드는 비용은 2달러 미만입니다. 이 연습 단계에서는 다음 리소스를 사용합니다.

- 전송 및 저장량이 100MB 미만인 S3 버킷

  자세한 내용은 Amazon S3 요금을 참조하세요.
- 몇 시간 동안의 10GiB EBS 스토리지와 `t2.medium` 인스턴스 한 개가 있는 OpenSearch Service 도메인

  자세한 내용은 Amazon OpenSearch Service 요금을 참조하세요.
- Amazon Transcribe에 대한 호출 여러 개

  자세한 내용은 Amazon Transcribe 요금을 참조하세요.
- Amazon Comprehend에 대한 자연어 처리 호출 여러 개

  자세한 내용은 Amazon Comprehend 요금을 참조하세요.

주제

- 1단계: 사전 조건 구성
- 2단계: 샘플 코드 복사
- (선택 사항) 3단계: 샘플 데이터 인덱싱
- 4단계: 데이터 분석 및 시각화
- 5단계: 리소스 정리 및 다음 단계

1단계: 사전 조건 구성

계속하려면 먼저 다음 리소스를 확보해야 합니다.

전제 조건	설명
Amazon S3 버킷	자세한 내용은 Amazon Simple Storage Service 사용 설명서에서 버킷 생성을 참조하세요.
OpenSearch Service 도메인	데이터의 대상 주소입니다. 자세한 내용은 OpenSearch Service 도메인 생성을 참조하세요.

이러한 리소스가 아직 없는 경우 다음 AWS CLI 명령을 사용하여 만들 수 있습니다.

```
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**
>
> 이 명령은 us-west-2 리전을 사용하지만, Amazon Comprehend가 지원하는 아무 리전이나 사용할 수 있습니다. 자세한 내용은 AWS 일반 참조 섹션을 참조하십시오.

2단계: 샘플 코드 복사

1. 다음 Python 3 샘플 코드를 복사하여 call-center.py라는 새 파일에 붙여넣습니다.

```
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. 처음 여섯 개의 변수를 업데이트합니다.
3. 다음 명령을 사용하여 필요한 패키지를 설치합니다.

```
pip install boto3
pip install requests
pip install requests_aws4auth
```

4. MP3를 `call-center.py`와 동일한 디렉터리에 넣고 스크립트를 실행합니다. 샘플 출력은 다음과 같습니다.

```
$ 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`는 다음 몇 가지 작업을 수행합니다.

1. 이 스크립트는 S3 버킷에 오디오 파일(이 경우에는 MP3지만, Amazon Transcribe는 다른 형식도 지원함)을 업로드합니다.
2. 오디오 파일의 URL을 Amazon Transcribe로 보낸 다음 녹취 작업이 완료되기를 기다립니다.

   녹취 작업의 완료 시간은 오디오 파일의 길이에 따라 달라집니다. 몇 초가 아니라 몇 분이 걸립니다.

   > **Tip**
   >
   > 녹취 품질을 높이기 위해 Amazon Transcribe에 대한 사용자 지정 어휘를 구성할 수 있습니다.

3. 녹취 작업이 완료되면 스크립트가 녹취록을 추출하고, 5,000자로 정리한 다음 키워드 및 감정 분석을 위해 Amazon Comprehend로 보냅니다.
4. 마지막으로 이 스크립트는 전체 녹취록, 키워드, 감정 분석, 현재 타임스탬프 등을 JSON 문서에 추가하고 OpenSearch Service에서 이를 인덱싱합니다.

> **Tip**
>
> LibriVox의 퍼블릭 도메인 오디오북을 테스트에 이용할 수 있습니다.

(선택 사항) 3단계: 샘플 데이터 인덱싱

다수의 통화 레코딩을 바로 사용할 수 없는 경우 `call-center.py`에서 생성하는 것에 해당하는 샘플 문서를 sample-calls.zip으로 인덱스할 수 있습니다.

1. `bulk-helper.py`라는 이름의 파일을 만듭니다.

```
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. `host` 및 `region`의 처음 두 변수를 업데이트합니다.
3. 다음 명령을 사용하여 필요한 패키지를 설치합니다.

```
pip install opensearch-py
```

4. sample-calls.zip을 다운로드하여 압축을 풉니다.
5. `sample-calls.bulk`를 `bulk-helper.py`와 동일한 디렉터리에 넣고 도움말을 실행합니다. 샘플 출력은 다음과 같습니다.

```
$ 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
}
```

4단계: 데이터 분석 및 시각화

OpenSearch Service에 데이터가 있으므로 OpenSearch Dashboards를 사용하여 시각화할 수 있습니다.

1. `https://search-domain.region.es.amazonaws.com/_dashboards`로 이동합니다.
2. OpenSearch Dashboards를 사용하려면 먼저 인덱스 패턴이 있어야 합니다. Dashboards는 인덱스 패턴을 사용하여 분석 범위를 하나 이상의 인덱스로 좁혀 줍니다. `call-center.py`에서 생성된 `support-calls` 인덱스를 일치시키려면 스택 관리(Stack Management), 인덱스 패턴(Index Patterns)으로 이동하여 `support*`의 인덱스 패턴을 정의한 다음, 다음 단계(Next step)를 선택합니다.
3. 시간 필터 필드 이름(Time Filter field name)에서 타임스탬프(timestamp)를 선택합니다.
4. 이제 시각화를 생성할 수 있습니다. 시각화(Visualize)를 선택한 다음, 새 시각화를 추가합니다.
5. 파이 차트와 `support*` 인덱스 패턴을 선택합니다.
6. 시각화의 기본값은 기본이므로 조각 분할(Split Slices)을 선택하여 보다 흥미로운 시각화를 만듭니다.

   집계(Aggregation)에서 조건(Terms)을 선택합니다. 필드(Field)에서 sentiment.keyword를 선택합니다. 그런 다음 변경 사항 적용(Apply changes) 및 저장(Save)을 선택합니다.

7. 시각화(Visualize) 페이지로 돌아가서 다른 시각화를 추가합니다. 이번에는 가로 막대 차트를 선택합니다.

8. 계열 분할(Split Series)을 선택합니다.

   집계(Aggregation)에서 조건(Terms)을 선택합니다. 필드에서 keywords.keyword를 선택하고 크기(Size)를 20으로 변경합니다. 그런 다음 변경 사항 적용(Apply Changes) 및 저장(Save)을 선택합니다.

9. 시각화(Visualize) 페이지로 돌아가서 마지막 시각화인 세로 막대 차트를 추가합니다.
10. 계열 분할(Split Series)을 선택합니다. 집계(Aggregation)에서 날짜 히스토그램(Date Histogram)을 선택합니다. 필드(Field)에서 타임스탬프(timestamp)를 선택하고 간격(Interval)을 매일(Daily)로 변경합니다.
11. 지표 및 축(Metrics & Axes)을 선택하고 모드(Mode)를 정상(normal)으로 변경합니다.
12. 변경 사항 적용(Apply Changes) 및 저장(Save)을 선택합니다.

13. 이제 시각화 세 개를 Dashboards 대시보드에 추가할 수 있습니다. 대시보드(Dashboard)를 선택하고, 대시보드를 만들고, 시각화를 추가합니다.

5단계: 리소스 정리 및 다음 단계

불필요한 요금 부과를 피하려면 S3 버킷과 OpenSearch Service 도메인을 삭제하세요. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 버킷 삭제 및 이 가이드의 OpenSearch Service 도메인 삭제 섹션을 참조하세요.

녹취록은 MP3 파일보다 필요한 디스크 공간이 훨씬 적습니다. MP3 보존 기간을 단축하고(예: 통화 레코딩 3개월에서 1개월로 단축) 스토리지 비용을 절감할 수 있습니다.

또한 AWS Step Functions 및 Lambda를 사용하여 트랜스크립션 프로세스를 자동화하거나, 인덱싱 전에 메타데이터를 추가하거나, 정확한 사용 사례에 맞게 더 복잡한 시각화를 만들 수 있습니다.

Amazon OpenSearch Service 이름 변경 - 변경 사항 요약

2021년 9월 8일에 검색 및 분석 제품군의 이름이 Amazon OpenSearch Service로 변경되었습니다. OpenSearch Service는 OpenSearch 및 레거시 Elasticsearch OSS를 지원합니다. 다음 섹션에서는 이름 변경과 함께 변경된 서비스의 여러 부분과 도메인이 계속 제대로 작동하도록 하기 위해 수행해야 하는 작업에 대해 설명합니다.

이러한 변경 사항 중 일부는 도메인을 Elasticsearch에서 OpenSearch로 업그레이드할 때만 적용됩니다. Billing and Cost Management 콘솔과 같은 다른 경우에는 환경이 즉시 변경됩니다.

단, 이 목록이 전부는 아닙니다. 제품의 다른 부분도 변경되었지만 이러한 업데이트가 가장 적합합니다.

새로운 API 버전

새로운 버전의 OpenSearch Service 구성 API(2021년 1월 1일)는 기존 Elasticsearch OSS뿐만 아니라 OpenSearch와 함께 작동합니다. 21개의 API 작업이 보다 간결하고 엔진에 구애받지 않는 이름으로 대체되었지만(예:`CreateElasticsearchDomain`이 `CreateDomain`으로 변경 됨) OpenSearch Service는 두 가지 API 버전을 계속 지원합니다.

앞으로 새 API 작업을 사용하여 도메인을 생성하고 관리하는 것이 좋습니다. 새 API 작업을 사용하여 도메인을 생성할 때 `EngineVersion` 파라미터를 단순한 버전 번호가 아닌 `Elasticsearch_X.Y` 또는 `OpenSearch_X.Y`의 형식으로 지정해야 합니다. 버전을 지정하지 않을 경우 기본값은 최신 버전의 OpenSearch로 설정됩니다.

AWS CLI 를 사용하여 도메인을 생성하고 관리하려면 `aws opensearch ...`를 버전 1.20.40 이상으로 업그레이드합니다. 새로운 CLI 형식은 OpenSearch CLI 참조를 참조하세요.

인스턴스 유형의 이름 변경

이제 Amazon OpenSearch Service 인스턴스 유형의 형식은 `<type>.<size>.search`입니다(예: `m6g.large.`*`elasticsearch`*가 아닌 `m6g.large.`*`search`*). 별도의 조치를 할 필요는 없습니다. 기존 도메인은 API 및 Billing and Cost Management 콘솔에서 새 인스턴스 유형을 자동으로 참조하기 시작합니다.

예약 인스턴스(RI)가 있는 경우 계약은 변경의 영향을 받지 않습니다. 이전 구성 API 버전은 이전 명명 형식과 계속 호환되지만 새 API 버전을 사용하려면 새 형식을 사용해야 합니다.

액세스 정책 변경 사항

다음 섹션에서는 액세스 정책을 업데이트하기 위해 수행해야 하는 작업에 대해 설명합니다.

IAM 정책

이름이 바뀐 API 작업을 사용하려면 IAM 정책을 업데이트하는 것이 좋습니다. 그러나 OpenSearch Service는 이전 API 권한을 내부적으로 복제하여 기존 정책을 계속 준수합니다. 예를 들어, 현재 `CreateElasticsearchDomain` 작업을 수행할 수 있는 권한이 있는 경우 이제 `CreateElasticsearchDomain`(이전 API 작업) 및 `CreateDomain`(새 API 작업)을 모두 호출할 수 있습니다. 명시적 거부에도 동일하게 적용됩니다. 업데이트된 API 작업 목록은 정책 요소 참조를 참조하세요.

SCP 정책

서비스 제어 정책(SCP)은 표준 IAM에 비해 복잡성을 다시 한번 가중합니다. SCP 정책이 중단되는 것을 방지하려면 이전 및 새로운 API 작업을 모두 각 SCP 정책에 추가해야 합니다. 예를 들어, 사용자가 현재 `CreateElasticsearchDomain`에 대한 허용 권한이 있는 경우, `CreateDomain`에 대한 허용 권한도 부여하여 이들이 계속 도메인을 생성할 수 있도록 해야 합니다. 명시적 거부에도 동일하게 적용됩니다.

예제:

```
"Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:CreateElasticsearchDomain",
        "es:CreateDomain"
         ...
      ],
    },
      "Effect": "Deny",
      "Action:" [
        "es:DeleteElasticsearchDomain",
        "es:DeleteDomain"
         ...
```

새로운 리소스 유형

OpenSearch Service는 다음과 같은 새로운 리소스 유형을 도입합니다.

리소스	설명
`AWS::OpenSearchService::Domain`	Amazon OpenSearch Service 도메인을 나타냅니다. 이 리소스는 서비스 수준에 존재하며 도메인에서 실행되는 소프트웨어에만 국한되지 않습니다. AWS CloudFormation 및 AWS Resource Groups와 같은 서비스에 적용되며, 서비스 전체에 대한 리소스를 생성하고 관리합니다. CloudFormation 내에 정의된 도메인을 Elasticsearch에서 OpenSearch로 업그레이드하는 방법은 CloudFormation 사용 설명서의 설명을 참조하세요.
`AWS::OpenSearch::Domain`	도메인에서 실행 중인 OpenSearch/Elasticsearch 소프트웨어를 나타냅니다. 이 리소스는 AWS CloudTrail 및 AWS Config와 같은 서비스에 적용되며, OpenSearch Service 전체가 아닌 도메인에서 실행 중인 소프트웨어를 참조합니다. 이제 이러한 서비스에는 Elasticsearch를 실행하는 도메인(`AWS::Elasticsearch::Domain`)과 OpenSearch를 실행하는 도메인(`AWS::OpenSearch::Domain`)에 대한 별도의 리소스 유형이 포함됩니다.

> **Note**
>
> 하나 이상의 도메인을 OpenSearch로 업그레이드하는 경우에도 AWS Config에서 몇 주 동안 기존 `AWS::Elasticsearch::Domain` 리소스 유형의 데이터를 계속 볼 수 있습니다.

Kibana의 이름이 OpenSearch Dashboards로 변경

Kibana의 AWS 대안인 OpenSearch Dashboards는 OpenSearch와 함께 작동하도록 설계된 오픈 소스 시각화 도구입니다. Elasticsearch에서 OpenSearch로 도메인을 업그레이드하면 `/_plugin/kibana` 엔드포인트가 `/_dashboards`로 변경됩니다. OpenSearch Service는 모든 요청을 새 엔드포인트로 리디렉션하지만 IAM 정책에서 Kibana 엔드포인트를 사용하는 경우 새로운 `/_dashboards` 엔드포인트도 포함하도록 해당 정책을 업데이트합니다.

the section called “OpenSearch 대시보드에 대한 SAML 인증”을 사용하는 경우 도메인을 OpenSearch로 업그레이드하기 전에 자격 증명 공급자(IdP)에 구성된 모든 Kibana URL을 `/_plugin/kibana`에서 `/_dashboards`로 변경해야 합니다. 가장 일반적인 URL은 Assertion Consumer Service(ACS) 및 수신자 URL입니다.

OpenSearch Dashboards의 기본 `kibana_read_only` 역할이 `opensearch_dashboards_read_only`(으)로 이름이 변경되었으며 `kibana_user` 역할이 `opensearch_dashboards_user`(으)로 이름이 변경되었습니다. 변경 사항은 모든 서비스 소프트웨어 R20211203 이상이 설치된 새로 생성된 OpenSearch 1.x도메인에 적용됩니다. 기존 도메인을 서비스 소프트웨어 R20211203으로 업그레이드하 경우 역할 이름은 동일하게 유지됩니다.

CloudWatch 지표의 이름 변경

OpenSearch를 실행하는 도메인에 대한 CloudWatch 지표가 몇 가지 변경되었습니다. 도메인을 OpenSearch로 업그레이드하면 지표가 자동으로 변경되고 현재 CloudWatch 경보가 중단됩니다. 클러스터를 Elasticsearch 버전에서 OpenSearch 버전으로 업그레이드하기 전에 새 지표를 사용하도록 CloudWatch 경보를 업데이트해야 합니다.

다음 지표가 변경되었습니다.

원래 지표 이름	새 이름
`KibanaHealthyNodes`	`OpenSearchDashboardsHealthyNodes`
`KibanaConcurrentConnections`	`OpenSearchDashboardsConcurrentConnections`
`KibanaHeapTotal`	`OpenSearchDashboardsHeapTotal`
`KibanaHeapUsed`	`OpenSearchDashboardsHeapUsed`

원래 지표 이름	새 이름
KibanaHeapUtilization	OpenSearchDashboardsHeapUtilization
KibanaOS1MinuteLoad	OpenSearchDashboardsOS1MinuteLoad
KibanaRequestTotal	OpenSearchDashboardsRequestTotal
KibanaResponseTimesMaxInMillis	OpenSearchDashboardsResponseTimesMaxInMillis
ESReportingFailedRequestSysErrCount	KibanaReportingFailedRequestSysErrCount
ESReportingRequestCount	KibanaReportingRequestCount
ESReportingFailedRequestUserErrCount	KibanaReportingFailedRequestUserErrCount
ESReportingSuccessCount	KibanaReportingSuccessCount
ElasticsearchRequests	OpenSearchRequests

OpenSearch Services가 Amazon CloudWatch로 전송하는 지표의 전체 목록은 the section called “클러스터 지표 모니터링”를 참조하세요.

Billing and Cost Management 콘솔 변경 사항

결제 및 비용 관리 콘솔 및 비용 및 사용 보고서의 기록 데이터는 이전 서비스 이름을 계속 사용하므로 데이터를 검색할 때 Amazon OpenSearch Service와 레거시 Elasticsearch 이름 모두에 대한 필터를 사용해야 합니다. 기존의 저장된 보고서가 있는 경우 필터를 업데이트하여 OpenSearch Service도 포함하도록 합니다. Elasticsearch의 사용량이 감소하고 OpenSearch의 사용량이 증가하면 처음에 알림이 표시될 수 있지만 며칠 이내에 사라집니다.

서비스 이름 외에도 다음 필드는 모든 보고서, 청구서, 가격 목록 API 작업에서 변경됩니다.

필드	이전 형식	행 형식
인스턴스 유형	`m5.large.elasticsearch`	`m5.large.search`
제품군	Elasticsearch 인스턴스 Elasticsearch 볼륨	Amazon OpenSearch Service 인스턴스 Amazon OpenSearch Service 볼륨
가격 설명	c5.18xlarge.elasticsearch 인스턴스 시간(또는 부분적인 시간)당 5.098 USD - EU	c5.18xlarge.search 인스턴스 시간(또는 부분적인 시간)당 5.098 USD - EU
인스턴스 패밀리	`ultrawarm.elasticsearch`	`ultrawarm.search`

새로운 이벤트 형식

OpenSearch Service가 Amazon EventBridge와 Amazon CloudWatch에 전송하는 이벤트 형식이 변경되었습니다. 특히 `detail-type` 필드가 변경되었습니다. 소스 필드(`aws.es`)는 동일하게 유지됩니다. 각 이벤트 유형에 대한 전체 형식은 the section called “이벤트 모니터링” 섹션을 참조하세요. 이전 형식에 따라 달라지는 기존 이벤트 규칙이 있는 경우 새 형식에 맞게 업데이트해야 합니다.

변경되지 않는 것은 무엇입니까?

나열되지 않은 기능 중 다음 기능은 동일하게 유지됩니다.

- 서비스 보안 주체(`es.amazonaws.com`)
- 공급 업체 코드
- 도메인 ARN
- 도메인 엔드포인트

시작하기: 도메인을 OpenSearch 1.x로 업그레이드

OpenSearch 1.x는 Elasticsearch 버전 6.8 및 7.x에서의 업그레이드를 지원합니다. 도메인을 업그레이드하는 방법에 대한 지침은 the section called “도메인 업그레이드(콘솔)” 섹션을 참조하세요. AWS CLI 또는 구성 API를 사용하여 도메인을 업그레이드하는 경우를 `TargetVersion`로 지정해야 합니다`OpenSearch_1.x`.

OpenSearch 1.x에 호환성 모드 사용 설정이라는 추가 도메인 설정이 도입되었습니다. 특정 Elasticsearch OSS 클라이언트 및 플러그인은 연결하기 전에 클러스터 버전을 확인하기 때문에 호환성 모드에서는 OpenSearch가 해당 버전을 7.10으로 보고하도록 설정하여 이러한 클라이언트가 계속 작동하도록 합니다.

OpenSearch 도메인을 처음 생성하거나 Elasticsearch 버전에서 OpenSearch로 업그레이드할 때 호환성 모드를 활성화할 수 있습니다. 설정되지 않은 경우 파라미터의 기본값은 도메인을 생성할 때 `false`, `true`도메인을 업그레이드할 때입니다.

구성 API를 사용하여 호환성 모드를 활성화하려면, `override_main_response_version`을 `true`로 설정합니다.

```
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"
   }
}
```

기존 OpenSearch 도메인에서 호환성 모드를 활성화 또는 비활성화하려면 OpenSearch _cluster/settings API 작업을 사용해야 합니다.

```
PUT /_cluster/settings
{
  "persistent" : {
    "compatibility.override_main_response_version" : true
  }
}
```

Amazon OpenSearch Service 문제 해결

이 주제에서는 일반적인 Amazon OpenSearch Service 문제를 식별하고 해결하는 방법에 대해 설명합니다. AWS Support에 문의하기 전에 이 단원의 정보를 참조하세요.

OpenSearch Dashboards에 액세스할 수 없습니다.

OpenSearch Dashboards 엔드포인트는 서명된 요청을 지원하지 않습니다. 해당 도메인의 액세스 제어 정책에서 일부 IAM 역할에만 액세스 권한을 부여하고 Amazon Cognito 인증을 구성하지 않은 경우, Dashboards에 액세스하려고 하면 다음과 같은 오류가 발생할 수 있습니다.

```
"User: anonymous is not authorized to perform: es:ESHttpGet"
```

OpenSearch Service 도메인에서 VPC 액세스를 사용하는 경우에는 이 오류가 발생하지 않지만 요청 시간이 초과될 수 있습니다. 이 문제를 바로잡는 방법과 사용 가능한 각종 구성 옵션에 대해 알아보려면 the section called “대시보드에 대한 액세스 제어 ”, the section called “VPC 도메인 액세스 정책에 대하여” 및 the section called “ID 및 액세스 관리” 섹션을 참조하세요.

VPC 도메인에 액세스할 수 없습니다.

the section called “VPC 도메인 액세스 정책에 대하여” 및 the section called “VPC 도메인 테스트” 섹션을 참조하세요.

읽기 전용 상태의 클러스터

이전 Elasticsearch 버전에 비해, OpenSearch 및 Elasticsearch 7.x는 클러스터 조정 시 다른 시스템을 사용합니다. 이 새로운 시스템에서 클러스터가 쿼럼을 잃으면 조치를 할 때까지 클러스터를 사용할 수 없습니다. 쿼럼 손실은 두 가지 형태를 취할 수 있습니다.

- 클러스터가 전용 프라이머리 노드를 사용하는 경우 절반 이상을 사용할 수 없으면 쿼럼 손실이 발생합니다.
- 클러스터가 전용 프라이머리 노드를 사용하지 않는 경우 절반 이상의 데이터 노드를 사용할 수 없으면 쿼럼 손실이 발생합니다.

쿼럼 손실이 발생한 경우 클러스터에 둘 이상의 노드가 있으면 OpenSearch Service가 쿼럼을 복원하고 클러스터를 읽기 전용 상태로 설정합니다. 여기에는 두 가지 옵션이 있습니다.

- 읽기 전용 상태를 제거하고 클러스터를 그대로 사용합니다.
- 스냅샷에서 클러스터 또는 개별 인덱스를 복원합니다.

클러스터를 그대로 사용하려면 다음 요청을 사용하여 클러스터 상태가 녹색인지 확인합니다.

```
GET _cat/health?v
```

클러스터 상태가 빨간색이면 스냅샷에서 클러스터를 복원하는 것이 좋습니다. 문제 해결 단계는 the section called “빨간색 클러스터 상태” 섹션을 참조하세요. 클러스터 상태가 녹색이면 다음 요청을 사용하여 모든 예상 인덱스가 있는지 확인합니다.

```
GET _cat/indices?v
```

그런 다음 몇 가지 검색을 실행하여 예상 데이터가 있는지 확인합니다. 예상 데이터가 있으면 다음 요청을 사용하여 읽기 전용 상태를 제거할 수 있습니다.

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.blocks.read_only": false
  }
}
```

쿼럼 손실이 발생한 경우 클러스터에 노드가 하나만 있으면 OpenSearch Service가 노드를 교체하고 클러스터를 읽기 전용 상태로 설정하지 않습니다. 그렇지 않은 경우에는 방법이 동일합니다. 클러스터를 그대로 사용하거나 스냅샷에서 복원합니다.

두 상황 모두 OpenSearch Service는 AWS Health Dashboard에 두 개의 이벤트를 전송합니다. 첫 번째 이벤트는 쿼럼의 손실을 알려줍니다. 두 번째 이벤트는 OpenSearch Service가 쿼럼을 성공적으로 복원한 후에 발생합니다. 사용에 대한 자세한 내용은 AWS Health 사용 설명서를 AWS Health Dashboard참조하세요.

빨간색 클러스터 상태

빨간색 클러스터 상태는 하나 이상의 기본 샤드와 복제본이 노드에 할당되어 있지 않음을 나타냅니다. OpenSearch Service는 상태와 관계없이 모든 인덱스의 자동 스냅샷을 만들려고 시도하지만 빨간색 클러스터 상태가 지속되는 동안에는 스냅샷이 실패합니다.

빨간색 클러스터 상태의 가장 흔한 원인은 실패한 클러스터 노드 및 OpenSearch 처리 작업 부하가 지속해서 높아서 발생한 프로세스 충돌입니다.

> **Note**
>
> OpenSearch Service는 클러스터 상태와 관계없이 14일 동안 자동 스냅샷을 저장합니다. 따라서, 빨간색 클러스터 상태가 2주 이상 지속되면 마지막 정상적인 자동 스냅샷이 삭제되고 클러스터의 데이터가 영구적으로 손실될 수 있습니다. OpenSearch Service 도메인이 빨간색 클러스터 상태가 되면에 지원 문의하여 문제를 직접 해결할지 또는 지원 팀이 지원할지 문의할 수 있습니다. 빨간색 클러스터 상태가 발생할 경우 이를 알려주는 CloudWatch 경보를 설정할 수 있습니다.

빨간색 샤드는 빨간색 클러스터를 초래하고, 빨간색 인덱스는 빨간색 샤드를 초래합니다. 빨간색 클러스터 상태를 초래하는 인덱스를 식별하기 위해 OpenSearch는 몇 가지 유용한 API를 제공합니다.

- `GET /_cluster/allocation/explain`은 할당되지 않은 첫 번째 샤드를 선택하고, 노드에 할당되지 못한 이유를 확인하여 보여줍니다.

```
{
    "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`는 각 인덱스의 상태, 문서 수, 디스크 사용량을 보여줍니다.

```
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
```

빨간색 인덱스를 삭제하는 것은 빨간색 클러스터 상태를 해결하는 가장 빠른 방법입니다. 빨간색 클러스터 상태의 이유에 따라 OpenSearch Service 도메인을 확장하여 더 큰 인스턴스 유형, 더 많은 인스턴스 또는 더 많은 EBS 기반 스토리지를 사용하도록 한 후 문제가 발생한 인덱스를 다시 생성해 볼 수 있습니다.

문제가 발생한 인덱스가 삭제되지 않으면 스냅샷을 복원하거나, 인덱스에서 문서를 삭제하거나, 인덱스 설정을 변경하거나, 복제본 수를 줄이거나, 다른 인덱스를 삭제하여 디스크 공간을 확보합니다. OpenSearch Service 도메인을 재구성하기 전에 빨간색 클러스터 상태를 해결하는 것이 중요합니다. 빨간색 클러스터가 있는 도메인을 다시 구성할 경우 문제가 심각해져서 상태를 해결할 때까지 도메인이 처리 중 구성 상태로 중단될 수 있습니다.

빨간색 클러스터의 자동 수정

클러스터의 상태가 1시간 이상 지속적으로 빨간색으로 표시되면 OpenSearch Service는 할당되지 않은 샤드를 다시 라우팅하거나 이전 스냅샷에서 복원하여 자동으로 문제를 해결하려고 시도합니다.

하나 이상의 빨간색 인덱스를 수정하지 못하고 클러스터 상태가 총 14일 동안 빨간색으로 유지되는 경우 OpenSearch Service는 클러스터가 다음 기준 중 1개 이상을 충족하는 경우에만 추가 조치를 합니다.

- 가용 영역이 하나만 있음
- 전용 프라이머리 노드 없음
- 버스트 가능한 인스턴스 유형(T2 또는 T3) 포함

현재 클러스터가 이러한 기준 중 하나를 충족하면 OpenSearch Service가 이러한 인덱스를 수정하지 않으면 할당되지 않은 모든 샤드가 삭제됨을 설명하는 알림을 다음 7일 동안 매일 전송합니다. 21일 후에도 클러스터 상태가 여전히 빨간색으로 표시되면 OpenSearch Service는 모든 빨간색 인덱스에서 할당되지 않은 샤드(스토리지 및 컴퓨팅)를 삭제합니다. 이러한 각 이벤트에 대한 OpenSearch Service 콘솔의 Notifications(알림) 패널에서 알림을 수신합니다. 자세한 내용은 the section called “클러스터 상태 이벤트” 단원을 참조하십시오.

지속해서 과도한 처리 로드에서 복구

빨간색 클러스터 상태의 원인이 데이터 노드의 지속해서 과도한 처리 로드인지 확인하려면 다음 클러스터 지표를 모니터링합니다.

관련 측정치	설명	복구
JVMMemoryPressure	클러스터의 모든 데이터 노드에 사용되는 Java 힙의 비율을 지정합니다. 이 지표의 Maximum(최대) 통계를 모니터링하면서 Java 가비지 수집기가 충분한 메모리를 회수하지 못하여 메모리 압력 감소가 점차 작아지는 때를 찾습니다. 복합 쿼리 또는 큰 데이터 필드가 이러한 패턴의 원인일 수 있습니다. x86 인스턴스 유형은 일시 중지 시간을 짧게 하기 위해 애플리케이션 스레드와 함께 실행되는 Concurrent Mark Sweep(CMS) 가비지 수집기를 사용합니다. CMS가 정상 수집 중에 충분한 메모리를 회수하지 못하면 전체 가비지 수집이 트리거되어 애플리케이션이 일시 중지되므로 클러스터 안정성에 영향을 줄 수 있습니다. ARM 기반 Graviton 인스턴스 유형은 CMS와 유사한 Garbage-First(G1) 가비지 수집기를 사용하지만 추가적인 짧은 일시 중지 및 힙 조각 모음 기능을 사용하여 전체 가비지 수집의 필요성을 더욱 줄입니다. 두 경우 모두 전체 가비지 수집 중에 가비지 수집기가 회수할 수 있는 용량 이상으로 메모리 사용량이 계속 증가하면 메모리 부족 오류가 발생하며 OpenSearch가 작동을 멈춥니다. 모든 인스턴스 유형에서 사용량을 80% 미만으로 유지하는 것이 좋습니다.	JVM에 대해 메모리 회로 차단기를 설정합니다. 자세한 내용은 the section called “JVM OutOfMemoryError” 단원을 참조하십시오. 그래도 문제가 지속되면 불필요한 인덱스를 삭제하거나, 도메인에 대한 요청 수 또는 복잡성을 줄이거나, 인스턴스를 추가하거나, 더욱 큰 용량의 인스턴스 유형을 사용합니다.

관련 측정치	설명	복구
	_nodes/stats/jvm API는 JVM 통계와 메모리 풀 사용량, 그리고 가비지 수집 정보를 유용하게 요약하여 제공합니다. `GET domain-endpoint /_nodes/stats/jvm?pretty`	
CPUUtilization	클러스터의 데이터 노드에 사용되는 CPU 리소스의 비율을 지정합니다. 이 지표에 대한 Maximum(최대) 통계를 보고 지속해서 높은 사용 패턴이 있는지 찾습니다.	데이터 노드를 추가하거나 기존 데이터 노드의 인스턴스 유형 크기를 늘립니다.
Nodes(노드)	클러스터에 있는 노드 수를 지정합니다. 이 지표에 대한 Minimum(최소) 통계를 봅니다. 서비스에서 클러스터에 새 인스턴스 집합을 배포하는 경우 이 값이 변동됩니다.	데이터 노드를 추가합니다.

노란색 클러스터 상태

노란색 클러스터 상태는 모든 인덱스의 기본 샤드가 클러스터의 노드에 할당되어 있지만 하나 이상의 인덱스에 복제본 샤드가 할당되어 있지 않음을 나타냅니다. 단일 노드 클러스터는 OpenSearch Service가 복제본을 할당할 수 있는 다른 노드가 없기 때문에 항상 노란색 클러스터 상태로 초기화됩니다. 녹색 클러스터 상태가 되려면 노드 개수를 늘립니다. 자세한 내용은 the section called “도메인 크기 조정” 섹션을 참조하세요.

새 인덱스를 생성한 후 또는 노드 실패 후에 다중 노드 클러스터가 잠시 노란색 클러스터 상태가 될 수 있습니다. OpenSearch가 클러스터 전체에 데이터를 복제하면 이 상태는 자체 해결됩니다. 디스크 공간 부족이 노란색 클러스터 상태를 일으킬 수도 있습니다. 클러스터는 노드를 수용할 디스크 공간이 있는 경우에만 복제본 샤드를 배포할 수 있습니다.

ClusterBlockException

다음과 같은 이유로 `ClusterBlockException` 오류가 발생할 수 있습니다.

사용 가능한 스토리지 공간 부족

클러스터에 있는 하나 이상의 노드에 저장 공간이 최소값인 1) 사용 가능한 저장 공간의 20% 또는 2) 저장 공간 20GiB보다 작은 경우 문서 추가 및 인덱스 생성과 같은 기본적인 쓰기 작업이 실패하기 시작할 수 있습니다. the section called “스토리지 요구 사항 계산”에서는 OpenSearch Service가 디스크 공간을 사용하는 방법에 대한 요약을 제공합니다.

문제를 방지하려면 OpenSearch Service 콘솔의 `FreeStorageSpace` 지표를 모니터링하여 `FreeStorageSpace`가 특정 임계값 아래로 떨어지면 트리거하는 CloudWatch 경보를 생성합니다. 또한 `GET /_cat/allocation?v`은 샤드 할당 및 디스크 사용에 대한 유용한 요약 정보를 제공합니다. 스토리지 공간 부족과 관련된 문제를 해결하려면 더 큰 인스턴스 유형, 더 많은 인스턴스 또는 더 많은 EBS 기반 스토리지를 사용하도록 OpenSearch Service 도메인을 확장합니다.

높은 JVM 메모리 압력

JVMMemoryPressure 지표가 30분간 92%를 초과하면 OpenSearch Service는 보호 메커니즘을 트리거하고 모든 쓰기 작업을 차단하여 클러스터가 빨간색 상태가 되지 않도록 합니다. 보호가 설정되면 `ClusterBlockException` 오류가 뜨면서 쓰기 작업에 실패하고, 새 인덱스를 만들 수 없고, `IndexCreateBlockException` 오류가 발생합니다.

JVMMemoryPressure 지표가 5분간 88% 이하로 돌아가면 이 보호 조치는 비활성화되고 클러스터에 대한 쓰기 작업이 다시 허용됩니다.

높은 JVM 메모리 압력은 클러스터에 대한 요청 수가 급증하거나, 노드 간 샤드 할당이 불균형하거나, 클러스터에 샤드가 너무 많거나, 필드 데이터 또는 인덱스 매핑이 폭발적으로 증가하거나, 들어오는 부하를 처리할 수 없는 인스턴스 유형 등으로 인해 발생할 수 있습니다. 쿼리에 집계, 와일드카드 또는 광범위한 시간 범위를 사용하는 경우에도 발생할 수 있습니다.

클러스터에 대한 트래픽을 줄이고 JVM 메모리가 많이 소모되는 문제를 해결하려면, 다음 중 하나 이상을 시도해 보세요.

- 노드당 최대 힙 크기가 32GB가 되도록 도메인을 조정하세요.
- 오래되거나 사용되지 않는 인덱스를 삭제하여 샤드 수를 줄이세요.
- `POST index-name/_cache/clear?fielddata=true` API 작업으로 데이터 캐시를 지우세요. 캐시를 지우면 진행 중인 쿼리가 중단될 수 있다는 점에 유의하시기 바랍니다.

일반적으로, 이후 높은 JVM 메모리 압력을 방지하려면 다음 모범 사례를 따르세요.

- 텍스트 필드를 집계하거나 `keyword`에 대한 인덱스의 매핑 형식을 변경하지 않도록 하세요.
- 올바른 샤드 수 선택으로 검색 및 인덱싱 요청을 최적화하세요.
- 정기적으로 사용되지 않는 인덱스를 삭제하도록 인덱스 상태 관리(ISM) 정책을 설정하세요.

Multi-AZ with Standby로의 마이그레이션 오류

기존 도메인을 Multi-AZ with standby로 마이그레이션할 때 다음과 같은 문제가 발생할 수 있습니다.

대기 모드가 없는 도메인에서 대기 모드가 있는 도메인으로 마이그레이션하는 동안 인덱스, 인덱스 템플릿 또는 ISM 정책 생성

Multi-AZ without Standby에서 Multi-AZ with Standby로 도메인을 마이그레이션하는 동안 인덱스를 생성하고 인덱스 템플릿 또는 ISM 정책이 권장 데이터 복사 지침을 따르지 않으면 데이터 불일치가 발생하고 마이그레이션이 실패할 수 있습니다. 이러한 상황을 방지하려면 데이터 복사 횟수(프라이머리 노드와 복제본 모두 포함)가 3의 배수인 새 인덱스를 만드십시오. `DescribeDomainChangeProgress` API를 사용하여 마이그레이션 진행 상황을 확인할 수 있습니다. 복제본 개수 오류가 발생하는 경우 오류를 수정한 다음 AWS Support에 문의하여 마이그레이션을 다시 시도하세요.

잘못된 데이터 복사본 수

도메인에 적절한 수의 데이터 사본이 없는 경우 Multi-AZ with Standby로 마이그레이션하는 작업이 실패합니다.

JVM OutOfMemoryError

JVM `OutOfMemoryError`는 일반적으로 다음 JVM 회로 차단기 중 하나에 도달했음을 의미합니다.

회로 차단기	설명	클러스터 설정 속성
상위 차단기	모든 회로 차단기에 대해 허용되는 JVM 힙 메모의 총비율입니다. 기본값은 95%입니다.	`indices.breaker.total.limit`

회로 차단기	설명	클러스터 설정 속성
필드 데이터 차단기	메모리에 단일 데이터 필드를 로드하도록 허용된 JVM 힙 메모리의 비율입니다. 기본값은 40%입니다. 큰 필드가 포함된 데이터를 업로드하는 경우에는 이 제한을 늘려야 할 수 있습니다.	`indices.breaker.fielddata.limit`
요청 차단기	서비스 요청에 응답하는 데 사용되는 데이터 구조에 대해 허용되는 JVM 힙 메모리의 비율입니다. 기본값은 60%입니다. 서비스 요청에 집계 계산이 포함된 경우 이 제한을 늘려야 할 수 있습니다.	`indices.breaker.request.limit`

실패한 클러스터 노드

Amazon EC2 인스턴스가 예기치 않게 종료되고 다시 시작될 수 있습니다. 일반적으로 OpenSearch Service는 노드를 자동으로 다시 시작합니다. 그러나 OpenSearch 클러스터의 노드 하나 이상에서 실패 조건이 남아 있을 수 있습니다.

이러한 조건이 있는지 확인하려면 OpenSearch Service 콘솔에서 도메인 대시보드를 엽니다. Cluster health(클러스터 상태) 탭으로 이동한 후 Total nodes(총 노드) 지표를 찾습니다. 보고된 노드 수가 클러스터에 대해 구성한 노드 수보다 적은지 확인합니다. 이 지표가 하나 이상의 노드가 하루 이상 다운되었음을 표시하면 AWS Support에 문의하세요.

이런 문제가 발생할 경우 이를 알려주는 CloudWatch 경보를 설정할 수도 있습니다.

> **Note**
>
> 클러스터 구성 변경 중 그리고 서비스에 대한 정기 유지보수 중에는 Total nodes(총 노드) 지표가 정확하지 않습니다. 이는 예상된 동작입니다. 따라서 이 지표는 곧 정확한 클러스터 노드 개수를 보고합니다. 자세한 내용은 the section called “구성 변경” 섹션을 참조하세요.

예기치 않은 노드 종료 및 다시 시작으로부터 클러스터를 보호하려면 OpenSearch Service 도메인의 각 인덱스에 대해 복제본을 하나 이상 생성합니다.

최대 샤드 제한 초과

OpenSearch뿐만 아니라 Elasticsearch 7.x 버전의 기본 설정에는 노드당 1,000개 이하의 샤드가 있습니다. 새 인덱스 생성과 같은 요청으로 인해 이 제한을 초과하는 경우 OpenSearch/ElasticSearch에서 오류가 발생합니다. 이 오류가 발생한 경우 몇 가지 옵션이 있습니다.

- 더 많은 데이터 노드를 클러스터에 추가합니다.
- `_cluster/settings/cluster.max_shards_per_node` 설정을 늘립니다.
- _shrink API를 사용하여 노드의 샤드 수를 줄입니다.

도메인이 처리 상태에 멈춤

구성 변경 중일 때 OpenSearch Service 도메인은 “처리 중(Processing)” 상태가 됩니다. 구성 변경을 시작하면 도메인 상태가 처리 중(Processing)으로 변경되며 OpenSearch Service가 새 환경을 생성합니다. 새로운 환경에서 OpenSearch Service는 새로운 적용 가능한 노드(예: 데이터, 마스터 또는 UltraWarm) 세트를 시작합니다. 마이그레이션이 완료되면 이전 노드가 종료됩니다.

다음과 같은 상황 중 하나가 발생하는 경우 클러스터가 “처리 중(Processing)” 상태로 멈출 수 있습니다.

- 새 데이터 노드 집합이 시작되지 않습니다.

- 새 데이터 노드 집합으로의 샤드 마이그레이션이 실패했습니다.
- 오류가 발생하여 유효성 검사에 실패했습니다.

이러한 각 상황에 대한 자세한 해결 단계는 Amazon OpenSearch Service 도메인이 “처리 중(Processing)” 상태로 멈춘 이유는 무엇입니까?를 참조하세요.

낮은 EBS 버스트 밸런스

범용(SSD) 볼륨 중 하나의 EBS 버스트 밸런스가 70% 미만이면 OpenSearch Service에서 콘솔 알림을 보내고, 밸런스가 20% 미만으로 떨어지면 후속 알림을 보냅니다. 이 문제를 해결하려면 클러스터를 스

케일 업하거나, 읽기 및 쓰기 IOPS를 줄여 버스트 밸런스가 반영되도록 할 수 있습니다. gp3 볼륨 유형이 있는 도메인과 볼륨 크기가 1000GiB를 초과하는 gp2 볼륨이 있는 도메인의 경우 버스트 균형은 0으로 유지됩니다. 자세한 정보는 범용 SSD 볼륨(gp2)을 참조하세요. `BurstBalance` CloudWatch 지표를 사용하여 EBS 버스트 밸런스를 모니터링할 수 있습니다.

볼륨 크기 조정 중 EBS 지표 증가

Amazon Elastic Block Store 볼륨 크기를 수정할 때 , `Write Throughput`, 및 `Write Throughput Micro burstingDisk Queue Depth`와 같은 다양한 EBS 지표가 일시적으로 증가하는 것을 관찰할 수 있습니다`IOPS`. 이는 크기 조정 작업 중에 예상되는 동작이며 일반적으로 몇 초 동안 지속됩니다. 그러나 기간은 수정 중인 볼륨 크기에 따라 다를 수 있습니다.

지연 시간 문제와 요청 거부를 방지하려면 클러스터가 정상이고 네트워크 트래픽이 낮은 경우에만 EBS 볼륨 크기 조정을 수행합니다.

감사 로그를 활성화할 수 없음

OpenSearch Service 콘솔을 사용하여 감사 로그 게시를 활성화하려고 하면 다음 오류가 발생할 수 있습니다.

CloudWatch Logs 로그 그룹에 대해 지정된 리소스 액세스 정책은 Amazon OpenSearch Service에서 로그 스트림을 생성할 수 있는 충분한 권한을 부여하지 않습니다. 리소스 액세스 정책을 확인하세요.

이 오류가 발생하면 정책의 `resource` 요소에 올바른 로그 그룹 ARN 이 포함되었는지 확인합니다. 포함되었다면 다음 단계를 수행합니다.

1. 몇 분 정도 기다립니다.
2. 웹 브라우저에서 페이지를 새로 고칩니다.
3. Select existing group(기존 그룹 선택)을 선택합니다.
4. Existing log group(기존 로그 그룹)에서 오류 메시지를 받기 전에 생성한 로그 그룹을 선택합니다.
5. 액세스 정책 섹션에서 Select existing policy(기존 정책 선택)를 선택합니다.
6. Existing policy(기존 정책)에서 오류 메시지를 받기 전에 생성한 정책을 선택합니다.
7. Enable(활성화)를 선택합니다.

프로세스를 여러 번 반복한 후에도 오류가 계속되면 AWS Support에 문의하세요.

인덱스를 닫을 수 없음

OpenSearch Service는 OpenSearch 및 Elasticsearch 버전 7.4 이상에서만 `_close` API를 지원합니다. 이전 버전을 사용하고 있으며 스냅샷에서 인덱스를 복원하는 경우 기존 인덱스를 삭제할 수 있습니다(다시 인덱스를 만들기 전 또는 후).

클라이언트 라이선스 확인

Logstash 및 Beats의 기본 배포판에는 독점 라이선스 검사가 포함되어 있으며 오픈 소스 버전의 OpenSearch에 연결되지 않습니다. 이러한 클라이언트의 Apache 2.0(OSS) 배포판을 OpenSearch Service와 함께 사용해야 합니다.

요청 제한

지속해서 `403 Request throttled due to too many requests` 또는 `429 Too Many Requests` 오류가 발생하면 수직 확장을 고려합니다. Amazon OpenSearch Service는 페이로드로 인해 메모리 사용량이 Java 힙의 최대 크기를 초과할 경우 요청을 제한합니다.

노드에 SSH할 수 없음

SSH를 사용하여 OpenSearch 클러스터의 어떤 노드에도 액세스할 수 없고 `opensearch.yml`을 직접 수정할 수 없습니다. 대신 콘솔 AWS CLI또는 SDKs를 사용하여 도메인을 구성합니다. OpenSearch REST API를 사용하여 몇 가지 클러스터 수준 설정을 지정할 수도 있습니다. 자세한 내용은 Amazon OpenSearch Service API 참조 및 the section called “지원되는 연산자”을(를) 참조하세요.

클러스터 성능에 대한 더 자세한 통찰이 필요한 경우 CloudWatch에 오류 로그 및 느린 로그를 게시할 수 있습니다.

"객체 스토리지 클래스의 경우 유효하지 않음" 스냅샷 오류

OpenSearch Service 스냅샷은 S3 Glacier 스토리지 클래스를 지원하지 않습니다. 스냅샷을 나열하려 할 때 귀하의 S3 버킷이 객체를 S3 Glacier 스토리지 클래스로 전환하는 수명 주기를 갖고 있다면 이 에러가 발생합니다.

버킷으로부터 스냅샷을 복원하려면 S3 Glacier로부터 객체를 복원하고 새로운 버킷으로 복사한 뒤 스냅샷 리포지토리로 새로운 버킷을 등록합니다.

잘못된 호스트 헤더

OpenSearch Service에서는 클라이언트가 요청 헤드에 `Host`를 지정해야 합니다. 올바른 `Host` 값은 다음과 같이 `https://`가 없는 도메인 엔드포인트입니다.

```
Host: search-my-sample-domain-ih2lhn2ew2scurji.us-west-2.es.amazonaws.com
```

요청을 수행할 때 `Invalid Host Header` 오류가 발생하는 경우 클라이언트 또는 프록시의 `Host` 헤더에 OpenSearch Service 도메인 엔드포인트(예를 들어 IP 주소가 아님)가 포함되는지 확인합니다.

잘못된 M3 인스턴스 유형

OpenSearch Service에서는 OpenSearch 또는 Elasticsearch 버전 6.7 이상을 실행하는 기존 도메인에 M3 인스턴스를 추가하거나 수정할 수 없습니다. Elasticsearch 6.5 및 이전 버전에서 M3 인스턴스를 계속 사용할 수 있습니다.

최신 인스턴스 유형을 선택하는 것이 좋습니다. OpenSearch 또는 Elasticsearch 6.7 이상을 실행하는 도메인의 경우 다음 제한 사항이 적용됩니다.

- 기존 도메인에서 M3 인스턴스를 사용하지 않는 경우 더 이상 인스턴스로 변경할 수 없습니다.
- 기존 도메인을 M3 인스턴스 유형에서 다른 인스턴스 유형으로 변경하는 경우 다시 전환할 수 없습니다.

UltraWarm 활성화 후 핫 쿼리의 작동이 중지됨

도메인에 UltraWarm을 활성화할 때 `search.max_buckets` 설정에 기존 재정의가 없는 경우 OpenSearch Service는 값을 자동으로 `10000`으로 설정하여 메모리를 많이 사용하는 쿼리가 웜 노드를 포화시키는 것을 방지합니다. 핫 쿼리에서 10,000개 이상의 버킷을 사용하는 경우 UltraWarm을 활성화하면 핫 쿼리의 작동이 중지될 수 있습니다.

Amazon OpenSearch Service의 관리 특성으로 인해 이 설정을 수정할 수 없으므로 지원 사례를 열어 제한을 늘려야 합니다. 한도 증가에는 Premium Support 구독이 필요하지 않습니다.

업그레이드 후 다운그레이드할 수 없음

현재 위치 업그레이드는 되돌릴 수 없지만, AWS Support에 문의하면 새 도메인에서 자동 업그레이드 전 스냅샷을 복원하는 데 도움을 줄 수 있습니다. 예를 들어 도메인을 Elasticsearch 5.6에서 6.4로 업그

레이드하는 경우 AWS Support는 새 Elasticsearch 5.6 도메인에서 업그레이드 전 스냅샷을 복원하는 데 도움이 될 수 있습니다. 원래 도메인의 수동 스냅샷을 생성한 경우, 해당 단계를 직접 수행할 수 있습니다.

모든 AWS 리전에 대한 도메인의 요약 필요

다음 스크립트는 Amazon EC2 describe-regions AWS CLI 명령을 사용하여 OpenSearch Service를 사용할 수 있는 모든 리전 목록을 생성합니다. 그런 다음 각 리전에 대해 list-domain-names를 호출합니다.

```
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
```

각 리전에 대해 다음 출력을 수신합니다.

```
Listing domains in region:'us-west-2'...
[
  {
    "DomainName": "sample-domain"
  }
]
```

OpenSearch Service를 사용할 수 없는 리전은 "엔드포인트 URL에 연결할 수 없음"을 반환합니다.

OpenSearch Dashboards를 사용할 때 브라우저 오류

Dashboards를 사용하여 OpenSearch Service 도메인에서 데이터를 보려고 하면 브라우저가 HTTP 응답 객체에서 서비스 오류 메시지를 래핑합니다. 웹 브라우저에서 일반적으로 사용할 수 있는 개발자 도구(예: Chrome의 개발자 도구)를 사용하여 기본 서비스 오류를 확인하고 디버그 작업을 지원할 수 있습니다.

Chrome에서 서비스 오류를 보려면

1. Chrome 메뉴 모음에서 View(보기), Developer(개발자), Developer Tools(개발자 도구)를 차례로 선택합니다.
2. Network(네트워크) 탭을 선택합니다.

3. Status(상태) 열에서 상태가 500인 HTTP 세션을 선택합니다.

Firefox에서 서비스 오류를 보려면

1. 메뉴에서 Tools(도구), Web Developer(웹 개발자), Network(네트워크)를 차례로 선택합니다.
2. 상태가 500인 HTTP 세션을 선택합니다.
3. Response(응답) 탭을 선택하여 서비스 응답을 봅니다.

노드 샤드 및 스토리지 스큐

노드 샤드 스큐는 클러스터 내의 하나 이상의 노드에 다른 노드보다 훨씬 많은 샤드가 있는 경우입니다. 노드스토리지 스큐는 클러스터 내의 하나 이상의 노드에 다른 노드보다 훨씬 많은 스토리지(`disk.indices`)가 있는 경우입니다. 도메인에서 노드 하나를 교체했고 여전히 샤드를 노드에 할당 중인 경우처럼 이 두 조건 모두가 일시적으로 발생할 수 있습니다. 그러나 이러한 조건이 지속되는 경우 해결이 필요합니다.

두 가지 유형의 스큐를 모두 식별하려면 _cat/allocation API 작업을 실행하고 응답의 `shards` 및 `disk.indices` 항목을 비교합니다.

```
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
```

약간의 스토리지 스큐는 정상이지만 평균에서 10%를 초과하는 경우에는 주의해야 합니다. 샤드 배포가 왜곡되면 CPU, 네트워크 및 디스크 대역폭 사용량도 왜곡될 수 있습니다. 일반적으로 데이터가 많을수록 인덱싱 및 검색 작업이 늘어나기 때문에 가장 무거운 노드는 리소스가 가장 많이 사용되는 노드인 반면, 가벼운 노드는 리소스 활용도가 낮음을 나타냅니다.

해결 방법: 데이터 노드 수의 배수인 샤드 수를 사용하여 각 인덱스가 데이터 노드 간에 균등하게 분산되도록 합니다.

인덱스 샤드 및 스토리지 스큐

인덱스 샤드 스큐는 하나 이상의 노드가 다른 노드보다 인덱스의 샤드를 더 많이 보유하는 경우입니다. 인덱스 스토리지 스큐는 하나 이상의 노드가 인덱스의 총 스토리지 중 균형이 안 맞게 많은 양을 보유하는 경우입니다.

인덱스 스큐는 _cat/shards API 출력을 약간 조작해야 하기 때문에 노드 스큐보다 식별하기 어렵습니다. 클러스터 또는 노드 지표에 스큐 징후가 있는 경우 인덱스 스큐를 조사합니다. 인덱스 스큐의 일반 징후는 다음과 같습니다.

- 데이터 노드의 하위 집합에서 HTTP 429 오류 발생
- 데이터 노드 전체에서 고르지 않은 인덱스 또는 검색 작업 대기열
- 데이터 노드 전반에 걸쳐 균일한 JVM 힙 또는 CPU 사용률

해결 방법: 데이터 노드 수의 배수인 샤드 수를 사용하여 각 인덱스가 데이터 노드 간에 균등하게 분산되도록 합니다. 인덱스 저장소 또는 샤드 스큐가 계속 표시되는 경우, OpenSearch Service 도메인의 모든 블루/그린 배포에서 발생하는 샤드 재할당을 강제 실행해야 할 수 있습니다.

VPC 액세스를 선택한 후 허용되지 않은 작업

OpenSearch Service 콘솔을 사용하여 새 도메인을 만들 때 VPC 또는 공용 액세스를 선택하는 옵션이 있습니다. VPC access(VPC 액세스)를 선택할 경우, OpenSearch Service가 VPC 정보를 쿼리하는데, 올바른 정책이 없을 경우 쿼리가 실패합니다.

```
You are not authorized to perform this operation. (Service: AmazonEC2; Status Code:
 403; Error Code: UnauthorizedOperation
```

이 쿼리를 활성화하려면 `ec2:DescribeVpcs`, `ec2:DescribeSubnets` 및 `ec2:DescribeSecurityGroups` 작업에 대한 액세스 권한이 있어야 합니다. 이 요구 사항은 콘솔에만 해당됩니다. AWS CLI를 사용하여 VPC 엔드포인트가 있는 도메인을 생성하고 구성하는 경우 해당 작업에 액세스할 필요가 없습니다.

VPC 도메인 생성 후 로딩 단계에서 멈춤

VPC 액세스를 사용하는 새 도메인을 만든 후, 도메인의 Configuration state(구성 상태)가 Loading(로드 중)에서 더 이상 진행되지 않을 수 있습니다. 이 문제가 발생하면 해당 리전에 대해 AWS Security Token Service (AWS STS)가 비활성화되었을 수 있습니다.

VPC 엔드포인트를 VPC에 추가하려면 OpenSearch Service가 `AWSServiceRoleForAmazonOpenSearchService` 역할을 수임해야 합니다. 따라서 특정 리전에서 VPC 액세스를 사용하는 새 도메인을 생성하려면를 활성화해야 AWS STS 합니다. 활성화 및 비활성화에 대한 자세한 내용은 IAM 사용 설명서를 AWS STS참조하세요.

OpenSearch API에 대한 요청 거부됨

OpenSearch API에 태그 기반 액세스 제어를 도입하면서 이전에 없었던 액세스 거부 오류가 나타날 수 있습니다. 하나 이상의 액세스 정책에 `ResourceTag` 조건을 사용하는 `Deny`이(가) 포함되어 있고, 이러한 조건이 현재 적용되고 있어서 발생하는 것일 수 있습니다.

예를 들어 다음 정책은 도메인에 `environment=production` 태그가 있는 경우 구성 API의 `CreateDomain` 작업 액세스만 거부하는 데 사용되었습니다. `ESHttpPut`이(가) 작업 목록에도 포함되어 있었지만 해당 작업이나 다른 `ESHttp*` 작업에는 거부 문이 적용되지 않았습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:CreateDomain",
      "es:ESHttpPut"
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

OpenSearch HTTP 메서드에 대한 태그 지원이 추가되면서 위와 같은 IAM 자격 증명 기반 정책을 사용하면 연결된 사용자의 `ESHttpPut` 작업에 대한 액세스가 거부됩니다. 이전에는 태그 유효성 검사가 없어도 연결된 사용자가 PUT 요청을 계속 보낼 수 있었습니다.

도메인을 서비스 소프트웨어 R20220323 이상으로 업데이트한 후 액세스 거부 오류가 표시되기 시작하면 자격 증명 기반 액세스 정책을 확인하여 이러한 경우인지 확인하고 필요하다면 업데이트해 액세스를 부여합니다.

Alpine Linux에서 연결할 수 없음

Alpine Linux는 DNS 응답 크기를 512바이트로 제한합니다. Alpine Linux 버전 3.18.0 이하에서 OpenSearch Service 도메인에 연결하려고 할 때 도메인이 VPC에 있으며 20개가 넘는 노드가 있는 경우 DNS 확인이 실패할 수 있습니다. 3.18.0 이상의 Alpine Linux 버전을 사용하는 경우 20개 이상의 호스트를 해결할 수 있어야 합니다. 자세한 내용은 Alpine Linux 3.18.0 릴리스 정보를 참조하세요.

도메인이 VPC에 있는 경우 Debian, Ubuntu, CentOS, Red Hat Enterprise Linux 또는 Amazon Linux 2 등의 다른 Linux 배포를 사용하여 연결하는 것이 좋습니다.

Search Backpressure에 대한 요청이 너무 많음

CPU 기반 승인 제어는 트래픽의 유기적 증가와 급증 모두에 대해 현재 용량을 기준으로 노드에 대한 요청 수를 사전에 제한하는 게이트키핑 메커니즘입니다. 요청이 너무 많으면 거부 시 HTTP 429 “Too Many Requests” 상태 코드가 반환됩니다. 이 오류는 클러스터 리소스가 부족하거나, 리소스를 많이 사용하는 검색 요청 또는 의도하지 않은 워크로드 급증을 나타냅니다.

Search Backpression은 거부 사유를 제공하므로 리소스를 많이 사용하는 검색 요청을 세밀하게 조정하는 데 도움이 될 수 있습니다. 트래픽이 급증하는 경우 지수 백오프 및 지터를 사용하여 클라이언트 측 재시도를 하는 것이 좋습니다.

SDK를 사용할 때 인증서 오류

AWS SDKs 컴퓨터의 CA 인증서를 사용하기 때문에 SDK를 사용하려고 할 때 AWS 서버의 인증서를 변경하면 연결에 실패할 수 있습니다. 오류 메시지는 다양하지만 일반적으로 다음 텍스트가 포함되어 있습니다.

```
Failed to query OpenSearch
...
SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

컴퓨터의 CA 인증서와 운영 체제를 최신 상태로 유지하여 이러한 장애를 피할 수 있습니다. 본인 컴퓨터를 직접 관리하지 않는 기업 환경에서 이 문제가 발생하는 경우, 관리자에게 업데이트 프로세스를 문의해야 할 수 있습니다.

아래 목록에 최소한의 운영 체제 및 Java 버전이 나와 있습니다.

- 2005년 1월 이후 업데이트가 설치된 Microsoft Windows 버전의 경우, 신뢰할 수 있는 연결 목록에 필요한 CA가 하나 이상 들어 있습니다.
- Mac OS X 10.4 릴리스 5(2007년 2월), Mac OS X 10.5(2007년 10월) 및 그 이후 버전의 Java가 설치된 Mac OS X 10.4의 경우, 신뢰할 수 있는 연결 목록에 필요한 CA가 하나 이상 들어 있습니다.
- Red Hat Enterprise Linux 5(2007년 3월), 6, 7과 CentOS 5, 6, 7은 모두 신뢰할 수 있는 기본 CA 목록에 필요한 CA가 하나 이상 들어 있습니다.
- Java 1.4.2_12(2006년 5월), 5 업데이트 2(2005년 3월), 그리고 Java 6(2006년 12월), 7, 8을 포함한 이후의 모든 버전은 신뢰할 수 있는 기본 CA 목록에 필요한 CA가 하나 이상 들어 있습니다.

인증 기관은 다음 세 곳입니다.

- Amazon Root CA 1
- Starfield Services Root Certificate Authority - G2
- Starfield Class 2 Certification Authority

처음 두 기관의 루트 인증서는 Amazon Trust Services에서 구할 수 있지만, 컴퓨터를 최신 상태로 유지하는 것이 더욱 직접적인 해결책입니다. ACM 제공 인증서에 대한 자세한 내용은 AWS Certificate Manager FAQ 섹션을 참조하세요.

> **Note**
>
> 현재 us-east-1 리전의 OpenSearch Service 도메인은 다른 기관의 인증서를 사용합니다. 가까운 시일 안에 이 리전에서도 새 인증 기관을 사용하도록 업데이트할 예정입니다.

버전 호환성으로 인해 사용자 지정 플러그인 설치 실패

문제: 플러그인과 실행 중인 OpenSearch 인스턴스 간의 버전 불일치로 인해 플러그인 설치가 실패했습니다. 시스템에서 다음 오류를 반환합니다.

```
PluginValidationFailureReason : The provided plugin could not be loaded.
```

원인: 플러그인이 OpenSearch ${MAJOR}.$에 대해 컴파일되었습니다.{MINOR}.{PATCH}이지만 환경에서 OpenSearch ${MAJOR}.$를 실행 중입니다.{MINOR}.0. OpenSearch는 안정성과 보안을 위해 플러그인과 코어 OpenSearch 설치 간에 정확한 버전 일치가 필요합니다.

수정 방법: OpenSearch 버전 ${MAJOR}.$를 사용하여 플러그인 빌드클러스터 버전과 일치하는 {MINOR}.0.

OpenSearch 버전을 확인하고 업데이트하려면

1. 클러스터의 API 또는 대시보드를 사용하여 다음 명령을 실행합니다. ## ## ### ## 자신의 정보로 바꿉니다.

   API 요청:

   ```
   curl -X GET your-opensearch-endpoint/
   ```

   대시보드의 개발 도구 콘솔:

   ```
   GET /
   ```

   명령은 다음 형식으로 정보를 반환합니다.

   ```
   {
     "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. 버전 번호가 $*{MAJOR}*.$가 아닌 경우*{MINOR}*.0에서 다음을 수행하여 플러그인을 다시 빌드합니다.

   a. $*{MAJOR}*.$ 버전을 지정`descriptor.properties`하도록 플러그인을 업데이트합니다.*{MINOR}*.0.

   b. 프로젝트 유형에 대한 명령을 사용하여 플러그인을 다시 빌드합니다.

   c. 새로 빌드된 `.zip` 파일을 사용하여 update-package 명령을 실행합니다.

   d. associate-package 명령을 실행하여 `update-package` 이전 단계에서 명령을 실행할 때 생성된 최신 플러그인 버전을 연결합니다.

Amazon OpenSearch Service 문서 기록

다음 주제에서는 Amazon OpenSearch Service에서 변경된 중요 사항에 대해 설명합니다. 서비스 소프트웨어 업데이트에서 새 기능, 보안 패치, 버그 수정 및 기타 개선 사항에 대한 지원을 추가합니다. 새 기능을 사용하려면 도메인의 서비스 소프트웨어를 업데이트해야 할 수도 있습니다. 자세한 내용은 the section called “서비스 소프트웨어 업데이트” 단원을 참조하십시오.

서비스 기능은 서비스를 사용할 수 AWS 리전 있는에 점진적으로 배포됩니다. 이 문서는 첫 번째 릴리스에 대해서만 업데이트됩니다. 리전 가용성에 대한 정보를 제공하거나 후속 리전 롤아웃을 발표하지 않습니다. 서비스 기능의 리전 가용성에 대한 자세한 내용과 업데이트에 대한 알림을 구독하려면 What's New with AWS?를 참조하세요.

업데이트에 대한 알림을 받으려면 RSS 피드를 구독하시면 됩니다.

> **Note**
>
> 패치 릴리스: “-P”와 숫자로 끝나는 서비스 소프트웨어 버전(예: R20211203-P4)은 패치 릴리스입니다. 패치에는 성능 개선, 사소한 버그 수정, 보안 수정 또는 자세 개선이 포함될 수 있습니다. 패치에는 새로운 기능이나 주요 변경 사항이 포함되어 있지 않으므로 일반적으로 사용자 또는 문서에 직접적인 영향을 미치지 않으며 각 패치의 세부 사항이 이 문서 기록에 포함되지 않습니다.

변경 사항	설명	날짜
새로운 기능: SAML 속성 기반 세분화된 액세스 제어	Amazon OpenSearch Service는 SAML을 사용하여 세분화된 액세스 제어를 지원하여 ID 제공업체의 사용자 및 그룹을 OpenSearch 세분화된 액세스 제어 사용자 및 역할에 동적으로 매핑합니다. 이러한 역할의 범위를 특정 OpenSearch 도메인 및 서버리스 컬렉션으로 지정하고 인덱스 수준 권한 및 문서 수준 보안을 정의할 수 있습니다. 자세한 내용은 SAML 속	2025년 8월 8일

	성 기반 세분화된 액세스 제어 구성을 참조하세요.	
새 주제: OpenSearch UI 엔드포인트 및 할당량	Amazon OpenSearch UI는 현대화된 운영 분석 환경을 아시아 태평양(하이데라바드), 아시아 태평양(오사카), 아시아 태평양(서울), 유럽(밀라노), 유럽(취리히), 유럽(스페인), 미국 서부(캘리포니아 북부)를 비롯한 새로운 리전으로 확장하여 단일 엔드포인트에서 관리형 도메인 및 서버리스 컬렉션에 걸친 데이터 전반에 걸쳐 인사이트를 얻을 수 있습니다. OpenSearch UI는 버전 1.3 이상의 OpenSearch 도메인 및 OpenSearch 서버리스 컬렉션에 연결할 수 있습니다. 이제 22개 AWS 리전에서 사용할 수 있습니다.	2025년 8월 7일

새로운 기능: 자동 의미 체계 보강	자동 의미 체계 보강은 Amazon OpenSearch Service에서 의미 체계 검색 구현 및 기능을 간소화합니다. 의미 체계 검색은 키워드 일치뿐만 아니라 사용자 검색의 의도 및 컨텍스트 의미를 포함하는 쿼리 결과를 반환합니다. 자세한 내용은 자동 의미 체계 보강 정보를 참조하세요. 자동 의미 체계 보강은 다음과 같은 이점을 제공합니다. 간소화된 구현 기계 학습(ML) 전문 지식이나 복잡한 통합이 필요하지 않습니다. 자동화된 프로세스 의미 체계 보강은 데이터 수집 중에 자동으로 수행됩니다. 검색 관련성 개선 의미 체계 보강은 검색 결과의 품질과 컨텍스트 정확도를 향상시킵니다. 확장성 의미 체계 보강은 수동 개입 없이 대규모 데이터 세트에 의미 체계 검색을 적용합니다.	2025년 8월 6일

새로운 기능: Amazon OpenSearch Serverless의 Machine Learning	OpenSearch Serverless는 이제 RAG(검색 증강 생성) 및 의미 체계 검색과 같은 AI 워크플로를 위한 Machine Learning 기능을 지원합니다. 커넥터를 사용하여 타사 플랫폼 또는에서 호스팅되는 ML 알고리즘 및 원격 모델에 액세스합니다 AWS 서비스. 자세한 내용은 Amazon OpenSearch Serverless에서 Machine Learning 구성을 참조하세요. OpenSearch Serverless의 Machine Learning에는 다음 기능이 포함됩니다. 신경 검색 및 하이브리드 검색 보다 관련성이 높은 결과를 얻으려면 쿼리 컨텍스트와 의미를 이해하는 의미 체계 검색 작업을 수행합니다. 키워드 및 의미 체계 검색을 하이브리드 검색과 결합하여 관련성을 개선합니다. 자세한 내용은 OpenSearch Serverless에서 신경 검색 및 하이브리드 검색 구성을 참조하세요. 커넥터 및 모델 타사 ML 플랫폼 및에서 호스팅되는 모델에 연결합니다 AWS 서비스. 모델로 커넥터를 구성하고 배포하여	2025년 8월 6일

	예측 및 AI 워크플로를 실행합니다.
워크플로	여러 API 호출을 연결하여 복잡한 ML 설정 및 사전 처리 작업을 자동화합니다. 워크플로는 OpenSearch에서 AI 애플리케이션 구축을 간소화합니다. 자세한 내용은 OpenSearch Serverless에서 워크플로 구성을 참조하세요.

새로운 기능: OpenSearch Serverless 스냅샷	스냅샷은 재해 복구 기능을 제공하는 Amazon OpenSearch Serverless 컬렉션의 point-in-time 백업입니다. OpenSearch Serverless는 자동으로 컬렉션의 스냅샷을 생성하고 관리하여 비즈니스 연속성과 데이터 보호를 보장합니다. 주요 이점 • 수동 구성이 필요 없는 자동 시간별 백업 • 유지 관리 오버헤드 없음 • 추가 스토리지 비용 없음 • 우발적인 데이터 손실로부터 빠른 복구 • 스냅샷에서 특정 인덱스를 복원할 수 있는 기능 자세한 내용은 스냅샷을 사용하여 컬렉션 백업을 참조하세요.	2025년 7월 30일
OpenSearch Ingestion은 두 개의 새로운 소스를 지원합니다.	OpenSearch Ingestion은 이제 다음 소스를 지원합니다. • Amazon Aurora • Amazon RDS	2025년 7월 17일

새로운 기능: Amazon S3 Vectors와 OpenSearch Service 통합	이 통합은 OpenSearch Service에서 벡터 검색을 크게 향상시키는 다음 기능을 제공합니다. • Amazon S3 Vectors는 벡터 저장 및 쿼리를 기본적으로 지원하는 첫 번째 클라우드 객체 스토어를 제공합니다. S3 벡터는 의미와 유사성에 따라 쿼리할 수 있는 비용 효율적이고 탄력적이며 내구성이 뛰어난 벡터 스토리지를 제공합니다. 1초 미만의 쿼리 응답 시간과 벡터 업로드, 저장 및 쿼리 비용을 최대 90% 절감합니다. • Amazon OpenSearch Service는 Amazon S3를 벡터 인덱스의 벡터 엔진으로 사용할 수 있는 기능을 제공합니다. 이 기능을 사용하면 초 미만의 벡터 검색 기능을 저렴한 비용으로 유지하면서 벡터 데이터를 Amazon S3로 오프로드할 수 있습니다. 자세한 내용은 (미리 보기) Amazon S3 벡터와의 OpenSearch Service 통합을 참조하세요.	2025년 7월 15일

새 주제: OpenSearch Ingestion과 Atlassian 통합	이제 Amazon OpenSearch Ingestion 파이프라인을 사용하여 Atlassian Jira 및 Confluence 서비스의 데이터를 처리할 수 있습니다. 이 통합을 통해 전체 Jira 프로젝트와 Confluence 스페이스를 OpenSearch로 동기화하여 검색 가능한 통합 지식 기반을 생성할 수 있습니다. 통합은 지속적인 모니터링과 업데이트 자동 동기화를 통해 실시간 관련성을 유지하는 동시에 특정 프로젝트, 문제 유형 및 콘텐츠에 대한 유연한 필터링 옵션을 제공합니다. 자세한 내용은 Atlassian Services에서 OpenSearch Ingestion 파이프라인 사용을 참조하세요.	2025년 6월 5일

를 사용한 사용자 지정 플러그인 관리 AWS CLI	이제를 사용하여 Amazon OpenSearch Service용 사용자 지정 플러그인을 관리할 수 있습니다 AWS CLI. 이 기능을 사용하면 간소화된 명령줄 워크플로를 통해 사용자 지정 플러그인을 설치, 업데이트, 보호 및 제거할 수 있습니다. 또한 키를 사용하여 AWS KMS 플러그인을 보호하고 최신 버전의 OpenSearch로 업데이트할 때 플러그인 업그레이드를 관리할 수 있습니다. 자세한 내용은 및 Amazon OpenSearch Service AWS KMS 사용자 지정 패키지 통합을 사용하여 사용자 지정 플러그인 관리를 AWS CLI 참조하세요.	2025년 5월 27일
OpenSearch 2.18 및 2.19 지원	Amazon OpenSearch Service는 OpenSearch 버전 2.18 및 2.19를 지원합니다. 자세한 내용은 OpenSearch 2.19 및 2.18 릴리스 정보를 참조하세요. OpenSearch 2.18 및 2.19 작업에 대한 자세한 내용은 OpenSearch REST API 참조 또는 특정 플러그인에 대한 API 참조를 참조하세요.	2025년 4월 30일

OpenSearch Ingestion에서 파이프라인 역할 지정에 대한 새로운 지원	Amazon OpenSearch Ingestion에서 파이프라인을 생성하거나 업데이트할 때 더 이상 YAML 형식 파이프라인 구성에서 파이프라인 역할을 지정하지 않습니다. 대신 콘솔에서 새 파이프라인 역할 필드에 역할을 지정합니다. 를 사용하는 경우 새 `--pipeline-role-arn` 파라미터를 AWS CLI사용합니다. 자세한 정보는 다음의 주제를 참조하세요. • Amazon OpenSearch Ingestion에서 역할 및 사용자 설정 • Amazon OpenSearch Ingestion 파이프라인 생성 • 자습서: Amazon OpenSearch Ingestion을 사용하여 도메인에 데이터 수집 • 자습서: Amazon OpenSearch Ingestion을 사용하여 컬렉션에 데이터 수집 • CreatePipeline • UpdatePipeline	2025년 4월 22일

새 주제	문제 해결 주제 버전 호환성으로 인해 사용자 지정 플러그인 설치가 실패합니다. 오류 발생 시 취해야 할 단계를 해결합니다`PluginValidationFailureReason : The provided plugin could not be loaded .` 이 오류는 일반적으로 플러그인과 실행 중인 OpenSearch 인스턴스 간의 버전 불일치로 인해 발생합니다.	2025년 4월 21일
새로운 기능: 클러스터 간 검색과의 OpenSearch UI 호환성	OpenSearch UI는 이제 클러스터 간 검색과 호환됩니다. 이를 통해 OpenSearch UI를 하나의 에서 사용하여 다른 리전 AWS 리전 의 클러스터에 액세스할 수 있습니다. 이는 동일한 리전의 클러스터와 연결된 원격 클러스터로 구성하여 수행됩니다. 자세한 내용은 교차 클러스터 검색을 통한 교차 리전 및 교차 계정 데이터 액세스를 참조하세요.	2025년 4월 16일

수정된 콘텐츠: "Amazon OpenSearch Service에서 OpenSearch UI 사용"	확장된 절차, 추가 설정 세부 정보, OpenSearch UI에 대한 Amazon OpenSearch Service 지원의 새로운 릴리스 기록을 포함하여 Amazon OpenSearch Service에서 OpenSearch UI 사용 장을 점검 및 확장했습니다. https://docs.aws.amazon.com/opensearch-service/latest/developerguide/application-release-history.html OpenSearch	2025년 4월 15일
Amazon OpenSearch Service용 Amazon Q Developer	Amazon OpenSearch Service와의 Amazon Q 통합은 다음과 같은 생성 기능을 제공합니다. • 자연어를 사용하여 시각화 생성 • 알림 요약 및 인사이트 보기 • 검색 페이지에서 Amazon Q 생성 쿼리 결과 요약 보기 • 권장 이상 탐지기 보기 • OpenSearch 관련 질문에 대한 Amazon Q 채팅 액세스	2025년 3월 31일
AmazonOpenSearchServiceRolePolicy 관리형 정책이 업데이트되었습니다.	이 정책을 통해 OpenSearch는 OpenSearch에서만 관리하는 애플리케이션의 AWS IAM Identity Center 액세스 범위를 업데이트할 수 있습니다.	2025년 3월 28일

Amazon OpenSearch Service용 OR2 및 OM2 인스턴스	Amazon OpenSearch Service는 이제 OR2 및 OM2 인스턴스를 지원합니다. 내부 벤치마크에서 OR2는 OR1보다 최대 26% 더 나은 인덱싱 처리량을, R7보다 70% 더 나은 인덱싱 처리량을 보여주었습니다. OM2는 OR1보다 최대 15%, M7g보다 66% 더 나은 것으로 나타났습니다.	2025년 3월 25일
CloudWatch Logs 및 Security Lake에 대한 직접 쿼리	Amazon OpenSearch Service는 이제 CloudWatch Logs 및 Security Lake에서 데이터를 쿼리하기 위한 직접 쿼리를 지원합니다.	2024년 12월 1일
k-NN 인덱스	OpenSearch 버전 2.17부터 k-NN 인덱스를 UltraWarm 및 콜드 스토리지 계층으로 이동할 수 있습니다.	2024년 11월 13일
OpenSearch 2.17 지원	Amazon OpenSearch Service는 이제 OpenSearch 버전 2.17을 지원합니다. 이 버전에는 버전 2.16 및 2.17의 일부인 모든 기능이 포함되어 있습니다. 자세한 내용은 2.16 및 2.17 릴리스 정보를 참조하세요.	2024년 11월 13일
OpenSearch 2.15 지원	Amazon OpenSearch Service는 이제 OpenSearch 버전 2.15를 지원합니다. 이 버전에는 버전 2.14 및 2.15의 일부인 모든 기능이 포함되어 있습니다. 자세한 내용은 2.14 및 2.15 릴리스 정보를 참조하세요.	2024년 10월 11일

새 서비스 연결 역할	Amazon OpenSearch Service는 라는 서비스 연결 역할을 추가하여 Amazon OpenSearch IngestionAWSServiceRoleForOpensearchIngestionSelfManagedVpce 이 자체 관리형 VPC 엔드포인트가 있는 파이프라인에 Amazon CloudWatch 대해 지표 데이터를 로 전송할 수 있도록 합니다.	2024년 6월 12일
직접 쿼리	Amazon OpenSearch Service는 이제 OpenSearch 인덱스로 데이터를 수집할 필요 없이 Amazon S3에 저장된 데이터에 대해 직접 쿼리 실행을 지원합니다.	2024년 5월 22일
OpenSearch 2.13 지원	이제 Amazon OpenSearch Service는 OpenSearch 버전 2.13을 지원합니다. 이 버전에는 2.12 및 2.13 버전에 속한 모든 기능이 포함되어 있습니다. 자세한 내용은 2.12 및 2.13 릴리스 정보를 참조하세요.	2024년 5월 21일
Data Prepper 버전 2.7에 대한 Amazon OpenSearch Ingestion 지원	Amazon OpenSearch Ingestion에서 Data Prepper 버전 2.7에 대한 지원을 추가합니다. 자세한 내용은 2.7 release notes를 참조하세요.	2024년 4월 4일

AWS 서비스 OpenSearch Serverless 컬렉션에 대한 프라이빗 액세스	이제 Amazon Bedrock AWS 서비스과 같은 특정에 네트워크 액세스 정책 내에서 OpenSearch Serverless 컬렉션에 대한 액세스 권한을 부여할 수 있습니다.	2024년 3월 28일
인플레이스 EBS 업데이트	이제 Amazon OpenSearch Service에서 블루/그린 배포 없이 도메인에 대한 일부 EBS 변경을 수행할 수 있습니다.	2024년 2월 14일
구성 변경 가시성	이제 Amazon OpenSearch Service 콘솔에서 그리고 구성 API를 사용하여 도메인 구성 변경을 추적할 수 있습니다.	2024년 2월 6일
벡터 검색 컬렉션 정식 출시	Amazon OpenSearch Serverless 벡터 검색 컬렉션은 이제 정식 버전으로 제공됩니다. 미리 보기 단계에서 다음과 같은 중요한 개선이 이루어졌습니다. • 벡터 검색 컬렉션은 이제 각각 최대 128개의 차원을 가진 수십억 개의 벡터로 구성된 워크로드를 지원합니다. • OpenSearch 대시보드는 이제 벡터 검색 컬렉션을 지원합니다.	2023년 11월 29일
OR1 인스턴스	이제 Amazon OpenSearch Service는 OR1 인스턴스 유형을 지원합니다.	2023년 11월 29일

Amazon S3와의 직접 쿼리(평가판)	직접 쿼리는 트랜잭션 데이터가 Amazon S3 버킷에 기록된 후 몇 초 안에 Amazon OpenSearch Service에서 사용 가능하도록 만들기 위한 완전 관리형 솔루션을 제공합니다.	2023년 11월 29일
시계열 수집을 위한 10TiB 용량	Amazon OpenSearch Serverless는 시계열 수집을 위해 최대 10TiB의 인덱스 데이터에 대한 지원을 추가합니다. 또한 이 릴리스는 모든 유형의 컬렉션에 대해 200개 OCU의 최대 허용 용량을 지원하며 컬렉션을 생성할 때 대기 복제본을 비활성화할 수 있는 기능도 지원합니다.	2023년 11월 29일
OpenSearch 2.11 지원	이제 Amazon OpenSearch Service는 OpenSearch 버전 2.11을 지원합니다. 이 버전에는 2.10 및 2.11 버전에 속했던 모든 기능이 포함되어 있습니다. 자세한 내용은 2.10 및 2.11 릴리스 정보를 참조하세요.	2023년 11월 17일
Data Prepper 버전 2.6에 대한 Amazon OpenSearch Ingestion 지원	Amazon OpenSearch Ingestion에서 Data Prepper 버전 2.6에 대한 지원을 추가합니다. 자세한 내용은 2.6 release notes를 참조하세요. 또한 Amazon DynamoDB를 파이프라인 소스로 지정할 수 있습니다. 자세한 내용은 Amazon DynamoDB와 함께 OpenSearch Ingestion 파이프라인 사용을 참조하세요.	2023년 11월 17일

Data Prepper 버전 2.5에 대한 Amazon OpenSearch Ingestion 지원	Amazon OpenSearch Ingestion에서 Data Prepper 버전 2.5에 대한 지원을 추가합니다. 자세한 내용은 2.5 릴리스 정보를 참조하세요. 또한 이제 OpenSearch Service 도메인 또는 OpenSearch Serverless 컬렉션을 파이프라인 소스로 지정할 수 있습니다. 자세한 내용은 Data Prepper 설명서의 OpenSearch 소스 플러그인을 참조하세요.	2023년 11월 17일
원격 추론을 위한 CloudFormation 템플릿	의미 체계 검색을 위한 원격 추론 설정을 용이하게 하기 위해 Amazon OpenSearch Service는 콘솔에서 모델 프로비저닝 프로세스를 자동화하는 AWS CloudFormation 템플릿을 제공합니다.	2023년 11월 7일
서비스 연결 역할 정책 업데이트	서비스 연결 역할 정책 `AmazonOpenSearchServiceRolePolicy` 에서 IPv6 주소를 할당하고 할당 취소하는 데 필요한 권한을 추가합니다. 더 이상 사용되지 않는 Elasticsearch 정책 `AmazonElasticsearchServiceRolePolicy` 도 이전 버전과의 호환성을 보장하기 위해 업데이트되었습니다.	2023년 10월 26일

Amazon OpenSearch Serverless 수명 주기 정책	Amazon OpenSearch Serverless는 인덱스 수명 주기 정책을 도입하여 데이터 보존 및 삭제 관리를 간소화합니다. 이제 콘솔의 API 또는 구성 인터페이스를 사용하여 시계열 수집에 대한 데이터 보존 정책을 설정할 수 있으므로 오래된 데이터를 삭제하기 위해 일별 인덱스나 스크립트를 생성할 필요가 없습니다.	2023년 10월 25일
Im4gn 인스턴스 지원	이제 Amazon OpenSearch Service는 Im4gn 인스턴스 유형을 지원합니다. IM4gn 인스턴스는 대규모 데이터 세트를 관리하고 vCPU당 높은 스토리지 밀도를 필요로 하는 워크로드에 최적화되어 있습니다.	2023년 10월 20일
관리 옵션	Amazon OpenSearch Service는 이제 도메인 관련 문제를 해결해야 하는 경우 세분화된 제어를 제공하는 여러 관리 옵션을 제공합니다. 이러한 옵션에는 데이터 노드에서 OpenSearch 프로세스를 다시 시작하는 기능과 데이터 노드를 다시 시작하는 기능이 포함됩니다.	2023년 10월 17일

옵션 플러그인	Amazon OpenSearch Service는 네 가지 새로운 언어 분석기 플러그인, 즉 Nori(한국어), Sudachi(일본어), Pinyin(중국어) 및 STConvert 분석(중국어)과 함께 Amazon Personalize 검색 순위 플러그인에 대한 지원을 추가합니다.	2023년 10월 16일
OpenSearch 2.9 지원	이제 Amazon OpenSearch Service는 OpenSearch 버전 2.9를 지원합니다. 이 버전에는 2.8 및 2.9 버전에 속했던 모든 기능이 포함되어 있습니다. 자세한 내용은 2.8 및 2.9 릴리스 정보를 참조하세요.	2023년 10월 2일
ML 커넥터	Amazon OpenSearch Service는 기계 학습(ML) 커넥터에 대한 지원을 추가합니다. 커넥터를 사용하면 다른 AWS 서비스 또는 타사 기계 학습(ML) 플랫폼에서 호스팅되는 ML 모델에 쉽게 액세스할 수 있습니다.	2023년 9월 6일
Amazon OpenSearch Ingestion에서 Data Prepper 버전 2.4에 대한 지원 추가	Amazon OpenSearch Ingestion에서 Data Prepper 버전 2.4에 대한 지원을 추가합니다. 자세한 내용은 2.4 릴리스 정보를 참조하세요. 또한 이제 Amazon Managed Streaming for Apache Kafka (Amazon MSK)를 파이프라인 소스로 지정할 수 있습니다.	2023년 8월 31일

시계열 수집을 위한 6TiB 용량	Amazon OpenSearch Serverless는 시계열 수집을 위해 최대 6TiB의 인덱스 데이터에 대한 지원을 추가합니다. 또한 이번 릴리스는 검색 및 시계열 수집 모두에 대해 최대 허용 용량 100OCU를 지원합니다.	2023년 8월 15일
벡터 검색 수집	Amazon OpenSearch Serverless에는 벡터 검색 수집을 생성하는 옵션이 추가되며, 이 수집을 사용하여 벡터 임베딩을 저장하여 유사성 및 시맨틱 검색을 강화할 수 있습니다.	2023년 7월 26일
OpenSearch 2.7 지원	이제 Amazon OpenSearch Service는 OpenSearch 버전 2.7을 지원합니다. 이 버전에는 2.6 및 2.7 버전에 속했던 모든 기능이 포함되어 있습니다. 자세한 내용은 2.6 및 2.7 릴리스 정보를 참조하세요.	2023년 7월 10일
Data Prepper 2.3 지원	Amazon OpenSearch Ingestion에서 Data Prepper 버전 2.3에 대한 지원을 추가합니다. 자세한 내용은 2.3 릴리스 정보를 참조하세요. 또한 이제 Amazon Security Lake를 파이프라인 소스로 지정할 수 있습니다.	2023년 6월 26일

Multi-AZ with Standby	Amazon OpenSearch Service는 세 개의 가용 영역(AZ)에 도메인을 배포할 수 있는 옵션을 추가하며, 각 AZ에는 전체 데이터 사본이 포함되고 이러한 AZ 중 하나에 있는 노드가 예비 복제본으로 작동합니다. Multi-AZ with Standby 배포 옵션은 인프라 장애 발생 시 99.99%의 가용성과 일관된 성능을 제공합니다.	2023년 5월 3일
새 서비스 연결 역할	Amazon OpenSearch Service는 `AWSServiceRoleForAmazonOpenSearchIngestionService` 이라는 서비스 연결 역할을 추가하여 Amazon OpenSearch Ingestion이 지표 데이터를 Amazon CloudWatch에 전송할 수 있도록 합니다.	2023년 4월 26일
Amazon OpenSearch Ingestion	Amazon OpenSearch Ingestion은 OpenSearch Service 도메인과 OpenSearch Serverless 컬렉션에 실시간 로그 및 추적 데이터를 제공하는 완전 관리형 데이터 수집기입니다. OpenSearch Ingestion을 사용하면 Logstash 또는 Jaeger와 같은 타사 솔루션을 사용하여 도메인과 컬렉션으로 데이터를 수집할 필요가 없습니다.	2023년 4월 26일

OpenSearch 2.5 지원	이제 Amazon OpenSearch Service는 OpenSearch 버전 2.5를 지원합니다. 이 버전에는 2.4 및 2.5 버전에 속했던 모든 기능이 포함되어 있습니다. 자세한 내용은 2.4 및 2.5 릴리스 정보를 참조하세요.	2023년 3월 13일
사용량이 적은 유지 관리 기간	Amazon OpenSearch Service는 트래픽이 적은 일일 10시간의 시간대인 사용량이 적은 기간을 추가하여 블루/그린 배포가 필요한 서비스 소프트웨어 업데이트 및 자동 조정 최적화를 예약할 수 있습니다. 비수기 업데이트는 트래픽이 많은 기간 동안 클러스터의 전용 프라이머리 노드에 가해지는 부담을 최소화하는 데 도움이 됩니다. 2월 16일 이후에 생성된 새 도메인의 경우 비수기 기간은 현지 시간으로 오후 10시~오전 8시로 자동 구성됩니다. 기존 도메인의 경우 기간을 명시적으로 활성화해야 합니다.	2023년 2월 16일
도메인 생성 중 SAML 인증 구성	Amazon OpenSearch Service는 이제 도메인 생성 중 SAML 인증 구성을 지원합니다. 이전에는 도메인이 이미 생성된 후에 SAML 옵션을 구성해야 했습니다.	2023년 2월 1일

VPC 도메인에 대한 원격 재인덱스	Amazon OpenSearch Service는 두 도메인 간의 VPC 엔드포인트 연결 옵션을 추가합니다. 이제 원격 재인덱스를 사용하여 역방향 프록시 없이 VPC 도메인 간에 인덱스를 복사할 수 있습니다. 이 기능을 사용하려면 VPC 도메인에서 서비스 소프트웨어 R20221114 이상을 실행해야 합니다.	2023년 1월 31일
Amazon OpenSearch Serverless 정식 출시	이제 Amazon OpenSearch Serverless가 정식 출시되었습니다. 미리 보기 단계에서 다음과 같은 중요한 개선이 이루어졌습니다. • 이제 컬렉션 엔드포인트에서 트래픽이 감소하면 용량을 구성된 최소 OCU로 스케일 다운할 수 있습니다. • 인덱싱과 검색 모두에 허용되는 최대 OCU가 20개에서 50개로 늘어났습니다. 각 OCU에는 120GiB의 인덱스 데이터를 위한 충분한 핫 임시 스토리지가 포함되어 있습니다. • 이제 데이터 액세스 설정을 별도의 워크플로에서 구성할 필요 없이 컬렉션을 생성하는 동안 구성할 수 있습니다.	2023년 1월 25일

비동기식 모의 실습	이제 Amazon OpenSearch Service는 구성을 변경하기 전에 검증 확인을 수행할 수 있게 하고 변경 사항으로 인해 블루/그린 배포가 발생하는지 알려주는 비동기식 모의 실습을 지원합니다.	2023년 1월 19일
새 서비스 연결 역할	Amazon OpenSearch Service는 `AWSServiceRoleForAmazonOpenSearchServerless` 이라는 서비스 연결 역할을 추가하여 OpenSearch Serverless가 지표 데이터를 Amazon CloudWatch에 전송할 수 있도록 합니다.	2022년 11월 29일
Amazon OpenSearch Serverless 미리 보기	Amazon OpenSearch Serverless는 Amazon OpenSearch Service를 위한 온디맨드 Auto Scaling 서버리스 구성입니다. 서버리스는 OpenSearch 클러스터를 프로비저닝, 구성, 조정하는 데 따르는 운영상의 복잡성을 제거합니다.	2022년 11월 29일

OpenSearch 2.3 지원	이제 Amazon OpenSearch Service는 OpenSearch 버전 2.3을 지원합니다. 이 버전에는 2.0, 2.1, 2.2 버전에 속했던 모든 기능이 포함되어 있습니다. 자세한 내용은 2.0, 2.1, 2.2, 2.3 릴리스 노트를 참조하세요. 버전 2.3에는 주요 변경 사항이 포함되어 있습니다. 자세한 내용은 지원되는 업그레이드 경로를 참조하세요.	2022년 11월 15일
알림 플러그인 지원	이제 Amazon OpenSearch Service는 OpenSearch 플러그인의 모든 알림에 대한 중앙 위치를 제공하는 알림 플러그인을 지원합니다. 버전 2.0부터 알림 대상은 더 이상 사용되지 않으며 알림 채널로 대체되었습니다.	2022년 11월 15일
Kibana 7.1.1 지원	이제 Elasticsearch 7.1을 실행하는 Amazon OpenSearch Service 도메인은 버그 수정을 추가하고 보안을 개선하는 Kibana 7.1.1용 최신 패치 릴리스를 지원합니다. 7.1 도메인을 서비스 소프트웨어 R20221114로 업데이트하면 OpenSearch Service에서 자동으로 이 패치 릴리스로 업그레이드합니다.	2022년 11월 15일

Kibana 6.8.13 지원	이제 Elasticsearch 6.8을 실행하는 Amazon OpenSearch Service 도메인은 버그 수정을 추가하고 보안을 개선하는 Kibana 6.8.13용 최신 패치 릴리스를 지원합니다. 6.8 도메인을 서비스 소프트웨어 R20221114로 업데이트하면 OpenSearch Service에서 자동으로 이 패치 릴리스로 업그레이드합니다.	2022년 11월 15일
Kibana 6.3.2 지원	이제 Elasticsearch 6.3을 실행하는 Amazon OpenSearch Service 도메인은 버그 수정을 추가하고 보안을 개선하는 Kibana 6.3.2용 최신 패치 릴리스를 지원합니다. 6.3 도메인을 서비스 소프트웨어 R20221114로 업데이트하면 OpenSearch Service에서 자동으로 이 패치 릴리스로 업그레이드합니다.	2022년 11월 15일

AWS PrivateLink	Amazon OpenSearch Service 관리형 VPC 엔드포인트를 사용하면 인터넷을 통해 연결하는 대신 인터페이스 VPC 엔드포인트를 사용하여 OpenSearch Service VPC 도메인에 직접 연결할 수 있습니다. OpenSearch Service 관리형 VPC 엔드포인트는 엔드포인트가 프로비저닝된 VPC 내에서만 또는 라우팅 테이블 및 보안 그룹에서 허용하는 대로 엔드포인트가 프로비저닝된 VPC와 피어링된 모든 VPC에서 액세스할 수 있습니다. 인터페이스 VPC 엔드포인트에 연결하려면 VPC 도메인에서 서비스 소프트웨어 R20220928 또는 이후 버전을 실행해야 합니다.	2022년 11월 7일
버그 수정 및 성능 향상	서비스 소프트웨어 R20220928에는 향상된 SAML 로깅을 비롯한 버그 수정 및 성능 향상 기능이 포함되어 있습니다. 또한 이 업데이트는 기본 테넌트를 `Private`이 아닌 `Global`로 변경합니다.	2022년 10월 3일
개선된 API 참조	Amazon OpenSearch Service는 모든 것을 아우르는 개선된 구성 API 참조를 제공합니다. 새 참조에는 모든 사용 가능한 작업 및 데이터 유형, 샘플 요청 및 응답 구문, 지원되는 모든 언어에 대한 해당 SDK 참조 링크가 포함되어 있습니다.	2022년 9월 13일

블루/그린 검증	이제 Amazon OpenSearch Service는 블루/그린 배포 전에 검증을 수행하고 도메인을 업데이트할 수 없는 경우 검증 오류를 표시합니다.	2022년 8월 16일
OpenSearch 1.3 지원	Amazon OpenSearch Service는 이제 OpenSearch 버전 1.3을 지원합니다. 자세한 내용은 1.3 릴리스 정보를 참조하세요.	2022년 7월 27일
ML Commons 플러그인 지원	Amazon OpenSearch Service는 전송 및 REST API 호출을 통해 공통 기계 학습 알고리즘 세트를 제공하는 ML Commons 플러그인에 대한 지원을 추가합니다. PPL 명령을 통해 ML Commons 플러그인과 상호 작용할 수도 있습니다.	2022년 7월 27일
gp3 볼륨 지원	Amazon OpenSearch Service는 gp3 EBS 범용 SSD 볼륨 유형에 대한 지원을 추가합니다. 도메인을 생성하거나 수정할 때 추가 프로비저닝된 IOPS 및 처리량(Throughput)을 지정할 수 있습니다.	2022년 7월 26일
향상된 모범 사례 문서	Amazon OpenSearch Service 문서는 OpenSearch Service 도메인 생성과 운영에 대한 향상된 운영 모범 사례 및 일반 권장 사항을 제공합니다.	2022년 7월 6일

Service Quotas와 통합	이제 Service Quotas 콘솔에서 Amazon OpenSearch Service에 대한 할당량을 확인하고 할당량을 증가하도록 요청할 수 있습니다.	2022년 6월 29일
OpenSearch API에 대한 태그 기반 액세스 제어	이제 태그를 사용하여 OpenSearch API에 대한 액세스를 제어할 수 있습니다. 이전에는 구성 API에 대한 액세스를 제어하는 데만 태그를 사용할 수 있었습니다.	2022년 6월 16일
리전 간 클러스터 간 검색	이제 두 도메인이 모두 Elasticsearch 버전 7.10 이상 또는 모든 버전의 OpenSearch를 실행하는 AWS 리전 한에서 클러스터 간 검색이 지원됩니다.	2022년 6월 14일
단일 Kibana 5.6 지원	Amazon OpenSearch Service는 단일 Kibana 5.6.16에 대한 지원을 추가합니다. 단일 Kibana 5.6.16을 사용하면 Elasticsearch 버전 5.1, 5.3, 5.5 및 5.6 버전에 연결하는 동안 Kibana 5.6을 프런트 엔드로 사용할 수 있습니다. 단일 Kibana 5.6을 사용하려면 서비스 소프트웨어 R20220323 이상이어야 합니다.	2022년 4월 4일

R20220323-P1	Amazon OpenSearch Service에서 최근에 서비스 소프트웨어 업데이트 R20220323을 릴리스했지만 문제가 발생하여 업데이트가 롤백되었습니다. 도메인을 패치 릴리스 R20220323-P1 이상으로 업데이트하여 문제를 해결하는 것이 좋습니다.	2022년 4월 4일
OpenSearch 1.2 지원	Amazon OpenSearch Service는 이제 OpenSearch 버전 1.2를 지원합니다. 자세한 내용은 1.2 릴리스 정보를 참조하세요.	2022년 4월 4일
Observability	Amazon OpenSearch Service용 OpenSearch 대시보드의 기본 설치에는 Observability 플러그인이 포함되어 있습니다. 이 플러그인은 데이터를 탐색 및 쿼리하기 위해 파이프 처리 언어(PPL)를 사용하여 데이터 기반 이벤트를 시각화하는 데 사용할 수 있습니다. 플러그인에는 OpenSearch 1.2 이상, 서비스 소프트웨어 R20220323 이상이 필요합니다.	2022년 4월 4일

Kibana 7.7.1 지원	이제 Elasticsearch 버전 7.7을 실행하는 Amazon OpenSearch Service 도메인에서 Kibana 7.7에 대한 최신 패치 릴리스를 지원하므로 버그 수정이 추가되고 보안이 향상됩니다. 7.7 도메인을 서비스 소프트웨어 R20220323 이상으로 업데이트하면 OpenSearch Service에서 자동으로 이 패치 릴리스로 업그레이드합니다.	2022년 4월 4일
JVM 메모리 압력 지표 변경	Amazon OpenSearch Service에서는 메모리 사용률을 더욱 정확하게 반영하기 위해 `JVMMemoryPressure` CloudWatch 지표에 대한 로직을 변경했습니다. 이전에는 지표가 JVM 힙의 전 세대 메모리 풀만 고려했습니다. 이번 변경으로 지표가 최신 세대 메모리 풀도 고려합니다. 도메인을 서비스 소프트웨어 R20220323으로 업데이트하면 `JVMMemoryPressure` ,`MasterJVMMemoryPressure` 및/또는`WarmJVMMemoryPressure` 지표가 증가할 수 있습니다.	2022년 4월 4일
IK (Chinese) Analysis 플러그인이 포함된 사용자 지정 사전	Amazon OpenSearch Service는 이제 IK (Chinese) Analysis 플러그인과 함께 사용자 지정 사전을 사용할 수 있도록 지원합니다.	2022년 4월 4일

기존 도메인에 대한 클러스터 간 복제	Amazon OpenSearch Service에서는 2020년 6월 3일 이후에 생성된 도메인에 대해서만 클러스터 간 검색 및 클러스터 간 복제를 구현할 수 있다는 제한을 삭제했습니다. 이제 생성 시기와 관계없이 모든 도메인에서 이러한 기능을 활성화할 수 있습니다. 두 도메인 모두 서비스 소프트웨어 R20220323 이상이어야 합니다.	2022년 4월 4일
블루/그린 배포 가시성	이제 Amazon OpenSearch Service에서 블루/그린 배포의 진행 상황을 더욱 잘 파악할 수 있습니다. 콘솔에서 또는 구성 API를 사용하여 이러한 세부 정보를 모니터링할 수 있습니다.	2022년 1월 27일
기존 도메인에서의 세분화된 액세스 제어	이제 기존 도메인에서 세분화된 액세스 제어를 사용 설정할 수 있습니다. Open/IP 기반 액세스 정책에 대해 임시 마이그레이션 기간을 사용하도록 설정하여 역할을 생성하고 매핑하는 동안 사용자가 도메인에 계속 액세스하도록 할 수 있습니다. 기존 도메인에서의 세분화된 액세스 제어를 사용 설정하려면 R20211203 이상의 서비스 소프트웨어가 필요합니다.	2022년 1월 6일

이름 변경된 OpenSearch 대시보드 역할	서비스 소프트웨어 R20211203에서 `kibana_user` 역할이 `opensearch_dashboards_user`(으)로 이름이 변경되었으며 `kibana_read_only` 이(가) `opensearch_dashboards_read_only`(으)로 이름이 변경이 되었습니다. 이 변경 사항은 모든 새로 생성된 OpenSearch 1.x 도메인에 적용됩니다. 서비스 소프트웨어 R20211203으로 업그레이드하는 기존 OpenSearch 도메인의 경우 역할은 동일하게 유지됩니다.	2022년 1월 4일
OpenSearch 1.1 지원	Amazon OpenSearch Service는 이제 OpenSearch 버전 1.1을 지원합니다. 자세한 내용은 1.1 릴리스 정보를 참조하세요.	2022년 1월 4일
ISM 시각적 편집기	Amazon OpenSearch Service용 OpenSearch 대시보드의 기본 설치에서는 이제 ISM 정책에 대한 시각적 편집기를 지원합니다. 이 기능을 사용하려면 OpenSearch 1.1 이상이 필요합니다.	2022년 1월 4일

교차 서비스 혼동된 대리자 예방 업데이트	Amazon OpenSearch Service는 혼동된 대리자 문제를 방지하기 위해 IAM 리소스 정책에서 `aws:SourceArn` 및 `aws:SourceAccount` 글로벌 조건 컨텍스트 키 사용을 지원합니다. 이러한 조건 키를 추가하려면 도메인이 서비스 소프트웨어 R20211203 이상이어야 합니다.	2022년 1월 4일
Log4j 패치	서비스 소프트웨어 R20211203-P2는 CVE-2021-44228 및 CVE-2021-45046의 권고에 따라 OpenSearch Service에 사용되는 Log4j 버전을 업데이트합니다. 이 패치는 모든 버전의 OpenSearch 및 Elasticsearch를 실행하는 도메인에 적용됩니다. OpenSearch Service는 내부적으로 다양한 Log4j 버전을 계속 업데이트하며 반드시 최신 버전의 Log4j로 제한되는 것은 아닙니다. 도메인의 Log4j 버전은 도메인이 실행 중인 소프트웨어 버전에 따라 다릅니다. 단, Log4j 버전과 관계없이 R20211203-P2 이상을 실행하는 경우 도메인에 CVE-2021-44228 및 CVE-2021-45046 주소를 지정하는 데 필요한 Log4j 업데이트가 포함되어 있습니다.	2021년 12월 15일

클러스터 간 복제	클러스터 간 복제를 사용하면 특정 OpenSearch 서비스 도메인에서 다른 도메인으로 인덱스, 매핑 및 메타데이터를 복제할 수 있습니다. 클러스터 간 복제는 Elasticsearch 7.10 또는 OpenSearch 1.1 이상을 실행하는 도메인이 필요합니다.	2021년 10월 5일
새로운 AWS관리형 정책	Amazon OpenSearch Service의 출시에는 새로운 AWS관리형 정책과 이전 정책의 사용 중단이 포함됩니다.	2021년 9월 8일
Kibana 6.4.3 지원	이제 Elasticsearch 버전 6.4를 실행하는 레거시 Amazon OpenSearch Service 도메인에서 Kibana 6.4에 대한 최신 패치 릴리스를 지원하므로 버그 수정이 추가되고 보안이 향상됩니다. OpenSearch Service는 도메인을 이 패치 릴리스로 자동 업그레이드합니다.	2021년 9월 8일
데이터 스트림	Amazon OpenSearch Service에 데이터 스트림에 대한 지원이 추가되어 시계열 데이터 관리 프로세스를 단순화합니다. 데이터 스트림을 사용하려면 도메인이 OpenSearch 1.0 이상을 실행해야 합니다.	2021년 9월 8일

Amazon OpenSearch Service	AWS 는 Amazon OpenSearch Service의 이름을 변경하여 레거시 "Elasticsearch" 브랜딩을 제거합니다. Amazon OpenSearch Service는 OpenSearch 및 레거시 Elasticsearch OSS를 지원합니다. 클러스터를 생성할 때 어떤 검색 엔진을 사용할지 선택할 수 있습니다. OpenSearch Service는 소프트웨어의 최종 오픈 소스 버전인 Elasticsearch OSS 7.10과 광범위한 호환성을 제공합니다.	2021년 9월 8일
콜드 스토리지	콜드 스토리지는 자주 액세스하지 않는 데이터 또는 기록 데이터를 위한 새로운 스토리지 계층입니다. 콜드 인덱스는 S3 스토리지만 차지하며 연결된 계산이 없습니다. 콜드 스토리지에는 Elasticsearch 7.9 이상을 실행하는 도메인과 서비스 소프트웨어 R20210426 이상이 필요합니다.	2021년 5월 13일
ARM 기반 Graviton 인스턴스	이제 Amazon OpenSearch Service에서 ARM 기반 Graviton 인스턴스 유형(M6G, C6G, R6G, R6GD)을 지원합니다. Graviton 인스턴스 유형은 Elasticsearch 7.9 이상을 실행하는 신규 및 기존 도메인 및 서비스 소프트웨어 R20210331 이상에서 사용할 수 있습니다.	2021년 5월 4일

ISM 템플릿	Amazon OpenSearch Service에 ISM 템플릿에 대한 지원이 추가되어 인덱스가 정책에 정의된 패턴과 일치하는 경우 인덱스에 ISM 정책을 자동으로 연결할 수 있습니다. ISM 템플릿에는 서비스 소프트웨어 R20210426 이상이 필요합니다. 이 업데이트는 또한 `policy_id` 설정을 사용 중지하므로, 새로 생성된 인덱스에 ISM 정책을 적용하기 위해 더 이상 인덱스 템플릿을 사용할 수 없습니다. 이 업데이트로 인해 이 설정을 사용한 기존 CloudFormation 템플릿에 대한 주요 변경이 발생할 수 있습니다.	2021년 4월 27일
Elasticsearch 7.10 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 7.10을 지원합니다. 자세한 내용은 7.10 릴리스 정보를 참조하세요.	2021년 4월 21일
비동기 검색	이제 Amazon OpenSearch Service는 비동기 검색을 지원하므로, 백그라운드에서 검색 요청을 실행할 수 있습니다. 비동기 검색에는 Elasticsearch 7.10 이상을 실행하는 도메인과 서비스 소프트웨어 R20210331 이상이 필요합니다.	2021년 4월 21일

구성 API에 대한 태그 기반 액세스 제어	이제 AWS 태그를 사용하여 Amazon ES 구성 API에 대한 액세스를 제어할 수 있습니다.	2021년 3월 2일
자동 조정	Amazon OpenSearch Service에 자동 조정이 추가됩니다. 이 기능은 클러스터의 성능 및 사용량 지표를 사용하여 노드의 JVM 설정에 대한 변경 사항을 제안합니다. 자동 조정에는 Elasticsearch 6.7 이상을 실행하는 도메인과 서비스 소프트웨어 R20201117 이상이 필요합니다.	2021년 2월 24일
Trace Analytics	이제 Amazon OpenSearch Service용 Kibana의 기본 설치에 추적 분석 플러그인이 포함되어 분산 애플리케이션의 추적 데이터를 모니터링할 수 있습니다. 플러그인에는 Elasticsearch 7.9 이상을 실행하는 도메인과 서비스 소프트웨어 R20210201 이상이 필요합니다.	2021년 2월 17일

샤드 지표	샤드 상태(`Shards.active` , `Shards.unassigned` , `Shards.delayedUnassigned` `Shards.activePrimary` , `Shards.initializing` , `Shards.relocating`) 추적을 위해 Amazon OpenSearch Service에 다음 CloudWatch 지표가 추가됩니다. 지표는 서비스 소프트웨어 R20210201 이상이 설치된 도메인에서 사용할 수 있습니다.	2021년 2월 17일
Kibana 보고서	이제 Amazon OpenSearch Service용 Kibana의 기본 설치에서 검색, 시각화 및 대시보드 페이지에 대한 온디맨드 보고서를 지원합니다. 이 기능을 사용하려면 Elasticsearch 7.9 이상 및 서비스 소프트웨어 R20210201 이상이 필요합니다.	2021년 2월 17일
Kibana 5.6.16 지원	이제 Elasticsearch 버전 5.6을 실행하는 Amazon OpenSearch Service 도메인에서 Kibana 5.6에 대한 최신 패치 릴리스를 지원하므로 버그 수정이 추가되고 보안이 향상됩니다. Amazon ES는 도메인을 이 패치 릴리스로 자동 업그레이드합니다.	2021년 2월 17일

기존 도메인에 대한 암호화	이제 Amazon OpenSearch Service는 Elasticsearch 6.7 또는 이후 버전을 실행하는 기존 도메인에서 저장 데이터의 암호화 및 노드 간 암호화 활성화를 지원합니다. 이러한 설정을 활성화한 후에는 비활성화할 수 없습니다.	2021년 1월 27일
원격 재인덱스	이제 Amazon OpenSearch Service는 원격 재인덱스를 지원하므로, 원격 도메인에서 인덱스를 마이그레이션할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201117 이상이 필요합니다.	2020년 11월 24일
파이프 처리 언어	이제 Amazon OpenSearch Service는 파이프(|) 구문을 사용하여 Elasticsearch에 저장된 데이터를 쿼리할 수 있는 쿼리 언어인 파이프 처리 언어(PPL)를 지원합니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201117 이상이 필요합니다. 자세한 내용은 다음 섹션을 참조하세요.	2020년 11월 24일
Kibana 노트북	Amazon OpenSearch Service에 Kibana 노트북에 대한 지원이 추가되어 라이브 시각화와 설명 텍스트를 단일 인터페이스에서 결합할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201117 이상이 필요합니다.	2020년 11월 24일

Gantt 차트	이제 Amazon OpenSearch Service용 Kibana의 기본 설치에서 새로운 시각화 유형인 Gantt 차트를 지원합니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201117 이상이 필요합니다.	2020년 11월 24일
Elasticsearch 7.9 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 7.9를 지원합니다. 자세한 내용은 7.9 릴리스 정보를 참조하세요.	2020년 11월 24일
이상 탐지 업데이트	Amazon OpenSearch Service의 이상 탐지에 높은 카디널리티에 대한 지원이 추가되어 IP 주소, 제품 ID, 국가 코드 등과 같은 차원으로 이상 항목을 분류할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201117 이상이 필요합니다.	2020년 11월 24일
동적 사전 업데이트	이제 Amazon OpenSearch Service에서 재인덱싱하지 않고도 검색 분석기를 업데이트할 수 있습니다. 일부 또는 모든 도메인의 사전 파일을 업데이트할 수 있으며, Amazon ES에서 시간에 따라 패키지 버전을 추적하므로 변경된 내용과 시기에 대한 기록을 확인할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201019 이상이 필요합니다.	2020년 11월 17일

사용자 지정 엔드포인트	이제 Amazon OpenSearch Service는 사용자 지정 엔드포인트를 지원하므로 Amazon ES 도메인에 새 URL을 제공할 수 있습니다. 도메인을 스왑하는 경우 동일한 URL을 유지할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201019 이상이 필요합니다.	2020년 11월 5일
새로운 언어 플러그인	이제 Amazon OpenSearch Service는 서비스 소프트웨어 R20201019 또는 이후 버전을 통해 Elasticsearch 7.7 또는 이후 버전을 실행하는 도메인에서 IK(중국어) 분석, 베트남어 분석, 태국어 분석 플러그인을 지원합니다.	2020년 10월 28일
Elasticsearch 7.8 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 7.8을 지원합니다. 자세한 내용은 7.8 릴리스 정보를 참조하세요.	2020년 10월 28일
Kibana에 대한 SAML 인증	이제 Amazon OpenSearch Service는 Kibana에 대한 SAML 인증을 지원하므로, 타사 자격 증명 공급자를 사용하여 Kibana에 로그인하고 세분화된 액세스 제어를 관리하고 데이터를 검색하고 시각적 객체를 구축할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20201019 이상이 필요합니다.	2020년 10월 27일

T3 인스턴스	이제 Amazon OpenSearch Service는 `t3.small` 및 `t3.medium` 인스턴스 유형을 지원합니다.	2020년 9월 23일
감사 로그	이제 Amazon OpenSearch Service는 데이터의 감사 로그를 지원하므로 실패한 로그인 시도, 인덱스, 문서 및 필드에 대한 사용자 액세스 등을 추적할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20200910 이상이 필요합니다.	2020년 9월 16일
UltraWarm 업데이트	Amazon OpenSearch Service용 UltraWarm에는 새로운 지표, 새로운 설정, 더 큰 마이그레이션 대기열, 취소 API가 추가됩니다. 이러한 업데이트에는 서비스 소프트웨어 R20200910 이상이 필요합니다. 자세한 내용은 다음 섹션을 참조하세요.	2020년 9월 14일
순위 학습	이제 Amazon OpenSearch Service는 오픈 소스 순위 학습 플러그인을 지원하므로, 기계 학습 기술을 사용하여 검색 관련성을 개선할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20200721 이상이 필요합니다.	2020년 7월 27일

k-NN 코사인 유사도	이제 k-Nearest Neighbor (k-NN)에서 유클리드 거리 외에 코사인 유사도로 “가장 가까운 이웃”을 검색할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20200721 이상이 필요합니다.	2020년 7월 23일
gzip 압축	이제 Amazon OpenSearch Service는 대부분의 HTTP 요청 및 응답에 대한 gzip 압축을 지원하므로 대기 시간을 줄이고 대역폭을 절약할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20200721 이상이 필요합니다.	2020년 7월 23일
Elasticsearch 7.7 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 7.7을 지원합니다. 자세한 내용은 7.7 릴리스 정보를 참조하세요.	2020년 7월 23일
Kibana 맵 서비스	이제 Amazon OpenSearch Service용 Kibana의 기본 설치에 인도 및 중국 리전의 도메인을 제외한 WMS 맵 서버가 포함됩니다.	2020년 6월 18일
SQL 개선 사항	이제 Amazon OpenSearch Service에 대한 SQL 지원은 많은 새로운 작업, 데이터 탐색을 위한 전용 Kibana 사용자 인터페이스, 대화형 CLI를 지원합니다.	2020년 6월 3일

클러스터 간 검색	Amazon OpenSearch Service를 사용하면 연결된 여러 도메인에서 클러스터 간 쿼리 및 집계를 수행할 수 있습니다.	2020년 6월 3일
이상 탐지	Amazon OpenSearch Service를 사용하면 거의 실시간으로 이상을 자동 탐지할 수 있습니다.	2020년 6월 3일
UltraWarm	Amazon OpenSearch Service용 UltraWarm 스토리지는 공개 미리 보기 단계를 지나 현재 정식 버전으로 출시되어 있습니다. 이제이 기능은 더 광범위한 버전 및를 지원합니다 AWS 리전. 자세한 내용은 단원을 참조하십시오.	2020년 5월 5일
사용자 지정 사전	Amazon OpenSearch Service를 사용하면 클러스터에 사용할 사용자 지정 사전 파일을 업로드할 수 있습니다. 이러한 파일은 Elasticsearch에서 특정 고빈도 단어를 무시하거나 검색어를 동일하게 취급하도록 하여 검색 결과를 개선합니다.	2020년 4월 21일
Elasticsearch 7.4 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 7.4를 지원합니다. 자세한 내용은 지원되는 버전을 참조하세요.	2020년 3월 12일

k-NN	Amazon OpenSearch Service에 k-Nearest Neighbor(k-NN) 검색에 대한 지원이 추가됩니다. k-NN을 사용하려면 서비스 소프트웨어 R20200302 또는 이후 버전이 필요합니다.	2020년 3월 3일
인덱스 상태 관리	Amazon OpenSearch Service에는 인덱스 상태 관리(ISM)가 추가되어 특정 연령에 도달하면 인덱스를 삭제하는 것과 같이 주기적으로 수행되는 작업을 자동화할 수 있습니다. 이 기능을 사용하려면 서비스 소프트웨어 R20200302 이상이 필요합니다.	2020년 3월 3일
Elasticsearch 5.6.16 지원	이제 Amazon OpenSearch Service는 버그 수정을 추가하고 보안을 개선하는 버전 5.6에 대한 최신 패치 릴리스를 지원합니다. Amazon ES는 기존 5.6 도메인을 이 릴리스로 자동 업그레이드합니다. 이 Elasticsearch 릴리스에서는 버전이 5.6.17로 잘못 보고됩니다.	2020년 3월 2일
세분화된 액세스 제어	이제 Amazon OpenSearch Service는 세분화된 액세스 제어를 지원하여 인덱스, 문서, 필드 수준의 보안, Kibana 멀티테넌시, 클러스터에 대한 HTTP 기본 인증(옵션) 기능을 제공합니다.	2020년 2월 11일

UltraWarm 스토리지(미리 보기)	Amazon OpenSearch Service에 Amazon S3를 사용하는 새로운 웜 스토리지 계층인 UltraWarm 및 성능을 개선하는 정교한 캐싱 솔루션을 추가합니다. 활발하게 사용하지 않거나 쿼리 빈도가 낮은 인덱스의 경우, UltraWarm 스토리지는 GiB당 비용이 아주 저렴합니다.	2019년 12월 3일
중국 리전의 암호화 기능	현재 `cn-north-1` 중국(베이징) 및 `cn-northwest-1` 중국(닝샤) 리전에서 저장된 데이터의 암호화와 노드 간 암호화를 사용할 수 있습니다.	2019년 11월 20일
HTTPS 요구	이제 Amazon ES 도메인에 대한 모든 트래픽이 HTTPS를 통해 도착하도록 할 수 있습니다. 도메인을 구성할 때 HTTPS 요구 확인란을 선택합니다. 이 기능을 사용하려면 서비스 소프트웨어 R20190808 이상이 필요합니다.	2019년 10월 3일
Elasticsearch 7.1 및 6.8 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 7.1과 6.8을 지원합니다. 자세한 내용은 지원되는 버전을 참조하세요.	2019년 8월 13일

시간별 스냅샷	이제 Amazon OpenSearch Service는 데이터를 복원하기 위한 백업을 더 자주 만들 수 있도록 Elasticsearch 5.3 및 이후 버전을 실행하는 도메인의 일별 스냅샷 대신 시간별 스냅샷을 생성합니다.	2019년 7월 8일
Elasticsearch 6.7 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 6.7을 지원합니다. 자세한 내용은 지원되는 버전을 참조하세요.	2019년 5월 29일
SQL 지원	이제 Amazon OpenSearch Service에서 SQL을 사용하여 데이터를 쿼리할 수 있습니다. SQL 지원을 받으려면 서비스 소프트웨어 R20190418 이상이 필요합니다.	2019년 5월 15일
5시리즈 인스턴스 유형	이제 Amazon OpenSearch Service는 M5, C5, R5 인스턴스 유형을 지원합니다. 이 새로운 유형은 이전 세대의 인스턴스 유형에 비해 저렴한 가격으로 더 나은 성능을 발휘합니다. 자세한 설명은 제한을 참조하십시오.	2019년 4월 24일
Elasticsearch 6.5 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 6.5를 지원합니다.	2019년 4월 8일

알림	Amazon OpenSearch Service의 알림 기능은 하나 이상의 Amazon ES 인덱스의 데이터가 특정 조건을 충족하면 사용자에게 알립니다. 알림을 사용하려면 서비스 소프트웨어 R20190221 이상이 필요합니다.	2019년 3월 25일
가용 영역 3개 지원	이제 Amazon OpenSearch Service는 여러 리전에서 가용 영역 3개를 지원합니다. 이 릴리스에는 간소화된 콘솔 환경도 포함되어 있습니다. 다중 AZ를 사용하려면 서비스 소프트웨어 R20181023 이상이 필요합니다.	2019년 2월 7일
Elasticsearch 6.4 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 6.4를 지원합니다.	2019년 1월 23일
200 노드 클러스터	이제 Amazon ES를 사용하여 총 3PB 스토리지에 대해 최대 200개의 데이터 노드를 생성할 수 있습니다.	2019년 1월 22일
서비스 소프트웨어 업데이트	이제 Amazon ES에서는 도메인의 서비스 소프트웨어를 수동으로 업데이트하여 새로운 기능을 더 빠르게 활용하거나 트래픽이 적은 시간에 업데이트할 수 있습니다. 자세한 내용은 다음 섹션을 참조하세요.	2018년 11월 20일

새로운 CloudWatch 지표	이제 Amazon ES는 노드 수준 지표와 Amazon ES 콘솔의 새로운 클러스터 상태(Cluster health) 및 인스턴스 상태(Instance health) 탭을 사용할 수 있습니다.	2018년 11월 20일
중국(베이징) 지원	Amazon OpenSearch Service를 현재 cn-north-1 리전에서 사용할 수 있으며 M4, C4, R4 인스턴스 유형을 지원합니다.	2018년 10월 17일
노드 간 암호화	이제 Amazon OpenSearch Service는 Amazon ES가 데이터를 클러스터에 배포할 때 데이터를 암호화된 상태로 유지하는 노드 간 암호화를 지원합니다.	2018년 9월 18일
현재 위치 버전 업그레이드	이제 Amazon OpenSearch Service는 현재 위치 버전 업그레이드를 지원합니다.	2018년 8월 14일
Elasticsearch 6.3 및 5.6 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 6.3과 5.6을 지원합니다.	2018년 8월 14일
오류 로그	이제 Amazon ES에서 Amazon CloudWatch에 Elasticsearch 오류 로그를 게시할 수 있습니다.	2018년 7월 31일
중국(닝샤) 예약 인스턴스	이제 Amazon ES가 중국(닝샤) 리전에서 예약 인스턴스를 제공합니다.	2018년 5월 29일
예약 인스턴스	이제 Amazon ES에서 예약 인스턴스를 지원합니다.	2018년 5월 7일

이전 업데이트

다음 표에서는 2018년 5월 이전의 Amazon ES에 대한 중요 변경 사항을 설명합니다.

변경 사항	설명	날짜
Kibana에서의 Amazon Cognito 인증	이제 Amazon ES가 Kibana의 로그인 페이지를 보호합니다. 자세한 내용은 the section called “OpenSearch Dashboards에 대한 Amazon Cognito 인증” 섹션을 참조하세요.	2018년 4월 2일
Elasticsearch 6.2 지원	이제 Amazon OpenSearch Service는 Elasticsearch 버전 6.2를 지원합니다.	2018년 3월 14일
한국어 분석 플러그인	이제 Amazon ES가 메모리 최적화 버전의 Seunjeon 한국어 분석 플러그인을 지원합니다.	2018년 3월 13일
액세스 제어 즉시 업데이트	Amazon ES 도메인에 대한 액세스 제어 정책을 변경할 경우 즉시 적용됩니다.	2018년 3월 7일
페타바이트 규모	이제 Amazon ES에서 I3 인스턴스 유형 및 최대 1.5PB의 총 도메인 스토리지를 지원합니다. 자세한 내용은 the section called “페타바이트 규모” 섹션을 참조하세요.	2017년 12월 19일
저장된 데이터 암호화	이제 Amazon ES에서 저장된 데이터 암호화를 지원합니다. 자세한 내용은 the section called “저장 시 암호화” 섹션을 참조하세요.	2017년 12월 7일
Elasticsearch 6.0 지원	이제 Amazon ES가 Elasticsearch 버전 6.0을 지원합니다. 마이그레이션에 대한 고려 사항 및 지침은 the section called “도메인 업그레이드” 섹션을 참조하세요.	2017년 12월 6일
VPC 지원	이제 Amazon ES를 사용해 Amazon Virtual Private Cloud 내에서 도메인을 시작할 수 있습니다. VPC 지원은 추가 보안 계층을 제공하고 VPC 내에서 Amazon ES와 다른 서비스 간의 통신을 간소화합니다. 자세한 내용은 the section called “VPC 지원” 섹션을 참조하세요.	2017년 10월 17일

변경 사항	설명	날짜
느린 로그 게시	이제 Amazon ES에서 CloudWatch Logs에 대한 느린 로그 게시를 지원합니다. 자세한 내용은 the section called “로그 모니터링” 섹션을 참조하세요.	2017년 10월 16일
Elasticsearch 5.5 지원	이제 Amazon ES가 Elasticsearch 버전 5.5를 지원합니다. 이제 지원 에 문의하지 않고도 `_scripts` API를 사용하여 자동 스냅샷을 복원하고 스크립트를 저장할 수 있습니다.	2017년 9월 7일
Elasticsearch 5.3 지원	Amazon ES에서 Elasticsearch 버전 5.3에 대한 지원이 추가되었습니다.	2017년 6월 1일
클러스터당 인스턴스 수 및 EBS 용량 증가	이제 Amazon ES에서 클러스터당 최대 100개 노드와 150TB의 EBS 용량을 지원합니다.	2017년 4월 5일
캐나다(중부) 및 EU(런던) 지원	Amazon ES에 캐나다(중부), ca-central-1 및 EU(런던), eu-west-2 리전에 대한 지원이 추가되었습니다.	2017년 3월 20일
인스턴스 수 및 EBS 볼륨 증가	Amazon ES에서 인스턴스 수 및 EBS 볼륨 증가에 대한 지원이 추가되었습니다.	2017년 2월 21일
Elasticsearch 5.1 지원	Amazon ES에서 Elasticsearch 버전 5.1에 대한 지원이 추가되었습니다.	2017년 1월 30일
음성 분석 플러그인에 대한 지원	이제 Amazon ES는 음성 분석 플러그인과의 통합을 기본 제공하여 데이터에 대해 "유사한 음성" 쿼리를 실행할 수 있습니다.	2016년 12월 22일
미국 동부(오하이오) 지원	Amazon ES에 미국 동부(오하이오) us-east-2 리전에 대한 지원이 추가되었습니다.	2016년 10월 17일
새로운 성능 지표	Amazon ES에 성능 지표 `ClusterUsedSpace` 가 추가되었습니다.	2016년 7월 29일
Elasticsearch 2.3 지원	Amazon ES에서 Elasticsearch 버전 2.3에 대한 지원이 추가되었습니다.	2016년 7월 27일

변경 사항	설명	날짜
아시아 태평양(뭄바이) 지원	Amazon ES에 아시아 태평양(뭄바이) ap-south-1에 대한 지원이 추가되었습니다.	2016년 6월 27일
클러스터당 인스턴스 수 증가	Amazon ES에서 클러스터당 인스턴스의 최대 개수(인스턴스 수)가 10개에서 20개로 늘었습니다.	2016년 5월 18일
아시아 태평양(서울) 지원	Amazon ES에 아시아 태평양(서울) ap-northeast-2 리전에 대한 지원이 추가되었습니다.	2016년 1월 28일
Amazon ES	최초 릴리스.	2015년 10월 1일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.