亚马逊云科技

开发人员指南

亚马逊 OpenSearch 服务

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

亚马逊 OpenSearch 服务: 开发人员指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅 中国的 Amazon Web Services 服务入门 (PDF)。

Table of Contents

Amazon S OpenSearch ervice 是什么？ 1
Amazon OpenSearch 服务的特点 1
何时使用 3
Elasticsearch 支持的版本和 OpenSearch 3
标准支持和扩展支持 4
支持的标准版本和扩展版本 6
计算扩展支持费用 6
定价 7
相关服务 7
设置 10
授予权限 10
授予编程式访问权限 10
设置 Amazon CLI 11
打开 控制台 12
入门 13
创建域 13
上传数据以进行索引 15
选项 1：上传单个文档 15
选项 2：上传多个文档 15
搜索文档 16
从命令行搜索文档 16
使用 OpenSearch控制面板搜索文档 18
删除域 18
Amazon OpenSearch Ingestion 19
重要概念 19
优势 20
限制 21
支持的 Data Prepper 版本 21
扩缩管道 22
定价 24
支持的 Amazon Web Services 区域 24
设置角色和用户 24
管道角色 25
提取角色 28

向管道授予访问域的权限 30
向管道授予访问集合的权限 33
开始使用 OpenSearch Ingestion 39
教程：将数据摄取到域 39
教程：将数据摄取到集合 47
管道功能 53
持久缓冲功能 54
拆分 55
链接 56
死信队列 57
索引管理 59
End-to-end 确认 62
源背压 62
创建管道 63
先决条件和所需 IAM 角色 64
所需的 IAM 权限 64
指定管道版本 66
指定提取路径 66
创建管道 67
跟踪管道创建的状态 71
使用蓝图 72
查看管道 74
更新管道 76
注意事项 77
所需权限 77
更新管道 78
使用蓝绿部署进行管道更新 79
管理管道成本 79
停止管道 80
启动管道 81
删除管道 82
支持的插件和选项 83
支持的插件 83
无状态与有状态处理器 85
配置要求和限制 86
集成管道 91

构建摄取端点 91
创建摄取角色 92
Atlassian 服务 94
Amazon DynamoDB 109
Amazon DocumentDB 122
Confluent Cloud Kafka 136
Amazon MSK 146
Amazon S3 153
Amazon Security Lake 162
Fluent Bit 165
Fluentd 167
OpenTelemetry 收藏家 169
自行管理的 Kafka 171
自建集群 OpenSearch 178
Amazon Kinesis Data Streams 185
后续步骤 189
Amazon Lambda 190
在域和集合之间迁移数据 193
限制 193
OpenSearch 服务即来源 194
指定多个 OpenSearch 服务域接收器 196
将数据迁移到 OpenSearch 无服务器 VPC 集合 196
使用管理管道 Amazon SDKs 197
Python 197
OpenSearch 摄取中的安全性 201
为管道配置 VPC 访问 202
身份和访问管理 206
使用监控 CloudTrail 213
标记管道 217
所需权限 217
使用标签 (控制台) 218
使用标签 (Amazon CLI) 218
日志记录和监控 219
监控管道日志 219
监控管道指标 221
最佳实践 248

一般最佳实践 249
建议 CloudWatch 警报 249
Amazon OpenSearch 无服务器 255
优势 255
什么是 Amazon OpenSearch 无服务器？ 255
OpenSearch 无服务器的用例 256
工作方式 256
选择集合类型 258
定价 259
支持的 Amazon Web Services 区域 260
限制 260
比较 OpenSearch 服务和 OpenSearch 无服务器 260
教程： OpenSearch 无服务器入门 263
步骤 1：配置权限 264
步骤 2：创建集合 265
步骤 3：上传并搜索数据 266
步骤 4：删除集合 267
后续步骤 267
创建和管理集合 267
管理集合 268
使用向量搜索集合 276
使用数据生命周期策略 283
使用管理馆藏 Amazon SDKs 291
使用创建收藏夹 CloudFormation 302
管理容量限制 304
配置容量设置 306
最大容量限制 306
监控容量使用情况 307
将数据摄取到集合中 307
所需的最低权限 307
OpenSearch 摄取 308
Fluent Bit 308
Amazon Data Firehose 309
Go 310
Java 312
JavaScript 313

Logstash 316
Python 318
Ruby 320
其他客户端 320
OpenSearch 无服务器中的安全性 322
加密策略 324
网络策略 324
数据访问策略 325
IAM 和 SAML 身份验证 326
基础结构安全性 327
安全性入门 327
身份和访问管理 341
加密 357
网络访问 366
数据访问控制 377
VPC 端点 386
SAML 身份验证 397
合规性验证 405
标记集合 406
所需权限 406
标记集合（控制台） 406
标记集合（Amazon CLI） 407
受支持的操作和插件 407
受支持 OpenSearch 的 API 操作和权限 408
支持的 OpenSearch 插件 413
监控 OpenSearch 无服务器 414
使用监控 CloudWatch 415
使用监控 CloudTrail 419
使用监控 EventBridge 422
创建和管理域 426
创建 OpenSearch 服务域 426
创建 OpenSearch 服务域（控制台） 426
创建 OpenSearch 服务域 (Amazon CLI) 431
创建 OpenSearch 服务域 (Amazon SDKs) 433
创建 OpenSearch 服务域 (Amazon CloudFormation) 433
配置访问策略 433

高级集群设置 434
配置更改 434
通常会导致 blue/green 部署的更改 435
通常不会导致 blue/green 部署的更改 436
确定更改是否会导发蓝绿部署 436
跟踪配置更改 440
配置更改的阶段 442
蓝绿部署对性能的影响 444
配置更改的费用 444
对验证错误进行故障排除 445
服务软件更新 449
可选更新与必需更新 449
补丁更新 450
注意事项 450
启动更新 450
非高峰窗口 453
监控更新 454
当域不符合更新资格时 455
非高峰窗口 456
非高峰窗口服务软件更新 456
非高峰期自动调整优化 457
启用非高峰窗口 457
配置自定义非高峰窗口 458
查看计划的操作 458
重新计划操作 460
从自动调整维护窗口迁移 462
通知 463
开始使用通知 463
通知严重性 464
示例 EventBridge 事件 464
配置多 AZ 域 465
带待机功能的多可用区 465
不带待机功能的多可用区 467
可用区中断 470
VPC 支持 472
VPC 与公有域对比 472

限制 472
架构 473
创建索引快照 479
先决条件 479
注册手动快照存储库 483
手动创建快照 487
还原快照 488
删除手动快照 490
使用快照管理自动处理快照 490
使用索引状态管理自动执行快照 492
将 Curator 用于快照 492
升级域 493
支持的升级途径 493
升级域（控制台） 496
升级域（CLI） 496
升级域（SDK） 497
对验证失败进行故障排除 498
排查升级问题 499
使用快照迁移数据 501
创建自定义终端节点 506
新域的自定义终端节点 507
现有域的自定义终端节点 507
CNAME 映射 508
自动调整 508
更改类型 508
启用或禁用自动调整 509
计划自动调整增强功能 510
监控自动调整更改 511
标记域 511
标签示例 512
标记域（控制台） 512
标记域（Amazon CLI） 513
标记域（Amazon SDKs） 514
执行管理操作 515
在数据节点上重新启动 OpenSearch 进程 516
重启数据节点 516

重新启动仪表板流程 516
限制 517
使用直接查询 518
定价 518
限制 519
一般限制 519
亚马逊 S3 的限制 519
Amazon CloudWatch 日志的限制 519
Amazon 安全湖的限制 520
建议 520
一般建议 520
亚马逊 S3 建议 521
CloudWatch 日志推荐 521
安全湖建议 522
限额 522
亚马逊 S3 的配额 523
CloudWatch 日志配额 523
安全湖配额 524
支持 Amazon Web Services 区域 525
Amazon Web Services 区域 适用于亚马逊 S3 525
Amazon Web Services 区域 可用于 CloudWatch 日志 526
Amazon Web Services 区域 适用于安全湖 526
在 S3 中直接查询 527
创建 S3 数据源 527
配置 S3 数据源 535
在 CloudWatch 日志中直接查询 537
创建 CloudWatch 日志数据源 537
配置 CloudWatch 日志数据源 542
在安全湖中直接查询 544
创建安全湖数据源 544
配置安全湖数据源 550
管理数据来源 552
使用 CloudWatch 指标数据源进行监控 552
启用和禁用数据来源 555
用 Amazon 预算进行监控 555
删除数据来源 556

优化查询性能 ... 557
跳过索引 ... 557
实体化视图 ... 558
覆盖索引 ... 558
支持的 SQL 和 PPL 命令 ... 558
支持的 SQL 命令 ... 559
支持的 PPL 命令 ... 748
监控域 ... 921
监控集群指标 ... 922
在中查看指标 CloudWatch ... 923
解释S OpenSearch ervice 中的运行状况图表 ... 923
集群指标 ... 924
专用主节点指标 ... 930
专用协调器节点指标 ... 931
EBS 卷指标 ... 932
实例指标 ... 934
UltraWarm 指标 ... 945
冷存储指标 ... 950
OR1 指标 ... 951
提醒指标 ... 952
异常检测指标 ... 953
异步搜索指标 ... 954
自动调整指标 ... 956
带待机功能的多可用区指标 ... 957
时间点指标 ... 959
SQL 指标 ... 960
k-NN 指标 ... 961
跨集群搜索指标 ... 963
跨集群复制指标 ... 964
学习排名指标 ... 965
管道处理语言指标 ... 966
监控日志 ... 967
启用日志发布（控制台） ... 968
启用日志发布 (Amazon CLI) ... 970
启用日志发布 (Amazon SDKs) ... 972
启用日志发布 (CloudFormation) ... 972

设置搜索请求慢速日志阈值 974
设置分片慢速日志阈值 974
测试慢速日志 975
查看日志 975
监控审计日志 976
限制 977
启用审计日志 977
使用启用审核日志 Amazon CLI 979
使用配置 API 启用审计日志记录 979
审计日志图层和类别 979
审计日志设置 981
审计日志示例 984
使用 REST API 配置审计日志 987
监控事件 988
服务软件更新事件 989
自动调整事件 995
集群运行状况事件 1000
VPC 端点事件 1013
节点停用事件 1015
性能降级节点停用事件 1017
域错误事件 1019
教程：侦听 OpenSearch 服务事件 1021
教程：为可用更新发送 SNS 警报 1023
使用监控 CloudTrail 1024
中的亚马逊 OpenSearch 服务信息 CloudTrail 420
了解亚马逊 OpenSearch 服务日志文件条目 421
安全性 1029
数据保护 1030
静态加密 1030
Node-to-node 加密 1034
身份和访问管理 1034
策略的类型 1035
提出和签署 OpenSearch 服务请求 1043
当策略发生冲突时 1044
策略元素参考 1045
高级选项和 API 注意事项 1050

配置访问策略 1053
其他示例策略 1053
API 权限参考 1053
Amazon 托管策略 1054
防止跨服务混淆代理 1061
精细访问控制 1063
大局：精细的访问控制和服务安全 OpenSearch 1063
重要概念 1067
关于主用户 1067
启用精细访问控制 1068
以主用户身份访问 OpenSearch 仪表板 1071
管理权限 1073
推荐配置 1078
限制 1081
修改主用户 1082
其他主用户 1083
手动快照 1084
集成 1085
REST API 差异 1085
教程：使用 Cognito 身份验证的精细访问控制 1087
教程：使用基本身份验证的内部用户数据库 1091
合规性验证 1094
恢复能力 1095
JSON Web 令牌 1095
注意事项 1096
修改域访问策略 1096
配置 JWT 身份验证和授权 1097
使用 JWT 发送测试请求 1097
基础结构安全性 1099
使用 OpenSearch 服务托管的 VPC 端点 1099
仪表板的 SAML 身份验证 OpenSearch 1103
SAML 配置概述 1103
注意事项 1104
用于 VPC 域的 SAML 身份验证 1104
修改域访问策略 1105
配置 SP 或 IdP 发起的身份验证 1106

配置 SP 和 IdP 发起的身份验证 1112
配置 SAML 身份验证 (Amazon CLI) 1112
配置 SAML 身份验证（配置 API） 1113
SAML 故障排除 1113
禁用 SAML 身份验证 1116
IAM 身份中心支持 Amazon OpenSearch 服务 1117
控制面板的 Amazon Cognito 身份验证 OpenSearch 1119
先决条件 1120
将域配置为使用 Amazon Cognito 身份验证 1123
允许经过身份验证的角色 1127
配置身份提供商 1128
（可选）配置精细访问 1128
（可选）自定义登录页面 1129
（可选）配置高级安全 1129
测试 1129
限额 1130
常见配置问题 1130
禁用控制面板的 Amazon Cognito 身份验证 OpenSearch 1134
删除在控制面板中使用 Amazon Cognito 身份验证的域名 OpenSearch 1134
使用服务相关角色 1134
VPC 域和数据源创建角色 1135
集合创建角色 1138
管道创建角色 1140
代码示例 1144
Elasticsearch 客户端兼容性 1144
压缩 HTTP 请求 1145
启用 gzip 压缩 1145
必需的标头 1146
示例代码 (Python 3) 1146
使用 Amazon SDKs 1147
Java 1147
Python 1159
节点 1161
为数据建立索引 1165
索引的命名限制 1165
减小响应大小 1166

索引编解码器 1167
将流数据加载到 OpenSearch 服务中 1168
从 OpenSearch Ingestion 加载流数据 1168
从 Amazon S3 表中加载流数据 1168
从 Amazon Kinesis Data Streams 加载流数据 1174
从 Amazon DynamoDB 表中加载流数据 1178
从 Amazon Data Firehose 加载流数据 1182
正在加载来自亚马逊的流媒体数据 CloudWatch 1182
加载来自的流数据 Amazon IoT 1182
使用 Logstash 加载数据 1182
配置 1182
搜索数据 1186
URI 搜索 1186
请求正文搜索 1188
提升字段 1189
搜索结果突出显示 1190
计数 API 1192
对搜索结果进行分页 1192
时间点 1192
`from` 和 `size` 参数 1193
控制面板查询语言 1193
程序包 1195
所需的权限 1196
将程序包上传到 Amazon S3 1196
导入和关联程序包 1197
将包与配合使用 OpenSearch 1197
更新程序包 1202
使用新词典手动更新索引 1205
取消程序包关联并移除程序包 1207
自定义插件 1208
第三方插 1223
SQL 支持 1226
调用示例 1228
说明和差异 1228
SQL Workbench 1229
SQL CLI 1097

JDBC 驱动程序 1229
ODBC 驱动程序 1229
k-NN 搜索 1229
k-NN 入门 1230
k-NN 差异、调整和限制 1232
跨集群搜索 1233
限制 1234
跨集群搜索先决条件 1234
跨集群搜索定价 1234
设置连接 1234
移除连接 1236
设置安全性和示例演练 1236
OpenSearch 仪表板 1242
学习排名 1242
学习排名入门 1243
学习排名 API 1264
异步搜索 1270
搜索调用示例 1270
异步搜索权限 1272
异步搜索设置 1272
跨集群搜索 1272
UltraWarm 1274
时间点 1275
注意事项 1275
创建 PIT 1275
时间点权限 1277
PIT 设置 1278
跨集群搜索 1278
UltraWarm 1278
语义搜索 1278
并发分段搜索 1279
自然语言查询生成 1279
先决条件 1279
入门 1280
配置 权限 1280
配置自动化 1280

OpenSearch 仪表板 1282
控制对仪表板的访问权限 1282
使用代理从仪表板访问 OpenSearch 服务 1283
将仪表板配置为使用 WMS 地图服务器 1287
将本地仪表板服务器连接到 OpenSearch 服务 1288
在仪表板中管理索引 1289
其他功能 1290
OpenSearch 用户界面 1291
发行历史记录 1292
入门 1293
创建亚马逊 OpenSearch 服务应用程序所需的权限 1293
创建 应用程序 1297
管理应用程序管理员 1304
通过 IAM 启用 SAML 联合 1307
步骤 1：设置身份提供商应用程序 (Okta) 1307
第 2 步：为 Okta 设置 Amazon 配置 1311
步骤 3：在 IAM 中创建亚马逊 OpenSearch 服务访问策略 1313
步骤 4：使用 SAML 验证身份提供商启动的单点登录体验 1315
管理数据源关联和 VPC 访问权限 1319
将数据源与 OpenSearch UI 应用程序关联 1319
管理 VPC 中域的访问权限 1320
在 VPC 中配置对 OpenSearch 无服务器集合的访问权限 1321
使用亚马逊 OpenSearch 服务工作空间 1324
创建 OpenSearch UI 应用程序工作区 1324
工作空间隐私和协作者 1325
工作空间类型 1326
通过跨集群搜索实现跨区域和跨账户数据访问 1326
通过跨集群搜索设置跨区域和跨账户数据访问的访问权限 1328
在域之间创建连接 1330
通过跨集群搜索测试跨区域和跨账户数据访问的安全设置 1332
删除连接 1335
管理从 VPC 终端节点对 OpenSearch UI 的访问 1335
在 VPC 和 OpenSearch 用户界面之间创建私有连接 1335
更新 VPC 终端节点策略以允许访问 OpenSearch UI 应用程序 1337
在 VPC 终端节点策略中撤消对 OpenSearch UI 的访问权限 1337
管理索引 1339

UltraWarm 存储 1339
先决条件 1340
UltraWarm 存储要求和性能注意事项 1342
UltraWarm 定价 1342
启用 UltraWarm 1343
将索引迁移到 UltraWarm 存储 1345
自动执行迁移 1348
迁移调整 1348
取消迁移 1348
列出热索引和暖索引 1349
将温索引返回到热存储 1349
从快照恢复温索引 1349
暖索引的手动快照 1351
将温索引迁移到冷存储 1352
KNN 索引的最佳实践 1352
禁用 UltraWarm 1352
冷存储 1353
先决条件 1353
冷存储要求和性能注意事项 1355
冷存储定价 1355
启用冷存储 1355
在控制 OpenSearch 面板中管理冷索引 1357
将索引迁移到冷存储 1357
自动迁移到冷存储 1359
取消迁移到冷存储 1359
列出冷索引 1359
将冷索引迁移到温存储 1363
从快照恢复冷索引 1365
取消从冷存储迁移到热存储 1365
更新冷索引元数据 1365
删除冷索引 1366
禁用冷存储 1366
OpenSearch 优化存储 1366
限制 1367
进行微调以提高摄取吞吐量 1367
OpenSearch 优化后的实例与其他实例有何不同 1367

与 UltraWarm存储有何 OR1 不同 1368
为域配置实 OR1 例 1368
索引状态管理 1369
创建一个 ISM 策略 1370
示例策略 1371
ISM 模板 1375
差异 1375
教程：实现 ISM 过程的自动化 1377
索引汇总 1381
创建索引汇总作业 1381
索引转换 1382
创建索引转换任务 1383
跨集群复制 1384
限制 1385
先决条件 1385
权限要求 1386
设置跨集群连接 1387
开始复制 1388
确认复制 1388
暂停和恢复复制 1389
停止复制 1390
自动关注 1390
升级已连接的域 1392
远程重建索引 1392
先决条件 1393
在 OpenSearch 服务互联网域之间重新索引数据 1393
当远程域位于 VPC 中时，重新索引数据 1394
在非OpenSearch 服务域之间重新索引数据 1398
重新索引大型数据集 1399
远程重新索引设置 1400
数据流 1401
数据流入门 1401
监控数据 1405
警报 1405
提醒权限 1405
开始使用警报 1406

通知 1406
差异 1407
异常检测 1409
..... 1409
教程：使用异常检测功能检测高 CPU 使用率 1412
Amazon Q 支持 1415
设置。 Amazon Web Services 区域 1415
设置 Amazon Q 数据集 OpenSearch 成 1416
使用自然语言生成可视化效果 1416
查看警报摘要和见解 1418
开始前的准备工作 1420
查看警报摘要和见解 1421
在 “发现” 页面上查看 Amazon Q 生成的查询结果摘要 1421
描述异常检测器 1423
访问 Amazon Q 聊天室咨询 OpenSearch 服务问题 1423
机器学习 1427
连接器适用于 Amazon Web Services 服务 1427
先决条件 1427
创建 OpenSearch 服务连接器 1430
适用于外部平台的连接器 1432
先决条件 1433
创建 OpenSearch 服务连接器 1436
CloudFormation 模板集成 1438
先决条件 1438
Amazon SageMaker AI 模板 1439
Amazon Bedrock 模板 1440
不支持的 ML Commons 设置 1441
流框架插件 1441
在 OpenSearch 服务中创建 ML 连接器 1442
配置 权限 1449
安全分析 1451
安全分析组件和概念 1451
日志类型 1451
探测器 1452
规则 1452
调查发现 1452

警报 1452
探索安全分析 1452
配置 权限 1454
故障排除 1456
无此类索引错误 1456
可观察性 1457
使用事件分析来探索数据 1457
创建可视化 1459
使用跟踪分析功能更深入探索 1460
跟踪分析 1461
先决条件 1462
OpenTelemetry 收集器示例配置 1462
OpenSearch Ingestion 示例配置 1463
探索跟踪数据 1464
管道式处理语言 1466
..... 1466
最佳实践 1468
监控和提醒 1468
配置 CloudWatch 警报 1468
启用日志发布 1468
分片策略 1469
确定分片和数据节点数 1469
避免存储偏斜 1470
稳定性 1470
随时了解最新动态 OpenSearch 1470
提高快照性能 1471
启用专用主节点 1471
跨多个可用区进行部署 1471
控制摄取流量和缓冲 1472
为搜索工作负载创建映射 1472
使用索引模板 1473
使用索引状态管理来管理索引 1474
删除未使用的索引 1474
使用多个域以实现高可用性 1474
性能 1474
优化批量请求大小和压缩 1474

降低批量请求响应的大小 1475
优化刷新间隔时间 1475
启用自动调整 1475
安全性 1476
启用精细访问控制 1476
在 VPC 中部署域 1476
应用限制性访问策略 1476
启用静态加密 1476
启用 node-to-node加密 1477
使用监视器 Amazon Security Hub 1477
成本优化 1477
使用最新一代实例类型 1477
使用最新的 Amazon EBS gp3 卷 1477
时间序列日志数据的使用 UltraWarm 和冷存储 1478
检查有关预留实例的建议 1478
调整域大小 1478
计算存储要求 1479
选择分片数量 1480
选择实例类型和测试 1481
PB 规模 1483
专用协调器节点 1484
何时使用专用协调器节点 1484
架构和行为 1484
要求和限制 1485
配置专用协调器节点 1485
最佳实践 1486
专用主节点 1487
专用主节点的数量 1488
为专用主节点选择实例类型 1489
推荐的 CloudWatch 警报 1490
一般参考 1497
支持的实例类型 1497
当前一代实例类型 1497
上一代实例类型 1513
功能（按引擎版本） 1517
插件（按引擎版本） 1521

可选插件 1524
支持的操作 1524
值得注意的 API 差异 1524
限额 1575
UltraWarm 存储配额 1576
每个 AZ 的数据节点数 1489
按实例系列划分的节点总数限制 1576
EBS 卷大小配额 1577
网络配额 1587
分片大小配额 1594
分片计数配额 1594
Java 进程配额 1595
域策略配额 1595
预留实例 1595
购买预留实例（控制台） 1596
购买预留实例 (Amazon CLI) 1596
购买预留实例 (Amazon SDKs) 1599
调查费用 1601
其他支持的资源 1601
教程 1603
创建和搜索文档 1603
先决条件 1603
将文档添加到索引 1604
创建自动生成的 ID IDs 1605
使用 POST 命令更新文档 1606
执行批量操作 1607
搜索文档 1607
相关资源 1609
迁移到 OpenSearch 服务 1610
拍摄并上载快照 1610
创建域 1611
提供权限以访问 S3 存储桶。 1612
还原快照。 1614
创建搜索应用程序 1617
先决条件 1618
步骤 1：为示例数据建立索引 1618

步骤 2：创建并部署 Lambda 函数 1618
步骤 3：在 API Gateway 中创建 API 1621
步骤 4：（可选）修改域访问策略 1624
映射 Lambda 角色（如果使用精细访问控制） 1626
步骤 5：测试 Web 应用程序 1627
后续步骤 1629
可视化支持呼叫 1629
步骤 1：配置先决条件 1630
步骤 2：复制示例代码 1631
(可选) 步骤 3：索引示例数据 1636
步骤 4：分析和可视化您的数据 1637
步骤 5：清除资源和后续步骤 1641
Amazon OpenSearch 服务重命名 1642
新 API 版本 1642
重命名的实例类型 1642
访问策略更改 1642
IAM 策略 1643
SCP 策略 1643
新资源类型 1643
Kibana 重命名为控制面板 OpenSearch 1644
重命名的 CloudWatch 指标 1645
账单和成本管理控制台更改 1646
新事件格式 1647
什么保持不变？ 1647
开始吧：将您的域名升级到 OpenSearch 1.x 1647
故障排除 1649
无法访问 OpenSearch 仪表板 1649
无法访问 VPC 域 1649
集群处于只读状态 1649
红色集群状态 1650
自动修复红色集群 1652
从连续处理繁重负载恢复 1652
黄色集群状态 1654
ClusterBlockException 1654
缺少可用存储空间 1654
JVM 内存压力过高 1654

迁移到带待机功能的多可用区时出错 1655
从无备用域的域迁移到有备用域的域期间，创建索引、索引模板或 ISM 策略 1456
数据副本数量错误 1655
JVM OutOfMemoryError 1655
集群节点失败 1656
超过最大分片限制 1657
域卡在 Processing（正在处理）状态 1657
EBS 可爆发容量余额低 1657
在调整卷大小期间，EBS 指标会增加 1658
无法启用审核日志 1658
无法关闭索引 1658
客户端许可检查 1658
请求限制 1659
无法通过 SSH 登录节点 1659
“对象的存储类无效”快照错误 1659
主机标头无效 1659
M3 实例类型无效 1659
启用后，热门查询停止工作 UltraWarm 1660
升级后无法降级 1660
需要所有 Amazon Web Services 区域的域摘要 1660
使用 OpenSearch 仪表板时出现浏览器错误 1661
节点分片和存储偏斜 1661
索引分片和存储偏斜 1662
在选择 VPC 访问后出现未授权的操作 1662
在创建 VPC 域后卡在加载状态 1663
对 OpenSearch API 的请求被拒绝 1663
无法从 Alpine Linux 连接 1664
Search Backpressure 请求过多 1664
在使用开发工具包时出现证书错误 1664
由于版本兼容性，自定义插件安装失败 1665
文档历史记录 1668
早期更新 1702
Amazon 词汇表 1705
..... mdccvi

Amazon S OpenSearch ervice 是什么？

Amazon OpenSearch Service 是一项托管服务，让用户能够在 Amazon 云中轻松部署、运行并扩展 OpenSearch 集群。S OpenSearch ervice 域与 OpenSearch 集群是同义词。域是包含您指定的设置、实例类型、实例计数和存储资源的集群。Amazon S OpenSearch ervice 支持 OpenSearch 旧式 Elasticsearch OSS（最高 7.10，该软件的最终开源版本）。创建域时，您可以选择使用哪种搜索引擎。

OpenSearch是一种全面开源搜索和分析引擎，使用案例包括日志分析、实时应用程序监控、点击流分析等。有关更多信息，请参阅 OpenSearch 文档。

Amazon S OpenSearch ervic e 会为您的 OpenSearch 集群配置所有资源并启动它。它还自动检测和替换失败的S OpenSearch ervice 节点，减少与自管理基础设施相关的开销。您只需调用一次 API 或在控制台中单击几下就可扩展集群。

要开始使用 S OpenSearch ervice，您需要创建一个相当于 OpenSearch 集群的 OpenSearch 服务域。集群中的每个 EC2 实例都充当一个S OpenSearch ervice 节点。

您可以在几分钟内使用S OpenSearch ervice 控制台设置和配置一个域。如果您更喜欢编程访问，则可以使用Amazon CLIAmazon SDKs、或 Terraform。

Amazon OpenSearch 服务的特点

OpenSearch Service 包括以下功能：

Scale

- 大量 CPU、内存和存储容量配置，也称为实例类型，包括具有成本效益的 Graviton 实例。
- 支持多达 1002 个数据节点
- 高达 25 PB 的连接存储
- 为只读数据提供经济实惠UltraWarm的冷存储

安全性

- Amazon Identity and Access Management (IAM) 访问控制
- 与 Amazon VPC 和 VPC 安全组轻松集成
- 静态数据加密和 node-to-node加密
- 控制面板的 Amazon Cognito、HTTP 基本身份验证或 SAML 身份验证 OpenSearch
- 索引级、文档级和字段级安全性
- 审核日志
- 控制面板多租户

稳定性

- 资源具有大量的地理位置，也称为区域和可用区
- 在 Amazon 同一区域的两个或三个可用区之间的节点分配，也称为 Multi-AZ。
- 利用专用主节点来卸载集群管理任务
- 自动快照用于备份和还原S OpenSearch ervice 域

弹性

- SQL 支持与商业智能 (BI) 应用程序集成
- 自定义程序包以改善搜索结果

与热门服务的集成

- 使用 OpenSearch 控制面板进行数据可视化
- 与 Amazon CloudWatch 的集成，用于监控S OpenSearch ervice 域指标和设置警报

- 与的集成， Amazon CloudTrail 用于审核对S OpenSearch ervice 域的配置 API 调用
- 与 Amazon S3、Amazon Kinesis 和 Amazon DynamoDB 数据集成，将流数据加载到服务 OpenSearch
- 数据超过特定阈值时从 Amazon SNS 发出的警报

OpenSearch 与亚马逊 OpenSearch 服务相比，何时使用该服务

下表有助您确定预置的 Amazon S OpenSearch ervice 还是自行管理 OpenSearch 更适合您。

OpenSearch	亚马逊 OpenSearch 服务
• 您的组织愿意手动监控和维护自行预置的集群，并且拥有具备相应技能的人员。 • 您需要对代码拥有完全的编译级别控制。 • 您的组织希望或非常独特地使用开源软件。 • 您执行多云战略，需要不特定于供应商的技术。 • 您的团队有能力解决任何关键的生产问题。 • 您希望能够灵活地以任何需要的方式使用、修改和扩展产品。 • 您希望在新功能发布后立即使用这些功能。	• 您不想手动管理、监控和维护基础设施。 • 您想利用 Amazon S3 的持久性和低成本优势，通过跨存储层进行数据分层，从而管理不断增加的分析成本。 • 您想利用与 Amazon Web Services 服务 DynamoDB、Amazon DB（兼容 MongoDB）、IAM 和。 CloudWatch CloudFormation • 您希望能够在 Amazon Web Services 支持 进行预防性维护和出现生产问题时，轻松获得的帮助。 • 您想利用自我修复、主动维护、韧性和备份等功能。

Elasticsearch 支持的版本和 OpenSearch

OpenSearch 服务支持以下版本 OpenSearch：

- 2.19、2.17、2.11、2.7、2.2.7、2.2.7、2.7、2.7、2.7、2.7、2.7、2.7、2.2、2.2.7、2.2.2、2.2.2.2.2、2.2

OpenSearch 该服务支持以下版本的旧版 Elasticsearch：

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

我们建议升级到最新的可用 OpenSearch 版本，以便在性价比、功能丰富性和安全性改进方面充分利用 OpenSearch 服务。

标准支持和扩展支持

Amazon 为标准支持的版本提供错误修复和安全更新。对于扩展支持的版本，在标准支持结束后 Amazon 提供至少 12 个月的关键安全修复，按标准化实例小时数 (NIH) 收取固定费用。NIH 基于实例大小和使用时长。

如果域名运行的版本不再受标准支持，则会自动收取扩展支持费用。要避免这些费用，请升级到受支持的版本。

下表显示了旧版 Elasticsearch 版本 OpenSearch 的终止支持时间表。

OpenSearch 服务支持多个版本 OpenSearch 和旧版开源 Elasticsearch 版本。对于某些版本，我们已经发布了终止标准支持和延长支持日期。我们建议您升级到最新的可用 OpenSearch版本，以便在性价比、功能丰富性和安全性改进方面充分利用 OpenSearch 服务。下表提供了 Elasticsearch 和 OpenSearch 版本及其支持时间表的列表。

Elasticsearch 版本的终止支持时间表如下：

软件版本	Standards Support	Extends Suarch
Elas 版本 1.5 和 2.3	2025 年	2026 年
Elasticsearch 5.1 到 5.5	2025 年	2026 年
Elasticsearch 版本	2025 年	2028 年
Elasticsearch 版本 6.0 到 6.7	2025 年	2026 年

软件版本	Standards Support	Extends Suarch
Elasticsearch 6.8 版	未公告	未公告
Elasticsearch 版本 7.1 到 7.8	2025 年	2026 年
Elasticsearch 7.9 版	未公告	未公告
Elasticsearch 7.10 版	未公告	未公告

各 OpenSearch 版本的终止支持时间表如下：

软件版本	Standards Support	Extends Suarch
OpenSearch 版本 1.0 到 1.2	2025 年	2026 年
OpenSearch 版本 1.3	未公告	未公告
OpenSearch 版本 2.3 到 2.9	2025 年	2026 年
OpenSearch 版本 2.11 及更高版本	未公告	未公告

和 Elasticsearch 的标准支持 OpenSearch 和扩展支持

Amazon 为 Standard Support 涵盖的版本提供定期错误修复和安全更新。对于 Extended Support 下的版本，在标准支持结束后至少 12 个月内 Amazon 提供关键安全修复，每个标准化实例小时数 (NIH) 需额外支付固定费用。NIH 是根据实例大小（例如中型、大型）和实例小时数的因子计算得出的（有关示例，请参阅下面的计算延长支持费用部分）。如果域名运行的版本已终止标准支持，则会自动收取扩展支持费用。您可以升级到仍在标准支持范围内的最新版本，以避免延长支持费用。有关延期支持费用的更多信息，请参阅定价页面。有关扩展支持的一般信息，请参阅 Extended Supp ort 常见问题解答。

计算扩展支持费用

在扩展支持下运行版本的域将收取固定额外费用fee/Normalized Instance Hour (NIH), for example, $0.0065 in the US East (North Virginia) Region. NIH is computed as a factor of the instance size (e.g., medium, large), and the number of instance hours. For example, if you are running an m7g.medium.search instance for 24 hours in the US East (North Virginia) Region, which is priced at $0.068/Instance hour (on-demand), you will typically pay $1.632 ($0.068x24). If you are running a version that is in extended support, you will pay an additional $0.0065/NIH，计算方法为 0.0065 美元 x 24（实例小时数）x 2（大小标准化系数；中型实例为 2），24 小时延长支持为 0.312 美元。您将为 24 小时支付的总金额将是标准实例使用成本和延期支持成本的总和，即 1.944 美元（1.632 美元 +0.312 美元）。下表显示了 S OpenSearch ervice 中各种实例大小的标准化系数。

实例大小	标准化因子
nano	0.25
微型	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32

实例大小	标准化因子
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

Amazon S OpenSearch ervice 定价

对于 OpenSearch 服务，您需要按小时支付 EC2 实例和附加到您的实例的任何 EBS 存储卷的累计大小的费用。标准 Amazon 数据传输费用也适用。

但是，存在一些明显的数据传输异常。如果某个域使用多个可用区，则 S OpenSearch ervice 不会对可用区之间的流量进行计费。分片分配和重新平衡期间，域中会发生大量数据传输。 OpenSearch 既不会计量此流量，也不会收取此流量的费用。同样， OpenSearch 服务不对 UltraWarm/冷节点和 Amazon S3 之间的数据传输收费。

有关完整的定价详情，请参阅 Amazon S OpenSearch ervice 定价。有关配置更改期间产生的费用的信息，请参阅 the section called “配置更改的费用”。

相关服务

OpenSearch Service 通常用于以下服务：

Amazon CloudWatch

OpenSearch Service 域自动将指标发送到， CloudWatch 以便您可以监控域运行状况和性能。有关更多信息，请参阅 使用 Amazon 监控 OpenSearch 集群指标 CloudWatch。

CloudWatch 日志也可以反过来用。您可以将配置为 CloudWatch 将数据流式传输到 S OpenSearch ervice 以便分析。要了解更多信息，请参阅the section called “正在加载来自亚马逊的流媒体数据 CloudWatch”。

Amazon CloudTrail

使用 Amazon CloudTrail 获取您账户的S OpenSearch ervice 配置 API 调用和相关事件的历史记录。有关更多信息，请参阅 使用监控亚马逊 OpenSearch 服务 API 调用 Amazon CloudTrail。

Amazon Kinesis

Kinesis 是一种托管服务，能够实时进行大规模流数据处理。有关更多信息，请参阅 the section called “从 Amazon Kinesis Data Streams 加载流数据” 和 the section called “从 Amazon Data Firehose 加载流数据”。

Amazon S3

Amazon Simple Storage Service (Amazon S3) 提供 Internet 的存储服务。本指南提供了用于与 Amazon S3 集成的 Lambda示例代码。有关更多信息，请参阅 the section called “从 Amazon S3 表中加载流数据”。

Amazon IAM

Amazon Identity and Access Management (IAM) 是一项 Web 服务，可用于管理对您的S OpenSearch ervice 域的访问。有关更多信息，请参阅 the section called “身份和访问管理”。

Amazon Lambda

Amazon Lambda 是一项计算服务，可用于运行代码，而无需预置或管理服务器。本指南提供了 Lambda 示例代码，用于从DynamoDB、Amazon S3 和 Kinesis 流式传输数据。有关更多信息，请参阅 the section called “将流数据加载到 OpenSearch 服务中”。

Amazon DynamoDB

Amazon DynamoDB 是一种全托管 NoSQL 数据库服务，提供快速而可预测的性能，能够实现无缝扩展。要了解有关将数据流式传输到 S OpenSearch ervice 的更多信息，请参阅the section called “从 Amazon DynamoDB 表中加载流数据”。

Amazon QuickSight

您可以使用 QuickSight 控制面板将来自 S OpenSearch ervice 的数据可视化。有关更多信息，请参阅QuickSight 用户指南 QuickSight中的将 Amazon OpenSearch 服务与配合使用。

> **Note**
>
> OpenSearch 包括来自 Elasticsearch B.V. 的某些 Apache 许可的 Elasticsearch 代码和其他源代码。Elasticsearch B.V. 不是其他源代码的源。ELASTICSEARCH 是 Elasticsearch B.V. 的注册商标。

设置 Amazon S OpenSearch ervice

授予权限

在生产环境中，我们建议您使用更精细的策略。要了解有关访问管理的更多信息，请参阅 IAM 用户指南中的Amazon 资源访问管理。

要提供访问权限，请为您的用户、组或角色添加权限：

- 通过身份提供商在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供商创建角色（联合身份验证）的说明进行操作。
- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

授予编程式访问权限

如果用户需要在 Amazon 之外与交互，则需要编程式访问权限 Amazon Web Services Management Console。 Amazon APIs 和 Amazon Command Line Interface 需要访问密钥。可能的话，创建临时凭证，该凭证由一个访问密钥 ID、一个秘密访问密钥和一个指示凭证何时到期的安全令牌组成。

要向用户授予编程式访问权限，请选择以下选项之一。

哪个用户需要编程式访问权限？	目的	方式
IAM	使用短期证书签署对 Amazon CLI 或的编程请求 Amazon APIs（直接或使用 Amazon SDKs）。	按照 IAM 用户指南中的将临时证书与 Amazon 资源配合使用中的说明进行操作。
IAM	（不推荐使用） 使用长期证书签署对 Amazon CLI 或的编程请求 Amazon	按照《IAM 用户指南》中管理 IAM 用户的访问密钥中的说明进行操作。

哪个用户需要编程式访问权限？	目的	方式
	APIs（直接或使用 Amazon SDKs）。	

安装和配置 Amazon CLI

如果要使用 OpenSearch 服务 APIs，则必须安装最新版本的 Amazon Command Line Interface (Amazon CLI)。无需即可通过控制台使用 S OpenSearch ervice，按照中的步骤操作，在无 CLI 的情况下开始使用开始使用 Amazon S OpenSearch ervice。 Amazon CLI

要设置 Amazon CLI

1. 要安装 Amazon CLI 适用于 macOS、Linux 或 Windows 的最新版本，请参阅安装或更新最新版本的。 Amazon CLI
2. 要配置 Amazon CLI 和安全设置您的访问 Amazon Web Services 服务权限（包括 OpenSearch 服务），请参阅使用进行快速配置`aws configure`。
3. 要验证设置，请在 DataBrew 命令提示符处输入以下命令。

   ```
   aws opensearch help
   ```

   Amazon CLI 命令使用配置 Amazon Web Services 区域 中的默认，除非使用参数或配置文件另行设置。要 Amazon Web Services 区域 使用参数设置，您可以将`--region`参数添加到每个命令中。

   要 Amazon Web Services 区域 使用配置文件设置，请先在`~/.aws/config`文件或文件中添加命名配置`%UserProfile%/.aws/config`文件 (Microsoft Windows)。按 Amazon CLI的命名配置文件中的步骤执行操作。接下来，使用与以下示例类似的命令设置您 Amazon Web Services 区域的和其他设置。

   ```
   [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
   ```

打开 控制台

本部分中的大多数面向控制台的主题都是从 S OpenSearch ervic e 控制台开始的。如果您尚未登录 Amazon Web Services 账户，请登录，然后打开 S OpenSearch ervice 控制台并继续下一部分以继续 S OpenSearch ervice 入门。

开始使用 Amazon S OpenSearch ervice

开始之前，如果您还没有账户，请先先注册一个 Amazon Web Services 账户。设置帐户后，请完成 Amazon S OpenSearch ervice的入门教程。如果您在了解该服务时需要更多信息，请参考以下介绍性主题：

- 创建域。
- 根据工作负载调整域大小。
- 使用域访问策略或精细访问控制来控制对域的访问权限。
- 手动索引数据或来自其他 Amazon 服务的索引。
- 使用OpenSearch 控制面板搜索数据并创建可视化效果。
- 了解有关创建域的更多高级选项的信息。有关更多信息，请参阅 创建和管理域。
- 了解如何管理域中的索引。有关更多信息，请参阅 管理索引。
- 尝试使用 Amazon S OpenSearch ervice 的教程之一。有关更多信息，请参阅 教程。

有关从自行管理的 OpenSearch 集群迁移到 S OpenSearch ervice 的信息，请参阅the section called “迁移到 OpenSearch 服务”。

有关更多详细信息，请参阅创建和管理域和本指南中的其他主题。有关从自行管理的 OpenSearch 集群迁移到 S OpenSearch ervice 的信息，请参阅the section called “迁移到 OpenSearch 服务”。

您可以使用 OpenSearch 服务控制台 Amazon CLI、或 Amazon SDK 完成以下步骤。有关安装和设置的信息 Amazon CLI，请参阅Amazon Command Line Interface 用户指南。

创建 Amazon S OpenSearch ervice 域

> **Important**
>
> 这是一个简明教程，用于配置测试 Amazon S OpenSearch ervice 域。不要使用此流程创建生产域。有关相同流程的综合版本，请参阅创建和管理域。

S OpenSearch ervice 域与 OpenSearch 集群是同义词。域是包含您指定的设置、实例类型、实例计数和存储资源的集群。您可以使用控制台、或 Amazon CLI，来创建 OpenSearch 服务域 Amazon SDKs。

要使用控制台创建S OpenSearch ervice 域

1. 前往https://aws.amazon.com并选择 “登录到控制台”。
2. 在 “分析” 下，选择 “亚马逊 OpenSearch 服务”。
3. 选择 Create domain（创建域）。
4. 提供域的名称。本教程中的示例使用名称：movies。
5. 对于域创建方法，选择标准创建。

> **Note**
>
> 要使用最佳实践快速配置生产域，可以选择轻松创建。本教程用于开发和测试目的，我们将使用标准创建。

6. 对于模板，请选择开发/测试。
7. 对于部署选项，请选择带有待机状态的域。
8. 对于版本，请选择最新版本。
9. 对于现在，请忽略数据节点、冷热数据存储、专用主节点、快照配置和自定义端点部分。
10. 在教程中为简单起见，请使用公有访问域。在 Network（网络）下，选择 Public access（公有访问权限）。
11. 在精细访问控制设置中，选中启用精细访问控制复选框。选择创建主用户并提供用户名和密码。
12. 现在，忽略 SAML 身份验证和 Amazon Cognito 身份验证章节。
13. 对于 Access policy（访问策略），选择 Only use fine-grained access control（仅使用精细访问控制）。在本教程中，精细访问控制处理身份验证，而不是域访问策略。
14. 忽略其余设置，然后选择 Create（创建）。新域初始化流程通常需要 15-30 分钟，但可能需要更长的时间，具体取决于配置。域初始化后，请选择该域以打开其配置窗格。记下 General information（一般信息）下的域端点（例如，`https://search-my-domain.us-east-1.es.amazonaws.com`），您将会在下一个步骤中用到它。

下一步：将数据上传到 OpenSearch 服务域进行索引

将数据上传到 Amazon S OpenSearch ervice 以便编制索引

> **Important**
>
> 这是一个简明教程，用于将少量测试数据上传到 Amazon S OpenSearch ervice。有关在生产域中上传数据的详细信息，请参阅 为数据建立索引。

您可以使用命令行或大多数编程语言将数据上传到 S OpenSearch ervice 域。

为简化和方便起见，以下示例请求使用了 curl（常见的 HTTP 客户端）。像 curl 这样的客户端无法执行您的访问策略指定 IAM 用户或角色时所需的请求签名。要成功完成此步骤，必须使用精细访问控制以及主用户名和密码，就像在步骤 1 中配置的那样。

您可以在 Windows 上安装 curl 并通过命令提示符使用它，但建议您使用 Cygwin 或 Windows Subsystem for Linux 之类的工具。macOS 和大多数 Linux 发行版都预安装有 curl。

选项 1：上传单个文档

运行以下命令将单个文档添加到 movies 域：

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

在命令中，提供您在步骤 1 中创建的用户名和密码。

有关此命令的详细说明，以及如何向 S OpenSearch ervice 发出已签名的请求，请参阅为数据建立索引。

选项 2：上传多个文档

要上传包含多个文档的 JSON 文件到某个S OpenSearch ervice 域

1. 创建名为 `bulk_movies.json` 的本地文件。将以下内容粘贴到文件中，并添加一个尾部换行：

```
{ "index" : { "_index": "movies", "_id" : "2" } }
{"director": "Frankenheimer, John", "genre": ["Drama", "Mystery", "Thriller",
 "Crime"], "year": 1962, "actor": ["Lansbury, Angela", "Sinatra, Frank", "Leigh,
```

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

2. 在存储文件的本地目录中，运行以下命令，将其上传到 movies 域：

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

有关批量文件格式的更多信息，请参阅为数据建立索引。

下一步：搜索文档

在 Amazon S OpenSearch ervice 中搜索文档

要在 Amazon S OpenSearch ervice 域中搜索文档，请使用 OpenSearch 搜索 API。也可以使用OpenSearch 控制面板搜索域中的文档。

从命令行搜索文档

运行以下命令在 movies 域中搜索单词 mars：

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

如果您使用上一页的批量数据，请尝试搜索 rebel。

您可以看到类似以下内容的响应：

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

使用 OpenSearch控制面板搜索文档

OpenSearch 控制面板是一种流行的开源虚拟化工具，专为与之结合使用而设计 OpenSearch。它提供了一个有用的用户界面，供您搜索和监控您的索引。

使用控制面板从 S OpenSearch ervice 域中搜索文档

1. 导航到域的 OpenSearch 控制面板 URL。您可以在Ser OpenSearch vice 控制台中找到域的控制面板 URL。URL 遵循以下格式：

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

2. 使用您的主用户名和密码登录。
3. 要使用控制面板，您需要创建至少一个索引模式。控制面板使用这些模式来标识要分析的索引。打开左侧导航窗格，选择 Stack Management（堆栈管理），选择 Index Patterns（索引模式），然后选择 Create index pattern（创建索引模式）。在本教程中，请输入 movies。
4. 选择下一步，然后选择创建索引模式。创建模式后，您可以查看各种文档字段，例如 `actor` 和 `director`。
5. 返回到 Index Patterns（索引模式）选项卡，并确保 `movies` 已设置为默认值。如果不是，请选择该模式，然后选择星形图标以将其设为默认值。
6. 要开始搜索数据，请再次打开左侧导航菜单，然后选择 Discover（发现）。
7. 在搜索栏中，如果您上传了单个文档，请输入 mars，或者如果您上传了多个文档，输入 rebel，然后按 Enter。您可以尝试搜索其他词语，例如演员或导演姓名。

下一步：删除域

删除 Amazon S OpenSearch ervice 域

由于本教程中的 movies 域用于测试目的，因此在试用完毕后应将其删除，以避免产生费用。

要使用控制台删除 S OpenSearch ervice 域

1. 登录亚马逊 OpenSearch 服务控制台。
2. 在 Domains（域）下，选择 movies（电影）域。
3. 选择 Delete（删除），然后确认删除。

Amazon OpenSearch Ingestion 概述

Amazon OpenSearch Ingestion 是一个完全托管的无服务器数据收集器，可将实时日志、指标和跟踪数据流式传输到亚马逊 OpenSearch 服务域和 OpenSearch 无服务器集合。

借助 OpenSearch Ingestion，您不再需要像 Logstash 或 Jaeger 这样的第三方工具来摄取数据。您可以将数据生成器配置为将数据发送到 OpenSearch Ingestion，然后它会自动将其传输到您的指定域或集合。您也可以在交付数据之前转换数据。

由于 OpenSearch Ingestion 是无服务器的，因此您无需手动管理基础架构、修补软件或扩展集群。您可以直接在中配置摄取管道 Amazon Web Services Management Console，其余工作由 OpenSearch Ingestion 处理。

作为 Amazon OpenSearch 服务的一部分， OpenSearch Ingestion 由 Data Prepper 提供支持，Data Prepper 是一个开源数据收集器，可以筛选、丰富、转换、标准化和聚合数据，以进行下游分析和可视化。

Amazon OpenSearch Ingestion 中的关键概念

在开始使用 OpenSearch Ingestion 之前，了解这些关键概念会很有帮助。

管道

从 OpenSearch 摄取的角度来看，管道是指您在服务中创建的单个预配置数据收集器。OpenSearch 您可以将其视为整个 YAML 配置文件，其中包含一个或多个子管道。有关创建提取管道的步骤，请参阅the section called “创建管道”。

子管道

您可以在 YAML 配置文件中定义子管道。每个子管道都由一个来源、一个缓冲区、零个或多个处理器以及一个或多个接收器组成。您可以在单个 YAML 文件中定义多个子管道，每个子管道都有唯一的来源、处理器和接收器。为了便于监控 CloudWatch 和其他服务，我们建议您指定一个不同于其所有子管道的管道名称。

您可以在单个 YAML 文件中将多个子管道串在一起，这样一个子管道的源是另一个子管道，而其接收器是第三个子管道。有关示例，请参阅the section called “OpenTelemetry 收藏家”。

来源

子管道的输入组件。它定义了管道使用记录的机制。源可以处理事件，其方法是通过 HTTPS 接收事件，或从 Amazon S3 等外部端点读取事件。源有两种类型：基于推送的源和基于拉取的源。基于推送的来源（例如 HTTP 和OTel 日志）将记录流式传输到摄取端点。基于拉取的源（例如 OTel trace 和 S3）从源中提取数据。

处理器

中间处理单元，可以在将记录发布到接收器之前对其进行筛选、转换和扩充为所需格式。处理器是管道的可选组件。如果您未定义处理器，则记录将以源文件中定义的格式发布。可以有多个处理器。管道按照定义处理器的顺序来运行处理器。

sink

子管道的输出组件。它定义了子管道向其发布记录的一个或多个目的地。 OpenSearch 摄取支持 OpenSearch 服务域作为接收器。它还支持子管道作为接收器。这意味着您可以在单个 OpenSearch Ingestion 管道（YAML 文件）中将多个子管道串在一起。不支持将自行管理的 OpenSearch 集群作为接收器。

Buffer

处理器的一部分，在源和接收器之间充当缓冲层。您无法在管道中手动配置缓冲区。 OpenSearch 摄取使用默认的缓冲区配置。

路线

处理器的一部分，可让管道作者仅向不同接收器发送符合特定条件的事件。

有效的子管道定义必须包含源和接收器。有关每个管道元素的更多信息，请参阅配置参考。

Amazon OpenSearch Ingestion 的好处

OpenSearch 摄取有以下主要好处：

- 无需您手动管理自行预调配的管道。
- 根据您定义的容量限制自动扩缩管道。
- 通过安全补丁和错误补丁让您的管道保持最新状态。
- 提供将管道连接到您的虚拟私有云 (VPC) 的选项，以增加安全层。
- 可让您停止和启动管道以控制成本。
- 为常见用例提供管道配置蓝图，以帮助您更快地启动和运行。
- 允许您通过各种 Amazon SDKs 和 Inge OpenSearch stion API 以编程方式与您的管道进行交互。
- 支持 Amazon 中的性能监控 CloudWatch 和 CloudWatch 日志中的错误记录。

Amazon OpenSearch Ingestion 的局限性

OpenSearch 摄取有以下限制：

- 您只能将数据提取到运行 OpenSearch 1.0 或更高版本或 Elasticsearch 6.8 或更高版本的域中。如果您使用的是OTel 跟踪源，我们建议您使用 Elasticsearch 7.9 或更高版本，以便您可以使用仪表板插件。OpenSearch
- 如果管道正在写入 VPC 内的 OpenSearch 服务域，则必须在与该域 Amazon Web Services 区域 相同的环境中创建管道。
- 您只能在管道定义中配置单个数据来源。
- 您不能将自行管理的 OpenSearch 集群指定为接收器。
- 您无法将自定义端点指定为接收器。您仍然可以写入启用了自定义端点的域，但必须指定其标准端点。
- 您无法将选择加入区域内的资源指定为来源或接收器。
- 在管道配置中可以包含的参数有一些限制。有关更多信息，请参阅 the section called “配置要求和限制”。

支持的 Data Prepper 版本

OpenSearch Ingestion 目前支持以下主要版本的 Data Prepper：

- 2.x

使用代码编辑器创建管道时，请使用必填`version`选项指定要使用的 Data Prepper 的主要版本。例如，`version: "2"`。 OpenSearch Ingestion 会检索该主要版本支持的最新次要版本，并使用该版本配置管道。

如果您不使用代码编辑器创建管道， OpenSearch Ingestion 会自动使用支持的最新版本配置您的管道。

目前， OpenSearch Ingestion 使用版本 2.7 的 Data Prepper 配置管道。有关更多信息，请参阅 2.7 发布说明。 OpenSearch Ingestion 并不支持特定主要版本的每个次要版本。

更新管道配置时，如果支持新的次要版本的 Data Prepper， OpenSearch Ingestion 会自动将管道升级到工作流配置中指定的主要版本的最新支持次要版本。例如，您的工作流配置`version: "2"`中可能有， OpenSearch Ingestion 最初使用版本 2.6.0 配置了管道。添加对版本 2.7.0 的支持并更改工作流配置后， OpenSearch Ingestion 会将管道升级到 2.7.0 版。此过程可让您的管道及时获取最新错误修复和性能改进。 OpenSearch 除非您在工作流配置中手动更改该`version`选项，否则 Ingestion 无法更新管道的主要版本。有关更多信息，请参阅 the section called “更新管道”。

在 Amazon OpenSearch Ingestion 中扩展管道

OpenSearch Ingestion 会根据您指定的最小和最大摄取 OpenSearch 计算单位（摄取）自动扩展管道容量。 OCUs这样就无需手动配置和管理。

每个 Ingestion OCU 都是大约 15 GiB 内存和 2 v 的组合。CPUs 您可以为管道指定最小和最大 OCU 值， OpenSearch Ingestion 会根据这些限制自动扩展您的管道容量。

在创建管道时，您可以指定以下值：

- 最小容量 — 管道可以将容量减少到这个摄取 OCUs次数。指定的最小容量也是管道的起始容量。
- 最大容量-管道可以将容量增加到该摄取 OCUs次数。

确保管道的最大容量足够大，可以应对工作负载峰值，而且最小容量足够低，可以在管道不忙碌时最大限度地降低成本。根据您的设置， OpenSearch Ingestion 会自动调整管道的摄取数量，以处理采集 OCUs 工作负载。在任何特定时间，您只需为管道正在使用的 Ingestion OCUs 收取费用。

分配给您的 OpenSearch Ingestion 管道的容量会根据管道的处理要求和客户端应用程序生成的负载向上和向下扩展。当容量受到限制时， OpenSearch Ingestion 会通过分配更多计算单元（GiB 内存）来扩大规模。当您的管道处理较小的工作负载或根本不处理数据时，它可以缩减到最低配置的 Ingestion OCUs。

您可以指定至少 1 个摄入 OCU，为无状态管道指定最多 96 个摄入， OCUs 为有状态管道指定最多 48 个摄入。 OCUs 对于基于推送的来源，我们建议至少 2 次摄取 OCUs 。启用永久缓冲后，您可以指定最少 2，最大为 384 Ingestion。 OCUs

一个具有单一源、一个简单的 grok 模式和一个接收器的标准日志管道，每个计算单位的支持可达每秒最多 2 MiB。对于具有多个处理器的更为复杂的日志管道，每个计算单位支持的摄取负载可能更少。根据管道容量和资源利用率， OpenSearch Ingestion 扩展过程开始了。

为确保高可用性，Ingestion OCUs 分布在各个可用区 () AZs。的数量 AZs 取决于您指定的最小容量。

例如，如果您至少指定 2 个计算单元，则在任何给定时间 OCUs 使用的摄取均匀分布在 2 个计算单元。 AZs如果您至少指定 3 个或更多计算单元，则摄取将平均分布在 3 OCUs 个计算单元。 AZs我们建议您至少预配置两次 Ingestion，以确保采集管道 OCUs 的可用性达到 99.9%。

当管道处于`Create failed`、`Creating`、和状态 OCUs 时，您无需支付摄取费用。`Deleting` `Stopped`

有关配置和检索管道容量设置的说明，请参阅the section called “创建管道”。

OpenSearch 摄取定价

在任何特定时间，无论是否有数据流经管道 OCUs ，您都只需为分配给管道的摄取数量付费。OpenSearch Ingestion 可根据使用情况向上或向下扩展管道容量，从而立即适应您的工作负载。

有关全部定价详情，请参阅 Amazon OpenSearch 服务定价。

支持的 Amazon Web Services 区域

OpenSearch 该 OpenSearch 服务的子集提供摄取功能 Amazon Web Services 区域 ，可在中获得。有关支持区域的列表，请参阅中的亚马逊 OpenSearch 服务终端节点和配额Amazon Web Services 一般参考。

在 Amazon OpenSearch Ingestion 中设置角色和用户

Amazon OpenSearch Ingestion 使用各种权限模型和 IAM 角色来允许源应用程序写入管道，并允许管道写入接收器。在开始提取数据之前，您需要根据自己的用例创建一个或多个具有具体权限的 IAM 角色。

至少需要以下角色才能成功设置管道。

名称	描述
管道角色	管道角色为管道提供从源读取和写入域或集合接收器所需的权限。您可以手动创建管道角色，也可以让 OpenSearch Ingestion 为您创建该角色。
提取角色	提取角色包含对管道资源的 `osis:Ingest` 权限。此权限允许基于推送的源将数据摄取到管道中。

下图演示了典型的管道设置，其中诸如 Amazon S3 或 Fluent Bit 之类的数据来源使用不同的账户写入管道。在这种情况下，客户端需要担任提取角色才能访问管道。有关更多信息，请参阅 the section called “跨账户提取”。

欲了解简易设置指南，请参阅the section called “教程：将数据摄取到域”。

主题

- the section called “管道角色”
- the section called “提取角色”
- the section called “跨账户提取”

管道角色

管道需要一定的权限才能从其源代码读取和写入其接收器。这些权限取决于客户端应用程序或 Amazon Web Services 服务 正在写入管道的应用程序，以及接收器是 OpenSearch 服务域、 OpenSearch 无服务器集合还是 Amazon S3。此外，管道可能需要从源应用程序物理提取数据的权限（如果源是基于拉取的插件），以及写入 S3 死信队列的权限（如果启用）。

创建管道时，您可以选择指定手动创建的现有 IAM 角色，也可以让 OpenSearch Ingestion 根据您选择的源和接收器自动创建管道角色。下图显示了如何在中指定管道角色 Amazon Web Services Management Console。

主题

- 自动创建管道角色
- 手动创建管道角色

自动创建管道角色

你可以选择让 OpenSearch Ingestion 为你创建管道角色。它会根据配置的源和接收器自动识别角色所需的权限。它使用您输入的前缀和后缀`OpenSearchIngestion-`创建一个 IAM 角色。例如，如果您输入`PipelineRole`作为后缀， OpenSearch Ingestion 会创建一个名为的角色。`OpenSearchIngestion-PipelineRole`

自动创建管道角色可以简化设置过程并降低出现配置错误的可能性。通过自动创建角色，可以避免手动分配权限，从而确保应用正确的策略而不会出现安全配置错误的风险。这还可以通过实施最佳实践来节省时间并增强安全合规性，同时确保多个管道部署的一致性。

您只能让 OpenSearch Ingestion 在中自动创建管道角色。 Amazon Web Services Management Console如果您使用的是 Amazon CLI、 OpenSearch Ingestion API 或其中一个 SDKs，则必须指定手动创建的管道角色。

要让 OpenSearch Ingestion 为您创建角色，请选择创建并使用新的服务角色。

> **Important**
>
> 您仍然需要手动修改域或集合访问策略以授予对管道角色的访问权限。对于使用精细访问控制的域，您还必须将管道角色映射到后端角色。您可以在创建管道之前或之后执行这些步骤。有关说明，请参阅以下主题：
>
> - 为域配置数据访问权限
> - 为馆藏配置数据和网络访问权限

手动创建管道角色

如果您需要对权限进行更多控制以满足特定的安全或合规性要求，则可能更愿意手动创建管道角色。手动创建允许您定制角色以适应现有基础设施或访问管理策略。您也可以选择手动设置来将该角色与其他角色集成， Amazon Web Services 服务 或者确保其符合您的独特运营需求。

要选择手动创建的管道角色，请选择使用现有 IAM 角色并选择现有角色。该角色必须具有从选定源接收数据并写入所选接收器所需的所有权限。以下各节概述了如何手动创建管道角色。

主题

- 从来源读取的权限
- 写入域接收器的权限
- 向集合接收器写入的权限
- 写入 Amazon S3 或死信队列的权限

从来源读取的权限

OpenSearch 摄取管道需要权限才能读取和接收来自指定来源的数据。例如，对于亚马逊 DynamoDB 来源，它需要诸如和之类的权限。`dynamodb:DescribeTable` `dynamodb:DescribeStream`有关常见来源（例如 Amazon S3、Fluent Bit 和 Collecto OpenTelemetry r）的管道角色访问策略示例，请参阅the section called “集成管道”。

写入域接收器的权限

In OpenSearch gestion 管道需要权限才能写入配置为其接收器的 OpenSearch 服务域。这些权限包括能够描述域以及向其发送 HTTP 请求。公有域和 VPC 域的这些权限相同。有关创建管道角色并在域访问策略中指定管道角色的说明，请参阅允许管道访问域。

向集合接收器写入的权限

OpenSearch 摄取管道需要权限才能写入配置为其 OpenSearch 接收器的无服务器集合。这些权限包括能够描述集合以及向其发送 HTTP 请求。

首先，请确保您的管道角色访问策略授予所需的权限。然后，将此角色包含在数据访问策略中，并为其提供在集合中创建索引、更新索引、描述索引和写入文档的权限。有关完成其中每个步骤的说明，请参阅允许管道访问集合。

写入 Amazon S3 或死信队列的权限

如果您将 Amazon S3 指定为管道的接收目标，或者启用死信队列 (DLQ)，则管道角色必须允许其访问您指定为目标的 S3 存储桶。

为提供 DLQ 访问权限的管道角色附加单独的权限策略。至少必须向该角色授予对存储桶资源的`S3:PutObject`操作权限：

JSON

```
{
```

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

提取角色

摄取角色是一个 IAM 角色，它允许外部服务安全地与 OpenSearch 摄取管道交互并向其发送数据。对于基于推送的来源（例如 Amazon Security Lake），此角色必须授予将数据推送到管道的权限，包括`osis:Ingest`。对于基于拉取的源，例如 Amazon S3，角色必须允许 OpenSearch Ingestion 代入并使用必要的权限访问数据。

主题

- 基于推送的源的摄取角色
- 基于拉取的源的摄取角色
- 跨账户提取

基于推送的源的摄取角色

对于基于推送的来源，数据将从其他服务（例如 Amazon Security Lake 或 Amazon DynamoDB）发送或推送到摄取管道。在这种情况下，摄取角色至少需要获得与管道交互的`osis:Ingest`权限。

以下 IAM 访问策略演示了如何向摄取角色授予此权限：

JSON

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

```
        "osis:Ingest"
      ],
      "Resource": "arn:aws:osis:region:account-id:pipeline/pipeline-name/*"
    }
  ]
}
```

基于拉取的源的摄取角色

对于基于拉取的源， OpenSearch Ingestion 管道会主动从外部来源（例如 Amazon S3）提取或获取数据。在这种情况下，管道必须扮演一个 IAM 管道角色，该角色授予访问数据源的必要权限。在这些场景中，摄取角色与管道角色同义。

该角色必须包括允许 OpenSearch Ingestion 担任该角色的信任关系以及特定于数据源的权限。有关更多信息，请参阅 the section called “从来源读取的权限”。

跨账户提取

您可能需要从其他 Amazon Web Services 账户渠道（例如应用程序帐户）将数据提取到管道中。要配置跨账户提取，请在与管道相同的账户中定义一个提取角色，并在提取角色和应用程序账户之间建立信任关系：

JSON

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

然后，将您的应用程序配置为担任提取角色。应用程序账户必须向应用程序角色授予管道账户中摄取角色的AssumeRole权限。

有关详细步骤和 IAM policy 示例，请参阅the section called “提供跨账户摄取访问权限”。

授予 Amazon OpenSearch Ingestion 管道访问域名的权限

Amazon OpenSearch Ingestion 管道需要权限才能写入配置为其接收器的 OpenSearch 服务域。要提供访问权限，您需要为一个 Amazon Identity and Access Management (IAM) 角色配置一个限制性权限策略，该策略限制了对管道向其发送数据的域的访问权限。例如，您可能希望将摄取管道限制为仅含支持其用例所需的域和索引。

> **Important**
>
> 您可以选择手动创建管道角色，也可以让 OpenSearch Ingestion 在创建管道期间为您创建该角色。如果您选择自动创建角色， OpenSearch Ingestion 会根据您选择的来源和接收器将所有必需的权限添加到管道角色访问策略中。它使用您输入的前缀`OpenSearchIngestion-`和后缀在 IAM 中创建管道角色。有关更多信息，请参阅 the section called “管道角色”。
> 如果您让 OpenSearch Ingestion 为您创建管道角色，则在创建管道之前或之后，您仍然需要将该角色包含在域访问策略中，并将其映射到后端角色（如果该域使用精细访问控制）。有关说明，请参阅步骤 2。

主题

- 步骤 1：创建管道角色
- 步骤 2：为域配置数据访问权限

步骤 1：创建管道角色

管道角色必须附加权限策略，允许其向域接收器发送数据。它还必须具有允许 OpenSearch Ingestion 担任该角色的信任关系。有关如何附加角色策略的说明，请参阅 IAM 用户指南中的添加 IAM 身份权限。

以下示例策略演示了您可以在管道角色中为其提供写入单个域的最低权限：

JSON

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

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

如果计划重用角色写入多个域，则可以将域名替换为通配符 (*) 来扩大策略范围。

该角色必须具有以下信任关系，这允许 OpenSearch Ingestion 担任管道角色：

JSON

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

步骤 2：为域配置数据访问权限

为了使管道能够将数据写入域，该域必须具有允许管道角色访问该域的域级访问策略。

以下示例域访问策略允许名为的管道角色`pipeline-role`向名为的域写入数据`ingestion-domain`：

JSON

```
{
```

```
  "Version": "2012-10-17",
  "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/*"
    }
  ]
}
```

映射管道角色（仅适用于使用精细访问控制的域）

如果您的域使用精细访问控制进行身份验证，则需要采取额外步骤，为管道提供域访问权限。步骤因域配置而异：

- 场景 1：不同的主角色和管道角色 — 如果您使用 IAM Amazon 资源名称 (ARN) 作为主用户，并且该名称与管道角色不同，则需要将管道角色映射到 OpenSearch`all_access`后端角色。这会将管道角色添加为其他主用户。有关更多信息，请参阅其他主用户。
- 场景 2：内部用户数据库中的主用户-如果您的域使用内部用户数据库中的主用户和 OpenSearch 仪表板的 HTTP 基本身份验证，则无法将主用户名和密码直接传递到工作流配置中。而是将管道角色映射到后 OpenSearch`all_access`端角色。这会将管道角色添加为其他主用户。有关更多信息，请参阅其他主用户。
- 场景 3：相同的主角色和管道角色（不常见）— 如果您使用的 IAM ARN 作为主用户，并且它与您用作管道角色的 ARN 相同，则无需采取任何进一步的操作。管道具有写入域所需的权限。这种情况并不常见，因为大多数环境都使用管理员角色或其他角色作为主角色。

下图显示了如何将管道角色映射到后端角色：

授予 Amazon OpenSearch Ingestion 管道访问集合的权限

Amazon OpenSearch Ingestion 管道可以写入 OpenSearch 无服务器公共集合或 VPC 集合。要提供对集合的访问权限，您需要为一个 Amazon Identity and Access Management (IAM) 管道角色配置权限策略，以授予对集合的访问权限。管道扮演此角色是为了签署对 OpenSearch 无服务器集合接收器的请求。

> **Important**
>
> 您可以选择手动创建管道角色，也可以让 OpenSearch Ingestion 在创建管道期间为您创建该角色。如果您选择自动创建角色， OpenSearch Ingestion 会根据您选择的来源和接收器将所有必需的权限添加到管道角色访问策略中。它使用您输入的前缀`OpenSearchIngestion-`和后缀在 IAM 中创建管道角色。有关更多信息，请参阅 the section called “管道角色”。
>
> 如果您让 OpenSearch Ingestion 为您创建管道角色，则在创建管道之前或之后，您仍然需要将该角色包含在集合的数据访问策略中。有关说明，请参阅步骤 2。

在创建管道期间， OpenSearch Ingestion 会在管道和 OpenSearch Serverless 集合之间创建 Amazon PrivateLink 连接。来自管道的所有流量都将通过此 VPC 端点并路由到集合。要连接到集合，必须通过网络访问策略向端点授予访问该集合的权限。

主题

- 步骤 1：创建管道角色
- 步骤 2：为馆藏配置数据和网络访问权限

步骤 1：创建管道角色

管道角色必须附加权限策略，允许其向集合接收器发送数据。它还必须具有允许 OpenSearch Ingestion担任该角色的信任关系。有关如何附加角色策略的说明，请参阅 IAM 用户指南中的添加 IAM 身份权限。

以下示例策略演示了您可以在管道角色访问策略中为其提供的最低权限，以便其写入集合：

JSON

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

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

该角色必须具有以下信任关系，这允许 OpenSearch Ingestion 担任该角色：

JSON

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

步骤 2：为馆藏配置数据和网络访问权限

使用以下设置创建 OpenSearch 无服务器集合。有关创建集合的说明，请参阅the section called “创建集合”。

数据访问策略

为集合创建一个向管道角色授予所需权限的数据访问策略。例如：

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

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

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

网络访问策略

您在 OpenSearch Serverless 中创建的每个集合都至少有一个与之关联的网络访问策略。网络访问策略决定了是否可以通过互联网从公共网络访问该集合，还是必须以私有方式访问该集合。更多有关网络策略的信息，请参阅the section called “网络访问”。

在网络访问策略中，您只能指定 OpenSearch 无服务器托管的 VPC 终端节点。有关更多信息，请参阅the section called “VPC 端点”。但是，为了使管道能够写入集合，该策略还必须授予对 OpenSearch Ingestion 在管道和集合之间自动创建的 VPC 终端节点的访问权限。因此，如果您选择 OpenSearch Serverless 集合作为管道的目标接收器，则必须在 “网络策略名称” 字段中输入关联的网络策略的名称。

在创建管道期间， OpenSearch Ingestion 会检查指定的网络策略是否存在。如果它不存在，则 OpenSearch Ingestion 会创建它。如果确实存在， OpenSearch Ingestion 会通过向其添加新规则来对其进行更新。添加的规则会授予对连接管道和集合的 VPC 端点的访问权限。

例如：

```
{
```

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

在控制台中， OpenSearch Ingestion 添加到您的网络策略中的所有规则都命名为 Create b y Data Prepper：

> **Note**
>
> 通常，为集合指定公有访问权限的规则将会覆盖指定私有访问权限的规则。因此，如果策略已经配置了公共访问权限，那么 OpenSearch Ingestion 添加的这条新规则实际上并不会改变策略的行为。有关更多信息，请参阅 the section called “策略优先顺序”。

如果您停止或删除管道， OpenSearch Ingestion 会删除管道和集合之间的 VPC 终端节点。此外还会修改网络策略，将 VPC 端点从允许的端点列表中移除。如果您重启管道，则会重新创建 VPC 端点，并使用端点 ID 重新更新网络策略。

开始使用 Amazon OpenSearch Ingestion

Amazon OpenSearch Ingestion 支持将数据提取到托管 OpenSearch 服务域和无服务器集合中。OpenSearch 以下教程演示了完成管道设置和运行的基本步骤。

第一个教程向您展示了如何使用 Amazon OpenSearch Ingestion 配置简单的管道并将数据提取到亚马逊 OpenSearch 服务域中。

第二个教程向您展示如何使用 Amazon OpenSearch Ingestion 配置简单管道并将数据提取到 Ama OpenSearch zon Serverless 集合中。

> **Note**
>
> 如果未设置正确的权限，则管道创建将失败。创建管道之前，请参阅 the section called “设置角色和用户”，更深入地了解所需的角色。

主题

- 教程：使用 Amazon Ingestion 将数据提取到域中 OpenSearch
- 教程：使用 Amazon Ingestion 将数据提取到集合中 OpenSearch

教程：使用 Amazon Ingestion 将数据提取到域中 OpenSearch

本教程向您展示如何使用 Amazon OpenSearch Ingestion 配置简单的管道并将数据提取到亚马逊 OpenSearch 服务域中。管道是 OpenSearch Ingestion 预置和管理的资源。您可以使用管道筛选、丰富、转换、标准化和聚合数据，以便在 S OpenSearch ervice 中进行下游分析和可视化。

本教程将指导您完成快速启动并运行管道的基本步骤。有关更全面的说明，请参阅the section called “创建管道”。

在本教程中，您将完成以下步骤：

1. 创建域。
2. 创建管道。
3. 摄取一些样本数据。

在本教程中，您将创建以下资源：

- 管道写入ingestion-domain的名为的域
- 名为 ingestion-pipeline 的管道

所需的权限

要完成本教程，您的用户或角色必须附加具有以下最低权限的基于身份的策略。这些权限允许您创建管道角色并附加策略（iam:Create*和iam:Attach*）、创建或修改域（es:*）以及使用管道（osis:*）。

JSON

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

步骤 1：创建管道角色

首先，创建管道将扮演的角色以访问 OpenSearch 服务域接收器。在本教程后面的部分中，您将在管道配置中包含此角色。

要创建管道角色

1. 打开 Amazon Identity and Access Management 控制台，网址为https://console.aws.amazon.com/iamv2/。
2. 选择策略，然后选择创建策略。
3. 在本教程中，您将数据摄取到名为 `ingestion-domain` 的域，该域将在下一步中创建。选择 JSON，然后将下面的策略粘贴到编辑器中。使用您的账户 ID 替换 *`your-account-id`*，并在必要时修改区域。

   JSON

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

   如果您想将数据写入现有域，请使用您的域名替换 `ingestion-domain`。

> **Note**
>
> 为了简化本教程，我们使用了广泛访问策略。但是，在生产环境中，建议您对管道角色应用严格一些的访问策略。有关提供最低所需权限的策略示例，请参阅 the section called “向管道授予访问域的权限”。

4. 依次选择下一步、下一步，然后命名您的策略 pipeline-policy。
5. 选择创建策略。
6. 然后，创建一个角色并将策略附加到该角色。选择 角色，然后选择 创建角色。
7. 选择自定义信任策略，并将以下策略粘贴到编辑器中：

   JSON

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

8. 选择下一步。然后搜索并选择（您刚刚创建的）pipeline-policy。
9. 选择 “下一步” 并命名角色PipelineRole。
10. 选择创建角色。

请记住角色的 Amazon 资源名称（ARN）（例如，`arn:aws:iam::your-account-id:role/PipelineRole`）。您创建管道时需要用到。

步骤 2：创建域

首先，创建一个名为的域，`ingestion-domain`用于将数据采集到其中。

在https://console.aws.amazon.com/aos/家中导航至 Amazon S OpenSearch ervice 控制台，创建一个满足以下要求的域名：

- 正在运行 OpenSearch 1.0 或更高版本，或者运行 Elasticsearch 7.4 或更高版本
- 使用公有访问
- 不使用精准访问控制

> **Note**
>
> 这些要求是为了让本教程简单易懂。在生产环境中， and/or 您可以使用精细的访问控制来配置具有 VPC 访问权限的域。要使用精细访问控制，请参阅映射管道角色。

该域必须具有授予 `OpenSearchIngestion-PipelineRole` IAM 角色权限的访问策略，OpenSearch 服务将在下一步中为您创建该角色。管道将扮演此角色，以便将数据发送到域接收器。

确保该域具有以下域级访问策略，该策略授予管道角色访问该域的权限。替换成您自己的区域和账户 ID：

JSON

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

有关创建域级访问策略的更多信息，请参阅。the section called “基于资源的策略”

如果您已经创建了域，请修改其现有访问策略以向 `OpenSearchIngestion-PipelineRole` 提供上述权限。

步骤 3：创建管道

现在您已经有了域，可以创建管道了。

创建管道

1. 在亚马逊 OpenSearch 服务控制台中，从左侧导航窗格中选择 Pipelin es。
2. 选择 Create pipeline（创建管道）。
3. 选择空白管道，然后选择选择蓝图。
4. 在本教程中，我们将创建一个使用 HTTP 源插件的简单管道。插件接受 JSON 数组格式的日志数据。我们将指定单个 OpenSearch 服务域作为接收器，并将所有数据提取到`application_logs`索引中。

   在 “源” 菜单中，选择 “HTTP”。在路径中，输入 /logs。
5. 在本教程中，为简单起见，我们将配置管道的公共访问权限。对于源网络选项，请选择公共访问。有关配置 VPC 访问权限的更多信息，请参阅 the section called “为管道配置 VPC 访问”。
6. 选择下一步。
7. 对于处理者，输入日期并选择添加。
8. 启用 “从收到时间开始”。将所有其他设置保留为默认设置。
9. 选择下一步。
10. 配置接收器详细信息。对于OpenSearch 资源类型，请选择托管群集。然后选择您在上一节中创建的 OpenSearch 服务域。

    在索引名称中，输入应用程序日志。 OpenSearch 如果此索引尚不存在，Ingestion 会自动在域中创建该索引。
11. 选择下一步。
12. 命名管道摄取管道。将容量设置保留为其默认值。
13. 对于管道角色，选择创建并使用新的服务角色。管道角色为管道提供写入域接收器并从基于拉取的源中读取所需的权限。选择此选项即表示您允许 OpenSearch Ingestion 为您创建角色，而不是在 IAM 中手动创建角色。有关更多信息，请参阅 the section called “设置角色和用户”。
14. 对于服务角色名称后缀，请输入PipelineRole。在 IAM 中，角色的格式为`arn:aws:iam::`*`your-account-id`*`:role/OpenSearchIngestion-`**`PipelineRole`**。
15. 选择下一步。检查您的管道配置，然后选择创建管道。管道需要 5-10 分钟才能变为活动状态。

步骤 4：摄取一些示例数据

当管道状态为 `Active` 时，您可以开始将数据摄取到管道。您必须使用 Signature 版本 4 对向管道发出的所有 HTTP 请求进行签名。使用诸如 Postman 或 awscurl 之类的 HTTP 工具向管道发送一些数据。与将数据直接索引到域一样，将数据提取到管道始终需要 IAM 角色或 IAM 访问密钥和私有密钥。

> **Note**
>
> 签署请求的主体必须具有 `osis:Ingest` IAM 权限。

首先，从管道设置页面获取摄取 URL：

然后，摄取一些示例数据。以下请求使用 awscurl 向管道发送单个日志文件：

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

您现在会看到 `200 OK` 响应。如果您遇到身份验证错误，可能是因为您从与管道不同的账户提取数据。请参阅the section called “修复权限问题”。

现在，查询 `application_logs` 索引以确保成功提取您的日志条目：

```
awscurl --service es --region us-east-1 \
     -X GET \
```

```
    https://search-ingestion-domain.us-east-1.es.amazonaws.com/application_logs/
_search | json_pp
```

示例响应:


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

修复权限问题

如果您按照教程中的步骤进行操作，但在尝试提取数据时仍然看到身份验证错误，则可能是因为写入管道的角色与管道本身 Amazon Web Services 账户 不同。在这种情况下，您需要创建并担任一个角色，该角色让您专门用来摄取数据。有关说明，请参阅the section called “提供跨账户摄取访问权限”。

相关资源

本教程介绍了一个通过 HTTP 摄取单个文档的简单用例。在生产场景中，您将配置客户端应用程序（例如 Fluent Bit、Kubernetes 或 Collect OpenTelemetry or）以将数据发送到一个或多个管道。您的管道可能比本教程展示的简单示例复杂得多。

要开始配置您的客户端并摄取数据，请参阅以下资源：

- 创建和管理管道
- 将您的客户端配置为向 OpenSearch Ingestion 发送数据
- Data Prepper 文档

教程：使用 Amazon Ingestion 将数据提取到集合中 OpenSearch

本教程向您展示如何使用 Amazon OpenSearch Ingestion 配置简单管道并将数据提取到 Ama OpenSearch zon Serverless 集合中。管道是 OpenSearch Ingestion 预置和管理的资源。您可以使用管道筛选、丰富、转换、标准化和聚合数据，以便在 S OpenSearch ervice 中进行下游分析和可视化。

有关演示如何将数据提取到已配置的 OpenSearch 服务域的教程，请参阅。the section called “教程：将数据摄取到域”

您将在本教程中完成以下步骤:.

1. 创建集合。
2. 创建管道。
3. 摄取一些样本数据。

在本教程中，您将创建以下资源：

- 管道将写入的名为 `ingestion-collection` 的集合

- 名为 `ingestion-pipeline-serverless` 的管道

所需的权限

要完成本教程，您的用户或角色必须附加具有以下最低权限的基于身份的策略。这些权限允许您创建管道角色并附加策略（`iam:Create*`和`iam:Attach*`）、创建或修改集合（`aoss:*`）以及使用管道（`osis:*`）。

此外，还需要多个 IAM 权限才能自动创建管道角色并将其传递给 OpenSearch Ingestion，这样它就可以向集合写入数据。

JSON

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

步骤 1：创建收藏夹

首先，创建一个集合，将数据采集到其中。将集合命名为 `ingestion-collection`。

1. 在https://console.aws.amazon.com/aos/家中导航至 Amazon OpenSearch 服务控制台。
2. 选择左侧导航窗格中的集合，然后选择创建集合。
3. 将集合命名为 ingestion-collection。
4. 对于安全，选择标准创建。
5. 在网络访问设置下，将访问类型更改为公有。
6. 所有其他设置保留为默认值，然后选择 Next（下一步）。
7. 现在，配置集合的数据访问策略。取消选择“自动匹配访问策略设置”。
8. 对于定义方法，选择 JSON，然后将下面的策略粘贴到编辑器中。此策略具有两个作用：
   - 允许管道角色写入集合。
   - 允许从集合中读取。稍后，在将一些示例数据摄取到管道后，查询集合，以确保数据已成功摄取并写入索引。

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/ingestion-collection/*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:UpdateIndex",
          "aoss:DescribeIndex",
          "aoss:ReadDocument",
          "aoss:WriteDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::your-account-id:role/OpenSearchIngestion-PipelineRole",
      "arn:aws:iam::your-account-id:role/Admin"
    ],
    "Description": "Rule 1"
```

```
  }
]
```

9. 修改`Principal`元素以包含您的 Amazon Web Services 账户 ID。对于第二个委托人，请指定一个用户或角色，以便日后可以用来查询集合。
10. 选择下一步。命名访问策略，pipeline-collection-access然后再次选择 Next。
11. 查看集合配置并选择 Submit（提交）。

步骤 2：创建管道

现在你已经有了集合，就可以创建一个管道了。

创建管道

1. 在亚马逊 OpenSearch 服务控制台中，从左侧导航窗格中选择 Pipelin es。
2. 选择 Create pipeline（创建管道）。
3. 选择空白管道，然后选择选择蓝图。
4. 在本教程中，我们将创建一个使用 HTTP 源插件的简单管道。插件接受 JSON 数组格式的日志数据。我们将指定一个 OpenSearch Serverless 集合作为接收器，并将所有数据提取到索引中。`my_logs`

   在 “源” 菜单中，选择 “HTTP”。在路径中，输入 /logs。
5. 在本教程中，为简单起见，我们将配置管道的公共访问权限。对于源网络选项，请选择公共访问。有关配置 VPC 访问权限的更多信息，请参阅 the section called “为管道配置 VPC 访问”。
6. 选择下一步。
7. 对于处理者，输入日期并选择添加。
8. 启用 “从收到时间开始”。将所有其他设置保留为默认设置。
9. 选择下一步。
10. 配置接收器详细信息。对于OpenSearch 资源类型，选择集合（无服务器）。然后选择您在上一节中创建的 OpenSearch 服务集合。

    将网络策略名称保留为默认名称。在索引名称中，输入 my_logs。 OpenSearch 如果该索引尚不存在，Ingestion 会自动在集合中创建该索引。
11. 选择下一步。
12. 为管道命名ingestion-pipeline-serverless。将容量设置保留为其默认值。

13. 对于管道角色，选择创建并使用新的服务角色。管道角色为管道提供写入集合接收器以及从基于拉取的源中读取所需的权限。选择此选项即表示您允许 OpenSearch Ingestion 为您创建角色，而不是在 IAM 中手动创建角色。有关更多信息，请参阅 the section called “设置角色和用户”。
14. 对于服务角色名称后缀，请输入PipelineRole。在 IAM 中，角色的格式为`arn:aws:iam::`*`your-account-id`*`:role/OpenSearchIngestion-`**`PipelineRole`**。
15. 选择下一步。检查您的管道配置，然后选择创建管道。管道需要 5-10 分钟才能变为活动状态。

第 3 步：摄取一些示例数据

当管道状态为 `Active` 时，您可以开始将数据摄取到管道。您必须使用 Signature 版本 4 对向管道发出的所有 HTTP 请求进行签名。使用诸如 Postman 或 awscurl 之类的 HTTP 工具向管道发送一些数据。与通过索引将数据直接引入集合一样，将数据摄取到管道始终需要 IAM 角色或者 IAM 访问密钥和私有密钥。

> **Note**
>
> 签署请求的主体必须具有 `osis:Ingest` IAM 权限。

首先，从管道设置页面获取摄取 URL：

然后，将一些示例数据发送到摄取路径。以下示例请求使用 awscurl 向管道发送单个日志文件：

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

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

您现在会看到 200 OK 响应。

现在，查询 my_logs 索引，确保成功摄取日志条目：

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

示例响应:


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

```
}
```

相关资源

本教程介绍了一个通过 HTTP 摄取单个文档的简单用例。在生产场景中，您将配置客户端应用程序（例如 Fluent Bit、Kubernetes 或 Collect OpenTelemetry or）以将数据发送到一个或多个管道。您的管道可能比本教程展示的简单示例复杂得多。

要开始配置您的客户端并摄取数据，请参阅以下资源：

- 创建和管理管道
- 将您的客户端配置为向 OpenSearch Ingestion 发送数据
- Data Prepper 文档

Amazon OpenSearch Ingestion 中的管道功能概述

Amazon OpenSearch Ingestion 预置管道，这些管道由一个源、一个缓冲区、零个或多个处理器以及一个或多个接收器组成。提取管道由 Data Prepper 作为数据引擎提供支持。有关管道各个组件的概述，请参见 the section called “重要概念”。

以下各节概述了 Amazon OpenSearch Ingestion 中一些最常用的功能。

> **Note**
>
> 该列表不是管道可用功能的详尽列表。有关管道所有可用功能的综合文档，请参阅 Data Prepper 文档。请注意， OpenSearch Ingestion 对你可以使用的插件和选项施加了一些限制。有关更多信息，请参阅 the section called “支持的插件和选项”。

主题

- 持久缓冲功能
- 拆分
- 链接
- 死信队列
- 索引管理

- End-to-end 确认
- 源背压

持久缓冲功能

永久缓冲区将您的数据存储在跨多个可用区的基于磁盘的缓冲区中，以增强数据的持久性。您可以使用持久缓冲从所有支持的基于推送的来源提取数据，而无需设置独立的缓冲区。这些来源包括 HTTP 以及 OpenTelemetry 用于日志、跟踪和指标的来源。要启用永久缓冲，请在创建或更新管道时选择 “启用永久缓冲区”。有关更多信息，请参阅 the section called “创建管道”。

OpenSearch Ingestion 会动态确定 OCUs 用于持久缓冲的数量，同时考虑数据源、流式传输转换和接收目标。由于它会 OCUs 为缓冲分配一些资源，因此您可能需要增加最小值和最大值 OCUs 以保持相同的摄取吞吐量。管道在缓冲区中保留数据最多 72 小时。

如果您为管道启用永久缓冲，则默认的最大请求负载大小如下所示：

- HTTP 来源 — 10 MB
- OpenTelemetry 来源 — 4 MB

对于 HTTP 源，您可以将最大负载大小增加到 20 MB。请求负载大小包括整个 HTTP 请求，其中通常包含多个事件。每个事件的容量不能超过 3.5 MB。

具有持久缓冲功能的管道在计算单元和缓冲单元之间拆分配置的流水线单元。如果管道使用 CPU 密集型处理器（例如 grok、键值或拆分字符串），则它将以 1:1 的比例分配单位。 buffer-to-compute否则，它以 3:1 的比例分配它们，始终偏向于计算单位。

例如：

- 具有 grok 和 2 个最大单位的管道 — 1 个计算单元和 1 个缓冲单元
- 具有 grok 和 5 个最大单位的管道 — 3 个计算单元和 2 个缓冲单元
- 没有处理器且最多 2 个单元的流水线 — 1 个计算单元和 1 个缓冲单元
- 没有处理器且最多 4 个单元的流水线 — 1 个计算单元和 3 个缓冲单元
- 具有 grok 和 5 个最大单位的管道 — 2 个计算单元和 3 个缓冲单元

默认情况下，管道使用加密 Amazon 拥有的密钥 缓冲区数据。这些管道不需要任何额外的管道角色权限。

您也可以指定客户自主管理型密钥，并将以下 IAM 权限添加到管道角色：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KeyAccess",
            "Effect": "Allow",
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "arn:aws:kms:{region}:{aws-account-
id}:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

有关更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的客户托管密钥。

> **Note**
>
> 如果您禁用永久缓冲，您的管道将完全在内存缓冲上运行。

拆分

您可以将 OpenSearch Ingestion 管道配置为将传入事件拆分为子管道，从而允许您对同一个传入事件执行不同类型的处理。

以下示例管道将传入事件拆分到两个子管道。每个子管道都使用自己的处理器来丰富和操作数据，然后将数据发送到不同的 OpenSearch 索引。

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

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

logs_enriched_one_pipeline:
  source:
    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"
```

链接

您可以将多个子管道链接在一起，以便分块执行数据处理和扩充。换句话说，你可以在一个子管道中使用一定的处理能力来丰富传入的事件，然后将其发送到另一个子管道，使用不同的处理器进行进一步丰富，最后将其发送到其 OpenSearch 接收器。

在以下示例中，`log_pipeline`子管道使用一组处理器丰富传入的日志事件，然后将该事件发送到名为的 OpenSearch 索引。`enriched_logs`管道将相同的事件发送到`log_advanced_pipeline`子管道，子管道对其进行处理并将其发送到名`enriched_advanced_logs`为的其他 OpenSearch 索引。

```
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"
```

死信队列

死信队列 (DLQs) 是管道未能写入接收器的事件的目的地。在 OpenSearch Ingestion 中，您必须指定一个具有相应写入权限的 Amazon S3 存储桶才能用作 DLQ。您可以向管道中的每个接收器添加 DLQ 配置。当管道遇到写入错误时，它会在配置的 S3 存储桶中创建 DLQ 对象。DLQ 对象作为一组失败事件存在于 JSON 文件中。

满足以下任意条件时，管道会向 DLQ 写入事件：

- OpenSearch 接收器的最大重试次数已用完。 OpenSearch 要进行此设置，摄取至少需要 16。

- 由于错误情况，接收器正在拒绝事件。

配置

要为子管道配置死信队列，请在配置接收器目标时选择 “启用 S3 DLQ”。然后，为队列指定所需的设置。有关更多信息，请参阅 Data Prepper DLQ 文档中的配置。

写入此 S3 DLQ 的文件具有以下命名模式：

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

有关手动配置管道角色以允许访问 DLQ 写入的 S3 存储桶的说明，请参阅。the section called “写入 Amazon S3 或死信队列的权限”

示例

考虑以下示例 DLQ 文件：

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

以下是未能写入接收器并发送到 DLQ S3 存储桶进行进一步分析的数据示例：

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

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

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

索引管理

Amazon OpenSearch Ingestion 具有许多索引管理功能，包括以下功能。

创建索引

您可以在管道接收器中指定索引名称， OpenSearch Ingestion 在置备管道时会创建索引。如果索引已经存在，管道会将其用于索引传入事件。如果您停止并重启管道，或者更新其 YAML 配置，如果这些索引尚不存在，则管道会尝试创建新的索引。管道始终不会删除索引。

以下示例为接收器在预调配管道时创建两个索引：

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

生成索引名称和模式

您可以使用传入事件字段的变量来生成动态索引名称。在接收器配置中，使用格式 `string ${}` 表示字符串插值，并使用 JSON 指针从事件中提取字段。`index_type` 的选项是 `custom` 或 `management_disabled`。由于 OpenSearch 域名和 OpenSearch 无服务器集合`management_disabled`的`index_type`默认值为，因此可以将其保留为未设置。`custom`

例如，以下管道从传入事件中选择 `metadataType` 字段以生成索引名称。

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

以下配置继续每天或每小时生成一个新索引。

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

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

索引名称也可以是以日期-时间模式作为后缀的纯字符串，例如 `my-index-%{yyyy.MM.dd}`。当接收器向发送数据时 OpenSearch，它会将日期时间模式替换为 UTC 时间，并为每天创建一个新索引，例如。`my-index-2022.01.25`有关更多信息，请参阅DateTimeFormatter课程。

该索引名称也可以是带有/不带日期-时间样式后缀的格式化字符串，例如 `my-${index}-name`。当接收器向发送数据时 OpenSearch，它会将该`"${index}"`部分替换为正在处理的事件中的值。如果格式为 `"${index1/index2/index3}"`，则使用事件中的值代替字段 `index1/index2/index3`。

正在生成文档 IDs

管道可以在为文档编制索引时生成文档 ID OpenSearch。它可以 IDs 从传入事件中的字段中推断出这些文档。

此示例使用传入事件的 `uuid` 字段生成文档 ID。

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

在以下示例中，添加条目处理器合并传入事件的字段 `uuid` 和 `other_field` 以生成文档 ID。

该`create`操作可确保具有相同内容的文档 IDs 不会被覆盖。管道会丢弃重复的文档，而不会出现任何重试或 DLQ 事件。由于使用此操作的管道作者的目的在于避免更新现有文档，因而这一预期十分合理。

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

您可能需要将事件的文档 ID 设置为子对象中的字段。在以下示例中，s OpenSearch ink 插件使用子对象`info/id`生成文档 ID。

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

鉴于以下事件，管道将生成一个 `_id` 字段设置为 `json001` 的文档：

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

正在生成路由 IDs

你可以使用 sin OpenSearch k 插件中的`routing_field`选项将文档路由属性 (`_routing`) 的值设置为来自传入事件的值。

路由支持 JSON 指针语法，因此嵌套字段也可用，而不仅仅是顶级字段。

```
sink:
  - opensearch:
       ...
```

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

鉴于以下事件，插件将生成一个 `_routing` 字段设置为 `abcd` 的文档：

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

有关创建供管道在创建索引时使用的索引模板的说明，请参阅索引模板。

End-to-end 确认

OpenSearch Ingestion 使用end-to-end确认功能跟踪无状态管道中从源到接收器的传输，从而确保数据的持久性和可靠性。

> **Note**
>
> 目前，只有 S 3 源插件支持 end-to-end确认。

通过 end-to-end确认，管道源插件会创建一个确认集来监视一批事件。当这些事件成功发送到其接收器时，它会收到肯定应答，或者当任何事件无法发送到其接收器时，它会收到否定应答。

如果管道组件出现故障或崩溃，或者源未能收到确认，则源会超时并采取必要的操作，例如重试或记录失败。如果管道配置了多个接收器或多个子管道，则只有在将事件发送到所有子管道中的所有接收器之后，才会发送事件级别确认。如果接收器配置了 DLQ，则 end-to-end确认还会跟踪写入 DLQ 的事件。

要启用 end-to-end确认，请展开 Amazon S3 源配置中的其他选项，然后选择启用 end-to-end消息确认。

源背压

当管道忙于处理数据，或者其接收器暂时关闭或数据采集速度缓慢时，管道可能会承受背压。OpenSearch 根据管道使用的源插件，Ingestion 有不同的处理背压的方法。

HTTP 源

使用 HTTP 源插件的管道处理反向压力的方式会有所不同，具体取决于哪个管道组件处于拥塞状态：

- 缓冲区 — 当缓冲区已满时，管道开始将错误代码为 408 的 HTTP 状态 `REQUEST_TIMEOUT` 返回到源端点。缓冲区被释放后，管道将重新开始处理 HTTP 事件。
- 源线程 — 当所有 HTTP 源线程都忙于执行请求，并且未处理的请求队列大小已超过允许的最大请求数时，管道开始将错误代码为 429 的 HTTP 状态 `TOO_MANY_REQUESTS` 返回到源端点。当请求队列降至允许的最大队列大小以下时，管道将重新开始处理请求。

OTel 来源

当使用 OpenTelemetry 源（OTel 日志、OTel 指标和OTel 跟踪）的管道的缓冲区已满时，管道开始向源端点返回错误代码为 408 的 HTTP 状态`REQUEST_TIMEOUT`。缓冲区被释放后，管道将重新开始处理事件。

S3 源

当带有 S3 源的管道的缓冲区已满时，管道将停止处理 SQS 通知。缓冲区被释放后，管道将重新开始处理通知。

如果接收器关闭或无法采集数据，并且为源启用了 end-to-end确认功能，则管道将停止处理 SQS 通知，直到收到来自所有接收器的成功确认为止。

创建 Amazon OpenSearch Ingestion 管道

管道是 Amazon OpenSearch Ingestion 用来将数据从其来源（数据来源）移动到接收器（数据所在的地方）的机制。在 OpenSearch Ingestion 中，接收器将始终是单个亚马逊 OpenSearch 服务域，而您的数据来源可以是 Amazon S3、Fluent Bit 或 Collector 等客户端。 OpenTelemetry

有关更多信息，请参阅 OpenSearch 文档中的管道。

主题

- 先决条件和所需 IAM 角色
- 所需的 IAM 权限
- 指定管道版本
- 指定提取路径

- 创建管道
- 跟踪管道创建的状态
- 使用蓝图

先决条件和所需 IAM 角色

要创建 OpenSearch 摄取管道，您必须拥有以下资源：

- 一个名为管道角色的 IAM 角色， OpenSearch Ingestion 为了写入接收器而担任该角色。您可以提前创建此角色，也可以让 OpenSearch Ingestion 在创建管道时自动创建该角色。
- 充当接收器的 OpenSearch 服务域或 OpenSearch 无服务器集合。如果您要写入某个域，则该域必须运行的是 OpenSearch 1.0 或更高版本，或者运行 Elasticsearch 7.4 或更高版本。接收器必须具有向您的 IAM 管道角色授予相应权限的访问策略。

有关创建这些资源的说明，请参阅以下主题：

- the section called “向管道授予访问域的权限”
- the section called “向管道授予访问集合的权限”

> **Note**
>
> 如果您要写入使用精细访问控制的域，则需要完成一些额外的步骤。请参阅the section called “映射管道角色（仅适用于使用精细访问控制的域）”。

所需的 IAM 权限

OpenSearch Ingestion 使用以下 IAM 权限来创建管道：

- `osis:CreatePipeline` –创建管道。
- `osis:ValidatePipeline` – 检查管道配置是否有效。
- `iam:CreateRole`并且 `iam:AttachPolicy` — 让 OpenSearch Ingestion 自动为您创建管道角色。
- `iam:PassRole`— 将管道角色传递给 OpenSearch Ingestion，这样它就可以向域写入数据。此权限必须位于管道角色资源上，*或者仅当您计划在每个管道中使用不同的角色时。

例如，以下策略授予创建管道的权限：

JSON

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

OpenSearch Ingestion 还包括一项名为的权限`osis:Ingest`，该权限是使用签名版本 4 向管道发送签名请求所必需的。有关更多信息，请参阅 the section called “创建摄取角色”。

> **Note**
>
> 此外，第一个在账户中创建管道的用户必须拥有 `iam:CreateServiceLinkedRole` 操作的权限。有关更多信息，请参阅管道角色资源。

有关每项权限的更多信息，请参阅《服务授权参考 OpenSearch 》中的 “摄取操作、资源和条件密钥”。

指定管道版本

使用配置编辑器创建管道时，必须指定管道将运行的 Data Prepper 的主要版本。要指定版本，请在您的管道配置中包含以下 `version` 选项：

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

选择 “创建” 时， OpenSearch Ingestion 会确定您指定的主要版本的最新可用次要版本，并使用该版本预置管道。例如，如果您指定`version: "2"`，并且最新支持的 Data Prepper 版本为 2.1.1，则 OpenSearch Ingestion 会使用版本 2.1.1 来配置您的管道。我们不会公开显示您的管道正在运行的次要版本。

要在 Data Prepper 新的主要版本可用时升级管道，请编辑管道配置并指定新版本。您无法将管道降级到较早版本。

> **Note**
>
> OpenSearch 新版本的 Data Prepper 发布后，Ingestion 不会立即支持这些版本。在新版本公开发行和 OpenSearch Ingestion 支持新版本之间会有一些延迟。此外， OpenSearch Ingestion 可能明确不完全支持某些主要版本或次要版本。有关完整列表，请参阅the section called “支持的 Data Prepper 版本”。

每当你对启动 blue/green 部署的管道进行更改时， OpenSearch Ingestion 都可以将其升级到当前为管道配置的主要版本的最新次要版本。有关更多信息，请参阅the section called “使用蓝绿部署进行管道更新”。 OpenSearch 除非您在工作流配置中明确更新该`version`选项，否则 Ingestion 无法更改管道的主要版本。

指定提取路径

对于基于拉取的来源，例如OTel 跟踪和OTel 指标， OpenSearch Ingestion 需要在源配置中添加其他`path`选项。路径是字符串（例如，`/log/ingest`），它表示提取的 URI 路径。此路径定义用于向管道发送数据的 URI。

例如，假设你为带有 HTTP 源的管道指定了以下路径：

向管道中摄取数据时，必须在客户端配置中指定以下端点：`https://pipeline-name-abc123.us-west-2.osis.amazonaws.com/my/test_path`。

路径必须以斜杠 (/) 开头，可以包含特殊字符 '-'、'_'、'.'、'/' 以及 `${pipelineName}` 占位符。如果您使用`${pipelineName}`（例如`/${pipelineName}/test_path`），OpenSearch Ingestion 会将变量替换为关联子管道的名称。

创建管道

本节介绍如何使用 OpenSearch 服务控制台和创建 OpenSearch 摄取管道。 Amazon CLI

控制台

要创建管道，请在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台，然后选择创建管道。

您可以选择空白管道，也可以选择配置蓝图。蓝图包括针对各种常见用例的预配置管道。有关更多信息，请参阅 the section called “使用蓝图”。

选择选择蓝图。

配置源

1. 如果您从空白管道开始，请从下拉菜单中选择一个来源。可用来源可能包括其他 Amazon Web Services 服务 OpenTelemetry、或 HTTP。有关更多信息，请参阅 the section called “集成管道”。
2. 根据您选择的来源，为源配置其他设置。例如，要使用 Amazon S3 作为来源，您必须指定管道接收消息的 Amazon SQS 队列的 URL。有关支持的源插件列表及其文档链接，请参阅the section called “支持的插件和选项”。
3. 对于某些源，必须指定源网络选项。选择 VPC 访问或公共访问。如果您选择 Public access (公有访问权限)，请跳至下一步。如果您选择 VPC 访问，请配置以下设置：

设置	描述
端点管理	选择是要自己创建虚拟私有云 (VPC) 终端节点，还是让 OpenSearch Ingestion 为您创建终端节点。端点管理默认为由 OpenSearch Ingestion 管理的端点。
VPC	选择您要使用的 VPC 的 ID。VPC 和管道必须位于同一 Amazon Web Services 区域中。
子网	选择一个或多个子网。 OpenSearch 服务将在子网中放置 VPC 终端节点和弹性网络接口。
安全组	选择一个或多个 VPC 安全组，允许所需的应用程序通过管道暴露的端口（80 或 443）和协议（HTTP 或 HTTPs）到达接 OpenSearch 入管道。
VPC 挂载选项	如果您的源是自主管理型端点，请将您的管道挂载到某个 VPC。选择提供的默认 CIDR 选项之一，或使用自定义 CIDR。

有关更多信息，请参阅 the section called “为管道配置 VPC 访问”。

4. 选择下一步。

配置处理器

向您的管道中添加一个或多个处理器。处理器是子管道中的组件，允许您在将记录发布到域或集合接收器之前筛选、转换和丰富事件。有关支持的处理器列表及其文档链接，请参阅the section called “支持的插件和选项”。

您可以选择 “操作” 并添加以下内容：

- 条件路由-根据特定条件将事件路由到不同的接收器。有关更多信息，请参阅条件路由。
- 子流水线 — 每个子流水线都是单个源、零个或多个处理器和一个接收器的组合。只有一个子管道可以有外部源。所有其他管道的源必须是整体管道配置中的其他子管道。单个管道配置可以包含 1-10 个子管道。

选择下一步。

配置接收器

选择管道发布记录的目的地。每个子管道必须包含至少一个接收器。您最多可以向管道添加 10 个水槽。

对于 OpenSearch 水槽，请配置以下字段：

设置	描述
网络策略名称 （仅限无服务器接收器）	如果您选择了 OpenSearch 无服务器集合，请输入网络策略名称。OpenSearch Ingestion 要么创建策略（如果不存在），要么使用允许访问连接管道和集合的 VPC 终端节点的规则对其进行更新。有关更多信息，请参阅 the section called “向管道授予访问集合的权限”。
索引名	管道发送数据的索引的名称。 OpenSearch 如果此索引尚不存在，则 Ingestion 会创建该索引。
索引映射选项	选择管道如何将文档及其字段存储和索引到 OpenSearch 接收器中。如果选择“动态映射”，则在为文档编制索引时会自动 OpenSearch 添加字段。如果选择“自定义映射”，请输入索引映射模板。有关更多信息，请参阅索引模板。
启用 DLQ	为管道配置 Amazon S3 死信队列 (DLQ)。有关更多信息，请参阅 the section called “死信队列”。
其他设置	为 OpenSearch 水槽配置高级选项。有关更多信息，请参阅 Data Prepper 文档中的配置选项。

要添加 Amazon S3 接收器，请选择添加接收器和 Amazon S3。有关更多信息，请参阅 the section called “Amazon S3 作为目标”。

选择下一步。

配置管道

配置以下其他管道设置：

设置	描述
管道名称	管道的唯一名称。

设置	描述
永久缓冲区	永久缓冲区将您的数据存储在跨多个可用区的基于磁盘的缓冲区中。有关更多信息，请参阅 the section called “持久缓冲功能”。 如果启用永久缓冲，请选择加密缓冲区数据的 Amazon Key Management Service 密钥。
管道容量	最小和最大管道容量，以摄取 OpenSearch 计算单位 () OCUs 为单位。有关更多信息，请参阅 the section called “扩缩管道”。
管道角色	IAM 角色，为管道提供写入接收器和从基于拉取的源中读取所需的权限。你可以自己创建角色，也可以让 OpenSearch Ingestion 根据你选择的用例为你创建角色。 有关更多信息，请参阅 the section called “设置角色和用户”。
标签	向您的管道添加一个或多个标签。有关更多信息，请参阅 the section called “标记管道”。
日志发布选项	启用将管道日志发布到 Amazon CloudWatch 日志。建议您启用日志发布，以便更轻松地解决管道问题。有关更多信息，请参阅 the section called “监控管道日志”。

选择下一步。，然后查看您的工作流配置并选择创建管道。

OpenSearch Ingestion 运行异步进程来构建管道。当管道状态为 `Active` 时，您可以开始将数据提取到管道。

Amazon CLI

create-pipel ine 命令接受以字符串形式或.yaml 或.json 文件中的管道配置。如果您以字符串形式提供配置，则必须使用 `\n` 转义每一个新行。例如，`"log-pipeline:\n source:\n http:\n processor:\n - grok:\n ...`

以下示例命令采用以下配置创建管道：

- 最少 4 次摄入 OCUs，最多 10 次摄入 OCUs
- 在虚拟私有云 (VPC) 中预调配

- 启用日志发布

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

OpenSearch Ingestion 运行异步进程来构建管道。当管道状态为 `Active` 时，您可以开始将数据提取到管道。要检查管道的状态，请使用GetPipeline命令。

OpenSearch 摄取 API

要使用 OpenSearch Ingestion API 创建 OpenSearch 摄取管道，请调用该操作。CreatePipeline

成功创建管道后，您可以配置客户端并开始将数据提取到您的 OpenSearch 服务域中。有关更多信息，请参阅 the section called “集成管道”。

跟踪管道创建的状态

在 OpenSearch Ingestion 配置管道并准备好接收数据时，您可以跟踪管道的状态。

控制台

最初创建管道后，当 OpenSearch Ingestion 为采集数据做准备时，它会经历多个阶段。要查看管道创建的各个阶段，请选择管道名称以查看其管道设置页面。在状态下，选择查看详细信息。

管道要经过以下几个阶段才可以摄取数据：

- 验证 — 验证管道配置。此阶段完成后，所有验证均已成功。
- 创建环境 — 准备和预调配资源。此阶段完成后，即创建了新的管道环境。
- 部署管道 — 部署管道。此阶段完成后，管道已成功部署。
- 检查管道运行状况 — 检查管道的运行状况。此阶段完成后，所有运行状况检查均已通过。
- 启用流量 — 允许管道摄取数据。此阶段完成后，您可以开始将数据提取到管道。

CLI

使用get-pipeline-change-progress命令检查管道的状态。以下 Amazon CLI 请求检查名为的管道的状态`my-pipeline`：

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

响应：

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

OpenSearch 摄取 API

要使用 OpenSearch Ingestion API 跟踪管道创建的状态，请调用该操作。GetPipelineChangeProgress

使用蓝图

您可以使用配置蓝图，而不是从头开始创建管道定义，这些蓝图是针对常见摄取方案（例如 Trace Analytics 或 Apache 日志）的预配置模板。配置蓝图可帮助您轻松预调配管道，而不必从头开始编写配置。

控制台

要使用管道蓝图

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。

2. 在左侧导航窗格中选择管道，然后选择创建管道。
3. 从用例列表中选择蓝图，然后选择选择蓝图。管道配置会填充您所选用例的子管道。

   初始的管道蓝图是无效的。您需要根据所选来源指定其他设置。

CLI

要使用获取所有可用蓝图的列表 Amazon CLI，请发送list-pipeline-blueprints请求。

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

该请求返回所有可用蓝图的列表。

要获取有关特定蓝图的更多详细信息，请使用以下get-pipeline-blueprint命令：

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

该请求返回 Apache 日志管道蓝图的内容：

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

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

OpenSearch 摄取 API

要使用 OpenSearch Ingestion API 获取有关管道蓝图的信息，请使用和操作。ListPipelineBlueprintsGetPipelineBlueprint

查看 Amazon OpenSearch Ingestion 管道

您可以使用 Amazon Web Services Management Console、或 Ingestion API，查看有关 Amazon OpenSearch Ingestion 管道的 OpenSearch 详细信息。 Amazon CLI

控制台

查看管道

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择管道。
3. （可选）要查看具有特定状态的管道，请选择任何状态并选择要筛选的状态类型。

   管道可以具有以下状态：

   - `Active`— 管道处于活动状态，并且可以摄取数据。
   - `Creating`— 正在创建管道。
   - `Updating`— 正在更新管道。
   - `Deleting`— 正在删除管道。
   - `Create failed`— 无法创建管道。
   - `Update failed`— 无法更新管道。
   - `Stop failed`— 无法停止管道。
   - `Start failed`— 管道无法启动。
   - `Stopping`— 正在停止管道。
   - `Stopped`— 管道已停止，可以随时重新启动。
   - `Starting`— 管道正在启动。

当管道处于Create failed、Creating、和状态 OCUs 时，您无需支付摄取费用。Deleting Stopped

CLI

要使用查看管道 Amazon CLI，请发送 list-pipelines 请求：

```
aws osis list-pipelines
```

该请求返回所有现有管道的列表：

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

要获取有关单个管道的信息，请使用 get-pipeline 命令：

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

该请求返回指定管道的配置信息：

```
{
    "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

要使用 OpenSearch Ingestion API 查看 OpenSearch 摄取管道，请调用和操作。ListPipelinesGetPipeline

更新 Amazon OpenSearch Ingestion 管道

您可以使用 Amazon Web Services Management Console、或 OpenSearch Ingestion API 更新 Amazon OpenSearch Ingestion 管道。 Amazon CLI OpenSearch 当你更新管道 blue/green 配置时，Ingestion 会启动部署。有关更多信息，请参阅 the section called “使用蓝绿部署进行管道更新”。

主题

- 注意事项

- 所需权限
- 更新管道
- 使用蓝绿部署进行管道更新

注意事项

更新管道时，请注意以下事项：

- 您无法更新管道的名称或网络设置。
- 如果管道写入 VPC 域接收器，则在创建管道后将无法返回，也无法将接收器更改为其他 VPC 域。您必须删除，然后使用新的接收器重新创建管道。您仍然可以将接收器从 VPC 域切换到公有域、从公有域切换到 VPC 域，或者从一个公有域切换到另一个公有域。
- 您可以随时在公共 OpenSearch 服务域和 OpenSearch 无服务器集合之间切换管道接收器。
- 当您更新管道的源、处理器或接收器配置时， OpenSearch Ingestion 会启动部署。 blue/green 有关更多信息，请参阅 the section called “使用蓝绿部署进行管道更新”。
- 当您更新管道的源、处理器或接收器配置时， OpenSearch Ingestion 会自动将您的管道升级到该管道正在运行的 Data Prepper 主版本支持的最新次要版本。此过程可让您的管道及时获取最新错误修复和性能改进。
- 管道停止后，仍然可以对管道进行更新。

所需权限

OpenSearch Ingestion 使用以下 IAM 权限来更新管道：

- `osis:UpdatePipeline` – 更新管道。
- `osis:ValidatePipeline` – 检查管道配置是否有效。
- `iam:PassRole`— 将管道角色传递给 OpenSearch Ingestion，这样它就可以向域写入数据。只有在更新工作流配置时才需要此权限，而在修改其他设置（例如日志发布或容量限制）时才需要此权限。

例如，以下策略授予更新管道的权限：

JSON

```
{
```

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

更新管道

您可以使用 Amazon Web Services Management Console、或 OpenSearch Ingestion API 更新 Amazon OpenSearch Ingestion 管道。 Amazon CLI

控制台

更新管道

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择管道。
3. 选择一个管道以打开其设置。然后，选择其中一个 “编辑” 选项。
4. 完成更改后，选择 Save (保存)。

CLI

要使用更新管道 Amazon CLI，请发送更新管道请求。以下示例请求上传新配置文件并更新最小和最大容量值：

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

OpenSearch 摄取 API

要使用 OpenSearch Ingestion API 更新 OpenSearch 摄取管道，请调用该操作。UpdatePipeline

使用蓝绿部署进行管道更新

OpenSearch 更新管道配置时，Ingestion 会启动蓝/绿部署流程。

Blue/green refers to the practice of creating a new environment for pipeline updates and routing traffic to the new environment after those updates are complete. The practice minimizes downtime and maintains the original environment in the event that deployment to the new environment is unsuccessful. Blue/green部署本身不会对性能产生任何影响，但是如果您的工作流配置以改变性能的方式发生变化，则性能可能会发生变化。

OpenSearch 在部署期间，Ingestion 会阻止自动缩放。 blue/green 在将旧管道重定向到新管道之前，只需继续为其支付流量费。重定向流量后，只需支付新管道费用。永远不需要同时为两条管道付费。

更新管道的源、处理器或接收器配置时， OpenSearch Ingestion 可以自动将您的管道升级到该管道正在运行的主要版本的最新支持次要版本。例如，您的工作流配置`version: "2"`中可能有， OpenSearch Ingestion 最初使用版本 2.1.0 配置了管道。添加对版本 2.1.1 的支持并且您更改工作流配置后， OpenSearch Ingestion 会将您的管道升级到 2.1.1 版。

此过程可让您的管道及时了解最新的错误修复和性能改进。 OpenSearch 除非您在工作流配置中手动更改该`version`选项，否则 Ingestion 无法更新管道的主要版本。

管理 Amazon OpenSearch Ingestion 管道成本

您可以在 Amazon Ingestion 中启动和停止 OpenSearch 摄取管道，以根据需要控制数据流。停止管道会在保留配置的同时停止数据处理，因此无需重新配置即可重新启动管道。这可以帮助优化成本、管理资源使用情况或解决问题。当你停止管道时， OpenSearch Ingestion 不会处理传入的数据，但之前摄取的数据在中仍然可用。 OpenSearch

启动和停止简化了用于开发、测试或不需要持续可用性的类似活动的管道的设置和停用过程。管道停止时，不计入任何 Ingestion OCU 小时数。您仍然可以更新已停止的管道，它们会自动接收次要版本更新和安全性补丁。重新启动管道会恢复对新传入数据的处理。

> **Note**
>
> 如果您的管道容量过剩但需要保持运行，请考虑调整其最大容量限制，而不是停止并重启管道。这可以帮助管理成本，同时确保管道继续高效地处理数据。有关更多详细信息，请参阅 the section called “扩缩管道”。

以下主题说明了如何使用 Amazon Web Services Management Console、 Amazon CLI和 OpenSearch Ingestion API 启动和停止管道。

主题

- 停止 Amazon OpenSearch Ingestion 渠道
- 启动 Amazon OpenSearch Ingestion 渠道

停止 Amazon OpenSearch Ingestion 渠道

要使用 In OpenSearch gestion 管道或执行管理，请始终从活动管道开始，接着停止管道，然后重新启动管道。管道停止时，不计入 Ingestion OCU 小时数。

控制台

停止管道

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在导航窗格中，选择管道，然后选择管道。您可以从该页面中执行停止操作，或者导航到要停止的管道的详细信息页面。
3. 在操作中，选择停止管道。

   如果无法停止和启动管道，则停止管道操作不可用。

Amazon CLI

要使用停止管道 Amazon CLI，请使用以下参数调用停止-管道命令：

- `--pipeline-name` – 管道的名称。

Example

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

OpenSearch Ingestion API

要使用 OpenSearch Ingestion API 停止管道，请使用以下参数调用StopPipeline操作：

- `PipelineName` – 管道的名称。

启动 Amazon OpenSearch Ingestion 渠道

启动 OpenSearch Ingestion 管道时，始终从已处于停止状态的管道开始。管道保留其配置设置，例如容量限制、网络设置和日志发布选项。

重新启动管道通常需要几分钟时间。

控制台

启动管道

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在导航窗格中，选择管道，然后选择管道。您可以从该页面中执行启动操作，或者导航到要启动的管道的详细信息页面。
3. 对于操作，选择启动管道。

Amazon CLI

要使用启动管道 Amazon CLI，请使用以下参数调用启动-管道命令：

- `--pipeline-name` – 管道的名称。

Example

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

OpenSearch Ingestion API

要使用 OpenSearch Ingestion API 启动 OpenSearch Ingestion 管道，请使用以下参数调用StartPipeline操作：

- `PipelineName` – 管道的名称。

删除 Amazon OpenSearch Ingestion 管道

您可以使用 Amazon Web Services Management Console、或 OpenSearch Ingestion API 删除 Amazon OpenSearch Ingestion 管道。 Amazon CLI当管道的状态为 `Creating` 或 `Updating` 时，无法删除管道。

控制台

删除管道

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择管道。
3. 选择要删除的管道，然后选择操作、删除。
4. 确认删除并选择删除。

CLI

要使用删除管道 Amazon CLI，请发送删除管道请求：

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

OpenSearch Ingestion API

要使用 OpenSearch Ingestion API 删除 OpenSearch Ingestion 管道，请使用以下参数调用DeletePipeline操作：

- `PipelineName` – 管道的名称。

Amazon OpenSearch Ingestion 管道支持的插件和选项

Amazon OpenSearch Ingestion 支持开源 Data Prepper 中的源、处理器和接收器仅相当于开源 OpenSearch Dat a Prepper 的一小部分。此外， OpenSearch Ingestion 对支持的每个插件的可用选项提出了一些限制。以下各节介绍 OpenSearch Ingestion 支持的插件和相关选项。

> **Note**
>
> OpenSearch Ingestion 不支持任何缓冲区插件，因为它会自动配置默认缓冲区。如果您在管道配置中添加缓冲区，将收到验证错误。

主题

- 支持的插件
- 无状态与有状态处理器
- 配置要求和限制

支持的插件

OpenSearch Ingestion 支持以下 Data Prepper 插件：

源：

- DocumentDB
- DynamoDB

- HTTP
- Kafka
- Kinesis
- OpenSearch
- OTel 日志
- OTel 指标
- OTel 跟踪
- S3

处理器：

- 添加条目
- 聚合
- 异常探测器
- Amazon Lambda
- 转换条目类型
- 复制值
- CSV
- 日期
- 延迟
- Decompress
- 删除条目
- Dissect
- 删除事件
- 展平
- Geo IP
- Grok
- 键值
- 列表到地图
- 小写字符串
- Map to list
- Mutate event（处理器系列）
- Mutate string（处理器系列）
- 混淆处理
- OTel 指标
- OTel 追踪组
- OTel 跟踪
- Parse Ion
- 解析 JSON
- Parse XML

- 重命名密钥
- 路线
- Select entries
- 服务地图
- 拆分赛事
- 拆分字符串
- 字符串转换器
- 替换字符串
- 跟踪对等转发服务器
- Translate
- 修剪字符串
- Truncate
- 大写字符串
- 用户代理
- 写 JSON

接收器：

- OpenSearch（支持 OpenSearch 服务、 OpenSearch 无服务器和 Elasticsearch 6.8 或更高版本）
- S3

接收器编解码器：

- Avro
- NDJSON
- JSON
- Parquet

无状态与有状态处理器

无状态处理器执行转换和筛选等操作，而有状态处理器则执行聚合等操作，会记住上一次运行的结果。OpenSearch Ingestion 支持有状态的处理器聚合和服务映射。所有其他受支持的处理器均为无状态处理器。

对于仅包含无状态处理器的管道，最大容量限制为 96 个 Ingest OCUs ion。对于包含任何有状态处理器的管道，最大容量限制为 48 个摄 OCUs取。但是，如果管道启用了永久缓冲区，则仅 OCUs 包含无状态处理器时最多可以有 192 个摄取，包含任何有状态处理器时最多可以有 192 个摄取 OCUs 。有关更多信息，请参阅 the section called “扩缩管道”。

End-to-end 仅无状态处理器支持确认。有关更多信息，请参阅 the section called “End-to-end 确认”。

配置要求和限制

除非下文另有说明，否则上文列出的受支持插件的 Data Prepper 配置参考中描述的所有选项均允许在 OpenSearch Ingestion 管道中使用。以下各节介绍了 OpenSearch Ingestion 对某些插件选项提出的限制。

> **Note**
>
> OpenSearch Ingestion 不支持任何缓冲区插件，因为它会自动配置默认缓冲区。如果您在管道配置中添加缓冲区，将收到验证错误。

许多选项都由 OpenSearch Ingestion 在内部配置和管理，例如`authentication`和。`acm_certificate_arn`其他选项（例如，`thread_count` 和 `request_timeout`），一旦手动更改，则会影响性能。因此，将在内部设置这些值，以确保实现管道的最佳性能。

最后，某些选项无法传递到 OpenSearch Ingestion（例如`ism_policy_file`和）`sink_template`，因为在开源 Data Prepper 中运行时它们是本地文件。这些值不受支持。

主题

- 常规管道选项
- Grok 处理器
- HTTP 源
- OpenSearch 下沉
- OTel 指标源、 OTel 跟踪源和 OTel 日志源
- OTel 跟踪组处理器
- OTel 跟踪处理器
- 服务映射处理器

- S3 源

常规管道选项

以下常规管道选项由 OpenSearch Ingestion 设置，在管道配置中不受支持：

- `workers`
- `delay`

Grok 处理器

以下 Grok 处理器选项不受支持：

- `patterns_directories`
- `patterns_files_glob`

HTTP 源

HTTP 源插件具有以下要求和限制：

- `path` 选项为必填项。路径是字符串（例如，`/log/ingest`），它表示日志摄取的 URI 路径。此路径定义用于向管道发送数据的 URI。例如 `https://log-pipeline.us-west-2.osis.amazonaws.com/log/ingest`。路径必须以斜杠 (/) 开头，而且可以包含特殊字符 '-'、'_'、'.'、'/' 以及 `${pipelineName}` 占位符。
- 以下 HTTP 源选项由 OpenSearch Ingestion 设置，在管道配置中不受支持：
  - `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 下沉

OpenSearch接收器插件具有以下要求和限制。

- aws 选项为必填项，必须包含以下选项：
  - sts_role_arn
  - region
  - hosts
  - serverless（如果接收器是 OpenSearch 无服务器集合）
- sts_role_arn 选项必须指向 YAML 定义文件中每个接收器的同一角色。
- 该hosts选项必须指定 OpenSearch 服务域端点或 OpenSearch 无服务器集合端点。您不能为域指定自定义端点；必须是标准端点。
- 如果 hosts 选项为无服务器集合端点，则必须将 serverless 选项设置为 true。此外，如果 YAML 定义文件包含 index_type 选项，则必须将其设置为 management_disabled，否则验证将失败。
- 不支持以下选项：
  - username
  - password
  - cert
  - proxy
  - dlq_file - 如果要将失败事件卸载到死信队列 (DLQ)，则必须使用 dlq 选项并指定 S3 存储桶。
  - ism_policy_file
  - socket_timeout
  - template_file
  - insecure

OTel 指标源、 OTel 跟踪源和 OTel 日志源

OTel 指标源、OTel 跟踪源和OTel 日志源插件具有以下要求和限制：

- path 选项为必填项。路径是字符串（例如，/log/ingest），它表示日志摄取的 URI 路径。此路径定义用于向管道发送数据的 URI。例如 https://log-pipeline.us-west-2.osis.amazonaws.com/*log/ingest*。路径必须以斜杠 (/) 开头，而且可以包含特殊字符 '-'、'_'、'.'、'/' 以及 ${pipelineName} 占位符。
- 以下选项由 OpenSearch Ingestion 设置，在管道配置中不受支持：
  - 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 跟踪组处理器

OTel 跟踪组处理器具有以下要求和限制：

- aws 选项为必填项，必须包含以下选项：
  - sts_role_arn

  - `region`
  - `hosts`
- 该`sts_role_arn`选项指定的角色与您在 OpenSearch 接收器配置中指定的管道角色相同。
- 不支持 `username`、`password`、`cert` 和 `insecure` 选项。
- `aws_sigv4` 选项为必填项，必须设置为 true。
- 不支持 sin OpenSearch k 插件`serverless`中的选项。Otel 跟踪组处理器目前不适用于 OpenSearch 无服务器集合。
- 管道配置主体中的 `otel_trace_group` 处理器数量不能超过 8 个。

OTel 跟踪处理器

OTel 跟踪处理器具有以下要求和限制：

- `trace_flush_interval` 选项的值不能超过 300 秒。

服务映射处理器

服务映射处理器具有以下要求和限制：

- `window_duration` 选项的值不能超过 300 秒。

S3 源

S3 源插件具有以下要求和限制：

- `aws` 选项为必填项，必须包含 `region` 和 `sts_role_arn` 选项。
- `records_to_accumulate` 选项的值不能超过 200。
- `maximum_messages` 选项的值不能超过 10。
- 如果指定，则 `disable_bucket_ownership_validation` 选项必须设置为 false。
- 如果指定，则 `input_serialization` 选项必须设置为 `parquet`。

将 Amazon OpenSearch Ingestion 管道与其他服务和应用程序集成

要成功将数据提取到 Amazon OpenSearch Ingestion 管道，您必须将您的客户端应用程序（源）配置为将数据发送到管道终端节点。您的来源可能是客户端，例如 Fluent Bit 日志、 OpenTelemetry 收集器或简单的 S3 存储桶。每个客户端的确切配置各不相同。

源配置期间（与直接向 OpenSearch 服务域或 OpenSearch 无服务器集合发送数据相比）的重要区别在于 Amazon 服务名称 (`osis`) 和主机端点，它们必须是管道端点。

构建摄取端点

要将数据摄入管道，请将其发送到摄取端点。要找到提取 URL，请导航至管道设置页面并复制摄取 URL。

要为基于拉取的来源（例如OTel 跟踪和OTel 指标）构建完整的摄取终端节点，请将管道配置中的提取路径添加到摄取网址。

例如，假设管道配置的摄取路径如下所示：

您在客户端配置中指定的完整摄取端点将采用以下格式：`https://`*`ingestion-pipeline-abcdefg`*`.us-east-1.osis.amazonaws.com`*`/my/test_path`*。

创建摄取角色

所有对 OpenSearch Ingestion 的请求都必须使用签名版本 4 进行签名。至少，必须向签署请求的角色授予`osis:Ingest`操作权限，从而允许其向 OpenSearch 摄取管道发送数据。

例如，以下 Amazon Identity and Access Management (IAM) 策略允许相应的角色向单个管道发送数据：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:region:account-id:pipeline/pipeline-name"
    }
  ]
}
```

> **Note**
>
> 要将角色用于所有管道，请将 `Resource` 元素的 ARN 替换为通配符 (*)。

提供跨账户摄取访问权限

> **Note**
>
> 只能为公有管道（而非 VPC 管道）提供跨账户摄取访问权限。

您可能需要将来自其他渠道的数据提取到管道中 Amazon Web Services 账户，例如存放源应用程序的帐户。如果写入管道的主体与管道本身的账户不同，则需要将主体配置为信任另一个 IAM 角色，以将数据摄取到管道中。

配置跨账户摄取权限

1. 在与管道相同的范围内创建具有`osis:Ingest`权限的摄取角色（ Amazon Web Services 账户 如上一节所述）。有关说明，请参阅创建 IAM 角色。
2. 为摄取角色附加信任策略，允许其他账户主体担任此角色：

   JSON

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

3. 在另一个账户中，配置您的客户端应用程序（例如，Fluent Bit）担任摄取角色。为使配置生效，应用程序账户必须向应用程序用户或角色授予担任摄取角色的权限。

   以下基于身份的示例策略允许附加主体担任管道账户的 `ingestion-role`：

   JSON

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

然后，客户端应用程序可以使用该AssumeRole操作来假设数据`ingestion-role`并将其摄取到关联的管道中。

在 Atlas OpenSearch sian 服务中使用采集管道

你可以使用 Atlassian Jira 和 Confluence 源插件将来自 Atlassian 服务的数据提取到你的摄取管道中。OpenSearch 这些集成使您能够通过同步完整的 Jira 项目和 Confluence 空间来创建统一的可搜索知识库，同时通过持续监控和自动同步更新来保持实时相关性。

Integrating with Jira

将您的 Jira 内容集成到，借助强大的上下文搜索功能，改变您的 Jira 体验。 OpenSearchData Prepper Atlassian Jira 源插件使您能够通过同步完整的 Jira 项目来创建统一的可搜索知识库，同时通过持续监控和自动同步更新来保持实时相关性。这种集成允许通过灵活的筛选选项对特定项目、问题类型和状态进行数据同步，确保只导入您需要的信息。

为了确保安全可靠的连接，该插件支持多种身份验证方法，包括基本的 API 密钥身份验证和 OAuth2身份验证，并通过使用存储在中的密钥管理凭据增加了安全性 Amazon Secrets Manager。它还具有自动令牌续订功能，可实现不间断的访问，从而确保持续运行。此集成基于 Atlassian 的 API 版本 2，使团队能够通过高级搜索功能从 Jira 数据中解锁宝贵 OpenSearch的见解。

Integrating with Confluence

OpenSearch 通过 Data Prepper 的 Confluence 源插件将 Atlassian Confluence 内容集成到 Confluenc e 源插件中，增强团队的知识管理和协作能力。这种集成使您能够创建一个集中、可搜索的集体知识存储库，从而提高信息发现和团队工作效率。通过同步 Confluence 内容并持续监控更新，该插件可确保您的 OpenSearch 索引保持 up-to-date完整且完整。

该集成提供了灵活的筛选选项，允许您有选择地从特定空间或页面类型导入内容，根据组织的需求定制同步内容。该插件支持基本的 API 密钥和 OAuth2 身份验证方法，可以选择通过安全地管理凭据 Amazon Secrets Manager。该插件的自动令牌续订功能可确保不间断的访问和无缝操作。此集成基于 Atlassian 的 Confluence API，使团队能够在其 Confluence 内容中利用 OpenSearch高级搜索功能，从而提高组织内部的信息可访问性和利用率。

主题

- 先决条件
- 配置管道角色
- Jira 连接器管道配置
- Confluence 连接器管道配置
- 数据一致性

- 限制
- Atlass CloudWatch ian 连接器的指标
- 使用 2.0 将 Amazon OpenSearch Ingestion 管道连接到 Atlassian Jira 或 Confluence OAuth

先决条件

在创建 OpenSearch Ingestion 管道之前，请完成以下步骤：

1. 选择以下选项之一，为您的 Jira 网站准备证书。 OpenSearch 摄取只需要对内容进行`ReadOnly`授权。

   a. 选项 1：API 密钥 — 登录您的 Atlassian 账户并使用以下主题中的信息生成您的 API 密钥：

      - 管理您的 Atlassian 账户的 API 代币

   b. 选项 2： OAuth2— 登录您的 Atlassian 帐户并使用中的信息。the section called “使用 2.0 将 Amazon OpenSearch Ingestion 管道连接到 Atlassian Jira 或 Confluence OAuth ”

2. 在中创建密钥 Amazon Secrets Manager以存储在上一步中创建的证书。按照步骤进行操作时，请做出以下选择：

   - 对于密钥类型，请选择其他密钥类型。
   - 对于密钥/值对，根据您选择的授权类型创建以下对：

   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
   }
   ```

   创建密钥后，复制密钥的 Amazon 资源名称 (ARN)。您将把它包含在管道角色权限策略中。

配置管道角色

在管道中传递的角色必须附加以下策略才能读取和写入在先决条件部分中创建的密钥。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecretReadWrite",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:PutSecretValue",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "secret-arn-created-in-step-2"
        }
    ]
}
```

该角色还应附有访问和写入您选择的接收器的策略。例如，如果您选择 OpenSearch 作为接收器，则策略将类似于以下内容：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "OpenSearchWritePolicy",
            "Effect": "Allow",
            "Action": "aoss:*",
            "Resource": "aoss-collection-arn"
        }
    ]
}
```

Jira 连接器管道配置

您可以使用预配置的 Atlassian Jira 蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

将 *placeholder values* 替换为您自己的信息。

```
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 Amazon 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 源代码中属性的关键：

1. 主机：您的 Jira 云或本地 URL。一般来说，它看起来像`https://your-domain-name.atlassian.net/`。
2. 致谢：保证数据一直传送到接收器。
3. 身份验证：描述您希望管道如何访问您的 Jira 实例。选择`Basic`或`OAuth2`并指定引用密钥中密钥的相应密钥属性。 Amazon
4. 过滤器：此部分可帮助您选择要提取和同步的 Jira 数据的哪一部分。
   a. p roject：在该`include`部分中列出要同步的项目密钥。否则，请在该`exclude`部分下方列出要排除的项目。在任何给定时间仅提供一个包含或排除选项。
   b. issue_type：你要同步的特定问题类型。遵循适合您需求的类似`include`或`exclude`模式。请注意，附件将显示为指向原始附件的锚链接，但不会提取附件内容。
   c. 状态：您要为数据提取查询应用的特定状态筛选器。如果您指定`include`，则只有具有这些状态的票证才会被同步。如果您指定`exclude`，则将同步除列出的已排除状态的门票以外的所有门票。

Confluence 连接器管道配置

您可以使用预配置的 Atlassian Confluence 蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

```
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 Amazon 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 源中的关键属性：

1. 主机：您的 Confluence 云或本地网址。一般来说，它看起来像 `https://your-domain-name.atlassian.net/`
2. 致谢：保证数据一直传送到接收器。
3. 身份验证：描述您希望管道如何访问您的 Confluence 实例。选择`Basic`或`OAuth2`并指定引用密钥中密钥的相应密钥属性。 Amazon
4. 过滤器：此部分可帮助您选择要提取和同步的 Confluence 数据的哪一部分。
   a. 空间：在该`include`部分中列出要同步的空格键。否则，请在该`exclude`部分下方列出要排除的空格。在任何给定时间仅提供一个包含或排除选项。
   b. page_typ e：您要同步的特定页面类型（如页面、博客文章或附件）。遵循适合您需求的类似`include`或`exclude`模式。请注意，附件将显示为指向原始附件的锚链接，但不会提取附件内容。

数据一致性

根据管道 YAML 中指定的过滤器，选定的项目（或空间）将被提取一次，并完全同步到目标接收器。然后，持续的变更监控将在变更发生时捕捉变化并更新接收器中的数据。一个例外情况是，变更监控只会同步`update`操作，`create`而不会同步`delete`操作。

限制

- 用户删除操作不会同步。数据一旦记录在接收器中，将保留在接收器中。如果在接收器设置中指定了 ID 映射，则更新将使用新的更改覆盖现有内容。

- 使用不支持以下内容的旧版 Atlassian 软件的本地实例与该来源 APIs 不兼容：
  - Jira Search API 版本 3
    - `rest/api/3/search`
    - `rest/api/3/issue`
  - Confluence
    - `wiki/rest/api/content/search`
    - `wiki/rest/api/content`
    - `wiki/rest/api/settings/systemInfo`

Atlass CloudWatch ian 连接器的指标

类型：Jira 连接器指标

来源	指标	指标类型
acknowledgementSetSuccesses。count	计数器	如果启用了确认功能，则此指标将提供成功同步的票证数量。
acknowledgementSetFailures。count	计数器	如果启用了确认功能，则此指标将提供同步失败的票证数量。
crawlingTime.avg	计时器	浏览所有新更改所花费的时间。
ticketFetchLatency.avg	计时器	取票 API 延迟的平均值。
ticketFetchLatency.max	计时器	取票 API 延迟的最大值。

来源	指标	指标类型
已申请门票。COUNT	计数器	发出的取票请求数。
ticketRequestedFailed。count	计数器	失败的取票请求数。
ticketRequestedSuccess。count	计数器	成功获取票证的请求数。
searchCallLatency.avg	计时器	搜索 API 调用延迟平均值。
searchCallLatency.max	计时器	搜索 API 调用延迟时间上限。
searchResultsFound。count	计数器	在给定搜索调用中找到的项目数。
searchRequestFailed。count	计数器	搜索 API 调用失败次数。
authfailures.count	计数器	身份验证失败计数。

类型：Confluence 连接器指标

来源	指标	指标类型
acknowledgementSetSuccesses。count	计数器	如果启用了确认功能，则此指标将提供成功同步的页数。
acknowledgementSetFailures。count	计数器	如果启用了确认功能，则此指标将提供同步失败的页数。
crawlingTime.avg	计时器	浏览所有新更改所花费的时间。
pageFetchLatency.avg	计时器	内容提取 API 延迟（平均值）。
pageFetchLatency.max	计时器	内容提取 API 延迟（最大）。
请求的页面数。计数	计数器	内容获取 API 的调用次数。
pageRequestFailed。count	计数器	内容提取 API 的失败请求数。
pageRequestedSuccess。count	计数器	成功请求内容获取 API 的次数。
searchCallLatency.avg	计时器	搜索 API 调用延迟平均值。

来源	指标	指标类型
searchCallLatency.max	计时器	搜索 API 调用延迟上限
searchResultsFound。count	计数器	在给定搜索调用中找到的项目数。
searchRequestsFailed。count	计数器	搜索 API 调用失败次数。
authfailures.count	计数器	身份验证失败计数。

使用 2.0 将 Amazon OpenSearch Ingestion 管道连接到 Atlassian Jira 或 Confluence OAuth

使用本主题中的信息来帮助您使用 2.0 身份验证配置 Amazon OpenSearch Ingestion 管道并将其连接到 Jira 或 Confluence 账户。 OAuth 在完成使用 Atlassian S the section called “先决条件” ervices 的 OpenSearch 摄取管道但选择不使用 API 密钥凭证时执行此任务。

主题

- 创建 OAuth 2.0 集成应用程序
- 生成并刷新 Atlassian 开发者访问令牌

创建 OAuth 2.0 集成应用程序

使用以下步骤帮助您在 Atlassian 开发者网站上创建 OAuth 2.0 集成应用程序。

创建 OAuth 2.0 集成应用程序

1. 通过 myapps/ 登录到你的 Atlassian 开发者账户。 https://developer.atlassian.com/console/
2. 选择创建，OAuth 2.0 集成。
3. 在名称中，输入一个名称以标识应用程序的用途。

4. 选中 “我同意接受 Atlassian 开发者条款的约束” 复选框，然后选择 “创建”。
5. 在左侧导航栏中，选择 “授权”，然后选择 “添加”。
6. 在 “回调 URL” 中，输入任意 URL，例如**`https://www.amazon.com`**或**`https://www.example.com`**，然后选择 “保存更改”。
7. 在左侧导航栏中，选择 “权限” 页面，然后在 Jira API 行中选择 “添加”，然后选择 “配置”。然后选择所有 “经典范围” 读取权限（列表如下），然后选择 “保存”
8. 选择 “精细范围” 选项卡，然后选择 “编辑范围” 以打开 “编辑 Jira API” 对话框。
9. 选择您正在使用的源插件的权限：

   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. 选择保存。

有关相关信息，请参阅 At lassian 开发者网站上的实施 OAuth 2.0 (3LO) 和确定操作所需的范围。

生成并刷新 Atlassian 开发者访问令牌

使用以下步骤帮助您在 Atlassian 开发者网站上生成和刷新 Atlassian 开发者访问令牌。

生成和刷新 Jira 访问令牌

1. 通过 myapps/ 登录到你的 Atlassian 开发者账户。 https://developer.atlassian.com/console/
2. 选择您在中创建的应用程序the section called “创建 OAuth 2.0 集成应用程序”。
3. 在左侧导航栏中，选择 “授权”。
4. 从页面底部复制精细的 Atlassian API 授权网址值，然后将其粘贴到您选择的文本编辑器中。

   URL 的格式如下所示：

```
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. 对于`state=YOUR_USER_BOUND_VALUE`，将参数值更改为您选择的任何值，例如 state="**`sample_text`**"。

   有关更多信息，请参阅状态参数的用途？ 在 Atlassian 开发者网站上。

6. 请注意，该`scope`部分列出了您在之前的任务中选择的精细范围。例如：`scope=read%3Ajira-work%20read%3Ajira-user%20offline_access`

   `offline_access`表示您要生成`refresh_token`。

7. 打开 Web 浏览器窗口，然后在浏览器窗口的地址栏中输入您复制的授权 URL。
8. 目标页面打开后，请验证信息是否正确，然后选择 "接受" 以重定向到您的 Jira 或 Confluence 主页。
9. 主页加载完毕后，复制此页面的 URL。它包含您的应用程序的授权码。您可以使用此代码生成访问令牌。后面的整个部分`code=`是授权码。
10. 使用以下 curl 命令生成访问令牌。将 *`placeholder values`* 替换为您自己的信息。

> **Tip**
>
> 您也可以使用第三方服务，例如 Postman。

```
curl --request POST --url 'https://auth.atlassian.com/oauth/token' \
--header 'Content-Type: application/json' \
--data '{"grant_type": "authorization_code",
"client_id": "YOUR_CLIENT_ID",
"client_secret": "YOUR_CLIENT_SECRET",
"code": "AUTHORIZATION_CODE",
"redirect_uri": "YOUR_CALLBACK_URL"}'
```

对此命令的响应包括`access_code`和的值`refresh_token`。

在 Amaz OpenSearch on DynamoDB 上使用采集管道

您可以使用 D ynamoDB 插件将表事件（例如创建、更新和删除）流式传输到亚马逊服务域和 OpenSearch 亚马逊 OpenSearch 无服务器集合。该管道使用变更数据捕获 (CDC) 进行大规模、低延迟的流式传输。

无论有没有完整的初始快照，您都可以处理 DynamoDB 数据。

- 有了完整快照，DynamoDB point-in-time 使用恢复 (PITR) 来创建备份并将其上传到 Amazon S3。OpenSearch 然后，摄取将快照索引到一个或多个 OpenSearch索引中。为了保持一致性，管道会将所有 DynamoDB 更改与同步。 OpenSearch此选项要求您同时启用 PITR 和 DynamoDB Streams。
- 没有快照 — OpenSearch Ingestion 仅流式传输新的 DynamoDB 事件。如果您已经有快照或者需要在没有历史数据的情况下进行实时流式传输，请选择此选项。此选项要求您仅启用 DynamoDB Streams。

有关更多信息，请参阅《开发者指南》中的 DynamoDB 零 ETL 与 OpenSearch 亚马逊服务的集成。Amazon DynamoDB

主题

- 先决条件
- 步骤 1：配置管道角色
- 步骤 2：创建管道
- 数据一致性
- 映射数据类型
- 限制
- DynamoDB 的推荐 CloudWatch 警报

先决条件

要设置管道，您必须有一个已启用 DynamoDB Streams 的 DynamoDB 表。您的流应使用 `NEW_IMAGE` 流视图类型。但是，`NEW_AND_OLD_IMAGES`如果这种流视图类型适合您的用例，OpenSearch Ingestion 管道也可以使用流式传输事件。

如果您使用的是快照，则还必须在表上启用 point-in-time恢复。有关更多信息，请参阅 Amazon DynamoD B 开发者指南中的创建表、启用 point-in-time恢复和启用流。

步骤 1：配置管道角色

设置 DynamoDB 表后，设置要在管道配置中使用的管道角色，并在该角色中添加以下 DynamoDB 权限：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowRunExportJob",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:ExportTableToPointInTime"
            ],
            "Resource": [
                "arn:aws:dynamodb:region:account-id:table/my-table"
            ]
        },
        {
            "Sid": "allowCheckExportjob",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeExport"
            ],
            "Resource": [
                "arn:aws:dynamodb:region:account-id:table/my-table/export/*"
            ]
        },
        {
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeStream",
                "dynamodb:GetRecords",
                "dynamodb:GetShardIterator"
            ],
            "Resource": [
                "arn:aws:dynamodb:region:account-id: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/*"
            ]
        }
    ]
}
```

您也可以使用 Amazon KMS 客户管理的密钥对导出数据文件进行加密。要解密导出的对象，请在管道的导出配置中指定 `s3_sse_kms_key_id` 作为密钥 ID，格式如下：`arn:aws:kms:region:account-id:key/my-key-id`。以下策略包括使用客户自主管理型密钥时的必需权限：

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

步骤 2：创建管道

然后，您可以配置如下所示的 OpenSearch 采集管道，将 DynamoDB 指定为来源。此示例管道使用 PITR 快照从 `table-a` 中摄取数据，然后从 DynamoDB Streams 摄取事件。`LATEST` 的起始位置指示管道应从 DynamoDB Streams 读取最新数据。

```
version: "2"
cdc-pipeline:
  source:
    dynamodb:
      tables:
      - table_arn: "arn:aws:dynamodb:region:account-id:table/table-a"
        export:
          s3_bucket: "my-bucket"
```

```
          s3_prefix: "export/"
        stream:
          start_position: "LATEST"
      aws:
        region: "us-east-1"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.region.es.amazonaws.com"]
      index: "${getMetadata(\"table-name\")}"
      index_type: custom
      normalize_index: true
      document_id: "${getMetadata(\"primary_key\")}"
      action: "${getMetadata(\"opensearch_action\")}"
      document_version: "${getMetadata(\"document_version\")}"
      document_version_type: "external"
```

您可以使用预先配置的 DynamoDB 蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

数据一致性

OpenSearch Ingestion 支持 end-to-end确认，以确保数据的持久性。管道读取快照或流时，它会动态创建分区以进行并行处理。当管道在摄取 OpenSearch 域或集合中的所有记录后收到确认信息时，该管道会将该分区标记为已完成。

如果要收录到 OpenSearch 无服务器搜索集合中，可以在管道中生成文档 ID。如果要采集到 OpenSearch 无服务器时间序列集合中，请注意该管道不会生成文档 ID。

In OpenSearch gestion 管道还将传入的事件操作映射到相应的批量索引操作中，以帮助采集文档。这样可以保持数据的一致性，因此 DynamoDB 中的每一次数据更改都与中的相应文档更改保持一致。OpenSearch

映射数据类型

OpenSearch 服务将每个传入文档中的数据类型动态映射到 DynamoDB 中的相应数据类型。下表显示了 S OpenSearch ervice 如何自动映射各种数据类型。

数据类型	OpenSearch	DynamoDB
数字	OpenSearch 自动映射数值数据。如果该数字是整数，则将其 OpenSearch 映	DynamoDB 支持数字。

数据类型	OpenSearch	DynamoDB
	射为长值。如果数字是小数，则将其 OpenSearch 映射为浮点值。 OpenSearch 根据第一个发送的文档动态映射各种属性。如果您在 DynamoDB 中为同一属性混合了多种数据类型（例如整数和小数），则映射可能会失败。 例如，如果您的第一个文档的属性为整数，而后来的文档具有与小数相同的属性，OpenSearch 则无法收录第二个文档。在这些情况下，应提供一个显式的映射模板，如下所示： `{ "template": { "mappings": { "properties": { "MixedNumberAttribute": { "type": "float" } } } } }` 如果需要双精度，请使用字符串类型字段映射。不存在支持 38 位精度的等效数字类型 OpenSearch。	
数字集	OpenSearch 自动将数字集映射到由长值或浮点值组成的数组中。与标量数字一样，这取决于摄取的第一个数字是整数还是小数。您可以像映射标量字符串一样提供数字集的映射。	DynamoDB 支持表示数字集的类型。

数据类型	OpenSearch	DynamoDB
字符串	OpenSearch 自动将字符串值映射为文本。在某些情况下（例如枚举值），您可以映射到关键字类型。 以下示例说明如何将`PartType`名为的 DynamoDB 属性映射到关键字。OpenSearch `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	DynamoDB 支持字符串。
字符串集	OpenSearch 自动将字符串集映射到字符串数组中。您可以像映射标量字符串一样提供字符串集的映射。	DynamoDB 支持表示字符串集的类型。

数据类型	OpenSearch	DynamoDB
二元	OpenSearch 自动将二进制数据映射为文本。您可以提供映射以将它们写成二进制字段 OpenSearch。 以下示例说明如何将`ImageData` 名为的 DynamoDB 属性映射到 OpenSearch 二进制字段。 `{ "template": { "mappings": { "properties": { "ImageData": { "type": "binary" } } } } }`	DynamoDB 支持二进制类型属性。
二进制集	OpenSearch 自动将二进制集作为文本映射到二进制数据数组中。您可以像映射标量二进制一样提供数字集的映射。	DynamoDB 支持表示二进制值集的类型。
布尔值	OpenSearch 将 DynamoDB 布尔类型映射为 OpenSearch 布尔类型。	DynamoDB 支持布尔类型属性。
Null	OpenSearch 可以采集 DynamoDB 空类型的文档。它将该值作为空值保存在文档中。此类型没有映射，并且此字段未编制索引或不可搜索。 如果对空类型使用相同的属性名称，然后更改为其他类型（例如字符串），则会为第一个非空值 OpenSearch 创建动态映射。后续值仍然可以是 DynamoDB 空值。	DynamoDB 支持空类型属性。

数据类型	OpenSearch	DynamoDB
Map	OpenSearch 将 DynamoDB 映射属性映射到嵌套字段。嵌套字段内也适用相同的映射。 以下示例将嵌套字段中的字符串映射到中的关键字类型 OpenSearch： `{ "template": { "mappings": { "properties": { "AdditionalDescriptions": { "properties": { "PartType": { "type": "keyword" } } } } } } }`	DynamoDB 支持映射类型属性。
列表	OpenSearch 根据列表中的内容，为 DynamoDB 列表提供了不同的结果。 当列表包含所有相同类型的标量类型（例如，所有字符串的列表）时，则将该列表作为 OpenSearch 该类型的数组提取。这适用于字符串、数字、布尔值和空类型。其中每种类型的限制与该类型标量的限制相同。 您还可以使用与映射相同的映射，为映射列表提供映射。 您无法提供混合类型的列表。	DynamoDB 支持列表类型属性。

数据类型	OpenSearch	DynamoDB
设置	OpenSearch 根据 DynamoDB 集合中的内容，为 DynamoDB 集提供不同的结果。 当一个集合包含所有相同类型的标量类型（例如，所有字符串的集合）时，则将该集合作为该类型的数组进行 OpenSearch 摄取。这适用于字符串、数字、布尔值和空类型。其中每种类型的限制与该类型标量的限制相同。 您还可以使用与映射相同的映射，为映射集合提供映射。 您无法提供混合类型的集合。	DynamoDB 支持表示集合的类型。

我们建议您在摄取管道中配置死信队列 (DLQ)。 OpenSearch 如果您已配置队列，S OpenSearch ervice 会将所有因动态映射失败而无法载入的失败文档发送到队列。

如果自动映射失败，则可以在管道配置中使用 `template_type` 和 `template_content` 来定义显式映射规则。或者，您可以在启动管道之前直接在搜索域或集合中创建映射模板。

限制

在为 DynamoDB 设置 OpenSearch 摄取管道时，请考虑以下限制：

- OpenSearch 采集与 DynamoDB 的集成目前不支持跨区域接入。您的 DynamoDB 表 OpenSearch 和摄取管道必须相同。 Amazon Web Services 区域
- 您的 DynamoDB 表 OpenSearch 和摄取管道必须相同。 Amazon Web Services 账户
- 一个 OpenSearch 摄取管道仅支持一个 DynamoDB 表作为其来源。
- DynamoDB Streams 仅在日志中存储最多 24 小时的数据。如果从大型表的初始快照中摄取数据需要 24 小时或更长时间，则会丢失一些初始数据。为了缓解这种数据丢失，请估计表的大小并配置适当的 OpenSearch 摄取管道计算单元。

DynamoDB 的推荐 CloudWatch 警报

建议使用以下 CloudWatch 指标来监控您的摄取管道的性能。这些指标可能有助您确定处理的导出数据量、处理的流事件量、处理导出和流事件时的错误数以及写入目标的文档数量。您可以设置 CloudWatch 警报，以便在其中一个指标在指定时间内超过指定值时执行操作。

指标	描述
`dynamodb-pipeline.BlockingBuffer.bufferUsage.value`	指示正在使用的缓冲区数量。
`dynamodb-pipeline.dynamodb.activeExportS3ObjectConsumers.value`	显示正在积极处理待导出 OCUs 的 Amazon S3 对象的总数。
`dynamodb-pipeline.dynamodb.bytesProcessed.count`	处理的 DynamoDB 源字节数。
`dynamodb-pipeline.dynamodb.changeEventsProcessed.count`	处理的 DynamoDB 流更改事件数量。
`dynamodb-pipeline.dynamodb.changeEventsProcessingErrors.count`	处理的 DynamoDB 更改事件中的错误数。
`dynamodb-pipeline.dynamodb.exportJobFailure.count`	失败的导出任务提交尝试次数。
`dynamodb-pipeline.dynamodb.exportJobSuccess.count`	已成功提交的导出任务数量。
`dynamodb-pipeline.dynamodb.exportRecordsProcessed.count`	处理的导出记录总数。
`dynamodb-pipeline.dynamodb.exportRecordsTotal.count`	从 DynamoDB 导出的记录总数，这对于跟踪数据导出量至关重要。
`dynamodb-pipeline.dynamodb.exportS3ObjectsProcessed.count`	从 Amazon S3 成功处理的导出数据文件总数。

指标	描述
dynamodb-pipeline.opensearch.bulkBadRequestErrors.count	批量请求期间由于请求格式错误而导致的错误计数。
dynamodb-pipeline.opensearch.bulkRequestLatency.avg	向发出的批量写入请求的平均延迟 OpenSearch。
dynamodb-pipeline.opensearch.bulkRequestNotFoundErrors.count	由于找不到目标数据而失败的批量请求数。
dynamodb-pipeline.opensearch.bulkRequestNumberOfRetries.count	采集管道为写 OpenSearch 入集群而进行的重试次数。 OpenSearch
dynamodb-pipeline.opensearch.bulkRequestSizeBytes.sum	向发出的所有批量请求的总大小（以字节为单位） OpenSearch。
dynamodb-pipeline.opensearch.documentErrors.count	向发送文档时出现的错误数 OpenSearch。导致错误的文件将发送给 DLQ。
dynamodb-pipeline.opensearch.documentsSuccess.count	成功写入 OpenSearch 集群或集合的文档数。
dynamodb-pipeline.opensearch.documentsSuccessFirstAttempt.count	第一次尝试成功编入索引 OpenSearch 的文档数。
dynamodb-pipeline.opensearch.documentsVersionConflictErrors.count	处理过程中由于文档版本冲突而导致的错误计数。
dynamodb-pipeline.opensearch.PipelineLatency.avg	OpenSearch Ingestion 管道通过从源读取数据到写入目标来处理数据的平均延迟。
dynamodb-pipeline.opensearch.PipelineLatency.max	通过从源读取数据到写入目标来处理数据的 OpenSearch Ingestion 管道的最大延迟。

指标	描述
dynamodb-pipeline.opensearch.recordsIn.count	成功摄入 OpenSearch的记录数。该指标对于跟踪正在处理和存储的数据量至关重要。
dynamodb-pipeline.opensearch.s3.dlqS3RecordsFailed.count	未能写入 DLQ 的记录数。
dynamodb-pipeline.opensearch.s3.dlqS3RecordsSuccess.count	写入 DLQ 的记录数。
dynamodb-pipeline.opensearch.s3.dlqS3RequestLatency.count	Amazon S3 死信队列请求的延迟测量次数。
dynamodb-pipeline.opensearch.s3.dlqS3RequestLatency.sum	Amazon S3 死信队列的所有请求的总延迟
dynamodb-pipeline.opensearch.s3.dlqS3RequestSizeBytes.sum	向 Amazon S3 死信队列发出的所有请求的总大小（以字节为单位）。
dynamodb-pipeline.recordsProcessed.count	管道中处理的记录总数，这是衡量整体吞吐量的关键指标。
dynamodb.changeEventsProcessed.count	没有从 DynamoDB 流收集任何记录。这可能是由于桌上没有活动、正在进行导出或访问 DynamoDB 直播时出现问题。
dynamodb.exportJobFailure.count	尝试触发向 S3 的导出操作失败。
dynamodb-pipeline.opensearch.bulkRequestInvalidInputErrors.count	OpenSearch 由于输入无效而导致的批量请求错误计数，这对于监控数据质量和操作问题至关重要。

指标	描述
`opensearch.EndToEndLatency.avg`	端到端延迟高于从 DynamoDB 数据流读取所需的延迟。这可能是由于 OpenSearch 集群规模不足，或者管道 OCU 的最大容量对于 DynamoDB 表上的 WCU 吞吐量来说过低。这种端到端的延迟在导出后会很高，而且随着时间的推移，它会赶上最新的 DynamoDB 流，因此会随着时间的推移而降低。

在 Amazon DocumentDB 中使用 OpenSearch 摄取管道

您可以使用 D ocumentDB 插件将文档更改（例如创建、更新和删除）流式传输到亚马逊 OpenSearch 服务。该管道支持变更数据捕获 (CDC)（如果有），或者支持高规模、低延迟流媒体的 API 轮询。

无论有没有完整的初始快照，您都可以处理数据。完整快照会捕获整个 Amazon DocumentDB 馆藏并将其上传到 Amazon S3。然后，管道将数据发送到一个或多个 OpenSearch 索引。在摄取快照后，管道会同步正在进行的更改以保持一致性，并最终赶上近乎实时的更新。

如果您已经有来自其他来源的完整快照，或者只需要处理新事件，则可以在没有快照的情况下进行流式传输。在这种情况下，管道直接从 Amazon DocumentDB 变更流读取，无需初始批量加载。

如果您启用直播，则必须在 Amazon DocumentDB 馆藏上启用更改流。但是，如果您只执行满载或导出，则不需要更改流。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 按照《Amazon DocumentDB 开发者指南》中 Create an Amazon DocumentDB cluster 说明的步骤操作，创建一个具有数据读取权限的 Amazon DocumentDB 集群。如果您使用 CDC 基础设施，请将您的 Amazon DocumentDB 集群配置为发布变更流。
2. 在 Amazon DocumentDB 集群上启用 TLS。
3. 设置私有地址空间的 VPC CIDR 以用于 OpenSearch 接入。
4. 使用在您的亚马逊文档数据库集群上设置身份验证。 Amazon Secrets Manager按照自动轮换 Amazon DocumentDB 密码中的步骤启用密钥轮换。有关更多信息，请参阅 Amazon DocumentDB 中使用基于角色的访问控制和安全访问数据库。

5. 如果您使用更改流订阅 Amazon DocumentDB 集合中的数据更改，请使用 `change_stream_log_retention_duration` 参数将保留期延长至最多 7 天，以避免数据丢失。默认情况下，更改流事件将在事件录制后存储 3 小时，这对于大型集合来说是不够的。要修改更改流保留期，请参阅修改更改流日志保留期限。
6. 创建 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅the section called “创建 OpenSearch 服务域”和the section called “创建集合”。
7. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的 Amazon DocumentDB 集群写入您的域或集合。

   以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

   JSON

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

   要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

完成 Amazon DocumentDB 管道先决条件设置后，配置管道角色以在管道配置中使用，并为该角色添加以下 Amazon DocumentDB 权限：

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:region:account-id:secret:secret-
name"]
        },
        {
            "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/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals":
                    {
                        "aws:RequestTag/OSISManaged": "true"
                    }
            }
        }
    ]
}
```

您必须为用于创建 OpenSearch 摄取管道的 IAM 角色提供上述 Amazon EC2 权限，因为管道使用这些权限在您的 VPC 中创建和删除网络接口。该管道只能通过此网络接口访问 Amazon DocumentDB 集群。

步骤 2：创建管道

然后，您可以配置如下所示的 OpenSearch 摄取管道，该管道将 Amazon DocumentDB 指定为来源。请注意，为了填充索引名称，`getMetadata` 函数会将 *`documentdb_collection`* 用作元数据键。要在不使用 `getMetadata` 方法的情况下使用其他索引名称，则可以使用配置 `index: "my_index_name"`。

```
version: "2"
documentdb-pipeline:
  source:
    documentdb:
      acknowledgments: true
      host: "https://docdb-cluster-id.us-east-1.docdb.amazonaws.com"
      port: 27017
      authentication:
        username: ${aws_secrets:secret:username}
        password: ${aws_secrets:secret:password}
      aws:
      s3_bucket: "bucket-name"
      s3_region: "bucket-region"
      s3_prefix: "path" #optional path for storing the temporary data
      collections:
        - collection: "dbname.collection"
          export: true
          stream: true
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      index: "${getMetadata(\"documentdb_collection\")}"
      index_type: custom
      document_id: "${getMetadata(\"primary_key\")}"
      action: "${getMetadata(\"opensearch_action\")}"
      document_version: "${getMetadata(\"document_version\")}"
      document_version_type: "external"
extension:
  aws:
    secrets:
      secret:
        secret_id: "my-docdb-secret"
```

```
        region: "us-east-1"
        refresh_interval: PT1H
```

您可以使用预先配置的 Amazon DocumentDB 蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

如果您使用创建管道，则还必须将管道连接到 VPC，才能使用 Amazon DocumentDB 作为来源。Amazon Web Services Management Console 为此，请找到 “源网络选项” 部分，选中 “连接到 VPC” 复选框，然后从提供的默认选项中选择您的 CIDR。您可以按照 RFC 1918 当前最佳实践中的定义，使用私有地址空间中的任何 CIDR。

要提供自定义 CIDR，请从下拉菜单中选择其他。为避免 OpenSearch Ingestion 和 Amazon DocumentDB 之间的 IP 地址冲突，请确保亚马逊 DocumentDB VPC CIDR 不同于用于摄取的 CIDR。OpenSearch

有关更多信息，请参阅为管道配置 VPC 访问权限。

数据一致性

该管道通过持续轮询或接收来自 Amazon DocumentDB 集群的更改以及更新索引中的相应文档来确保数据一致性。 OpenSearch

OpenSearch Ingestion 支持 end-to-end确认，以确保数据的持久性。管道读取快照或流时，它会动态创建分区以进行并行处理。当管道在摄取 OpenSearch 域或集合中的所有记录后收到确认信息时，该管道会将该分区标记为已完成。

如果要收录到 OpenSearch 无服务器搜索集合中，可以在管道中生成文档 ID。如果要采集到 OpenSearch Serverless 时间序列集合，请注意管道不会生成文档 ID，因此您必须在工作流接收器配置`document_id: "${getMetadata(\"primary_key\")}"`中省略文档 ID。

In OpenSearch gestion 管道还将传入的事件操作映射到相应的批量索引操作中，以帮助采集文档。这样可以保持数据的一致性，从而使 Amazon DocumentDB 中的每一次数据更改都与中的相应文档更改保持一致。 OpenSearch

映射数据类型

OpenSearch 服务将每个传入文档中的数据类型动态映射到 Amazon DocumentDB 中的相应数据类型。下表显示了 S OpenSearch ervice 如何自动映射各种数据类型。

数据类型	OpenSearch	Amazon DocumentDB
整数	OpenSearch 自动将 Amazon DocumentDB 的整数值映射到 OpenSearch整数。 OpenSearch 根据第一个发送的文档动态映射字段。如果 Amazon DocumentDB 中的同一属性混合了多种数据类型，则自动映射可能会失败。 例如，如果您的第一个文档的属性很长，而后来的文档具有与整数相同的属性，OpenSearch 则无法收录第二个文档。对于这些情况，应提供选择最灵活数值类型的显式映射模板，如下所示： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB 支持整型。
长整型	OpenSearch 自动将 Amazon DocumentDB 的长整数值映射到 OpenSearch多头。 OpenSearch 根据第一个发送的文档动态映射字段。如果 Amazon DocumentDB 中的同一属性混合了多种数据类型，则自动映射可能会失败。	Amazon DocumentDB 支持长整型。

数据类型	OpenSearch	Amazon DocumentDB
	例如，如果您的第一个文档的属性很长，而后来的文档具有与整数相同的属性，OpenSearch 则无法收录第二个文档。对于这些情况，应提供选择最灵活数值类型的显式映射模板，如下所示： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	
字符串	OpenSearch 自动将字符串值映射为文本。在某些情况下（例如枚举值），您可以映射到关键字类型。 以下示例说明如何将名为的 Amazon DocumentDB 属性映射`PartType`到关键词。 OpenSearch `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	Amazon DocumentDB 支持字符串。

数据类型	OpenSearch	Amazon DocumentDB
双精度	OpenSearch 自动将 Amazon DocumentDB 双精度值映射到双精度值。OpenSearch OpenSearch 根据第一个发送的文档动态映射字段。如果 Amazon DocumentDB 中的同一属性混合了多种数据类型，则自动映射可能会失败。 例如，如果您的第一个文档的属性很长，而后来的文档具有与整数相同的属性，OpenSearch 则无法收录第二个文档。对于这些情况，应提供选择最灵活数值类型的显式映射模板，如下所示： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB 支持双精度浮点数。

数据类型	OpenSearch	Amazon DocumentDB
日期	默认情况下，日期映射到中的整数 OpenSearch。您可以定义自定义映射模板以将日期映射到 OpenSearch 日期。 `{ "template": { "mappings": { "properties": { "myDateField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB 支持日期。
Timestamp	默认情况下，时间戳映射到中的整数。OpenSearch您可以定义自定义映射模板以将日期映射到 OpenSearch日期。 `{ "template": { "mappings": { "properties": { "myTimestampField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB 支持时间戳。
布尔值	OpenSearch 将 Amazon DocumentDB 布尔类型映射为 OpenSearch 布尔类型。	Amazon DocumentDB 支持布尔类型属性。

数据类型	OpenSearch	Amazon DocumentDB
十进制	OpenSearch 将 Amazon DocumentDB 映射属性映射到嵌套字段。嵌套字段内也适用相同的映射。 以下示例将嵌套字段中的字符串映射到中的关键字类型 OpenSearch： `{ "template": { "mappings": { "properties": { "myDecimalField": { "type": "double" } } } } }` 使用此自定义映射，您可以对该字段进行双精度查询和聚合。原始值保留了 OpenSearch文档`_source`属性的全部精度。如果没有此映射，则默认 OpenSearch 使用文本。	Amazon DocumentDB 支持小数。
正则表达式	正则表达式类型会创建嵌套字段。这包括 *<myFieldName>* `.pattern` 和 *<myFieldName>* `.options`。	Amazon DocumentDB 支持正则表达式。

数据类型	OpenSearch	Amazon DocumentDB
二进制数据	OpenSearch 自动将 Amazon DocumentDB 二进制数据映射到 OpenSearch文本。您可以提供映射以将它们写成二进制字段 OpenSearch。 以下示例说明如何将名为的 Amazon DocumentDB 字段映射`imageData` 到 OpenSearch 二进制字段。 `{ "template": { "mappings": { "properties": { "imageData": { "type": "binary" } } } } }`	Amazon DocumentDB 支持二进制数据字段。
ObjectId	具有 ObjectID 类型的字段会映射到 OpenSearch 文本字段。该值将是 objectId 的字符串表示形式。	Amazon DocumentDB 支持 objectIds。
Null	OpenSearch 可以收录 Amazon DocumentDB 空类型的文档。它将该值作为空值保存在文档中。此类型没有映射，并且此字段未编制索引或不可搜索。 如果对空类型使用相同的属性名称，然后更改为其他类型（例如字符串），则会为第一个非空值 OpenSearch 创建动态映射。后续值仍然可以是 Amazon DocumentDB 空值。	Amazon DocumentDB 支持空值类型字段。

数据类型	OpenSearch	Amazon DocumentDB
未定义	OpenSearch 可以收录 Amazon DocumentDB 未定义类型的文档。它将该值作为空值保存在文档中。此类型没有映射，并且此字段未编制索引或不可搜索。 如果对未定义的类型使用相同的字段名称，然后更改为其他类型（例如字符串），则会为第一个未定义的值 OpenSearch 创建动态映射。后续值仍然可以是 Amazon DocumentDB 未定义值。	Amazon DocumentDB 支持未定义类型字段。
MinKey	OpenSearch 可以收录亚马逊 DocumentDB minKey 类型的文档。它将该值作为空值保存在文档中。此类型没有映射，并且此字段未编制索引或不可搜索。 如果对 minKey 类型使用相同的字段名称，然后更改为其他类型（例如字符串），则会为第一个非 MinKey 值 OpenSearch 创建动态映射。后续值仍然可以是 Amazon DocumentDB minKey 值。	Amazon DocumentDB 支持 minKey 类型字段。
MaxKey	OpenSearch 可以收录亚马逊 DocumentDB MaxKey 类型的文档。它将该值作为空值保存在文档中。此类型没有映射，并且此字段未编制索引或不可搜索。 如果对 MaxKey 类型使用相同的字段名称，然后更改为其他类型（例如字符串），则会为第一个非 MaxKey 值 OpenSearch 创建动态映射。后续值仍然可以是 Amazon DocumentDB maxKey 值。	Amazon DocumentDB 支持 maxKey 类型字段。

我们建议您在摄取管道中配置死信队列 (DLQ)。 OpenSearch 如果您已配置队列，S OpenSearch ervice 会将所有因动态映射失败而无法载入的失败文档发送到队列。

如果自动映射失败，则可以在管道配置中使用 `template_type` 和 `template_content` 来定义显式映射规则。或者，您可以在启动管道之前直接在搜索域或集合中创建映射模板。

限制

在为 Amazon DocumentDB 设置 OpenSearch 摄取管道时，请考虑以下限制：

- OpenSearch Ingestion 与 Amazon DocumentDB 的集成目前不支持跨区域提取。您的 Amazon DocumentDB 集群和 OpenSearch 摄取管道必须处于相同的位置。 Amazon Web Services 区域
- OpenSearch Ingestion 与 Amazon DocumentDB 的集成目前不支持跨账户提取。您的 Amazon DocumentDB 集群和 OpenSearch 摄取管道必须处于相同的位置。 Amazon Web Services 账户
- 一个 OpenSearch 摄取管道仅支持一个 Amazon DocumentDB 集群作为其来源。
- OpenSearch Ingestion 与亚马逊 DocumentDB 的集成特别支持基于亚马逊文档数据库实例的集群。但不支持 Amazon DocumentDB 弹性集群。
- OpenSearch Ingestion 集成仅支持 Amazon Secrets Manager 作为您的 Amazon DocumentDB 集群的身份验证机制。
- 无法通过更新现有管道配置来从不同的数据库或集合摄取数据。相反，必须创建新管道。

推荐的 CloudWatch 警报

为了获得最佳性能，我们建议您在创建 OpenSearch 采集管道时使用以下 CloudWatch 警报来访问作为来源的 Amazon DocumentDB 集群。

CloudWatch 警报	描述
<pipeline-name> .docucmentdb.凭证已更改	该指标表示 Amazon 密钥的轮换频率。
<pipeline-name> .docmentdb。 executorRefreshErrors	该指标指示刷新 Amazon 密钥失败。
<pipeline-name> .docmentdb。 exportRecordsTotal	该指标指示从 Amazon DocumentDB 导出的记录数量。
<pipeline-name> .docmentdb。 exportRecordsProcessed	该指标表示 OpenSearch 摄取管道处理的记录数。

CloudWatch 警报	描述
<pipeline-name> .docmentdb。 exportRecordProcessing错误	该指标表示从 Amazon DocumentDB 集群读取数据时 OpenSearch 摄取管道中出现的处理错误数量。
<pipeline-name> .docmentdb。 exportRecordsSuccess总计	该指标指示成功处理的导出记录总数。
<pipeline-name> .docmentdb。 exportRecordsFailed总计	该指标指示处理失败的导出记录总数。
<pipeline-name> .docucmentdb.bytes 已收到	该指标表示 OpenSearch 摄取管道接收的总字节数。
<pipeline-name> .docucmentdb.bytes已处理	该指标表示 OpenSearch 摄取管道处理的总字节数。
<pipeline-name> .docmentdb。 exportPartitionQuery总计	该指标指示导出分区总数。
<pipeline-name> .docmentdb。 streamRecordsSuccess总计	该指标指示该流中成功处理的记录总数。
<pipeline-name> .docmentdb。 streamRecordsFailed总计	该指标指示该流中处理失败的记录总数。

在 Confluent Cl OpenSearch oud Kafka 中使用采集管道

您可以使用 OpenSearch 摄取管道将数据从 Confluent Cloud Kafka 集群流式传输到 OpenSearch 亚马逊服务域和无服务器集合。 OpenSearch OpenSearch Ingestion 支持公用和私有网络配置，用于将数据从 Confluent Cloud Kafka 集群流式传输到由服务或无服务器管理的域或集合。 OpenSearch OpenSearch

连接到 Confluent Cloud 公有 Kafka 集群

您可以使用 OpenSearch 摄取管道从具有公共配置的 Confluent Cloud Kafka 集群迁移数据，这意味着可以公开解析域 DNS 名称。为此，请设置一个 OpenSearch 采集管道，将 Confluent Cloud 公共

Kafka 集群作为源，将 OpenSearch 服务或 OpenSearch 无服务器作为目标。这将处理您从自行管理的源集群到托 Amazon管目标域或集合的流式数据。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 创建充当源的 Confluent Cloud Kafka 集群。集群应包含您要提取到 S OpenSearch ervice 中的数据。
2. 创建要将数据迁移到的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅the section called “ 创建 OpenSearch 服务域”和the section called “创建集合”。
3. 使用 Amazon Secrets Manager在 Confluent Cloud Kafka 集群上设置身份验证。按照 Rotate Amazon Secrets Manager secrets 中的步骤启用密钥轮换。
4. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的自管理集群写入您的域或集合。

   以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

   JSON

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

要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

设置 Confluent Cloud Kafka 集群管道先决条件后，配置要在管道配置中使用的管道角色，添加写入 OpenSearch 服务域或 OpenSearch 无服务器集合的权限，以及从 Secrets Manager 读取密钥的权限。

管理网络接口需要以下权限：

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:*:account-id:network-interface/*",
                "arn:aws:ec2:*:account-id:subnet/*",
                "arn:aws:ec2:*:account-id: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" }
            }
        }
    ]
}
```

以下是从 Amazon Secrets Manager 服务中读取机密所需的权限：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": ["secretsmanager:GetSecretValue"],
            "Resource": ["arn:aws:secretsmanager:region:account-
id:secret:,secret-name"]
        }
    ]
}
```

写入亚马逊 OpenSearch 服务域需要以下权限：

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-id:role/pipeline-role"
```

```
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

步骤 2：创建管道

然后，你可以配置如下所示的 OpenSearch 摄取管道，它将你的 Confluent Cloud Kafka 指定为来源。

您可以将多个 OpenSearch 服务域指定为数据的目的地。此功能允许有条件地路由或将传入数据复制到多个 OpenSearch 服务域。

您还可以将数据从源 Confluent Kafka 集群迁移到无服务器 V OpenSearch PC 集合。务必要在管道配置中提供网络访问策略。您可以使用 Confluent 架构注册表来定义 Confluent 架构。

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

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

您可以使用预先配置的蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

连接到 VPC 中的 Confluent Cloud Kafka 集群

您还可以使用 OpenSearch 摄取管道从 VPC 中运行的 Confluent Cloud Kafka 集群迁移数据。为此，请设置一个 OpenSearch 采集管道，将 Confluent Cloud Kafka 集群作为源，将 OpenSearch 服务或 OpenSearch 无服务器作为目标。这将处理从 Confluent Cloud Kafka 源集群到 Amazon托管式目标域或集合的流式数据传输。

OpenSearch Ingestion 支持在 Confluent 中以所有支持的网络模式配置的 Confluent Cloud Kafka 集群。 OpenSearch Ingestion 支持以下网络配置模式作为来源：

- Amazon VPC 对等连接
- Amazon PrivateLink 适用于专用集群
- Amazon PrivateLink 适用于企业集群
- Amazon Transit Gateway

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 使用 VPC 网络配置创建 Confluent Cloud Kafka 集群，其中包含您要提取到服务中的数据。OpenSearch
2. 创建要将数据迁移到的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅。有关更多信息，请参阅the section called “ 创建 OpenSearch 服务域”和the section called “创建集合”。
3. 使用 Amazon Secrets Manager在 Confluent Cloud Kafka 集群上设置身份验证。按照 Rotate Amazon Secrets Manager secrets 中的步骤启用密钥轮换。

4. 获取有权访问 Confluent Cloud Kafka 集群的 VPC 的 ID。选择 OpenSearch Ingestion 要使用的 VPC CIDR。

> **Note**
>
> 如果您使用创建管道，则还必须将您的 OpenSearch 摄取管道连接到您的 VPC 才能使用 Confluent Cloud Kafka 集群。 Amazon Web Services Management Console 要执行此操作，请找到网络配置部分，选择连接到 VPC 复选框，然后从提供的任意一个默认选项中选择 CIDR，或者选择自己的 CIDR。您可以按照 RFC 1918 当前最佳实践中的定义，使用私有地址空间中的任何 CIDR。
> 要提供自定义 CIDR，请从下拉菜单中选择其他。为避免接入和自行管理之间的 IP 地址冲突 OpenSearch，请确保自管理 VP OpenSearch C CIDR 与用于 OpenSearch 摄取的 CIDR 不同。 OpenSearch

5. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的自管理集群写入您的域或集合。

> **Note**
>
> 如果使用 Amazon PrivateLink 连接 Confluent Cloud Kafka，则需要配置 VPC DHCP 选项。应启用 DNS 主机名和 DNS 解析。
> 具体而言，请使用以下选项集值：
>
> ```
> domain-name: aws.private.confluent.cloud
> domain-name-servers: AmazonProvidedDNS
> ```
>
> 此更改可确保 Confluent PrivateLink 终端节点的 DNS 解析在 VPC 内正常运行。

以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

JSON

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

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

要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

完成管道先决条件设置后，配置管道角色以在管道配置中使用，并为该角色添加以下权限：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": ["arn:aws:secretsmanager:region:account-id:secret:secret-
name"]
        },
        {
            "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/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals":
                    {
                        "aws:RequestTag/OSISManaged": "true"
                    }
            }
        }
    ]
}
```

您必须为用于创建 OpenSearch 摄取管道的 IAM 角色提供上述 Amazon EC2 权限，因为管道使用这些权限在您的 VPC 中创建和删除网络接口。该管道只能通过此网络接口访问 Kafka 集群。

步骤 2：创建管道

然后，您可以配置如下所示的 OpenSearch 采集管道，将 Kafka 指定为来源。

您可以将多个 OpenSearch 服务域指定为数据的目的地。此功能允许有条件地路由或将传入数据复制到多个 OpenSearch 服务域。

您还可以将数据从源 Confluent Kafka 集群迁移到无服务器 V OpenSearch PC 集合。务必要在管道配置中提供网络访问策略。您可以使用 Confluent 架构注册表来定义 Confluent 架构。

```
 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"
```

将 OpenSearch 摄取管道与 Amazon Managed Streaming for Apache Kafka

你可以使用 Kafka 插件将来自亚马逊 Apache Managed Streaming for Apache Kafka（亚马逊 MS OpenSearch K）的数据提取到你的摄取管道中。在 Amazon MSK 中，您可以构建并运行使用 Apache Kafka 的应用程序来处理流数据。 OpenSearch Ingestion Amazon PrivateLink 用于连接亚马逊 MSK。您可以从 Amazon MSK 和 Amazon MSK 无服务器集群摄取数据。这两个流程之间的唯一区别是在设置管道之前必须执行的先决条件步骤。

主题

- 预配置的 Amazon MSK 先决条件
- Amazon MSK 无服务器先决条件
- 步骤 1：配置管道角色
- 步骤 2：创建管道
- 步骤 3：（可选）使用 Amazon Glue 架构注册表
- 步骤 4：（可选）为 Amazon MSK 管道配置推荐的计算单位 (OCUs)

预配置的 Amazon MSK 先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 按照《Amazon Managed Streaming for Apache Kafka 开发人员指南》中 Creating a cluster 说明的步骤，创建一个由 Amazon MSK 预置的集群。对于 Broker 类型，请选择除`t3`类型之外的任何选项，因为 OpenSearch Ingestion 不支持这些类型。
2. 集群处于活动状态后，请按照开启多 VPC 连接中的步骤执行操作。
3. 按照将集群策略附加到 MSK 集群的步骤附加以下策略之一，具体取决于集群与管道是否位于同一 Amazon Web Services 账户。此策略允许 OpenSearch Ingestion 创建与您的 Amazon MSK 集群的 Amazon PrivateLink 连接并从 Kafka 主题中读取数据。确保使用自身 ARN 更新 `resource`。

   当集群与管道位于同一 Amazon Web Services 账户时，适用以下策略：

   JSON

```
{
```

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

如果您的 Amazon MSK 集群与您的管道 Amazon Web Services 账户 不同，请改为附加以下策略。请注意，只有预置的 Amazon MSK 集群才能进行跨账户访问，Amazon MSK 无服务器集群不支持跨账户访问。的 ARN Amazon `principal` 应该是您为工作流配置提供的相同管道角色的 ARN：

JSON

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

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

4. 按照创建主题中的步骤创建 Kafka 主题。确保*BootstrapServerString*这是私有终端节点（单 VPC）引导程序 URLs之一。`--replication-factor` 的值应为 2 或 3，具体取决于 Amazon MSK 集群包含的可用区数量。`--partitions` 的值至少应为 10。
5. 按照生成和使用数据中的步骤生成和使用数据。再说一遍，请确保*BootstrapServerString*这是您的私有终端节点（单 VPC）引导程序 URLs之一。

Amazon MSK 无服务器先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 按照《Amazon Managed Streaming for Apache Kafka 开发人员指南》中 Create an MSK Serverless cluster 说明的步骤，创建一个 Amazon MSK 无服务器集群。
2. 集群处于活动状态后，按照 Attach a cluster policy to the MSK cluster 中的步骤附加以下策略。确保使用自身 ARN 更新 `resource`。

   JSON

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

```
        ],
        "Resource": "arn:aws:kafka:region:account-id:cluster/cluster-name/cluster-id"
      }
    ]
}
```

此策略允许 OpenSearch Ingestion 创建与您的 Amazon MSK 无服务器集群的 Amazon PrivateLink 连接并从 Kafka 主题中读取数据。当您的集群和管道处于相同状态时，此政策适用 Amazon Web Services 账户，这必须是正确的，因为 Amazon MSK Serverless 不支持跨账户访问。

3. 按照创建主题中的步骤创建 Kafka 主题。确保*BootstrapServerString*这是您的简单身份验证和安全层 (SASL) IAM 引导程序 URLs之一。`--replication-factor` 的值应为 2 或 3，具体取决于 Amazon MSK 无服务器集群包含的可用区数量。`--partitions` 的值至少应为 10。
4. 按照生成和使用数据中的步骤生成和使用数据。再说一遍，请确保*BootstrapServerString*这是您的简单身份验证和安全层 (SASL) IAM 引导程序 URLs之一。

步骤 1：配置管道角色

预置好 Amazon MSK 集群或设置好无服务器集群后，在管道角色中添加要在管道配置中使用的以下 Kafka 权限：

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:region:account-id:cluster/cluster-name/cluster-id"
            ]
        },
```

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

步骤 2：创建管道

然后，你可以配置如下所示的 OpenSearch 摄取管道，将 Kafka 指定为来源：

```
version: "2"
log-pipeline:
  source:
    kafka:
      acknowledgements: true
      topics:
      - name: "topic-name"
        group_id: "grouplambd-id"
      aws:
        msk:
          arn: "arn:aws:kafka:region:account-id:cluster/cluster-name/cluster-id"
        region: "us-west-2"
  processor:
  - grok:
```

```
      match:
        message:
        - "%{COMMONAPACHELOG}"
  - date:
      destination: "@timestamp"
      from_time_received: true
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
      index: "index_name"
      aws_region: "region"
      aws_sigv4: true
```

您可以使用预先配置的 Amazon MSK 蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

步骤 3：（可选）使用 Amazon Glue 架构注册表

当您将 OpenSearch Ingestion 与 Amazon MSK 配合使用时，可以将 AVRO 数据格式用于架构注册表中托管的架构。 Amazon Glue 在 Amazon Glue 架构注册表中，您可以集中发现、控制和演变数据流架构。

要使用此选项，请在管道配置中启用架构 `type`：

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

您还必须在您的管道角色中提供 Amazon Glue 读取访问权限。您可以使用名为的 Amazon 托管策略AWSGlueSchemaRegistryReadonlyAccess。此外，您的注册表必须 Amazon Web Services 账户与您的 OpenSearch 摄取管道位于同一区域中。

步骤 4：（可选）为 Amazon MSK 管道配置推荐的计算单位 (OCUs)

每个计算单位的每个主题有一个使用者。代理在给定主题的使用者之间均衡分配分区。但是，当分区数量大于使用者数量时，Amazon MSK 将要求每个使用者托管多个分区。 OpenSearch Ingestion 具有内置的 auto Scaling，可以根据 CPU 使用率或管道中的待处理记录数量向上或向下扩展。

为实现最佳性能，请将分区分布在多个计算单位中以便并行处理。如果主题有大量分区（例如，超过 96 个，这是 OCUs 每个管道的最大分区），我们建议您将管道配置为 1— OCUs 96。因为它将根据需要自动扩缩。如果主题包含的分区数量较少（例如，少于 96 个），则最大计算单位应与分区数量相同。

当管道包含多个主题时，请选择分区数最多的主题作为参考来配置最大计算单位。通过向同一主题和使用者组添加另一个具有新集的 OCUs 管道，您可以几乎线性地扩展吞吐量。

在 Amaz OpenSearch on S3 中使用采集管道

借助 OpenSearch Ingestion，您可以将 Amazon S3 用作源或目标。当您使用 Amazon S3 作为数据源时，会将数据发送到 OpenSearch 摄取管道。当您使用 Amazon S3 作为目标时，会将数据从 OpenSearch 摄取管道写入到一个或多个 S3 存储桶。

主题

- Amazon S3 作为源
- Amazon S3 作为目标
- 将 Amazon S3 跨账户作为源

Amazon S3 作为源

您可以通过两种方式使用 Amazon S3 作为源处理数据：S3-SQS 处理和计划扫描。

如果您需要在文件写入 S3 后近实时扫描文件，请使用 S3-SQS 处理。您可以配置 Amazon S3 存储桶，在存储桶中存储或修改对象时随时触发事件。使用一次性扫描或定期计划扫描批处理 S3 存储桶中的数据。

主题

- 先决条件
- 步骤 1：配置管道角色
- 步骤 2：创建管道

先决条件

要使用 Amazon S3 作为预定扫描或 S3-SQS 处理的 OpenSearch 摄取管道的来源，请先创建一个 S3 存储桶。

> **Note**
>
> 如果 OpenSearch Ingestion 管道中用作源的 S3 存储桶位于不同的存储桶中 Amazon Web Services 账户，则还需要对该存储桶启用跨账户读取权限。这样管道将可读取和处理数据。要启用跨账户权限，请参阅 Amazon S3 用户指南中的存储桶拥有者授予跨账户存储桶权限。

如果您的 S3 存储桶位于多个账户中，请使用 `bucket_owners` 映射。有关示例，请参阅 OpenSearch文档中的跨账户 S3 访问权限。

要设置 S3-SQS 处理，还需要执行以下步骤：

1. 创建 Amazon SQS 队列。
2. 在以 SQS 队列为目标的 S3 存储桶上启用事件通知。

步骤 1：配置管道角色

与其他将数据推送到管道的源插件不同，S3 源插件采用基于读取的架构，管道从源中拉取数据。

因此，为使管道能够从 S3 读取，必须在管道的 S3 源配置中指定一个可以同时访问 S3 存储桶和 Amazon SQS 队列的角色。管道将担任此角色，以便从队列中读取数据。

> **Note**
>
> 在 S3 源配置中指定的角色必须是管道角色。因此，管道角色必须包含两个单独的权限策略，一个用于写入接收器，另一个用于从 S3 源中拉取。您必须在所有管道组件中使用相同的 `sts_role_arn`。

以下示例策略显示了使用 S3 作为源所需的权限：

JSON

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

```
    {
       "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:account-id:MyS3EventSqsQueue"
    }
  ]
}
```

必须将以下权限附加到在 S3 源插件配置的 `sts_role_arn` 选项中指定的 IAM 角色：

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

步骤 2：创建管道

设置权限后，您可以根据您的 Amazon S3 用例配置 OpenSearch 摄取管道。

S3-SQS 处理

要设置 S3-SQS 处理，请配置您的管道，指定 S3 作为源并设置 Amazon SQS 通知：

```
version: "2"
s3-pipeline:
  source:
```

```
    s3:
      notification_type: "sqs"
      codec:
        newline: null
      sqs:
        queue_url: "https://sqs.us-east-1amazonaws.com/account-id/ingestion-queue"
      compression: "none"
      aws:
        region: "region"
  processor:
  - grok:
      match:
        message:
        - "%{COMMONAPACHELOG}"
  - date:
      destination: "@timestamp"
      from_time_received: true
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
      index: "index-name"
      aws:
        region: "region"
```

如果您在处理 Amazon S3 上的小文件时发现 CPU 利用率较低，可考虑通过修改 `workers` 选项的值来提高吞吐量。有关更多信息，请参阅 S3 plugin configuration options。

计划扫描

要设置计划扫描，请使用适用于所有 S3 存储桶的扫描级别或存储桶级别的计划来配置管道。存储桶级别计划或扫描间隔配置始终覆盖扫描级别配置。

您可以使用一次性扫描（非常适合数据迁移）或定期扫描（非常适合批处理）配置计划扫描。

要将管道配置为从 Amazon S3 读取，请使用预先配置的 Amazon S3 蓝图。您可以编辑管道配置的 `scan` 部分以满足计划需求。有关更多信息，请参阅 the section called “使用蓝图”。

一次性扫描

一次性计划扫描运行一次。在您的工作流配置中，您可以使用`start_time`和`end_time`来指定何时扫描存储桶中的对象。或者，您也可以使用 `range` 指定相对于当前时间的时间间隔，以该时间间隔扫描存储桶中的对象。

例如，范围设置为 PT4H 将扫描最近四个小时内创建的所有文件。要配置再次运行一次性扫描，必须先停止管道，然后再重新启动。如果未配置范围，则还必须更新开始时间和结束时间。

以下配置为所有存储桶及这些存储桶中的所有对象设置一次性扫描：

```
version: "2"
log-pipeline:
  source:
    s3:
      codec:
        csv:
      compression: "none"
      aws:
        region: "region"
      acknowledgments: true
      scan:
        buckets:
          - bucket:
              name: my-bucket
              filter:
                include_prefix:
                  - Objects1/
                exclude_suffix:
                  - .jpeg
                  - .png
          - bucket:
              name: my-bucket-2
              key_prefix:
                include:
                  - Objects2/
                exclude_suffix:
                  - .jpeg
                  - .png
      delete_s3_objects_on_read: false
  processor:
    - date:
        destination: "@timestamp"
        from_time_received: true
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
        index: "index-name"
        aws:
          region: "region"
```

```
        dlq:
          s3:
            bucket: "dlq-bucket"
            region: "us-east-1"
```

以下配置设置在指定时段内对所有存储桶运行一次性扫描。这意味着 S3 仅处理创建时间在此时段内的对象。

```
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
```

以下配置设置扫描级别和存储桶级别一次性扫描。存储桶级别开始时间和结束时间将覆盖扫描级别开始时间和结束时间。

```
scan:
  start_time: 2023-01-21T18:00:00.000Z
  end_time: 2023-04-21T18:00:00.000Z
  buckets:
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-1
        filter:
          include:
```

```
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

停止管道会移除对管道在停止之前所扫描对象的任何先前引用。如果停止单个扫描管道，则将在启动后重新扫描所有对象，即使这些已被扫描。如果您需要停止单个扫描管道，建议在重新启动管道之前更改时间段。

如果您需要按开始时间和结束时间筛选对象，则停止和启动管道是唯一的选择。如果您不需要按开始时间和结束时间进行筛选，则可以按名称筛选对象。按名称筛选对象不需要停止和启动管道。要执行此操作，请使用 `include_prefix` 和 `exclude_suffix`。

定期扫描

定期计划扫描按定期计划时间间隔对您指定的 S3 存储桶运行扫描。只能在扫描级别配置间隔，因为不支持单独执行存储桶级别配置。

在您的工作流配置中，`interval`指定了重复扫描的频率，可以介于 30 秒到 365 天之间。始终在创建管道时运行首次扫描。`count` 定义扫描实例总数。

以下配置设置定期扫描，两次扫描之间的延迟为 12 小时：

```
scan:
  scheduling:
    interval: PT12H
    count: 4
  buckets:
    - bucket:
        name: my-bucket-1
        filter:
          include:
```

```
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

Amazon S3 作为目标

要将数据从 OpenSearch 摄取管道写入 S3 存储桶，请使用预配置的 S3 蓝图创建带有 S3 接收器的管道。该管道将选择性数据路由到 OpenSearch 接收器，同时将所有数据发送到 S3 中进行存档。有关更多信息，请参阅 the section called “使用蓝图”。

创建 S3 接收器时，您可以从各种不同的接收器编解码器指定首选格式。例如，如果要以列式格式写入数据，请选择 Parquet 或 Avro 编解码器。如果您更喜欢基于行的格式，请选择 JSON 或 NDJSON。要将数据写入指定架构中的 S3，您还可以使用 Avro 格式在接收器编解码器中定义内联架构。

以下示例在 S3 接收器中定义内联架构：

```
- 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"}
          ]
        }
```

定义此架构时，请指定管道向接收器发送的不同类型事件中可能存在的所有键的超集。

例如，如果事件可能缺少键，则在架构中添加值为 `null` 的键。Null 值声明允许架构处理非统一数据（一些事件具有这些键，另一些事件则没有）。当传入事件确实存在这些键时，则将键值写入接收器。

此架构定义充当筛选器，仅允许将定义的键发送到接收器，并从传入事件中删除未定义的键。

您也可以在接收器中使用 `include_keys` 和 `exclude_keys` 筛选路由到其他接收器的数据。两个筛选器互斥，因此在架构中一次只能使用一个筛选器。此外，不能在用户定义的架构中使用它们。

要使用上述筛选条件创建管道，请使用预先配置的接收器筛选条件蓝图。有关更多信息，请参阅 the section called “使用蓝图”。

将 Amazon S3 跨账户作为源

您可以通过 Amazon S3 授予跨账户访问权限，这样 OpenSearch Ingestion 管道就可以访问另一个账户中的 S3 存储桶作为来源。要启用跨账户访问，请参阅《Amazon S3 用户指南》中的 Bucket owner granting cross-account bucket permissions。授予访问权限后，请确保您的管道角色具有所需的权限。

然后，您可以使用创建管道，`bucket_owners`以启用跨账户访问作为来源的 Amazon S3 存储桶：

```
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"
```

在 Amaz OpenSearch on Security Lake 中使用采集管道

您可以使用 S3 源插件将来自 A mazon Security Lake 的数据提取到您的 OpenSearch 摄取管道中。Security Lake 会自动将来自 Amazon 环境、本地环境和 SaaS 提供商的安全数据集中到专门构建的数据湖中。您可以创建订阅，将数据从 Security Lake 复制到您的 OpenSearch 摄取管道，然后由该渠道将其写入您的 OpenSearch 服务域或 OpenSearch 无服务器集合。

要将管道配置为从 Security Lake 读取，请使用预先配置的 Security Lake 蓝图。蓝图包括用于从 Security Lake 摄取开放式网络安全架构框架 (OCSF) parquet 文件的默认配置。有关更多信息，请参阅 the section called “使用蓝图”。

主题

- 使用以 Amaz OpenSearch on Security Lake 为来源的摄取管道
- 使用以 Amaz OpenSearch on Security Lake 为接收器的摄取管道

使用以 Amaz OpenSearch on Security Lake 为来源的摄取管道

您可以使用 OpenSearch Ingestion 管道中的 Amazon S3 源插件从 Amazon Security Lake 提取数据。Security Lake 会自动将来自 Amazon 环境、本地系统和 SaaS 提供商的安全数据集中到专门构建的数据湖中。

Amazon Security Lake 在管道中具有以下元数据属性：

- `bucket_name`：安全湖为存储安全数据而创建的 Amazon S3 存储桶的名称。
- `path_prefix`：在 Security Lake IAM 角色策略中定义的自定义源名称。
- `region`：Secur Amazon Web Services 区域 ity Lake S3 存储桶所在的位置。
- `accountID`：启用安全湖的 Amazon Web Services 账户 ID。
- `sts_role_arn`：打算与 Security Lake 一起使用的 IAM 角色的 ARN。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

- 启用 Security Lake。
- 在 Security Lake 中创建订阅用户。
  - 选择要摄取到管道的源。

- 对于订阅者凭证，请添加您打算在 Amazon Web Services 账户 哪里创建管道的 ID。对于外部 ID，请指定 `OpenSearchIngestion-{accountid}`。
- 对于数据访问方法，请选择 S3。
- 有关通知详细信息，请选择 SQS 队列。

创建订阅用户时，Security Lake 将自动创建两个内联权限策略，一个用于 S3，一个用于 SQS。策略采用以下格式：`AmazonSecurityLake-{12345}-S3` 和 `AmazonSecurityLake-{12345}-SQS`。要允许管道访问订阅用户源，必须将必需权限与管道角色进行关联。

配置管道角色

在 IAM 中创建新的权限策略，仅组合 Security Lake 自动创建的两个策略中的必需权限。以下示例策略显示了 OpenSearch 摄取管道从多个 Security Lake 来源读取数据所需的最低权限：

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3:::aws-security-data-lake-region-abcde/aws/
LAMBDA_EXECUTION/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-region-abcde/aws/S3_DATA/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-region-abcde/aws/VPC_FLOW/1.0/
*",
            "arn:aws:s3:::aws-security-data-lake-region-abcde/aws/ROUTE53/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-region-abcde/aws/
SH_FINDINGS/1.0/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sqs:ReceiveMessage",
            "sqs:DeleteMessage"
```

```
            ],
            "Resource":[
               "arn:aws:sqs:region:account-id:AmazonSecurityLake-abcde-Main-Queue"
            ]
         }
      ]
   }
```

> **Important**
>
> Security Lake 不负责为您管理管道角色策略。如果向 Security Lake 订阅中添加源或从中删除源，则必须手动更新政策。Security Lake 将为每个日志源创建分区，因此您需要在管道角色中手动添加或删除权限。

必须将以下权限附加到在 S3 源插件配置的 `sts_role_arn` 选项的 `sqs` 下指定的 IAM 角色。

```
version: "2"
source:
  s3:
    ...
    sqs:
      queue_url: "https://sqs.us-east-1amazonaws.com/account-id/
AmazonSecurityLake-abcde-Main-Queue"
    aws:
      ...
processor:
  ...
sink:
  - opensearch:
      ...
```

创建管道

向管道角色添加权限后，使用预先配置的 Security Lake 蓝图创建管道。有关更多信息，请参阅 the section called “使用蓝图”。

必须在 `s3` 源配置中指定 `queue_url` 选项，即要读取的 Amazon SQS 队列 URL。要设置 URL 格式，请在订阅用户配置中找到订阅端点，将 `arn:aws:` 更改为 `https://`。例如 `https://sqs.us-east-1amazonaws.com/account-id/AmazonSecurityLake-abdcef-Main-Queue`。

在 S3 源配置中指定的 `sts_role_arn` 必须是管道角色 ARN。

使用以 Amaz OpenSearch on Security Lake 为接收器的摄取管道

使用 OpenSearch Ingestion 中的 Amazon S3 接收器插件将数据从任何支持的来源发送到亚马逊安全湖。Security Lake 在专用数据湖中收集和存储来自 Amazon本地环境和 SaaS 提供商的安全数据。

要将您的管道配置为将日志数据写入 Security Lake，请使用预配置的防火墙流量日志蓝图。该蓝图包括一个默认配置，用于检索存储在 Amazon S3 存储桶中的原始安全日志或其他数据、处理记录并对其进行标准化。然后，它将数据映射到开放网络安全架构框架 (OCSF)，并将转换后的符合 OCSF 的数据发送到 Security Lake。

该管道具有以下元数据属性：

- `bucket_name`：安全湖为存储安全数据而创建的 Amazon S3 存储桶的名称。
- `path_prefix`：在 Security Lake IAM 角色策略中定义的自定义源名称。
- `region`：Secur Amazon Web Services 区域 ity Lake S3 存储桶所在的位置。
- `accountID`：启用安全湖的 Amazon Web Services 账户 ID。
- `sts_role_arn`：打算与 Security Lake 一起使用的 IAM 角色的 ARN。

先决条件

在创建用于向 Security Lake 发送数据的管道之前，请执行以下步骤：

- 启用和配置 Amazon Security Lake：设置 Amazon Security Lake 以集中来自各种来源的安全数据。有关说明，请参阅使用控制台启用安全湖。

  选择来源时，选择 “收录特定 Amazon 来源”，然后选择一个或多个要采集的日志和事件源。
- 设置权限：为管道角色配置将数据写入 Security Lake 所需的权限。有关更多信息，请参阅管道角色。

创建管道

使用预配置的 Security Lake 蓝图创建管道。有关更多信息，请参阅使用蓝图创建管道。

将 OpenSearch 采集管道与 Fluent Bit 配合使用

此示例 Fluent Bit 配置文件将日志数据从 Fluent Bit 发送到 OpenSearch 摄取管道。有关摄取日志数据的更多信息，请参阅 Data Prepper 文档中的日志分析。

请注意以下几点：

- host 值必须是管道端点。例如 *pipeline-endpoint*.*us-east-1*osis.amazonaws.com。
- aws_service 值必须为 osis。
- 该aws_role_arn值是 Amazon IAM 角色的 ARN，供客户端代入并用于签名版本 4 身份验证。

```
[INPUT]
  name                  tail
  refresh_interval      5
  path                  /var/log/test.log
  read_from_head        true

[OUTPUT]
  Name http
  Match *
  Host pipeline-endpoint.us-east-1osis.amazonaws.com
  Port 443
  URI /log/ingest
  Format json
  aws_auth true
  aws_region region
  aws_service osis
  aws_role_arn arn:aws:iam::account-id:role/ingestion-role
  Log_Level trace
  tls On
```

然后，您可以配置如下所示的 OpenSearch 采集管道，该管道以 HTTP 为源：

```
version: "2"
unaggregated-log-pipeline:
  source:
    http:
      path: "/log/ingest"
  processor:
    - grok:
        match:
          log:
            - "%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:network_node}
 %{NOTSPACE:network_host} %{IPORHOST:source_ip}:%{NUMBER:source_port:int} ->
 %{IPORHOST:destination_ip}:%{NUMBER:destination_port:int} %{GREEDYDATA:details}"
    - grok:
```

```
        match:
          details:
            - "'%{NOTSPACE:http_method} %{NOTSPACE:http_uri}' %{NOTSPACE:protocol}"
            - "TLS%{NOTSPACE:tls_version} %{GREEDYDATA:encryption}"
            - "%{NUMBER:status_code:int} %{NUMBER:response_size:int}"
    - delete_entries:
        with_keys: ["details", "log"]

  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
        index: "index_name"
        index_type: custom
        bulk_size: 20
        aws:
          region: "region"
```

将 OpenSearch 采集管道与 Fluentd 配合使用

Fluentd是一个开源数据收集生态系统，它提供了 SDKs 不同的语言和子项目，例如Fluent Bit。此示例 Fluentd 配置文件将日志数据从 Fluentd 发送到摄取管道。 OpenSearch 有关摄取日志数据的更多信息，请参阅 Data Prepper 文档中的日志分析。

请注意以下几点：

- `endpoint` 值必须是管道端点。例如 *pipeline-endpoint*.*us-east-1*osis.amazonaws.com/apache-log-pipeline/logs。
- `aws_service` 值必须为 `osis`。
- 该`aws_role_arn`值是 Amazon IAM 角色的 ARN，供客户端代入并用于签名版本 4 身份验证。

```
<source>
  @type tail
  path logs/sample.log
  path_key log
  tag apache
  <parse>
    @type none
  </parse>
</source>

<filter apache>
```

```
  @type record_transformer
  <record>
    log ${record["message"]}
  </record>
</filter>

<filter apache>
  @type record_transformer
  remove_keys message
</filter>

<match apache>
  @type http
  endpoint pipeline-endpoint.us-east-1osis.amazonaws.com/apache-log-pipeline/logs
  json_array true

  <auth>
    method aws_sigv4
    aws_service osis
    aws_region region
    aws_role_arn arn:aws:iam::account-id:role/ingestion-role
  </auth>

  <format>
    @type json
  </format>

  <buffer>
    flush_interval 1s
  </buffer>
</match>
```

然后，您可以配置如下所示的 OpenSearch 采集管道，该管道以 HTTP 为源：

```
version: "2"
apache-log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log:
```

```
            - "%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:network_node}
 %{NOTSPACE:network_host} %{IPORHOST:source_ip}:%{NUMBER:source_port:int} ->
 %{IPORHOST:destination_ip}:%{NUMBER:destination_port:int} %{GREEDYDATA:details}"
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1es.amazonaws.com"]
        index: "index_name"
        aws_region: "region"
        aws_sigv4: true
```

将采集管道与 C OpenSearch ollector 配合使用 OpenTelemetry

此示例OpenTelemetry 配置文件从 OpenTelemetry收集器导出跟踪数据并将其发送到 OpenSearch 摄取管道。有关摄取跟踪数据的更多信息，请参阅 Data Prepper 文档中的跟踪分析。

请注意以下几点：

- `endpoint` 值必须包含管道端点。例如 `https://pipeline-endpoint.us-east-1osis.amazonaws.com`。
- `service` 值必须为 `osis`。
- OTLP/HTTP 导出器的`compression`选项必须与管道 OpenTelemetry 源上的`compression`选项相匹配。

```
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]
```

然后，您可以配置如下所示的 OpenSearch Ingestion 管道，该管道将OTel 跟踪插件指定为来源：

```
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"
```

有关其他示例管道，请参阅预先配置的跟踪分析蓝图。有关更多信息，请参阅 the section called “使用蓝图”。

在 Kafka 中使用 OpenSearch 采集管道

您可以使用 Kafka 插件将数据从自行管理的 Kafka 集群流式传输到亚马逊 OpenSearch 服务域和 OpenSearch 无服务器集合。 OpenSearch Ingestion 支持来自配置有公共或私有 (VPC) 网络的 Kafka 集群的连接。本主题概述了设置采集管道的先决条件和步骤，包括配置网络设置和身份验证方法，例如双向 TLS (mTLS)、SASL/SCRAM 或 IAM。

从公共 Kafka 集群迁移数据

您可以使用 OpenSearch Ingestion 管道从公共自管 Kafka 集群迁移数据，这意味着可以公开解析域 DNS 名称。为此，请设置一个以自管理的 Kafka 为源，将 OpenSearch 服务或 OpenSearch 无服务器作为目标的 OpenSearch 摄取管道。这将处理您从自行管理的源集群到托 Amazon管目标域或集合的流式数据。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 使用公有网络配置创建自主管理型 Kafka 集群。集群应包含您要采集到 S OpenSearch ervice 中的数据。
2. 创建要将数据迁移到的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅the section called “ 创建 OpenSearch 服务域”和the section called “创建集合”。
3. 使用在您的自管理集群上设置身份验证。 Amazon Secrets Manager按照 Rotate Amazon Secrets Manager secrets 中的步骤启用密钥轮换。
4. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的自管理集群写入您的域或集合。

   以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

   JSON

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

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

要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

设置 Kafka 管道先决条件后，配置要在工作流配置中使用的管道角色，添加写入 OpenSearch 服务域或 OpenSearch 无服务器集合的权限，以及从 Secrets Manager 读取密钥的权限。

步骤 2：创建管道

然后，您可以配置如下所示的 OpenSearch 采集管道，将 Kafka 指定为来源。

您可以将多个 OpenSearch 服务域指定为数据的目的地。此功能允许有条件地路由或将传入数据复制到多个 OpenSearch 服务域。

您还可以将数据从源 Confluent Kafka 集群迁移到无服务器 V OpenSearch PC 集合。务必要在管道配置中提供网络访问策略。您可以使用 Confluent 架构注册表来定义 Confluent 架构。

```
version: "2"
kafka-pipeline:
  source:
    kafka:
      encryption:
        type: "ssl"
      topics:
```

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

您可以使用预先配置的蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

从 VPC 中的 Kafka 集群迁移数据

您还可以使用 OpenSearch 摄取管道从 VPC 中运行的自管 Kafka 集群迁移数据。为此，请设置一个以自管理的 Kafka 为源，将 OpenSearch 服务或 OpenSearch 无服务器作为目标的 OpenSearch 摄取管道。这将处理您从自行管理的源集群到托 Amazon管目标域或集合的流式数据。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 使用 VPC 网络配置创建自管理 Kafka 集群，其中包含您要提取到服务中的数据。 OpenSearch
2. 创建要将数据迁移到的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅创建 OpenSearch 服务域和创建集合。
3. 使用在您的自管理集群上设置身份验证。 Amazon Secrets Manager按照 Rotate Amazon Secrets Manager secrets 中的步骤启用密钥轮换。
4. 获取有权访问自主管理型 Kafka 的 VPC 的 ID。选择 OpenSearch Ingestion 要使用的 VPC CIDR。

> **Note**
>
> 如果您使用创建管道，则还必须将您的 OpenSearch 摄取管道连接到您的 VPC 才能使用自我管理的 Kafka。 Amazon Web Services Management Console 要执行此操作，请找到网络配置部分，选择连接到 VPC 复选框，然后从提供的任意一个默认选项中选择 CIDR，或者选择自己的 CIDR。您可以按照 RFC 1918 当前最佳实践中的定义，使用私有地址空间中的任何 CIDR。
> 要提供自定义 CIDR，请从下拉菜单中选择其他。为避免接入和自行管理之间的 IP 地址冲突 OpenSearch，请确保自管理 VP OpenSearch C CIDR 与用于 OpenSearch 摄取的 CIDR 不同。 OpenSearch

5. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的自管理集群写入您的域或集合。

   以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

   JSON

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

```
        ]
      }
    ]
  }
```

要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

完成管道先决条件设置后，配置管道角色以在管道配置中使用，并为该角色添加以下权限：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": ["arn:aws:secretsmanager:region:account-id:secret:secret-
name"]
        },
        {
            "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/*"
            ]
```

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

您必须为用于创建 OpenSearch 摄取管道的 IAM 角色提供上述 Amazon EC2 权限，因为管道使用这些权限在您的 VPC 中创建和删除网络接口。该管道只能通过此网络接口访问 Kafka 集群。

步骤 2：创建管道

然后，您可以配置如下所示的 OpenSearch 采集管道，将 Kafka 指定为来源。

您可以将多个 OpenSearch 服务域指定为数据的目的地。此功能允许有条件地路由或将传入数据复制到多个 OpenSearch 服务域。

您还可以将数据从源 Confluent Kafka 集群迁移到无服务器 V OpenSearch PC 集合。务必要在管道配置中提供网络访问策略。您可以使用 Confluent 架构注册表来定义 Confluent 架构。

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

您可以使用预先配置的蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

使用 Amazon OpenSearch Ingestion 从自行管理的 OpenSearch 集群迁移数据

您可以使用带有自我管理 OpenSearch 或 Elastic OpenSearch search 的 Amazon Ingestion 管道将数据迁移到亚马逊 OpenSearch 服务域和无服务器集合。 OpenSearch OpenSearch Ingestion 支持公用和私有网络配置，用于迁移来自自管理 OpenSearch 和 Elasticsearch 的数据。

从公共 OpenSearch集群迁移

您可以使用 OpenSearch Ingestion 管道从具有公共配置的自管理集群 OpenSearch 或 Elasticsearch 集群迁移数据，这意味着可以公开解析域 DNS 名称。为此，请设置一个以自我管理 OpenSearch 或 Elast OpenSearch icsearch 为源，以 OpenSearch 服务或 OpenSearch 无服务器为目标的采集管道。这可以有效地将您的数据从自我管理的源集群迁移到托管的 Amazon目标域或集合。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 创建包含您要迁移的数据的自管理集群 OpenSearch 或 Elastisearch 集群，并配置公有 DNS 名称。有关说明，请参阅 OpenSearch 文档中的创建集群。
2. 创建要将数据迁移到的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅the section called “ 创建 OpenSearch 服务域”和the section called “创建集合”。
3. 使用在您的自管理集群上设置身份验证。 Amazon Secrets Manager按照 Rotate Amazon Secrets Manager secrets 中的步骤启用密钥轮换。
4. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的自管理集群写入您的域或集合。

   以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::pipeline-account-id:role/pipeline-role"
      },
```

```
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:region:account-id:domain/domain-name"
      ]
    }
  ]
}
```

要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

设置好 OpenSearch 管道先决条件后，配置要在管道配置中使用的管道角色，添加写入 OpenSearch 服务域或 OpenSearch 无服务器集合的权限，以及从 Secrets Manager 读取密钥的权限。

步骤 2：创建管道

然后，您可以配置如下所示的 OpenSearch Ingestion 管道，该管道指定 OpenSearch 为来源。

您可以将多个 OpenSearch 服务域指定为数据的目的地。此功能允许有条件地路由或将传入数据复制到多个 OpenSearch 服务域。

您还可以将数据从源集群 OpenSearch 或 Elasticsearch 集群迁移到 OpenSearch 无服务器 VPC 集合。务必要在管道配置中提供网络访问策略。

```
version: "2"
opensearch-migration-pipeline:
  source:
    opensearch:
      acknowledgments: true
      host: [ "https://my-self-managed-cluster-name:9200" ]
      indices:
        include:
          - index_name_regex: "include-.*"
        exclude:
          - index_name_regex: '\..*'
      authentication:
        username: ${aws_secrets:secret:username}
```

```
        password: ${aws_secrets:secret:password}
        scheduling:
           interval: "PT2H"
           index_read_count: 3
           start_time: "2023-06-02T22:01:30.00Z"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
          #Uncomment the following lines if your destination is an OpenSearch
 Serverless collection
          #serverless: true
          # serverless_options:
          #     network_policy_name: "network-policy-name"
      index: "${getMetadata(\"opensearch-index\")}"
      document_id: "${getMetadata(\"opensearch-document_id\")}"
      enable_request_compression: true
      dlq:
        s3:
          bucket: "bucket-name"
          key_path_prefix: "apache-log-pipeline/logs/dlq"
          region: "us-east-1"
extension:
  aws:
    secrets:
      secret:
        secret_id: "my-opensearch-secret"
        region: "us-east-1"
        refresh_interval: PT1H
```

您可以使用预先配置的蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

从 VPC 中的 OpenSearch集群迁移数据

您还可以使用 OpenSearch 摄取管道从 VPC 中运行的自管理集群 OpenSearch 或 Elasticsearch 集群迁移数据。为此，请设置一个以自我管理 OpenSearch 或 Elast OpenSearch icsearch 为源，以 OpenSearch 服务或 OpenSearch 无服务器为目标的采集管道。这可以有效地将您的数据从自我管理的源集群迁移到托管的 Amazon目标域或集合。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 使用包含要迁移的数据的 VPC 网络配置创建自管理集群 OpenSearch 或 Elastisearch 集群。
2. 创建要将数据迁移到的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅创建 OpenSearch 服务域和创建集合。
3. 使用在您的自管理集群上设置身份验证。 Amazon Secrets Manager按照 Rotate Amazon Secrets Manager secrets 中的步骤启用密钥轮换。
4. 获取有权访问自管 OpenSearch 或弹性搜索的 VPC 的 ID。选择 OpenSearch Ingestion 要使用的 VPC CIDR。

> **Note**
>
> 如果您使用创建管道，则还必须将您的 OpenSearch 摄取管道连接到您的 VPC 才能使用自我管理 OpenSearch 或 Elasticsearch。 Amazon Web Services Management Console 为此，请找到 “源网络选项” 部分，选中 “连接到 VPC” 复选框，然后从提供的默认选项中选择您的 CIDR。您可以按照 RFC 1918 当前最佳实践中的定义，使用私有地址空间中的任何 CIDR。
> 要提供自定义 CIDR，请从下拉菜单中选择其他。为避免接入和自行管理之间的 IP 地址冲突 OpenSearch，请确保自管理 VP OpenSearch C CIDR 与用于 OpenSearch 摄取的 CIDR 不同。 OpenSearch

5. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的自管理集群写入您的域或集合。

   以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

   JSON

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

```
        "Resource": [
          "arn:aws:es:region:account-id:domain/domain-name"
        ]
      }
    ]
  }
```

要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

完成管道先决条件设置后，配置管道角色以在管道配置中使用，并为该角色添加以下权限：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": ["arn:aws:secretsmanager:region:account-id:secret:secret-
name"]
        },
        {
            "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/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals":
                    {
                        "aws:RequestTag/OSISManaged": "true"
                    }
            }
        }
    ]
}
```

您必须为用于创建 OpenSearch 摄取管道的 IAM 角色提供上述 Amazon EC2 权限，因为管道使用这些权限在您的 VPC 中创建和删除网络接口。管道只能通过此网络接口访问 OpenSearch 集群。

步骤 2：创建管道

然后，您可以配置如下所示的 OpenSearch Ingestion 管道，该管道指定 OpenSearch 为来源。

您可以将多个 OpenSearch 服务域指定为数据的目的地。此功能允许有条件地路由或将传入数据复制到多个 OpenSearch 服务域。

您还可以将数据从源集群 OpenSearch 或 Elasticsearch 集群迁移到 OpenSearch 无服务器 VPC 集合。务必要在管道配置中提供网络访问策略。

```
version: "2"
opensearch-migration-pipeline:
  source:
    opensearch:
      acknowledgments: true
      host: [ "https://my-self-managed-cluster-name:9200" ]
      indices:
        include:
          - index_name_regex: "include-.*"
        exclude:
          - index_name_regex: '\..*'
      authentication:
        username: ${aws_secrets:secret:username}
        password: ${aws_secrets:secret:password}
        scheduling:
           interval: "PT2H"
           index_read_count: 3
           start_time: "2023-06-02T22:01:30.00Z"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      aws:
          region: "us-east-1"
          #Uncomment the following lines if your destination is an OpenSearch
 Serverless collection
          #serverless: true
          # serverless_options:
          #     network_policy_name: "network-policy-name"
      index: "${getMetadata(\"opensearch-index\")}"
      document_id: "${getMetadata(\"opensearch-document_id\")}"
      enable_request_compression: true
      dlq:
        s3:
          bucket: "bucket-name"
          key_path_prefix: "apache-log-pipeline/logs/dlq"
          region: "us-east-1"
extension:
  aws:
    secrets:
      secret:
        secret_id: "my-opensearch-secret"
        region: "us-east-1"
        refresh_interval: PT1H
```

您可以使用预先配置的蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。

将 OpenSearch 摄取管道与 Amazon Kinesis Data Streams 配合使用

您可以使用 Kinesis 插件将数据从 Amazon Kinesis Data Streams 流传输到亚马逊 OpenSearch 服务域和无服务器集合。 OpenSearch 该管道从 Amazon Kinesis 提取记录并将其发送 OpenSearch到，并根据直播名称和当前日期自动生成索引。

连接亚马逊 Kinesis Data Streams

您可以使用 OpenSearch Ingestion 管道通过公共配置从 Amazon Kinesis Data Streams 迁移数据，这意味着可以公开解析域 DNS 名称。为此，请设置一个以 Amazon K OpenSearch inesis Data Streams 为源，以 OpenSearch 服务或 OpenSearch 无服务器为目标的摄取管道。这将处理您从自行管理的源集群到托 Amazon管目标域或集合的流式数据。

先决条件

在创建 OpenSearch Ingestion 管道之前，请执行以下步骤：

1. 创建充当源的 Amazon Kinesis 数据流。该流应包含您要采集到 S OpenSearch ervice 中的数据。
2. 创建要将数据迁移到的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅创建 OpenSearch 服务域和创建集合。
3. 使用对您的 Amazon Kinesis 数据流设置身份验证。 Amazon Secrets Manager按照 Rotate Amazon Secrets Manager secrets 中的步骤启用密钥轮换。
4. 将基于资源的策略附加到域，或将数据访问策略附加到集合。这些访问策略允许 OpenSearch Ingestion 将数据从您的自管理集群写入您的域或集合。

   以下示例域访问策略允许您在下一步中创建的管道角色将数据写入域。确保使用自身 ARN 更新 `resource`。

   JSON

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

要创建具有正确权限的 IAM 角色来访问向集合或域写入数据，请参阅the section called “设置角色和用户”。

步骤 1：配置管道角色

设置 Amazon Kinesis Data Streams 管道先决条件后，配置要在管道配置中使用的管道角色，添加写入服务域 OpenSearch 或无服务器集合的权限，以及从 Secrets Manager 读取密钥的权限。OpenSearch

写入 Amazon S3 存储桶、域和集合需要以下权限：

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:region:account-id:stream/stream-name"
            ]
        }
    ]
}
```

如果为直播启用了服务器端加密，则以下 KMS 策略将解密直播记录：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowDecryptionOfCustomManagedKey",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:region:account-id:key/key-id"
        }
    ]
}
```

为使管道能够将数据写入域，域必须具有域级访问策略，以允许 sts_role_arn 管道角色访问域。以下示例域访问策略允许您在上一步中创建的名为`pipeline-role`的管道角色向名为的域写入数据`ingestion-domain`：

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

```
    }
  ]
}
```

步骤 2：创建管道

然后，您可以配置一个将 Amazon K OpenSearch inesis 指定为来源的摄取管道。可用的元数据属性有：

- `stream_name`: 正在从中提取记录的 Kinesis 数据流的名称。
- `partition_key`: 正在提取的 Kinesis 数据流记录的分区键。
- `sequence_number`：正在提取的 Kinesis 数据流记录的序列号。
- `sub_sequence_number`：正在提取的 Kinesis 数据流记录的子序列号。

您可以将多个 OpenSearch 服务域指定为数据的目的地。此功能允许有条件地路由或将传入数据复制到多个 OpenSearch 服务域。

您也可以将数据从 Amazon Kinesis 迁移到 OpenSearch 无服务器 VPC 集合。 OpenSearch Ingestion 控制台上有一个用于创建管道的蓝图。要创建管道，您可以使用以下`Amazon-KinesisDataStreamsPipeline`蓝图。

```
version: "2"
     kinesis_data_streams_pipeline:
       source:
         kinesis_data_streams:
           acknowledgments: true
           codec:
             newline:
           streams:
             - stream_name: "<stream name>"
             - stream_name: "<stream name>"
           aws:
             region: "region"
       sink:
         - opensearch:
             hosts: [ "https://search-mydomain.region.es.amazonaws.com" ]
             index: "index_${getMetadata(\"stream-name\")}"
             document_id: "${getMetadata(\"partition_key\")}"
             aws:
               sts_role_arn: "<<arn:aws:iam::123456789012:role/Example-Role>>"
```

```
                region: "region"

                s3:
                  bucket: "dlq-bucket-name"
                  region: "region"
```

您可以使用预先配置的蓝图来创建此管道。有关更多信息，请参阅 the section called “使用蓝图”。您还可以查看开源 Opensearch 文档，了解其他配置选项。要了解更多信息，请参阅配置选项。

数据一致性

OpenSearch Ingestion 支持 end-to-end确认，以确保数据的持久性。当管道从 Kinesis 读取流记录时，它会根据与流关联的分片动态分配读取流记录的工作。Pipeline 在收录 OpenSearch 域或集合中的所有记录后收到确认后，将自动对流进行检查点。这将避免对直播记录进行重复处理。

> **Note**
>
> 如果你想根据直播名称创建索引，你可以在 opensearch sink 部分中将索引定义为 “index_${getMetadata (\”*stream_name*\”)}”。

（可选）为 Kinesis Data Streams 管道配置推荐的计算单位 (OCUs)

创建 Kinesis 源管道时，建议至少使用 2 个计算单元 (OCU)。这将允许每个分片处理的 Kinesis 数据流记录在计算单元之间均匀分布，从而确保流记录摄取的低延迟机制。

也可以将 OpenSearchKinesis 数据流源管道配置为从多个流中提取流记录。建议为每个新数据流添加一个额外的计算单元。

> **Note**
>
> 如果您的管道的计算单元 (OCU) 多于管道中配置的流集中的分片数，则某些计算单元可能会处于空闲状态，而不会为每个分片处理任何流记录。

后续步骤

将数据导出到管道后，您可以从配置为管道接收器的 OpenSearch 服务域中查询数据。以下资源可帮助您开始使用：

- 可观察性
- the section called “跟踪分析”
- the section called “管道式处理语言”

将 OpenSearch 摄取管道与 Amazon Lambda

使用Amazon Lambda 处理器使用自定义代码丰富来自 OpenSearch Ingestion 支持的任何源或目标的数据。使用 Lambda 处理器，您可以应用自己的数据转换或扩展，然后将处理过的事件返回到您的管道进行进一步处理。该处理器支持自定义数据处理，并允许您在数据通过管道之前完全控制数据的操作方式。

> **Note**
>
> Lambda 处理器处理的单个事件的有效负载大小限制为 5 MB。此外，Lambda 处理器仅支持 JSON 数组格式的响应。

先决条件

在使用 Lambda 处理器创建管道之前，请创建以下资源：

- 一种 Amazon Lambda 用于丰富和转换源数据的函数。有关说明，请参阅创建您的第一个 Lambda 函数。
- 将成为管道接收器的 OpenSearch 服务域或 OpenSearch 无服务器集合。有关更多信息，请参阅the section called “ 创建 OpenSearch 服务域”和the section called “创建集合”。
- 一种管道角色，包括写入域或集合接收器的权限。有关更多信息，请参阅 the section called “管道角色”。

  管道角色还需要附加权限策略，允许其调用管道配置中指定的 Lambda 函数。例如：

  JSON

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

```
            "Action": [
                "lambda:invokeFunction",
                "lambda:InvokeAsync",
                "lambda:ListFunctions"
            ],
            "Resource": "arn:aws:lambda:region:account-id:function:function-
name"

        }
    ]
}
```

创建管道

要 Amazon Lambda 用作处理器，请配置 OpenSearch Ingestion 管道并指定`aws_lambda`为处理器。您也可以使用Amazon Lambda 自定义扩充蓝图来创建管道。有关更多信息，请参阅 the section called “使用蓝图”。

以下示例管道从 HTTP 源接收数据，使用日期处理器和处理器对其进行丰富，然后将 Amazon Lambda 处理后的数据提取到域中。 OpenSearch

```
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
```

以下示例 Amazon Lambda 函数通过向提供的事件数组中的每个元素添加新的键值对 (`"transformed": "true"`) 来转换传入的数据，然后发回修改后的版本。

```
import json

def lambda_handler(event, context):
    input_array = event.get('events', [])
    output = []
    for input in input_array:
        input["transformed"] = "true";
        output.append(input)

    return output
```

批处理

管道向 Lambda 处理器发送批处理事件，并动态调整批量大小以确保其保持在 5 MB 限制以下。

以下是管道批处理的示例：

```
batch:
    key_name: "events"

input_arrary = event.get('events', [])
```

> **Note**
>
> 创建管道时，请确保 Lambda 处理器配置中的`key_name`选项与 Lambda 处理程序中的事件密钥相匹配。

条件筛选

条件筛选允许您根据事件数据中的特定条件控制 Amazon Lambda 处理器何时调用 Lambda 函数。当你想有选择地处理某些类型的事件而忽略其他类型的事件时，这特别有用。

以下示例配置使用条件筛选：

```
processors:
  - aws_lambda:
      function_name: "my-lambda-function"
      aws:
        region: "region"
      lambda_when: "/sourceIp == 10.10.10.10"
```

使用 Amazon OpenSearch Ingestion 在域名和集合之间迁移数据

您可以使用 OpenSearch Ingestion 管道在亚马逊 OpenSearch 服务域或 OpenSearch 无服务器 VPC 集合之间迁移数据。为此，您需要设置一个管道，将一个域或集合配置为源，将另一个域或集合配置为接收器。这可以有效地将数据从一个域或集合迁移到另一个域或集合。

要迁移数据，您必须拥有以下资源：

- 源 OpenSearch 服务域或 OpenSearch 无服务器 VPC 集合。此域或集合包含要迁移的数据。如果您使用的是域名，则该域名必须运行 OpenSearch 版本 1.0 或更高版本，或者运行 Elasticsearch 版本 7.4 或更高版本。该域还必须具有向您的管道角色授予相应权限的访问策略。
- 您要向其迁移数据的单独域或 VPC 集合。该域或集合将充当管道接收器。
- 一个管道角色， OpenSearch Ingestion 将使用它来读取和写入你的收藏或域名。请将此角色的 Amazon 资源名称（ARN）包含在管道配置中。有关更多信息，请参阅以下资源：
  - the section called “向管道授予访问域的权限”
  - the section called “向管道授予访问集合的权限”

主题

- 限制
- OpenSearch 服务即来源
- 指定多个 OpenSearch 服务域接收器
- 将数据迁移到 OpenSearch 无服务器 VPC 集合

限制

将 OpenSearch 服务域或 OpenSearch 无服务器集合指定为接收器时，以下限制适用：

- 一个管道不能写入多个 VPC 域。

- 您只能将数据迁移到使用 VPC 访问权限的 OpenSearch 无服务器集合或从中迁移数据。不支持公有集合。
- 您不能在单个管道配置中指定 VPC 和公有域的组合。
- 在单个管道配置中，您最多可以指定 20 个非管道接收器。
- 在单个管道配置 Amazon Web Services 区域 中，最多可以指定三个不同的接收器。
- 如果任何接收器的停机时间过长或未预置足够的容量来接收传入的数据，则具有多个接收器的管道的处理速度可能会逐渐下降。

OpenSearch 服务即来源

您指定为源的域或集合是要迁移的数据的来源。

在 IAM 中创建管道角色

要创建 In OpenSearch gestion 管道，必须先创建一个管道角色来授予域名或集合之间的读写权限。为此，请执行以下步骤：

1. 在 IAM 中创建一个新权限策略以附加到该管道角色。确保允许从源读取数据并将数据写入接收器的权限。有关为 OpenSearch 服务域设置 IAM 管道权限的更多信息，请参阅the section called “向管道授予访问域的权限”和the section called “向管道授予访问集合的权限”。
2. 在管道角色中指定以下权限，以从源中读取：

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"es:ESHttpGet",
         "Resource":[
            "arn:aws:es:region:account-id:domain/domain-name/",
            "arn:aws:es:region:account-id:domain/domain-name/_cat/indices",
            "arn:aws:es:region:account-id:domain/domain-name/_search",
            "arn:aws:es:region:account-id:domain/domain-name/_search/scroll",
            "arn:aws:es:region:account-id:domain/domain-name/*/_search"
         ]
      },
      {
         "Effect":"Allow",
```

```
            "Action":"es:ESHttpPost",
            "Resource":[
               "arn:aws:es:region:account-id:domain/domain-name/*/_search/
point_in_time",
               "arn:aws:es:region:account-id:domain/domain-name/*/_search/scroll"
            ]
         },
         {
            "Effect":"Allow",
            "Action":"es:ESHttpDelete",
            "Resource":[
               "arn:aws:es:region:account-id:domain/domain-name/_search/
point_in_time",
               "arn:aws:es:region:account-id:domain/domain-name/_search/scroll"
            ]
         }
      ]
}
```

创建管道

将策略附加到管道角色后，使用AWSOpenSearchDataMigrationPipeline迁移蓝图创建管道。该蓝图包括用于在 OpenSearch 服务域或集合之间迁移数据的默认配置。有关更多信息，请参阅 the section called “使用蓝图”。

> **Note**
>
> OpenSearch Ingestion 使用您的源域版本和发行版来确定使用哪种机制进行迁移。某些版本支持该`point_in_time`选项。 OpenSearch Serverless 之所以使用该`search_after`选项，是因为它不支持`point_in_time`或`scroll`。

迁移过程中可能正在创建新索引，也可能在迁移过程中更新文档。因此，您可能需要对域索引数据执行一次或多次扫描，以获取新的或更新的数据。

通过在管道配置中配置 `index_read_count` 和 `interval`，指定要运行的扫描次数。以下示例说明如何执行多次扫描：

```
scheduling:
    interval: "PT2H"
```

```
    index_read_count: 3
    start_time: "2023-06-02T22:01:30.00Z"
```

OpenSearch Ingestion 使用以下配置来确保您的数据写入同一个索引并保持相同的文档 ID：

```
index: "${getMetadata(\"opensearch-index\")}"
document_id: "${getMetadata(\"opensearch-document_id\")}"
```

指定多个 OpenSearch 服务域接收器

您可以指定多个公共 OpenSearch 服务域作为数据的目的地。您可以使用此功能执行条件路由或将传入的数据复制到多个 OpenSearch 服务域。您最多可以指定 10 个不同的公共 OpenSearch 服务域作为接收器。

在以下示例中，传入的数据有条件地路由到不同的 OpenSearch 服务域：

```
...
  route:
    - 2xx_status: "/response >= 200 and /response < 300"
    - 5xx_status: "/response >= 500 and /response < 600"
  sink:
    - opensearch:
        hosts: [ "https://search-response-2xx.region.es.amazonaws.com" ]
        aws:
          region: "us-east-1"
        index: "response-2xx"
        routes:
          - 2xx_status
    - opensearch:
        hosts: [ "https://search-response-5xx.region.es.amazonaws.com" ]
        aws:
          region: "us-east-1"
        index: "response-5xx"
        routes:
          - 5xx_status
```

将数据迁移到 OpenSearch 无服务器 VPC 集合

您可以使用 OpenSearch Ingestion 将数据从源 OpenSearch 服务域或 OpenSearch 无服务器集合迁移到 VPC 集合接收器。必须在管道配置中提供网络访问策略。有关将数据提取到 OpenSearch 无服务器 VPC 集合的更多信息，请参阅。the section called “教程：将数据摄取到集合”

将数据迁移到 VPC 集合

1. 创建 OpenSearch 无服务器集合。有关说明，请参阅the section called “教程：将数据摄取到集合”。
2. 为集合创建网络策略，指定对集合端点和控制面板端点的 VPC 访问权限。有关说明，请参阅the section called “网络访问”。
3. 如果您还没有管道角色，请创建一个。有关说明，请参阅the section called “管道角色”。
4. 创建管道。有关说明，请参阅the section called “使用蓝图”。

使用与 Amazon OpenSearch Ingestion 互动 Amazon SDKs

本节包含一个关于如何使用 Amazon Ingestion 与 Amazon OpenSearch Ingestion 进行交互的示例。Amazon SDKs 该代码示例演示了如何创建域和管道，以及如何将数据摄取到管道中。

主题

- Python

Python

以下示例脚本使用 适用于 Python (Boto3) 的 Amazon SDK 创建 IAM 管道角色、用于向其写入数据的域以及用于摄取数据的管道。然后，它使用 `requests` HTTP 库将示例日志文件提取到管道中。

要安装所需依赖项，请运行以下命令：

```
pip install boto3
pip install botocore
pip install requests
pip install requests-auth-aws-sigv4
```

在脚本中，将的所有实例替换为您的 Amazon Web Services 账户 ID。*account-id*

```
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 Ingestion 中的安全

云安全 Amazon 是重中之重。作为 Amazon 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 Amazon 的责任。责任共担模式将其描述为云的 安全性和云中 的安全性：

- 云安全 — Amazon 负责保护在 Amazon 云中运行 Amazon 服务的基础架构。 Amazon 还为您提供可以安全使用的服务。作为 Amazon 合规性计划的一部分，第三方审核人员将定期测试和验证安全性的有效性。
- 云端安全-您的责任由您使用的 Amazon 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 OpenSearch Ingestion 时如何应用分担责任模型。以下主题向您介绍如何配置 OpenSearch Ingestion 以满足您的安全和合规性目标。您还将学习如何使用其他 Amazon 服务来帮助您监控和保护您的 OpenSearch Ingestion 资源。

主题

- 为 Amazon OpenSearch Ingestion 管道配置 VPC 访问
- 适用于 Amazon OpenSearch Ingestion 的 Identity and Access 管理
- 使用记录 Amazon OpenSearch Ingestion API 调用 Amazon CloudTrail

为 Amazon OpenSearch Ingestion 管道配置 VPC 访问

您可以使用接口 VPC 端点访问 Amazon OpenSearch Ingestion 管道。VPC 是专供您的使用的虚拟网络 Amazon Web Services 账户。它在逻辑上与 Amazon 云中的其他虚拟网络隔绝。通过 VPC 端点来访问管道，有利于在 OpenSearch Ingestion 和 VPC 中的其他服务之间安全地进行通信，而无需 Internet 网关、NAT 设备或 VPN 连接。 Amazon 云中的所有流量都会保持安全。

OpenSearch Ingestion 通过创建由提供支持的接口端点来建立此私有连接。 Amazon PrivateLink在创建管道过程中，我们会在您指定的每个子网中创建一个端点网络接口。这些是请求者托管式网络接口，用作发往 OpenSearch Ingestion 管道的流量的入口点。您还可以选择自行创建和管理接口端点。

使用 VPC，您可以强制在 VPC 边界内通过 OpenSearch Ingestion 管道传输数据，而不是通过公共互联网传输。非 VPC 内部管道通过面向公众的端点和互联网收发数据。

使用 VPC 访问的管道可以写入公有或 VPC Serv OpenSearch ice 域，以及公有或 VPC 访问的公有或 VPC OpenSearch 无服务器集合。

主题

- 注意事项
- 限制
- 先决条件
- 为管道配置 VPC 访问权限
- 自主管理型 VPC 端点
- VPC 访问的服务相关角色

注意事项

为管道配置 VPC 时，请考虑以下事项。

- 管道无需与其接收器位于同一 VPC 中。也不需要在两者之间建立连接 VPCs。 OpenSearch Ingestion 负责为您建立连接。

- 您只能为管道指定一个 VPC。
- 与公有管道不同的是，VPC 管道必须与其写入的域或集合接收器位于同一 Amazon Web Services 区域 中。
- 您可以选择将管道部署到 VPC 的一个、两个或三个子网中。子网分布在部署 Ingestion OpenSearch 计算单位 (OCUs) 的可用区中。
- 如果您只在一个子网中部署管道，则可用区出现故障时，您将无法摄取数据。为确保高可用性，我们建议您使用两个或三个子网配置管道。
- 指定安全组是可选的。如果您未提供安全组， OpenSearch Ingestion 将使用 VPC 中指定的默认安全组。

限制

使用 VPC 访问的管道存在以下限制。

- 创建管道后，将无法更改其网络配置。如果您在 VPC 中启动管道，则后续无法将其更改为公共端点，反之亦然。
- 您可以使用接口 VPC 端点或公有端点启动管道，但不能同时使用 VPC 端点和公有端点启动管道。在创建管道时只能选择其一。
- 在预置使用 VPC 访问的管道后，将不能将其移到其他 VPC 中，也不能更改其子网或安全组设置。
- 如果管道写入使用 VPC 访问的域或集合接收器，则在管道创建后将无法返回，也无法更改接收器（VPC 或公有）。必须删除，然后使用新的接收器重新创建管道。您仍然可以从公有接收器切换到使用 VPC 访问的接收器。
- 您无法提供对 VPC 管道的跨账户摄取访问权限。

先决条件

在预置使用 VPC 访问的管道之前，必须先执行以下操作：

- 创建 VPC

  要创建 VPC，可以使用 Amazon VPC 控制台、 Amazon CLI 或其中一个 Amazon SDKs。有关更多信息，请参阅 Amazon VPC 用户指南 VPCs中的使用。如果您已有 VPC，请跳过此步骤。
- 预留 IP 地址

  OpenSearch Ingestion 在创建管道期间指定的每个子网中安装一个弹性网络接口。每个网络接口都与一个 IP 地址关联。必须为每个子网预留一个用于网络接口的 IP 地址。

为管道配置 VPC 访问权限

您可以在 OpenSearch 服务控制台中或使用为管道启用 VPC 访问 Amazon CLI。

控制台

您可以在管道创建期间配置 VPC 访问权限。在源网络选项下，选择 VPC 访问并配置以下设置：

设置	描述
端点管理	选择是要自己创建 VPC 端点，还是让 OpenSearch Ingestion 为您创建端点。
VPC	选择要使用的虚拟私有云 (VPC) 的 ID。VPC 和管道必须位于同一 Amazon Web Services 区域中。
子网	选择一个或多个子网。 OpenSearch 服务将 VPC 终端节点和弹性网络界面放入子网中。
安全组	选择一个或多个 VPC 安全组，以允许您所需的应用程序在管道公开的端口（80 或 44443）和协议（HTTP 或 HTTPs）上访问 OpenSearch Ingestion 管道。
VPC 挂载选项	如果您的来源需要跨VPC通信，例如Amazon DocumentDB、 OpenSearch自我管理或 Confluent Kafka OpenSearch ，Ingestion 会在您指定的子网中创建弹性网络接口 ENIs ()，以便连接到这些源。 OpenSearch Ingestion ENIs 在每个可用区中使用来到达指定的来源。连接到 VPC 选项将接 OpenSearch 入数据平面 VPC 连接到您指定的 VPC。 为托管 VPC 选择 CIDR 预留以部署网络接口。

CLI

要使用配置 VPC 访问 Amazon CLI，请指定`--vpc-options`参数：

```
aws osis create-pipeline \
  --pipeline-name vpc-pipeline \
  --min-units 4 \
  --max-units 10 \
  --vpc-options
 SecurityGroupIds={sg-12345678,sg-9012345},SubnetIds=subnet-1212234567834asdf \
```

```
  --pipeline-configuration-body "file://pipeline-config.yaml"
```

自主管理型 VPC 端点

创建管道时，您可以使用端点管理功能来创建使用自主管理型端点或服务托管端点的管道。端点管理是可选的，并且会默认选择由 OpenSearch Ingestion 托管的端点。

要在中创建使用自主管理型 VPC 端点的管道 Amazon Web Services Management Console，请参阅使用 OpenSearch 服务控制台创建管道。要在中创建使用自主管理型 VPC 端点的管道 Amazon CLI，您可以在 create- pipeline 命令中使用`--vpc-options`参数：

```
--vpc-options SubnetIds=subnet-abcdef01234567890,VpcEndpointManagement=CUSTOMER
```

您可以在指定端点服务时自行创建到管道的端点。要查找端点服务，请使用 get-pipeline 命令，这将返回类似于以下内容的响应：

```
"vpcEndpointService" : "com.amazonaws.osis.us-east-1.pipeline-
id-1234567890abcdef1234567890",
"vpcEndpoints" : [
  {
    "vpcId" : "vpc-1234567890abcdef0",
    "vpcOptions" : {
      "subnetIds" : [ "subnet-abcdef01234567890", "subnet-021345abcdef6789" ],
      "vpcEndpointManagement" : "CUSTOMER"
    }
  }
```

使用响应`vpcEndpointService`中的创建带有 Amazon Web Services Management Console 或的 VPC 终端节点 Amazon CLI。

如果您使用自主管理型 VPC 端点，则必须在您的 VPC 中启用 DNS 属性 `enableDnsSupport` 和 `enableDnsHostnames`。请注意，如果您的管道使用自主管理型端点，并且您停止并重启了管道，则必须在您的账户中重新创建该 VPC 端点。

VPC 访问的服务相关角色

服务相关角色是一种独特的 IAM 角色类型，它将权限委派给服务，使之能够代表您创建和管理资源。如果您选择服务托管型 VPC 端点， OpenSearch Ingestion 需要名为的服务相关角色才能AWSServiceRoleForAmazonOpenSearchIngestionService访问您的 VPC、创建管道端点以及将网络接口放入您的 VPC 的子网中。

如果您选择自主管理型 VPC 端点， OpenSearch Ingestion 需要一个名为的服务相关角色。AWSServiceRoleForOpensearchIngestionSelfManagedVpce有关这些角色及其权限和删除方法的更多信息，请参阅the section called “管道创建角色”。

OpenSearch Ingestion 将在您创建摄取管道时自动创建角色。要成功完成自动创建，在账户中创建第一个管道的用户必须拥有 `iam:CreateServiceLinkedRole` 操作权限。有关更多信息，请参阅 IAM 用户指南中的服务相关角色权限。创建角色后，您可以在 Amazon Identity and Access Management (IAM) 控制台中查看角色。

适用于 Amazon OpenSearch Ingestion 的 Identity and Access 管理

Amazon Identity and Access Management (IAM) Amazon Web Services 服务 可帮助管理员安全地控制对 Amazon 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 OpenSearch 摄取资源。您可以使用 IAM Amazon Web Services 服务 ，无需支付额外费用。

主题

- 基于身份的摄取策略 OpenSearch
- 针对 OpenSearch 摄取的政策措施
- 用于 OpenSearch 摄取的策略资源
- Amazon OpenSearch Ingestion 的政策条件密钥
- 含摄入的 ABA OpenSearch C
- 在 OpenSearch 摄取中使用临时证书
- 用于 OpenSearch 摄取的服务相关角色
- 基于身份的摄取策略示例 OpenSearch

基于身份的摄取策略 OpenSearch

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

基于身份的摄取策略示例 OpenSearch

要查看 OpenSearch Ingestion 基于身份的策略的示例，请参阅。the section called “基于身份的策略示例”

针对 OpenSearch 摄取的政策措施

支持策略操作：是

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 Amazon API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

OpenSearch Ingestion 中的策略操作在操作前使用以下前缀：

```
osis
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "osis:action1",
      "osis:action2"
         ]
```

您可以使用通配符 (*) 指定多个操作。例如，要指定以单词 `List` 开头的所有操作，包括以下操作：

```
"Action": "osis:List*"
```

要查看 OpenSearch Ingestion 基于身份的策略的示例，请参阅。 无服务器基于身份的策略示例 OpenSearch

用于 OpenSearch 摄取的策略资源

支持策略资源：是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符（*）指示语句应用于所有资源。

```
"Resource": "*"
```

Amazon OpenSearch Ingestion 的政策条件密钥

支持特定于服务的策略条件键：否

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 Amazon 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 Amazon 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM 策略元素：变量和标签。

Amazon 支持全局条件密钥和特定于服务的条件密钥。要查看所有 Amazon 全局条件键，请参阅 IAM 用户指南中的Amazon 全局条件上下文密钥。

要查看 OpenSearch 摄取条件密钥列表，请参阅《服务授权参考》中的 Amazon OpenSearch Ingestion 条件密钥。要了解您可以使用哪些操作和资源使用条件密钥，请参阅 Amazon OpenSearch Ingestion 定义的操作。

含摄入的 ABA OpenSearch C

支持 ABAC（策略中的标签）：是

基于属性的访问控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在中 Amazon，这些属性称为标签。您可以将标签附加到 IAM 实体（用户或角色）和许多 Amazon 资源。标记实体和资源

是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

有关为 OpenSearch Ingestion 资源添加标签的更多信息，请参阅。the section called “标记管道”

在 OpenSearch 摄取中使用临时证书

支持临时凭证：是

当你使用临时证书登录时，有些 Amazon Web Services 服务 不起作用。有关更多信息，包括哪些 Amazon Web Services 服务 适用于临时证书，请参阅 IAM 用户指南中的Amazon Web Services 服务与 IA M 配合使用的信息。

如果您使用除用户名和密码之外的任何方法登录，则 Amazon Web Services Management Console 使用的是临时证书。例如，当您 Amazon 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的从用户切换到 IAM 角色（控制台）。

您可以使用 Amazon CLI 或 Amazon API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 Amazon。 Amazon 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

用于 OpenSearch 摄取的服务相关角色

支持服务相关角色：是

服务相关角色是一种与服务相关联的 Amazon Web Services 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 Amazon Web Services 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

OpenSearch 摄取使用名为的服务相关角色。`AWSServiceRoleForAmazonOpenSearchIngestionService`名为 `AWSServiceRoleForOpensearchIngestionSelfManagedVpce` 的服务相关角色也适用于使用自主管理型 VPC 端点的管道。有关创建和管理 OpenSearch Ingestion 服务相关角色的详细信息，请参阅。the section called “管道创建角色”

基于身份的摄取策略示例 OpenSearch

默认情况下，用户和角色无权创建或修改 OpenSearch Ingestion 资源。他们也无法使用 Amazon Web Services Management Console、 Amazon Command Line Interface (Amazon CLI) 或 Amazon API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 Amazon OpenSearch Ingestion 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》中的 Amazon OpenSearch Ingestion 的操作、资源和条件密钥。 ARNs

主题

- 策略最佳实践
- 在控制台中使用 OpenSearch Ingestion
- 管理 OpenSearch 摄取管道
- 将数据摄取到摄取管道中 OpenSearch

策略最佳实践

基于身份的策略非常强大。它们决定是否有人可以在您的账户中创建、访问或删除 OpenSearch 摄取资源。这些操作可能会使 Amazon Web Services 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 OpenSearch 摄取资源。这些操作可能会使 Amazon Web Services 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 Amazon 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的Amazon 托管策略。它们在你的版本中可用 Amazon Web Services 账户。我

们建议您通过定义针对您的用例的 Amazon 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 Amazon 托管式策略或工作职能的Amazon 托管式策略。

- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 Amazon Web Services 服务，例如 Amazon CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA)-如果 Amazon Web Services 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

在控制台中使用 OpenSearch Ingestion

要在 OpenSearch 服务控制台中访问 OpenSearch Ingestion，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您 Amazon 账户中的 OpenSearch Ingestion 资源的详细信息。如果您创建的基于身份的策略比所需的最低权限更严格，则无法为具有该策略的实体（如 IAM 角色）正常运行控制台。

对于仅调用 Amazon CLI 或 Amazon API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

以下策略允许用户在 OpenSearch 服务控制台中访问 OpenSearch Ingestion：

JSON

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

```
            "Effect": "Allow",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ]
        }
    ]
}
```

或者，您可以使用the section called “AmazonOpenSearchIngestionReadOnlyAccess” Amazon Web Services 托管策略，该策略授予对所有 OpenSearch Ingestion 资源的只读访问权限。 Amazon Web Services 账户

管理 OpenSearch 摄取管道

此策略是 “管道管理员” 策略的一个示例，该策略允许用户管理和管理 Amazon OpenSearch Ingestion 管道。用户可以创建、查看和删除管道。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:region:123456789012: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"
        }
    ]
}
```

将数据摄取到摄取管道中 OpenSearch

此示例策略允许用户或其他实体将数据提取到其账户中的 Amazon OpenSearch Ingestion 管道中。用户无法修改管道。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:region:123456789012:pipeline/*",
            "Action": [
                "osis:Ingest"
            ],
            "Effect": "Allow"
        }
    ]
}
```

使用记录 Amazon OpenSearch Ingestion API 调用 Amazon CloudTrail

Amazon OpenSearch Ingestion 与集成 Amazon CloudTrail，后者是在 Ingestion 中提供用户、角色或服务所采取操作的记录的 Amazon OpenSearch 服务。

CloudTrail 将针对 OpenSearch Ingestion 的所有 API 调用作为事件捕获。捕获的调用包括来自 OpenSearch 服务控制台 OpenSearch Ingestion 部分的调用，以及针对 In OpenSearch gestion API 操作的代码调用。

如果您创建跟踪，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 OpenSearch Ingestion 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台的 Event history (事件历史记录) 中查看最新事件。

通过使用收集的信息 CloudTrail，您可以确定向 OpenSearch Ingestion 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间，以及其他详细信息。

要了解更多信息 CloudTrail，请参阅Amazon CloudTrail 用户指南。

OpenSearch 中的摄取信息 CloudTrail

CloudTrail 在您创建账户 Amazon Web Services 账户 时已在您的账户上启用。当 OpenSearch Ingestion 中发生活动时，该活动将记录在事件中，并与其他 Amazon 服务 CloudTrail 事件一同保存在 Event histion 中。您可以在 Amazon Web Services 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录中的事件（包括 OpenSearch Ingestion 的事件），请 Amazon Web Services 账户创建跟踪。通过跟踪 CloudTrail ，您可以将日志文件传送至 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 Amazon Web Services 区域。

此跟踪记录来自 Amazon 分区中的所有区域的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 Amazon 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取操作。有关更多信息，请参阅下列内容：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 为配置 Amazon SNS 通知 CloudTrail
- 从多个区域接收 CloudTrail 日志文件和从多个账户中接收 CloudTrail 日志文件

所有 OpenSearch 摄取操作均由 In OpenSearch g estion CloudTrail API 参考记录并记录在内。例如，对 `CreateCollection`、`ListCollections` 和 `DeleteCollection` 操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可以帮助您确定：

- 请求是使用根用户凭证还是 Amazon Identity and Access Management (IAM) 用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 Amazon 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 OpenSearch 摄取日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。

事件表示来自任何源的单个请求。它包括有关所请求操作的信息、操作的日期和时间、请求参数等。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了`DeletePipeline`操作。

```
{
    "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"
}
```

标记 Amazon OpenSearch Ingestion 管道

标签允许您将任意信息分配给 Amazon OpenSearch Ingestion 管道，以便您可以对该信息进行分类和筛选。标签是您或为 Amazon 资源分配的元数据 Amazon 标记。每个标签均包含一个键 和一个值。对于您分配的标签，需要定义键和值。例如，您可以将键定义为 `stage`，将一个资源的值定义为 `test`。

标签可帮助您：

- 标识和整理您的 Amazon 资源。许多 Amazon 服务支持标记，因此，您可以将同一标签分配给来不同服务的资源，以指示这些资源是相关的。例如，您可以将相同标签分配给您分配给 Amaz OpenSearch on S OpenSearch ervice 域的 Ingestion 管道。
- 跟踪您的 Amazon 成本。您可以在 Amazon 账单与成本管理 控制面板上激活这些标签。 Amazon 使用标签对您的成本进行分类，并向您提供每月成本分配报告。有关更多信息，请参阅《Amazon Billing 用户指南》https://docs.amazonaws.cn/awsaccountbilling/latest/aboutv2/中的使用成本分配标签。
- 使用基于属性的访问控制限制管道访问。有关更多信息，请参阅 IAM 用户指南中的根据标签密钥控制访问。

在 OpenSearch Ingestion 中，主要资源为管道。您可以使用 OpenSearch 服务控制台、 Amazon CLI、 OpenSearch Ingestion APIs 或在 Amazon SDKs 管道中添加、管理和删除标签。

主题

- 所需权限
- 使用标签 (控制台)
- 使用标签 (Amazon CLI)

所需权限

OpenSearch Ingestion 使用以下 Amazon Identity and Access Management Access Analyzer (IAM) 权限来标记管道：

- `osis:TagResource`
- `osis:ListTagsForResource`
- `osis:UntagResource`

有关每项权限的更多信息，请参阅《服务授权参考 OpenSearch 》中的 “摄取操作、资源和条件密钥”。

使用标签 (控制台)

控制台是标记管道的最简单方法。

创建标签

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择管道。
3. 选择您要将标签添加到的管道，然后转到标签选项卡。
4. 选择 Manage（管理）和 Add new tag（添加新标签）。
5. 输入一个标签键和可选的值。
6. 选择保存。

要删除标签，请按照相同步骤操作并在 Manage tags（管理标签）页面中选择 Remove（删除）。

有关使用控制台处理标签的更多信息，请参阅《Amazon 管理控制台入门指南》中的标签编辑器。

使用标签 (Amazon CLI)

要使用标记管道 Amazon CLI，`TagResource`请发送请求：

```
aws osis tag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tags Key=service,Value=osis Key=source,Value=otel
```

使用 `UntagResource` 命令从管道中删除标签：

```
aws osis untag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tag-keys service
```

使用 `ListTagsForResource` 命令查看现有的管道标签：

```
aws osis list-tags-for-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
```

使用 Amazon 记录和监控 Amazon OpenSearch Ingestion CloudWatch

Amazon OpenSearch Ingestion 将指标和日志发布到 Amazon。 CloudWatch

主题

- 监控管道日志
- 监控管道指标

监控管道日志

您可以为 Amazon OpenSearch Ingestion 管道启用日志记录，以公开管道操作和摄取活动期间出现的错误和警告消息。 OpenSearch Ingestion 将所有日志发布到 A ma CloudWatch zon Logs。 CloudWatch 日志可以监控日志文件中的信息，并在达到特定阈值时通知您。您还可以在高持久性存储中检索您的日志数据。有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南。

来自 OpenSearch Ingestion 的日志可能指示请求处理失败、从源到接收器的身份验证错误以及其他有助于排除故障的警告。对于日志， OpenSearch Ingestion 使用`INFO`、`WARN``ERROR`、和的日志级别。`FATAL`我们建议为所有管道启用日志发布。

所需权限

为使 OpenSearch Ingestion 将日志发送到 Log CloudWatch s，必须以具有特定 IAM 权限的用户身份登录。

您需要以下 CloudWatch Logs 权限才能创建和更新日志传送资源：

```
{a
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
```

```
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:GetLogDelivery",
                "logs:ListLogDeliveries"
            ]
        }
    ]
}
```

启用日志发布

您可以在现有管道上启用日志发布，也可以在创建管道时启用日志发布。有关在管道创建期间启用日志发布的步骤，请参阅 the section called “创建管道”。

控制台

在现有管道上启用发布日志

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择管道。
3. 打开要启用日志的管道，然后选择操作、编辑日志发布选项。
4. 启用 “发布到 CloudWatch 日志”。
5. 创建新日志组或选择现有日志组。我们建议您将名称设置为路径格式，例如 `/aws/vendedlogs/OpenSearchIngestion/pipeline-name/audit-logs`。此格式可以更轻松地应用 CloudWatch 访问策略，为特定路径（例如）下的所有日志组授予权限`/aws/vendedlogs/OpenSearchIngestion`。

   > **Important**
   >
   > 必须在日志组名称中包含前缀 `vendedlogs`，否则创建失败。

6. 选择保存。

CLI

要使用启用日志发布 Amazon CLI，请发送以下请求：

```
aws osis update-pipeline \
  --pipeline-name my-pipeline \
  --log-publishing-options  IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="/
aws/vendedlogs/OpenSearchIngestion/pipeline-name"}
```

监控管道指标

您可以使用 Amazon 监控 Amazon OpenSearch Ingestion。Amazon CloudWatch Ingestion 管道会收集原始数据并将其处理为易读且近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

OpenSearch Ingestion 控制台在每个管道的性能选项卡上显示一系列基于原始数据的图表。CloudWatch

OpenSearch Ingestion 将报告绝大多数支持插件的指标。如果特定插件下方未显示专属表，则意味着插件未报告任何特定于插件的指标。管道指标发布在 `AWS/OSIS` 命名空间中。

主题

- 通用指标
- 缓冲区指标
- Signature V4 指标
- 有界阻塞缓冲区指标
- Otel 跟踪源指标
- Otel 指标源指标
- Http 指标
- S3 指标
- 聚合指标
- 日期指标
- Lambda 指标
- Grok 指标
- Otel 跟踪原始指标
- Otel 跟踪组指标
- 服务映射有状态指标

- OpenSearch 指标
- 系统和计量指标

通用指标

以下指标适用于所有处理器和接收器。

每个指标均以子管道名称和插件名称为前缀，格式为 < *sub_pipeline_name* >< >< >< > *plugin*。*metric_name*例如，名为 my-pipeline 的子管道的 recordsIn.count 指标和日期处理器的全名为 my-pipeline.date.recordsIn.count。

指标后缀	描述
recordsIn.count	记录进入管道组件的入口。此指标适用于处理器和接收器。 相关统计数据：总计 维度：PipelineName
recordsOut.count	从管道组件输出记录的出口。此指标适用于处理器和源。 相关统计数据：总计 维度：PipelineName
timeElapsed.count	执行管道组件期间记录的数据点计数。此指标适用于处理器和接收器。 相关统计数据：总计 维度：PipelineName
timeElapsed.sum	执行管道组件花费的总时间。此指标适用于处理器和接收器（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
timeElapsed.max	执行管道组件花费的最长时间。此指标适用于处理器和接收器（以毫秒为单位）。

指标后缀	描述
	相关统计数据：最大值 维度：PipelineName

缓冲区指标

以下指标适用于 OpenSearch Ingestion 为所有管道自动配置的默认有界阻塞缓冲区。

每个指标均以子管道名称和缓冲区名称为前缀，格式为 < *sub_pipeline_name* >< >< >< > *buffer_name*。*metric_name*例如，名为 my-pipeline 的子管道的 recordsWritten.count 指标的全名为 my-pipeline.BlockingBuffer.recordsWritten.count。

指标后缀	描述
recordsWritten.count	写入缓冲区的记录数。 相关统计数据：总计 维度：PipelineName
recordsRead.count	从缓冲区读取的记录数。 相关统计数据：总计 维度：PipelineName
recordsInFlight.value	从缓冲区读取的未检查记录数。 相关统计数据：Average 维度：PipelineName
recordsInBuffer.value	缓冲区当前包含的记录数。 相关统计数据：Average 维度：PipelineName
recordsProcessed.count	从缓冲区读取并由管道处理的记录数。

指标后缀	描述
	相关统计数据：总计 维度：PipelineName
recordsWriteFailed.count	管道无法写入接收器的记录数。 相关统计数据：总计 维度：PipelineName
writeTimeElapsed.count	写入缓冲区时记录的数据点计数。 相关统计数据：总计 维度：PipelineName
writeTimeElapsed.sum	写入缓冲区花费的总时间（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
writeTimeElapsed.max	写入缓冲区花费的最长时间（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
writeTimeouts.count	缓冲区写入超时计数。 相关统计数据：总计 维度：PipelineName
readTimeElapsed.count	从缓冲区读取时记录的数据点计数。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
readTimeElapsed.sum	从缓冲区读取花费的总时间（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
readTimeElapsed.max	从缓冲区读取花费的最长时间（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
checkpointTimeElapsed.count	检查点时记录的数据点计数。 相关统计数据：总计 维度：PipelineName
checkpointTimeElapsed.sum	检查点花费的总时间（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
checkpointTimeElapsed.max	检查点花费的最长时间（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName

Signature V4 指标

以下指标适用于管道摄取端点，并与源插件（http、otel_trace 和 otel_metrics）相关联。向摄取端点发送的所有请求必须使用 Signature 版本 4 签名。这些指标有助于您在连接管道时识别授权问题，或者确认是否已成功进行身份验证。

每个指标均以子管道名称和 osis_sigv4_auth 为前缀。例如 *sub_pipeline_name*.osis_sigv4_auth.httpAuthSuccess.count。

指标后缀	描述
`httpAuthSuccess.count`	向管道发送的成功 Signature V4 请求数。 相关统计数据：总计 维度：`PipelineName`
`httpAuthFailure.count`	向管道发送的失败 Signature V4 请求数。 相关统计数据：总计 维度：`PipelineName`
`httpAuthServerError.count`	向管道发送并返回服务器错误的 Signature V4 请求数。 相关统计数据：总计 维度：`PipelineName`

有界阻塞缓冲区指标

以下指标适用于有界阻塞缓冲区。每个指标均以子管道名称和 `BlockingBuffer` 为前缀。例如 *`sub_pipeline_name`*`.BlockingBuffer.bufferUsage.value`。

指标后缀	描述
`bufferUsage.value`	基于缓冲区记录数计算得出的 `buffer_size` 使用率。`buffer_size` 表示写入缓冲区的最大记录数以及正在进行的未检查最大记录数。 相关统计数据：Average 维度：`PipelineName`

Otel 跟踪源指标

以下指标适用于OTel 跟踪源。每个指标均以子管道名称和 `otel_trace_source` 为前缀。例如 *`sub_pipeline_name`*`.otel_trace_source.requestTimeouts.count`。

指标后缀	描述
requestTimeouts.count	超时请求数。 相关统计数据：总计 维度：PipelineName
requestsReceived.count	插件收到的请求数。 相关统计数据：总计 维度：PipelineName
successRequests.count	插件已成功处理的请求数。 相关统计数据：总计 维度：PipelineName
badRequests.count	插件已处理的无效格式请求数。 相关统计数据：总计 维度：PipelineName
requestsTooLarge.count	内容中的 span 数大于缓冲区容量的请求数。 相关统计数据：总计 维度：PipelineName
internalServerError.count	采用自定义异常类型的插件处理的请求数。 相关统计数据：总计 维度：PipelineName
requestProcessDuration.count	插件处理请求时记录的数据点计数。 相关统计数据：总计

指标后缀	描述
	维度：PipelineName
requestProcessDuration.sum	插件处理的请求的总延迟（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
requestProcessDuration.max	插件处理的请求的最大延迟（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
payloadSize.count	传入请求的有效负载大小的分布计数（以字节为单位）。 相关统计数据：总计 维度：PipelineName
payloadSize.sum	传入请求的有效负载大小的总分布（以字节为单位）。 相关统计数据：总计 维度：PipelineName
payloadSize.max	传入请求的有效负载大小的最大分布（以字节为单位）。 相关统计数据：最大值 维度：PipelineName

Otel 指标源指标

以下指标适用于OTel 指标源。每个指标均以子管道名称和 otel_metrics_source 为前缀。例如 *sub_pipeline_name*.otel_metrics_source.requestTimeouts.count。

指标后缀	描述
requestTimeouts.count	超时插件请求总数。 相关统计数据：总计 维度：PipelineName
requestsReceived.count	插件收到的请求总数。 相关统计数据：总计 维度：PipelineName
successRequests.count	插件成功处理（200 响应状态代码）的请求数。 相关统计数据：总计 维度：PipelineName
requestProcessDuration.count	插件处理的请求的延迟计数（以秒为单位）。 相关统计数据：总计 维度：PipelineName
requestProcessDuration.sum	插件处理的请求的总延迟（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
requestProcessDuration.max	插件处理的请求的最大延迟（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
payloadSize.count	传入请求的有效负载大小的分布计数（以字节为单位）。 相关统计数据：总计

指标后缀	描述
	维度：PipelineName
payloadSize.sum	传入请求的有效负载大小的总分布（以字节为单位）。 相关统计数据：总计 维度：PipelineName
payloadSize.max	传入请求的有效负载大小的最大分布（以字节为单位）。 相关统计数据：最大值 维度：PipelineName

Http 指标

以下指标适用于 HTTP 源。每个指标均以子管道名称和 http 为前缀。例如 *sub_pipeline_name*.http.requestsReceived.count。

指标后缀	描述
requestsReceived.count	/log/ingest 端点收到的请求数。 相关统计数据：总计 维度：PipelineName
requestsRejected.count	插件拒绝（429 响应状态代码）的请求数。 相关统计数据：总计 维度：PipelineName
successRequests.count	插件成功处理（200 响应状态代码）的请求数。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
badRequests.count	插件处理的内容类型或格式无效（400 响应状态代码）的请求数。 相关统计数据：总计 维度：PipelineName
requestTimeouts.count	HTTP 源服务器中超时（415 响应状态代码）的请求数。 相关统计数据：总计 维度：PipelineName
requestsTooLarge.count	内容中事件大小大于缓冲区容量（413 响应状态代码）的请求数。 相关统计数据：总计 维度：PipelineName
internalServerError.count	采用自定义异常类型的插件（500 响应状态代码）处理的请求数。 相关统计数据：总计 维度：PipelineName
requestProcessDuration.count	插件处理的请求的延迟计数（以秒为单位）。 相关统计数据：总计 维度：PipelineName
requestProcessDuration.sum	插件处理的请求的总延迟（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
requestProcessDuration.max	插件处理的请求的最大延迟（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
payloadSize.count	传入请求的有效负载大小的分布计数（以字节为单位）。 相关统计数据：总计 维度：PipelineName
payloadSize.sum	传入请求的有效负载大小的总分布（以字节为单位）。 相关统计数据：总计 维度：PipelineName
payloadSize.max	传入请求的有效负载大小的最大分布（以字节为单位）。 相关统计数据：最大值 维度：PipelineName

S3 指标

以下指标适用于 S3 源。每个指标均以子管道名称和 s3 为前缀。例如 *sub_pipeline_name*.s3.s3ObjectsFailed.count。

指标后缀	描述
s3ObjectsFailed.count	插件无法读取的 S3 对象总数。 相关统计数据：总计 维度：PipelineName
s3ObjectsNotFound.count	因 S3 Not Found错误导致插件无法读取的 S3 对象的数量。这些指标也将计入 s3ObjectsFailed 指标。

指标后缀	描述
	相关统计数据：总计 维度：PipelineName
s3ObjectsAccessDenied.count	因 S3 Access Denied 或 Forbidden 错误导致插件无法读取的 S3 对象的数量。这些指标也将计入 s3ObjectsFailed 指标。 相关统计数据：总计 维度：PipelineName
s3ObjectReadTimeElapsed.count	插件对 S3 对象执行 GET 请求、解析请求并将事件写入缓冲区花费的时间。 相关统计数据：总计 维度：PipelineName
s3ObjectReadTimeElapsed.sum	插件对 S3 对象执行 GET 请求、解析请求并将事件写入缓冲区花费的总时间（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
s3ObjectReadTimeElapsed.max	插件对 S3 对象执行 GET 请求、解析请求并将事件写入缓冲区花费的最长时间（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
s3ObjectSizeBytes.count	S3 对象大小的分布计数（以字节为单位）。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
s3ObjectSizeBytes.sum	S3 对象大小的总分布（以字节为单位）。 相关统计数据：总计 维度：PipelineName
s3ObjectSizeBytes.max	S3 对象大小的最大分布（以字节为单位）。 相关统计数据：最大值 维度：PipelineName
s3ObjectProcessedBytes.count	插件处理的 S3 对象的分布计数（以字节为单位）。 相关统计数据：总计 维度：PipelineName
s3ObjectProcessedBytes.sum	插件处理的 S3 对象的总分布（以字节为单位）。 相关统计数据：总计 维度：PipelineName
s3ObjectProcessedBytes.max	插件处理的 S3 对象的最大分布（以字节为单位）。 相关统计数据：最大值 维度：PipelineName
s3ObjectsEvents.count	插件收到的 S3 事件的分布计数。 相关统计数据：总计 维度：PipelineName
s3ObjectsEvents.sum	插件收到的 S3 事件的总分布。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
s3ObjectsEvents.max	插件收到的 S3 事件的最大分布。 相关统计数据：最大值 维度：PipelineName
sqsMessageDelay.count	S3 记录创建对象的事件时间到完全解析对象所记录的数据点计数。 相关统计数据：总计 维度：PipelineName
sqsMessageDelay.sum	S3 记录创建对象的事件时间到完全解析对象的总时间（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
sqsMessageDelay.max	S3 记录创建对象的事件时间到完全解析对象的最长时间（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
s3ObjectsSucceeded.count	插件成功读取的 S3 对象数量。 相关统计数据：总计 维度：PipelineName
sqsMessagesReceived.count	插件从队列中收到的 Amazon SQS 消息数量。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
sqsMessagesDeleted.count	插件从队列中删除的 Amazon SQS 消息数量。 相关统计数据：总计 维度：PipelineName
sqsMessagesFailed.count	插件无法解析的 Amazon SQS 消息数量。 相关统计数据：总计 维度：PipelineName

聚合指标

以下指标适用于聚合处理器。每个指标均以子管道名称和 aggregate 为前缀。例如 *sub_pipeline_name*.aggregate.actionHandleEventsOut.count。

指标后缀	描述
actionHandleEvents Out.count	对已配置操作调用 handleEvent 返回的事件数。 相关统计数据：总计 维度：PipelineName
actionHandleEvents Dropped.count	对已配置操作调用 handleEvent 返回的事件数。 相关统计数据：总计 维度：PipelineName
actionHandleEvents ProcessingErrors.count	针对导致错误的已配置操作调用 handleEvent 的次数。 相关统计数据：总计 维度：PipelineName
actionConcludeGrou pEventsOut.count	对已配置操作调用 concludeGroup 返回的事件数。

指标后缀	描述
	相关统计数据：总计 维度：PipelineName
actionConcludeGroupEventsDropped.count	对已配置操作调用 condludeGroup 未返回的事件数。 相关统计数据：总计 维度：PipelineName
actionConcludeGroupEventsProcessingErrors.count	针对导致错误的已配置操作调用 concludeGroup 的次数。 相关统计数据：总计 维度：PipelineName
currentAggregateGroups.value	当前组数。当组结束时，此量规会减小；当事件发起创建新组时，此量规会增大。 相关统计数据：Average 维度：PipelineName

日期指标

以下指标适用于日期处理器。每个指标均以子管道名称和 date 为前缀。例如 *sub_pipeline_name*.date.dateProcessingMatchSuccess.count。

指标后缀	描述
dateProcessingMatchSuccess.count	与 match 配置选项中指定的至少一个模式匹配的记录数。 相关统计数据：总计 维度：PipelineName
dateProcessingMatchFailure.count	与 match 配置选项中指定的任何模式均不匹配的记录数。

指标后缀	描述
	相关统计数据：总计 维度：PipelineName

Lambda 指标

以下指标适用于Amazon Lambda处理器。每个指标均以子管道名称和 lambda 为前缀。例如 *sub_pipeline_name*.lambda.recordsSuccessfullySentToLambda.count。

指标后缀	描述
recordsSuccessfullySentToLambda.count	Lambda 函数成功处理的记录数。 相关统计数据：总计 维度：PipelineName
recordsFailedToSendToLambda.count	未能发送到 Lambda 函数的记录数。 相关统计数据：总计 维度：PipelineName
lambdaFunctionLatency.avg lambdaFunctionLatency.max	Lambda 函数调用的延迟。 相关统计数据：平均值和最大值 维度：PipelineName
numberOfRequestsSucceeded.count	成功的 Lambda 调用请求总数。 相关统计数据：总计 维度：PipelineName
numberOfRequestsFailed.count	Lambda 调用失败的总数。 相关统计数据：总计

指标后缀	描述
	维度：PipelineName
requestPayloadSize.avg	发送到 Lambda 的请求负载的大小。 相关统计数据：Average 维度：PipelineName
responsePayloadSize.avg	从 Lambda 收到的响应负载的大小。 相关统计数据：Average 维度：PipelineName

Grok 指标

以下指标适用于 Grok 处理器。每个指标均以子管道名称和 grok 为前缀。例如 *sub_pipeline_name*.grok.grokProcessingMatch.count。

指标后缀	描述
grokProcessingMatch.count	从 match 配置选项中找到至少一个模式匹配的记录数。 相关统计数据：总计 维度：PipelineName
grokProcessingMismatch.count	与 match 配置选项中指定的任何模式均不匹配的记录数。 相关统计数据：总计 维度：PipelineName
grokProcessingErrors.count	记录处理错误数量。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
grokProcessingTimeouts.count	匹配超时的记录数。 相关统计数据：总计 维度：PipelineName
grokProcessingTime.count	当单个记录与 match 配置选项中的模式匹配时记录的数据点计数。 相关统计数据：总计 维度：PipelineName
grokProcessingTime.sum	每条记录与 match 配置选项中的模式进行匹配花费的总时间（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
grokProcessingTime.max	每条记录与 match 配置选项中的模式进行匹配花费的最长时间（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName

Otel 跟踪原始指标

以下指标适用于OTel 跟踪原始处理器。每个指标均以子管道名称和 otel_trace_raw 为前缀。例如 *sub_pipeline_name*.otel_trace_raw.traceGroupCacheCount.value。

指标后缀	描述
traceGroupCacheCount.value	跟踪组缓存中的跟踪组数量。 相关统计数据：总计

指标后缀	描述
	维度：PipelineName
spanSetCount.value	span 集集合中的 span 集数量。 相关统计数据：总计 维度：PipelineName

Otel 跟踪组指标

以下指标适用于OTel 跟踪组处理器。每个指标均以子管道名称和 otel_trace_group 为前缀。例如 *sub_pipeline_name*.otel_trace_group.recordsInMissingTraceGroup.count。

指标后缀	描述
recordsInMissingTraceGroup.count	缺少追踪组字段的入口记录数量。 相关统计数据：总计 维度：PipelineName
recordsOutFixedTraceGroup.count	成功填充跟踪组字段的出口记录数量。 相关统计数据：总计 维度：PipelineName
recordsOutMissingTraceGroup.count	缺少追踪组字段的出口记录数量。 相关统计数据：总计 维度：PipelineName

服务映射有状态指标

以下指标适用于服务映射有状态处理器。每个指标均以子管道名称和 service-map-stateful 为前缀。例如 *sub_pipeline_name*.service-map-stateful.spansDbSize.count。

指标后缀	描述
`spansDbSize.value`	当前窗口持续时间及上一窗口持续时间内 MapDB 中的 span 内存字节大小。 相关统计数据：Average 维度：`PipelineName`
`traceGroupDbSize.value`	当前窗口持续时间及上一窗口持续时间内 MapDB 中的跟踪组内存字节大小。 相关统计数据：Average 维度：`PipelineName`
`spansDbCount.value`	当前窗口持续时间及上一窗口持续时间内 MapDB 中的 span 计数。 相关统计数据：总计 维度：`PipelineName`
`traceGroupDbCount.value`	当前窗口持续时间及上一窗口持续时间内 MapDB 中的跟踪组计数。 相关统计数据：总计 维度：`PipelineName`
`relationshipCount.value`	当前窗口持续时间及上一窗口持续时间内存储的关系计数。 相关统计数据：总计 维度：`PipelineName`

OpenSearch 指标

以下指标适用于OpenSearch水槽。每个指标均以子管道名称和 `opensearch` 为前缀。例如 *`sub_pipeline_name`*`.opensearch.bulkRequestErrors.count`。

指标后缀	描述
bulkRequestErrors.count	发送批量请求时遇到的错误总数。 相关统计数据：总计 维度：PipelineName
documentsSuccess.count	通过批量请求（包括重试）成功发送到 Serv OpenSearch ice 的文档数量。 相关统计数据：总计 维度：PipelineName
documentsSuccessFirstAttempt.count	首次尝试通过批量请求成功发送到 Serv OpenSearch ice 的文档数量。 相关统计数据：总计 维度：PipelineName
documentErrors.count	批量请求发送失败的文档数量。 相关统计数据：总计 维度：PipelineName
bulkRequestFailed.count	失败的批量请求数量。 相关统计数据：总计 维度：PipelineName
bulkRequestNumberOfRetries.count	失败批量请求的重试次数。 相关统计数据：总计 维度：PipelineName
bulkBadRequestErrors.count	发送批量请求时遇到的 Bad Request 错误数量。

指标后缀	描述
	相关统计数据：总计 维度：PipelineName
bulkRequestNotAllowedErrors.count	发送批量请求时遇到的 Request Not Allowed 错误数量。 相关统计数据：总计 维度：PipelineName
bulkRequestInvalidInputErrors.count	发送批量请求时遇到的 Invalid Input 错误数量。 相关统计数据：总计 维度：PipelineName
bulkRequestNotFoundErrors.count	发送批量请求时遇到的 Request Not Found 错误数量。 相关统计数据：总计 维度：PipelineName
bulkRequestTimeoutErrors.count	发送批量请求时遇到的 Request Timeout 错误数量。 相关统计数据：总计 维度：PipelineName
bulkRequestServerErrors.count	发送批量请求时遇到的 Server Error 错误数量。 相关统计数据：总计 维度：PipelineName
bulkRequestSizeBytes.count	批量请求的有效负载大小的分布计数（以字节为单位）。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
bulkRequestSizeBytes.sum	批量请求的有效负载大小的总分布（以字节为单位）。 相关统计数据：总计 维度：PipelineName
bulkRequestSizeBytes.max	批量请求的有效负载大小的最大分布（以字节为单位）。 相关统计数据：最大值 维度：PipelineName
bulkRequestLatency.count	请求（包括重试）发送到插件时记录的数据点计数。 相关统计数据：总计 维度：PipelineName
bulkRequestLatency.sum	发送到插件的请求（包括重试）的总延迟（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
bulkRequestLatency.max	发送到插件的请求（包括重试）的最大延迟（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName
s3.dlqS3RecordsSuccess.count	成功发送到 S3 死信队列的记录数。 相关统计数据：总计 维度：PipelineName

指标后缀	描述
s3.dlqS3RecordsFailed.count	未能发送到 S3 死信队列的记录数。 相关统计数据：总计 维度：PipelineName
s3.dlqS3RequestSuccess.count	S3 死信队列请求成功次数。 相关统计数据：总计 维度：PipelineName
s3.dlqS3RequestFailed.count	S3 死信队列请求失败次数。 相关统计数据：总计 维度：PipelineName
s3.dlqS3RequestLatency.count	请求（包括重试）发送到 S3 死信队列时记录的数据点计数。 相关统计数据：总计 维度：PipelineName
s3.dlqS3RequestLatency.sum	发送到 S3 死信队列的请求（包括重试）的总延迟（以毫秒为单位）。 相关统计数据：总计 维度：PipelineName
s3.dlqS3RequestLatency.max	发送到 S3 死信队列的请求（包括重试）的最大延迟（以毫秒为单位）。 相关统计数据：最大值 维度：PipelineName

指标后缀	描述
s3.dlqS3RequestSizeBytes.count	S3 死信队列请求的有效负载大小的分布计数（以字节为单位）。 相关统计数据：总计 维度：PipelineName
s3.dlqS3RequestSizeBytes.sum	S3 死信队列请求的有效负载大小的总分布（以字节为单位）。 相关统计数据：总计 维度：PipelineName
s3.dlqS3RequestSizeBytes.max	S3 死信队列请求的有效负载大小的最大分布（以字节为单位）。 相关统计数据：最大值 维度：PipelineName

系统和计量指标

以下指标适用于整个 OpenSearch Ingestion 系统。这些指标没有任何前缀。

指标	描述
system.cpu.usage.value	所有数据节点的可用 CPU 使用率。 相关统计数据：Average 维度：PipelineName 、area、id
system.cpu.count.value	所有数据节点的 CPU 使用总量。 相关统计数据：Average 维度：PipelineName 、area、id

指标	描述
jvm.memory.max.value	可用于内存管理的最大内存量（以字节为单位）。 相关统计数据：Average 维度：PipelineName 、area、id
jvm.memory.used.value	使用的内存总量（以字节为单位）。 相关统计数据：Average 维度：PipelineName 、area、idsigna
jvm.memory.committed.value	提供给 Java 虚拟机 (JVM) 使用的内存量（以字节为单位）。 相关统计数据：Average 维度：PipelineName 、area、id
computeUnits	管道正在使用的 Ingestion OpenSearch 计算单位（Ingestion OCUs）数量。 相关统计数据：Max、Sum、Average 维度：PipelineName

Amazon OpenSearch Ingestion 的最佳实践

本主题提供创建和管理 Amazon OpenSearch Ingestion 管道的一些最佳实践，并包括适用于许多用例的一般准则。每个工作负载都是独一无二的，具有独特的特征，因此不存在完全适合所有使用案例的万能建议。

主题

- 一般最佳实践
- 建议 CloudWatch 警报

一般最佳实践

以下一般最佳实践适用于创建和管理管道。

- 为确保高可用性，请使用两个或三个子网配置 VPC 管道。如果您只在一个子网中部署管道，则可用区出现故障时，您将无法摄取数据。
- 在每个管道中，建议将子管道的数量限制在 5 个或更少。
- 如果您使用的是 S3 源插件，请使用大小一致的 S3 文件以获得最佳性能。
- 如果您使用的是 S3 源插件，请为 S3 存储桶中每 0.25 GB 的文件大小增加 30 秒的额外可见性超时时间，以获得最佳性能。
- 在管道配置中加入死信队列 (DLQ)，这样您就可以卸载失败的事件并使其可供分析。如果您的接收器由于映射不正确或其他问题而拒绝数据，则可以将数据路由到 DLQ 以进行故障排除并修复问题。

建议 CloudWatch 警报

CloudWatch 当一段时间内 Marc CloudWatch h 值超出指定的值时，警报将会执行某个操作。例如，如果您的集群运行状况超过一分钟，您可能需要 Amazon `red`给您发送电子邮件。本部分包括一些 Amazon OpenSearch Ingestion 建议警报及其响应方式。

有关配置警报的更多信息，请参阅亚马逊 CloudWatch 用户指南中的创建亚马逊 CloudWatch警报。

警报	事务
`computeUnits` 最大 = 配置的 `maxUnits` 达到 15 分钟，连续 3 次	管道已达到最大容量，可能需要 `maxUnits` 更新。增加管道的最大容量
`opensearch.documentErrors.count` 总计 = *`{sub_pipeline_name}`* `.opensearch.recordsIn.count` 总计达到 1 分钟，连续 1 次	该管道无法写 OpenSearch 入。检查管道权限并确认域或集合运行状况良好。您还可以检查死信队列 (DLQ) 中是否存在失败的事件（如果已配置）。

警报	事务
bulkRequestLatency.max 最大 >= x 达到 1 分钟，连续 1 次	该管道在向 OpenSearch 接收器发送数据时遇到了高延迟。这可能是由于接收器过小，或者分片策略不佳，从而导致接收器落后。持续的高延迟会影响管道性能，并可能给客户端带来反向压力。
httpAuthFailure.count 总计 >= 1 达到 1 分钟，连续 1 次	未对提取请求进行身份验证。确认所有客户端均已正确启用签名版本 4 身份验证。
system.cpu.usage.value 平均 >= 80% 达到 15 分钟，连续 3 次	CPU 持续的高使用率可能会出现问题。考虑增加管道的最大容量。
bufferUsage.value 平均 >= 80% 达到 15 分钟，连续 3 次	缓存持续的高使用率可能会出现问题。考虑增加管道的最大容量。

您可能会考虑的其他警报

请考虑根据您经常使用的 Amazon OpenSearch Ingestion 功能配置以下警报。

警报	事务
dynamodb.exportJobFailure.count 总计为 1	尝试触发导出到 Amazon S3 失败。
opensearch.EndtoEndLatency.avg 平	从 DynamoDB 流中读取时，EndtoEndLatency 高于预期值。这可能是由于， OpenSearch 集群规模过小，或者管道 OCU 的最大容量过低，无法满足 DynamoDB 表中的 WCU 吞吐量。 EndtoEndLatency 导出

警报	事务
均 > X 达到 15 分钟，连续 4 次	后会更高，但随着时间的推移，它会随着时间的推移而降低，因为它会赶上最新的 DynamoDB 流。
`dynamodb.changeEventsProcessed.count` 总计 == 0 达到 X 分钟	没有从 DynamoDB 流收集任何记录。这可能是因为表中没有任何活动，或者访问 DynamoDB 流时出现问题。
`opensearch.s3.dlqS3RecordsSuccess.count` 总计 >= `opensearch.documentSuccess.count` 总计达到 1 分钟，连续 1 次	发送到 DLQ 的记录数量多于 OpenSearch 接收器。查看 S OpenSearch ink 插件指标，以调查和确定根本原因。
`grok.grokProcessingTimeouts.count` 总计 = recordsIn.countsum 达到 1 分钟，连续 5 次	当 Grok 处理器尝试模式匹配时，所有数据都超时。这可能会影响性能并减慢您的管道速度。考虑调整模式以减少超时。
`grok.grokProcessingErrors.count` 总计 >= 1 达到 1 分钟，连续 1 次	Grok 处理器无法将模式与管道中的数据进行匹配，从而导致错误。查看您的数据和 Grok 插件配置，以确保模式匹配符合预期。

警报	事务
grok.grokProcessingMismatch.count 总计 = recordsIn.count sum 达到 1 分钟，连续 5 次	Grok 处理器无法将模式与管道中的数据进行匹配。查看您的数据和 Grok 插件配置，以确保模式匹配符合预期。
date.dateProcessingMatchFailure.count 总计 = recordsIn.count sum 达到 1 分钟，连续 5 次	数据处理器无法将模式与管道中的数据进行匹配。查看您的数据和日期插件配置，以确保模式符合预期。
s3.s3ObjectsFailed.count 总计 >= 1 达到 1 分钟，连续 1 次	此问题的原因在于 S3 对象不存在或管道权限不足。查看 s3Objects NotFound.count 和 s3ObjectsAccessDenied.count 指标以确定根本原因。确认 S3 对象存在和/或更新权限。
s3.sqsMessagesFailed.count 总计 >= 1 达到 1 分钟，连续 1 次	S3 插件无法处理 Amazon SQS 消息。如果您在 SQS 队列上启用了 DLQ，请查看失败消息。队列可能正在接收管道正在尝试处理的无效数据。
http.badRequests.count 总计 >= 1 达到 1 分钟，连续 1 次	客户端发送的请求不正确。确认所有客户端都发送了正确的负载。
http.requestsTooLarge.count 总计 >= 1 达到 1 分钟，连续 1 次	来自 HTTP 源插件的请求包含的数据过多，超过了缓冲区容量。调整客户的批量大小。

警报	事务
`http.internalServerError.count` 总计 >= 0 达到 1 分钟，连续 1 次	HTTP 源插件在接收事件时出现问题。
`http.requestTimeouts.count` 总计 >= 0 达到 1 分钟，连续 1 次	源超时可能是由于管道预调配不足所致。考虑增加管道 `maxUnits` 以处理额外的工作负载。
`otel_trace.badRequests.count` 总计 >= 1 达到 1 分钟，连续 1 次	客户端发送的请求不正确。确认所有客户端都发送了正确的负载。
`otel_trace.requestsTooLarge.count` 总计 >= 1 达到 1 分钟，连续 1 次	来自 Otel Trace 源插件的请求包含的数据过多，超过了缓冲区容量。调整客户的批量大小。
`otel_trace.internalServerError.count` 总计 >= 0 达到 1 分钟，连续 1 次	Otel Trace 源插件在接收事件时出现问题。
`otel_trace.requestTimeouts.count` 总计 >= 0 达到 1 分钟，连续 1 次	源超时可能是由于管道预调配不足所致。考虑增加管道 `maxUnits` 以处理额外的工作负载。

警报	事务
otel_metrics.requestTimeouts.count 总计 >= 0 达到 1 分钟，连续 1 次	源超时可能是由于管道预调配不足所致。考虑增加管道 maxUnits 以处理额外的工作负载。

Amazon OpenSearch 无服务器

Amazon OpenSearch Serverless 是一种针对亚马逊 OpenSearch 服务的按需自动缩放配置。与需要手动容量管理的预配置 OpenSearch 域不同， OpenSearch Serverless 集合会根据应用程序的需求自动扩展计算资源。

OpenSearch Serverless 为不频繁、间歇性或不可预测的工作负载提供了经济实惠的解决方案。它通过根据应用程序的使用情况自动扩展计算容量来优化成本。无服务器集合使用与预配置服务域相同的高容量、分布式和高可用性存储量。 OpenSearch

OpenSearch 无服务器集合始终是加密的。您可以选择加密密钥，但不能禁用加密。有关更多信息，请参阅 the section called “加密”。

优势

OpenSearch 无服务器具有以下优点：

- 比配置更简单 — OpenSearch Serverless 消除了管理 OpenSearch 集群和容量的大部分复杂性。它会自动调整集群的大小和进行微调，并负责处理分片和索引生命周期管理。它还管理服务软件更新和 OpenSearch版本升级。所有更新和升级都是无中断的。
- 经济实惠 — 使用 OpenSearch 无服务器时，您只需为消耗的资源付费。这样就无需为峰值工作负载进行提前预调配和超额预调配。
- 高可用性 — OpenSearch Serverless 通过冗余支持生产工作负载，以防可用区中断和基础设施故障。
- 可扩展 — OpenSearch Serverless 会自动扩展资源，以保持持续快速的数据摄取率和查询响应时间。

什么是 Amazon OpenSearch 无服务器？

Amazon OpenSearch Serverless 是 Amazon OpenSearch 服务的按需无服务器选项，它消除了配置、配置和调整 OpenSearch 集群的操作复杂性。它非常适合那些不愿自行管理集群或缺乏用于大规模部署的专用资源和专业知识的组织。借助 OpenSearch Serverless，您可以搜索和分析大量数据，而无需管理底层基础架构。

OpenSearch Serverless 集合是一组 OpenSearch 索引，它们协同工作以支持特定的工作负载或用例。与需要手动配置的自我管理 OpenSearch 集群相比，集合简化了操作。

集合使用与配置的 OpenSearch 服务域相同的高容量、分布式和高可用性存储，但通过消除手动配置和调整，进一步降低了复杂性。集合中的数据在传输过程中经过加密。 OpenSearch Serverless 还支持 OpenSearch 仪表板，为数据分析提供了一个接口。

目前，无服务器集合运行 OpenSearch 版本 2.0.x。随着新版本的发布， OpenSearch Serverless 会自动升级集合以加入新功能、错误修复和性能改进。

OpenSearch Serverless 支持与 OpenSearch 开源套件相同的采集和查询 API 操作，因此您可以继续使用现有的客户端和应用程序。您的客户端必须与 OpenSearch 2.x 兼容，才能使用 OpenSearch Serverless。有关更多信息，请参阅 the section called “将数据摄取到集合中”。

主题

- OpenSearch 无服务器的用例
- 工作方式
- 选择集合类型
- 定价
- 支持的 Amazon Web Services 区域
- 限制
- 比较 OpenSearch 服务和 OpenSearch 无服务器

OpenSearch 无服务器的用例

OpenSearch 无服务器支持两个主要用例：

- 日志分析：日志分析部分专注于分析大量半结构化、机器生成的时间序列数据，以了解操作和用户行为。
- 全文搜索：全文搜索部分可为内部网络中的应用程序（内容管理系统、法律文档）和面向互联网的应用程序（如电子商务网站内容搜索）提供支持。

在创建集合时，您可以选择以下应用场景之一。有关更多信息，请参阅 the section called “选择集合类型”。

工作方式

传统 OpenSearch 集群只有一组执行索引和搜索操作的实例，并且索引存储与计算容量紧密结合。相比之下， OpenSearch Serverless 使用云原生架构，将索引（提取）组件与搜索（查询）组件分开，Amazon S3 作为索引的主要数据存储。

这种分离架构使您可以独立扩展搜索和索引功能，并且独立于 S3 中的索引数据。这种架构还为摄取和查询操作提供了隔离，使它们可以同时运行，而不会发生资源争用。

当您将数据写入集合时， OpenSearch Serverless 会将其分发到索引计算单元。索引计算单位将摄取传入数据，并将索引移至 S3。当您对集合数据执行搜索时， OpenSearch Serverless 会将请求路由到保存所查询数据的搜索计算单元。搜索计算单位直接从 S3 下载索引数据（如果这些数据尚未在本地缓存），运行搜索操作，然后执行聚合。

下图阐明了这种分离架构：

OpenSearch 用于数据摄取、搜索和查询的无服务器计算容量以 OpenSearch 计算单位 () 来衡量。OCUs每个 OCU 是 6GiB 内存和相应的虚拟 CPU（vCPU）以及创建到 Amazon S3 的数据管道的组合。每个 OCU 都包含足够的热临时存储，可存储 120 GiB 的索引数据。

当你创建第一个集合时， OpenSearch Serverless 会实例化两个 OCUs ——一个用于索引，一个用于搜索。为了确保高可用性，它还会在另一个可用区中启动一组备用节点。出于开发和测试目的，您可以禁用集合的启用冗余设置，这将消除两个备用副本，只实例化两个副本。 OCUs默认情况下，冗余活动副本处于启用状态，这意味着账户中的第一个集合总共实例化了四个 OCUs 副本。

即使在任何集合端点上没有任何活动，它们也会 OCUs 存在。所有后续集合都共享这些 OCUs。当您在同一个账户中创建其他集合时， OpenSearch Serverless 只会根据您指定的容量限制，根据需要添加更多 OCUs 用于搜索和收录的集合，以支持这些集合。容量会随着计算使用量的减少而缩减。

有关这些费用如何计费的信息 OCUs，请参阅the section called “定价”。

选择集合类型

OpenSearch Serverless 支持三种主要的集合类型：

时间序列 — 日志分析部分，用于实时分析大量半结构化、机器生成的数据，提供有关运营、安全、用户行为和业务绩效的见解。

搜索 — 全文搜索，支持内部网络中的应用程序，例如内容管理系统和法律文件存储库，以及面向互联网的应用程序，例如电子商务网站搜索和内容发现。

矢量搜索 — 矢量嵌入的语义搜索简化了矢量数据管理，并支持机器学习 (ML) 增强型搜索体验。它支持生成式 AI 应用程序，例如聊天机器人、个人助理和欺诈检测。

在首次创建集合时，您可以选择一种集合类型：

您选择的集合类型取决于您计划摄取到集合中的数据类型，以及您计划如何查询这些数据。在创建集合后，您将无法更改集合类型。

集合类型有以下显著差异：

- 对于搜索和向量搜索集合，所有数据都将存储在热存储中，以确保快速查询响应时间。时间序列集合使用热存储和温存储的组合，其中最新数据保存在热存储中，以优化访问频率更高的数据的查询响应时间。
- 对于时间序列和向量搜索集合，无法按自定义文档 ID 编制索引，也无法按 upsert 请求进行更新。此操作是为搜索应用场景保留的。您可以改为按文档 ID 进行更新。有关更多信息，请参阅 the section called “受支持 OpenSearch 的 API 操作和权限”。
- 对于搜索和时间序列集合，不能使用 k-NN 类型的索引。

定价

Amazon 向您收取以下 OpenSearch 无服务器组件的费用：

- 数据摄取计算
- 搜索和查询计算
- 保留在 Amazon S3 中的存储

它按小时计费 OCU，按每秒精度计费。在您的账户对账单中，您会看到一个以 OCU 小时为单位的计算条目，上面有一个用于数据提取的标签和一个用于搜索的标签。 Amazon 还会按月向您收取存储在 Amazon S3 中的数据的费用。它不会向您收取使用 OpenSearch 仪表板的费用。

创建集合并启用冗余活动副本时，您需要支付至少 2 OCUs （0.5 OCU x 2）的摄取费用和 1 个 OCU（0.5 OCU x 2）的搜索费用。如果您禁用冗余的活动副本，则需要为账户中的第一个集合支付至少 1 个 OCU（0.5 OCU x 2）的费用。所有后续集合都可以共享这些集合 OCUs。

OpenSearch Serverless 根据支持您的馆藏所需的计算能力和存储空间以 1 OCU 为增量添加额外 OCUs 内容。您可以 OCUs 为账户配置最大数量以控制成本。

> **Note**
>
> 具有唯一性的收藏 Amazon KMS keys 无法 OCUs 与其他收藏共享。

OpenSearch Serverless 尝试使用所需的最低资源来应对不断变化的工作负载。随时 OCUs 配置的数量可能会有所不同，而且不准确。随着时间的推移， OpenSearch Serverless使用的算法将继续改进，以更好地最大限度地减少系统使用量。

有关全部定价详情，请参阅 Amazon OpenSearch 服务定价。

支持的 Amazon Web Services 区域

OpenSearch 无服务器 Amazon Web Services 区域 在该 OpenSearch 服务的子集中可用。有关支持区域的列表，请参阅中的亚马逊 OpenSearch 服务终端节点和配额Amazon Web Services 一般参考。

限制

OpenSearch 无服务器有以下限制：

- 不支持某些 OpenSearch API 操作。请参阅the section called “受支持 OpenSearch 的 API 操作和权限”。
- 不支持某些 OpenSearch 插件。请参阅the section called “支持的 OpenSearch 插件”。
- 目前无法将您的数据从托管 OpenSearch 服务域自动迁移到无服务器集合。您必须将数据从域重新索引到集合。
- 不支持跨账户存取集合。您不能将来自其他账户的集合包括在您的加密或数据访问策略中。
- 不支持自定义 OpenSearch 插件。
- 您无法拍摄或恢复 OpenSearch 无服务器集合的快照。
- 不支持跨区域搜索和复制。
- 对于您在一个账户和区域中可以拥有的无服务器资源的数量存在限制。请参阅OpenSearch 无服务器配额。
- 向量搜索集合中索引的刷新间隔约为 60 秒。搜索集合和时间序列集合中索引的刷新间隔约为 10 秒。
- 分片数、间隔数和刷新间隔不可修改，由 OpenSearch Serverless 处理。分片策略基于集合类型和流量。例如，时间序列集合根据写入流量瓶颈扩展主分片。
- 支持 2.1 及以下 OpenSearch 版本中可用的地理空间功能。

比较 OpenSearch 服务和 OpenSearch 无服务器

在 OpenSearch Serverless 中，某些概念和功能与预配置 OpenSearch 服务域的相应功能不同。例如，一个重要的区别是 OpenSearch Serverless 没有集群或节点的概念。

下表描述了 OpenSearch Serverless 中的重要功能和概念与预配置 OpenSearch 服务域中的等效功能有何不同。

功能	OpenSearch 服务	OpenSearch 无服务器
域与集合	索引保存在域中，这些域是预先配置的群集。 OpenSearch 有关更多信息，请参阅 创建和管理域。	索引保存在集合中，集合是代表特定工作负载或应用场景的索引的逻辑分组。 有关更多信息，请参阅 the section called “管理集合”。
节点类型和容量管理	您可以使用符合成本和性能规格的节点类型构建集群。您必须计算自己的存储需求，并针对您的域选择实例类型。 有关更多信息，请参阅 the section called “调整域大小”。	OpenSearch Serverless 会根据您的容量使用情况自动为您的账户扩展和配置额外的计算单元。 有关更多信息，请参阅 the section called “管理容量限制”。
Billing	您需要为实例的每小时使用量以及连接到您的 EC2 实例的任何 EBS 存储卷的累积大小付费。 有关更多信息，请参阅 the section called “定价”。	您需要以 OCU 小时数为单位，为用于数据摄取的计算、用于搜索和查询的计算以及保留在 S3 中的存储付费。 有关更多信息，请参阅 the section called “定价”。
加密	对于域，静态加密是可选的。 有关更多信息，请参阅 the section called “静态加密”。	对于集合，静态加密是必需的。 有关更多信息，请参阅 the section called “加密”。
数据访问控制	针对域中数据的访问权限由 IAM policy 和精细访问控制决定。	针对集合中数据的访问权限由数据访问策略决定。
支持的 OpenSearch 操作	OpenSearch 服务支持所有 OpenSearch API 操作的子集。 有关更多信息，请参阅 the section called “支持的操作”。	OpenSearch 无服务器支持不同的 OpenSearch API 操作子集。 有关更多信息，请参阅 the section called “受支持的操作和插件”。

功能	OpenSearch 服务	OpenSearch 无服务器
控制面板登录	使用用户名和密码登录。 有关更多信息，请参阅 the section called “以主用户身份访问 OpenSearch 仪表板”。	如果您已登录 Amazon 控制台并导航到控制面板 URL，则会自动登录。 有关更多信息，请参阅 the section called “访问 OpenSearch 仪表板”。
APIs	使用 OpenSearch 服务 API 操作以编程方式与OpenSearch 服务交互。	使用 OpenSearch 无服务器 API 操作以编程方式与OpenSearch 无服务器交互。
网络访问	域的网络设置适用于域端点和 OpenSearch 仪表板端点。两者的网络访问紧密结合。	域端点和 OpenSearch 仪表板端点的网络设置是分离的。您可以选择不配置 OpenSearch 仪表板的网络访问权限。 有关更多信息，请参阅 the section called “网络访问”。
签署请求	使用 OpenSearch 高级和低级 REST 客户端签署请求。将服务名称指定为 `es`。	目前， OpenSearch Serverless 支持 OpenSearch 服务支持的一部分客户端。 在签署请求时，请将服务名称指定为 `aoss`。`x-amz-content-sha256` 标头是必需的。有关更多信息，请参阅 the section called “其他客户端”。
OpenSearch 版本升级	随着新版本的上市，您可以手动升级您的域名。 OpenSearch由您负责确保您的域符合升级要求，并且您已解决任何重大更改。	OpenSearch Serverless 会自动将您的收藏升级到新 OpenSearch 版本。升级不一定会在新版本推出后立即进行。
服务软件更新	当服务软件更新推出时，您可以手动将其应用于您的域。	OpenSearch Serverless 会自动更新您的集合，以使用最新的错误修复、功能和性能改进。

功能	OpenSearch 服务	OpenSearch 无服务器
VPC 访问	您可以在 VPC 中预调配您的域。 您还可以创建其他OpenSearch 服务托管 VPC 终端节点来访问该域。	您可以为账户创建一个或多个OpenSearch 无服务器托管的 VPC 终端节点。然后，将这些端点包括在网络策略中。
SAML 身份验证	您可以基于每个域启用 SAML 身份验证。 有关更多信息，请参阅 the section called “仪表板的 SAML 身份验证 OpenSearch ”。	您可以在账户级别配置一个或多个 SAML 提供商，然后将关联的用户和组包含 IDs 在数据访问策略中。 有关更多信息，请参阅 the section called “SAML 身份验证”。
传输层安全性 (TLS)	OpenSearch 服务支持 TLS 1.2，但建议您使用 TLS 1.3。	OpenSearch 无服务器支持 TLS 1.2，但建议您使用 TLS 1.3。

教程：开始使用 Amazon OpenSearch Serverless

本教程将引导您完成快速启动并运行 Amazon OpenSearch Serverless 搜索集合的基本步骤。搜索集合可为内部网络应用程序和面向 Internet 的应用程序（例如，电子商务网站搜索和内容搜索）提供支持。

要了解如何使用向量搜索集合，请参阅 the section called “使用向量搜索集合”。有关使用集合的更多详细信息，请参阅the section called “管理集合”和本指南中的其他主题。

在本教程中，您将完成以下步骤：

1. 配置权限
2. 创建集合
3. 上传和搜索数据
4. 删除集合

> **Note**
>
> 我们建议您仅使用 ASCII 字符作为您的`IndexName`字符。如果您不使用 ASCII 字符`IndexName`，则对于非 ASCII 字符，输入 CloudWatch 指标将转换为网址编码格式。`IndexName`

步骤 1：配置权限

要完成本教程并全面使用 OpenSearch Serverless，您必须拥有正确的 IAM 权限。在本教程中，您将创建一个集合、上传和搜索数据，然后删除该集合。

您的用户或角色必须已经附加基于身份的策略，并且具有以下最低权限：

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": "*"
    }
  ]
}
```

有关 OpenSearch 无服务器 IAM 权限的更多信息，请参阅the section called “身份和访问管理”。

步骤 2：创建集合

集合是一组 OpenSearch 索引，它们协同工作以支持特定的工作负载或用例。

创建 OpenSearch 无服务器集合

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择左侧导航窗格中的 Collections（集合），然后选择 Create collection（创建集合）。
3. 将该集合命名为 movies（电影）。
4. 对于集合类型，选择 Search（搜索）。有关更多信息，请参阅选择集合类型。
5. 对于安全，选择标准创建。
6. 在加密下，选择使用 Amazon 拥有的密钥。这是 OpenSearch Serv Amazon KMS key erless用来加密你的数据的。
7. 在 Network（网络）下，配置集合的网络设置。
   - 对于访问权限类型，选择 Public（公共）。
   - 对于资源类型，选择 “启用 OpenSearch 端点访问权限” 和 “启用对 OpenSearch 仪表板的访问”。由于您将使用 OpenSearch 仪表板上传和搜索数据，因此需要同时启用两者。
8. 选择下一步。
9. 在 Configure data access（配置数据访问权限）中设置集合的访问设置。数据访问策略允许用户和角色访问集合中的数据。在本教程中，我们将为单个用户提供索引和搜索 movies（电影）集合中的数据所需的权限。

   创建一条规则，提供针对影片集合的访问权限。将该规则命名为 Movies collection access（电影集合访问权限）。
10. 选择添加委托人、IAM 用户和角色，然后选择您将用于登录 OpenSearch 控制面板和索引数据的用户或角色。选择保存。
11. 在 Index permissions（索引权限）下，选择所有权限。
12. 选择下一步。
13. 在访问策略设置中，选择 Create a new data access policy（创建新的数据访问策略）并将策略命名为 movies（影片）。
14. 选择下一步。
15. 查看集合设置并选择 提交。稍等几分钟，等集合状态变为 `Active`。

步骤 3：上传并搜索数据

您可以使用 P ostm an 或 curl 将数据上传到 OpenSearch 无服务器集合。为简洁起见，这些示例使用 OpenSearch 仪表板控制台中的开发工具。

索引和搜索“movies”（电影）集合中的数据

1. 选择左侧导航窗格中的 Collections（集合），然后选择 movies（电影）集合，以打开其详细信息页面。
2. 为该集合选择 OpenSearch 仪表板 URL。该 URL 采用 `https://dashboards.{region}.aoss.amazonaws.com/_login/?collectionId={collection-id}` 格式。
3. 在 “ OpenSearch 控制面板” 中，打开左侧导航窗格并选择 “开发工具”。
4. 要创建名为 movies-index 的单个索引，请发送以下请求：

```
PUT movies-index
```

5. 要将单个文档索引到 movies-index 中，请发送以下请求：

```
PUT movies-index/_doc/1
{
  "title": "Shawshank Redemption",
  "genre": "Drama",
  "year": 1994
}
```

6. 要在 OpenSearch 仪表板中搜索数据，您需要配置至少一种索引模式。 OpenSearch 使用这些模式来确定要分析哪些索引。打开左侧导航窗格，选择 Stack Management（堆栈管理），选择 Index Patterns（索引模式），然后选择 Create index pattern（创建索引模式）。在本教程中，请输入 movies。
7. 选择下一步，然后选择创建索引模式。创建模式后，您可以查看各种文档字段，例如 `title` 和 `genre`。
8. 要开始搜索数据，请再次打开左侧导航窗格，然后选择 Discover（发现），或者使用 Dev Tools（开发工具）中的搜索 API。

步骤 4：删除集合

由于 movies（电影）集合用于测试目的，因此请确保在您完成试验后将其删除。

删除 OpenSearch 无服务器集合

1. 返回亚马逊 OpenSearch 服务控制台。
2. 选择左侧导航窗格中的 Collections（集合），然后选择 movies（电影）集合。
3. 选择 Delete（删除），然后确认删除。

后续步骤

现在您已知道如何创建集合和索引数据，您可能想尝试以下一些练习：

- 查看用于创建集合的更多高级选项。有关更多信息，请参阅 the section called “管理集合”。
- 了解如何配置安全策略以大规模管理集合安全性。有关更多信息，请参阅 the section called “ OpenSearch 无服务器中的安全性”。
- 发现将数据索引到集合中的其他方法。有关更多信息，请参阅 the section called “将数据摄取到集合中”。

创建和管理 Amazon OpenSearch 无服务器馆藏

您可以使用控制台、 Amazon CLI 和 API、和创建 Amazon OpenSearch Serverless 集合。 Amazon SDKs Amazon CloudFormation

主题

- 管理 Amazon OpenSearch 无服务器馆藏
- 使用向量搜索集合
- 在 Amazon OpenSearch Serverless 中使用数据生命周期策略
- 使用与 Amazon OpenSearch 无服务器互动 Amazon SDKs
- Amazon CloudFormation 用于创建 Amazon OpenSearch 无服务器集合

管理 Amazon OpenSearch 无服务器馆藏

Amazon OpenSearch 无服务器中的集合是代表分析工作负载的一个或多个索引的逻辑分组。OpenSearch 无服务器可以自动管理和调整集合，只需很少的手动输入。

主题

- 所需权限
- 创建集合
- 访问 OpenSearch 仪表板
- 查看集合
- 删除集合

所需权限

OpenSearch 无服务器使用以下 Amazon Identity and Access Management （IAM）权限来创建和管理集合。您可以指定 IAM 条件，以将用户限制到特定集合。

- `aoss:CreateCollection`：创建集合。
- `aoss:ListCollections`：列出当前账户中的集合。
- `aoss:BatchGetCollection`：获取有关一个或多个集合的详细信息。
- `aoss:UpdateCollection`：修改集合。
- `aoss:DeleteCollection`：删除集合。

以下基于身份的示例访问策略将为用户提供管理名为 `Logs` 的单个集合所需的最低权限：

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

之所以包括 `aoss:CreateAccessPolicy` 和 `aoss:CreateSecurityPolicy`，是因为需要加密、网络和数据访问策略才能使集合正常运行。有关更多信息，请参阅 the section called “身份和访问管理”。

> **Note**
>
> 如果您要在账户中创建第一个集合，则还需要 `iam:CreateServiceLinkedRole` 权限。有关更多信息，请参阅 the section called “集合创建角色”。

创建集合

您可以使用控制台或创建无服务器集合。 Amazon CLI 这些步骤介绍如何创建搜索集合或时间序列集合。要创建向量搜索集合，请参阅 the section called “使用向量搜索集合”。

创建集合（控制台）

使用控制台创建集合

1. 在家中导航至 Amazon OpenSearch 服务控制台 https://console.aws.amazon.com/aos//。
2. 展开左侧导航窗格中的 Serverless（无服务器），然后选择 Collections（集合）。
3. 选择 Create collection（创建集合）。

4. 为集合提供名称和描述。名称必须符合以下标准：

   - 对于您的账户和而言是唯一的 Amazon Web Services 区域
   - 以小写字母开头
   - 包含 3 到 32 个字符
   - 只包含小写字母 a-z、数字 0–9 和连字符（-）

5. 选择集合类型：

   - Time series（时间序列）：专注于分析机器生成的大量半结构化数据的日志分析部分。至少 24 小时的数据存储在热索引中，其余数据仍保留在温存储中。
   - Search（搜索）：全文搜索，支持内部网络中的应用程序和面向互联网的应用程序。所有搜索数据都将存储在热存储中，以确保快速查询响应时间。
   - 向量搜索 – 对简化向量数据管理的矢量嵌入进行语义搜索。支持机器学习（ML）增强搜索体验和生成式人工智能应用程序，例如聊天机器人、个人助理和欺诈检测。

   有关更多信息，请参阅 the section called “选择集合类型”。

6. 对于部署类型下，选择集合的冗余设置。默认情况下，每个集合都有冗余，因此索引和搜索 OpenSearch Compute Units（计算单元OCUs）在不同的可用区中都有自己的备用副本。对于开发和测试目的，您可以选择禁用冗余，从而将集合 OCUs 中的数量减少到两个。有关更多信息，请参阅 the section called “工作方式”。
7. 对于安全，选择标准创建。
8. 对于 E ncryption（加密），选择用于加密数据的密 Amazon KMS 钥。 OpenSearch 如果您输入的集合名称与加密策略中定义的模式相匹配，则 Serverless 将通知您。您可以选择保留此匹配项，也可以使用唯一的加密设置将其覆盖。有关更多信息，请参阅 the section called “加密”。
9. 对于网络访问设置，配置集合的网络访问权限。

   - 对于访问类型，选择公有或私有。

     如果选择私有的，请指定哪些 VPC 端点和 Amazon Web Services 服务 可以访问该集合。

     - 用于访问的 VPC 端点 – 指定一个或多个可用于访问的 VPC 端点。要创建 VPC 端点，请参阅the section called “VPC 端点”。
     - Amazon Web Services 服务 私有访问 — 选择一个或多个要允许访问的支持的服务。
   - 对于 Resource type（资源类型），选择用户是可以通过集合OpenSearch端点访问（通过 curl、Postman 等进行 API 调用）、可以通过控制OpenSearch 面板端点访问（通过控制台使用可视化和进行 API 调用），还是可以同时通过两者访问。

> **Note**
>
> Amazon Web Services 服务 私有访问仅适用于 OpenSearch端点，不适用于 OpenSearch 控制面板端点。

OpenSearch 如果您输入的集合名称与网络策略中定义的模式相匹配，则 Serverless 将通知您。您可以选择保留此匹配项，也可以使用自定义网络设置将其覆盖。有关更多信息，请参阅 the section called “网络访问”。

10. （可选）将一个或多个标签添加到集合。有关更多信息，请参阅 the section called “标记集合”。
11. 选择下一步。
12. 为集合配置数据访问规则，该规则定义谁可以访问集合中的数据。对于创建的每条规则，请执行以下步骤：

    - 选择 Add principals（添加主体），然后选择一个或多个 IAM 角色或 SAML 用户和组，授予其数据访问权限。
    - 在 Grant permissions（授予权限）下，选择要授予关联主体的别名、模板和索引权限。有关权限及其允许的访问权限的完整列表，请参阅 the section called “受支持 OpenSearch 的 API 操作和权限”。

    OpenSearch 如果您输入的集合名称与数据访问策略中定义的模式相匹配，则 Serverless 将通知您。您可以选择保留此匹配项，也可以使用唯一的数据访问设置将其覆盖。有关更多信息，请参阅 the section called “数据访问控制”。
13. 选择下一步。
14. 对于 Data access policy settings（数据访问策略设置），选择如何处理刚才创建的规则。您可以使用它们来创建新的数据访问策略，也可以将其添加到现有策略中。
15. 选择下一步。
16. 查看集合配置并选择 Submit（提交）。

在 OpenSearch 无服务器创建集合时，集合状态更改`Creating`为。

创建集合（CLI）

在使用创建集合之前 Amazon CLI，您必须拥有一个加密策略，其资源模式必须与集合的预期名称相匹配。例如，如果您计划将您的集合命名为 logs-application，则可以创建如下所示的加密策略：

```
aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource
\":[\"collection\/logs-application\"]}],\"AWSOwnedKey\":true}"
```

如果您计划将该策略用于其他集合，则可以扩大该规则的范围，如 `collection/logs*` 或 `collection/*`。

您还需要以网络策略的形式为集合配置网络设置。使用前面的 logs-application 示例，您可以创建以下网络策略：

```
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}]"
```

> **Note**
>
> 您可以在创建集合后创建网络策略，但我们建议您事先创建网络策略。

要创建收藏夹，CreateCollection请发送请求：

```
aws opensearchserverless create-collection --name "logs-application" --type SEARCH --
description "A collection for storing log data"
```

对于 `type`，请指定 `SEARCH` 或 `TIMESERIES`。有关更多信息，请参阅 the section called “选择集合类型”。

示例响应

```
{
    "createCollectionDetail": {
        "id": "07tjusf2h91cunochc",
        "name": "books",
        "description":"A collection for storing log data",
        "status": "CREATING",
```

```
        "type": "SEARCH",
        "kmsKeyArn": "auto",
        "arn": "arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
        "createdDate": 1665952577473
    }
}
```

如果您在该请求中未指定集合类型，则其默认为 `TIMESERIES`。如果您的集合使用 Amazon 拥有的密钥进行加密，则 `kmsKeyArn` 将是 `auto` 而非 ARN。

> **Important**
>
> 在创建集合后，除非该集合与某一数据访问策略相匹配，否则您将无法访问它。有关创建数据访问策略的说明，请参阅the section called “数据访问控制”。

访问 OpenSearch 仪表板

使用创建集合后 Amazon Web Services Management Console，您可以导航到该集合的 Dashboard OpenSearch s URL。您可以通过选择左侧导航窗格中的集合，然后选择该集合以打开其详细信息页面，来查找控制面板 URL。该 URL 采用 `https://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc` 格式。导航到 URL 后，将自动登录控制面板。

如果已有 OpenSearch 控制面板 URL 可用但未添加到 Amazon Web Services Management Console，则从浏览器调用控制面板 URL 将重定向到控制台。输入 Amazon 凭证后，将自动登录控制面板。有关访问 SAML 集合的信息，请参阅使用 SAML 访问 OpenSearch 控制面板。

控制 OpenSearch 面板控制台的超时时间为一小时且不可配置。

> **Note**
>
> 2023 年 5 月 10 日， OpenSearch 推出 OpenSearch 控制面板通用全局端点。现在，您可以使用采用该格式的 URL 在浏览器中导航到 OpenSearch 仪表板`https://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc`。为确保向后兼容性，我们将继续支持格式为现有集合特定 Dash OpenSearch boards 端点`https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/_dashboards`。

查看集合

您可以在 Amazon S OpenSearch ervice 控制台的 Collections（集合）选项卡 Amazon Web Services 账户 上，查看中的现有集合。

要列出收藏品及其收藏 IDs，请发送ListCollections请求。

```
aws opensearchserverless list-collections
```

示例响应

```
{
   "collectionSummaries":[
      {
         "arn":"arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "id":"07tjusf2h91cunochc",
         "name":"my-collection",
         "status":"CREATING"
      }
   ]
}
```

要限制搜索结果，请使用集合筛选器。此请求将筛选针对处于 `ACTIVE` 状态的集合的响应：

```
aws opensearchserverless list-collections --collection-filters '{ "status": "ACTIVE" }'
```

要获取有关一个或多个集合的更多详细信息，包括 OpenSearch 端点和 Dashboar OpenSearch ds 端点，请发送BatchGetCollection请求：

```
aws opensearchserverless batch-get-collection --ids ["07tjusf2h91cunochc",
 "1iu5usc4rame"]
```

> **Note**
>
> 您可以在该请求中包括 `--names` 或 `--ids`，但不能同时包括两者。

示例响应

```
{
   "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":[]
}
```

删除集合

删除集合将删除该集合中的所有数据和索引。删除集合后，您将无法恢复它们。

使用控制台删除集合

1. 在 Amazon S OpenSearch ervice 控制台的 Collections（集合）面板中，选择要删除的集合。

2. 选择 Delete（删除），然后确认删除。

要使用删除收藏夹 Amazon CLI，DeleteCollection请发送请求：

```
aws opensearchserverless delete-collection --id 07tjusf2h91cunochc
```

示例响应

```
{
   "deleteCollectionDetail":{
      "id":"07tjusf2h91cunochc",
      "name":"my-collection",
      "status":"DELETING"
   }
}
```

使用向量搜索集合

OpenSearch Serverless 中的向量搜索集合类型提供了可扩展且性能高的相似度搜索功能。它使您可以轻松构建现代机器学习 (ML) 增强搜索体验和生成式人工智能 (AI) 应用程序，而无需管理底层的向量数据库基础架构。

向量搜索集合的用例包括图像搜索、文档搜索、音乐检索、产品推荐、视频搜索、基于位置的搜索、欺诈检测和异常检测。

由 OpenSearch 于 Serverless 的矢量引擎由中的 k 最近邻 (k-nn) 搜索功能提供支持 OpenSearch，因此您可以通过无服务器环境的简单性获得相同的功能。该引擎支持 k-nn 插件 API。通过这些操作，您可以利用全文搜索、高级筛选、聚合、地理空间查询、嵌套查询来更快地检索数据，并增强搜索结果。

向量引擎提供距离指标，例如欧几里得距离、余弦相似度和点积相似度，并且可以容纳 16,000 个维度。您可以存储具有各种元数据数据类型的字段，例如数字、布尔值、日期、关键字和地理点。您还可以存储带有描述性信息的文本字段，以便为存储的向量添加更多上下文。将数据类型放在一起可以降低复杂性、提高可维护性，并避免数据重复、版本兼容性挑战和许可问题。

> **Note**
>
> Amazon OpenSearch Serverless 支持 Faiss 16 位标量量化，可用于在 32 位浮点向量和 16 位向量之间进行转换。要了解更多信息，请参阅 Faiss 16 位标量量化。您还可以使用二进制向量来降低内存成本。有关更多信息，请参阅二进制向量。

开始使用向量搜索集合

在本教程中，您将完成以下步骤来实时存储、搜索和检索向量嵌入：

1. 配置权限
2. 创建集合
3. 上传和搜索数据
4. 删除集合

步骤 1：配置权限

要完成本教程（以及一般使用 OpenSearch 无服务器），您必须拥有正确的 Amazon Identity and Access Management (IAM) 权限。在本教程中，您将创建一个集合、上传和搜索数据，然后删除该集合。

您的用户或角色必须已经附加基于身份的策略，并且具有以下最低权限：

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": "*"
    }
  ]
}
```

有关 OpenSearch 无服务器 IAM 权限的更多信息，请参阅the section called “身份和访问管理”。

步骤 2：创建集合

集合是一组 OpenSearch 索引，它们协同工作以支持特定的工作负载或用例。

创建 OpenSearch 无服务器集合

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择左侧导航窗格中的 Collections（集合），然后选择 Create collection（创建集合）。
3. 命名集合住房。
4. 对于集合类型，选择向量搜索。有关更多信息，请参阅 the section called “选择集合类型”。
5. 在部署类型下，清除启用冗余（活动副本）。这会在开发或测试模式下创建一个集合，并将集合中的 OpenSearch 计算单元 (OCUs) 数量减少到两个。如果要在本教程中创建生产环境，则请选中该复选框。
6. 在安全下，选择轻松创建以简化您的安全配置。默认情况下，向量引擎中的所有数据在传输过程中和静止状态下都经过加密。向量引擎支持精细的 IAM 权限，因此您可以定义谁可以创建、更新和删除加密、网络、集合和索引。
7. 选择 下一步。
8. 查看集合设置并选择 提交。稍等几分钟，等集合状态变为 `Active`。

步骤 3：上传并搜索数据

索引是具有通用数据架构的文档集合，它为您提供了一种存储、搜索和检索向量嵌入和其他字段的方法。您可以使用 OpenSearch 仪表板中的开发工具控制台或 Po stm an 或 awscurl 等 HTTP 工具，创建数据并将其上传到 OpenSearch 无服务器集合中的索引。本教程使用的是 Dev Tools。

索引和搜索“movies”（电影）集合中的数据

1. 要为新集合创建单个索引，请在 Dev Tools 控制台中发送以下请求。默认情况下，这将创建一个带有 `nmslib` 引擎和欧几里得距离的索引。

```
PUT housing-index
{
   "settings": {
      "index.knn": true
   },
   "mappings": {
```

```
      "properties": {
         "housing-vector": {
            "type": "knn_vector",
            "dimension": 3
         },
         "title": {
            "type": "text"
         },
         "price": {
            "type": "long"
         },
         "location": {
            "type": "geo_point"
         }
      }
   }
}
```

2. 要将单个文档索引到 housing-index 中，请发送以下请求：

```
POST housing-index/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. 要搜索与索引中的属性相似的属性，请发送以下查询：

```
GET housing-index/_search
{
    "size": 5,
    "query": {
        "knn": {
            "housing-vector": {
                "vector": [
                    10,
                    20,
```

```
                    30
                ],
                "k": 5
            }
        }
    }
}
```

步骤 4：删除集合

由于住房集合用于测试目的，因此请确保在您完成试验后将其删除。

删除 OpenSearch 无服务器集合

1. 返回亚马逊 OpenSearch 服务控制台。
2. 选择左侧导航窗格中的集合，然后选择属性集合。
3. 选择删除，然后确认删除。

经过筛选的搜索

您可以使用筛选条件来优化语义搜索结果。要创建索引并对文档执行经过筛选的搜索，请使用以下说明替换上一个教程中的上传和搜索数据。其他步骤保持不变。有关筛选条件的更多信息，请参阅使用筛选条件进行 k-NN 搜索。

索引和搜索“movies”（电影）集合中的数据

1. 要为您的集合创建单个索引，请在 Dev Tools 控制台中发送以下请求：

```
PUT housing-index-filtered
{
  "settings": {
    "index.knn": true
  },
  "mappings": {
    "properties": {
      "housing-vector": {
        "type": "knn_vector",
        "dimension": 3,
        "method": {
          "engine": "faiss",
```

```
          "name": "hnsw"
        }
      },
      "title": {
        "type": "text"
      },
      "price": {
        "type": "long"
      },
      "location": {
        "type": "geo_point"
      }
    }
  }
}
```

2. 要将单个文档编入索引 housing-index-filtered，请发送以下请求：

```
POST housing-index-filtered/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. 要搜索西雅图给定价格和给定距离范围内的公寓数据，请发送以下请求：

```
GET housing-index-filtered/_search
{
  "size": 5,
  "query": {
    "knn": {
      "housing-vector": {
        "vector": [
          0.1,
          0.2,
          0.3
        ],
```

```
        "k": 5,
        "filter": {
          "bool": {
            "must": [
              {
                "query_string": {
                  "query": "Find me 2 bedroom apartment in Seattle under $3000 ",
                  "fields": [
                    "title"
                  ]
                }
              },
              {
                "range": {
                  "price": {
                    "lte": 3000
                  }
                }
              },
              {
                "geo_distance": {
                  "distance": "100miles",
                  "location": {
                    "lat": 48,
                    "lon": 121
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}
```

十亿级工作负载

向量搜索集合支持包含数十亿个向量的工作负载。您无需出于扩展目的重新编制索引，因为自动扩缩功能会为您执行此操作。如果您有数百万个（或更多）具有大量维度的向量，并且需要超过 200 个 OCUs，请联系 Su Amazon pp ort 以提高您的账户的最大 OpenSearch 计算单位 (OCUs)。

限制

向量搜索集合具有以下限制：

- 向量搜索集合不支持 Apache Lucene ANN 引擎。
- 向量搜索集合仅支持带有 Faiss 的 HNSW 算法，不支持 IVF 和 IVFQ。
- 向量搜索集合不支持预热、统计数据和模型训练 API 操作。
- 向量搜索集合不支持内联脚本或存储脚本。
- 矢量搜索集合中 Amazon Web Services Management Console 没有索引计数信息。
- 向量搜索集合上索引的刷新间隔为 60 秒。

后续步骤

现在您已知道如何创建向量搜索集合和索引数据，您可能想尝试以下一些练习：

- 使用 OpenSearch Python 客户端处理矢量搜索集合。请参阅本教程GitHub。
- 使用 OpenSearch Java 客户端处理矢量搜索集合。请参阅本教程GitHub。
- 设置 LangChain 为 OpenSearch 用作矢量存储。 LangChain 是一个开源框架，用于开发由语言模型支持的应用程序。有关更多信息，请参阅 LangChain 文档。

在 Amazon OpenSearch Serverless 中使用数据生命周期策略

Amazon OpenSearch Serverless 中的数据生命周期策略定义了 S OpenSearch erverless 在时间序列集合中保留数据的时间。例如，您可以将策略设置为在 OpenSearch Serverless 删除日志数据之前将其保留 30 天。

您可以为每个时间序列集合中的每个索引配置单独的策略 Amazon Web Services 账户。 OpenSearch Serverless 将文档至少保留您在策略中指定的期限。然后，它会尽力自动删除文档，通常在 48 小时或保留期的 10% 内（以较长者为准）。

只有时间序列集合支持数据生命周期策略。搜索和矢量搜索集合不是。

主题

- 数据生命周期策略
- 所需的权限

- 策略优先顺序
- 策略语法
- 创建数据生命周期策略
- 更新数据生命周期策略
- 删除数据生命周期策略

数据生命周期策略

在数据生命周期策略中，您可以指定一系列规则。数据生命周期策略允许您管理与符合这些规则的索引或集合相关的数据的保留期。这些规则定义了索引或索引组中数据的保留期。每条规则都由资源类型 (`index`)、保留期和保留期适用的资源（索引）列表组成。

您可以使用以下格式之一定义保留期：

- `"MinIndexRetention": "24h"`— OpenSearch Serverless 以小时或天为单位保留指定时间段的索引数据。您可以将此时间段设置为从 `24h` 到 `3650d`。
- `"NoMinIndexRetention": true`— OpenSearch 无服务器无限期保留索引数据。

在以下示例策略中，第一条规则指定集合 `marketing` 内所有索引的保留期为 15 天。第二条规则规定，`finance` 集合中以 `log` 开头的所有索引名称均未设置保留期，并且将无限期保留。

```
{
   "lifeCyclePolicyDetail": {
      "type": "retention",
      "name": "my-policy",
      "policyVersion": "MTY4ODI0NTM2OTk1N18x",
      "policy": {
         "Rules": [
            {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/*"
            ],
            "MinIndexRetention": "15d"
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/finance/log*"
```

```
            ],
            "NoMinIndexRetention": true
         }
         ]
      },
      "createdDate": 1688245369957,
      "lastModifiedDate": 1688245369957
   }
}
```

在以下示例策略规则中， OpenSearch Serverless 无限期保留账户内所有集合的所有索引中的数据。

```
{
   "Rules": [
      {
         "ResourceType": "index",
         "Resource": [
            "index/*/*"
         ]
      }
   ],
   "NoMinIndexRetention": true
}
```

所需的权限

OpenSearch 无服务器的生命周期策略使用以下 Amazon Identity and Access Management (IAM) 权限。您可以指定 IAM 条件，以限制用户使用与具体集合和索引相关的生命周期策略。

- `aoss:CreateLifecyclePolicy` —创建生命周期策略。
- `aoss:ListLifecyclePolicies` — 列出当前账户中的所有数据生命周期策略。
- `aoss:BatchGetLifecyclePolicy` — 查看与账户或策略名称相关的数据生命周期策略。
- `aoss:BatchGetEffectiveLifecyclePolicy` — 查看给定资源（`index` 是唯一受支持的资源）的数据生命周期策略。
- `aoss:UpdateLifecyclePolicy` — 修改给定的数据生命周期策略，并更改其保留设置或资源。
- `aoss:DeleteLifecyclePolicy` — 删除数据生命周期策略。

以下基于身份的访问策略允许用户查看所有数据生命周期策略，并使用资源模式 `collection/application-logs` 更新策略：

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": "*"
        }
    ]
}
```

策略优先顺序

在某些情况下，数据生命周期策略规则可能会在策略内部或策略之间重叠。发生这种情况时，具有更具体的资源名称或模式的索引规则，将覆盖具有更通用的资源名称或模式的任何索引（对于两个规则通用）规则。

例如，在以下策略中，两个规则都适用于索引 `index/sales/logstash`。在这种情况下，第二条规则优先，因为 `index/sales/log*` 与 `index/sales/logstash` 匹配的时间最长。因此，OpenSearch Serverless 不为索引设置任何保留期。

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

```
        {
           "ResourceType":"index",
           "Resource":[
              "index/sales/*",
           ],
           "MinIndexRetention": "15d"
        },
        {
           "ResourceType":"index",
           "Resource":[
              "index/sales/log*",
           ],
           "NoMinIndexRetention": true
        }
     ]
  }
```

策略语法

提供一条或多条规则。这些规则定义了 OpenSearch 无服务器索引的数据生命周期设置。

每个规则包含以下元素。您可以在每条规则中提供 `MinIndexRetention` 或 `NoMinIndexRetention`，但不能两者兼而有之。

元素	描述
资源类型	该规则适用于的资源类型。数据生命周期策略唯一受支持的选项是 `index`。
资源	资源名称 and/or 模式列表。模式由前缀和通配符 (*) 组成，允许关联的权限应用于多个资源。例如 `index/<collection-name|pattern> /<index-name|pattern>` 。
MinIndexRetention	索引中文档的最短保留期限，以天 (d) 或小时 (h) 为单位。时间下限是 `24h`，上限是 `3650d`。
NoMinIndexRetention	如果是`true`，则 OpenSearch 无服务器会无限期保留文档。

在以下示例中，第一条规则适用于该autoparts-inventory模式 (index/autoparts-inventory/*) 下的所有索引，并要求在执行任何操作（例如删除或存档）之前，数据必须至少保留 20 天。

第二条规则针对与auto*/gear模式 (index/auto*/gear) 匹配的索引，将最小保留期设置为 24 小时。

第三条规则专门适用于tires索引，没有最低保留期限，这意味着可以立即删除或存档该索引中的数据，也可以根据其他标准进行删除。这些规则有助于管理保留时间各不相同或没有保留限制的索引数据的保留。

```
{
  "Rules": [
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/*"
      ],
      "MinIndexRetention": "20d"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/auto*/gear"
      ],
      "MinIndexRetention": "24h"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/tires"
      ],
      "NoMinIndexRetention": true
    }
  ]
}
```

创建数据生命周期策略

要创建数据生命周期策略，您需要根据指定的标准定义管理数据的保留和删除的规则。

控制台

创建数据生命周期策略

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择数据生命周期策略。
3. 选择创建数据生命周期策略。
4. 输入策略的描述性名称。
5. 对于数据生命周期，选择添加并选择策略的集合和索引。

   首先选择索引所属的集合。然后，要么从列表中选择索引，要么输入索引模式。要选择所有收藏作为来源，请输入星号 (*)。
6. 对于数据保留，您可以选择无限期保留数据，也可以取消选择 “无限制”（从不删除）并指定一个时间段，在此时间段之后， OpenSearch Serverless 会自动从 Amazon S3 中删除数据。
7. 选择 “保存”，然后选择 “创建”。

Amazon CLI

要使用创建数据生命周期策略 Amazon CLI，请使用带有以下选项的create-lifecycle-policy命令：

- `--name`— 策略的名称。
- `--type`— 策略的类型。当前，唯一可用的值是`retention`。
- `--policy`— 数据生命周期策略。此参数接受内联策略和.json 文件。必须将内联策略编码为 JSON 转义字符串。要在文件中提供策略，请使用格式`--policy file://my-policy.json`。

Example

```
aws opensearchserverless create-lifecycle-policy \
  --name my-policy \
  --type retention \
  --policy "{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":[\"index/autoparts-
inventory/*\"],\"MinIndexRetention\": \"81d\"},{\"ResourceType\":\"index\",\"Resource
\":[\"index/sales/orders*\"],\"NoMinIndexRetention\":true}]}"
```

更新数据生命周期策略

要更新数据生命周期策略，您可以修改现有规则以反映数据保留或删除要求的变化。这使您可以随着数据管理需求的变化调整策略。

在您更新策略和 OpenSearch Serverless 开始强制执行新的保留期之间可能会有几分钟的延迟。

控制台

更新数据生命周期策略

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择数据生命周期策略。
3. 选择要更新的数据生命周期策略，然后选择编辑。
4. 使用可视化编辑器或 JSON 编辑器修改策略。
5. 选择保存。

Amazon CLI

要使用更新数据生命周期策略 Amazon CLI，请使用update-lifecycle-policy命令。

您必须在请求中包含`--policy-version`参数。您可以使用 list-lifecycle-policies 或 batch-get-lifecycle-policy 命令检索策略版本。我们建议加入最新的策略版本，以防止意外覆盖他人所做的更改。

以下请求使用新的策略 JSON 文档更新数据生命周期策略。

Example

```
aws opensearchserverless update-lifecycle-policy \
  --name my-policy \
  --type retention \
  --policy-version MTY2MzY5MTY1MDA3Ml8x \
  --policy file://my-new-policy.json
```

删除数据生命周期策略

当您删除数据生命周期策略时， OpenSearch Serverless 将不再在任何匹配的索引上强制执行该策略。

控制台

删除数据生命周期策略

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择数据生命周期策略。
3. 选择要删除的策略，然后选择删除并确认删除。

Amazon CLI

要使用删除数据生命周期策略 Amazon CLI，请使用delete-lifecycle-policy命令。

Example

```
aws opensearchserverless delete-lifecycle-policy \
  --name my-policy \
  --type retention
```

使用与 Amazon OpenSearch 无服务器互动 Amazon SDKs

本节包含关于如何使用与 Amazon OpenSearch 无服务器 Amazon SDKs 进行交互的示例。这些代码示例演示如何创建安全策略和集合，以及如何查询集合。

> **Note**
> 我们目前正在扩建这些代码示例。如果您想贡献代码示例（Java、Go 等），请直接在GitHub存储库中打开拉取请求。

主题

- Python
- JavaScript

Python

以下示例脚本使用 适用于 Python (Boto3) 的 Amazon SDK 以及适用于 Python 的 opensearch-py 客户端来创建加密、网络和数据访问策略，创建相匹配的集合，以及索引某些示例数据。

要安装所需依赖项，请运行以下命令：

```
pip install opensearch-py
pip install boto3
pip install botocore
pip install requests-aws4auth
```

在脚本中，将 `Principal` 元素替换为签署请求的用户或角色的 Amazon 资源名称（ARN）。您也可以选择修改 `region`。

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3
import botocore
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

client = boto3.client('opensearchserverless')
service = 'aoss'
region = 'us-east-1'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def createEncryptionPolicy(client):
    """Creates an encryption policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Encryption policy for TV collections',
            name='tv-policy',
            policy="""
                {
                    \"Rules\":[
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[
                                \"collection\/tv-*\"
                            ]
                        }
                    ],
                    \"AWSOwnedKey\":true
```

```
                }
                """,
            type='encryption'
        )
        print('\nEncryption policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] The policy name or rules conflict with an existing
 policy.')
        else:
            raise error


def createNetworkPolicy(client):
    """Creates a network policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Network policy for TV collections',
            name='tv-policy',
            policy="""
                [{
                    \"Description\":\"Public access for TV collection\",
                    \"Rules\":[
                        {
                            \"ResourceType\":\"dashboard\",
                            \"Resource\":[\"collection\/tv-*\"]
                        },
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[\"collection\/tv-*\"]
                        }
                    ],
                    \"AllowFromPublic\":true
                }]
                """,
            type='network'
        )
        print('\nNetwork policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
```

```
                '[ConflictException] A network policy with this name already exists.')
        else:
            raise error


def createAccessPolicy(client):
    """Creates a data access policy that matches all collections beginning with tv-"""
    try:
        response = client.create_access_policy(
            description='Data access policy for TV collections',
            name='tv-policy',
            policy="""
                [{
                    \"Rules\":[
                        {
                            \"Resource\":[
                                \"index\/tv-*\/*\"
                            ],
                            \"Permission\":[
                                \"aoss:CreateIndex\",
                                \"aoss:DeleteIndex\",
                                \"aoss:UpdateIndex\",
                                \"aoss:DescribeIndex\",
                                \"aoss:ReadDocument\",
                                \"aoss:WriteDocument\"
                            ],
                            \"ResourceType\": \"index\"
                        },
                        {
                            \"Resource\":[
                                \"collection\/tv-*\"
                            ],
                            \"Permission\":[
                                \"aoss:CreateCollectionItems\"
                            ],
                            \"ResourceType\": \"collection\"
                        }
                    ],
                    \"Principal\":[
                        \"arn:aws:iam::123456789012:role\/Admin\"
                    ]
                }]
                """,
            type='data'
```

```
        )
        print('\nAccess policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] An access policy with this name already exists.')
        else:
            raise error


def createCollection(client):
    """Creates a collection"""
    try:
        response = client.create_collection(
            name='tv-sitcoms',
            type='SEARCH'
        )
        return(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] A collection with this name already exists. Try
 another name.')
        else:
            raise error


def waitForCollectionCreation(client):
    """Waits for the collection to become active"""
    response = client.batch_get_collection(
        names=['tv-sitcoms'])
    # Periodically check collection status
    while (response['collectionDetails'][0]['status']) == 'CREATING':
        print('Creating collection...')
        time.sleep(30)
        response = client.batch_get_collection(
            names=['tv-sitcoms'])
    print('\nCollection successfully created:')
    print(response["collectionDetails"])
    # Extract the collection endpoint from the response
    host = (response['collectionDetails'][0]['collectionEndpoint'])
    final_host = host.replace("https://", "")
    indexData(final_host)
```

```
def indexData(host):
    """Create an index and add some sample data"""
    # Build the OpenSearch client
    client = OpenSearch(
        hosts=[{'host': host, 'port': 443}],
        http_auth=awsauth,
        use_ssl=True,
        verify_certs=True,
        connection_class=RequestsHttpConnection,
        timeout=300
    )
    # It can take up to a minute for data access rules to be enforced
    time.sleep(45)

    # Create index
    response = client.indices.create('sitcoms-eighties')
    print('\nCreating index:')
    print(response)

    # Add a document to the index.
    response = client.index(
        index='sitcoms-eighties',
        body={
            'title': 'Seinfeld',
            'creator': 'Larry David',
            'year': 1989
        },
        id='1',
    )
    print('\nDocument added:')
    print(response)


def main():
    createEncryptionPolicy(client)
    createNetworkPolicy(client)
    createAccessPolicy(client)
    createCollection(client)
    waitForCollectionCreation(client)


if __name__ == "__main__":
```

```
    main()
```

JavaScript

以下示例脚本使用适用于 Node.js JavaScript 中的 SDK 以及适用于 opensearch-js 客户端来创建加密、网络和数据访问策略、创建相匹配的集合、创建索引，以及索引某些示例数据。 JavaScript

要安装所需依赖项，请运行以下命令：

```
npm i aws-sdk
npm i aws4
npm i @opensearch-project/opensearch
```

在脚本中，将 `Principal` 元素替换为签署请求的用户或角色的 Amazon 资源名称（ARN）。您也可以选择修改 `region`。

```
var AWS = require('aws-sdk');
var aws4 = require('aws4');
var {
    Client,
    Connection
} = require("@opensearch-project/opensearch");
var {
    OpenSearchServerlessClient,
    CreateSecurityPolicyCommand,
    CreateAccessPolicyCommand,
    CreateCollectionCommand,
    BatchGetCollectionCommand
} = require("@aws-sdk/client-opensearchserverless");
var client = new OpenSearchServerlessClient();

async function execute() {
    await createEncryptionPolicy(client)
    await createNetworkPolicy(client)
    await createAccessPolicy(client)
    await createCollection(client)
    await waitForCollectionCreation(client)
}

async function createEncryptionPolicy(client) {
    // Creates an encryption policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
```

```
            description: 'Encryption policy for TV collections',
            name: 'tv-policy',
            type: 'encryption',
            policy: " \
        { \
            \"Rules\":[ \
                { \
                    \"ResourceType\":\"collection\", \
                    \"Resource\":[ \
                        \"collection\/tv-*\" \
                    ] \
                } \
            ], \
            \"AWSOwnedKey\":true \
        }"
        });
        const response = await client.send(command);
        console.log("Encryption policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] The policy name or rules conflict with an
 existing policy.');
        } else
            console.error(error);
    };
}

async function createNetworkPolicy(client) {
    // Creates a network policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
            description: 'Network policy for TV collections',
            name: 'tv-policy',
            type: 'network',
            policy: " \
            [{ \
                \"Description\":\"Public access for television collection\", \
                \"Rules\":[ \
                    { \
                        \"ResourceType\":\"dashboard\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    }, \
                    { \
```

```
                        \"ResourceType\":\"collection\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    } \
                ], \
                \"AllowFromPublic\":true \
            }]"
        });
        const response = await client.send(command);
        console.log("Network policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] A network policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createAccessPolicy(client) {
    // Creates a data access policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateAccessPolicyCommand({
            description: 'Data access policy for TV collections',
            name: 'tv-policy',
            type: 'data',
            policy: " \
            [{ \
                \"Rules\":[ \
                    { \
                        \"Resource\":[ \
                            \"index\/tv-*\/*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateIndex\", \
                            \"aoss:DeleteIndex\", \
                            \"aoss:UpdateIndex\", \
                            \"aoss:DescribeIndex\", \
                            \"aoss:ReadDocument\", \
                            \"aoss:WriteDocument\" \
                        ], \
                        \"ResourceType\": \"index\" \
                    }, \
                    { \
```

```
                        \"Resource\":[ \
                            \"collection\/tv-*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateCollectionItems\" \
                        ], \
                        \"ResourceType\": \"collection\" \
                    } \
                ], \
                \"Principal\":[ \
                    \"arn:aws:iam::123456789012:role\/Admin\" \
                ] \
            }]"
        });
        const response = await client.send(command);
        console.log("Access policy created:");
        console.log(response['accessPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] An access policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createCollection(client) {
    // Creates a collection to hold TV sitcoms indexes
    try {
        var command = new CreateCollectionCommand({
            name: 'tv-sitcoms',
            type: 'SEARCH'
        });
        const response = await client.send(command);
        return (response)
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] A collection with this name already
 exists. Try another name.');
        } else
            console.error(error);
    };
}
```

```
async function waitForCollectionCreation(client) {
    // Waits for the collection to become active
    try {
        var command = new BatchGetCollectionCommand({
            names: ['tv-sitcoms']
        });
        var response = await client.send(command);
        while (response.collectionDetails[0]['status'] == 'CREATING') {
            console.log('Creating collection...')
            await sleep(30000) // Wait for 30 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        console.log('Collection successfully created:');
        console.log(response['collectionDetails']);
        // Extract the collection endpoint from the response
        var host = (response.collectionDetails[0]['collectionEndpoint'])
        // Pass collection endpoint to index document request
        indexDocument(host)
    } catch (error) {
        console.error(error);
    };
}

async function indexDocument(host) {

    var client = new Client({
        node: host,
        Connection: class extends Connection {
            buildRequestObject(params) {
                var request = super.buildRequestObject(params)
                request.service = 'aoss';
                request.region = 'us-east-1'; // e.g. us-east-1
                var body = request.body;
                request.body = undefined;
                delete request.headers['content-length'];
                request.headers['x-amz-content-sha256'] = 'UNSIGNED-PAYLOAD';
                request = aws4.sign(request, AWS.config.credentials);
                request.body = body;
```

```
                return request
            }
        }
    });

    // Create an index
    try {
        var index_name = "sitcoms-eighties";

        var response = await client.indices.create({
            index: index_name
        });

        console.log("Creating index:");
        console.log(response.body);

        // Add a document to the index
        var document = "{ \"title\": \"Seinfeld\", \"creator\": \"Larry David\", \"year
\": \"1989\" }\n";

        var response = await client.index({
            index: index_name,
            body: document
        });

        console.log("Adding document:");
        console.log(response.body);
    } catch (error) {
        console.error(error);
    };
}

execute()
```

Amazon CloudFormation 用于创建 Amazon OpenSearch 无服务器集合

您可以使用 Amazon CloudFormation 创建 Amazon OpenSearch 无服务器资源，如集合、安全策略和 VPC 端点。有关全面的 OpenSearch 无服务器 CloudFormation 参考，请参阅《Amazon CloudFormation 用户指南》中的 Amazon OpenSearch Serverless。

以下示例： CloudFormation 模板将创建简单的数据访问策略、网络策略和安全策略，以及相匹配的集合。这是使 Amazon OpenSearch 无服务器快速启动和运行以及预调配创建和使用集合所需元素的好方法。

> **Important**
>
> 此示例使用公共网络访问权限，建议不要将其用于生产工作负载。我们建议使用 VPC 访问来保护您的集合。有关更多信息，请参阅AWS::OpenSearchServerless::VpcEndpoint和the section called “VPC 端点”。

```
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
```

管理 Amazon OpenSearch Serverless 的容量限制

有了 Amazon OpenSearch Serverless，您不必自己管理容量。 OpenSearch Serverless 会根据当前的工作负载自动扩展您账户的计算容量。无服务器计算容量以OpenSearch 计算单位 (OCUs) 来衡量。每个 OCU 是 6GiB 内存和相应的虚拟 CPU（vCPU）以及创建到 Amazon S3 的数据管道的组合。有关 OpenSearch Serverless 中分离架构的更多信息，请参阅。the section called “工作方式”

当您创建第一个集合时， OpenSearch Serverless 会 OCUs 根据您的冗余设置进行实例化。默认情况下，冗余活动副本处于启用状态，这意味着总共实例化了四个 OCUs 副本（两个用于索引，两个用于搜索），以确保备用节点位于另一个可用区域的高可用性。出于开发和测试目的，您可以禁用集合的启用冗余设置，这将消除备用副本，只实例化两个副本 OCUs （一个用于索引，一个用于搜索）。即使没有索引或搜索活动，它们也 OCUs 始终存在。所有后续集合都可以共享这些集合 OCUs （具有唯一 Amazon KMS 密钥的集合除外，这些集合会实例化自己的集合）。 OCUs如果需要，随着索引和搜索使用量的增长， OpenSearch Serverles OCUs s 会自动扩展并添加更多内容。当您的集合终端节点上的流量减少时，容量会缩小到您的数据大小 OCUs 所需的最小数量。对于搜索和时间序列集合，空闲时 OCUs 所需的数量与数据大小和索引计数成正比。对于向量，存储矢量图取决于内存 (RAM) 和存储索引的磁盘空间。如果未处于空闲状态，OCU 要求会同时考虑这两个因素。

向量集合将索引数据保存在OCU本地存储中。达到 OCU RAM 限制的速度比 OCU 磁盘限制的速度快，从而导致矢量集合受到 RAM 空间的限制。最多会缩减至 1 个 OCU [0.5 个 OCU x 2] 用于索引，和 1 个 OCU [0.5 个 OCU x 2] 用于搜索。扩展还会影响您的集合或索引所需的分片数量。每个 OCU

可以支持指定数量的分片。索引的数量应与分片计数成正比。 OCUs 所需的基础总数是所需的最大数据、内存和分片量。有关更多信息，请参阅Amazon 大数据博客上任何规模的 Amazon OpenSearch Serverless 经济实惠的搜索功能。

对于搜索和向量搜索集合，所有数据都将存储在热索引中，以确保快速查询响应时间。时间序列集合使用热存储和温存储的组合，在热存储中保留最新数据，以优化访问频率更高的数据的查询响应时间。有关更多信息，请参阅 the section called “选择集合类型”。

> **Note**
>
> 矢量搜索集合无法 OCUs 与搜索集合和时间序列集合共享，即使矢量搜索集合使用与搜索集合或时间序列集合相同的 KMS 密钥。 OCUs 将为您的第一个矢量集合创建一组新的矢量图集。向 OCUs 量集合在相同的 KMS 密钥集合之间共享。

为了管理馆藏容量和控制成本，您可以为当前账户和区域指定索引和搜索的总体最大容量，OpenSearch Serverless 会根据这些规格自动扩展您的馆藏资源。

由于索引和搜索容量单独扩缩，因此您可以为每种容量指定账户级限制：

- 最大索引容量 — OpenSearch Serverless 可以将索引容量增加到这个数量。 OCUs
- 最大搜索容量 — OpenSearch Serverless可以将搜索容量增加到此数量. OCUs

> **Note**
>
> 目前，容量设置仅应用于账户级。您无法配置每个集合的容量限制。

您的目标是确保最大容量足以处理工作负载峰值。根据您的设置， OpenSearch Serverless 会自动扩展馆藏 OCUs 的数量，以处理索引和搜索工作量。

主题

- 配置容量设置
- 最大容量限制
- 监控容量使用情况

配置容量设置

要在 OpenSearch 无服务器控制台中配置容量设置，请在左侧导航窗格中展开无服务器，然后选择控制面板。在 Capacity management（容量管理）下，指定最大索引和搜索容量：

要使用配置容量 Amazon CLI，UpdateAccountSettings请发送请求：

```
aws opensearchserverless update-account-settings \
    --capacity-limits '{ "maxIndexingCapacityInOCU": 8,"maxSearchCapacityInOCU": 9 }'
```

最大容量限制

一个集合可以包含的最大索引总数为 1000。对于所有三种类型的馆藏，用于索引的默认最大 OCU 容量为 10 OCUs ，搜索的最大 OCU 容量 OCUs 为 10。一个账户允许的最小 OCU 容量为 1 个 OCU [0.5 OCU x 2] 用于索引，1 个 OCU [0.5 OCU x 2] 用于搜索。对于所有馆藏，允许的最大索引容量为 1,700， OCUs 搜索的最大允许容量为 1,700 OCUs 。您可以按 2 的整数倍，将 OCU 计数配置为从 1 到允许的最大容量之间的任意数字。

每个 OCU 都包含足够的临时热存储空间，可存放 120 GiB 的索引数据。 OpenSearch 在搜索和矢量搜索集合中，Serverless 支持每个索引最多 1 TiB 的数据，在时间序列集合中，每个索引最多支持 100 TiB 的热门数据。对于时间序列集合，您仍然可以摄取更多数据，这些数据可作为暖数据存储在 S3 中。

有关所有配额的列表，请参阅OpenSearch 无服务器配额。

监控容量使用情况

您可以监控`SearchOCU`和`IndexingOCU`账户级别的 CloudWatch 指标，以了解您的收款规模是如何扩展的。建议您配置警报，以便在您的账户接近与容量相关的指标阈值时通知您，使您能够相应调整容量设置。

您还可以使用这些指标确定您的最大容量设置是否合适，或者是否需要调整它们。分析这些指标，以便您将精力集中在优化集合效率上。有关 OpenSearch Serverless 发送到的指标的更多信息 CloudWatch，请参阅the section called “监控 OpenSearch 无服务器”。

将数据摄取到 Ama OpenSearch zon 无服务器集合中

这些章节详细介绍了支持将数据摄取到 Ama OpenSearch zon 无服务器集合中的摄取管线。它们还介绍了一些您可用于与 OpenSearch API 操作进行交互的客户端。您的客户端应与 OpenSearch 2.x 兼容，才能与 OpenSearch Serverless 集成。

主题

- 所需的最低权限
- OpenSearch 摄取
- Fluent Bit
- Amazon Data Firehose
- Go
- Java
- JavaScript
- Logstash
- Python
- Ruby
- 与其他客户端签署 HTTP 请求

所需的最低权限

为将数据摄取到 S OpenSearch erverless 集合中，必须在数据访问策略中为写入数据的主体分配以下最低权限：

```
[
    {
```

```
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/target-collection/logs"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:WriteDocument",
               "aoss:UpdateIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

如果您计划写入到其他索引，则权限可以更广。例如，您可以允许针对所有索引 (index/ /*) 或索引子集 (index/ *target-collection* /*) 的权限，而不是指定单个目标索引。*target-collection logs**

有关所有可用 OpenSearch API 操作及其相关权限的参考，请参阅the section called “受支持的操作和插件”。

OpenSearch 摄取

您可以使用 Amazon OpenSearch Ingestion 将数据直接发送到 OpenSearch 无服务器集合。您可以配置数据生产者，将数据发送到 OpenSearch Ingestion，后者自动将数据传输到您指定的集合。您还可以配置 OpenSearch Ingestion 以在传输数据之前转换数据。有关更多信息，请参阅 Amazon OpenSearch Ingestion。

OpenSearch 摄取管道需要权限才能写入配置为其接收器的 S OpenSearch erver无服务器集合。这些权限包括能够描述集合以及向其发送 HTTP 请求。有关使用 OpenSearch Ingestion 向集合添加数据的说明，请参阅。the section called “向管道授予访问集合的权限”

要开始使用 OpenSearch Ingestion，请参阅。the section called “教程：将数据摄取到集合”

Fluent Bit

你可以使用 F Amazon or Fluent Bit 图像和OpenSearch 输出插件将数据提取到 OpenSearch 无服务器集合中。

> **Note**
> 您必须有 2.30.0 或更高版本的 Amazon 适用于 Fluent Bit 映像的才能与 Serverless 集成。OpenSearch

示例配置：

配置文件的此示例输出部分显示了如何使用 OpenSearch 无服务器集合作为目标。添加 `AWS_Service_Name` 参数（即 `aoss`）十分重要。`Host` 是集合端点。

```
[OUTPUT]
    Name  opensearch
    Match *
    Host  collection-endpoint.us-west-2.aoss.amazonaws.com
    Port  443
    Index  my_index
    Trace_Error On
    Trace_Output On
    AWS_Auth On
    AWS_Region <region>
    AWS_Service_Name aoss
    tls     On
    Suppress_Type_Name On
```

Amazon Data Firehose

Firehose 支持将 OpenSearch 无服务器作为传输目标。有关将数据发送到 OpenSearch 无服务器的说明，请参阅《亚马逊数据 Fireh ose 开发者指南》中的创建 Kinesis Data Firehose 传送流并 OpenSearch 选择无服务器作为目的地。

您提供给 Firehose 用于传输的 IAM 角色，必须在具有目标集合的 `aoss:WriteDocument` 最低权限的数据访问策略中指定，并且您必须具有预先存在的索引以向其发送数据。有关更多信息，请参阅 the section called “所需的最低权限”。

在将数据发送到 OpenSearch Serverless 之前，您可能需要对数据执行转换。要了解有关使用 Lambda 函数执行此任务的更多信息，请参阅此同一指南中的 Amazon Kinesis Data Firehose 数据转换。

Go

以下示例代码使用适用于 Go 的 opensearch-g o 客户端与指定的 OpenSearch Server无服务器集合建立安全连接，并创建单个索引。必须提供 `region` 和 `host` 的值。

```
package main

import (
  "context"
  "log"
  "strings"
  "github.com/aws/aws-sdk-go-v2/aws"
  "github.com/aws/aws-sdk-go-v2/config"
  opensearch "github.com/opensearch-project/opensearch-go/v2"
  opensearchapi "github.com/opensearch-project/opensearch-go/v2/opensearchapi"
  requestsigner "github.com/opensearch-project/opensearch-go/v2/signer/awsv2"
)

const endpoint = "" // serverless collection endpoint

func main() {
 ctx := context.Background()

 awsCfg, err := config.LoadDefaultConfig(ctx,
  config.WithRegion("<AWS_REGION>"),
  config.WithCredentialsProvider(
   getCredentialProvider("<AWS_ACCESS_KEY>", "<AWS_SECRET_ACCESS_KEY>",
 "<AWS_SESSION_TOKEN>"),
  ),
 )
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an AWS request Signer and load AWS configuration using default config folder
 or env vars.
 signer, err := requestsigner.NewSignerWithService(awsCfg, "aoss") // "aoss" for Amazon
 OpenSearch Serverless
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an opensearch client and use the request-signer
```

```
 client, err := opensearch.NewClient(opensearch.Config{
  Addresses: []string{endpoint},
  Signer:    signer,
 })
 if err != nil {
  log.Fatal("client creation err", err)
 }

 indexName := "go-test-index"

  // define index mapping
 mapping := strings.NewReader(`{
  "settings": {
    "index": {
         "number_of_shards": 4
         }
       }
  }`)

 // create an index
 createIndex := opensearchapi.IndicesCreateRequest{
  Index: indexName,
    Body: mapping,
 }
 createIndexResponse, err := createIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("Error ", err.Error())
  log.Println("failed to create index ", err)
  log.Fatal("create response body read err", err)
 }
 log.Println(createIndexResponse)

 // delete the index
 deleteIndex := opensearchapi.IndicesDeleteRequest{
  Index: []string{indexName},
 }

 deleteIndexResponse, err := deleteIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("failed to delete index ", err)
  log.Fatal("delete index response body read err", err)
 }
 log.Println("deleting index", deleteIndexResponse)
}
```

```
func getCredentialProvider(accessKey, secretAccessKey, token string)
 aws.CredentialsProviderFunc {
 return func(ctx context.Context) (aws.Credentials, error) {
  c := &aws.Credentials{
   AccessKeyID:     accessKey,
   SecretAccessKey: secretAccessKey,
   SessionToken:    token,
  }
  return *c, nil
 }
}
```

Java

以下示例代码使用适用于 Java 的 opensearch- java 客户端与指定 OpenSearch 无服务器集合建立安全连接，并创建单个索引。必须提供 `region` 和 `host` 的值。

与 OpenSearch 服务域相比，重要的区别在于服务名称（`aoss`而不是`es`）。

```
// import OpenSearchClient to establish connection to OpenSearch Serverless collection
import org.opensearch.client.opensearch.OpenSearchClient;

SdkHttpClient httpClient = ApacheHttpClient.builder().build();
// create an opensearch client and use the request-signer
OpenSearchClient client = new OpenSearchClient(
    new AwsSdk2Transport(
        httpClient,
        "...us-west-2.aoss.amazonaws.com", // serverless collection endpoint
        "aoss" // signing service name
        Region.US_WEST_2, // signing service region
        AwsSdk2TransportOptions.builder().build()
    )
);

String index = "sample-index";

// create an index
CreateIndexRequest createIndexRequest = new
 CreateIndexRequest.Builder().index(index).build();
CreateIndexResponse createIndexResponse = client.indices().create(createIndexRequest);
System.out.println("Create index reponse: " + createIndexResponse);
```

```
// delete the index
DeleteIndexRequest deleteIndexRequest = new
 DeleteIndexRequest.Builder().index(index).build();
DeleteIndexResponse deleteIndexResponse = client.indices().delete(deleteIndexRequest);
System.out.println("Delete index reponse: " + deleteIndexResponse);

httpClient.close();
```

以下示例代码会重新建立安全连接，然后搜索索引。

```
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

以下示例代码使用适用 JavaScript 于的 opensearch-js 客户端，建立与指定的 OpenSearch Server无服务器集合的安全连接、创建一个索引、添加文档，并搜索索引。必须提供 `node` 和 `region` 的值。

与 OpenSearch 服务域相比，重要的区别在于服务名称（`aoss`而不是`es`）。

Version 3

此示例使用适用于 SDK JavaScript in Node.js 的 SDK 版本 3。

```
const { defaultProvider } = require('@aws-sdk/credential-provider-node');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () => {
                const credentialsProvider = defaultProvider();
                return credentialsProvider();
            },
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index })).body) {
        console.log((await client.indices.create({ index })).body);
    }

    // add a document to the index
    const document = { foo: 'bar' };
    const response = await client.index({
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}
```

```
main();
```

Version 2

此示例使用了 Node.js JavaScript 中开发工具包的版本 2。

```
const AWS = require('aws-sdk');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () =>
                new Promise((resolve, reject) => {
                    AWS.config.getCredentials((err, credentials) => {
                        if (err) {
                            reject(err);
                        } else {
                            resolve(credentials);
                        }
                    });
                }),
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index })).body) {
        console.log((await client.indices.create({
            index
        })).body);
    }

    // add a document to the index
    const document = {
        foo: 'bar'
    };
    const response = await client.index({
```

```
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}

main();
```

Logstash

您可以使用 Logstash OpenSearch 插件，将日志发布到 OpenSearch 无服务器集合。

要使用 Logstash 将数据发送到无服务器 OpenSearch

1. 使用 Docker 或 Linux 安装该logstash-output-opensearch插件的 2.0.0 或更高版本。

   Docker

   Docker 将托管已预安装 OpenSearch 输出插件的 Logstash OSS 软件：opensearchproject/-output-plugin。logstash-oss-with-opensearch您可以像任何其他映像一样拉取该映像：

   ```
   docker pull opensearchproject/logstash-oss-with-opensearch-output-plugin:latest
   ```

   Linux

   首先，请安装最新版本的 Logstash（如果您尚未这样做）。然后，安装版本 2.0.0 的输出插件：

   ```
   cd logstash-8.5.0/
   bin/logstash-plugin install --version 2.0.0 logstash-output-opensearch
   ```

   如果已安装该插件，请将其更新到最新版本：

   ```
   bin/logstash-plugin update logstash-output-opensearch
   ```

从该插件的版本 2.0.0 开始， Amazon SDK 使用版本 3。如果您使用的是 8.4.0 之前的 Logstash 版本，则必须移除所有预安装的插件，然后安装该 Amazon 插件：`logstash-integration-aws`

```
/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. 为使 OpenSearch 输出插件能与 OpenSearch Serverless 配合使用，您必须对 logstash.conf 的`opensearch`输出部分进行以下修改：

   - 在 `auth_type` 下，将 `aoss` 指定为 `service_name`。
   - 为 `hosts` 指定您的集合端点。
   - 添加参数 `default_server_major_version` 和 `legacy_template`。这些参数是该插件与 OpenSearch Server无服务器配合使用所必需的。

```
output {
  opensearch {
    hosts => "collection-endpoint:443"
    auth_type => {
      ...
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

此示例配置文件从 S3 存储桶中的文件获取其输入，然后将它们发送到 S OpenSearch erverless 集合：

```
input {
  s3  {
```

```
    bucket => "my-s3-bucket"
    region => "us-east-1"
  }
}

output {
  opensearch {
    ecs_compatibility => disabled
    hosts => "https://my-collection-endpoint.us-east-1.aoss.amazonaws.com:443"
    index => my-index
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

3. 然后，使用新配置运行 Logstash，以测试该插件：

```
bin/logstash -f config/test-plugin.conf
```

Python

以下示例代码使用适用于 Python 的 opensearch-py 客户端，建立与指定 OpenSearch 无服务器集合的安全连接、创建一个索引，以及搜索该索引。必须提供 `region` 和 `host` 的值。

与 OpenSearch 服务域相比，重要的区别在于服务名称（`aoss`而不是`es`）。

```
from opensearchpy import OpenSearch, RequestsHttpConnection, AWSV4SignerAuth
import boto3

host = ''  # serverless collection endpoint, without https://
region = ''  # e.g. us-east-1

service = 'aoss'
credentials = boto3.Session().get_credentials()
auth = AWSV4SignerAuth(credentials, region, service)
```

```
# create an opensearch client and use the request-signer
client = OpenSearch(
    hosts=[{'host': host, 'port': 443}],
    http_auth=auth,
    use_ssl=True,
    verify_certs=True,
    connection_class=RequestsHttpConnection,
    pool_maxsize=20,
)

# create an index
index_name = 'books-index'
create_response = client.indices.create(
    index_name
)

print('\nCreating index:')
print(create_response)

# index a document
document = {
  'title': 'The Green Mile',
  'director': 'Stephen King',
  'year': '1996'
}

response = client.index(
    index = 'books-index',
    body = document,
    id = '1'
)


# delete the index
delete_response = client.indices.delete(
    index_name
)

print('\nDeleting index:')
print(delete_response)
```

Ruby

`opensearch-aws-sigv4`Gem 提供对 OpenSearch Serverless 以及Serv OpenSearch ice 的即时访问权限。它具有 opensearch-ruby 客户端的所有功能，因为它是这款 Gem 的依赖项。

在实例化 Sigv4 签名程序时，指定 `aoss` 为服务名称：

```
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)
```

与其他客户端签署 HTTP 请求

如果与其他客户端构建 HTTP 请求，在对 OpenSearch Serverless 集合签署请求时，以下要求适用。

- 必须将服务名称指定为 `aoss`。
- 所有 Amazon 签名版本 4 请求都需要 `x-amz-content-sha256` 标头。它将提供请求负载的哈希。如果有请求负载，请将该值设置为其安全哈希算法 (SHA) 加密哈希 (SHA256)。如果没有请求负载，请将该值设置为 `e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855`，它是空字符串的哈希。

主题

- 使用 cURL 创建索引
- 使用 Postman 创建索引

使用 cURL 创建索引

以下示例请求使用客户端 URL 请求库（cURL）将单个文档发送到集合中名为 `movies-index` 的索引：

```
curl -XPOST \
    --user "$AWS_ACCESS_KEY_ID":"$AWS_SECRET_ACCESS_KEY" \
    --aws-sigv4 "aws:amz:us-east-1:aoss" \
    --header "x-amz-content-sha256: $REQUEST_PAYLOAD_SHA_HASH" \
    --header "x-amz-security-token: $AWS_SESSION_TOKEN" \
    "https://my-collection-endpoint.us-east-1.aoss.amazonaws.com/movies-index/_doc" \
    -H "Content-Type: application/json" -d '{"title": "Shawshank Redemption"}'
```

使用 Postman 创建索引

下图演示了如何使用 Postman 将请求发送到集合。有关身份验证的说明，请参阅 Authentic Amazon ation with Signature with Sig

Amazon OpenSearch Serverless 中的安全概述

Amazon OpenSearch Serverless 中的安全性与亚马逊 OpenSearch 服务中的安全性有根本的区别，具体体现在以下方面：

功能	OpenSearch 服务	OpenSearch 无服务器
数据访问控制	数据访问权限由 IAM policy 和精细访问控制决定。	数据访问权限由数据访问策略决定。
静态加密	对于域，静态加密是可选的。	对于集合，静态加密是必需的。
安全设置和管理	必须为每个域单独配置网络、加密和数据访问权限。	您可以使用安全策略大规模管理多个集合的安全设置。

下图说明了构成功能集合的安全组件。集合必须具有已分配的加密密钥、网络访问权限设置，以及相匹配的数据访问策略，该策略授予针对其资源的权限。

主题

- 加密策略
- 网络策略
- 数据访问策略
- IAM 和 SAML 身份验证
- 基础结构安全性
- 开始使用 Amazon OpenSearch Serverless 中的安全功能
- 适用于 Amazon OpenSearch Serverless 的身份和访问管理
- Amazon OpenSearch 无服务器中的加密
- Amazon OpenSearch Serverless 的网络访问
- Amazon OpenSearch 无服务器的数据访问控制
- 使用接口终端节点访问 Amazon OpenSearch Serverless ()Amazon PrivateLink
- 适用于 Amazon 无服务器的 SAM OpenSearch L 身份验证
- Amazon OpenSearch 无服务器的合规性验证

加密策略

加密策略定义了您的馆藏是使用还是使用客户托管密钥进行加密。 Amazon 拥有的密钥 加密策略由两个组件组成：资源模式和加密密钥。资源模式将定义该策略适用于哪个或哪些集合。加密密钥决定如何保护关联的集合。

要将一个策略应用于多个集合，请在策略规则中包含通配符 (*)。例如，以下策略适用于名称以“logs”（日志）开头的所有集合。

加密策略可以简化创建和管理集合的过程，尤其是当您以编程方式创建和管理集合时。您只需指定名称即可创建集合，并在创建时自动为其分配加密密钥。

网络策略

网络策略定义您的集合是必须以私有方式访问，还是可以通过互联网从公共网络访问。私有馆藏可以通过 OpenSearch 无服务器托管的 VPC 终端节点进行访问，也可以通过诸如 Amazon Bedrock Amazon Web Services 服务 之类的 Amazon Web Services 服务 特定终端节点使用私有访问权限进行访问。就像加密策略一样，网络策略也可以应用于多个集合，这使您可以大规模管理很多集合的网络访问权限。

网络策略由两个组件组成：访问权限类型和资源类型。访问类型可以是公有也可以是私有。资源类型决定了您选择的访问权限是适用于集合端点、 OpenSearch 仪表板端点还是两者兼而有之。

如果您计划在网络策略中配置 VPC 访问权限，则必须先创建一个或多个OpenSearch 无服务器托管的 VPC 终端节点。这些终端节点允许您像访问您的 VPC 一样访问 OpenSearch 无服务器，无需使用互联网网关、NAT 设备、VPN 连接或 Amazon Direct Connect 连接。

对的私有访问权限 Amazon Web Services 服务 只能应用于集合的 OpenSearch 端点，不能应用于 OpenSearch 仪表板端点。 Amazon Web Services 服务 无法被授予对 OpenSearch 仪表板的访问权限。

数据访问策略

数据访问策略定义您的用户如何访问您的集合中的数据。数据访问策略通过自动为匹配特定模式的集合和索引分配访问权限，帮助您大规模管理集合。多个策略可应用于一个资源。

数据访问策略由一组规则组成，每条规则包含三个组件：资源类型、授予的资源和一组权限。资源类型可以是集合或索引。授予的资源可以是带有通配符 (*) 的 collection/index 名称或模式。权限列表指定了策略向哪OpenSearch 些 API 操作授予访问权限。此外，策略还包含主体列表，用于指定要向其授予访问权限的 IAM 角色、用户和 SAML 身份。

Selected principals

Principals

arn:aws:iam::478253424788:user/Administrator

saml/478253424788/myprovider/user/Annie

Granted resources and permissions (2)

Granted resources	Resource type	Permissions
collection/autopartsinventory	collection	aoss:CreateCollectionItems aoss:UpdateCollectionItems
index/test-collection/*	index	aoss:ReadDocument aoss:DescribeIndex

有关数据访问策略格式的更多信息，请参阅策略语法。

在创建数据访问策略之前，您必须拥有一个或多个 IAM 角色、用户或 SAML 身份，才能在策略中为其提供访问权限。有关详细信息，请参阅下一节。

> **Note**
>
> 将您的集合从公共访问切换到私有访问权限将移除 OpenSearch 无服务器集合控制台中的 “索引” 选项卡。

IAM 和 SAML 身份验证

IAM 主体和 SAML 身份是数据访问策略的构建基块之一。在访问策略的 `principal` 语句中，您可以包含 IAM 角色、用户和 SAML 身份。然后，向这些主体授予您在关联的策略规则中指定的权限。

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
      "Principal":[
```

```
            "arn:aws:iam::123456789012:user/Dale",
            "arn:aws:iam::123456789012:role/RegulatoryCompliance",
            "saml/123456789012/myprovider/user/Annie"
        ]
    }
]
```

您可以直接在 OpenSearch 无服务器中配置 SAML 身份验证。有关更多信息，请参阅 the section called “SAML 身份验证”。

基础结构安全性

Amazon OpenSearch Serverless 受 Amazon 全球网络安全的保护。有关 Amazon 安全服务以及如何 Amazon 保护基础设施的信息，请参阅Amazon 云安全。要使用基础设施安全的最佳实践来设计您的 Amazon 环境，请参阅 S Amazon ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 Amazon 已发布的 API 调用通过网络访问 Amazon OpenSearch Serverless。客户端必须支持传输层安全性（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。有关 TLS 1.3 支持的密码列表，请参阅《弹性负载均衡文档》中的 TLS protocols and ciphers。

此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用 Amazon Security Token Service（Amazon STS）生成临时安全凭证来对请求进行签名。

开始使用 Amazon OpenSearch Serverless 中的安全功能

以下教程可帮助您开始使用 Amazon OpenSearch Serverless。这两个教程将完成相同的基本步骤，但一个使用控制台，另一个则使用 Amazon CLI。

请注意，这些教程中的应用场景已简化。网络和安全策略非常开放。在生产工作负载中，建议您配置更强大的安全功能，如 SAML 身份验证、VPC 访问权限和限制性数据访问策略。

主题

- 教程：Amazon OpenSearch Serverless（控制台）安全入门
- 教程：亚马逊 OpenSearch 无服务器 (CLI) 安全入门

教程：Amazon OpenSearch Serverless（控制台）安全入门

本教程将引导您完成使用 Amazon OpenSearch Serverless 控制台创建和管理安全策略的基本步骤。

在本教程中，您将完成以下步骤：

1. 配置权限
2. 创建加密策略
3. 创建网络策略
4. 配置数据访问策略
5. 创建集合
6. 上传和搜索数据

本教程将引导您完成使用设置收藏夹的过程 Amazon Web Services Management Console。有关使用中的相同步骤 Amazon CLI，请参阅the section called “教程：安全性入门（CLI）”。

步骤 1：配置权限

> **Note**
>
> 如果您已经在使用更广泛的基于身份的策略，如 `Action":"aoss:*"` 或 `Action":"*"`，则可以跳过此步骤。但在生产环境中，建议您遵循最低权限原则，仅分配完成任务所需的最低权限。

要完成本教程，您必须拥有正确的 IAM 权限。您的用户或角色必须已经附加基于身份的策略，并且具有以下最低权限：

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": "*"
      }
    ]
}
```

有关 OpenSearch 无服务器权限的完整列表，请参阅the section called “身份和访问管理”。

步骤 2：创建加密策略

加密策略指定了 Amazon KMS OpenSearch Serverless 用于加密集合的密钥。您可以使用 Amazon 托管式密钥 或不同的密钥对集合进行加密。在本教程中，为简单起见，我们将使用 Amazon 托管式密钥对集合进行加密。

创建加密策略

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 展开左侧导航窗格中的 Serverless（无服务器），然后选择 Encryption policies（加密策略）。
3. 选择 Create encryption policy（创建加密策略）。
4. 将该策略命名为 books-policy。对于描述，请输入 Encryption policy for books collection（适用于书籍集合的加密策略）。
5. 在 Resources（资源）下，输入 books（书籍），您将使用它为您的集合命名。如果您想扩大范围，可以添加星号 (`books*`)，以使该策略适用于以“books”（书籍）一词开头的所有集合。
6. 对于 “加密”，保留选中 “使用 Amazon Web Services 自有密钥”。
7. 选择创建。

步骤 3：创建网络策略

网络策略决定您的馆藏是否可通过互联网从公共网络访问，或者是否必须通过 OpenSearch 无服务器托管的 VPC 终端节点进行访问。在本教程中，我们将配置公共访问权限。

创建网络策略

1. 选择左侧导航窗格中的 Network policies（网络策略），然后选择 Create network policy（创建网络策略）。
2. 将该策略命名为 books-policy。对于描述，请输入 Network policy for books collection（适用于书籍集合的网络策略）。

3. 在 Rule 1（规则 1）下，将规则命名为 Public access for books collection（针对书籍集合的公共访问权限）。
4. 在本教程中，为简单起见，我们将配置针对 books（书籍）集合的公共访问权限。对于访问权限类型，选择 Public（公共）。
5. 我们将从 OpenSearch 仪表板访问该集合。为此，您需要为仪表板和 OpenSearch 端点配置网络访问权限，否则仪表板将无法运行。

   对于资源类型，启用对 OpenSearch端点的访问和对 OpenSearch仪表板的访问权限。
6. 在两个输入框中，输入 Collection Name = books（集合名称 = 书籍）。此设置将缩小该策略的范围，使其仅适用于单个集合 (books)。您的规则应如下所示：

7. 选择创建。

步骤 4：创建数据访问策略

您的集合数据将不可访问，直到您配置数据访问权限为止。数据访问策略与您在步骤 1 中配置的 IAM 基于身份的策略是分开的。它们允许用户访问集合中的实际数据。

在本教程中，我们将为单个用户提供将数据索引到 books（书籍）集合所需的权限。

创建数据访问策略

1. 选择左侧导航窗格中的 Data access policies（数据访问策略），然后选择 Create access policy（创建访问策略）。

2. 将该策略命名为 books-policy。对于描述，请输入 Data access policy for books collection（适用于书籍集合的数据访问策略）。
3. 为策略定义方法选择 JSON，然后将以下策略粘贴到 JSON 编辑器中。

   将委托人 ARN 替换为用于登录 OpenSearch 控制面板和索引数据的账户的 ARN。

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/books/*"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:DescribeIndex",
               "aoss:ReadDocument",
               "aoss:WriteDocument",
               "aoss:UpdateIndex",
               "aoss:DeleteIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

   此策略将为单个用户提供在 books（书籍）集合中创建索引、索引某些数据和以及搜索这些数据所需的最低权限。
4. 选择创建。

步骤 5：创建集合

您已经配置了加密和网络策略，现在您可以创建相匹配的集合，安全设置将自动应用于该集合。

创建 OpenSearch 无服务器集合

1. 选择左侧导航窗格中的 Collections（集合），然后选择 Create collection（创建集合）。

2. 将该集合命名为 books（书籍）。
3. 对于集合类型，选择 Search（搜索）。
4. 在 “加密” 下， OpenSearch Serverless 会通知您集合名称与`books-policy`加密策略匹配。
5. 在 “网络访问设置” 下， OpenSearch Serverless 会通知您集合名称与`books-policy`网络策略匹配。
6. 选择下一步。
7. 在数据访问策略选项下， OpenSearch Serverless 会通知您集合名称与`books-policy`数据访问策略相匹配。
8. 选择下一步。
9. 查看集合配置并选择 Submit（提交）。集合通常需要不到一分钟的时间来初始化。

步骤 6：上传和搜索数据

您可以使用 Postman 或 curl 将数据上传到 OpenSearch 无服务器集合。为简洁起见，这些示例使用 OpenSearch 仪表板控制台中的开发工具。

索引和搜索集合中的数据

1. 选择左侧导航窗格中的 Collections（集合），然后选择 books（书籍）集合以打开其详细信息页面。
2. 为该集合选择 OpenSearch 仪表板 URL。该 URL 采用 `https://collection-id.us-east-1.aoss.amazonaws.com/_dashboards` 格式。
3. 使用您在数据Amazon 访问策略中为委托人指定的访问密钥和密钥登录控制面 OpenSearch 板。
4. 在 “ OpenSearch 控制面板” 中，打开左侧导航菜单，然后选择 “开发工具”。
5. 要创建名为 books-index 的单个索引，请运行以下命令：

```
PUT books-index
```

6. 要将单个文档索引到 books-index 中，请运行以下命令：

```
PUT books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

7. 要在 OpenSearch 仪表板中搜索数据，您需要配置至少一种索引模式。 OpenSearch 使用这些模式来确定要分析的索引。打开控制面板主菜单，选择堆栈管理，选择索引模式，然后选择创建索引模式。对于本教程，请输入 books-index。
8. 选择下一步，然后选择创建索引模式。创建模式后，您可以查看各种文档字段，例如 `author` 和 `title`。
9. 要开始搜索您的数据，请再次打开主菜单，然后选择 Discover（发现），或者使用搜索 API。

教程：亚马逊 OpenSearch 无服务器 (CLI) 安全入门

为了安全起见，本教程将引导您完成控制台入门教程中描述的步骤，但使用 Amazon CLI 的是而不是 OpenSearch 服务控制台。

在本教程中，您将完成以下步骤：

1. 创建 IAM 权限策略
2. 将 IAM policy 附到 IAM 角色上

3. 创建加密策略
4. 创建网络策略
5. 创建集合
6. 配置数据访问策略
7. 检索集合端点
8. 将数据上载到您的连接
9. 在您的连接中搜索数据

本教程的目标是使用相当简单的加密、网络和数据访问设置来设置单个 OpenSearch Serverless 集合。例如，我们将配置公共网络访问权限，一个 Amazon 托管式密钥 用于加密，以及一个向单个用户授予最低权限的简化数据访问策略。

在生产场景中，应考虑实施更强大的配置，包括 SAML 身份验证、自定义加密密钥和 VPC 访问权限。

开始使用 OpenSearch 无服务器中的安全策略

1.

> **Note**
>
> 如果您已经在使用更广泛的基于身份的策略，如 `Action":"aoss:*"` 或 `Action":"*"`，则可以跳过此步骤。但在生产环境中，建议您遵循最低权限原则，仅分配完成任务所需的最低权限。

首先，创建一个具有执行本教程中步骤所需的最低权限的 Amazon Identity and Access Management 策略。我们将该策略命名为 `TutorialPolicy`：

```
aws iam create-policy \
  --policy-name TutorialPolicy \
  --policy-document "{\"Version\": \"2012-10-17\",\"Statement\":
 [{\"Action\": [\"aoss:ListCollections\",\"aoss:BatchGetCollection\",
\"aoss:CreateCollection\",\"aoss:CreateSecurityPolicy\",\"aoss:GetSecurityPolicy\",
\"aoss:ListSecurityPolicies\",\"aoss:CreateAccessPolicy\",\"aoss:GetAccessPolicy\",
\"aoss:ListAccessPolicies\"],\"Effect\": \"Allow\",\"Resource\": \"*\"}]}"
```

示例响应

```
{
```

```
    "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. 将 `TutorialPolicy` 附加到 IAM 角色，该角色将在集合中索引并搜索数据。我们将该用户命名为 `TutorialRole`：

```
aws iam attach-role-policy \
  --role-name TutorialRole \
  --policy-arn arn:aws:iam::123456789012:policy/TutorialPolicy
```

3. 在创建集合之前，您需要创建加密策略，该策略要将 Amazon 拥有的密钥 分配给您会在后续步骤中创建的 books（书籍）集合。

   发送以下请求，为 books（书籍）集合创建加密策略：

```
aws opensearchserverless create-security-policy \
  --name books-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/books\"]}],\"AWSOwnedKey\":true}"
```

   示例响应

```
{
    "securityPolicyDetail": {
        "type": "encryption",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDAwNTk5MF8x",
        "policy": {
            "Rules": [
                {
                    "Resource": [
```

```
                        "collection/books"
                    ],
                    "ResourceType": "collection"
                }
            ],
            "AWSOwnedKey": true
        },
        "createdDate": 1669240005990,
        "lastModifiedDate": 1669240005990
    }
}
```

4. 创建网络策略，该策略将提供针对 books（书籍）集合的公共访问权限：

```
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}]"
```

示例响应

```
{
    "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. 创建 books（书籍）集合：

```
aws opensearchserverless create-collection --name books --type SEARCH
```

示例响应

```
{
    "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. 创建数据访问策略，该策略将提供索引和搜索 books（书籍）集合中的数据的最低权限。将主体 ARN 替换为步骤 1 中 `TutorialRole` 的 ARN：

```
aws opensearchserverless create-access-policy \
  --name books-policy \
  --type data \
  --policy "[{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":
[\"index\/books\/books-index\"],\"Permission\":[\"aoss:CreateIndex
\",\"aoss:DescribeIndex\",\"aoss:ReadDocument\",\"aoss:WriteDocument
\",\"aoss:UpdateIndex\",\"aoss:DeleteIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:role\/TutorialRole\"]}]"
```

示例响应

```
{
    "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
    }
}
```

现在，`TutorialRole` 应该能够索引和搜索 books（书籍）集合中的文档。

7. 要调用 OpenSearch API，您需要收集端点。发送以下请求以检索 `collectionEndpoint` 参数：

```
aws opensearchserverless batch-get-collection --names books
```

示例响应

```
{
    "collectionDetails": [
        {
            "id": "8kw362bpwg4gx9b2f6e0",
            "name": "books",
            "status": "ACTIVE",
            "type": "SEARCH",
            "description": "",
            "arn": "arn:aws:aoss:us-
east-1:123456789012:collection/8kw362bpwg4gx9b2f6e0",
            "createdDate": 1665765327107,
            "collectionEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com",
            "dashboardEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com/_dashboards"
        }
    ],
    "collectionErrorDetails": []
}
```

> **Note**
>
> 在集合状态更改为 ACTIVE 之前，您将无法看到集合端点。在成功创建集合之前，可能需要进行多次调用以检查状态。

8. 使用 Postman 或 curl 等 HTTP 工具，将数据索引到 books（书籍）集合中。我们将创建一个名为 books-index 的索引，并添加一个文档。

   使用 TutorialRole 的凭证，将以下请求发送到您在上一步中检索到的集合端点。

```
PUT https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

   示例响应

```
{
  "_index" : "books-index",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 0,
    "successful" : 0,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 0
}
```

9. 要开始搜索您的集合中的数据，请使用搜索 API。以下查询将执行基本搜索：

```
GET https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_search
```

示例响应

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

适用于 Amazon OpenSearch Serverless 的身份和访问管理

Amazon Identity and Access Management (IAM) Amazon Web Services 服务 可帮助管理员安全地控制对 Amazon 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 OpenSearch 无服务器资源。您可以使用 IAM Amazon Web Services 服务 ，无需支付额外费用。

主题

- 无服务器基于身份的策略 OpenSearch
- OpenSearch 无服务器的策略操作
- OpenSearch 无服务器策略资源
- Amazon OpenSearch 无服务器的策略条件密钥
- 带有无服务器功能的 ABA OpenSearch C
- 在 OpenSearch 无服务器中使用临时证书
- 无服务器的服务相关角色 OpenSearch
- 其他策略类型
- 无服务器基于身份的策略示例 OpenSearch
- 对 Amazon OpenSearch 无服务器的 IAM 身份中心支持

无服务器基于身份的策略 OpenSearch

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

无服务器基于身份的策略示例 OpenSearch

要查看基于身份的 OpenSearch 无服务器策略的示例，请参阅。the section called “基于身份的策略示例”

OpenSearch 无服务器的策略操作

支持策略操作：是

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 Amazon API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

OpenSearch Serverless 中的策略操作在操作前使用以下前缀：

```
aoss
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "aoss:action1",
      "aoss:action2"
         ]
```

您可以使用通配符 (*) 指定多个操作。例如，要指定以单词 `Describe` 开头的所有操作，包括以下操作：

```
"Action": "aoss:List*"
```

要查看基于身份的 OpenSearch 无服务器策略的示例，请参阅。 无服务器基于身份的策略示例 OpenSearch

OpenSearch 无服务器策略资源

支持策略资源：是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符（*）指示语句应用于所有资源。

```
"Resource": "*"
```

Amazon OpenSearch 无服务器的策略条件密钥

支持特定于服务的策略条件键：是

管理员可以使用 Amazon JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 `Condition` 元素（或 `Condition` 块）中，可以指定语句生效的条件。`Condition` 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 `Condition` 元素，或在单个 `Condition` 元素中指定多个键，则 Amazon 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 Amazon 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM 策略元素：变量和标签。

Amazon 支持全局条件密钥和特定于服务的条件密钥。要查看所有 Amazon 全局条件键，请参阅 IAM 用户指南中的Amazon 全局条件上下文密钥。

除了基于属性的访问控制 (ABAC) 之外， OpenSearch Serverless 还支持以下条件键：

- `aoss:collection`
- `aoss:CollectionId`
- `aoss:index`

即使在为访问策略和安全策略提供权限时，您也可以使用这些条件键。例如：

```
[
   {
      "Effect":"Allow",
      "Action":[
         "aoss:CreateAccessPolicy",
         "aoss:CreateSecurityPolicy"
      ],
      "Resource":"*",
      "Condition":{
         "StringLike":{
            "aoss:collection":"log"
         }
      }
   }
]
```

在此示例中，该条件适用于此类策略：包含与集合名称或模式相匹配的规则。这些条件具有以下行为：

- `StringEquals`：适用于此类策略：具有包含确切的资源字符串“log”（即 `collection/log`）的规则。
- `StringLike`：适用于此类策略：具有包含资源字符串的规则，该资源字符串包括字符串“log”（即 `collection/log`，但也包括 `collection/logs-application` 或 `collection/applogs123`）。

> **Note**
>
> Collection（集合）条件键不适用于索引级别。例如，在上述策略中，该条件不适用于包含资源字符串 `index/logs-application/*` 的访问或安全策略。

要查看 OpenSearch 无服务器条件密钥列表，请参阅《服务授权参考》中的 Amazon OpenSearch Serverless 条件密钥。要了解您可以使用哪些操作和资源使用条件密钥，请参阅 Amazon OpenSearch Serverless 定义的操作。

带有无服务器功能的 ABA OpenSearch C

支持 ABAC（策略中的标签）：是

基于属性的访问控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在中 Amazon，这些属性称为标签。您可以将标签附加到 IAM 实体（用户或角色）和许多 Amazon 资源。标记实体和资源

是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

有关标记 OpenSearch 无服务器资源的更多信息，请参阅。the section called “标记集合”

在 OpenSearch 无服务器中使用临时证书

支持临时凭证：是

当你使用临时证书登录时，有些 Amazon Web Services 服务 不起作用。有关更多信息，包括哪些 Amazon Web Services 服务 适用于临时证书，请参阅 IAM 用户指南中的Amazon Web Services 服务与 IA M 配合使用的信息。

如果您使用除用户名和密码之外的任何方法登录，则 Amazon Web Services Management Console 使用的是临时证书。例如，当您 Amazon 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的从用户切换到 IAM 角色（控制台）。

您可以使用 Amazon CLI 或 Amazon API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 Amazon。 Amazon 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

无服务器的服务相关角色 OpenSearch

支持服务相关角色：是

服务相关角色是一种与服务相关联的 Amazon Web Services 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 Amazon Web Services 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建和管理 OpenSearch 无服务器服务相关角色的详细信息，请参阅。the section called “集合创建角色”

其他策略类型

Amazon 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 服务控制策略 (SCPs)- SCPs 是指定组织或组织单位 (OU) 的最大权限的 JSON 策略 Amazon Organizations。 Amazon Organizations 是一项用于对您的企业拥有的多个 Amazon 账户进行分组和集中管理的服务。如果您启用组织中的所有功能，则可以将服务控制策略 (SCPs) 应用于您的任何或所有帐户。SCP 限制成员账户中实体的权限，包括每个 Amazon 账户的根用户。有关 Organization SCPs s 和的更多信息，请参阅《Amazon Organizations 用户指南》中的服务控制策略。
- 资源控制策略 (RCPs) — RCPs 是 JSON 策略，您可以使用它来设置账户中资源的最大可用权限，而无需更新附加到您拥有的每个资源的 IAM 策略。RCP 限制成员账户中资源的权限，并可能影响身份（包括 Amazon 账户根用户）的有效权限，无论他们是否属于您的组织。有关 Organizations 的更多信息 RCPs，包括支持的 Amazon 服务列表 RCPs，请参阅《Amazon Organizations 用户指南》中的资源控制策略 (RCPs)。

无服务器基于身份的策略示例 OpenSearch

默认情况下，用户和角色无权创建或修改 OpenSearch 无服务器资源。他们也无法使用 Amazon Web Services Management Console、 Amazon Command Line Interface (Amazon CLI) 或 Amazon API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 Amazon OpenSearch Serverless 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》中的 Amazon OpenSearch Serverless 的操作、资源和条件密钥。 ARNs

主题

- 策略最佳实践
- 在控制台中使用 OpenSearch 无服务器
- 管理 OpenSearch 无服务器集合
- 查看 OpenSearch 无服务器集合
- 使用 OpenSearch API 操作
- 用于 OpenSearch API 操作的 ABAC

策略最佳实践

基于身份的策略非常强大。它们决定是否有人可以在您的账户中创建、访问或删除 OpenSearch 无服务器资源。这些操作可能会使 Amazon Web Services 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 OpenSearch 无服务器资源。这些操作可能会使 Amazon Web Services 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 Amazon 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的Amazon 托管策略。它们在你的版本中可用 Amazon Web Services 账户。我们建议您通过定义针对您的用例的 Amazon 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 Amazon 托管式策略或工作职能的Amazon 托管式策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 Amazon Web Services 服务，例如 Amazon CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA)-如果 Amazon Web Services 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

在控制台中使用 OpenSearch 无服务器

要在 OpenSearch 服务控制台中访问 OpenSearch Serverless，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您 Amazon 账户中 OpenSearch 无服务器资源的详细信息。如果您创建的基于身份的策略比所需的最低权限更严格，则无法为具有该策略的实体（如 IAM 角色）正常运行控制台。

对于仅调用 Amazon CLI 或 Amazon API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

以下策略允许用户在 OpenSearch 服务控制台中访问 OpenSearch Serverless：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "aoss:ListCollections",
                "aoss:BatchGetCollection",
                "aoss:ListAccessPolicies",
                "aoss:ListSecurityConfigs",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:ListVpcEndpoints",
                "aoss:GetAccessPolicy",
                "aoss:GetAccountSettings",
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy"
            ]
        }
    ]
}
```

管理 OpenSearch 无服务器集合

此政策是 “馆藏管理员” 策略的一个示例，该策略允许用户管理和管理 Amazon OpenSearch Serverless 馆藏。用户可以创建、查看和删除集合。

JSON

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

```
        {
            "Resource": "arn:aws:aoss:region:123456789012:collection/*",
            "Action": [
                "aoss:CreateCollection",
                "aoss:DeleteCollection",
                "aoss:UpdateCollection"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:ListCollections",
                "aoss:CreateAccessPolicy",
                "aoss:CreateSecurityPolicy"
            ],
            "Effect": "Allow"
        }
    ]
}
```

查看 OpenSearch 无服务器集合

此示例策略允许用户查看其账户中所有 Amazon OpenSearch Serverless 集合的详细信息。用户无法修改集合或任何相关联的安全策略。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:ListCollections",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:BatchGetCollection"
            ],
            "Effect": "Allow"
```

```
        }
    ]
}
```

使用 OpenSearch API 操作

数据平面 API 操作由你在 OpenSearch Serverless 中使用的函数组成，这些函数用于从服务中获取实时价值。控制面板 API 操作由用于设置环境的函数组成。

要通过浏览器访问 Amazon OpenSearch Serverless 数据平面 APIs 和 OpenSearch 控制面板，您需要为集合资源添加两个 IAM 权限。这些权限是 `aoss:APIAccessAll` 和 `aoss:DashboardsAccessAll`。

> **Note**
>
> 从 2023 年 5 月 10 日起， OpenSearch Serverless 需要这两个新的 IAM 权限才能使用集合资源。该`aoss:APIAccessAll`权限允许访问数据平面，该`aoss:DashboardsAccessAll`权限允许通过浏览器访问 OpenSearch 仪表板。未能添加这两个新 IAM 权限会导致出现 403 错误。

此示例策略允许用户访问其账户中指定集合的数据平面 APIs ，以及访问其账户中所有集合的 OpenSearch 控制面板。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": "aoss:APIAccessAll",
            "Resource": "arn:aws:aoss:region:account-id:collection/collection-id"
        },
        {
            "Effect": "Allow",
            "Action": "aoss:DashboardsAccessAll",
            "Resource": "arn:aws:aoss:region:account-id:dashboards/default"
        }
    ]
```

```
}
```

两者兼而有之，`aoss:APIAccessAll`并`aoss:DashboardsAccessAll`授予对集合资源的完整 IAM 权限，而 Dashboards 权限还提供 OpenSearch 仪表板访问权限。每个权限都彼此独立，因此明确拒绝 `aoss:APIAccessAll` 不会妨碍对资源（包括开发工具）的 `aoss:DashboardsAccessAll`访问。否认也是如此`aoss:DashboardsAccessAll`。 OpenSearch Serverless 支持以下全局条件键：

- `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`

以下是在委托人的 IAM 策略的条件块`aws:SourceIp`中使用数据平面调用的示例：

```
"Condition": {
    "IpAddress": {
         "aws:SourceIp": "203.0.113.0"
    }
}
```

以下是在委托人的 IAM 策略的条件块`aws:SourceVpc`中使用数据平面调用的示例：

```
"Condition": {
    "StringEquals": {
        "aws:SourceVpc": "vpc-0fdd2445d8EXAMPLE"
    }
}
```

此外，还支持以下 OpenSearch 无服务器特定的密钥：

- `aoss:CollectionId`
- `aoss:collection`

以下是在委托人的 IAM 策略的条件块`aoss:collection`中使用数据平面调用的示例：

```
"Condition": {
    "StringLike": {
         "aoss:collection": "log-*"
    }
}
```

用于 OpenSearch API 操作的 ABAC

基于身份的策略允许您使用标签来控制对 Amazon OpenSearch Serverless 数据平面的访问。 APIs APIs 如果集合带有`team:devops`标签，则以下策略允许附加的委托人访问数据平面：

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "aoss:APIAccessAll",
            "Resource": "arn:aws:aoss:region:account-id:collection/collection-
id",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/team": "devops"
                }
            }
        }
    ]
}
```

以下策略是拒绝附加委托人访问数据平面 APIs 和仪表板访问的示例（如果集合带有`environment:production`标签）：

```
{
    "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 Serverless 不支持数据平面`RequestTag` APIs和`TagKeys`全局条件密钥。

对 Amazon OpenSearch 无服务器的 IAM 身份中心支持

对 Amazon OpenSearch 无服务器的 IAM 身份中心支持

您可以使用 IAM 身份中心委托人（用户和群组）通过亚马逊应用程序访问亚马逊 OpenSearch 无服务器数据。 OpenSearch 要启用 Amazon OpenSearch Serverless 的 IAM 身份中心支持，您需要启用 IAM 身份中心。要详细了解如何执行此操作，请参阅什么是 IAM 身份中心？

创建 IAM 身份中心实例后，客户账户管理员需要为 Amazon OpenSearch Serverless 服务创建 IAM 身份中心应用程序。这可以通过调用CreateSecurityConfig:来完成。客户账户管理员可以指定将使用哪些属性来授权请求。使用的默认属性是`UserId`和 `GroupId`.

Amazon OpenSearch Serverless 的 IAM 身份中心集成使用以下 Amazon IAM 身份中心 (IAM) 权限：

- `aoss:CreateSecurityConfig`— 创建 IAM 身份中心提供商
- `aoss:ListSecurityConfig`— 列出当前账户中的所有 IAM 身份中心提供商。
- `aoss:GetSecurityConfig`— 查看 IAM 身份中心提供商信息。
- `aoss:UpdateSecurityConfig`— 修改给定的 IAM 身份中心配置
- `aoss:DeleteSecurityConfig`— 删除 IAM 身份中心提供商。

以下基于身份的访问策略可用于管理所有 IAM Identity Center 配置：

JSON

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Action": [
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
```

```
}
```

> **Note**
>
> 该`Resource`元素必须是通配符。

创建 IAM 身份中心提供商（控制台）

您可以创建 IAM 身份中心提供商以启用 OpenSearch应用程序身份验证。要为 OpenSearch 控制面板启用 IAM 身份中心身份验证，请执行以下步骤：

1. 登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航面板上，展开无服务器并选择身份验证。
3. 选择 IAM 身份中心身份验证。
4. 选择 “编辑”
5. 选中 “通过 IAM 身份中心进行身份验证” 旁边的复选框。
6. 从下拉菜单中选择用户和群组属性键。用户属性将用于根据`UserNameUserId`、和对用户进行授权`Email`。组属性将用于根据`GroupName`和对用户进行身份验证`GroupId`。
7. 选择 IAM 身份中心实例。
8. 选择 “保存”

创建 IAM 身份中心提供商 (Amazon CLI)

要使用 Amazon Command Line Interface (Amazon CLI) 创建 IAM 身份中心提供商，请使用以下命令：

```
aws opensearchserverless create-security-config \
--region us-east-2 \
--name "iamidentitycenter-config" \
--description "description" \
--type "iamidentitycenter" \
--iam-identity-center-options '{
    "instanceArn": "arn:aws:sso:::instance/ssoins-99199c99e99ee999",
    "userAttribute": "UserName",
    "groupAttribute": "GroupId"
```

```
}'
```

启用 IAM 身份中心后，客户只能修改用户和群组的属性。

```
aws opensearchserverless update-security-config \
--region us-east-1 \
--id <id_from_list_security_configs> \
--config-version <config_version_from_get_security_config> \
--iam-identity-center-options-updates '{
    "userAttribute": "UserId",
    "groupAttribute": "GroupId"
}'
```

要使用查看 IAM 身份中心提供商 Amazon Command Line Interface，请使用以下命令：

```
aws opensearchserverless list-security-configs --type iamidentitycenter
```

删除 IAM 身份中心提供商

IAM Identity Center 提供两个提供商实例，一个用于您的组织账户，另一个用于您的成员账户。如果您需要更改 IAM Identity Center 实例，则需要通过 `DeleteSecurityConfig` API 删除您的安全配置，并使用新的 IAM Identity Center 实例创建新的安全配置。以下命令可用于删除 IAM 身份中心提供商：

```
aws opensearchserverless delete-security-config \
--region us-east-1 \
--id <id_from_list_security_configs>
```

授予 IAM 身份中心访问收集数据的权限

启用 IAM Identity Center 提供商后，您可以更新收集数据访问策略以包含 IAM 身份中心委托人。需要按以下格式更新 IAM 身份中心委托人：

```
[
   {
"Rules":[
       ...
      ],
      "Principal":[
         "iamidentitycenter/<iamidentitycenter-instance-id>/user/<UserName>",
         "iamidentitycenter/<iamidentitycenter-instance-id>/group/<GroupId>"
```

```
        ]
    }
]
```

> **Note**
>
> Amazon OpenSearch Serverless 仅支持一个用于所有客户集合的 IAM 身份中心实例，并且单个用户最多可以支持 100 个群组。如果您尝试使用的实例数量超过允许的数量，则会遇到与数据访问策略授权处理不一致的问题，并会收到一条403错误消息。

您可以授予针对集合、索引或两者的访问权限。如果您希望不同的用户拥有不同的权限，则需要创建多个规则。有关可用权限的列表，请参阅亚马逊 OpenSearch 服务中的 Identity and Access 管理。有关如何格式化访问策略的信息，请参阅授予 SAML 身份访问集合数据的权限。

IAM Identity Center 提供两个提供商实例，一个用于您的组织账户，另一个用于您的成员账户。如果您需要更改 IAM Identity Center 实例，则需要通过 `DeleteSecurityConfig` API 删除您的安全配置，并使用新的 IAM Identity Center 实例创建新的安全配置。以下命令可用于删除 IAM 身份中心提供商：

```
aws opensearchserverless delete-security-config \
--region us-east-1 \
--id <id_from_list_security_configs>
```

Amazon OpenSearch 无服务器中的加密

静态加密

您创建的每个 Amazon OpenSearch Serverless 集合都受到静态数据加密的保护，这是一项有助于防止未经授权访问您的数据的安全功能。静态加密使用 Amazon Key Management Service (Amazon KMS) 来存储和管理您的加密密钥。它使用具有 256 位密钥（AES-256）的高级加密标准算法执行加密。

主题

- 加密策略
- 注意事项
- 所需权限
- 客户托管密钥的密钥策略

- OpenSearch 无服务器如何使用授权 Amazon KMS
- 创建加密策略（控制台）
- 创建加密策略（Amazon CLI）
- 查看加密策略
- 更新加密策略
- 删除加密策略

加密策略

借助加密策略，您可以通过自动将加密密钥分配给与某个具体名称或模式相匹配的新建集合来大规模管理多个集合。

在创建加密策略时，您可以指定前缀（它是基于通配符的匹配规则，如 `MyCollection*`），也可以输入一个集合名称。然后，当您创建与该名称或前缀模式相匹配的集合时，会将该策略和相应的 KMS 密钥自动分配给该集合。

加密策略包含以下元素：

- `Rules`：一条或多条集合匹配规则，每条规则都包含以下子元素：
  - `ResourceType`：目前唯一选项为“collection”（集合）。加密策略仅适用于集合资源。
  - `Resource`：策略将适用于的一个或多个集合名称或模式，格式为 `collection/<collection name|pattern>`。

- `AWSOwnedKey`：是否使用 Amazon 拥有的密钥。
- `KmsARN`：如果您将 `AWSOwnedKey` 设置为 false，请指定用于加密关联集合的 KMS 密钥的 Amazon 资源名称（ARN）。如果包含此参数，则 OpenSearch Serverless 会忽略该`AWSOwnedKey`参数。

以下示例策略会将客户托管的密钥分配给名为 `autopartsinventory` 的任何未来集合，以及以“sales”（销售）一词开头的集合：

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

即使策略与集合名称相匹配，如果资源模式包含通配符 (*)，您也可以在集合创建期间选择覆盖此自动分配。如果您选择覆盖自动密钥分配， OpenSearch Serverless 会为您创建一个名为 a uto-< ***collection-name*** > 的加密策略并将其附加到集合中。该策略最初仅适用于一个集合，但您可以将其修改为包括其他集合。

如果您修改策略规则以不再与某个集合相匹配，则不会从该集合取消分配关联的 KMS 密钥。该集合仍将使用其初始加密密钥进行加密。如果您要更改某个集合的加密密钥，则必须重新创建该集合。

如果来自多个策略的规则与某个集合相匹配，则将使用更具体的规则。例如，如果一个策略包含 `collection/log*` 的规则，而另一个策略包含 `collection/logSpecial` 的规则，则使用第二个策略的加密密钥，因为它更具体。

如果名称或前缀已存在于其他策略中，则不能在策略中使用该名称或前缀。 OpenSearch 如果您尝试在不同的加密策略中配置相同的资源模式，Serverless 会显示错误。

注意事项

在为集合配置加密时，请考虑以下事项：

- 所有无服务器集合都需要静态加密。
- 您可以选择使用客户托管的密钥或 Amazon 拥有的密钥。如果您选择客户托管的密钥，建议您启用 automatic key rotation（自动密钥轮换）。
- 在创建集合后，您将无法更改该集合的加密密钥。首次设置 Amazon KMS 收藏夹时，请仔细选择要使用的收藏夹。
- 一个集合只能匹配一个加密策略。
- 具有唯一 KMS 密钥的集合无法与其他集合共享 OpenSearch 计算单位 (OCUs)。每个拥有唯一密钥的集合都需要自己的 4 OCUs.
- 如果您更新加密策略中的 KMS 密钥，更改不会影响与已分配的 KMS 密钥相匹配的现有集合。
- OpenSearch Serverless 不会明确检查用户对客户托管密钥的权限。如果用户有权通过数据访问策略访问集合，则他们将能够摄取和查询使用关联密钥加密的数据。

所需权限

OpenSearch 无服务器的静态加密使用以下 Amazon Identity and Access Management (IAM) 权限。您可以指定 IAM 条件，以将用户限制到特定集合。

- `aoss:CreateSecurityPolicy`：创建加密策略。
- `aoss:ListSecurityPolicies`：列出所有加密策略及其附加到的集合。
- `aoss:GetSecurityPolicy`：查看某个具体加密策略的详细信息。
- `aoss:UpdateSecurityPolicy`：修改加密策略。
- `aoss:DeleteSecurityPolicy`：删除加密策略。

以下基于身份的示例访问策略为用户提供了使用资源模式 `collection/application-logs` 管理加密策略所需的最低权限。

JSON

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

```
            "aoss:UpdateSecurityPolicy",
            "aoss:DeleteSecurityPolicy",
            "aoss:GetSecurityPolicy"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aoss:collection":"application-logs"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "aoss:ListSecurityPolicies"
         ],
         "Resource":"*"
      }
   ]
}
```

客户托管密钥的密钥策略

如果您选择客户托管密钥来保护集合，则 OpenSearch Serverless 将获得代表做出选择的委托人使用 KMS 密钥的权限。该委托人（用户或角色）必须拥有 S OpenSearch erverless 所需的 KMS 密钥的权限。您可以在密钥策略或 IAM policy 中提供这些权限。

OpenSearch Serverless 至少需要对客户托管密钥具有以下权限：

- kms: DescribeKey
- kms: CreateGrant

例如：

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 无服务器创建具有 kms: GenerateDataKey 和 km s: Dec rypt 权限的授权。

有关更多信息，请参阅 Amazon Key Management Service 开发人员指南中的在 Amazon KMS中使用密钥策略。

OpenSearch 无服务器如何使用授权 Amazon KMS

OpenSearch Serverless 需要获得授权才能使用客户托管密钥。

当您使用新密钥在账户中创建加密策略时， OpenSearch Serverless 会通过向发送CreateGrant请求来 Amazon KMS代表您创建授权。中的授权 Amazon KMS 用于授予对客户账户中 KMS 密钥的 OpenSearch 无服务器访问权限。

OpenSearch Serverless 需要获得授权才能使用您的客户托管密钥进行以下内部操作：

- 向发送DescribeKey请求 Amazon KMS 以验证提供的对称客户托管密钥 ID 是否有效。
- 向 KMS 密钥发送GenerateDataKey请求以创建用于加密对象的数据密钥。
- 将 Decrypt 请求发送 Amazon KMS 到以解密加密的数据密钥，以便它们可用于加密您的数据。

您可以随时撤销授予访问权限，或删除服务对客户托管密钥的访问权限。如果这样做， OpenSearch Serverless 将无法访问由客户托管密钥加密的任何数据，这会影响依赖该数据的所有操作，从而导致异步工作流程中`AccessDeniedException`出现错误和故障。

OpenSearch 当给定的客户托管密钥与任何安全策略或集合没有关联时，Serverless 会在异步工作流程中停用授权。

创建加密策略（控制台）

在加密策略中，您可以指定 KMS 密钥和一系列将应用该策略的集合模式。在创建集合时，将为与该策略中定义的模式之一相匹配的任何新集合分配相应的 KMS 密钥。建议您在开始创建集合之前，先创建加密策略。

创建 OpenSearch 无服务器加密策略

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 在左侧导航面板上，展开 Serverless（无服务器），然后选择 Encryption policies（加密策略）。
3. 选择 Create encryption policy（创建加密策略）。
4. 为策略提供名称和描述。
5. 在 Resources（资源）下，为此加密策略输入一个或多个资源模式。将为当前 Amazon Web Services 账户 和区域中与其中某一模式相匹配的任何新建集合自动分配此策略。例如，如果您输入（`ApplicationLogs`不带通配符），然后使用该名称创建集合，则该策略和相应的 KMS 密钥将分配给该集合。

   您还可以提供前缀，如 `Logs*`，它会将该策略分配给名称以 `Logs` 开头的任何新集合。通过使用通配符，您可以大规模管理多个集合的加密设置。
6. 在 Encryption（加密）下，选择要使用的 KMS 密钥。
7. 选择创建。

下一步：创建集合

在配置一个或多个加密策略后，您可以开始创建与这些策略中定义的规则相匹配的集合。有关说明，请参阅the section called “创建集合”。

在创建集合的加密步骤中， OpenSearch Serverless 会通知您输入的名称与加密策略中定义的模式相匹配，并自动将相应的 KMS 密钥分配给该集合。如果资源模式包含通配符 (*)，则您可以选择覆盖匹配，然后选择您自己的密钥。

创建加密策略（Amazon CLI）

要使用 OpenSearch 无服务器 API 操作创建加密策略，您需要指定资源模式和 JSON 格式的加密密钥。该CreateSecurityPolicy请求接受内联策略和.json 文件。

加密策略采用以下格式。此示例 `my-policy.json` 文件将与名为 `autopartsinventory` 的任何未来集合以及名称以 `sales` 开头的任何集合相匹配。

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

要使用服务拥有的密钥，请将 `AWSOwnedKey` 设置为 `true`：

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":true
}
```

以下请求将创建加密策略：

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy file://my-policy.json
```

然后，使用 CreateCollectionAPI 操作创建一个或多个与其中一个资源模式相匹配的集合。

查看加密策略

在创建集合之前，您可能想预览账户中的现有加密策略，以查看哪个加密策略的资源模式与您的集合名称相匹配。以下ListSecurityPolicies请求列出了您账户中的所有加密策略：

```
aws opensearchserverless list-security-policies --type encryption
```

该请求将返回有关所有已配置的加密策略的信息。使用 `policy` 元素的内容查看策略中定义的模式规则：

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

要查看有关特定策略（包括 KMS 密钥）的详细信息，请使用GetSecurityPolicy命令。

更新加密策略

如果您更新加密策略中的 KMS 密钥，则更改将仅适用于与已配置的名称或模式相匹配的新建集合。它不会影响已经分配 KMS 密钥的现有集合。

这同样适用于策略匹配规则。如果您添加、修改或删除规则，则更改将仅适用于新建集合。如果您修改策略的规则，使其不再与集合的名称相匹配，则现有集合不会失去已分配它们的 KMS 密钥。

要在 OpenSearch Serverless 控制台中更新加密策略，请选择加密策略，选择要修改的策略，然后选择编辑。进行更改，然后选择保存。

要使用 OpenSearch 无服务器 API 更新加密策略，请使用UpdateSecurityPolicy操作。以下请求将使用新策略 JSON 文档更新加密策略：

```
aws opensearchserverless update-security-policy \
```

```
    --name sales-inventory \
    --type encryption \
    --policy-version 2 \
    --policy file://my-new-policy.json
```

删除加密策略

在删除加密策略时，当前使用该策略中定义的 KMS 密钥的任何集合都不受影响。要在 OpenSearch 无服务器控制台中删除策略，请选择该策略并选择删除。

你也可以使用以下DeleteSecurityPolicy操作：

```
aws opensearchserverless delete-security-policy --name my-policy --type encryption
```

传输中加密

在 OpenSearch Serverless 中，集合中的所有路径在传输过程中都使用传输层安全 1.2 (TLS) 和行业标准 AES-256 密码进行加密。也可以通过 TLS 1.2 访问 Opensearch 的所有控制面板 APIs和控制面板。TLS 是一组行业标准的加密协议，用于加密通过网络交换的信息。

Amazon OpenSearch Serverless 的网络访问

Amazon OpenSearch Serverless 馆藏的网络设置决定了该馆藏是否可以从公共网络通过互联网访问，还是必须以私密方式访问。

私有访问适用于以下一种或两种情形：

- OpenSearch 无服务器托管的 VPC 终端节点
- 支持， Amazon Web Services 服务 例如 Amazon Bedrock

您可以分别为集合的OpenSearch端点及其对应的OpenSearch 仪表板端点配置网络访问权限。

网络访问权限是允许从不同源网络进行访问的隔离机制。例如，如果集合的 OpenSearch Dashboards 端点可以公开访问，但 OpenSearch API 端点不可访问，则用户在从公共网络连接时只能通过仪表板访问集合数据。如果他们尝试 OpenSearch APIs直接从公共网络呼叫，他们将被屏蔽。网络设置可以用于源类型到资源类型的此类排列。Amazon OpenSearch Serverless 同时支持 IPv4 和 IPv6连接。

主题

- 网络策略

- 注意事项
- 配置网络策略所需的权限
- 策略优先顺序
- 创建网络策略（控制台）
- 创建网络策略（Amazon CLI）
- 查看网络策略
- 更新网络策略
- 删除网络策略

网络策略

网络策略使您能够通过为符合策略中定义的规则的集合自动分配网络访问权限设置，从而大规模管理许多集合。

在网络策略中，您可以指定一系列规则。这些规则定义了对集合端点和 OpenSearch 仪表板端点的访问权限。每条规则都由访问类型（公共或私有）和资源类型（集合和/或 OpenSearch 仪表板端点）组成。对于每种资源类型（`collection` 和 `dashboard`），您可以指定一系列规则，用于定义策略将适用于哪些集合。

在此示例策略中，第一条规则针对以 `marketing*` 一词开头的所有集合，指定了对集合端点和控制面板端点的 VPC 端点访问权限。此外还指定了 Amazon Bedrock 访问权限。

> **Note**
>
> Amazon Web Services 服务 诸如 Amazon Bedrock 之类的私有访问权限仅适用于集合的 OpenSearch 终端节点，不适用于 OpenSearch 控制面板终端节点。即使`ResourceType`是`dashboard`，也 Amazon Web Services 服务 无法授予对 OpenSearch 仪表板的访问权限。

第二条规则指定针对 `finance` 集合的公共访问权限，但仅适用于集合端点（无控制面板访问权限）。

```
[
   {
      "Description":"Marketing access",
      "Rules":[
         {
```

```
            "ResourceType":"collection",
            "Resource":[
               "collection/marketing*"
            ]
         },
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/marketing*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   },
   {
      "Description":"Sales access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

此政策仅为以 “财务” 开头的集合提供对 OpenSearch 仪表板的公开访问权限。任何直接访问 OpenSearch API 的尝试都将失败。

```
[
  {
    "Description": "Dashboards access",
    "Rules": [
      {
        "ResourceType": "dashboard",
```

```
        "Resource": [
          "collection/finance*"
        ]
      }
    ],
    "AllowFromPublic": true
  }
]
```

网络策略可以应用于现有集合以及未来集合。例如，您可以创建一个集合，然后使用与集合名称相匹配的规则创建网络策略。在创建集合之前，无需创建网络策略。

注意事项

在为集合配置网络访问权限时，请考虑以下事项：

- 如果您计划为集合配置 VPC 终端节点访问权限，则必须先创建至少一个OpenSearch 无服务器托管的 VPC 终端节点。
- 对的私有访问权限 Amazon Web Services 服务 仅适用于集合的 OpenSearch端点，不适用于 OpenSearch 仪表板端点。即使`ResourceType`是`dashboard`，也 Amazon Web Services 服务 无法授予对 OpenSearch 仪表板的访问权限。
- 如果集合可以从公共网络访问，则也可以从所有 OpenSearch 无服务器托管的 VPC 终端节点和所有终端节点访问该集合。 Amazon Web Services 服务
- 多个网络策略可以应用于一个集合。有关更多信息，请参阅 the section called “策略优先顺序”。

配置网络策略所需的权限

OpenSearch 无服务器的网络访问使用以下 Amazon Identity and Access Management (IAM) 权限。您可以指定 IAM 条件，以将用户限制到与特定集合相关的网络策略。

- `aoss:CreateSecurityPolicy`：创建网络访问策略。
- `aoss:ListSecurityPolicies`：列出当前账户中的所有网络策略。
- `aoss:GetSecurityPolicy`：查看网络访问策略规范。
- `aoss:UpdateSecurityPolicy`：修改给定的网络访问策略，并更改 VPC ID 或公共访问指定。
- `aoss:DeleteSecurityPolicy`：删除网络访问策略（将其与所有集合分离后）。

以下基于身份的访问策略允许用户查看所有网络策略，并使用资源模式 `collection/application-logs` 更新策略：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aoss:UpdateSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": "application-logs"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:ListSecurityPolicies",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> 此外， OpenSearch Serverless 需要`aoss:APIAccessAll`和`aoss:DashboardsAccessAll`权限才能使用集合资源。有关更多信息，请参阅 the section called “使用 OpenSearch API 操作”。

策略优先顺序

在某些情况下，网络策略规则可能会在策略内部或策略之间重叠。发生这种情况时，对于以下两条规则共用的任何集合，指定公有访问权限的规则将覆盖指定私有访问权限的规则。

例如，在以下策略中，两条规则都指定针对 `finance` 集合的网络访问权限，但一条规则指定 VPC 访问权限，而另一条规则指定公共访问权限。在这种情况下，公共访问权限仅会针对“finance”（财务）集合（因为该集合同时存在于两条规则中）覆盖 VPC 访问权限，因此可以从公共网络访问“finance”（财务）集合。“sales”（销售）集合将具有从指定端点进行 VPC 访问的权限。

```
[
   {
      "Description":"Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/sales",
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ]
   },
   {
      "Description":"Rule 2",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

如果不同规则中的多个 VPC 端点适用于一个集合，则这些规则具有累加性，可以从所有指定端点访问该集合。如果您设置`AllowFromPublic`为`true`但同时提供一个或多个`SourceVPCEs`或`SourceServices`，则 OpenSearch Serverless 会忽略 VPC 终端节点和服务标识符，关联的集合将具有公开访问权限。

创建网络策略（控制台）

网络策略可以应用于现有策略以及未来策略。建议您在开始创建集合之前，先创建网络策略。

创建 OpenSearch 无服务器网络策略

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 在左侧导航面板上，展开 Serverless（无服务器），然后选择 Network policies（网络策略）。
3. 选择 Create network policy（创建网络策略）。
4. 为策略提供名称和描述。
5. 提供一条或多条规则。这些规则定义了您的 OpenSearch 无服务器集合及其 OpenSearch 仪表板端点的访问权限。

   每个规则包含以下元素：

元素	描述
Rule name（规则名称）	描述规则内容的名称。例如，“VPC access for marketing team”（针对营销团队的 VPC 访问权限）。
Access type（访问类型）	选择公有或私有访问。选择选择以下一项或两项： • 用于访问的 VPC 终端节点 — 指定一个或多个OpenSearch 无服务器托管的 VPC 终端节点 — 托管 VPC 终端节点。 • Amazon Web Services 服务 私人访问权限- 选择一个或多个支持的访问权限 Amazon Web Services 服务。
资源类型	选择是提供对 OpenSearch 端点（允许调用 OpenSearch API）、 OpenSearch 仪表板

元素	描述
	（允许访问可视化效果和 OpenSearch 插件用户界面）的访问权限，还是两者兼而有之。 **Note** Amazon Web Services 服务 私有访问权限仅适用于集合的 OpenSearch 端点，不适用于 OpenSearch 仪表板端点。即使您选择 “OpenSearch 控制面板”，也 Amazon Web Services 服务只能被授予终端节点访问权限。

对于您选择的每种资源类型，您可以选择要应用策略设置的现有集合， and/or 创建一个或多个资源模式。资源模式由前缀和通配符 (*) 组成，用于定义策略设置将应用于哪些集合。

例如，如果您包括名为 `Marketing*` 的模式，则名称以“Marketing”（营销）开头的任何新建或现有集合都将自动应用此策略中的网络设置。一个通配符 (*) 可将策略应用于所有当前和未来集合。

此外，您可以指定不带通配符的 f ut ure 集合的名称，例如`Finance`。 OpenSearch Serverless 会将策略设置应用于任何新创建的具有该名称的集合。

6. 当您对策略配置感到满意时，选择 Create（创建）。

创建网络策略（Amazon CLI）

要使用 OpenSearch 无服务器 API 操作创建网络策略，您需要以 JSON 格式指定规则。该CreateSecurityPolicy请求接受内联策略和.json 文件。所有集合和模式都必须采用 `collection/<collection name|pattern>` 形式。

> **Note**
> 资源类型`dashboards`仅允许访问 OpenSearch 仪表板的权限，但是为了使 OpenSearch 仪表板正常运行，您还必须允许来自相同来源的集合访问权限。有关示例，请参阅下面的第二个策略。

要指定私有访问，请包含以下一个或两个元素：

- `SourceVPCEs`— 指定一个或多个 OpenSearch 无服务器托管 VPC 终端节点。
- `SourceServices`— 指定一个或多个支持的标识符 Amazon Web Services 服务。目前支持以下服务的标识符：
  - `bedrock.amazonaws.com` – Amazon Bedrock

以下示例网络策略向一个 VPC 端点和 Amazon Bedrock 授予了对集合端点的私有访问权限，但仅限以前缀 `log*` 开头的集合。经过身份验证的用户无法登录 OpenSearch 控制面板；他们只能通过编程方式访问集合端点。

```
[
   {
      "Description":"Private access for log collections",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/log*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   }
]
```

以下策略为名为的单个集合提供对 OpenSearch 终端节点和 OpenSearch 仪表板的公共访问权限`finance`。如果集合不存在，则网络设置将在创建该集合时应用于该集合。

```
[
   {
      "Description":"Public access for finance collection",
      "Rules":[
         {
            "ResourceType":"dashboard",
```

```
            "Resource":[
               "collection/finance"
            ]
         },
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

以下请求将创建上述网络策略：

```
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}]"
```

要在 JSON 文件中提供策略，请使用 `--policy file://my-policy.json` 格式

查看网络策略

在创建集合之前，您可能想预览账户中的现有网络策略，以查看哪个网络策略的资源模式与您的集合名称相匹配。以下ListSecurityPolicies请求列出了您账户中的所有网络政策：

```
aws opensearchserverless list-security-policies --type network
```

该请求将返回有关所有已配置的网络策略的信息。要查看某一具体策略中定义的模式规则，请在响应的 `securityPolicySummaries` 元素内容中查找策略信息。请注意本政策type的name和，并在GetSecurityPolicy请求中使用这些属性来接收包含以下策略详细信息的响应：

```
{
    "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
        }
    ]
}
```

要查看有关特定策略的详细信息，请使用GetSecurityPolicy命令。

更新网络策略

当您修改针对网络的 VPC 端点或公共访问权限指定时，所有相关集合都会受到影响。要在 OpenSearch Serverless 控制台中更新网络策略，请展开网络策略，选择要修改的策略，然后选择编辑。进行更改，然后选择保存。

要使用 OpenSearch 无服务器 API 更新网络策略，请使用UpdateSecurityPolicy命令。您必须在请求中包括策略版本。您可以使用 `ListSecurityPolicies` 或 `GetSecurityPolicy` 命令检索策略版本。包括最新策略版本可以确保您不会无意中覆盖其他人所做的更改。

以下请求将使用新策略 JSON 文档更新网络策略：

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type network \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

删除网络策略

必须先将网络策略与所有集合分离，然后才能删除网络策略。要在 OpenSearch 无服务器控制台中删除策略，请选择该策略并选择删除。

你也可以使用以下DeleteSecurityPolicy命令：

```
aws opensearchserverless delete-security-policy --name my-policy --type network
```

Amazon OpenSearch 无服务器的数据访问控制

借助 Amazon OpenSearch Serverless 中的数据访问控制，您可以允许用户访问集合和索引，无论其访问机制或网络来源如何。您可以提供对 IAM 角色和 SAML 身份的访问权限。

您可以通过适用于集合和索引资源的数据访问策略管理访问权限。数据访问策略通过自动为匹配某个具体模式的集合和索引分配访问权限，帮助您大规模管理集合。多个数据访问策略可应用于一个资源。请注意，您必须为馆藏制定数据访问策略才能访问 OpenSearch 控制面板 URL。

主题

- 数据访问策略与 IAM policy 对比
- 配置数据访问策略所必需的 IAM 权限
- 策略语法
- 受支持的策略权限
- OpenSearch 仪表板上的示例数据集
- 创建数据访问策略（控制台）
- 创建数据访问策略（Amazon CLI）
- 查看数据访问策略
- 更新数据访问策略
- 删除数据访问策略
- 跨账户数据访问

数据访问策略与 IAM policy 对比

数据访问策略在逻辑上与 Amazon Identity and Access Management (IAM) 策略是分开的。IAM 权限控制针对无服务器 API 操作的访问权限，如 `CreateCollection` 和 `ListAccessPolicies`。数据访问策略控制对 OpenSearch Serverless 支持的OpenSearch 操作（例如`PUT <index>`或`GET _cat/indices`）的访问权限。

控制数据访问策略 API 操作访问权限的 IAM 权限，如 `aoss:CreateAccessPolicy` 和（`aoss:GetAccessPolicy`将在下一节中介绍），不会影响数据访问策略中指定的权限。

例如，假设 IAM policy 拒绝用户为 `collection-a` 创建数据访问策略，但允许他们为所有集合 (*) 创建数据访问策略：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "collection-a"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

如果用户创建了一个数据访问策略，该策略允许针对所有集合（`collection/*` 或 `index/*/*`）的某些权限，则该策略将适用于所有集合，包括集合 A。

> **Important**
>
> 在数据访问策略中获得权限不足以访问您的 OpenSearch Serverless 集合中的数据。还必须授予关联主体访问 IAM 权限 `aoss:APIAccessAll` 和 `aoss:DashboardsAccessAll` 的权限。这两个权限都授予对集合资源的完全访问权限，而仪表板权限还提供对 OpenSearch 仪表板的访问权限。如果主体没有这两种 IAM 权限，则其在尝试向集合发送请求时将收到 403 错误。有关更多信息，请参阅 the section called “使用 OpenSearch API 操作”。

配置数据访问策略所必需的 IAM 权限

OpenSearch 无服务器的数据访问控制使用以下 IAM 权限。您可以指定 IAM 条件，以将用户限制到特定访问策略名称。

- `aoss:CreateAccessPolicy`：创建访问策略。
- `aoss:ListAccessPolicies`：列出所有访问策略。
- `aoss:GetAccessPolicy`：查看有关某个具体访问策略的详细信息。
- `aoss:UpdateAccessPolicy`：修改访问策略。
- `aoss:DeleteAccessPolicy`：删除访问策略。

以下基于身份的访问策略允许用户查看所有访问策略，并更新包含资源模式 `collection/logs` 的策略。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:GetAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "aoss:UpdateAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": [
                        "logs"
                    ]
                }
```

```
            }
        }
    ]
}
```

> **Note**
>
> 此外， OpenSearch Serverless 需要`aoss:APIAccessAll`和`aoss:DashboardsAccessAll`权限才能使用集合资源。有关更多信息，请参阅 the section called “使用 OpenSearch API 操作”。

策略语法

数据访问策略包括一组规则，每条规则均包含以下元素：

元素	描述
`ResourceType`	权限适用于的资源类型（集合或索引）。别名和模板权限位于集合级别，而创建、修改和搜索数据的权限位于索引级别。有关更多信息，请参阅受支持的策略权限。
`Resource`	资源名称 and/or 模式列表。模式是前缀后跟通配符 (*)，它们允许将关联权限应用于多个资源。 • 集合采用 `collection/` *`<name|pattern>`* 格式。 • 索引采用 `index/`*`<collection-name|pattern>`* `/`*`<index-name|pattern/>`* 格式。
`Permission`	要为指定资源授予的权限的列表。有关权限以及它们允许执行的 API 操作的完整列表，请参阅the section called “受支持 OpenSearch 的 API 操作和权限”。
`Principal`	要向其授予访问权限的一个或多个主体的列表。委托人可以是 IAM 角色 ARNs 或 SAML 身份。这些主体必须位于当前 Amazon Web Services 账户内。数据访问策略不直接支持跨账户访问，但您可以在策略中加入一个角色，让其他 Amazon Web Services 账户 用户在拥有馆藏的账户中担任该角色。有关更多信息，请参阅 the section called “跨账户数据访问”。

以下示例策略授予针对名为 autopartsinventory 的集合以及任何以前缀 sales* 开头的集合的别名和模板权限。它还授予针对 autopartsinventory 集合内所有索引以及以前缀 orders* 开头的 salesorders 集合内任何索引的读写权限。

```
[
   {
      "Description": "Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/autopartsinventory",
               "collection/sales*"
            ],
            "Permission":[
               "aoss:CreateCollectionItems",
               "aoss:UpdateCollectionItems",
               "aoss:DescribeCollectionItems"
            ]
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/autopartsinventory/*",
               "index/salesorders/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/Dale",
         "arn:aws:iam::123456789012:role/RegulatoryCompliance",
         "saml/123456789012/myprovider/user/Annie",
         "saml/123456789012/anotherprovider/group/Accounting"
      ]
   }
]
```

您不能在策略中显式拒绝访问权限。因此，所有策略权限都具有累加性。例如，如果一个策略授予用户 aoss:ReadDocument，而另一个策略授予 aoss:WriteDocument，则该用户将同时拥有这两种权

限。如果第三个策略授予同一用户 `aoss:*`，则该用户可以针对关联索引执行所有操作；限制性较大的权限不会覆盖限制性较小的权限。

受支持的策略权限

数据访问策略中支持以下权限。有关每种权限允许的 OpenSearch API 操作，请参阅the section called “受支持 OpenSearch 的 API 操作和权限”。

集合权限

- `aoss:CreateCollectionItems`
- `aoss:DeleteCollectionItems`
- `aoss:UpdateCollectionItems`
- `aoss:DescribeCollectionItems`
- `aoss:*`

索引权限

- `aoss:ReadDocument`
- `aoss:WriteDocument`
- `aoss:CreateIndex`
- `aoss:DeleteIndex`
- `aoss:UpdateIndex`
- `aoss:DescribeIndex`
- `aoss:*`

OpenSearch 仪表板上的示例数据集

OpenSearch 仪表板提供了带有可视化效果、仪表板和其他工具的示例数据集，可帮助您在添加自己的数据之前浏览仪表板。要根据此示例数据创建索引，您需要一个数据访问策略，该策略提供对要使用的数据集的权限。以下策略使用通配符 (*) 来提供对所有三个示例数据集的权限。

```
[
  {
    "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>"
    ]
  }
]
```

创建数据访问策略（控制台）

您可以使用可视化编辑器或以 JSON 格式创建数据访问策略。在创建集合时，将为与该策略中定义的模式之一相匹配的任何新集合分配相应的权限。

创建 OpenSearch 无服务器数据访问策略

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，展开 Serverless，然后在 “安全” 下，选择 “数据访问策略”。
3. 选择 Create access policy（创建访问策略）。
4. 为策略提供名称和描述。
5. 为策略中的第一条规则提供名称。例如，Logs collection access（日志集合访问权限）。
6. 选择 Add principals（添加主体），然后选择一个或多个 IAM 角色或 SAML 用户和组，授予其数据访问权限。

   > **Note**
   >
   > 要从下拉菜单中选择主体，您必须拥有 `iam:ListUsers` 和 `iam:ListRoles` 权限（对于 IAM 主体）和 `aoss:ListSecurityConfigs` 权限（对于 SAML 身份）。

7. 选择 Grant（授予），然后选择要授予关联主体的别名、模板和索引权限。有关权限及其允许的访问权限的完整列表，请参阅 the section called “受支持 OpenSearch 的 API 操作和权限”。

8. （可选）为策略配置其他规则。
9. 选择创建。在创建策略与强制执行权限之间，可能会有大约一分钟的延迟。如果该延迟超过 5 分钟，请联系 Amazon Web Services 支持。

> **Important**
>
> 如果策略仅包含索引权限（不包含集合权限），您可能仍会看到一条有关匹配集合的消息，内容为 `Collection cannot be accessed yet. Configure data access policies so that users can access the data within this collection`。您可以忽略此警告。允许的主体仍然可以在集合上执行其分配的索引相关操作。

创建数据访问策略（Amazon CLI）

要使用 OpenSearch 无服务器 API 创建数据访问策略，请使用`CreateAccessPolicy`命令。该命令同时接受内联策略和 .json 文件。必须以 JSON 转义字符串的形式编码内联策略。

以下请求将创建数据访问策略：

```
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\"]}]"
```

要在 .json 文件中提供策略，请使用 `--policy file://my-policy.json` 格式。

策略中包含的委托人现在可以使用他们被授予访问权限的OpenSearch 操作。

查看数据访问策略

在创建集合之前，您可能想预览账户中的现有数据访问策略，以查看哪个数据访问策略的资源模式与您的集合名称相匹配。以下ListAccessPolicies请求列出了您账户中的所有数据访问政策：

```
aws opensearchserverless list-access-policies --type data
```

该请求将返回有关所有已配置的数据访问策略的信息。要查看某一具体策略中定义的模式规则，请在响应的 `accessPolicySummaries` 元素内容中查找策略信息。请注意本政策type的name和，并在GetAccessPolicy请求中使用这些属性来接收包含以下策略详细信息的响应：

```
{
    "accessPolicyDetails": [
        {
            "type": "data",
            "name": "my-policy",
            "policyVersion": "MTY2NDA1NDE4MDg1OF8x",
            "description": "My policy",
            "policy": "[{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection/autopartsinventory\",\"collection/sales*\"],
\"Permission\":[\"aoss:UpdateCollectionItems\"]},{\"ResourceType\":\"index\",
\"Resource\":[\"index/autopartsinventory/*\",\"index/salesorders/orders*\"],
\"Permission\":[\"aoss:ReadDocument\",\"aoss:DescribeIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:user/Shaheen\"]}]",
            "createdDate": 1664054180858,
            "lastModifiedDate": 1664054180858
        }
    ]
}
```

您可以包括资源筛选器，将结果限制为包含特定集合或索引的策略：

```
aws opensearchserverless list-access-policies --type data --resource
 "index/autopartsinventory/*"
```

要查看有关特定策略的详细信息，请使用GetAccessPolicy命令。

更新数据访问策略

在更新数据访问策略时，所有关联集合都将受到影响。要在 OpenSearch Serverless 控制台中更新数据访问策略，请选择数据访问控制，选择要修改的策略，然后选择编辑。进行更改，然后选择保存。

要使用 OpenSearch 无服务器 API 更新数据访问策略，`UpdateAccessPolicy`请发送请求。必须包括策略版本，您可以使用 `ListAccessPolicies` 或 `GetAccessPolicy` 命令检索策略版本。包括最新策略版本可以确保您不会无意中覆盖其他人所做的更改。

以下UpdateAccessPolicy请求使用新的策略 JSON 文档更新数据访问策略：

```
aws opensearchserverless update-access-policy \
```

```
    --name sales-inventory \
    --type data \
    --policy-version MTY2NDA1NDE4MDg10F8x \
    --policy file://my-new-policy.json
```

在更新策略与强制执行新权限之间，可能会有几分钟的延迟。

删除数据访问策略

在删除数据访问策略时，所有关联集合都将失去该策略中定义的访问权限。在删除策略之前，请确保您的 IAM 和 SAML 用户拥有针对集合的适当访问权限。要在 OpenSearch 无服务器控制台中删除策略，请选择该策略并选择删除。

你也可以使用以下DeleteAccessPolicy命令：

```
aws opensearchserverless delete-access-policy --name my-policy --type data
```

跨账户数据访问

虽然无法创建具有跨账户身份或跨账户集合的数据访问策略，但仍然可以通过代入角色选项来设置跨账户访问。例如，假设 *account-a* 拥有一个 *account-b* 需要访问的集合，则 *account-b* 中的用户可以代入 *account-a* 中的角色。该角色必须具有 IAM 权限 `aoss:APIAccessAll` 和 `aoss:DashboardsAccessAll`，并且必须包含在 *account-a* 上的数据访问策略中。

使用接口终端节点访问 Amazon OpenSearch Serverless ()Amazon PrivateLink

您可以使用 Amazon PrivateLink 在您的 VPC 和 Amazon OpenSearch Serverless 之间创建私有连接。您可以像在 VPC 中一样访问 OpenSearch 无服务器，无需使用互联网网关、NAT 设备、VPN 连接或 Amazon Direct Connect 连接。您的 VPC 中的实例不需要公有 IP 地址即可访问 OpenSearch 无服务器。有关 VPC 网络访问的更多信息，请参阅 Amazon OpenSearch Serverless 的网络连接模式。

您可以通过创建由 Amazon PrivateLink提供支持的接口端点来建立此私有连接。我们将在您为接口端点指定的每个子网中创建一个端点网络接口。这些是请求者管理的网络接口，用作发往 Serverless 的流量的入口点。 OpenSearch

有关更多信息，请参阅《Amazon PrivateLink 指南》中的通过 Amazon PrivateLink访问 Amazon Web Services 服务。

主题

- 集合端点 DNS 解析
- VPCs 和网络访问策略
- VPCs 和端点策略
- 注意事项
- 所需权限
- 为 OpenSearch 无服务器创建接口端点
- Amazon OpenSearch 无服务器的共享 VPC 设置

集合端点 DNS 解析

当您创建 VPC 终端节点时，该服务会创建一个新的 Amazon Route 53 私有托管区域并将其连接到 VPC。此私有托管区域包含一条记录，用于将 OpenSearch 无服务器集合的通配符 DNS 记录 (`*.aoss.us-east-1.amazonaws.com`) 解析为用于终端节点的接口地址。您只需要一个 VPC 中的一个 OpenSearch 无服务器 VPC 终端节点即可访问每个 Amazon Web Services 区域 VPC 中的所有集合和控制面板。每个带有 S OpenSearch erverless 终端节点的 VPC 都有自己的私有托管区域。

OpenSearch Serverless 还会为该地区的所有集合创建一条公共 Route 53 通配符 DNS 记录。DNS 名称解析为 OpenSearch 无服务器公有 IP 地址。中 VPCs 没有 OpenSearch 无服务器 VPC 终端节点的客户端或公共网络中的客户端可以使用公共 Route 53 解析器，并使用这些 IP 地址访问集合和控制面板。VPC 终端节点的 IP 地址类型（IPv4 IPv6、或 Dualstack）是根据您为无服务器创建接口终端节点时提供的子网确定的。 OpenSearch

> **Note**
>
> OpenSearch Serverless 会为服务域解析创建额外的 Amazon Route 53 私有托管区域 `<region>.opensearch.amazonaws.com`（`）。 OpenSearch 您可以使用中的update-vpc-endpoint命令将现有 IPv4 VPC 终端节点更新为 Dualstack。 Amazon CLI

给定 VPC 的 DNS 解析器地址是 VPC CIDR 的辅助 IP 地址。VPC 中的任何客户端均需使用该解析器获取任何集合的 VPC 端点地址。解析器使用由 OpenSearch Serverless 创建的私有托管区域。使用该解析器足以处理任何账户中的所有集合。也可以对一些集合端点使用 VPC 解析器，对另一些集合端点使用公有解析器，尽管通常并不需要这样做。

VPCs 和网络访问策略

要向您的集合 OpenSearch APIs 和仪表板授予网络权限，您可以使用 OpenSearch 无服务器网络访问策略。您可以从 VPC 端点或公有 Internet 控制此网络访问。由于网络策略仅控制流量权限，因此还必须设置数据访问策略，指定对集合及其索引中的数据执行操作的权限。将 OpenSearch 无服务器 VPC 终端节点视为服务的接入点，将网络访问策略视为集合和仪表板的网络级访问点，将数据访问策略视为对集合中数据进行任何操作的精细访问控制的接入点。

由于您可以在网络策略 IDs 中指定多个 VPC 终端节点，因此我们建议您为每个需要访问集合的 VPC 创建一个 VPC 终端节点。这些 Amazon 账户 VPCs可以属于与拥有 OpenSearch Serverless 集合和网络策略的账户不同的账户。我们不建议您在两个账户之间创建 VPC-to-VPC对等或其他代理解决方案，以便一个账户的 VPC 可以使用另一个账户的 VPC 终端节点。相较于每个 VPC 都有自己的端点，这种做法的安全性和成本效益会有所下降。即使其他 VPC 的管理员已在网络策略中设置对第一个 VPC 端点的访问权限，也无法轻易查看第一个 VPC。

VPCs 和端点策略

Amazon OpenSearch Serverless 支持以下方面的终端节点策略。 VPCs终端节点策略是您附加到 VPC 终端节点的基于 IAM 资源的策略，用于控制哪些 Amazon 委托人可以使用该终端节点访问您的 Amazon 服务。有关更多信息，请参阅使用端点策略控制对 VPC 端点的访问。

要使用端点策略，必须先创建接口端点。您可以使用无服务器控制台或 OpenSearch 无服务器 API 创建接口端点。 OpenSearch 创建接口端点后，您需要将端点策略添加到端点。有关更多信息，请参阅使用接口终端节点访问 Amazon OpenSearch Serverless（Amazon PrivateLink）。

> **Note**
>
> 您不能直接在 OpenSearch 服务控制台中定义终端节点策略。

端点策略不会覆盖或取代您可能已配置的其他基于身份的策略、基于资源的策略、网络策略或数据访问策略。有关如何更新端点策略的更多信息，请参阅使用端点策略控制对 VPC 端点的访问。

默认情况下，端点策略授予对 VPC 端点的完全访问权限。

```
{
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

尽管默认 VPC 端点策略授予端点完全访问权限，但您可以配置 VPC 端点策略以允许访问特定角色和用户。为此，请参阅以下示例：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012",
                    "987654321098"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

您可以指定一个 OpenSearch 无服务器集合作为条件元素包含在您的 VPC 终端节点策略中。为此，请参阅以下示例：

JSON

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

```
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": [
                        "coll-abc"
                    ]
                }
            }
        }
    ]
}
```

支持 S `aoss:CollectionId` upport

```
Condition": {
         "StringEquals": {
               "aoss:CollectionId": "collection-id"
          }
}
```

您可以在 VPC 端点策略中使用 SAML 身份来确定 VPC 端点访问权限。您必须在 VPC 端点策略的主体部分使用通配符 (`*`)。为此，请参阅以下示例：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:SamlGroups": [
                        "saml/123456789012/idp123/group/football",
                        "saml/123456789012/idp123/group/soccer",
                        "saml/123456789012/idp123/group/cricket"
                    ]
```

```
                }
            }
        }
    ]
}
```

此外，您可以将端点策略配置为包含特定的 SAML 主体策略。为此，请参阅以下内容：

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:SamlPrincipal": [
                        "saml/123456789012/idp123/user/user1234"]
                    }
                }
            }
        ]
    }
```

有关在 Amazon Serverless 中使用 SAML 身份验证的更多信息，请参阅 Amazon OpenSearch Serverless 的 SAML 身份验证。 OpenSearch

您还可以在同一 VPC 端点策略中包含 IAM 和 SAML 用户。为此，请参阅以下示例：

JSON

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

```
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aoss:SamlGroups": [
                        "saml/123456789012/idp123/group/football",
                        "saml/123456789012/idp123/group/soccer",
                        "saml/123456789012/idp123/group/cricket"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

您也可以 EC2 通过接口 VPC 终端节点从亚马逊访问亚马逊的 OpenSearch 无服务器集合。有关更多信息，请参阅从 Amazon 访问 OpenSearch 无服务器集合 EC2（通过接口 VPC 终端节点）。

注意事项

在为 OpenSearch Serverless 设置接口终端节点之前，请考虑以下事项：

- OpenSearch Serverless 支持通过接口端点调用所有支持OpenSearch 的 API 操作（不是配置 API 操作）。
- 为 OpenSearch Serverless 创建接口终端节点后，您仍需要将其包含在网络访问策略中，以便它能够访问无服务器集合。
- 默认情况下，允许通过接口端点对 OpenSearch Serverless 进行完全访问。您可以将安全组与端点网络接口关联，以控制通过接口终端节点流向 OpenSearch Serverless 的流量。

- 单个最多 Amazon Web Services 账户 可以有 50 个 OpenSearch 无服务器 VPC 终端节点。
- 如果网络策略启用通过公有 Internet 访问集合 API 或控制面板，则任何 VPC 和公有 Internet 均可访问您的集合。
- 如果您位于本地且在 VPC 之外，则不能直接使用 DNS 解析器进行 OpenSearch 无服务器 VPC 终端节点解析。如果需要 VPN 访问权限，则 VPC 需要 DNS 代理解析程序供外部客户端使用。Route 53 提供入站端点选项，您可以使用此选项解析从本地网络或其他 VPC 对您的 VPC 执行的 DNS 查询。
- OpenSearch Serverless 创建并连接到 VPC 的私有托管区域由服务管理，但它会显示在您的 Amazon Route 53 资源中并计入您的账户。
- 有关其他注意事项，请参阅 Amazon PrivateLink 指南中的注意事项。

所需权限

OpenSearch 无服务器的 VPC 访问使用以下 Amazon Identity and Access Management (IAM) 权限。您可以指定 IAM 条件，以将用户限制到特定集合。

- `aoss:CreateVpcEndpoint`：创建 VPC 端点。
- `aoss:ListVpcEndpoints`：列出所有 VPC 端点。
- `aoss:BatchGetVpcEndpoint`：查看有关 VPC 端点子集的详细信息。
- `aoss:UpdateVpcEndpoint`：修改 VPC 端点。
- `aoss:DeleteVpcEndpoint`：删除 VPC 端点。

此外，您需要以下 Amazon EC2 和 Route 53 权限才能创建 VPC 终端节点。

- `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`

为 OpenSearch 无服务器创建接口端点

您可以使用控制台或无服务器 API 为 OpenSearch 无服务器创建接口终端节点。 OpenSearch

为 OpenSearch 无服务器集合创建接口端点

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，展开 Serverless（无服务器），然后选择 VPC endpoints（VPC 端点）。
3. 选择 Create VPC endpoint（创建 VPC 端点）。
4. 为端点提供名称。
5. 对于 VPC，请选择您要从中访问 OpenSearch 无服务器的 VPC。
6. 对于子网，请选择一个您要从中访问 OpenSearch 无服务器的子网。
   - 端点的 IP 地址和 DNS 类型取决于子网类型
     - Dualstack：如果所有子网都同时具有和地址范围 IPv4 IPv6
     - IPv6: 如果所有子网都 IPv6 只是子网
     - IPv4: 如果所有子网都有 IPv4 地址范围
7. 对于 Security groups（安全组），选择要与端点网络接口关联的安全组。这是一个关键步骤，您可以在该步骤中限制您授权进入端点的入站流量的端口、协议和源。确保安全组规则允许将使用 VPC 终端节点与 OpenSearch Serverless 通信的资源与终端节点网络接口通信。
8. 选择创建端点。

要使用 OpenSearch 无服务器 API 创建 VPC 终端节点，请使用`CreateVpcEndpoint`命令。

> **Note**
>
> 在创建端点后，记下其 ID（例如，`vpce-abc123def4EXAMPLE`）。为给端点提供针对集合的访问权限，您必须将此 ID 包含在一个或多个网络访问策略中。

在创建接口端点后，必须通过网络访问策略为其提供针对集合的访问权限。有关更多信息，请参阅 the section called “网络访问”。

Amazon OpenSearch 无服务器的共享 VPC 设置

您可以使用亚马逊虚拟私有云 (VPC) Virtual Private Cloud 与组织中的 Amazon Web Services 账户 其他人共享 VPC 子网，也可以在多个资源之间共享网络基础设施，例如 VPN。 Amazon Web Services 账户

目前，除非您是共享 VPC 的所有者，否则 Amazon OpenSearch Serverless 不支持创建与共享 VPC 的 Amazon PrivateLink 连接。 Amazon PrivateLink 也不支持在之间共享连接 Amazon Web Services 账户。

但是，基于 OpenSearch Serverless 灵活的模块化架构，您仍然可以设置共享 VPC。这是因为 OpenSearch 无服务器网络基础设施与单个集合（OpenSearch 服务）基础架构是分开的。因此，您可以为一个 VPC 所在的账户创建一个 Amazon PrivateLink VPCe 终端节点，然后在其他账户的网络策略中使用一个 VPCe ID 来限制仅来自该共享 VPC 的流量。

以下过程涉及所有者账户和消费者账户。

所有者账户充当普通网络账户，您可以在其中设置 VPC 并与其他账户共享。消费者账户是在所有者账户与其共享的 VPC 中创建和维护其 OpenSearch 无服务器集合的账户。

先决条件

在设置共享 VPC 之前，请确保满足以下要求：

- 目标所有者账户必须已经在 Amazon Virtual Private Cloud 中设置了 VPC、子网、路由表和其他必需的资源。有关更多信息，请参阅《Amazon VPC 用户指南》。
- 目标所有者账户和消费者账户必须属于中的同一个组织 Amazon Organizations。有关更多信息，请参阅 Amazon Organizations 《用户指南》。

在所有者 account/common 网络账户中设置共享 VPC。

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 按照the section called “创建接口端点”中的步骤操作。与您一样，请进行以下选择：
   - 选择与组织中的使用者账户共享的 VPC 和子网。
3. 创建终端节点后，记下生成的 VPCe ID，并将其提供给要在使用者帐户中执行设置任务的管理员。

   VPCe IDs 是格式的`vpce-abc123def4EXAMPLE`。

在消费者账户中设置共享 VPC

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 如果您还没有收藏夹，请使用中的the section called “管理集合”信息创建收藏夹。
3. 使用中的信息the section called “创建网络策略（控制台）”创建网络策略。与您一样，请进行以下选择。

   > **Note**
   > 您也可以为此更新现有的网络策略。

   a. 对于访问类型，选择 VPC（推荐）。
   b. 对于用于访问的 VPC 终端节点，请选择所有者账户向您提供的 VPCe ID，格式为`vpce-abc123def4EXAMPLE`。
   c. 在 “资源类型” 区域中，执行以下操作：
      - 选中 “允许访问 OpenSearch终端节点” 复选框，然后选择集合名称或集合模式以允许从该共享 VPC 进行访问。
      - 选中 “允许访问 OpenSearch控制面板” 复选框，然后选择用于启用从该共享 VPC 进行访问的集合名称或集合模式。
4. 要创建新策略，请选择创建。对于现有策略，请选择更新。

适用于 Amazon 无服务器的 SAM OpenSearch L 身份验证

借助 Amazon S OpenSearch erverless 的 SAML 身份验证，您可以使用现有的身份提供商为无服务器集合的 OpenSearch 控制面板终端节点提供单点登录 (SSO)。

SAML 身份验证允许您使用第三方身份提供商登录 OpenSearch 仪表板以索引和搜索数据。 OpenSearch Serverless 支持使用 SAML 2.0 标准的提供商，例如 IAM 身份中心、Okta、Keycloak、Active Directory 联合身份验证服务 (AD FS) 和 Auth0。您可以将 IAM Identity Center 配置为同步来自其他身份来源（例如 Okta 和 Microsoft Entra ID）的用户和群组。 OneLogin有关 IAM Identity Center 支持的身份来源列表及其配置步骤，请参阅《IAM Identity Center 用户指南》中的 Getting started tutorials。

> **Note**
>
> SAML 身份验证仅适用于通过 Web 浏览器访问 OpenSearch 仪表板。经过身份验证的用户只能通过 OpenSearch 仪表板中的开发工具向 OpenSearch API 操作发出请求。您的 SAML 凭证不允许您直接向 OpenSearch API 操作发出 HTTP 请求。

要设置 SAML 身份验证，首先需要配置 SAML 身份提供者（IdP）。然后，您可以在数据访问策略中包括来自该 IdP 的一个或多个用户。此策略向其授予对集合 and/or 索引的某些权限。然后，用户可以登录 OpenSearch 仪表板并执行数据访问策略中允许的操作。

主题

- 注意事项
- 所需权限
- 创建 SAML 提供者（控制台）
- 访问 OpenSearch 仪表板
- 授予 SAML 身份针对集合数据的访问权限

- 创建 SAML 提供者（Amazon CLI）
- 查看 SAML 提供者
- 更新 SAML 提供者
- 删除 SAML 提供者

注意事项

在配置 SAML 身份验证时，请考虑以下事项：

- 不支持已签名和已加密的请求。
- 不支持已加密的断言。
- 不支持 IdP 发起的身份验证和注销。
- 对于非 IAM 身份（例如 Amazon S OpenSearch erverless 和 SAML 中的 SAML 以及亚马逊服务的基本内部用户授权），服务控制策略 (SCP) 将不适用或不进行评估。 OpenSearch

所需权限

OpenSearch 无服务器的 SAML 身份验证使用以下 Amazon Identity and Access Management (IAM) 权限：

- `aoss:CreateSecurityConfig`：创建 SAML 提供者。
- `aoss:ListSecurityConfig`：列出当前账户中的所有 SAML 提供者。
- `aoss:GetSecurityConfig`：查看 SAML 提供者信息。
- `aoss:UpdateSecurityConfig`：修改给定的 SAML 提供者配置，包括 XML 元数据。
- `aoss:DeleteSecurityConfig`：删除 SAML 提供者。

以下基于身份的访问策略允许用户管理所有 IdP 配置：

JSON

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

```
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

请注意，`Resource` 元素必须是通配符。

创建 SAML 提供者（控制台）

这些步骤说明了如何创建 SAML 提供者。这为仪表板启用了使用服务提供商 (SP) 启动的身份验证的 SAML 身份验证。 OpenSearch 不支持 IdP 发起的身份验证。

为仪表板启用 SAML 身份验证 OpenSearch

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航面板上，展开 Serverless（无服务器），然后选择 SAML authentication（SAML 身份验证）。
3. 选择 Add SAML provider（添加 SAML 提供者）。
4. 为提供者提供名称和描述。

> **Note**
>
> 您指定的名称可公开访问，当用户登录 OpenSearch 控制面板时，该名称将显示在下拉菜单中。确保该名称易于识别，并且不会泄露有关您的身份提供者的敏感信息。

5. 在 Configure your IdP（配置您的 IdP）下，复制断言使用者服务（ACS）URL。
6. 使用您刚刚复制的 ACS URL 配置您的身份提供者。术语和步骤因提供者而异。请参阅提供程序的文档。

   例如，在 Okta 中，您可以创建 SAML 2.0 web application（SAML 2.0 Web 应用程序），并将 ACS URL 指定为 Single Sign On URL（单点登录 URL）、Recipient URL（收件人 URL）和 Destination URL（目标 URL）。对于 Auth0，您可以在 “允许的回调 URLs” 中指定它。

7. 如果您的 IdP 有用于受众限制的字段，则请提供受众限制。受众限制是 SAML 断言中的一个值，用于指定断言适用于哪些受众。对于 OpenSearch 无服务器，请指定`aws:opensearch:<aws account id>`。例如 `aws:opensearch:123456789012`。

   受众限制字段的名称因提供者而异。对于 Okta，该字段为 Audience URI (SP Entity ID)（受众 URI（SP 实体 ID））。对于 IAM Identity Center，该字段为 Application SAML audience（应用程序 SAML 受众）。

8. 如果您使用的是 IAM Identity Center，则还需要指定以下属性映射：`Subject=${user:name}`，格式为 `unspecified`。

9. 配置了身份提供程序后，它会生成 IdP 元数据文件。此 XML 文件包含有关提供者的信息，如 TLS 证书、单点登录端点和身份提供者的实体 ID。

   复制 IdP 元数据文件中的文本，并将其粘贴到 Provide metadata from your IdP（提供来自您的 IdP 的元数据）字段下。也可以选择从 XML 文件导入并上载文件。元数据文件应如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>s
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST" Location="idp-sso-url"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

10. 保持自定义用户 ID 属性字段为空，以将 SAML 断言的 `NameID` 元素用于用户名。如果您的断言不使用此标准元素，而是将用户名作为自定义属性，请在此处指定该属性。属性区分大小写。仅支持一个用户属性。

以下示例显示了 SAML 断言中 NameID 的覆盖属性：

```
<saml2:Attribute Name="UserId" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
  <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="xs:string">annie</saml2:AttributeValue>
</saml2:Attribute>
```

11. （可选）在 Group attribute（组属性）字段中指定自定义属性，如 role 或 group。仅支持一个组属性。没有默认群组属性。如果您未指定组属性，则您的数据访问策略只能包含用户主体。

    以下示例显示了 SAML 断言中的组属性：

```
<saml2:Attribute Name="department"
 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
    <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:string">finance</saml2:AttributeValue>
</saml2:Attribute>
```

12. 默认情况下， OpenSearch 控制面板会在 24 小时后注销用户。您可以通过指定OpenSearch 仪表板超时将此值配置为 1 到 12 小时（15 到 720 分钟）之间的任意数字。如果您尝试将超时设置为等于或小于 15 分钟，您的会话将重置为一小时。
13. 选择 Create SAML provider（创建 SAML 提供者）。

访问 OpenSearch 仪表板

配置 SAML 提供商后，与该提供商关联的所有用户和群组都可以导航到 OpenSearch 仪表板端点。所有集合的控制面板 URL 格式为 *collection-endpoint*/_dashboards/。

如果您启用了 SAML，则选择中的链接 Amazon Web Services Management Console 会将您定向到 IdP 选择页面，您可以在其中使用 SAML 凭据登录。首先，使用下拉列表选择身份提供者：

然后，使用您的 IdP 凭证登录。

如果您未启用 SAML，则选择中的链接 Amazon Web Services Management Console 将引导您以 IAM 用户或角色的身份登录，但没有 SAML 选项。

授予 SAML 身份针对集合数据的访问权限

在您创建 SAML 提供者后，您仍然需要向基础用户和组授予针对集合中数据的访问权限。您可以通过数据访问策略授予访问权限。在您向用户提供访问权限之前，他们将无法读取、写入或删除您的集合中的任何数据。

要授予访问权限，请创建数据访问策略并在`Principal`语句 IDs 中指定您的 SAML 用户 and/or 组：

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

```
        ...
      ],
      "Principal":[
         "saml/987654321098/myprovider/user/Shaheen",
         "saml/987654321098/myprovider/group/finance"
      ]
   }
]
```

您可以授予针对集合、索引或两者的访问权限。如果您希望不同用户拥有不同权限，请创建多条规则。有关可用权限的列表，请参阅受支持的策略权限。有关如何设置访问策略格式的信息，请参阅策略语法。

创建 SAML 提供者（Amazon CLI）

要使用 OpenSearch 无服务器 API 创建 SAML 提供商，请发送请求：CreateSecurityConfig

```
aws opensearchserverless create-security-config \
    --name myprovider \
    --type saml \
    --saml-options file://saml-auth0.json
```

将包括元数据 XML 在内的 `saml-options` 指定为 .json 文件中的键值映射。必须以 JSON 转义字符串的形式编码元数据 XML。

```
{
   "sessionTimeout": 70,
   "groupAttribute": "department",
   "userAttribute": "userid",
   "metadata": "<EntityDescriptor xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ... ... ... IDPSSODescriptor>\r\n<\/EntityDescriptor>"
}
```

查看 SAML 提供者

以下ListSecurityConfigs请求列出了您账户中的所有 SAML 提供商：

```
aws opensearchserverless list-security-configs --type saml
```

该请求返回有关所有现有 SAML 提供者的信息，包括您的身份提供者生成的完整 IdP 元数据：

```
{
   "securityConfigDetails": [
      {
         "configVersion": "MTY2NDA1MjY4NDQ5M18x",
         "createdDate": 1664054180858,
         "description": "Example SAML provider",
         "id": "saml/123456789012/myprovider",
         "lastModifiedDate": 1664054180858,
         "samlOptions": {
            "groupAttribute": "department",
            "metadata": "<EntityDescriptor xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ... ... ... IDPSSODescriptor>\r\n<\/EntityDescriptor>",
            "sessionTimeout": 120,
            "userAttribute": "userid"
         }
      }
   ]
}
```

要查看有关某个具体提供者的详细信息，包括未来更新的 `configVersion`，请发送 `GetSecurityConfig` 请求。

更新 SAML 提供者

要使用 OpenSearch 无服务器控制台更新 SAML 提供商，请选择 SAML 身份验证，选择您的身份提供商，然后选择编辑。您可以修改所有字段，包括元数据和自定义属性。

要通过 OpenSearch Serverless API 更新提供商，UpdateSecurityConfig请发送请求并包含要更新的策略的标识符。还必须包括配置版本，您可以使用 `ListSecurityConfigs` 或 `GetSecurityConfig` 命令检索配置版本。包括最新版本可以确保您不会无意中覆盖其他人所做的更改。

以下请求更新了提供者的 SAML 选项：

```
aws opensearchserverless update-security-config \
    --id saml/123456789012/myprovider \
    --type saml \
    --saml-options file://saml-auth0.json \
    --config-version MTY2NDA1MjY4NDQ5M18x
```

将您的 SAML 配置选项指定为 .json 文件中的键值映射。

> **Important**
>
> 对 SAML 选项的更新不是增量的。如果您在更新时没有为 `SAMLOptions` 对象中的参数指定值，则现有值将被空值覆盖。例如，如果当前配置包含 `userAttribute` 的值，然后您进行了更新但不包括此值，则该值将从配置中移除。在您通过调用 `GetSecurityConfig` 操作进行更新之前，请确保您知道现有值是什么。

删除 SAML 提供者

在您删除 SAML 提供者时，对数据访问策略中关联用户和组的任何引用都将不再起作用。为避免混淆，建议您在删除端点之前，先移除对访问策略中端点的所有引用。

要使用 OpenSearch 无服务器控制台删除 SAML 提供商，请选择身份验证，选择提供商，然后选择删除。

要通过 OpenSearch 无服务器 API 删除提供商，请发送DeleteSecurityConfig请求：

```
aws opensearchserverless delete-security-config --id saml/123456789012/myprovider
```

Amazon OpenSearch 无服务器的合规性验证

作为多项合规性计划的一部分，第三方审计员将评估 Amazon OpenSearch Serverless 的安全 Amazon 性和合规性。这些计划包括 SOC、PCI 和 HIPAA。

要了解是否在特定合规性计划范围内，请参阅Amazon Web Services 服务 合性计划范围内的，然后选择您感兴趣的合规性计划。 Amazon Web Services 服务 有关一般信息，请参阅合规计划。

您可以使用下载第三方审计报告 Amazon Artifact。有关更多信息，请参阅中的 “下载报告” Amazon Artifact。

您在使用时的合规性责任 Amazon Web Services 服务 由您的数据的敏感性、您公司的合规性目标以及适用的法律法规决定。 Amazon 提供以下资源来帮助实现合规性：

- Security & Compliance：这些解决方案实施指南讨论了架构考虑因素，并提供了部署安全性和合规性功能的步骤。
- 合规性资源 — 此业务手册和指南集合可能适用于您的行业和位置。
- Amazon Config 开发人员指南中的@@ 使用规则评估资源 — Amazon Config 服务评估您的资源配置对内部实践、行业指南和法规的遵循情况。

- Amazon Security Hub— 这 Amazon Web Services 服务 可以全面了解您的安全状态 Amazon。Security Hub 通过安全控制措施评估您的 Amazon 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控制措施的列表，请参阅 Security Hub 控制措施参考。
- GuardDutyAmazon：该通过监控您的 Amazon Web Services 账户环境中是否存在可疑和恶意活动，来 Amazon Web Services 服务 检测您的、工作负载、容器和数据面临的潜在威胁。GuardDuty 可以通过满足某些合规性框架规定的入侵检测要求，来协助您满足各种合规性要求，如 PCI DSS。

标记 Amazon OpenSearch 无服务器集合

标签允许您将任意信息分配给 Amazon OpenSearch Server集合，以便您可以对该信息进行分类和筛选。标签是您或为 Amazon 资源分配的元数据 Amazon 标记。

每个标签均包含一个键 和一个值。对于您分配的标签，需要定义键和值。例如，您可以将键定义为 `stage`，将一个资源的值定义为 `test`。

您可以借助标签来标识和组织 Amazon 资源。许多 Amazon 服务支持标记，因此，您可以将同一标签分配给来不同服务的资源，以指示这些资源是相关的。例如，您可以将相同标签分配给您分配给 Amazon OpenSearch Service 域的 Serv OpenSearch ice。

在 OpenSearch Server中，主要资源是一个集合。您可以使用 OpenSearch 服务控制台、、OpenSearch Serverless API 操作或在 Amazon SDKs 集合中添加、管理和移除标签。 Amazon CLI

所需权限

OpenSearch 无服务器使用以下 Amazon Identity and Access Management Access Analyzer （IAM）权限来标记集合：

- `aoss:TagResource`
- `aoss:ListTagsForResource`
- `aoss:UntagResource`

标记集合（控制台）

控制台是标记集合的最简单方法。

创建标签（控制台）

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 展开左侧导航窗格中的 Serverless（无服务器），然后选择 Collections（集合）。
3. 选择您要将标签添加到的集合，然后转到 Tags（标签）选项卡。
4. 选择 Manage（管理）和 Add new tag（添加新标签）。
5. 输入一个标签键和可选的值。
6. 选择保存。

要删除标签，请按照相同步骤操作并在 Manage tags（管理标签）页面中选择 Remove（删除）。

有关使用控制台处理标签的更多信息，请参阅《Amazon 管理控制台入门指南》中的标签编辑器。

标记集合（Amazon CLI）

要使用为集合添加标签 Amazon CLI，TagResource请发送请求：

```
aws opensearchserverless tag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tags Key=service,Value=aoss Key=source,Value=logs
```

使用ListTagsForResource以下命令查看集合的现有标签：

```
aws opensearchserverless list-tags-for-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
```

使用UntagResource以下命令从集合中移除标签：

```
aws opensearchserverless untag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tag-keys service
```

Amazon OpenSearch 无服务器中受支持的操作和插件

Amazon OpenSearch 无服务器支持各种 OpenSearch 插件，以及中提供的索引、搜索和元数据 API 操作的 OpenSearch子集。您可以将权限包括在数据访问策略中表的左列，以限制对某些操作的访问权限。

主题

- 受支持 OpenSearch 的 API 操作和权限
- 支持的 OpenSearch 插件

受支持 OpenSearch 的 API 操作和权限

下表列出了 OpenSearch 无服务器支持的 API 操作，及其相应的数据访问策略权限：

数据访问策略权限	OpenSearch API 操作	描述和注意事项
`aoss:CreateIndex`	PUT <index>	创建索引。有关更多信息，请参阅 Create index（创建索引）。 **Note** 此权限也适用于在 OpenSearch 控制面板上使用示例数据创建索引。
`aoss:DescribeIndex`	• GET <index> • GET <index>/_mapping • GET <index>/_mappings • GET <index>/_setting • GET <index>/_setting/<setting> • GET <index>/_settings • GET <index>/_settings/<setting> • GET _cat/indices • GET _mapping • GET _mappings • GET _resolve/index/<index> • HEAD <index>	描述索引。有关更多信息，请参阅以下资源： • 获取索引 • 获取映射 • 获取设置 • Index exists • CAT indices（响应中不含 `health` 或 `status` 字段。）

数据访问策略权限	OpenSearch API 操作	描述和注意事项
`aoss:WriteDocument`	• DELETE <index>/_doc/<id> • POST <index>/_bulk • POST <index>/_create/<id>（仅适用于搜索集合类型） • POST <index>/_doc • POST <index>/_update/<id>（仅适用于搜索集合类型） • POST _bulk • PUT <index>/_create/<id>（仅适用于搜索集合类型） • PUT <index>/_doc/<id>（仅适用于搜索集合类型）	编写和更新文档。有关更多信息，请参阅以下资源： • 批量 • 为数据编制索引 Note 有些操作仅允许针对 SEARCH 类型的集合执行。有关更多信息，请参阅 the section called “选择集合类型”。

数据访问策略权限	OpenSearch API 操作	描述和注意事项
`aoss:ReadDocument`	• 获取 <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/_eplain • 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	阅读文档。有关更多信息，请参阅以下资源： • 执行文本分析 • 获取文档 • 计数 • 查询 DSL • 排名评估 • 分析 API • 解释 • 时间点 • SQL 和 PPL

数据访问策略权限	OpenSearch API 操作	描述和注意事项
	• POST <index>/_search • POST _analyze • POST _field_caps • POST _search • 删除 /_search/point_in_time/_all • 删除 /_search/point_in_time	
`aoss:DeleteIndex`	DELETE <target>	删除索引。有关更多信息，请参阅 Delete index（删除索引）。
`aoss:UpdateIndex`	• POST _mapping • POST <index>/_mapping/ • POST <index>/_mappings/ • POST <index>/_setting • POST <index>/_settings • POST _setting • POST _settings • PUT _mapping • PUT <index>/_mapping • PUT <index>/_mappings/ • PUT <index>/_setting • PUT <index>/_settings • PUT _setting • PUT _settings	更新索引设置。有关更多信息，请参阅以下资源： • 映射 • 更新设置
`aoss:CreateCollectionItems`	POST _aliases	创建索引别名。有关更多信息，请参阅 Create aliases（创建别名）。

数据访问策略权限	OpenSearch API 操作	描述和注意事项
`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> • HEAD _alias/<alias> • HEAD _component_template/<component-template> • HEAD _index_template/<name> • HEAD <index>/_alias/<alias>	描述别名和索引模板。有关更多信息，请参阅以下资源： • 管理别名 • 索引模板

数据访问策略权限	OpenSearch API 操作	描述和注意事项
`aoss:UpdateCollectionItems`	• POST <index>/_alias/<alias> • POST <index>/_aliases/<alias> • POST _component_template/<component-template> • POST _index_template/<index-template> • PUT <index>/_alias/<alias> • PUT <index>/_aliases/<alias> • PUT _component_template/<component-template> • PUT _index_template/<index-template>	更新别名和索引模板。有关更多信息，请参阅以下资源： • 索引别名 • 索引模板
`aoss:DeleteCollectionItems`	• DELETE <index>/_alias/<alias> • DELETE _component_template/<component-template> • DELETE _index_template/<index-template> • DELETE <index>/_aliases/<alias>	删除别名和索引模板。有关更多信息，请参阅以下资源： • 删除别名 • 删除模板

支持的 OpenSearch 插件

OpenSearch serverless-cache OpenSearch 无服务器可为您自动部署和管理插件。

分析插件

- ICU 分析
- 日语 (kuromoji) 分析
- 韩语（Nori）分析
- 拼音分析
- 智能中文分析
- Stempel Polish 分析

- 乌克兰语分析

映射器插件

- 映射器大小
- 映射器 Murmur3
- 映射器带注释的文本

脚本插件

- Painless
- 表达式
- Mustache

此外， OpenSearch 无服务器包括所有以模块形式交付的插件。

监控 Amazon OpenSearch 无服务器

监控是保持 Amazon OpenSearch Serverless 和您的其他 Amazon 解决方案的可靠性、可用性和性能的重要方面。 Amazon 提供以下监控工具来监控 OpenSearch Serverless、在出现错误时进行报告并在适当的时候自动采取措施：

- Amazon 实时 CloudWatch监控您的 Amazon 资源以及 Amazon 在上运行的应用程序。您可以收集和跟踪指标，创建自定义的控制平面，以及设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。

  例如，您可以具有Amazon EC2 实例的 CloudWatch 跟踪 CPU 使用率或其他指标并且在需要时自动启动新实例。有关更多信息，请参阅 Amazon CloudWatch 用户指南。
- Amazon CloudTrail 捕获由某个 Amazon Web Services 账户发出或代表该账户发出的 API 调用和相关事件。它会将日志文件传送到您指定的 Amazon S3 桶。您可以标识哪些用户和账户调用了 Amazon、从中发出调用的源 IP 地址以及调用的发生时间。有关更多信息，请参阅 Amazon CloudTrail 《用户指南》。
- Amazon EventBridge 提供近乎实时的系统事件流，这些系统事件介绍了您的 OpenSearch 服务域中的变化。您可以创建规则，以监控某些事件和在这些事件发生 Amazon Web Services 服务 时在其他中触发自动操作。有关更多信息，请参阅 Amazon EventBridge 用户指南。

使用 OpenSearch Amazon 监控无服务器 CloudWatch

您可以使用 CloudWatch监控 Amazon OpenSearch Serverless。Amazon Serverless 会收集原始数据并将其处理为易读且近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。

此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

OpenSearch Serverless 在`AWS/AOSS`命名空间中报告以下指标。

指标	描述
`ActiveCollection`	表示集合是否处于活动状态。值为 1 表示集合处于 `ACTIVE` 状态。此值将在成功创建集合时发出，并且一直保持为 1，直到您删除该集合为止。该指标的值不能为 0。 相关统计数据：最大值 维度：`ClientId`、`CollectionId` 、`CollectionName` 频率：60 秒
`DeletedDocuments`	已删除文档的总数。 相关统计数据：平均值、总计 维度：`ClientId`、`CollectionId` 、`CollectionName` 、`IndexId`、`IndexName` 频率：60 秒
`IndexingOCU`	用于摄取集合数据的 OpenSearch 计算单位（OCUs）的数量。此指标适用于账户级。 相关统计数据：总计 维度：`ClientId` 频率：60 秒

指标	描述
IngestionDataRate	集合或索引的索引速率，以 GiB/秒为单位。此指标仅适用于批量索引请求。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 频率：60 秒
IngestionDocumentErrors	为集合或索引摄取期间的文档错误总数。在成功完成批量索引请求后，编写者将处理该请求，并针对该请求中所有失败的文档发出错误。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 频率：60 秒
IngestionDocumentRate	将文档摄取到集合或索引的每秒速率。此指标仅适用于批量索引请求。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 频率：60 秒

指标	描述
IngestionRequestErrors	集合的批量索引请求错误总数。 OpenSearch 当批量索引请求因任何原因（如身份验证或可用性问题）失败时，Serverless 将发出此指标。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 频率：60 秒
IngestionRequestLatency	针对集合进行批量写入操作的延迟，以秒为单位。 相关统计数据：最小值、最大值、平均值 维度：ClientId、CollectionId 、CollectionName 频率：60 秒
IngestionRequestRate	集合收到的批量写入操作的总数。 相关统计数据：最小值、最大值、平均值 维度：ClientId、CollectionId 、CollectionName 频率：60 秒
IngestionRequestSuccess	针对集合成功执行索引操作的总数。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 频率：60 秒

指标	描述
SearchableDocuments	集合或索引中可搜索文档的总数。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 频率：60 秒
SearchRequestErrors	集合每分钟查询错误的总数。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 频率：60 秒
SearchRequestLatency	针对集合完成搜索操作所需的平均时间，以毫秒为单位。 相关统计数据：最小值、最大值、平均值 维度：ClientId、CollectionId 、CollectionName 频率：60 秒
SearchOCU	用于搜索集合数据的 OpenSearch 计算单位（OCUs）的数量。此指标适用于账户级。 相关统计数据：总计 维度：ClientId 频率：60 秒

指标	描述
SearchRequestRate	每分钟针对集合的搜索请求总数。 相关统计数据：平均值、最大值、总计 维度：ClientId、CollectionId 、CollectionName 频率：60 秒
StorageUsedInS3	使用的 Amazon S3 存储空间量（以字节为单位）。OpenSearch 无服务器在 Amazon S3 中存储索引数据。您必须在一分钟时选择保留期，以获取准确值。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 、IndexId、IndexName 频率：60 秒
2xx, 3xx, 4xx, 5xx	针对产生给定 HTTP 响应代码（2xx、3xx、4xx、5xx）的集合的请求数量。 相关统计数据：总计 维度：ClientId、CollectionId 、CollectionName 频率：60 秒

使用记录 OpenSearch 无服务器 API 调用 Amazon CloudTrail

Amazon OpenSearch Serverless 与 Amazon CloudTrail集成，后者是一项服务，它提供用户、角色或 Amazon 服务在无服务器中所采取的操作的记录。

CloudTrail 以事件的形式捕获 OpenSearch 无服务器的所有 API 调用。捕获的调用包括来自 OpenSearch 服务控制台无服务器部分的调用，以及针对 Ser OpenSearch verless API 操作的代码调用。

如果您创建跟踪，则可以使 Events 持续传输到 Amazon S3 存储桶（包括 OpenSearch 无服务器的事件）。 CloudTrail 如果您不配置跟踪，则仍可在 CloudTrail 控制台的 Event history (事件历史记录) 中查看最新事件。

通过使用收集的信息 CloudTrail，您可以确定向 OpenSearch Serverless 发出了什么请求、发出请求的 IP 地址、发出请求的人、发出请求的时间以及其他详细信息。

要了解更多信息 CloudTrail，请参阅Amazon CloudTrail 用户指南。

OpenSearch 中的无服务器信息 CloudTrail

CloudTrail 在您创建账户 Amazon Web Services 账户 时已在您的账户上启用。当 OpenSearch Serverless 中发生活动时，该活动将记录在事件中，并与其他 Amazon 服务 CloudTrail 事件一同保存在 Event history（事件历史记录）中。您可以在中查看、搜索和下载最新事件 Amazon Web Services 账户。有关更多信息，请参阅事件历史记录查看 CloudTrail 事件。

要持续记录中的事件（包括 OpenSearch Serverless 的事件），请 Amazon Web Services 账户创建跟踪。通过跟踪 CloudTrail ，可将日志文件传送至 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 Amazon Web Services 区域。

此跟踪记录来自 Amazon 分区中的所有区域的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 Amazon 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取操作。有关更多信息，请参阅下列内容：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 为配置 Amazon SNS 通知 CloudTrail
- 从多个区域接收 CloudTrail 日志文件和从多个账户中接收 CloudTrail 日志文件

所有 OpenSearch 无服务器操作都由OpenSearch 无服务器 API 参考记录 CloudTrail 并记录在案。例如，对`CreateCollection``ListCollections`、和`DeleteCollection`操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可以帮助您确定：

- 请求是使用根用户凭证还是 Amazon Identity and Access Management (IAM) 用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其它 Amazon 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 OpenSearch 无服务器日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。

事件表示来自任何源的单个请求。它包括有关所请求操作的信息、操作的日期和时间、请求参数等。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了`CreateCollection`操作。

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"AssumedRole",
      "principalId":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/test-user",
      "accountId":"123456789012",
      "accessKeyId":"access-key",
      "sessionContext":{
         "sessionIssuer":{
            "type":"Role",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:role/Admin",
            "accountId":"123456789012",
            "userName":"Admin"
         },
         "webIdFederationData":{

         },
         "attributes":{
            "creationDate":"2022-04-08T14:11:34Z",
            "mfaAuthenticated":"false"
         }
      }
   },
   "eventTime":"2022-04-08T14:11:49Z",
   "eventSource":"aoss.amazonaws.com",
   "eventName":"CreateCollection",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"AWS Internal",
   "userAgent":"aws-cli/2.1.30 Python/3.8.8 Linux/5.4.176-103.347.amzn2int.x86_64 exe/
x86_64.amzn.2 prompt/off command/aoss.create-collection",
```

```
    "errorCode":"HttpFailureException",
    "errorMessage":"An unknown error occurred",
    "requestParameters":{
       "accountId":"123456789012",
       "name":"test-collection",
       "description":"A sample collection",
       "clientToken":"d3a227d2-a2a7-49a6-8fb2-e5c8303c0718"
    },
    "responseElements": null,
    "requestID":"12345678-1234-1234-1234-987654321098",
    "eventID":"12345678-1234-1234-1234-987654321098",
    "readOnly":false,
    "eventType":"AwsApiCall",
    "managementEvent":true,
    "recipientAccountId":"123456789012",
    "eventCategory":"Management",
    "tlsDetails":{
       "clientProvidedHostHeader":"user.aoss-sample.us-east-1.amazonaws.com"
    }
}
```

使用 Amazon 监控 OpenSearch 无服务器事件 EventBridge

Amazon S OpenSearch ervice 可与 Amazon 集成， EventBridge 以将影响您的域的特定事件通知您。 Amazon 服务 EventBridge 中的事件将近实时传输到。同样的事件也会发送到 Amazon 的前身 Amazon CloudWatch Events 中 EventBridge。您可以编写规则来指示您关注的事件，并指示要在事件匹配规则时执行的自动化操作。可自动激活的操作示例如下：

- 调用函数数数数数数 Amazon Lambda
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams
- 激活 Amazon Step Functions 状态机
- 通知 Amazon SNS 主题或 Amazon SQS 队列

有关更多信息，请参阅《亚马逊 EventBridge 用户指南》 EventBridge中的 “亚马逊入门”。

设置通知

您可以使用Amazon 用户通知在 OpenSearch Serverless 事件发生时接收通知。事件是 OpenSearch 无服务器环境发生变化的指标，例如当您达到 OCU 使用量的最大限制时。 Amazon EventBridge 接收

事件并将通知发送到 Amazon Web Services Management Console 通知中心和您选择的配送渠道。当事件与指定的规则匹配时，会收到通知。

OpenSearch 计算单位 (OCU) 事件

OpenSearch EventBridge 当以下 OCU 相关事件发生时，无服务器会将事件发送到。

OCU 使用量接近最大限制

OpenSearch 当您的搜索或索引 OCU 使用率达到容量限制的 75% 时，无服务器会发送此事件。您的 OCU 使用量是根据您配置的容量限制和当前的 OCU 消耗量计算得出的。

示例

以下是该类型的示例事件（搜索 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."
  }
}
```

以下是该类型的示例事件（索引 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 使用量已达到最大限制

OpenSearch 当您的搜索或索引 OCU 使用率达到容量限制的 100% 时，无服务器会发送此事件。您的 OCU 使用量是根据您配置的容量限制和当前的 OCU 消耗量计算得出的。

示例

以下是该类型的示例事件（搜索 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."
  }
}
```

以下是该类型的示例事件（索引 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."
  }
}
```

创建和管理 Amazon OpenSearch 服务域名

本章介绍如何创建和管理 Amazon OpenSearch 服务域。域等同 Amazon于开源 OpenSearch 集群。在创建域时，您需要指定其设置、实例类型、实例数量以及存储分配。有关开源集群的更多信息，请参阅 OpenSearch 文档中的创建集群。

与入门教程中的简短说明不同，本章描述了所有选项，并提供了相关的参考信息。您可以按照 OpenSearch 服务控制台、() 或 Amazon Command Line Interface (Amazon CLI) 的说明完成每个过程 Amazon SDKs。

创建 OpenSearch 服务域

本节介绍如何使用 OpenSearch 服务控制台或使用 Amazon CLI 带`create-domain`命令的 OpenSearch 服务域来创建服务域。

创建 OpenSearch 服务域（控制台）

使用以下过程通过控制台创建 OpenSearch 服务域。

创建 OpenSearch 服务域（控制台）

1. 前往https://aws.amazon.com并选择 “登录到控制台”。
2. 在 “分析” 下，选择 “亚马逊 OpenSearch 服务”。
3. 选择 Create domain（创建域）。
4. 对于域名，输入一个域名。名称必须符合以下标准：
   - 您的账户独一无二且 Amazon Web Services 区域
   - 以小写字母开头
   - 包含 3 到 28 个字符
   - 只包含小写字母 a-z、数字 0-9 和连字符 (-)
5. 对于域创建方法，选择标准创建。
6. 对于模板，选择与您的域的目的最匹配的选项：
   - 用于需要高可用性和高性能的工作负载的生产域。这些域使用多可用区（带或不带待机功能）和专用主节点提高可用性。
   - 用于开发或测试的开发/测试域。开发/测试域可以使用多可用区（带或不带待机功能）或单个可用区。

> **Important**
>
> 不同部署类型在后续页面上显示不同的选项。这些步骤包括所有选项。

7. 对于部署选项，选择带备用域的域配置三可用区域，其中一个可用区中的节点作为备用域。此选项强制执行大量最佳实践，例如指定数据节点计数、主节点计数、实例类型、副本计数和软件更新设置。
8. 对于版本，请选择要使用的版本 OpenSearch 或旧版 Elasticsearch OSS。我们建议您选择最新版本的 OpenSearch。有关更多信息，请参阅 the section called “Elasticsearch 支持的版本和 OpenSearch”。

   （可选）如果您为域选择了 OpenSearch 版本，请选择启用兼容模式以将其版本 OpenSearch报告为 7.10，这允许某些在连接之前检查版本的 Elasticsearch OSS 客户端和插件继续使用该服务。
9. 对于 Instance type，为数据节点选择实例类型。有关更多信息，请参阅 the section called “支持的实例类型”。

   > **Note**
   >
   > 并非所有可用区都支持所有实例类型。如果您选择带或不带待机功能的多可用区，我们建议您选择最新一代实例类型，如 R5 或 I3。
10. 对于 Number of nodes (节点数)，选择数据节点数。

    有关最大值，请参阅OpenSearch 服务域和实例配额。单节点集群适合开发和测试，但不应用于生产工作负载。有关更多指导，请参阅the section called “调整域大小”和the section called “配置多 AZ 域”。

    > **Note**
    >
    > （可选）专用协调器节点支持所有 OpenSearch 版本和 ElasticSearch 版本 6.8 到 7.10。专用协调器节点可用于启用了专用集群管理器的域。要启用专用协调器节点，您需要选择实例类型和数量。根据最佳实践，专用协调器节点的实例系列应与数据节点相同（基于 Intel 处理器的实例或基于 Graviton 处理器的实例）。
11. 对于存储类型，选择 Amazon EBS。列表中可用的卷类型取决于您选择的实例类型。有关创建超大型域的指南，请参阅 the section called “PB 规模”。

12. 对于 EBS 存储，配置以下附加设置。根据您选择的卷类型，某些设置可能不会显示。

设置	描述
EBS 卷类型	在通用型（SSD）– gp3 与通用型（SSD）– gp2 之间，或上一代预调配 IOPS（SSD）与磁介质（标准）之间选择。
每个节点的 EBS 存储大小	输入要附加到每个数据节点的 EBS 卷。 EBS volume size (EBS 卷大小) 是按节点计算的。您可以通过将数据节点数乘以 EBS 卷大小来计算 OpenSearch 服务域的集群总大小。EBS 卷最小容量和最大容量取决于指定的 EBS 卷类型及其挂载到的实例类型。要了解更多信息，请参阅 EBS 卷大小限制。
预置 IOPS	如果您选择了预配置 IOPS SSD 卷类型，请输入该卷可以 I/O 支持的每秒操作数 (IOPS)。

13. （可选）如果您选择了gp3卷类型，请展开高级设置并指定超出存储价格所含范围的其他 IOPS（每个数据节点每预配置 3 TiB 卷大小最多可获得 16,000 个）和吞吐量（ MiB/s 每个数据节点每预配置 3 TiB 卷大小最多可达 1,000 个），但需支付额外费用。有关更多信息，请参阅 Amazon OpenSearch 服务定价。
14. （可选）要启用UltraWarm 存储，请选择启用 UltraWarm 数据节点。每种实例类型都有其可处理的最大存储量。将该数量乘以总可寻址温存储的温数据节点数。
15. （可选）要启用冷存储，选择启用冷存储。必须启用 UltraWarm 才能启用冷存储。
16. 如果使用带待机功能的多可用区，则已启用三个专用主节点。选择所需的主节点类型。如果选择不带待机功能的多可用区的域，请选择启用专用主节点，然后选择所需的主节点类型和数量。专用主节点可增强集群稳定性，对于包含 10 个以上实例的域而言是必需的。对于生产域，我们建议使用 3 个专用主节点。

> **Note**
>
> 对于专用主节点和数据节点，可以选择不同的实例类型。例如，您可以为数据节点选择通用或存储优化实例，但为专用主节点选择计算优化实例。

17. （可选）对于运行 OpenSearch 或 Elasticsearch 5.3 及更高版本的域名，快照配置无关紧要。有关自动制作快照的更多信息，请参阅the section called “创建索引快照”。

18. 如果您想使用自定义端点，而不是标准的 `https://search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-east-1.es.amazonaws.com` 之一，选择启用定制终端节点并提供名称和证书。有关更多信息，请参阅 the section called “创建自定义终端节点”。
19. 在 Network（网络）下，选择 VPC access（VPC 访问权限）或 Public access（公有访问权限）。如果您选择 Public access (公有访问权限)，请跳至下一步。如果选择 VPC access（VPC 访问权限），请确保您满足先决条件，然后配置以下设置：

设置	描述
VPC	选择要使用的虚拟私有云 (VPC) 的 ID。VPC 和域必须相同 Amazon Web Services 区域，并且必须选择租期设置为 “默认” 的 VPC。 OpenSearch 服务尚不支持使用专用租赁 VPCs 的服务。
子网	选择子网。如果您启用了多可用区，则必须选择两个或三个子网。OpenSearch 服务将在子网中放置 VPC 终端节点和弹性网络接口。 必须在子网中为网络接口预留足够的 IP 地址。有关更多信息，请参阅在 VPC 子网中预留 IP 地址。
安全组	选择一个或多个 VPC 安全组，允许所需的应用程序通过该域公开的端口（80 或 443）和协议（HTTP 或 HTTPS）访问 OpenSearch 服务域。有关更多信息，请参阅 the section called “VPC 支持”。
IAM 角色	保留默认角色。 OpenSearch 服务使用此预定义角色（也称为服务相关角色）来访问您的 VPC，并在 VPC 的子网中放置 VPC 终端节点和网络接口。有关更多信息，请参阅用于 VPC 访问的服务相关角色。
IP 地址类型	选择双堆栈 IPv4 或 IP 地址类型。双堆栈允许您跨 IPv4 IPv6 地址类型共享域资源，这是推荐的选项。如果您将 IP 地址类型设置为双堆栈，则日后无法更改地址类型。

20. 启用或禁用精细访问控制：
    - 如果要使用 IAM 进行用户管理，请选择 Set IAM ARN as master user (将 IAM ARN 设置为主用户)，然后为 IAM 角色指定 ARN。
    - 如果要使用内部用户数据库，请选择创建主用户，并指定用户名和密码。

无论您选择哪个选项，主用户都可以访问集群中的所有索引和所有 OpenSearch APIs索引。有关选择哪个选项的指南，请参阅the section called “重要概念”。

如果您禁用精细访问控制，则仍可以通过将域放置在 VPC 中并/或应用限制性访问策略来控制对域的访问。您必须启用 node-to-node加密和静态加密才能使用精细的访问控制。

> **Note**
>
> 我们强烈建议您启用精细访问控制以保护域中的数据。精细访问控制提供群集、索引、文档和字段级别的安全性。

21. （可选）如果要对 OpenSearch 仪表板使用 SAML 身份验证，请选择启用 SAML 身份验证并为该域配置 SAML 选项。有关说明，请参阅the section called “仪表板的 SAML 身份验证 OpenSearch ”。
22. （可选）如果您想对 OpenSearch 控制面板使用 Amazon Cognito 身份验证，请选择启用 Amazon Cognito 身份验证。然后选择要用于 OpenSearch 控制面板身份验证的 Amazon Cognito 用户池和身份池。有关创建这些资源的指南，请参阅the section called “控制面板的 Amazon Cognito 身份验证 OpenSearch ”。
23. 对于访问策略，选择访问策略或配置您自己的访问策略。如果选择创建自定义策略，则可以自行配置或从另一个域导入策略。有关更多信息，请参阅 the section called “身份和访问管理”。

> **Note**
>
> 如果您启用了 VPC 访问，则无法使用基于 IP 的策略。但您可以使用安全组来控制哪些 IP 地址可以访问该域。有关更多信息，请参阅 the section called “关于 VPC 域的访问策略”。

24. （可选）如果要求对域的所有请求都通过 HTTPS 到达，请选中 Require HTTPS for all traffic to the domain (要求到域的所有流量都使用 HTTPS)。要启用 node-to-node加密，请选择 N ode-to-node 加密。有关更多信息，请参阅 the section called “Node-to-node 加密”。要启用静态数据加密，请选中启用静态数据的加密复选框。如果选择带待机功能的多可用区部署选项，则预先选择这些选项。
25. （可选）选择 “使用 Amazon 自有密钥”，让 S OpenSearch ervice 代表您创建 Amazon KMS 加密密钥（或使用已创建的密钥）。否则，请选择您自己的 KMS 密钥。有关更多信息，请参阅 the section called “静态加密”。

26. 对于非高峰时段，选择开始时间以安排需要部署的服务软件更新和自动调整优化。 blue/green 非高峰时段更新有助于最大限度地减少高流量时段对集群专用主节点造成的压力。
27. 对于 Auto-Tune，请选择是否允许 S OpenSearch ervice 建议对您的域进行与内存相关的配置更改，以提高速度和稳定性。有关更多信息，请参阅 the section called “自动调整”。

    （可选）选择非高峰时段以安排一个循环时段，自动调整将在此时段内更新域。
28. （可选）选择自动软件更新以启用自动软件更新。
29. （可选）添加标签来描述您的域，以便您可以对该信息进行分类和筛选。有关更多信息，请参阅 the section called “标记域”。
30. （可选）展开和配置 Advanced cluster settings（高级集群设置）。有关这些选项的摘要，请参阅the section called “高级集群设置”。
31. 选择创建。

创建 OpenSearch 服务域 (Amazon CLI)

您可以使用，而不是使用控制台创建 OpenSearch 服务域 Amazon CLI。有关语法，请参阅 Amazon CLI 命令参考 a 中的 Amazon OpenSearch 服务。

示例命令

第一个示例演示了以下 OpenSearch 服务域配置：

- 使用 OpenSearch 版本 1.2 创建名为 mylogs 的 OpenSearch 服务域
- 用 `r6g.large.search` 实例类型的两个实例填充该域
- 使用 100GiB 通用型（SSD）gp3 EBS 卷作为每个数据节点的存储
- 允许匿名访问，但只能从单个 IP 地址访问：192.0.2.0/32

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.2 \
    --cluster-config  InstanceType=r6g.large.search,InstanceCount=2 \
    --ebs-options
 EBSEnabled=true,VolumeType=gp3,VolumeSize=100,Iops=3500,Throughput=125 \
    --access-policies '{"Version": "2012-10-17", "Statement": [{"Action": "es:*",
 "Principal":"*","Effect": "Allow", "Condition": {"IpAddress":{"aws:SourceIp":
["192.0.2.0/32"]}}}]}'
```

下一个示例演示了以下 OpenSearch 服务域配置：

- 使用 Elasticsearch 版本 7.10 创建名为 my logs 的 OpenSearch 服务域
- 用 `r6g.large.search` 实例类型的六个实例填充该域
- 使用 100GiB 通用型（SSD）gp2 EBS 卷作为每个数据节点的存储
- 仅限单个用户访问该服务，该用户由用户的 Amazon Web Services 账户 ID 识别：555555555555
- 跨三个可用区分配实例

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version Elasticsearch_7.10 \
    --cluster-config
 InstanceType=r6g.large.search,InstanceCount=6,ZoneAwarenessEnabled=true,ZoneAwarenessConfig={A
 \
    --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
 "Principal": {"AWS": "arn:aws:iam::555555555555:root" }, "Action":"es:*", "Resource":
 "arn:aws:es:us-east-1:555555555555:domain/mylogs/*" } ] }'
```

下一个示例演示了以下 OpenSearch 服务域配置：

- 使用 OpenSearch 版本 1.0 创建名为 mylogs 的 OpenSearch 服务域
- 用 `r6g.xlarge.search` 实例类型的十个实例填充该域
- 用 `r6g.large.search` 实例类型的三个实例作为专用主节点填充该域
- 使用 100GiB 预配置 IOPS EBS 卷作为存储，用每个数据节点 1000 IOPS 的基准性能进行配置
- 限制对单个用户和单个子资源的访问，`_search` API

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.0 \
    --cluster-config
 InstanceType=r6g.xlarge.search,InstanceCount=10,DedicatedMasterEnabled=true,DedicatedMasterTyp
 \
    --ebs-options EBSEnabled=true,VolumeType=io1,VolumeSize=100,Iops=1000 \
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
 "Principal": { "AWS": "arn:aws:iam::555555555555:root" }, "Action": "es:*",
 "Resource": "arn:aws:es:us-east-1:555555555555:domain/mylogs/_search" } ] }'
```

> **Note**
>
> 如果您尝试创建 OpenSearch 服务域，但同名的域已存在，则 CLI 不会报告错误。相反，它会返回现有域的详细信息。

创建 OpenSearch 服务域 (Amazon SDKs)

Amazon SDKs（安卓和 iOS 除外 SDKs）支持《亚马逊 OpenSearch 服务 API 参考》中定义的所有操作，包括`CreateDomain`。有关代码示例，请参阅 the section called “使用 Amazon SDKs”。有关安装和使用的更多信息 Amazon SDKs，请参阅Amazon 软件开发套件。

创建 OpenSearch 服务域 (Amazon CloudFormation)

OpenSearch 服务与一项服务集成 Amazon CloudFormation，该服务可帮助您对 Amazon 资源进行建模和设置，从而减少创建和管理资源和基础架构所花费的时间。您可以创建一个描述您要创建的 OpenSearch 域的模板，并为您 CloudFormation 置备和配置该域。有关更多信息，包括 OpenSearch 域名的 JSON 和 YAML 模板示例，请参阅Amazon CloudFormation 用户指南中的亚马逊 OpenSearch 服务资源类型参考。

配置访问策略

Amazon OpenSearch 服务提供了多种配置 OpenSearch 服务域访问权限的方法。有关更多信息，请参阅the section called “身份和访问管理”和the section called “精细访问控制”。

控制台提供了预配置的访问策略，您可以针对自己域的特定需求自定义这些策略。您也可以从其他 OpenSearch 服务域导入访问策略。有关这些访问策略如何与 VPC 访问交互的信息，请参阅the section called “关于 VPC 域的访问策略”。

配置访问策略 (控制台)

1. 转到 https://aws.amazon.com，然后选择 Sign In to the Console (登录控制台)。
2. 在 “分析” 下，选择 “亚马逊 OpenSearch 服务”。
3. 在导航窗格中的 Domains（域）下，选择要更新的域。
4. 选择 Actions（操作）和 Edit security configuration（编辑安全配置）。
5. 编辑访问策略 JSON，或导入预配置的选项。
6. 选择保存更改。

高级集群设置

使用高级选项来配置以下内容：

请求体中的索引

指定是否允许在 HTTP 请求正文中显式引用索引。将此属性设置为 `false` 可防止用户绕过子资源的访问控制。默认情况下，该值为 `true`。有关更多信息，请参阅 the section called “高级选项和 API 注意事项”。

字段数据缓存分配

指定分配到字段数据的 Java 堆空间的百分比。默认情况下，此设置为 JVM 堆的 20%。

> **Note**
>
> 许多客户查询轮换每日索引。我们建议您在大多数用例中将 `indices.fielddata.cache.size` 配置为 JVM 堆的 40% 来开始基准测试。但是，如果您有非常大的索引，您可能需要大型字段数据缓存。

最大子句数

指定 Lucene 布尔查询中允许的子句的最大数量。默认值为 1024。如果查询具有的子句数超过了允许的子句数，则会导致 `TooManyClauses` 错误。有关更多信息，请参阅 Lucene 文档。

在 Amazon OpenSearch 服务中进行配置更改

更新域名时，Amazon S OpenSearch ervice 使用蓝/绿部署流程。 blue/green 部署为域更新创建了一个空闲环境，该环境会复制生产环境，并在更新完成后将用户路由到新环境。在 blue/green 部署中，蓝色环境是当前的生产环境。绿色环境是闲置环境。

将数据从蓝色环境迁移至绿色环境。当新环境准备就绪时，S OpenSearch ervice 会切换环境，将绿色环境提升为新的生产环境。切换时不会丢失数据。此实践可在部署到新环境未成功的情况下最大程度地减少停机时间并维护原始环境。

主题

- 通常会导致 blue/green 部署的更改
- 通常不会导致 blue/green 部署的更改

- 确定更改是否会导发蓝绿部署
- 跟踪配置更改
- 配置更改的阶段
- 蓝绿部署对性能的影响
- 配置更改的费用
- 对验证错误进行故障排除

通常会导致 blue/green 部署的更改

以下操作会导致 blue/green 部署：

- 更改实例类型
- 启用精细访问控制
- 执行服务软件更新
- 启用或禁用专用主节点
- 启用或禁用不带待机功能的多可用区
- 更改存储类型、卷类型或卷大小
- 选择不同的 VPC 子网
- 添加或删除 VPC 安全组
- 添加或移除专用协调器节点
- 为控制面板启用或禁用 Amazon Cognito 身份验证 OpenSearch
- 选择不同的 Amazon Cognito 用户池或身份池
- 修改高级设置
- 升级到新 OpenSearch 版本（在部分或全部升级过程中，OpenSearch 仪表板可能不可用）
- 启用静态数据加密或 node-to-node加密
- 启用或禁用 UltraWarm 或冷存储
- 禁用自动调整和回滚其更改
- 关联和取消关联可选插件与域
- 增加具有两个专用主节点的多可用区域的专用主节点数量
- 减少 EBS 卷大小
- 更改 EBS 卷大小、IOPS 或吞吐量（前提是您上次进行的更改正在执行或距离上次更改的时间不到 6 小时）

- 允许向发布审核日志 CloudWatch。

对于带待机功能的多可用区的域，一次只能提出一个更改请求。如果已有更改正在执行，则会拒绝新请求。您可以使用 `DescribeDomainChangeProgress` API 查看当前更改状态。

通常不会导致 blue/green 部署的更改

在大多数情况下，以下操作不会导致 blue/green 部署：

- 修改访问策略
- 修改自定义端点
- 更改传输层安全性协议 (TLS) 策略
- 更改自动快照时间
- 启用或禁用 Require HTTPS (要求 HTTPS)
- 在不回滚其更改的情况下启用自动调整或禁用它
- 如果您的域有专用的主节点，请更改数据节点或 UltraWarm 节点数量
- 如果域具有专用主节点，则更改专用主实例类型或节点数量（具有两个专用主节点的多可用区域除外）
- 启用或禁用向发布错误日志或慢速日志 CloudWatch
- 禁止向发布审核日志 CloudWatch
- 将卷大小增加到不超过每个数据节点 3 TiB，更改卷类型、IOPS 或吞吐量
- 添加或删除标签

> **Note**
>
> 存在一些异常情况，具体取决于您的服务软件版本。如果您想确保更改不会导致 blue/green 部署，请在更新域名之前进行试运行（如果此选项可用）。某些更改不提供试运行选项。一般而言，我们建议您在流量高峰时段之外对集群进行更改。

确定更改是否会导发蓝绿部署

您可以测试某些类型的计划配置更改，以确定它们是否会导致 blue/green 部署，而不必承诺这些更改。发起配置更改之前，请使用控制台或 API 运行验证检查，以确保域符合更新条件。

Console

验证配置更改

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择 Domains（域）。
3. 选择要进行配置更改的域。随即打开域详细信息页面。选择 Actions（操作）下拉菜单，然后选择 Edit cluster configuration（编辑集群配置）。
4. 对域进行更改，例如更改实例类型或节点数量。
5. 在试运行分析下，选择运行。试运行会验证您的配置更改是否存在错误，并确定是否需要 blue/green 部署。
6. 试运行完成后，结果将与试运行 ID 一起显示在页面底部。分析表明配置更改是否需要 blue/green 部署。

   每次试运行都会覆盖前一次试运行。要保留每次运行的详细信息，请保存其试运行 ID。试运行的有效期为 90 天，或者直到您更新配置为止。
7. 要继续更新配置，请选择 Save changes（保存更改）。否则，选择取消。任一选项都会带您返回到 Cluster configuration（集群配置）选项卡。在此选项卡上，您可以选择 Dry run details（试运行详细信息）以查看最新试运行的详细信息。本页还包括试运行前的配置和试运行配置之间的 side-by-side比较。

API

您可以通过配置 API 来执行试运行验证。要使用 API 测试更改，请将 `DryRun` 设置为 `true`，将 `DryRunMode` 设置为 `Verbose`。除了确定更改是否会启动蓝绿部署外，Verbose 模式还会运行验证检查。例如，此UpdateDomainConfig请求测试启用后生成的部署类型 UltraWarm：

```
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"
```

```
}
```

该请求将运行验证检查并返回更改将会导发的部署类型，但实际上不会执行更新：

```
{
   "ClusterConfig": {
     ...
    },
   "DryRunResults": {
      "DeploymentType": "Blue/Green",
      "Message": "This change will require a blue/green deployment."
    }
}
```

可能的部署类型如下：

- Blue/Green：此更改将会导致蓝绿部署。
- DynamicUpdate：此更改不会导致蓝绿部署。
- Undetermined：域仍处于正在处理状态，因此无法确定部署类型。
- None：未发生配置更改。

如果验证失败，将返回验证失败列表。

```
{
   "ClusterConfig":{
      "..."
   },
   "DryRunProgressStatus":{
      "CreationDate":"2023-01-12T01:14:33.847Z",
      "DryRunId":"db00ca39-48b2-4774-bbd3-252cf094d205",
      "DryRunStatus":"failed",
      "UpdateDate":"2023-01-12T01:14:33.847Z",
      "ValidationFailures":[
         {
            "Code":"Cluster.Index.WriteBlock",
            "Message":"Cluster has index write blocks."
         }
      ]
   }
}
```

如果状态仍为pending，则可以在后续DescribeDryRunProgress调用的 UpdateDomainConfig 响应中使用试运行 ID 来检查验证状态。

```
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."
    }
}
```

要在不进行验证检查的情况下运行试运行分析，请在使用配置 API 时将 DryRunMode 设置为 Basic。

Python

以下 Python 代码使用 UpdateDomainConfigAPI 执行试运行验证检查，如果检查成功，则在不进行试运行的情况下调用相同的 API 来开始更新。如果检查失败，脚本将输出错误并停止。

```
import time
import boto3

client = boto3.client('opensearch')

response = client.UpdateDomainConfig(
    ClusterConfig={
        'WarmCount': 3,
        'WarmEnabled': True,
        'WarmCount': 123,
    },
    DomainName='test-domain',
    DryRun=True,
    DryRunMode='Verbose'
)
```

```
dry_run_id = response.DryRunProgressStatus.DryRunId

retry_count = 0

while True:

    if retry_count == 5:
        print('An error occured')
        break

    dry_run_progress_response = client.DescribeDryRunProgress('test-domain',
 dry_run_id)
    dry_run_status = dry_run_progress_response.DryRunProgressStatus.DryRunStatus

    if dry_run_status == 'succeeded':
        client.UpdateDomainConfig(
            ClusterConfig={
            'WarmCount': 3,
            'WarmEnabled': True,
            'WarmCount': 123,
        })
        break

    elif dry_run_status == 'failed':
        validation_failures_list =
 dry_run_progress_response.DryRunProgressStatus.ValidationFailures
        for item in validation_failures_list:
            print(f"Code: {item['Code']}, Message: {item['Message']}")
        break

    retry_count += 1
    time.sleep(30)
```

跟踪配置更改

您可以一次请求一项配置更改，也可以将多项更改分组到一个请求中。使用控制台中的域处理状态和配置更改状态字段来跟踪配置更改。请等待域名状态变成，`Active`然后再请求其他更改。

域名可以具有以下处理状态：

- `Active`— 未进行任何配置更改。您可以提交新的配置更改请求。

- `Creating`— 正在创建域。
- `Modifying`— 正在进行配置更改，例如添加新的数据节点、EBS、gp3、IOPS 配置或设置 KMS 密钥。
- `Upgrading engine version`— 正在进行引擎版本升级。
- `Updating service software`— 服务软件更新正在进行中。
- `Deleting`— 域名正在被删除。
- `Isolated`— 域名已暂停。

一个域可以具有以下配置更改状态：

- `Pending`— 已提交配置更改请求。
- `Initializing`— 该服务正在初始化配置更改。
- `Validating`— 该服务正在验证请求的更改和所需的资源。
- `Awaiting user inputs`— 该服务需要继续进行配置更改，例如实例类型更改。您可以编辑配置更改。
- `Applying changes`— 服务正在应用所请求的配置更改。
- `Cancelled`— 配置更改已取消。选择取消回滚所有更改。
- `Completed`— 请求的配置更改已成功完成。
- `Validation failed`— 请求的配置更改未能完成。未应用任何配置更改。

> **Note**
>
> 验证失败可能是由于您的域中存在红色索引、所选实例类型不可用或磁盘空间不足所致。有关验证错误的列表，请参阅the section called “对验证错误进行故障排除”。出现验证失败事件时，您可以取消、重试或编辑配置更改。

配置更改完成后，域状态将变回为`Active`。

您可以查看集群运行状况和 Amazon CloudWatch 指标，发现域更新发生时，集群中的节点数量会暂时增加（通常是翻一番）。在下图中，您可以看到配置更改期间的节点从 11 个翻倍至 22 个，然后在更新完成后返回至 11 个。

这种临时的增加可能会对集群的专用主节点造成压力，其要管理的节点数可能突然增加。它还会增加搜索和索引延迟，因为 S OpenSearch ervice 会将数据从旧集群复制到新集群。在集群上保持足够的容量以处理与这些 blue/green 部署相关的开销非常重要。

> **Important**
>
> 在配置更改和服务维护期间，您不会 产生任何额外费用。您只需要为您的集群请求的节点数量付费。有关具体信息，请参阅the section called “配置更改的费用”。

为防止专用主节点过载，您可以使用 Amazon CloudWatch 指标监控使用情况。有关推荐的最大值，请参阅the section called “推荐的 CloudWatch 警报”。

配置更改的阶段

在您启动配置更改后，S OpenSearch ervice 会通过一系列步骤来更新您的域名。您可以在控制台中的配置更改状态下查看配置更改的进度。更新经过的确切步骤取决于您正在进行的更改的类型。此外，您还可以使用 DescribeDomainChangeProgress API 操作监控配置更改。

以下是在配置更改期间更新可能经历的阶段：

阶段名称	描述
验证	验证域是否符合更新条件，如有必要，显示验证问题。
创建新环境	完成必要的先决条件并创建开始 blue/gree

阶段名称	描述
	n 部署所需的资源。
预置新节点	在新环境中创建新的实例组。
新节点上的流量路由	将流量重定向到新创建的数据节点。
旧节点上的流量路由	禁用旧数据节点上的流量。
准备要删除的节点	准备移除节点。此步骤仅在您将域缩小（例如，从 8 个节点缩小到 6 个节点）时才会发生。
将分片复制到新节点	将分片从旧节点移动到新节点。
终止节点	删除分片后，终止并删除旧节点。
删除较旧资源	删除与旧环境关联的资源（例如负载均衡器）。

阶段名称	描述
动态更新	当更新不需要 blue/green 部署并且可以动态应用时显示。
应用与专用主节点相关的更改	当专用主实例类型或计数更改时显示。
应用与卷相关的更改	卷大小、类型、IOPS 和吞吐量发生变化时显示。

蓝绿部署对性能的影响

在 blue/green 部署期间，您的 Amazon S OpenSearch ervice 集群可用于传入的搜索和索引请求。但是，您可能会遇到以下性能问题：

- 由于集群需要管理更多的节点，领导节点的使用量暂时增加。
- 由于 OpenSearch 服务将数据从旧节点复制到新节点，因此增加了搜索和索引延迟。
- 随着 blue/green 部署期间集群负载的增加，对传入请求的拒绝率增加。
- 为避免延迟问题和请求被拒绝，您应该在集群运行状况良好且网络流量较低时运行 blue/green 部署。

配置更改的费用

如果您更改域的配置，S OpenSearch ervice 会按中所述创建一个新集群the section called “配置更改”。在从旧群集迁移到新群集时，会产生以下费用：

- 如果您更改实例类型，第一个小时两个集群都会收费。第一个小时后，您只需为新群集付费。EBS 卷不会收取两次费用，因为它们是您的集群的一部分，因此它们的计费遵循实例计费。

示例：您将配置从三个 m3.xlarge 实例更改为四个 m4.large 实例。在第一个小时中，两个集群都需要收费 (3 个 m3.xlarge + 4 个 m4.large)。第一个小时后，您只需为新集群付费 (4 个 m4.large)。

- 如果您未更改实例类型，第一个小时您只需要为最大的集群付费。第一个小时后，您只需为新集群付费。

  示例： 您将配置从六个 m3.xlarge 实例更改为三个 m3.xlarge 实例。在第一个小时，您只需要为最大的集群付费 (6 个 m3.xlarge)。第一个小时后，您只需为新集群付费 (3 个 m3.xlarge)。

对验证错误进行故障排除

当您启动配置更改或执行 OpenSearch 或 Elasticsearch 版本升级时，Serv OpenSearch ice 会首先执行一系列验证检查，以确保您的域符合更新条件。如果其中任何一项检查失败，您将在控制台中收到通知，其中包含在更新域之前必须修复的特定问题。下表列出了 OpenSearch 服务可能出现的域名问题以及解决这些问题的步骤。

事务	错误代码	故障排除步骤
未找到安全组	SecurityGroupNotFound	与您的 OpenSearch 服务域关联的安全组不存在。要解决此问题，请使用指定的名称创建安全组。
未找到子网	SubnetNotFound	与您的 OpenSearch 服务域关联的子网不存在。要解决此问题，请在您的 VPC 中创建子网。
未配置服务相关角色	SLRNotConfigured	未配置服务的 OpenSearch 服务相关角色。服务相关角色由 S OpenSearch ervice 预定义，包括该服务代表您调用其他 Amazon 服务所需的所有权限。如果该角色不存在，您可能需要手动创建角色。
没有足够的 IP 地址	InsufficientFreeIPsForSubnets	您的一个或多个 VPC 子网没有足够的 IP 地址，因此无法更新您的域。要计算您需要的 IP 地址数量，请参阅the section called “在 VPC 子网中预留 IP 地址”。
Cognito 用户群体不存在	CognitoUserPoolNotFound	OpenSearch 服务找不到 Amazon Cognito 用户池。确认您已创建一个用户池并具有正确的 ID。要查找该 ID，您可以使用 Amazon Cognito 控制台或以下 Amazon CLI 命令：

事务	错误代码	故障排除步骤
		`aws cognito-idp list-user-pools --max-results 60 --region us-east-1`
Cognito 身份群体不存在	`CognitoIdentityPoolNotFound`	OpenSearch 服务找不到 Cognito 身份池。确认您已创建一个用户池并具有正确的 ID。要查找该 ID，您可以使用 Amazon Cognito 控制台或以下 Amazon CLI 命令： `aws cognito-identity list-identity-pools --max-results 60 --region us-east-1`
找不到用户群体的 Cognito 域	`CognitoDomainNotFound`	用户池没有域名。您可以使用 Amazon Cognito 控制台或以下 Amazon CLI 命令进行配置： `aws cognito-idp create-user-pool-domain --domain my-domain --user-pool-id id`
未配置 Cognito 角色	`CognitoRoleNotConfigured`	未配置向 OpenSearch 服务授予配置 Amazon Cognito 用户和身份池以及使用它们进行身份验证的权限的 IAM 角色。使用适当的权限集和信任关系配置角色。您可以使用控制台为您创建默认CognitoAccessForAmazonOpenSearch角色，也可以使用或 Amazon SDK 手动配置角色。 Amazon CLI
无法描述用户群体	`UserPoolNotDescribable`	指定的 Amazon Cognito 角色无权描述与您的域关联的用户群体。确保角色权限策略允许 `cognito-identity:DescribeUserPool` 操作。请参阅 the section called “关于 CognitoAccessForAmazonOpenSearch 角色”获取完全权限策略。
无法描述身份群体	`IdentityPoolNotDescribable`	指定的 Amazon Cognito 角色无权描述与您的域关联的身份群体。确保角色权限策略允许 `cognito-identity:DescribeIdentityPool` 操作。请参阅 the section called “关于 CognitoAccessForAmazonOpenSearch 角色”获取完全权限策略。

事务	错误代码	故障排除步骤
无法描述用户群体和身份群体	`CognitoPoolsNotDescribable`	指定的 Amazon Cognito 角色无权描述与您的域关联的用户群体和身份群体。确保角色权限策略允许 `cognito-identity:DescribeIdentityPool` 和 `cognito-identity:DescribeUserPool` 操作。请参阅 the section called “关于 CognitoAccessForAmazonOpenSearch 角色”获取完全权限策略。
KMS 密钥未启用。	`KMSKeyNotEnabled`	用于加密您的域名的 Amazon Key Management Service (Amazon KMS) 密钥已禁用。立即重新启用密钥。
自定义证书未处于 ISSUED（已签发）状态	`InvalidCertificate`	如果您的域使用自定义终端节点，则可以通过在 Amazon Certificate Manager (ACM) 中生成 SSL 证书或导入自己的证书来保护该终端节点。证书状态必须为 Issued（已签发）。如果您收到此错误，请在 ACM 控制台中检查证书状态。如果状态为“Expired（已过期）”、“Failed（失败）”、“Inactive（非活动）”或“Pending validation（待验证）”，请参阅 ACM 故障排除文档解决问题。
容量不足，无法启动所选实例类型	`InsufficientInstanceCapacity`	请求的实例类型容量不可用。例如，您可能请求了五个`i3.16xlarge.search` 节点，但是 S OpenSearch ervice 没有足够`i3.16xlarge.search` 的主机可用，因此无法完成请求。在 S OpenSearch ervice 中查看支持的实例类型，然后选择其他实例类型。
集群中的红色索引	`RedCluster`	集群中一个或多个索引的状态为红色，导致集群的整体状态为红色。要对此问题进行故障排除和修复，请参阅the section called “红色集群状态”。
内存断路器，请求过多	`TooManyRequests`	您的域名有太多的搜索和写入请求，因此 S OpenSearch ervice 无法更新其配置。您可以减少请求数量，将实例的 RAM 纵向扩展至高达 64 GiB，或者通过添加实例横向扩展。
新配置无法保存数据（磁盘空间不足）	`InsufficientStorageCapacity`	配置的存储大小无法保存域中的所有数据。要解决此问题，请选择更大的卷，删除未使用的索引，或者增加集群中的节点数量以立即释放磁盘空间。

事务	错误代码	故障排除步骤

固定到特定节点的分片	ShardMovementBlocked	您的域中的一个或多个索引已附加到特定节点，并且无法重新分配。发生这种情况的最可能原因是您配置了分片分配筛选，其使您能够指定允许哪些节点托管特定索引的分片。
		
		要解决此问题，请从所有受影响的索引中删除分片分配筛选条件：
		
		`PUT my-index/_settings`
		`{`
		`  "settings": {`
		`  "index.routing.allocation.require._name": null`
		`  }`
		`}`

新配置无法保存所有分片（分片数）	TooManyShards	您的域上的分片数过高，这会阻止 S OpenSearch ervice 将其移至新配置。要解决此问题，请通过添加与当前集群节点具有相同配置类型的节点来横向扩展域。请注意，最大 EBS 卷大小取决于节点的实例类型。 要防止将来出现此问题，请参阅 the section called “选择分片数量”并定义适合您的使用案例的分片策略。
与您的域关联的子网不支持 IPv4 地址	ResultCodeIPv4BlockNotExists	要解决此问题，请根据配置的域 IP 地址类型在 VPC 中创建子网或更新现有子网。如果您的域IPv4仅使用地址类型，请使用 IPv4仅限子网。如果您的域使用双堆栈模式，请使用双堆栈子网。
与您的域关联的子网不支持 IPv6 地址	ResultCodeIPv6BlockNotExists	要解决此问题，请根据配置的域 IP 地址类型在 VPC 中创建子网或更新现有子网。如果您的域IPv4仅使用地址类型，请使用 IPv4仅限子网。如果您的域使用双堆栈模式，请使用双堆栈子网。

Amazon Serv OpenSearch ice 中的服务软件更新

> **Note**
>
> 有关每个主要（非补丁）服务软件更新中所做的更改和新增功能说明，请参阅发布说明。

Amazon OpenSearch Service 定期发布增加功能或以其他方式改进您的域的服务软件更新。控制台中的 Notifications（通知）面板是查看是否有可用的更新或检查更新状态的最简单方法。每个通知都包含有关服务软件更新的详细信息。所有服务软件更新均使用蓝绿部署，以尽可能减少停机时间。

服务软件更新不同于 OpenSearch 版本升级。有关升级到更高版本的的信息 OpenSearch，请参阅the section called “升级域”。

可选更新与必需更新

OpenSearch 服务支持两大类服务软件更新：

可选更新

可选服务软件更新通常包括增强功能以及新特性或新功能支持。域不会强制执行可选更新，可选更新安装不存在硬性截止日期要求。通过电子邮件和控制台通知传达更新发布日期。您可以选择立即应用更新，也可以将其重新安排到更合适的日期和时间。您也可以在域的非高峰时段安排更新。绝大多数软件更新为可选更新。

无论是否计划更新，如果您对域做出更改并触发蓝绿部署，则 S OpenSearch ervice 自动为您更新服务软件。

您可以配置域，在非高峰时段自动应用可选更新。启用此选项后， OpenSearch 服务在可选更新可用后至少等待 13 天，然后计划在 72 小时（三天）后进行更新。计划更新时，将收到控制台通知，您可以选择重新安排在未来的某个日期执行更新。

要启用自动软件更新，请在创建或更新域时选择启用自动软件更新。要使用配置相同的设置 Amazon CLI，请在创建或更新域`true`时设置为`--software-update-options`。

必需更新：

必需服务软件更新通常包括重要安全修复或其他强制更新，以确保域的持续完整性和功能性。例如，必需更新包括 Log4j 常见漏洞和风险 (CVEs) 以及强制执行实例元数据服务版本 2 ()。IMDSv2一年的强制更新次数通常少于三次。

OpenSearch Service 会自动安排以上更新，并在计划更新前 72 小时（三天）通过电子邮件和控制台通知向您发送通知。您可以选择立即应用更新，也可以将其重新安排到更合适的日期和时间以及允许时间范围内的某个时间。您也可以在域的下一非高峰时段安排更新。如果未对必需更新采取任何操作，也没有执行任何域更改触发蓝绿部署，在超出指定的截止日期（通常是发布后 14 天）后，S OpenSearch ervice 有权在域的非高峰时段随时启动更新。

无论计划何时更新，如果您对域做出更改并触发蓝绿部署，则 S OpenSearch ervice 自动为您更新域。

补丁更新

以 “-P” 和数字结尾的服务软件版本（例如 R20211203-*P4*）是补丁版本。补丁可能包括性能改进、次要错误修复以及安全修复或状态改进。补丁发行版不包含新功能或重大更改，并且通常不会对用户产生直接或明显的影响。服务软件通知将说明补丁版本是可选补丁还是必需补丁。

注意事项

决定是否更新域时，请考虑以下事项：

- 通过手动更新域，您可以更快地利用新功能。在选择更新时，S OpenSearch ervice 将请求放置在队列中，并在有时间时开始更新。
- 在启动服务软件更新时，S OpenSearch ervice 会在更新开始和完成时发送通知。
- 软件更新使用蓝绿部署，以尽可能减少停机时间。更新可能会临时使集群的专用主节点紧张，因此请确保保持足够的容量来处理相关开销。
- 通常在几分钟内完成更新，但如果您的系统负载过重，也可能需要几个小时甚至几天。考虑在配置的非高峰时段更新您的域，以避免长时间更新。

启动服务软件更新

您可以通过服务控制台 Amazon CLI、或其中一个请求 OpenSearch 服务软件更新 SDKs。

控制台

请求服务软件更新

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择域名以打开其配置。

3. 选择操作、更新，然后选择以下选项之一：

   - 立即应用更新 - 如果有可用容量，则立即安排在当前小时执行更新。如果容量不足，我们会提供其他可用时段供您选择。
   - 计划在非高峰时段执行更新 - 仅当为域启用非高峰时段时段时可用。计划在域配置的非高峰时段执行更新。无法保证下一个时段执行更新。视容量而定，可能会在接下来的几天内更新。有关更多信息，请参阅 the section called “非高峰窗口”。
   - 计划特定日期和时间更新 - 计划在特定日期和时间执行更新。如果出于容量原因指定的时间不可用，则可以选择其他时段。

   如果计划在未来的某个日期（域的非高峰期之内或之外）执行更新，则可以随时重新安排更新。有关说明，请参阅the section called “重新计划操作”。

4. 选择确认。

Amazon CLI

发送启动服务软件更新的start-service-software-update Amazon CLI 请求。此示例将更新立即添加到队列中：

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "NOW"
```

响应：

```
{
    "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**
>
> 在请求更新之后，您可以在很小的时间范围内将其取消。此`PENDING_UPDATE`状态的持续时间可能会有很大的不同，并取决于 Amazon Web Services 区域 和 S OpenSearch ervice 正在执行的并发更新的数量。要取消更新，请使用控制台或`cancel-service-software-update` Amazon CLI 命令。

如果请求失败并出现 `BaseException`，则表示出于容量原因指定的时间不可用，您必须指定其他时间。 OpenSearch 服务将在响应中提供其他可用时段建议。

Amazon SDKs

此示例 Python 脚本使用中的 d escribe_domain 和 start_service_software_update 方法 适用于 Python (Boto3) 的 Amazon SDK 来检查域是否符合服务软件更新的条件，如果是，则启动更新。您必须为 `domain_name` 提供一个值。

```
import boto3
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)

domain_name = ''  # The name of the domain to check and update

client = boto3.client('opensearch', config=my_config)


def getUpdateStatus(client):
    """Determines whether the domain is eligible for an update"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    sso = response['DomainStatus']['ServiceSoftwareOptions']
    if sso['UpdateStatus'] == 'ELIGIBLE':
```

```
        print('Domain [' + domain_name + '] is eligible for a service software update
 from version ' +
              sso['CurrentVersion'] + ' to version ' + sso['NewVersion'])
        updateDomain(client)
    else:
        print('Domain is not eligible for an update at this time.')


def updateDomain(client):
    """Starts a service software update for the eligible domain"""
    response = client.start_service_software_update(
        DomainName=domain_name
    )
    print('Updating domain [' + domain_name + '] to version ' +
          response['ServiceSoftwareOptions']['NewVersion'] + '...')
    waitForUpdate(client)


def waitForUpdate(client):
    """Waits for the domain to finish updating"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    status = response['DomainStatus']['ServiceSoftwareOptions']['UpdateStatus']
    if status == 'PENDING_UPDATE' or status == 'IN_PROGRESS':
        time.sleep(30)
        waitForUpdate(client)
    elif status == 'COMPLETED':
        print('Domain [' + domain_name +
              '] successfully updated to the latest software version')
    else:
        print('Domain is not currently being updated.')

def main():
    getUpdateStatus(client)
```

计划在非高峰时段执行软件更新

2023 年 2 月 16 日之后创建的每个 OpenSearch 服务域，当地时间每天晚上 10:00 到上午 8:00 之间的 10 小时视为非高峰时段。 OpenSearch 服务使用在此时段安排域服务软件更新。非高峰时段更新有助于最大限度地减少高流量时段对集群专用主节点造成的压力。 OpenSearch 未经您的同意，服务无法在上述 10 小时之外的时间启动更新。

- 对于可选更新，S OpenSearch ervice 将通知更新发布日期，提示您在即将到来的非高峰时段安排更新。
- 对于必需更新，S OpenSearch ervice 将在即将到来的非高峰时段自动安排更新并提前三天通知您。您可以重新安排更新（在非高峰时段之内或之外执行），但只能在要求的时间范围内完成更新。

对于每个域，您可以选择使用自定义时间覆盖默认开始时间晚上 10:00。有关说明，请参阅the section called “配置自定义非高峰窗口”。

控制台

计划在即将到来的非高峰时段执行更新

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择域名以打开其配置。
3. 选择操作、更新。
4. 选择计划在非高峰时段执行更新。
5. 选择确认。

您可以在非高峰时段选项卡上查看计划操作并随时重新安排时间。请参阅the section called “查看计划的操作”。

CLI

要使用计划在即将到来的非高峰时段执行更新 Amazon CLI，请发送StartServiceSoftwareUpdate请求并指定`OFF_PEAK_WINDOW``--schedule-at`参数：

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "OFF_PEAK_WINDOW"
```

监控服务软件更新

OpenSearch 当服务软件更新可用、必须执行、已启动、已完成或失败时，服务就会发送通知。您可以在 OpenSearch 服务控制台的通知面板中查看这些通知。如果更新是可选的，通知严重性为`Informational`，如果需要，严重性为`High`。

OpenSearch 服务还会向 Amazon 发送服务软件事件 EventBridge。您可以使用配置 EventBridge 在收到事件时发送电子邮件或执行特定操作的规则。有关示例演练的信息，请参阅the section called “教程：为可用更新发送 SNS 警报”。

要了解发送到 Amazon 的每个服务软件事件的格式 EventBridge，请参阅the section called “服务软件更新事件”。

当域不符合更新资格时

如果您的域处于下表中所示的任何状态，则可能没有资格进行服务软件更新。

状态	描述
域正在处理中	域正在接受配置更改。在操作完成后检查更新资格。
红色集群状态	集群中的一个或多个索引为红色。有关问题排查步骤，请参阅the section called “红色集群状态”。
高错误率	在尝试处理请求时， OpenSearch 集群返回大量 5 xx 错误。此问题通常是因为同时读取或写入了过多的请求。请考虑减少流向集群的流量或扩展您的域。
裂脑	脑裂意味着您的 OpenSearch 集群有多个主节点并且已拆分成两个绝不会自行重新联接的集群。您可以通过使用推荐数量的专用主节点避免裂脑。为了帮助您从裂脑恢复，请联系 Amazon Web Services 支持。
Amazon Cognito 集成问题	您的域使用 OpenSearch 仪表板身份验证， OpenSearch 服务找不到一个或多个 Amazon Cognito 资源。如果缺少 Amazon Cognito 用户池，则通常会出现此问题。要更正此问题，请重新创建缺少的资源并将 S OpenSearch ervice 域配置为使用它。
其他 服务问题	OpenSearch 服务本身的问题可能会导致您的域显示没有更新资格。如果上述情况都不适用于您的域且该问题持续超过一天，请联系 Amazon Web Services 支持。

定义 Amazon OpenSearch Service 非高峰窗口

您在创建 Amazon S OpenSearch ervice 域时，可以定义每天 10 小时的非高峰时段。 OpenSearch 服务在此窗口内计划服务软件更新，以及自动调整尽可能在流量相对较低的窗口内进行蓝绿部署的优化。蓝绿部署是指创建用于域更新的新环境，并在这些更新完成后将用户路由至新环境的流程。

尽管蓝绿部署不会造成中断，但为了最大限度地减少蓝绿部署消耗资源时对性能的任何潜在影响，建议您在域配置的非高峰窗口计划这些部署。不在非高峰窗口进行诸如节点替换之类的更新或者需要立即部署到域中的更新。

您可以修改非高峰窗口的开始时间，但不能修改窗口时长。

> **Note**
>
> 非高峰窗口于 2023 年 2 月 16 日推出。默认情况下，在此日期之前创建的所有域都禁用非高峰窗口。您必须手动启用和配置这些域的非高峰窗口。默认情况下，在此日期之后创建的所有域都启用非高峰窗口。启用域的非高峰窗口后，您将无法再将其禁用。

非高峰窗口服务软件更新

OpenSearch Service 支持两大类服务软件更新：可选更新和必需更新。两种类型都需要蓝绿部署。不会对您的域强制执行可选更新，但如果您在指定的截止日期（通常是更新可用两周）之前不采取任何操作，则会自动安装必需更新。有关更多信息，请参阅 the section called “可选更新与必需更新”。

启动可选更新后，您可以选择立即应用更新，计划在随后的非高峰窗口进行，或者指定自定义的日期和时间来应用更新。

对于必需更新， OpenSearch 服务会自动计划在非高峰时段执行更新的日期和时间。您会在计划更新前的三天收到通知，而且您也可以在必需的部署期限内将其重新安排到稍后的日期和时间。有关说明，请参阅the section called “重新计划操作”。

非高峰期自动调整优化

自动调整之前使用维护窗口来计划必需进行蓝绿部署的更改。在引入非高峰窗口之前已经启用自动调整和维护窗口的域将继续使用维护窗口进行这些更新，除非您将其迁移到使用非高峰窗口。

建议您将域迁移到使用非高峰窗口，因为该窗口用于计划域上诸如服务软件更新之类的其他活动。有关说明，请参阅the section called “从自动调整维护窗口迁移”。将域迁移到非高峰窗口后，您将无法恢复到使用维护窗口。

2023 年 2 月 16 日之后创建的所有域都将使用非高峰窗口（而不是传统的维护窗口）来计划blue/green deployments. You can't disable the off-peak window for a domain. For a list of Auto-Tune optimizations that require blue/green部署，请参阅the section called “更改类型”。

启用非高峰窗口

默认情况下，在 2023 年 2 月 16 日（引入非高峰窗口）之前创建的所有域都禁用该功能。您必须手动为这些域启用该功能。启用非高峰窗口后，您将无法再将其禁用。

控制台

要启用域的非高峰窗口

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择域的名称以打开其配置。
3. 导航至非高峰窗口选项卡，然后选择编辑。
4. 您可以使用协调世界时 (UTC) 格式自定义开始时间。例如，要配置美国西部（俄勒冈州）区域的开始时间为晚上 11:30，请指定 07:30。
5. 选择保存更改。

CLI

要使用修改非高峰窗口 Amazon CLI，UpdateDomainConfig请发送请求：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
```

```
  --off-peak-window-options 'Enabled=true,
 OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

如果您未指定自定义窗口的开始时间，则默认为世界标准时间 00:00。

配置自定义非高峰窗口

您可以采用协调世界时 (UTC) 指定域的自定义非高峰窗口。例如，如果您要为域配置美国东部（弗吉尼亚州北部）区域的非高峰窗口开始时间为晚上 11:00，请指定 04:00 UTC。

控制台

要修改域的非高峰窗口

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择域的名称以打开其配置。
3. 导航至非高峰窗口选项卡。您可以查看已配置的非高峰窗口以及即将为域执行的计划操作列表。
4. 选择编辑，然后采用 UTC 指定新的开始时间。例如，要配置美国东部（弗吉尼亚州北部）区域的开始时间晚上 9:00，请指定 02:00 UCT。
5. 选择保存更改。

CLI

要使用配置自定义非高峰窗口 Amazon CLI，请发送UpdateDomainConfig请求，并以 24 小时时间格式指定小时和分钟。

例如，以下请求将窗口开始时间更改为 UTC 凌晨 2:00：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

如果您未指定窗口开始时间，则创建域的 Amazon Web Services 区域 默认时间为当地时间晚上 10:00。

查看计划的操作

您可以查看每个域的当前计划、正在进行或待处理的所有操作。操作的严重程度可分为`HIGH`、`MEDIUM`和`LOW`。

操作具有以下状态：

- `Pending update`— 操作正在待处理的队列中。
- `In progress`— 操作当前正在进行中。
- `Failed`— 操作未能完成。
- `Completed`— 操作已成功完成。
- `Not eligible`— 仅适用于服务软件更新。更新无法继续，因为集群运行状况不佳。
- `Eligible`— 仅适用于服务软件更新。该域符合更新条件。

控制台

OpenSearch 服务控制台显示域配置中的所有计划操作，以及每个操作的严重程度和当前状态。

要查看域的计划操作

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择域的名称以打开其配置。
3. 导航至非高峰窗口选项卡。
4. 在计划操作下，查看域的当前计划、正在进行或待处理的所有操作。

CLI

要使用查看计划操作 Amazon CLI，ListScheduledActions请发送请求：

```
aws opensearch list-scheduled-actions \
  --domain-name my-domain
```

响应：

```
{
    "ScheduledActions": [
        {
            "Cancellable": true,
            "Description": "The Deployment type is : BLUE_GREEN.",
            "ID": "R20220721-P13",
            "Mandatory": false,
```

```
            "Severity": "HIGH",
            "ScheduledBy": "CUSTOMER",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
            "Type": "SERVICE_SOFTWARE_UPDATE",
        },
        {
            "Cancellable": true,
            "Description": "Amazon Opensearch will adjust the young generation JVM
 arguments on your domain to improve performance",
            "ID": "Auto-Tune",
            "Mandatory": true,
            "Severity": "MEDIUM",
            "ScheduledBy": "SYSTEM",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
            "Type": "JVM_HEAP_SIZE_TUNING",
        }
    ]
}
```

重新计划操作

OpenSearch 服务会通知您计划服务软件更新和自动调整优化。您可以选择立即应用更新，也可以将其重新安排到稍后的日期和时间。

> **Note**
>
> OpenSearch 服务可以在您选择的时间后一小时内安排操作。例如，如果您选择在下午 5 点应用更新，则更新会在下午 5 点到 6 点之间进行。

控制台

要重新安排操作

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择域的名称以打开其配置。
3. 导航至非高峰窗口选项卡。
4. 在计划操作下，依次选择操作和重新安排。

5. 请选择以下选项之一：
   - 立即应用更新 - 如果有可用容量，则立即安排在当前小时执行更新。如果容量不足，我们会提供其他可用时段供您选择。
   - 将其安排在非高峰窗口 - 标记要在即将到来的非高峰窗口中进行操作。无法保证在下一个窗口会执行更改。视容量而定，可能会在接下来的几天内更新。
   - 重新安排此更新 - 允许您指定应用更改的自定义日期和时间。如果出于容量原因指定的时间不可用，则可以选择其他时段。
   - 取消计划更新 - 取消更新。此选项只适用于可选服务软件更新。它不适用于自动调整操作或必需的软件更新。
6. 选择保存更改。

CLI

要使用重新安排操作 Amazon CLI，请发送请求。UpdateScheduledAction要检索操作 ID，请发送 `ListScheduledActions` 请求。

以下请求将服务软件更新重新安排在某个具体的日期和时间：

```
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
```

响应：

```
{
   "ScheduledAction": {
      "Cancellable": true,
      "Description": "Cluster status is updated.",
      "Id": "R20220721-P13",
      "Mandatory": false,
      "ScheduledBy": "CUSTOMER",
      "ScheduledTime": 1677348395000,
      "Severity": "HIGH",
      "Status": "PENDING_UPDATE",
      "Type": "SERVICE_SOFTWARE_UPDATE"
```

```
    }
}
```

如果请求失败并出现`SlotNotAvailableException`，则表示出于容量原因指定的时间不可用，您必须指定其他时间。 OpenSearch 服务将在响应中提供其他可用时段建议。

从自动调整维护窗口迁移

如果域是在 2023 年 2 月 16 日之前创建，则可以使用维护窗口来计划需要蓝绿部署的自动调整优化。您可以迁移现有的自动调整域，以改为使用非高峰窗口。

> **Note**
>
> 将域迁移到使用非高峰窗口后，您将无法恢复到使用维护窗口。

控制台

要将域迁移到使用非高峰窗口

1. 在 Amazon S OpenSearch ervice 控制台中，选择域的名称以打开其配置。
2. 转到自动调整选项卡，选择编辑。
3. 选择迁移到非高峰窗口。
4. 对于开始时间 (UTC)，采用协调世界时间 (UTC) 为非高峰窗口的每日开始时间。
5. 选择 Save changes（保存更改）。

CLI

要使用从自动调整维护窗口迁移到非高峰窗口 Amazon CLI，请发UpdateDomainConfig送请求：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=[]
```

要将域从自动调整维护窗口迁移到非高峰窗口，必须打开非高峰窗口。您可以在单独的请求中或在同一个请求中启用非高峰窗口。有关说明，请参阅the section called “启用非高峰窗口”。

亚马逊 OpenSearch 服务中的通知

Amazon S OpenSearch ervice 中的通知包含有关您的域名的性能和运行状况的重要信息。OpenSearch 服务会通知您有关服务软件更新、Auto-Tune 增强功能、集群运行状况事件和域错误的信息。通知适用于所有版本的 Elasticsearch OSS。 OpenSearch

您可以在 OpenSearch 服务控制台的通知面板中查看通知。所有 OpenSearch 服务通知也都显示在 Amazon EventBridge 中。有关通知的完整列表和示例事件，请参阅the section called “监控事件”。

开始使用通知

创建域时会自动启用通知。转到 OpenSearch 服务控制台的通知面板以监控和确认通知。每个通知包括发布时间、相关域、严重性和状态级别以及简要说明等信息。您可以在控制台中查看长达 90 天的历史通知。

访问 Notifications（通知）面板或确认通知后，您可能会收到关于没有权限执行 `es:ListNotificationsV2` 或 `es:UpdateNotificationStatus` 的错误消息。要解决此问题，请在 IAM 中向您的用户或角色授予以下权限：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "es:UpdateNotificationStatus",
      "es:ListNotifications"
    ],
    "Resource": "arn:aws:es:*:123456789012:domain/*"
  }]
}
```

IAM 控制台抛出一个错误（“IAM does not recognize one or more actions." [IAM 无法识别一项或多项操作。]”），您可以放心地忽略该错误。您也可以限制对某些域进行 `es:UpdateNotificationStatus` 操作。要了解更多信息，请参阅the section called “策略元素参考”。

通知严重性

OpenSearch 服务中的通知可以是信息性的，与您已经采取的任何操作或域名的操作有关，也可以是可操作的，需要您采取特定的操作，例如应用强制性的安全补丁。每个通知均包含与其关联的严重性，可以是 `Informational`、`Low`、`Medium`、`High` 或 `Critical`。下表汇总了每种严重性：

严重性	描述	示例
`Informational`	与您的域操作相关的信息。	• 可用服务软件更新 • 自动调整已开始
`Low`	建议操作，但如果不采取任何操作，则不会对域可用性或性能产生不利影响。	• 自动调整已取消 • 高分片数警告
`Medium`	如果不采取建议的操作则可能会产生影响，但会带来更长的时间窗口。	• 服务软件更新失败 • 超出分片数限制
`High`	需要采取紧急操作以免产生不利影响。	• 已要求服务软件更新 • KMS 密钥不可访问
`Critical`	需要立即采取操作以免产生不利影响，或从中还原。	当前无可用

示例 EventBridge 事件

以下示例显示了发送给 Amazon 的 OpenSearch 服务通知事件 EventBridge。由于更新可选，通知的严重性为 `Informational`：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
```

```
  "detail": {
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] available."
  }
}
```

在 Ama OpenSearch zon 服务中配置多可用区域

为了防止数据丢失并最大限度地减少 OpenSearch 服务中断时的 Amazon Service 集群停机时间，您可以在同一区域的两个或三个可用区中分配节点，这种配置称为多可用区。可用区是每个 Amazon 区域内的隔离位置。

对于运行生产工作负载的域，建议使用带待机功能的多可用区部署选项，该选项可创建以下配置：

- 跨三个区域部署域。
- 为专用主节点和数据节点选择最新一代实例类型。
- 三个专用主节点和三个（或三倍）数据节点。
- 您的域中每个索引至少有两个副本，或者三倍的数据副本（包括主节点和副本）。

本节的其余部分提供了有关这些配置的解释和背景。

带待机功能的多可用区

带备用模式的多可用区是 Amazon Ser OpenSearch vice 域的部署选项，可提供 99.99% 的可用性、稳定的生产工作负载性能，并简化域配置和管理。当您使用带待机功能的多可用区时，域可以抵御基础设施故障，而不会影响性能或可用性。此部署选项通过强制执行一些最佳实践（例如指定的数据节点数、主节点数、实例类型、副本数量、软件更新设置和开启自动调整）来达到这一标准。

> **Note**
>
> 带备用模式的多可用区适用于 1.3 及更高 OpenSearch 版本，并且需要至少有三个可用区的区域。

当您将多可用区与备用区域配合使用时，S OpenSearch ervice 会跨三个可用区创建一个域，每个可用区都包含完整的数据副本，并且数据在每个区域中均匀分布。您的域让其中一个区中的节点处于待机状

态，这意味着它们不响应搜索请求。当 S OpenSearch ervice 检测到底层基础设施出现故障时，它会在不到一分钟的时间内自动激活备用节点。域可以继续提供索引和搜索请求服务，而影响仅限于执行故障转移所耗费的时间。但不会对数据或资源进行重新分配，所以集群性能不会受到影响，而且也不会出现可用性下降的风险。带待机功能的多可用区完全免费。

您可以通过两种方式在 Amazon Web Services Management Console上创建带有待机状态的域。首先，您可以使用 E asy create 创建方法创建域，然后 S OpenSearch ervice 将自动使用预先确定的配置，其中包括以下内容：

- 三个可用区，其中一个可处于待机状态
- 三个专用的主节点和数据节点
- 已在域上启用自动调整
- GP3 数据节点的存储

您也可以选择标准创建创建方法，然后选择不带待机状态的域作为部署选项。您可以自定义域，同时仍必需使用具有待机状态的域的关键功能，例如三个区和三个主节点。建议选择三倍（可用区数量）的数据节点数。

创建域后，您可以导航到域详细信息页面，然后在集群配置选项卡中，确认可用区下显示带待机功能的 3-AZ。

如果您在将现有域迁移到带待机功能的多可用区时遇到问题，请参阅故障排除指南中的迁移到带待机功能的多可用区时出错。

限制

在设置带待机功能的多可用区的域时，请考虑以下限制：

- 节点上的分片总数不能超过 1000，集群上的分片总数不能超过 75000，单个分片的大小不能超过 65 GB。
- 带备用模式的多可用区仅适用`m5`于`c5`、`r5`、`r6g`、`r7g`、`c6g`、`m6g`、`r6gd`和`i3`实例类型。有关所支持实例类型的更多信息，请参阅支持的实例类型。
- 您只能使用预配置 IOPs SSD、通用型 SSD (GP3) 或带待机状态的实例备份存储。
- 如果您在带有备用域的多可用区UltraWarm上启用，则温节点的数量必须是正在使用的可用区数量的倍数。

不带待机功能的多可用区

OpenSearch 服务仍支持不带备用状态的多可用区，可用性高达 99.9%。节点分布在各个可用区中，可用性取决于可用区的数量和数据副本。而在带有待机功能的情况下，您必须按照最佳实践配置您的域，而在没有待机功能的情况下，您可以自己选择可用区、节点和副本的数量。除非您的现有工作流程会因创建带有待机功能的域而中断，否则我们不建议使用此选项。

如果您选择此选项，我们仍然建议您选择三个可用区，以保持对节点、磁盘和单可用区故障的弹性。发生故障时，集群会在剩余资源之间重新分配数据，以保持可用性和冗余。这种数据移动会增加集群的资源使用量，并可能对性能产生影响。如果集群的大小不正确，其可用性可能会降低，这在很大程度上违背了多可用区的目的。

配置未开启备用域的唯一方法 Amazon Web Services Management Console 是选择标准创建方法，然后选择不带备用状态的域作为部署选项。

分片分配

如果您启用不带待机功能的多可用区，则应该为集群中的每个索引创建至少一个副本。如果没有副本，S OpenSearch ervice 就无法将您的数据副本分发到其他可用区。幸运的是，所有索引的默认配置均为副本数量等于 1。如下图所示，S OpenSearch ervice 会尽最大努力将主分片及其对应的副本分片分配到不同的区域。

除了按可用区分配分片外，S OpenSearch ervice 还按节点分发分片。但是，某些域配置可能会导致分片计数不平衡。考虑以下域：

- 5 个数据节点
- 5 个主分片
- 2 个副本
- 3 个可用区

在这种情况下，S OpenSearch ervice 必须超载一个节点，才能将主分片和副本分片分布到各个区域，如下图所示。

为避免这些可能导致单个节点紧张并损害性能的情况，如果您计划每个索引有两个或更多副本，建议您选择带待机功能的多可用区或一个为三的倍数的实例计数。

专用主节点分配

即使您在配置域时选择了两个可用区，S OpenSearch ervice 也会自动在三个可用区之间分配专用主节点。此分发有助于在区域遇到服务中断时防止集群停机。如果您使用推荐的三个专用主节点并且一个可

用区域关闭，则您的集群仍具有配额数量 (2) 的专用主节点，并且可以选择新的主节点。下图演示了此配置。

如果您选择三个可用区中不可用的较旧一代实例类型，则以下方案适用：

- 如果您为域选择了三个可用区，则 OpenSearch 服务会引发错误。请选择其他实例类型，然后重试。
- 如果您为域选择了两个可用区，S OpenSearch ervice 会将专用主节点分布在两个区域中。

可用区中断

可用区中断很少见，但仍会出现。下表列出了中断期间的不同多 AZ 配置和行为。表中的最后一行适用于带待机功能的多可用区，而所有其他行的配置仅适用于不带待机功能的多可用区。

一个区域中的可用区数量	您选择的可用区数量	专用主节点的数量	一个可用区遇到中断时的行为
2 或更多	2	0	停机时间。您的集群丢失了一半的数据节点，并且必须替换剩余可用区中的至少一个，然后才能选择主节点。
2	2	3	停机几率为 50/50。 OpenSearch 服务将两个专用主节点分配到一个可用区，将一个分配到另一个可用区： • 如果具有一个专用主节点的可用区遇到中断，则剩余可用区中的两个专用主节点可以选择主节点。 • 如果具有两个专用主节点的可用区遇到中断，则在剩余可用区恢复之前，集群不可用。
3 或更多	2	3	没有停机时间。 OpenSearch Service 会自动将专用主节点分布到三个可用区，因此剩下的两个专用主节点可以选出一个主节点。
3 或更多	3	0	无停机时间。大约三分之二的数据节点仍可用于选择主节点。
3 或更多	3	3	无停机时间。其余两个专用主节点可以选择主节点。

在所有配置中，无论原因如何，节点故障都可能导致集群的其余数据节点经历一段时间的负载增加，而 Ser OpenSearch vice 会自动配置新节点以替换现在缺少的节点。

例如，如果三区域配置中发生可用区中断，则三分之二的数据节点必须处理所有集群请求。当它们处理这些请求时，其余节点也会在新节点上线时将分片复制到新节点上，这可能会进一步影响性能。如果可用性对您的工作负载至关重要，请考虑向集群中添加资源以缓解此问题。

> **Note**
>
> OpenSearch 服务以透明方式管理多可用区域，因此您无法手动模拟可用区中断。

在 VPC 内启动您的亚马逊 OpenSearch 服务域

您可以将诸如亚马逊 OpenSearch 服务域之类的 Amazon 资源启动到虚拟私有云 (VPC) 中。VPC 是专为您服务的虚拟网络 Amazon Web Services 账户。它在逻辑上与 Amazon 云中的其他虚拟网络隔离。在 VPC 中放置 OpenSearch 服务域可实现 OpenSearch 服务与 VPC 内的其他服务之间的安全通信，无需互联网网关、NAT 设备或 VPN 连接。所有流量都安全地保存在 Amazon 云中。

> **Note**
>
> 如果您将 OpenSearch 服务域置于 VPC 内，则您的计算机必须能够连接到 VPC。此连接通常采用以下格式：VPN、过渡网关、托管网络或代理服务器。您无法从 VPC 外部访问您的域。

VPC 与公有域对比

以下是 VPC 域与公有域不同的一些方式。每个差异都有更加详细的介绍。

- 由于进行了逻辑隔离，与使用公共端点的域相比，驻留在 VPC 中的域有一层额外的安全性。
- 虽然可以从任何连接互联网的设备访问公有域，但 VPC 域需要某种形式的 VPN 或代理服务器。
- 与公共域相比，VPC 域在 控制台中显示的信息较少。具体而言，集群运行状况选项卡中不包含分片信息；不存在索引选项卡。
- 域端点采用不同的形式（`https://search-domain-name` 与 `https://vpc-domain-name`）。
- 您无法将基于IP的访问策略应用于驻留在 VPC 中的域，因为安全组已强制实施基于 IP 的访问策略。

限制

在 VPC 内运营 OpenSearch 服务域有以下限制：

- 如果在 VPC 中启动了新域，以后就不能将其切换为使用公共终端节点。反过来也是如此：如果使用公共终端节点创建了域，则以后就不能将该域放入 VPC 中，您必须创建一个新的域，然后迁移数据。
- 可以在 VPC 中启动域，也可以使用公共终端节点，但两者不能同时进行。在创建域时只能选择其一。
- 您无法在使用专用租赁的 VPC 内启动您的域。您必须使用具有设置为 Default 的租赁的 VPC。
- 将域放入 VPC 中之后，不能再将其移到其他 VPC 中，但可以更改子网和安全组设置。

- 要访问位于 VPC 内的域的默认 OpenSearch 仪表板安装，用户必须有权访问该 VPC。此过程因网络配置而异，但可能涉及连接到 VPN 或托管网络或使用代理服务器或过渡网关。要了解更多信息，请参阅 the section called “关于 VPC 域的访问策略”，Amazon VPC 用户指南和 the section called “控制对仪表板的访问权限 ”。

架构

为了提供支持 VPCs， OpenSearch 服务将终端节点放置在您的 VPC 的一个、两个或三个子网中。如果您为域启用多个可用区，则每个子网必须位于同一区域中的不同可用区中。如果您只使用一个可用区，S OpenSearch ervice 会将一个终端节点放入一个子网中。

下图显示了一个可用区的 VPC 架构。

下图显示了两个可用区的 VPC 架构。

OpenSearch 服务还会在 VPC 中为您的每个数据节点放置一个弹性网络接口 (ENI)。 OpenSearch 服务会根据您的子网地址范围为每个 ENI 分配一个私有 IP IPv4 地址。该服务还会分配这些 IP 地址的公有 DNS 主机名 (域终端节点)。您必须使用公共 DNS 服务将终端节点 (DNS 主机名) 解析为数据节点的相应 IP 地址：

- 如果您的 VPC 通过将`enableDnsSupport`选项设置为`true`（默认值）来使用亚马逊提供的 DNS 服务器，则 OpenSearch 服务终端节点的解析将成功。
- 如果您的 VPC 使用私有 DNS 服务器，并且该服务器可以访问公共权威 DNS 服务器来解析 DNS 主机名，则 OpenSearch 服务终端节点的解析也将成功。

因为 IP 地址可能会发生更改，所以您应定期解析域终端节点，以便可以始终访问正确的数据节点。我们建议您将 DNS 解析时间间隔设置为一分钟。如果您在使用客户端，还应确保客户端中的 DNS 缓存已清除。

从公有访问迁移到 VPC 访问

在创建域时，您会指定它是应该具有公共终端节点还是驻留在 VPC 中。一旦创建，就无法再切换，而只能创建一个新域，手动重建索引或迁移您的数据。快照提供了一种迁移数据的便捷方法。有关拍摄和还原快照的信息，请参阅the section called “创建索引快照”。

关于 VPC 域的访问策略

将您的 OpenSearch 服务域置于 VPC 内可提供固有的强大安全层。使用公有访问权限创建域时，终端节点将采用以下形式：

```
https://search-domain-name-identifier.region.es.amazonaws.com
```

如“公有”标签所示，此终端节点可从任何连接 Internet 的设备访问，但您可以 (且应该) 控制对它的访问。如果您访问 Web 浏览器中的终端节点，您可能会收到一条 `Not Authorized` 消息，但请求将达到域。

当您使用 VPC 访问权限创建域时，终端节点看起来 类似于公有终端节点：

```
https://vpc-domain-name-identifier.region.es.amazonaws.com
```

但是，如果您尝试访问 Web 浏览器中的终端节点，您可能会发现请求超时。要执行更基本的 `GET` 请求，您的计算机必须能够连接到 VPC。此连接通常采用以下格式：VPN、过渡网关、托管网络或代理服务器。有关此连接可以采用的各种格式的详细信息，请参阅 Amazon VPC 用户手册中的 VPC 示例。有关专注于开发的示例，请参阅 the section called “测试 VPC 域”。

除了此连接要求外，还 VPCs 允许您通过安全组管理对域的访问。对于许多使用案例，这种安全功能的组合方式已足够，并且您可能愿意将开放访问策略应用于域。

使用开放访问政策并不意味着互联网上的任何人都可以访问 OpenSearch 服务域。相反，这意味着如果请求到达 OpenSearch 服务域并且相关的安全组允许，则该域将接受该请求。唯一例外的情况是如果您使用细粒度访问控制或指定 IAM 角色的访问策略。在这些情况下，意味着要让域接受某个请求，安全组必须允许该请求并且必须使用有效凭证的签署该请求。

> **Note**
>
> 由于安全组已经强制执行基于 IP 的访问策略，因此您无法将基于 IP 的访问策略应用于 VPC 内的 OpenSearch 服务域。如果使用公有访问权限，则基于 IP 的策略仍可用。

开始之前：VPC 访问的先决条件

在启用 VPC 与新 OpenSearch 服务域之间的连接之前，必须执行以下操作：

- 创建 VPC

  要创建您的 VPC，您可以使用 Amazon VPC 控制台、 Amazon CLI 或其中一个 Amazon SDKs。有关更多信息，请参阅 Amazon VPC 用户指南 VPCs中的使用。如果您已有 VPC，请跳过此步骤。

- 预留 IP 地址

  OpenSearch 服务通过将网络接口放置在 VPC 的子网中来实现 VPC 与域的连接。每个网络接口都与一个 IP 地址关联。必须在子网中为网络接口预留足够数量的 IP 地址。有关更多信息，请参阅在 VPC 子网中预留 IP 地址。

测试 VPC 域

VPC 增强的安全性可能会使连接到您的域并运行基本测试成为挑战。如果您已经拥有 OpenSearch 服务 VPC 域并且不想创建 VPN 服务器，请尝试以下过程：

1. 对于域的访问策略，请选择 Only use fine-grained access control（仅使用精细访问控制）。完成测试之后，您可以随时更新此设置。
2. 在与您的 OpenSearch 服务域相同的 VPC、子网和安全组中创建 Amazon Linux Amazon EC2 实例。

   由于此实例用于测试目的且需要做的工作非常少，因此请选择一种便宜的实例类型（如 `t2.micro`）。为此实例分配一个公有 IP 地址，并创建一个新的密钥对或选择一个现有的密钥对。如果您创建新的密钥，请将其下载到您的 `~/.ssh` 目录。

   要了解有关创建实例的更多信息，请参阅 Amazon EC2 Linux 实例入门。
3. 将 Internet 网关添加到 VPC。
4. 在 VPC 的路由表中，添加新的路由。对于 Destination (目的地)，指定一个 CIDR 块，其中包含您的计算机的公有 IP 地址。对于 Target (目标)，请指定您刚刚创建的 Internet 网关。

例如，您可能只为您的计算机指定 `123.123.123.123/32`，或者为一系列计算机指定 `123.123.123.0/24`。

5. 对于安全组，指定两条入站规则：

类型	协议	端口范围	来源
SSH (22)	TCP (6)	22	*your-cidr-block*
HTTPS (443)	TCP (6)	443	*your-security-group-id*

第一条规则允许您通过 SSH 连接到您的 EC2 实例。第二种允许 EC2 实例通过 HTTPS 与 OpenSearch 服务域通信。

6. 从终端运行以下命令：

```
ssh -i ~/.ssh/your-key.pem ec2-user@your-ec2-instance-public-ip -N -L
 9200:vpc-domain-name-identifier.region.es.amazonaws.com:443
```

此命令创建一个 SSH 隧道，该隧道通过 EC2 实例将发送到 https://localhost:9200 的请求转发到您的 OpenSearch 服务域。在命令中指定端口 9200 会模拟本地 OpenSearch 安装，但要使用任何你想要的端口。 OpenSearch 服务仅接受通过端口 80 (HTTP) 或 443 (HTTPS) 进行的连接。

该命令不提供反馈并无限期运行。要停止此命令，请按 `Ctrl + C`。

7. 在你的网络浏览器中导航到 https://localhost:9200/_dashboards/。您可能需要确认安全异常。

或者，您可以使用 https://localhost:9200curl、Postman 或您常用的编程语言将请求发送到 。

> **Tip**
>
> 如果您由于证书不匹配而遇到 curl 错误，请尝试使用 `--insecure` 标记。

在 VPC 子网中预留 IP 地址

OpenSearch 服务通过将网络接口放置在 VPC（如果您启用多个可用区，则为 VPC 的多个子网）中放置网络接口，从而将域连接到 VPC。每个网络接口都与一个 IP 地址关联。在创建 OpenSearch 服务域之前，每个子网中必须有足够数量的 IP 地址来容纳网络接口。

以下是基本公式：S OpenSearch ervice 在每个子网中预留的 IP 地址数量等于数据节点数量的三倍除以可用区数量。

示例

- 如果某个域在 3 个可用区中有 9 个数据节点，则每个子网的 IP 数为 9 * 3 / 3 = 9。
- 如果某个域在 2 个可用区中有 8 个数据节点，则每个子网的 IP 数为 8 * 3 / 2 = 12。
- 如果某个域在一个可用区中有 6 个数据节点，则每个子网的 IP 数为 6 * 3 / 1 = 18。

创建域时，S OpenSearch ervice 会保留 IP 地址，将一些 IP 地址用于域，其余的保留用于蓝/绿部署。您可以在 Amazon EC2 控制台的 “网络接口” 部分查看网络接口及其关联的 IP 地址。描述列显示网络接口与哪个 OpenSearch 服务域相关联。

> **Tip**
>
> 我们建议您为 OpenSearch 服务预留 IP 地址创建专用子网。通过使用专用子网，可以避免与其他应用程序和服务重叠，并确保在将来需要扩展集群时可以预留额外 IP 地址。要了解更多信息，请参阅在 VPC 中创建子网。

您也可以考虑配置专用的协调器节点，以减少您的 VPC 域所需的私有 IP 地址预留数量。 OpenSearch 将弹性网络接口 (ENI) 附加到您的专用协调器节点，而不是您的数据节点。专用协调器节点数通常为总数据节点数的 10% 左右。这样就可以减少为 VPC 域保留的私有 IP 地址数量。

VPC 访问的服务相关角色

服务相关角色是一种独特的 IAM 角色类型，它将权限委派给服务，使之能够代表您创建和管理资源。OpenSearch 服务需要服务相关角色才能访问您的 VPC、创建域终端节点以及将网络接口放置在您的 VPC 的子网中。

OpenSearch 当您使用服务控制台在 VPC 内创建域时， OpenSearch 服务会自动创建角色。为使这种自动创建成功，您必须具有 `iam:CreateServiceLinkedRole` 操作的权限。有关更多信息，请参阅 IAM 用户指南中的服务相关角色权限。

OpenSearch 服务创建角色后，您可以使用 IAM 控制台查看该角色 (`AWSServiceRoleForAmazonOpenSearchService`)。

有关此角色的权限以及如何删除它的完整信息，请参阅the section called “使用服务相关角色”。

在 Amazon OpenSearch 服务中创建索引快照

Amazon S OpenSearch ervice 中的快照是集群索引和状态的备份。状态包含集群设置、节点信息、索引设置和分片分配。

OpenSearch 服务快照有以下几种形式：

- 自动快照仅用于集群恢复。在发生红色群集状态或数据丢失时，您可以使用它们还原域。有关更多信息，请参阅下面的恢复快照。 OpenSearch 该服务将自动快照存储在预配置的 Amazon S3 存储桶中，无需额外付费。
- 手动快照用于集群恢复或者将数据从一个集群移动到另一个集群。必须启动手动快照。这些快照将存储在您自己的 Amazon S3 存储桶中，收取标准 S3 费用。如果您有来自自管理 OpenSearch 集群的快照，则可以使用该快照迁移到 OpenSearch 服务域。有关更多信息，请参阅迁移到 Amazon OpenSearch 服务。

所有 OpenSearch 服务域都会自动拍摄快照，但频率在以下方面有所不同：

- 对于运行 OpenSearch 或 Elasticsearch 5.3 及更高版本的域名，S OpenSearch ervice 会按小时自动拍摄快照，并将其中最多 336 张快照保留 14 天。由于其增量性质，每小时快照的破坏性较小。如果出现域问题，它们还提供更新的恢复点。
- 对于运行 Elasticsearch 5.1 及更早版本的域名，S OpenSearch ervice 会在您指定的时间内每天自动拍摄快照，最多保留 14 张快照，并且不会将任何快照数据保留超过 30 天。

如果您的集群进入红色状态，则所有自动快照都会失败，而集群状态仍然存在。如果您在两周内未解决问题，则可能会永久丢失集群内的数据。有关问题排查步骤，请参阅the section called “红色集群状态”。

先决条件

要手动创建快照，您必须使用 IAM 和 Amazon S3。确保您已满足以下先决条件，然后再尝试创建快照。

先决条件	描述
S3 存储桶	创建 S3 存储桶以存储 OpenSearch 服务域的手动快照。有关说明，请参阅《Amazon 简单存储服务用户指南》中的创建通用存储桶。 记住要在以下位置使用它的存储桶名称： • 附加到 IAM 角色的 IAM policy 的 `Resource` 语句 • 用于注册快照存储库的 Python 客户端（如果使用此方法） **Important** 请勿对此存储桶应用 S3 Glacier 生命周期规则。手动快照不支持 S3 Glacier 存储类。
IAM 角色	创建 IAM 角色以向 OpenSearch 服务委派权限。有关说明，请参阅 IAM 用户指南中的创建 IAM 角色（控制台）。本章剩余部分将此角色称为 `TheSnapshotRole` 。 附加 IAM policy 将下面的策略附加到 `TheSnapshotRole` 以允许访问 S3 存储桶： JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3::: amzn-s3-demo-bucket  "
      ]
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
```

先决条件	描述

```
        "s3:DeleteObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3::: amzn-s3-demo-bucket  /*"
      ]
    }
  ]
}
```

有关向角色关联策略的说明，请参阅 IAM 用户指南中的添加 IAM 身份权限（控制台）。

编辑信任关系

编辑的信任关系`TheSnapshotRole` 以在`Principal` 语句中指定 S OpenSearch ervice，如以下示例所示：

JSON

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

}
```

有关编辑信任关系的说明，请参阅 IAM 用户指南中的更新角色信任策略。

先决条件	描述

权限	要注册快照存储库，您需要能够传递TheSnapshotRole 给 OpenSearch 服务。还需要对 es:ESHttpPut 操作的访问权限。要授予这两个权限，请将以下策略附加到 IAM 角色，该角色的凭据用于签署请求：
	
	JSON
	
	``
	{
	  "Version": "2012-10-17",
	  "Statement": [
	    {
	      "Effect": "Allow",
	      "Action": "iam:PassRole",
	      "Resource": "arn:aws:iam:: 123456789012 :role/TheSnapsh
	otRole "
	    },
	    {
	      "Effect": "Allow",
	      "Action": "es:ESHttpPut",
	      "Resource": "arn:aws:es: region:123456789
	012 :domain/domain-name /*"
	    }
	  ]
	}
	``
	
	如果您的用户或角色没有 iam:PassRole 权限传递 TheSnapshotRole ，在下一步骤中尝试注册存储库时，您可能会遇到以下常见错误：
	
	``
	$ 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 "}
	``
	
	先决条件 482

注册手动快照存储库

您需要先向 S OpenSearch ervice 注册快照存储库，然后才能手动拍摄索引快照。此一次性操作要求您使用允许访问的凭据签署 Amazon 请求`TheSnapshotRole`，如中所述the section called “先决条件”。

步骤 1：在 OpenSearch 仪表板中映射快照角色（如果使用精细的访问控制）

注册存储库时，精细访问控制会引入额外的步骤。即使将 HTTP 基本身份验证用于所有其他目的，也需要将 `manage_snapshots` 角色映射到具有传递 `TheSnapshotRole` 的 `iam:PassRole` 权限的 IAM 角色。

1. 导航到您的 OpenSearch 服务域的 OpenSearch 仪表板插件。您可以在 OpenSearch 服务控制台的域控制面板上找到控制面板终端节点。
2. 从主菜单中选择安全、角色，然后选择 manage_snapshots 角色。
3. 选择映射的用户、管理映射。
4. 添加具有传递 `TheSnapshotRole` 权限的角色的 ARN。将角色置 ARNs 于 “后端角色” 下。

   ```
   arn:aws:iam::123456789123:role/role-name
   ```

5. 选择映射并确认在映射的用户下显示的用户或角色。

第 2 步：注册存储库

以下快照选项卡演示如何注册快照目录。有关在迁移到新域后加密手动快照和注册快照的特定选项，请参阅相关选项卡。

Snapshots

要注册快照存储库，请向 OpenSearch 服务域端点发送 PUT 请求。您可以使用 curl、示例 Python 客户端、Postman 或某种其他方式发送已签名请求以注册快照存储库。请注意，您不能在控制 OpenSearch 面板控制台中使用 PUT 请求来注册存储库。

请求的格式如下：

```
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**
>
> 存储库名称不能以“cs-”开头。此外，您不应该从多个域写入同一个存储库。应该只有一个域具有对存储库的写入权限。

如果域位于某个 Virtual Private Cloud（VPC）中，则必须将您的电脑连接到该 VPC，请求才能成功注册快照存储库。访问 VPC 因网络配置而异，但很可能包括连接到 VPN 或企业网络。要检查您是否可以访问 OpenSearch 服务域，请在 Web 浏览器`https://your-vpc-domain.region.es.amazonaws.com`中导航到并验证您是否收到了默认 JSON 响应。

当您的 Amazon S3 存储桶位于 Amazon Web Services 区域 其他 OpenSearch域中时，请将参数`"endpoint": "s3.amazonaws.com"`添加到请求中。

Encrypted snapshots

您目前无法使用 Amazon Key Management Service (KMS) 密钥对手动快照进行加密，但您可以使用服务器端加密 (SSE) 对其进行保护。

要为您用作快照存储库的存储桶启用使用 S3 托管密钥的 SSE，请将 `"server_side_encryption": true` 添加到 PUT 请求 `"settings"` 数据块。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用 Amazon S3 托管式密钥服务器端加密（SSE-S3）。

或者，您也可以在用作快照存储库的 S3 存储桶上使用 Amazon KMS 密钥进行服务器端加密。如果您使用这种方法，请确保为用于加密 S3 存储桶的 Amazon KMS 密钥提供`TheSnapshotRole`权限。有关更多信息，请参阅Amazon KMS中的密钥策略。

Domain migration

注册快照存储库是一次性操作。但要从一个域迁移到另一个域，您必须在旧域和新域中注册相同的快照存储库。存储库名称是任意的。

迁移到新域或使用多个域注册同一存储库时，请考虑以下准则：

- 在新域中注册存储库时，将 `"readonly": true` 添加到`"settings"` PUT 请求数据库。此设置可防止您意外覆盖旧域中的数据。应该只有一个域具有对存储库的写入权限。
- 如果您要将数据迁移到不同 Amazon Web Services 区域中的域（例如，从位于 us-east-2 的旧域和存储桶迁移到 us-west-2 中的新域），请将 `"region": "region"` 替换为 PUT 语句中的 `"endpoint": "s3.amazonaws.com"`，然后重试请求。

使用示例 Python 客户端

Python 客户端比简单的 HTTP 请求更容易自动化，并且具有更好的可重用性。如果您选择使用此方法注册快照存储库，请将下面的示例 Python 代码保存为 Python 文件，如 `register-repo.py`。客户端需要 适用于 Python (Boto3) 的 Amazon SDK、requests 和 requests-aws4auth 程序包。客户端包含其他快照操作的带注释示例。

更新示例代码中的以下变量：`host`、`region`、`path` 和 `payload`。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = '' # domain endpoint
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Register repository

path = '/_snapshot/my-snapshot-repo-name' # the OpenSearch API endpoint
url = host + path

payload = {
  "type": "s3",
  "settings": {
    "bucket": "amzn-s3-demo-bucket",
    "base_path": "my/snapshot/directory",
    "region": "us-west-1",
    "role_arn": "arn:aws:iam::123456789012:role/snapshot-role"
  }
}
```

```
headers = {"Content-Type": "application/json"}

r = requests.put(url, auth=awsauth, json=payload, headers=headers)

print(r.status_code)
print(r.text)

# # Take snapshot
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot'
# url = host + path
#
# r = requests.put(url, auth=awsauth)
#
# print(r.text)
#
# # Delete index
#
# path = 'my-index'
# url = host + path
#
# r = requests.delete(url, auth=awsauth)
#
# print(r.text)
#
# # Restore snapshot (all indexes except Dashboards and fine-grained access control)
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
# url = host + path
#
# payload = {
#   "indices": "-.kibana*,-.opendistro_security,-.opendistro-*",
#   "include_global_state": False
# }
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
#
# # Restore snapshot (one index)
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
```

```
# url = host + path
#
# payload = {"indices": "my-index"}
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
```

手动创建快照

快照不是即时发生的。它们需要时间才能完成，并且不能代表集群的完美 point-in-time视图。当快照正在进行时，您仍可以对文档编制索引并对集群发出其他请求，但新文档（和对现有文档的更新）通常不包含在快照中。快照包括 OpenSearch 启动快照时存在的主分片。根据快照线程池的大小，快照中可能包含时间略有不同的不同分片。有关快照最佳实践，请参阅 the section called “提高快照性能”。

快照存储和性能

OpenSearch 快照是增量的，这意味着它们仅存储自上次成功快照以来更改的数据。此增量性质意味着频繁快照与不频繁快照之间的磁盘使用率差异通常极其小。换句话说，一周内每小时快照（总共 168 个快照）占用的磁盘空间比一周结束时的一个快照所占用的磁盘空间并不高多少。此外，拍摄快照的频率越高，完成快照所需的时间就更少。例如，每日快照可能需要 20-30 分钟才能完成，而每小时快照可能在几分钟内即可完成。有些 OpenSearch 用户每半小时拍摄一次快照。

拍摄快照

创建参数时，您需要指定以下信息：

- 快照存储库的名称
- 快照名称

为了方便和简洁起见，本章中的示例使用 curl，这是一种常见的 HTTP 客户端。要向 curl 请求传递用户名和密码，请参阅入门教程。

如果访问策略指定用户或角色，您必须签署快照请求。对于 curl，您可以在 7.75.0 或更高版本中使用 `--aws-sigv4` 选项。您也可以使用示例 Python 客户端中的带注释示例将签名 HTTP 请求置于 curl 命令使用的同一端点。

要制作手动快照，请执行以下步骤：

1. 如果当前正在制作快照，则您无法制作快照。要进行检查，请运行以下命令：

```
curl -XGET 'domain-endpoint/_snapshot/_status'
```

2. 运行以下命令来手动创建快照：

```
curl -XPUT 'domain-endpoint/_snapshot/repository-name/snapshot-name'
```

要包括或排除某些索引并指定其他设置，请添加请求正文。有关请求结构，请参阅 OpenSearch 文档中的拍摄快照。

> **Note**
>
> 拍摄快照所需的时间会随着 OpenSearch 服务域的大小而增加。长时间运行的快照操作有时会遇到以下错误：`504 GATEWAY_TIMEOUT`。通常情况下，您可以忽略这些错误并等待操作成功完成。运行以下命令验证您的域中所有快照的状态：
>
> ```
> curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
> ```

还原快照

在恢复快照之前，请确保目标域不使用带待机功能的多可用区。启用待机状态会导致恢复操作失败。

> **Warning**
>
> 如果您使用索引别名，应在删除别名的索引前停止向该别名写入请求（或将别名切换至其他索引）。停止写入请求有助于避免以下情景：
>
> 1. 您删除某个索引，同时会删除它的别名。
> 2. 对于现已删除的别名的错误写入请求会创建一个与别名同名的新索引。
> 3. 由于与新索引的命名冲突，您无法再使用别名。如果将别名切换到其他索引，请在从快照中还原时指定 `"include_aliases": false`。

还原快照

1. 确定要还原的快照。确保此索引的所有设置（例如，自定义分析器软件包或分配要求设置）均与域兼容。要查看所有快照存储库，请运行以下命令：

```
curl -XGET 'domain-endpoint/_snapshot?pretty'
```

在确定存储库后，您可以运行以下命令查看所有快照：

```
curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
```

> **Note**
>
> 大多数自动快照存储在 `cs-automated` 存储库中。如果您的域对静态数据进行加密，这些快照将存储在 `cs-automated-enc` 存储库中。如果您没有看到要查找的手动快照存储库，请确保您已向域注册该存储库。

2. （可选）如果群集上的索引与快照中的索引之间存在命名冲突，请删除或重命名 OpenSearch 服务域中的一个或多个索引。您无法将索引的快照还原到已包含同名索引的 OpenSearch 集群中。

   如果索引命名冲突，您可以使用以下选项：

   - 删除现有 OpenSearch 服务域上的索引，然后恢复快照。
   - 从快照还原索引时为其重命名，之后为它们重新编制索引。要了解如何重命名索引，请参阅 OpenSearch 文档中的此示例请求。
   - 将快照还原到不同的 OpenSearch 服务域（只能使用手动快照）。

   以下命令将删除域中的所有现有索引：

```
curl -XDELETE 'domain-endpoint/_all'
```

   但是，如果您不打算还原所有索引，则可以仅删除一个索引：

```
curl -XDELETE 'domain-endpoint/index-name'
```

3. 要还原快照，请运行以下命令：

```
curl -XPOST 'domain-endpoint/_snapshot/repository-name/snapshot-name/_restore'
```

由于 OpenSearch 仪表板上的特殊权限和精细的访问控制索引，尝试恢复所有索引可能会失败，尤其是在您尝试从自动快照还原时。以下示例通过 `my-index` 快照存储库中的 `2020-snapshot` 来只还原一个索引 `cs-automated`：

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "my-index"}' \
-H 'Content-Type: application/json'
```

您还可能需要还原除控制面板和精细访问控制索引以外的所有索引：

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "-.kibana*,-.opendistro*"}' \
-H 'Content-Type: application/json'
```

使用 `rename_pattern` 和 `rename_replacement` 参数，即可在不删除快照数据的情况下恢复快照。有关这些参数的更多信息，请参阅 OpenSearch 文档中的恢复快照 API 请求字段和示例请求。

> **Note**
>
> 如果并非所有主分片都适用于涉及的索引，则快照的 `state` 可能为 `PARTIAL`。此值表示未成功存储至少一个分片中的数据。您仍可以从部分快照进行还原，但可能需要使用较旧的快照来还原任何缺失的索引。

删除手动快照

要删除手动快照，运行下列命令：

```
DELETE _snapshot/repository-name/snapshot-name
```

使用快照管理自动处理快照

您可以在 OpenSearch 控制面板中设置快照管理 (SM) 策略，以自动创建和删除定期快照。SM 可以为一组索引创建快照，而索引状态管理只能为每个索引创建一个快照。要在 OpenSearch 服务中使用 SM，您需要注册自己的 Amazon S3 存储库。有关注册存储库的说明，请参阅注册手动快照存储库。

在 SM 之前，S OpenSearch ervice 提供免费的自动快照功能，默认情况下该功能仍处于开启状态。此功能将快照发送到服务维护 `cs-*` 存储库。要停用此功能，请与 Amazon Web Services 支持联系。

有关 SM 功能的更多信息，请参阅 OpenSearch 文档中的快照管理。

SM 目前不支持基于多种索引类型创建快照。例如，如果尝试基于多个包含 * 的索引创建快照且部分索引位于暖层，则快照创建将失败。如果需要快照包含多种索引类型，请使用 ISM 快照操作，直到 SM 支持此选项。

配置 权限

如果您要从以前的 OpenSearch 服务域版本升级到 2.5，则可能无法在该域上定义快照管理安全权限。必须将非管理员用户映射到此角色，才能通过精细访问控制在域上使用快照管理。要手动创建快照管理角色，请执行下列步骤：

1. 在 “ OpenSearch 控制面板” 中，转至 “安全”，然后选择 “权限”。
2. 选择创建操作组并配置以下组：

组名	权限
`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. 选择角色和创建角色。
4. 将角色命名为 snapshot_management_role。

5. 对于集群权限，选择 `snapshot_management_full_access` 和 `snapshot_management_read_access`。
6. 选择创建。
7. 创建角色之后，将其映射到将管理快照的任何用户或后端角色。

注意事项

配置快照管理时，请考虑以下事项：

- 每个存储库允许使用一个策略。
- 一个策略最多允许 400 个快照。
- 如果域名处于红色状态、JVM 压力过高（85% 或以上）或快照功能卡住，则无法运行此功能。当集群的整体索引和搜索性能受到影响时，SM 也可能受到影响。
- 只有完成上一个快照操作后才能开始下一个快照操作，因此无法通过一项策略激活并发快照操作。
- 如果多个策略采用相同的计划，可能会导致资源激增。如果策略的快照索引重叠，则只能按顺序运行分片级别快照操作，因而可能引发级联性能问题。如果多个策略共享一个存储库，则该存储库的写入操作将激增。
- 除非存在特殊用例，否则我们建议您将快照操作自动化时间间隔设置为每小时不超过一次。

使用索引状态管理自动执行快照

您可以使用索引状态管理 (ISM) 快照操作根据索引的期限、大小或文档数量的变化自动触发索引的快照。如果需要为每个索引创建一个快照，ISM 是最佳选择。如果需要为一组索引创建快照，请参阅 使用快照管理自动处理快照。

要在 OpenSearch 服务中使用 SM，您需要注册自己的 Amazon S3 存储库。有关使用`snapshot`操作的 ISM 示例，请参阅示例策略。

将 Curator 用于快照

如果 ISM 不适用于索引和快照管理，则可以改用 Curator。它提供了高级筛选功能，可帮助简化复杂集群上的任务。使用 pip 安装 Curator：

```
pip install elasticsearch-curator
```

您可以使用 Curator 作为命令行界面 (CLI) 或 Python API。如果您使用 Python API，则必须使用旧式 elasticsearch-py 客户端的版本 7.13.4 或更早版本。它不支持 opensearch-py 客户端。

如果您使用 CLI，请在命令行处导出您的凭证并配置 `curator.yml`，如下所示：

```
client:
  hosts: search-my-domain.us-west-1.es.amazonaws.com
  port: 443
  use_ssl: True
  aws_region: us-west-1
  aws_sign_request: True
  ssl_no_validate: False
  timeout: 60

logging:
  loglevel: INFO
```

升级 Amazon S OpenSearch ervice

> **Note**
>
> OpenSearch 而且 Elasticsearch 版本升级不同于服务软件更新。有关为您的 OpenSearch 服务域更新服务软件的信息，请参阅the section called “服务软件更新”。

Amazon Ser OpenSearch vice 为运行 OpenSearch 1.0 或更高版本或 Elasticsearch 5.1 或更高版本的域提供了就地升级。如果您使用 Amazon Data Firehose 或 Amazon CloudWatch Logs 等 OpenSearch 服务将数据流式传输至 Service，请在迁移 OpenSearch 前检查这些服务是否支持较新版本的。

支持的升级途径

当前，S OpenSearch ervice 支持以下升级路径：

之前版本	目标版本
OpenSearch 1.x x	OpenSearch 2.。 x OpenSearch 如果域满足以下条件，则默认情况下，2.17 将在自动模式下启用并行区段搜索：

之前版本	目标版本
	• 之前没有明确设置并发搜索设置。 • 所有数据实例（热实例和温实例）的实例类型均为 2.xl 或更高。 • 超过 1 周的数据实例（热实例和温实例）的平均 p90 CPU 使用率低于 45%。 有关此处并发区段搜索设置的更多详细信息，请参阅并发区段搜索。 版本 2.3 具有以下重大更改： • 在版本 2.0 中，从所有 OpenSearch API 端点中移除了`type`参数。有关更多信息，请参阅 breaking changes（重大更改）。 • 如果您的域包含最初在 Elasticsearch 6.8 中创建的任何索引（热索引、冷索引），则这些索引与 2.3 不兼容。 UltraWarm OpenSearch 在升级到版本 2.3 之前，必须为不兼容的索引重新编制索引。对于不兼容 UltraWarm 或冷索引，请将它们迁移到热存储，重新索引数据，然后再将它们迁移回暖存储或冷存储。或者，如果您不再需要索引，可以删除它们。 如果您意外将域升级到版本 2.3，而没有先执行这些步骤，则无法将不兼容的索引从其当前存储层迁移出去。您唯一的选择是删除它们。
OpenSearch 1.。 x	OpenSearch 1.。 x
Elasticsearch 7.x	Elasticsearch 7 x 或 OpenSearch 1。 x **Important** OpenSearch 1.。 x 引入了许多重大更改。有关更多信息，请参阅 Amazon OpenSearch 服务重命名。

之前版本	目标版本
Elasticsearch 6.8	Elasticsearch 7 x 或 OpenSearch 1。 x **Important** Elasticsearch 7.0 和 OpenSearch 1.0 包含许多突发性更改。在启动就地升级之前，我们建议您手动拍摄 6 的快照。 x 域，在测试中将其恢复 7. x 或 OpenSearch 1。 x 域，并使用该测试域来识别潜在的升级问题。有关 OpenSearch 1.0 中的突破性变更，请参阅Amazon OpenSearch 服务重命名。 与 Elasticsearch 6.x 相似，索引只能包含一种映射类型，但该类型现在必须名为 `_doc`。因此，某些请求正文中 APIs 不再需要映射类型（例如 `_bulk` API）。 对于新索引，请使用自托管的 Elasticsearch 7。 x 和 OpenSearch 1。 x 的默认分片数为 1。 OpenSearch Elasticsearch 上的服务域 7. x 及更高版本保留之前的默认值 5。
Elasticsearch 6.x	Elasticsearch 6.x
Elasticsearch 5.6	Elasticsearch 6.x **Important** 在版本 6.x 中创建的索引不再支持多个映射类型。在版本 5.x 中创建的索引在还原到 6.x 集群后仍然支持多个映射类型。请检查您的代码是否仅为每个索引创建一个映射类型。 尽量减少从 Elasticsearch 5.6 升级到 6 期间的停机时间。 x，S OpenSearch ervice 将索引重新索引到`.kibana-6` `.kibana`、删除、创建名为的别名`.kibana`，并将新索引映射到新别名。`.kibana`
Elasticsearch 5.x	Elasticsearch 5.x

升级过程包括三个步骤：

1. 升级前检查- OpenSearch 服务检查可能阻碍升级的问题，除非这些检查成功，否则不会继续下一步。
2. 快照-S OpenSearch ervice 拍摄 OpenSearch 或 Elasticsearch 集群的快照，除非快照拍摄成功，否则不会继续下一步。如果升级失败，S OpenSearch ervice 将使用此快照将集群还原到其原始状态。有关更多信息，请参阅 the section called “升级后无法降级”。
3. 升级- OpenSearch 服务开始升级，这可能需要 15 分钟到几个小时才能完成。 OpenSearch 控制面板可能在部分或全部升级中不可用。

升级域（控制台）

升级过程是不可撤消的，并且无法暂停或取消。在升级过程中，您无法对域进行配置更改。在开始升级之前，请仔细确认您是否要继续。您可以使用这些相同步骤执行升级前检查而不实际开始升级。

如果集群具有专用主节点，则无需停机即可完成 OpenSearch 升级。否则，集群在选择主节点时可能会在升级后几秒钟无响应。

将域升级到更高版本 OpenSearch 或 Elasticsearch

1. 创建您的域的手动快照。此快照用作备份，如果需要恢复为使用以前的 OpenSearch 版本，您可以在新域上恢复
2. 前往https://aws.amazon.com并选择 “登录到控制台”。
3. 在 “分析” 下，选择 “亚马逊 OpenSearch 服务”。
4. 在导航窗格中的 Domains（域）下，选择要升级的域。
5. 选择 Actions（操作）和 Upgrade（升级）。
6. 选择要升级到的版本。如果要升级到 OpenSearch 版本，出现启用兼容模式选项。如果启用此设置，会将其版本 OpenSearch 报告为 7.10，以允许 Elasticsearch OSS 客户端和插件（如日志记录）继续使用 Amazon Service。 OpenSearch 可以稍后禁用此设置
7. 选择 Upgrade。
8. 检查域控制面板上的 Status（状态）以监控升级的状态。

升级域（CLI）

您可以通过以下操作为您的域识别 OpenSearch 或 Elasticsearch 的正确版本，启动就地升级，执行升级前检查，并查看进度：

- `get-compatible-versions` (`GetCompatibleVersions`)

- `upgrade-domain` (`UpgradeDomain`)
- `get-upgrade-status` (`GetUpgradeStatus`)
- `get-upgrade-history` (`GetUpgradeHistory`)

有关更多信息，请参阅 Amazon CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

升级域（SDK）

此示例使用中的OpenSearchService低级别 Python 客户端， Amazon SDK for Python (Boto) 以检查域是否符合升级到特定版本的条件，对其进行升级，并持续检查升级状态。

```
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()
```

对验证失败进行故障排除

启动 OpenSearch 或 Elasticsearch 版本升级时，Serv OpenSearch ice 首先执行一系列验证检查，以确保您的域符合升级条件。如果其中任何一项检查失败，您将收到通知，其中包含在升级域之前必须修

复的特定问题。有关潜在问题及其解决步骤的列表，请参阅 the section called “对验证错误进行故障排除”。

排查升级问题

就地 升级需要正常运行的域。您的域可能不符合升级条件或出于各种原因无法升级。下表显示了最常见的问题。

事务	描述
不支持可选插件	当您使用可选插件升级域时，S OpenSearch ervice 也会自动升级插件。因此，您的域的目标版本还必须支持这些可选插件。如果域安装了目标版本不可用的可选插件，则升级请求将失败。
一个节点的分片过多	OpenSearch，以及 7。 x 如果当前集群中的节点超过了此设置，S OpenSearch ervice 不允许您升级。有关问题排查选项，请参阅the section called “超过最大分片限制”。
域正在处理中	域正在接受配置更改。在操作完成后检查升级资格。
红色集群状态	集群中的一个或多个索引为红色。有关问题排查步骤，请参阅the section called “红色集群状态”。
高错误率	在尝试处理请求时，集群返回大量 5xx 错误。此问题通常是因为同时读取或写入了过多的请求。请考虑减少流向集群的流量或扩展您的域。
裂脑	脑裂意味着您的集群有多个主节点并且已拆分成两个绝不会自行重新联接的集群。您可以通过使用推荐数量的专用主节点避免裂脑。为了帮助您从裂脑恢复，请联系 Amazon Web Services 支持。
找不到主节点	OpenSearch 服务无法找到集群的主节点。如果您的域使用了多 AZ，一个可用区故障可能已导致集群失去仲裁节点数且无法选择新的主节点。如果该问题无法自行解决，请联系 Amazon Web Services 支持。
待处理任务过多	主节点处于高负载状态，且具有很多待处理任务。请考虑减少流向集群的流量或扩展您的域。

事务	描述
存储卷受损	一个或多个节点的磁盘卷无法正常运行。此问题通常与其他问题一起发生，例如高错误率或待处理任务过多。如果它是独立发生的且无法自行解决，请联系 Amazon Web Services 支持。
KMS 密钥问题	用于加密域的 KMS 密钥无法访问或丢失。有关更多信息，请参阅 the section called “监控对静态数据进行加密的域”。
快照拍摄正在进行中	域当前正在拍摄快照。在快照拍摄完成后检查升级资格。还要检查您是否可以列出手动快照存储库，在这些存储库中列出快照，并拍摄手动快照。如果 OpenSearch 服务无法检查快照拍摄是否正在进行中，升级可能会失败。
快照拍摄超时或失败	升级前快照拍摄所需的时间过长或失败。检查集群运行状况并重试。如果问题仍存在，请联系 Amazon Web Services 支持。
索引不兼容	一个或多个索引与目标版本不兼容。如果您从较旧版本 OpenSearch 或 Elasticsearch 迁移索引，则可能出现此问题。重建索引并重试。
高磁盘使用率	集群的磁盘使用率高于 90%。删除数据或扩展域，然后重试。
高 JVM 使用率	JVM 内存压力高于 75%。减少流向集群的流量或扩展域，然后重试。
OpenSearch 控制面板别名问题	`.dashboards` 已被配置为一个别名并映射到不兼容的索引，可能是来自早期版本 OpenSearch 控制面板的索引。重建索引，然后重试。
红色控制面板状态	OpenSearch 控制面板状态为红色。尝试在升级完成时使用控制面板。如果红色状态仍然存在，请手动解决该问题，然后重试。
跨集群兼容性	仅当升级之后源域与目标域之间保持跨集群兼容性时，才能升级。在升级过程中，会识别任何不兼容的连接。要继续，请删除远程域或删除不兼容的连接。请注意，如果域上的复制处于活动状态，则在删除连接后无法恢复复制。
其他 OpenSearch 服务问题	OpenSearch 服务本身的问题可能会导致您的域显示没有升级资格。如果上述情况都不适用于您的域且该问题持续超过一天，请联系 Amazon Web Services 支持。

使用快照迁移数据

就地升级是将域升级到更高版本 OpenSearch 或 Elasticsearch 版本的更方便、更快捷、更可靠的方法。如果您需要从 5.1 之前的 Elasticsearch 版本迁移或想要迁移到全新的集群，那么快照是一个很好的选择。

下表显示了如何使用快照将数据迁移到使用不同 OpenSearch 或 Elasticsearch 版本的域。有关制作和还原快照的更多信息，请参阅the section called “创建索引快照”。

之前版本	目标版本	迁移过程

OpenSearch 1.x
x
	OpenSearch
	2.。 x
		1. 请查看 OpenSearch 2.3 的重大更改，了解是否需要对索引或应用程序做出调整。
		2. 创建 1.3 or 2.x 域的手动快照。
		3. 创建 2.x 域，其版本高于原来的 1.3 或 2.x 域。
		4. 将快照从原始域还原到 2.x 域。在操作期间，您可能需要在新名称下还原 `.opensearch` 索引：
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`"indices": "*",`
		`"ignore_unavailable": true,`
		`"rename_pattern": ".opensearch",`
		`"rename_replacement": ".backup-opensearch"`
		`}`
		然后，您可以在新域上为 `.backup-opensearch` 重新建立索引，并为其分配别名 `.opensearch` 。请注意，`_restore`REST 调用不包括 `include_global_state` ，因为 `_restore` 中的默认值为 false。因此，测试域将不包含任何索引模板，也不会获得备份的完整状态。
		5. 如果您不再需要您的原始域，请将其删除。否则，您仍需为该域付费。

之前版本	目标版本	迁移过程

OpenSearch 1.。 x	OpenSearch 1.。 x	1. 创建 1.x 域的手动快照。
		2. 创建 1.x 域，其版本要高于原来的 1.x 域。
		3. 将快照从原始域还原到新的 1.x 域。在操作期间，您可能需要在新名称下还原 `.opensearch` 索引：
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`"indices": "*",`
		`"ignore_unavailable": true,`
		`"rename_pattern": ".opensearch",`
		`"rename_replacement": ".backup-opensearch"`
		`}`
		然后，您可以在新域上为 `.backup-opensearch` 重新建立索引，并为其分配别名 `.opensearch` 。请注意，`_restore`REST 调用不包括 `include_global_state` ，因为 `_restore` 中的默认值为 false。因此，测试域将不包含任何索引模板，也不会获得备份的完整状态。
		4. 如果您不再需要您的原始域，请将其删除。否则，您仍需为该域付费。

之前版本	目标版本	迁移过程

Elasticsearch 6.x 或 7.x	OpenSearch 1.。 x	1. 请查看 OpenSearch 1.0 的重大更改，了解是否需要对索引或应用程序做出调整。
		2. 创建 Elasticsearch 7.x 或 6.x域的手动快照。
		3. 创建一个 OpenSearch 1. x 域名。
		4. 将快照从 Elasticsearch 域还原到域。 OpenSearch 在操作期间，您可能需要在新名称下还原 `.elasticsearch` 索引：
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		然后，您可以在新域上为 `.backup-opensearch` 重新建立索引，并为其分配别名 `.elasticsearch` 。请注意，`_restore`REST 调用不包括 `include_global_state` ，因为 `_restore` 中的默认值为 false。因此，测试域将不包含任何索引模板，也不会获得备份的完整状态。
		5. 如果您不再需要您的原始域，请将其删除。否则，您仍需为该域付费。

之前版本	目标版本	迁移过程

Elasticsearch 6.x	Elasticsearch 7.x	1. 请查看 7.0 的重大更改，了解是否需要对索引或应用程序做出调整。
		2. 创建 6.x 域的手动快照。
		3. 创建 7.x 域。
		4. 将快照从原始域还原到 7.x 域。在操作期间，您可能需要在新名称下还原 `.opensearch` 索引：
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`"indices": "*",`
		`"ignore_unavailable": true,`
		`"rename_pattern": ".elasticsearch",`
		`"rename_replacement": ".backup-elasticsearch"`
		`}`
		然后，您可以在新域上为 `.backup-elasticsearch` 重新建立索引，并为其分配别名 `.elasticsearch` 。请注意，`_restore`REST 调用不包括 `include_global_state` ，因为 `_restore` 中的默认值为 false。因此，测试域将不包含任何索引模板，也不会获得备份的完整状态。
		5. 如果您不再需要您的原始域，请将其删除。否则，您仍需为该域付费。

Elasticsearch 6.x	Elasticsearch 6.8	1. 创建 6.x 域的手动快照。 2. 创建 6.8 域。 3. 将快照从原始域还原到 6.8 域。 4. 如果您不再需要您的原始域，请将其删除。否则，您仍需为该域付费。

之前版本	目标版本	迁移过程
Elasticsearch 5.x	Elasticsearch 6.x	1. 请查看 6.0 的重大更改，了解是否需要对索引或应用程序做出调整。 2. 创建 5.x 域的手动快照。 3. 创建 6.x 域。 4. 将快照从原始域还原到 6.x 域。 5. 如果您不再需要 5.x 域，请将其删除。否则，您仍需为该域付费。
Elasticsearch 5.x	Elasticsearch 5.6	1. 创建 5.x 域的手动快照。 2. 创建 5.6 域。 3. 将快照从原始域还原到 5.6 域。 4. 如果您不再需要您的原始域，请将其删除。否则，您仍需为该域付费。
Elasticsearch 2.3	Elasticsearch 6.x	Elasticsearch 2.3 快照与 6.x 不兼容。要将您的数据直接从 2.3 迁移到 6.x，则必须在新域中手动重新创建您的索引。 或者，您也可以执行本表中从 2.3 迁移到 5.x 的步骤，在新的 5.x 域中执行 `_reindex` 操作以将 2.3 索引转换为 5.x 索引，然后执行从 5.x 迁移到 6.x 的步骤。
Elasticsearch 2.3	Elasticsearch 5.x	1. 请查看 5.0 的重大更改，了解是否需要对索引或应用程序做出调整。 2. 创建 2.3 域的手动快照。 3. 创建 5.x 域。 4. 将快照从 2.3 域还原到 5.x 域。 5. 如果您不再需要 2.3 域，请将其删除。否则，您仍需为该域付费。

之前版本	目标版本	迁移过程
Elasticsearch	Elasticsearch 5.x	1.5 快照与 5.x 不兼容。要将您的数据从 1.5 迁移到 5.x，则必须在新域中手动重新创建您的索引。 **Important** 1.x 与 2.3 兼容，但 OpenSearch 服务 2.3 域不支持此`_reindex`操作。由于您无法为它们重新编制索引，因此 1.5 域中发出的索引仍无法从 2.3 快照还原到 5.x 域。
Elasticsearch	Elasticsearch 2.3	1. 使用迁移插件了解是否可以直接升级到 2.3 版。您可能需要在迁移前对数据进行更改。 a. 在 Web 浏览器中，打开 `http://domain-endpoint /_plugin/migration/` 。 b. 选择立即运行检查。 c. 检查结果，如果需要，按照说明更改您的数据。 2. 创建 1.5 域的手动快照。 3. 创建 2.3 域。 4. 将快照从 1.5 域还原到 2.3 域。 5. 如果您不再需要 1.5 域，请将其删除。否则，您仍需为该域付费。

为 Amazon S OpenSearch ervice 服务创建自定义终端节点

为您的 Amazon S OpenSearch ervice 域创建自定义终端节点可让您更轻松地参考您的 OpenSearch and OpenSearch 仪表板 URLs。您可以包括贵公司的品牌，也可以使用比标准 easier-to-remember终端节点更短。

如果您需要切换到新域，只需更新 DNS 以指向新 URL 并继续使用与之前相同的终端节点即可。

您可以通过以下方式保护自定义终端节点： Amazon Certificate Manager (ACM) 或导入您自己的一个。

新域的自定义终端节点

您可以使用服务控制台、 Amazon CLI或配置 API 为新 OpenSearch OpenSearch 服务阈启用自定义终端节点。

要自定义终端节点（控制台）

1. 从 OpenSearch 服务控制台中，选择 Create domain（创建域），然后提供该域的名称。
2. 在自定义端点下方，选择启用定制端点。
3. 适用于自定义主机名中，输入首选的自定义终端节点主机名。主机名应为完全限定域名 (FQDN)，例如 www.yourdomain.com 或者例如 example.yourdomain.com。

   > **Note**
   >
   > 如果您没有通配符证书，则可能需要为自定义端点的子域获取新证书。

4. 对于 Amazon certificate，选择要用于您的域的 SSL 证书。如果没有可用的证书，则可以将证书导入 ACM 或使用 ACM 预配证书。有关更多信息，请参阅 Amazon Certificate Manager 用户指南中的发布和管理证书。

   > **Note**
   >
   > 证书必须具有自定义终端节点名称，并且与 OpenSearch 服务域位于同一帐户中。证书状态应该为“ISSUED”。

   - 请按照接下来的步骤创建您的域，然后选择 Create（创建）。
   - 完成处理后，选择域以查看您的自定义终端节点。

   要使用 CLI 或配置 API，请使用 `CreateDomain` 和 `UpdateDomainConfig` 运算符。有关更多信息，请参阅Amazon CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

现有域的自定义终端节点

要将自定义终端节点添加到现有 OpenSearch 服务域，请选择 Edit（编辑）并执行上述步骤 2-4。

CNAME 映射

为 OpenSearch 服务域启用自定义终端节点后，您可以在 Amazon Route 53（或您偏好的 DNS 服务提供商）中创建 CNAME 映射。创建 CNAME 映射将使您能够将流量路由到自定义端点及其子域。如果没有此映射，您无法将流量路由到自定义端点。有关在 Route 53 中创建此映射的步骤，请参阅为新域配置 DNS 路由和为子域创建新的托管区。对于其他提供商，请参考其文档。

创建将自定义端点指向自动生成的域端点的 CNAME 记录。如果您的域是双堆栈，则可以将 CNAME 记录指向两个服务生成的端点中的任何一个。自定义端点的双栈功能取决于 CNAME 记录指向的服务生成端点。自定义端点主机名是 CNAME 记录的名称，域端点主机名是 CNAME 记录的值。

如果您将使用用于 OpenSearch仪表板的 SAML 身份验证，则必须使用新的 SSO URL 更新您的 IdP。

您可以使用 Amazon Route 53 创建别名记录类型，以将域的自定义端点指向某个双栈搜索端点。要创建别名记录类型，必须将域配置为使用双栈 IP 地址类型。您可以使用 Route 53 API 来执行此操作。

要使用 Route 53 API 创建别名记录类型，请指定域的别名目标。您可以在自定义终端节点部分的托管区（双栈）字段中找到域的别名目标，也可以使用 API 并复制 `DescribeDomain` API 并复制的值`DomainEndpointV2HostedZoneId`。 OpenSearch

自动调整 Amazon Servic OpenSearch e

Amazon Service OpenSearch 服务中的自动调整使用 OpenSearch 集群中的性能和使用情况指标建议与内存相关的配置更改，包括节点上的队列和缓存大小以及 Java 虚拟机（JVM）设置。这些可选更改可提高集群速度和稳定性。

一些更改会立即部署，另一些更改则安排在域的非高峰时段部署。您可以随时恢复到默认 Serv OpenSearch ice 设置。随着自动调整收集和分析域的性能指标，您可以在通知页上面的 Ser OpenSearch vice 控制台中查看其建议。

自动调整可在运行任何 Elasticsearch 6.7 或更高 OpenSearch版本的域 Amazon Web Services 区域 上的商业中在受支持的实例类型上使用。

更改类型

自动调整有两大类更改：

- 集群运行时应用的无中断更改。
- 需要蓝绿部署的更改将在域的非高峰时段应用。

根据您域的性能指标，“自动调整”可以建议对以下设置进行调整：

更改类型	类别	描述
JVM 堆大小	蓝/绿	默认情况下，Serv OpenSearch ice 将实例的 RAM 的 50% 用于 JVM 堆，最大堆大小为 32 GiB。 增加此百分比可以 OpenSearch 获得更多的内存，但为操作系统和其他进程留下的内存较少。较大的值可以减少垃圾回收暂停的数量，但会增加这些暂停的长度。
JVM 年轻一代设置	蓝/绿	JVM“年轻一代”设置会影响次要垃圾收集的频率。更频繁的次要收集可以减少主要收集和暂停的次数。
队列大小	无中断	默认情况下，搜索队列大小为 `1000`，写入队列大小为 `10000`。如果有其他堆可用于处理请求，则自动调整会自动缩放搜索和写入队列。
缓存大小	无中断	字段缓存监控堆上的数据结构，因此监控缓存的使用非常重要。自动调整可调整字段数据高速缓存大小，以避免内存不足和断路器问题。 这些区域有：分片请求缓存在节点级别进行管理，并且默认的最大大小为堆的 1%。自动调整可扩展分片请求高速缓存大小，以接受比配置的集群能够处理的更多搜索和索引请求。
请求大小	无中断	默认情况下，当正在进行的请求的合计大小超过 JVM 总数的 10% 时（`t2`实例类型为 2%，对于 1%`t3.small`），将会 OpenSearch 限制所有新`_bulk`请求`_search`和请求，直到现有请求完成。 自动调整将会根据系统上当前占用的 JVM 量自动调整此阈值，通常在 5-15% 之间。例如，如果 JVM 内存压力大，则自动调整可能会将阈值降至 5%，此时，您可能会看到更多拒绝，直到集群稳定和阈值提高。

启用或禁用自动调整

OpenSearch 默认情况下，Service 在新域中启用自动调整。要在现有域上启用或禁用自动调整，我们建议使用控制台，这简化了过程。启用自动调整不会导致蓝/绿部署。

您目前无法使用 Amazon CloudFormation启用或禁用自动调整。

控制台

对现有域启用自动调整

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 在导航窗格中的域下，选择域名以打开集群配置。
3. 如果尚未启用自动调整，请选择启用。
4. 或者，选择非高峰时段，安排在域配置的非高峰时段执行需要蓝绿部署的优化。有关更多信息，请参阅 the section called “计划自动调整增强功能”。
5. 选择保存更改。

CLI

要使用启用自动调整 Amazon CLI，UpdateDomainConfig请发送请求：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options DesiredState=ENABLED
```

计划自动调整增强功能

2023 年 2 月 16 日之前，自动调整使用维护时段计划需要蓝绿部署的更改。现在已弃用维护时段，取而代之的是非高峰时段，即每天域流量通常较低的 10 小时时段。您可以修改非高峰时段的默认开始时间，但不能修改长度。

在 2023 年 2 月 16 日推出非高峰时段之前启用自动调整维护时段的所有域均可继续沿用旧版无间断维护时段。但是，我们建议您迁移现有域，改用非高峰时段进行域维护。有关说明，请参阅the section called “从自动调整维护窗口迁移”。

控制台

计划在非高峰时段执行自动调整操作

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 在导航窗格中的域下，选择域名以打开集群配置。
3. 转到自动调整选项卡，选择编辑。
4. 如果尚未启用自动调整，请选择启用。
5. 在计划在非高峰时段执行优化下，选择非高峰时段。

6. 选择保存更改。

CLI

要配置域计划在配置的非高峰时段执行自动调整操作，请在请求`UseOffPeakWindow`中UpdateDomainConfig纳入：

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=null
```

监控自动调整更改

您可以在 Amazon CloudWatch中监控自动调整统计数据。有关指标的完整列表，请参阅the section called “自动调整指标”。

OpenSearch Service 将自动调整事件发送到 Amazon EventBridge。您可以使用配置 EventBridge 在收到事件时发送电子邮件或执行特定操作的规则。要查看发送到的每个自动调整事件的格式 EventBridge，请参阅the section called “自动调整事件”。

标记 Amazon S OpenSearch ervice 域

标签允许您将任意信息分配给 Amazon S OpenSearch ervice 域，以便您可以对该信息进行分类和筛选。标签是由您定义的键-值对，与 OpenSearch 服务域关联。您可以将具有类似标签的资源的费用分组在一起，使用标签来跟踪成本。 Amazon 不会对您的标签应用任何语义意义。标签严格按字符串进行解释。所有标签均包含以下元素：

标签元素	描述	必填
标签键	标签键是标签的名称。密钥必须是附加到的 OpenSearch 服务域的唯一密钥。有关对标签键和值的基本限制的列表，请参阅用户定义的标签限制。	是
标签值	标签值则是标签字符串值。标签值可为 `null`，且在标签集中不必具有唯一性。例如，在的标签集中，可以存在键-值对。project/Trinity and cost-center/Trinity有关对标签键和值的基本限制的列表，请参阅用户定义的标签限制。	否

每个 OpenSearch 服务域都有一个标签集，其中包含分配给该 OpenSearch 服务域的所有标签。Amazon 不会自动为 OpenSearch 服务域分配任何标签。标签集可以包含 0 到 50 个标签。如果使用与现有标记相同的键向域添加标记，则新值将覆盖旧值。

标签示例

您可以使用密钥定义类别，而值作为该类别中的项目。例如，可定义标签键`project`、标签值为`Salix`，指示 OpenSearch 服务域已分配给 Salix 项目。您也可使用标签通过或等键指定 OpenSearch 服务域用于测试`environment=test`或生产`environment=production`。我们建议使用一组具有一致性的标签键，以使跟踪与 OpenSearch 服务域关联的元数据变得更轻松。

您也可以使用标签来组织您的 Amazon 账单，使其反映您的成本结构。要执行此操作，请注册以获取包含标签键值的 Amazon Web Services 账户 账单。然后，如需查看组合资源的成本，请按有同样标签键值的资源组织您的账单信息。例如，您可以使用键-值对标记几个 OpenSearch 服务域，然后组织账单信息，查看在数个服务中的每个域的总成本。有关更多信息，请参阅 https://docs.amazonaws.cn/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html 账单和成本管理文档中的Amazon 使用成本分配标签。

> **Note**
>
> 对标签进行缓存以用于授权。因此，可能先用几分钟添加和更新 S OpenSearch ervice 域上的标签，然后标签才可用。

标记域（控制台）

控制台是标记域的最简单方法。

创建标签（控制台）

1. 转到 https://aws.amazon.com，然后选择 Sign In to the Console (登录控制台)。
2. 在 “分析” 下，选择 “亚马逊 OpenSearch 服务”。
3. 选择您要将标签添加到的域，然后转到 Tags（标签）选项卡。
4. 选择 Manage（管理）和 Add new tag（添加新标签）。
5. 输入一个标签键和可选的值。
6. 选择保存。

要删除标签，请按照相同步骤操作并在 Manage tags（管理标签）页面中选择 Remove（删除）。

有关使用控制台处理标签的更多信息，请参阅《Amazon 管理控制台入门指南》中的标签编辑器。

标记域（Amazon CLI）

您可以使用 Amazon CLI 带--add-tags命令的创建资源标签。

语法

```
add-tags --arn=<domain_arn> --tag-list Key=<key>,Value=<value>
```

参数	描述
`--arn`	附加标签的 OpenSearch 服务域的 Amazon Resource Name。
`--tag-list`	采用以下格式设置空格分隔的键值对：`Key=<key>,Value=<value>`

示例

以下示例为 logs 域创建两个标签：

```
aws opensearch add-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-list
 Key=service,Value=OpenSearch Key=instances,Value=m3.2xlarge
```

您也可以使用--remove-tags命令从 OpenSearch 服务域中删除标签。

语法

```
remove-tags --arn=<domain_arn> --tag-keys Key=<key>,Value=<value>
```

参数	描述
`--arn`	附加标签的 OpenSearch 服务域的 Amazon Resource Name (ARN)。
`--tag-keys`	设置您要从服务域中移除的空格分隔的键-值对。 OpenSearch

示例

以下示例从之前示例中创建的 logs 域中删除两个标签：

```
aws opensearch remove-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-
keys service instances
```

您也可以使用以下--list-tags命令查看 OpenSearch 服务域的现有标签：

语法

```
list-tags --arn=<domain_arn>
```

参数	描述
`--arn`	附加标签的 OpenSearch 服务域的 Amazon Resource Name (ARN)。

示例

以下示例列出了 logs 域的所有资源标签：

```
aws opensearch list-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs
```

标记域（Amazon SDKs）

Amazon SDKs（Android 和 iOS 除外 SDKs）支持《亚马逊 OpenSearch 服务 API 参考》中定义的所有操作，包括`AddTags``ListTags`、和`RemoveTags`操作。有关安装和使用的更多信息 Amazon SDKs，请参阅Amazon 软件开发工具包。

Python

此示例使用适用于 Python（Boto）的 Amazon SDK 中的OpenSearchService低级别 Python 客户端向域添加标签、列出附加到域的标签以及从域中删除标签。必须提供 `DOMAIN_ARN`、`TAG_KEY` 和 `TAG_VALUE` 的值。

```
import boto3
from botocore.config import Config  # import configuration

DOMAIN_ARN = ''  # ARN for the domain. i.e "arn:aws:es:us-east-1:123456789012:domain/
my-domain
TAG_KEY = ''  # The name of the tag key. i.e 'Smileyface'
TAG_VALUE = ''  # The value assigned to the tag. i.e 'Practicetag'
```

```
# defines the configurations parameters such as region

my_config = Config(region_name='us-east-1')
client = boto3.client('opensearch', config=my_config)


# defines the client variable

def addTags():
    """Adds tags to the domain"""

    response = client.add_tags(ARN=DOMAIN_ARN,
                               TagList=[{'Key': TAG_KEY,
                                         'Value': TAG_VALUE}])

    print(response)


def listTags():
    """List tags that have been added to the domain"""

    response = client.list_tags(ARN=DOMAIN_ARN)
    print(response)


def removeTags():
    """Remove tags that have been added to the domain"""

    response = client.remove_tags(ARN=DOMAIN_ARN, TagKeys=[TAG_KEY])

    print('Tag removed')
    return response
```

在 Amazon EFS KMS 密 OpenSearch 钥执行管理操作

Amazon OpenSearch Ser OpenSearch vice 提供了多种管理选项，如果您需要解决有关域的问题，可以使用这些选项进行精细控制。这些选项包括重启数据节点 OpenSearch

OpenSearch OpenSearch Service 会监控节点运行状况参数，并在出现异常时采取纠正措施以保持域的稳定。凭借重启节点上的 OpenSearch OpenSearch 流程和重启节点的管理选项，您可以控制其中一些缓解措施。

您可以使用 Amazon Web Services Management Console Amazon CLI、或 Amazon SDK 来执行这些操作。以下各节介绍如何使用控制台执行这些操作。

在 Amazon OpenSearch 服务中的节点上重新启动该 OpenSearch 进程

要重启节点上的 OpenSearch Kibana 流程

1. 导航到 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择 域。选择要使用的域的名称。
3. 打开域的详细信息页面后，导航至实例运行状况选项卡。
4. 在数据节点下，选择要重启流程的节点旁边的按钮。
5. 选择 “操作” 下拉列表并选择 “重启 OpenSearch /Elasticsearch 进程”。
6. 在模态上选择确认。
7. 要查看您启动的操作的状态，请选择节点的名称。打开节点详细信息页面后，选择节点名称下方的事件选项卡，查看与该节点关联的事件列表。

在 Amazon 服务中重启数据节点 OpenSearch

要重启数据节点

1. 导航到 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择 域。选择要使用的域的名称。
3. 打开域的详细信息页面后，导航至实例运行状况选项卡。
4. 在数据节点下，选择要重启流程的节点旁边的按钮。
5. 选择操作下拉列表，然后选择重启节点。
6. 在模态上选择确认。
7. 要查看您启动的操作的状态，请选择节点的名称。打开节点详细信息页面后，选择节点名称下方的事件选项卡，查看与该节点关联的事件列表。

在 Amazon OpenSearch 服务中的节点上重新启动 OpenSearch 控制面板流程

您可以重新启动 OpenSearch 仪表板（以前是 Kibana）进程，以从界面冻结、加载失败或可视化效果无响应等问题中恢复。重启 OpenSearch 仪表板的选项仅适用于正在积极运行仪表板进程的节点。

在大多数 OpenSearch 服务域中，此过程在专用的协调器节点上运行，而不是在数据节点上运行。因此，当您在控制台中打开 “操作” 下拉列表时，该选项通常仅适用于协调器节点。有关更多信息，请参阅 the section called “专用协调器节点”。

这种行为取决于您的域名的配置方式。

- 带有专用协调器节点的域-仪表板进程仅在这些节点上运行，只有它们显示重启选项。
- 没有专用协调器节点的域 — 在更简单或更旧的设置中，仪表板可能会改为在数据节点上运行，并且重启选项会出现在那里。
- 主节点 — 这些节点仅用于管理群集状态和选举。他们不运行仪表板，也从不显示重启选项。

要确定哪个节点正在运行 Dashboards 进程，请导航到您的域的集群配置部分并查看节点角色。重启选项仅适用于托管仪表板流程的节点。

要重启节点上的 Kibana 流程

1. 导航到 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择 域。选择要使用的域的名称。
3. 打开域的详细信息页面后，导航至实例运行状况选项卡。
4. 在数据节点下，选择要重启流程的节点旁边的按钮。
5. 选择操作下拉列表，然后选择重启控制面板/Kibana 流程。
6. 在模态上选择确认。
7. 要查看您启动的操作的状态，请选择节点的名称。打开节点详细信息页面后，选择节点名称下方的事件选项卡，查看与该节点关联的事件列表。

限制

管理选项具有以下限制：

- Elasticsearch 7.x 及更高版本支持管理选项。
- 管理选项不支持启用了带待机功能的多可用区的域。
- 这些选项包括重启数据节点 OpenSearch
- 在具有两个或更多数据节点的域上，支持控制面板和 Kibana 流程。
- 要在节点上重启 OpenSearch OpenSearch 流程或重启节点，该域不得处于红色状态，并且所有索引都必须配置副本，并且所有索引都必须配置副本。

使用 Amazon OpenSearch 服务直接查询

使用亚马逊 OpenSearch 服务直接查询来分析亚马逊 CloudWatch 日志、Amazon S3 和亚马逊安全湖中的数据，而无需建立摄取管道。这种零 ETL 集成允许您使用 OpenSearch SQL 或 PPL 在原地查询数据，并在 Discover 中对其进行探索。

首先，请在 OpenSearch 服务控制台中配置您的数据源。对于 Amazon S3，使用域连接并在查询工作台中使用 SQL 创建表。 CloudWatch 日志和 Security Lake 使用预先配置的源和 Amazon Glue Data Catalog 表。

直接查询定价

当您使用 OpenSearch 服务直接查询时，您需要分别支付 OpenSearch 服务费用以及用于在 Amazon S3、Amazon L CloudWatch ogs 和 Amazon Security Lake 上处理和存储数据的资源费用。当你运行直接查询时，你会看到每小时 OpenSearch 计算单位 (OCUs) 的费用，在账单上列为 DirectQuery OCU 使用类型。

直接查询有两种类型：交互式查询和索引视图查询。

- 交互式查询用于填充数据选择器并对 S3、L CloudWatch ogs 或 Security Lake 中的数据进行分析。

  对于 Amazon S3 直接查询，当您从 Discover 中运行新查询时， OpenSearch 服务会启动一个持续至少三分钟的新会话。 OpenSearch 服务使此会话保持活动状态，以确保后续查询快速运行。

  对于 CloudWatch 日志和 Security Lake 查询，Ser OpenSearch vice 使用单独的预热任务处理每个查询，而无需延长会话。
- 索引视图查询使用计算来维护服务中的索引视图。 OpenSearch 这些查询通常需要更长的时间，因为它们将不同数量的数据摄入到命名索引中。通过为数据编制索引，您可以让 future 的交互式查询更快地运行，或者解锁高级分析功能，例如需要索引才能引用的仪表板或警报。

  对于 Amazon S3 数据源，索引数据根据购买的实例类型存储在域中。对于与 L CloudWatch ogs 和 Security Lake 连接的数据源，索引数据存储在 OpenSearch 无服务器集合中，您需要为已编制索引的数据 (IndexingOCU)、搜索的数据 (SearchOCU) 和以 GB 存储的数据付费。

有关更多信息，请参阅 Amazon OpenSearch 服务定价中的 “直接查询” 和 “无服务器” 部分。

直接查询限制

一般限制

以下限制适用于 OpenSearch 服务直接查询。

- 某些数据类型不支持。支持的数据类型仅限于 Parquet、CSV 和 JSON。
- 如果数据结构随着时间的推移而发生变化，则需要更新索引视图或 out-of-the-box集成，以应对数据结构的变化。
- Amazon CloudFormation 尚不支持模板。
- OpenSearch 与使用直接查询相比，SQ OpenSearch L 和 PPL 语句在使用 OpenSearch 索引时有不同的限制。直接查询支持子查询和查找等 JOINs高级命令，而 OpenSearch 索引上对这些命令的支持有限或根本不存在。有关更多信息，请参阅 the section called “支持的 SQL 和 PPL 命令”。

亚马逊 S3 的限制

如果您直接在 Amazon S3 中查询数据，则适用以下额外限制：

- S3 的直接查询仅适用于运行 2.13 或更高 OpenSearch版本的 OpenSearch 服务域，并且需要 Amazon Glue Data Catalog访问权限。必须在 Quer OpenSearch y Workbench 中使用 SQL 重新创建现有 Amazon Glue Data Catalog 表。
- 直接查询 S3 需要您在 Amazon S3 上指定检查点存储桶。此存储桶维护您的索引视图的状态，包括上次刷新时间和最近提取的数据。
- 您的 OpenSearch 域名和 Amazon Glue Data Catalog 必须相同 Amazon Web Services 账户。您的 S3 存储桶可以位于不同的账户中（需要将条件添加到您的 IAM 策略中），但必须与您的域位于同一个 Amazon Web Services 区域 账户中。
- OpenSearch 使用 S3 的服务直接查询仅支持从 Query Workbench 生成的 Spark 表。Spark 流式传输不支持在 Amazon Glue Data Catalog 或 Athena 中生成的表，这是维护索引视图所必需的。
- OpenSearch 实例类型的网络负载限制为 10 MiB 或 100 MiB，具体取决于您选择的特定实例类型。

Amazon CloudWatch 日志的限制

如果您直接在 CloudWatch 日志中查询数据，则适用以下额外限制：

- 与 CloudWatch 日志的直接查询集成仅适用于 OpenSearch 服务集合和 OpenSearch 用户界面。
- OpenSearch 无服务器集合的网络有效载荷限制为 100 MiB。

- CloudWatch 日志支持 VPC Flow CloudTrail，以及从控制台安装的 Amazon WAF 仪表板集成。

Amazon 安全湖的限制

如果您直接在 Security Lake 中查询数据，则适用以下额外限制：

- 与 Security Lake 的直接查询集成仅适用于 OpenSearch 服务集合和 OpenSearch 用户界面。
- OpenSearch 无服务器集合的网络有效载荷限制为 100 MiB。
- Security Lake 的表格管理在 Lake Formation 中执行。
- Security Lake 仅支持将物化视图作为索引视图。不支持覆盖索引。

关于在 Amazon OpenSearch 服务中使用直接查询的建议

本页提供了使用亚马逊 OpenSearch 服务直接查询来分析来自 CloudWatch 日志、Amazon S3 和亚马逊安全湖的数据的建议。这些最佳实践可帮助您优化性能并确保高效查询，而无需摄取或复制数据。

主题

- 一般建议
- 亚马逊 S3 建议
- CloudWatch 日志推荐
- 安全湖建议

一般建议

在使用直接查询功能时，建议执行以下操作：

- 使用该`COALESCE SQL`函数来处理缺失的列并确保返回结果。

- 使用查询限制来确保不会提取太多数据。
- 如果您计划多次分析同一个数据集，请创建一个索引视图，以便在完成分析后将数据完全摄取 OpenSearch 和索引，然后将其丢弃。
- 当不再需要加速任务和索引时，将其删除。
- 不支持包含相同但仅在大小写上有所不同（例如`field1`和`FIELD1`）的字段名称的查询。

  例如，不支持以下查询：

```
Select AWSAccountId, AwsAccountId from LogGroup
Select a.@LogStream, b.@logStream from Table A INNER Join Table B ona.id = b.id
```

但是，由于两个日志组中的字段名 (@logStream) 相同，因此支持以下查询：

```
Select a.@logStream, b.@logStream from Table A INNER Join Table B on a.id = b.id
```

- 函数和表达式必须对字段名进行操作，并且是`SELECT`语句的一部分，该语`FROM`句中指定了日志组。

  例如，不支持以下查询：

  ```
  SELECT cos(10) FROM LogGroup
  ```

  支持此查询：

  ```
  SELECT cos(field1) FROM LogGroup
  ```

亚马逊 S3 建议

如果您使用亚马逊 OpenSearch 服务在 Amazon S3 中直接查询数据，我们还建议您采取以下措施：

- 使用年、月、日、小时的分区格式将数据摄取到 Amazon S3 中，以加快查询速度。
- 构建跳过索引时，请对基数较高的字段使用 Bloom 过滤器，对值范围较大的字段使用 min/max 索引。对于高基数字段，可以考虑使用基于值的方法来提高查询效率。
- 使用索引状态管理来维护实例化视图和覆盖索引的存储。

CloudWatch 日志推荐

如果您使用 Amazon Ser OpenSearch vice 在 CloudWatch 日志中引导查询数据，我们还建议您采取以下措施：

- 在一个查询中搜索多个日志组时，请使用适当的语法。有关更多信息，请参阅 the section called “多日志组函数”。
- 使用 SQL 或 PPL 命令时，请将某些字段用反引号括起来，以便成功查询它们。带有特殊字符（非字母和非数字）的字段需要使用反引号。例如，用反引号`@message`括`Operation.Export`,起来、和`Test::Field`。您无需用反引号将纯字母名称的列括起来。

包含简单字段的查询示例：

```
SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
LIMIT 1000;
```

附加了反引号的类似查询：

```
SELECT `@SessionToken`, `@Operation`, `@StartTime` FROM `LogGroup-A`
LIMIT 1000;
```

安全湖建议

如果您使用 Amazon Ser OpenSearch vice 在安全湖中直接查询数据，我们还建议您采取以下措施：

- 检查您的 Security Lake 状态，确保其平稳运行，没有任何问题。有关详细的故障排除步骤，请参阅 Amazon Security Lake 用户指南中的数据湖状态故障排除。
- 验证您的查询访问权限：
  - 如果您使用与 Security Lake 委托管理员账户不同的账户查询 Security Lake，请在 S ecurity Lake 中设置具有查询权限的订阅者。
  - 如果您使用同一个账户查询 Security Lake，请查看 Security Lake 中是否有任何关于向其注册托管 S3 存储桶的 LakeFormation消息。
- 浏览查询模板和预建仪表板，快速开始分析。
- 熟悉开放式网络安全架构框架 (OCSF) 和 Security Lake：
  - 查看 OCSF GitHub 存储库中 Amazon 源的架构映射示例
  - 通过访问Amazon 源版本 2 的 Security Lake 查询 (OCSF 1.1.0)，了解如何有效地查询 Security L ake
  - 使用分区:`accountid`、和`region`，提高查询性能 `time_dt`
- 熟悉 SQL 语法，Security Lake 支持使用该语法进行查询。有关更多信息，请参阅 the section called “支持的 SQL 命令”。

直接查询配额

您的账户具有以下与 OpenSearch 服务直接查询相关的配额。

亚马逊 S3 的配额

每次您启动对 Amazon S3 数据源的查询时， OpenSearch 服务都会打开一个会话并使其保持至少三分钟的活动状态。这通过消除后续查询中的会话启动时间来减少查询延迟。

描述	最大值	可以覆盖
每个域的连接数	10	是
每个域的数据来源数	20	是
每个域的索引数	5	是
每个数据来源的并行会话数	10	是
每次查询的最大 OCU	60	是
最大查询执行时间（分钟）	30	是
OCUs 每次加速度的最大值	20	是
最大临时存储	20	是

CloudWatch 日志配额

> **Note**
>
> 如果您想使用 L CloudWatch ogs Insights 进行直接查询，请务必参考the section called “CloudWatch 日志见解”。

描述	值	软限制？	备注
跨直接查询的账户级 TPS 限制 APIs	3 TPS	是	
数据源的最大数量	20	是	限额为每个 Amazon Web Services 账户。

描述	值	软限制？	备注
自动刷新索引或实例化视图的最大值	30	是	限制是按数据源计算的。
最大并发查询数	15	是	限制适用于处于待处理或运行状态的查询。 包括交互式查询（例如，数据检索命令，如`SELECT`）和索引查询（例如，`CREATE`/`ALTER`/之类的操作`DROP`）。
每次查询的最大并发 OCU	512	是	OpenSearch 计算单位 (OCU)。限制基于 15 个执行器和 1 个驱动程序，每个执行器都有 16 个 vCPU 和 32 GB 内存。表示并发处理能力。
最大查询执行时间（以分钟为单位）	60	否	限制适用于 Logs Insights 中的 OpenSearch PPL/SQL 查询。 CloudWatch
清除陈旧查询的期限 IDs	90 天	是	在此时间段之后， OpenSearch 服务会清除较旧条目的查询元数据。例如，对于超过 90 天的查询，调用 GetDirectQuery 或 GetDirectQueryResult 失败。

安全湖配额

描述	值	软限制？	备注
跨直接查询的账户级 TPS 限制 APIs	3 TPS	是	
数据源的最大数量	20	是	限额为每个 Amazon Web Services 账户。
自动刷新索引或实例化视图的最大值	30	是	限制适用于每个数据源。 仅包括自动刷新设置为 true 的索引和实例化视图 (MVs)。
最大并发查询数	30	是	限制适用于处于待处理或运行状态的查询。

描述	值	软限制？	备注
			包括交互式查询（例如，数据检索命令，如SELECT）和索引查询（例如，CREATE/ALTER/之类的操作DROP）。
每次查询的最大并发 OCU	512	是	OpenSearch 计算单位 (OCU)。限制基于 15 个执行器和 1 个驱动程序，每个执行器都有 16 个 vCPU 和 32 GB 内存。表示并发处理能力。
最大查询执行时间（以分钟为单位）	30	否	仅适用于交互式查询（例如，数据检索命令，例如SELECT）。对于REFRESH查询，限制为 6 小时。
清除陈旧查询的期限 IDs	90 天	是	在此时间段之后， OpenSearch 服务会清除较旧条目的查询元数据。例如，对于超过 90 天的查询，调用 GetDirectQuery或 GetDirectQueryResult 失败。

支持 Amazon Web Services 区域

Amazon S3、 CloudWatch 日志和安全湖中的 OpenSearch 服务直接查询支持以下内容 Amazon Web Services 区域 ：

Amazon Web Services 区域 适用于亚马逊 S3

- 亚太地区（香港）
- 亚太地区（孟买）
- 亚太地区（首尔）
- 亚太地区（新加坡）
- 亚太地区（悉尼）
- 亚太地区（东京）
- 加拿大（中部）
- 欧洲地区（法兰克福）
- 欧洲地区（爱尔兰）
- 欧洲地区（斯德哥尔摩）

- 美国东部（弗吉尼亚州北部）
- 美国东部（俄亥俄州）
- 美国西部（俄勒冈州）

Amazon Web Services 区域 可用于 CloudWatch 日志

- 亚太地区（孟买）
- 亚太地区（香港）
- 亚太地区（新加坡）
- 亚太地区（悉尼）
- 亚太地区（东京）
- 加拿大（中部）
- 欧洲地区（法兰克福）
- 欧洲地区（爱尔兰）
- 欧洲地区（斯德哥尔摩）
- 美国东部（弗吉尼亚州北部）
- 美国东部（俄亥俄州）
- 美国西部（俄勒冈州）
- 欧洲地区（巴黎）
- 欧洲地区（伦敦）
- 南美洲（圣保罗）

Amazon Web Services 区域 适用于安全湖

- 亚太地区（孟买）
- 亚太地区（新加坡）
- 亚太地区（悉尼）
- 亚太地区（东京）
- 加拿大（中部）
- 欧洲地区（法兰克福）
- 欧洲地区（爱尔兰）

- 欧洲地区（斯德哥尔摩）
- 美国东部（弗吉尼亚州北部）
- 美国东部（俄亥俄州）
- 美国西部（俄勒冈州）
- 欧洲地区（巴黎）
- 欧洲地区（伦敦）
- 南美洲（圣保罗）

直接查询 OpenSearch 服务中的 Amazon S3 数据

本节将引导您完成在 Amazon Ser OpenSearch vice 中创建和配置数据源集成的过程，使您能够高效地查询和分析存储在 Amazon S3 中的数据。

在接下来的页面中，您将学习如何设置 Amazon S3 直接查询数据源、浏览必要的先决条件，以及如何使用和 OpenSearch 服务 API 按照 step-by-step步骤操作。 Amazon Web Services Management Console 它还涵盖了重要的后续步骤，包括映射 Amazon Glue Data Catalog 角色和在 OpenSearch 仪表板中配置访问控制。

主题

- 在 OpenSearch 服务中创建 Amazon S3 数据源集成
- 在 OpenSearch 控制面板中配置和查询 S3 数据源

在 OpenSearch 服务中创建 Amazon S3 数据源集成

您可以通过 Amazon Web Services Management Console 或 API 为 OpenSearch 服务创建新的 Amazon S3 直接查询数据源。每个新数据源都使用 Amazon Glue Data Catalog 来管理代表 Amazon S3 存储桶的表。

主题

- 先决条件
- 过程
- 后续步骤
- 映射 Amazon Glue Data Catalog 角色
- 其他资源

先决条件

在开始之前，请确保您已经阅读了以下文档：

- the section called “亚马逊 S3 的限制”
- the section called “亚马逊 S3 建议”
- the section called “亚马逊 S3 的配额”

在创建数据源之前，您必须拥有以下资源 Amazon Web Services 账户：

- 版本为 2.13 或更高版本的 OpenSearch 域名。这是设置直接查询集成的基础。有关相关设置的说明，请参阅the section called “ 创建 OpenSearch 服务域”。
- 一个或多个 S3 存储桶。您需要指定包含要查询的数据的存储桶，以及用于存储查询检查点的存储桶。有关创建 S3 存储桶的说明，请参阅 Amazon S3 用户指南中的创建存储桶。
- （可选）一个或多个 Amazon Glue 表。在 Amazon S3 上查询数据需要您设置表 Amazon Glue Data Catalog 以指向 S3 数据。必须使用 OpenSearch Query Workbench 创建表。现有的 Hive 表不兼容。

  如果这是您首次设置 Amazon S3 数据源，则必须创建一个管理数据源来配置所有 Amazon Glue Data Catalog 表。为此，您可以安装 OpenSearch out-of-the-box集成或使用 OpenSearch Query Workbench 为高级用例创建自定义 SQL 表。有关为 VPC CloudTrail、和 Amazon WAF 日志创建表的示例，请参阅 VPC CloudTrail、和 GitHub Amazon WAF的文档。创建表后，您可以创建新的 Amazon S3 数据源并限制对有限表的访问权限。
- （可选）手动创建的 IAM 角色。您可以使用此角色来管理对数据源的访问权限。或者，您可以让 S OpenSearch ervice 自动为您创建具有所需权限的角色。如果您选择使用手动创建的 IAM 角色，请按照中的指导进行操作the section called “手动创建的 IAM 角色所需的权限”。

过程

您可以使用 Amazon Web Services Management Console 或 OpenSearch 服务 API 在网域上设置直接查询数据源。

要使用设置数据源 Amazon Web Services Management Console

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择 域。
3. 选择要为其设置新数据来源的域。随即打开域详细信息页面。

4. 选择一般域详细信息下方的连接选项卡，然后找到直接查询部分。
5. 选择 “配置数据源”。
6. 输入新数据源的名称和可选描述。
7. 选择带有 Amazon S3 Amazon Glue Data Catalog。
8. 在 IAM 权限访问设置下，选择如何管理访问权限。
   a. 如果要自动为此数据源创建角色，请执行以下步骤：
      i. 选择 “创建新角色”。
      ii. 输入 IAM 角色的名称。
      iii. 选择一个或多个包含您要查询的数据的 S3 存储桶。
      iv. 选择要在其中存储查询检查点的检查点 S3 存储桶。
      v. 选择一个或多个 Amazon Glue 数据库或表来定义可以查询哪些数据。如果尚未创建表，请提供对默认数据库的访问权限。
   b. 如果您想使用自己管理的现有角色，请按照以下步骤操作：
      i. 选择使用现有角色。
      ii. 从下拉菜单中选择现有角色。

> **Note**
> 使用自己的角色时，必须通过从 IAM 控制台附加必需的策略来确保其拥有所有必要的权限。有关更多信息，请参阅中的策略示例the section called “手动创建的 IAM 角色所需的权限”。

9. 选择 配置。这将打开带有 OpenSearch 仪表板 URL 的数据源详细信息屏幕。您可导航到此 URL 以完成后续步骤。

OpenSearch 服务 API

使用 AddDataSourceAPI 操作在您的网域中创建新的数据源。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource

{
```

```
    "DataSourceType": {
         "S3GlueDataCatalog": {
             "RoleArn": "arn:aws:iam::account-id:role/role-name"
         }
     }
    "Description": "data-source-description",
    "Name": "my-data-source"
}
```

后续步骤

访问 OpenSearch仪表板

创建数据源后，S OpenSearch ervice 会为您提供 OpenSearch仪表板链接。您可以使用它来配置访问控制、定义表、安装 out-of-the-box集成和查询数据。

有关更多信息，请参阅 the section called “配置 S3 数据源”。

映射 Amazon Glue Data Catalog 角色

如果您在创建数据源后启用了精细访问控制，则必须将非管理员用户映射到具有 Amazon Glue Data Catalog 访问权限的 IAM 角色才能运行直接查询。要手动创建可以映射到 IAM `glue_access` 角色的后端角色，请执行以下步骤：

> **Note**
>
> 索引用于针对数据来源的任何查询。对给定数据来源的请求索引具有读取权限的用户可以读取针对该数据来源的所有查询。对结果索引具有读取权限的用户可以读取针对该数据来源的所有查询的结果。

1. 从 “ OpenSearch 控制面板” 的主菜单中，选择 “安全”、“角色” 和 “创建角色”。
2. 将该角色命名为 glue_access。
3. 对于集群权限，选择 `indices:data/write/bulk*`、`indices:data/read/scroll` 和 `indices:data/read/scroll/clear`。
4. 对于索引，输入想要授予具有角色访问权限的用户的以下索引：
   - `.query_execution_request_`*`<name of data source>`*
   - `query_execution_result_`*`<name of data source>`*
   - `.async-query-scheduler`

- `flint_*`

5. 对于索引权限，选择 `indices_all`。
6. 选择创建。
7. 选择映射的用户、管理映射。
8. 在后端角色下，添加需要权限才能调用域的 Amazon Glue 角色的 ARN。

```
arn:aws:iam::account-id:role/role-name
```

9. 选择映射并确认在映射的用户下显示的角色。

有关表映射角色的更多信息，请参阅 the section called “将角色映射到用户”。

其他资源

手动创建的 IAM 角色所需的权限

在为您的域创建数据源时，您可以选择一个 IAM 角色来管理对数据的访问权限。您有两种选择：

1. 自动创建新的 IAM 角色
2. 使用您手动创建的现有 IAM 角色

如果您使用手动创建的角色，则需要为该角色附加正确的权限。这些权限必须允许访问特定的数据源，并允许 S OpenSearch ervice 代入该角色。这是必需的，这样 OpenSearch 服务才能安全地访问您的数据并与之交互。

以下示例策略演示了创建和管理数据来源所需的最低权限。如果您拥有更广泛的权限，例如 `s3:*` 或 `AdminstratorAccess` 策略，则这些权限包含示例策略中的最低权限。

在以下示例政策中，*`placeholder text`* 用您自己的信息替换。

JSON

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

```
            "Action":"es:ESHttp*",
            "Resource":"arn:aws:es:region:account:domain/<domain_name>/*"
        },
        {
            "Sid":"AmazonOpenSearchS3GlueDirectQueryReadAllS3Buckets",
            "Effect":"Allow",
            "Action":[
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:ListBucket"
            ],
            "Condition":{
                "StringEquals":{
                    "aws:ResourceAccount":"account"
                }
            },
            "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:account:table/*",
            "arn:aws:glue:us-east-1:account:database/*",
            "arn:aws:glue:us-east-1:account:catalog",
            "arn:aws:es:region:account:domain/domain_name"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceAccount":"account"
            }
         }
      },
      {
         "Sid":"ReadAndWriteActionsForS3CheckpointBucket",
         "Effect":"Allow",
         "Action":[
            "s3:ListMultipartUploadParts",
            "s3:DeleteObject",
            "s3:GetObject",
            "s3:PutObject",
            "s3:GetBucketLocation",
            "s3:ListBucket"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceAccount":"account"
            }
         },
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
      }
   ]
}
```

要支持不同账户中的 Amazon S3 存储桶，您需要在 Amazon S3 策略中包含一个条件并添加相应的账户。

在以下示例条件中，*placeholder text* 用您自己的信息替换。

```
"Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{accountId}}"
                }
```

该角色还必须具有指定目标 ID 的以下信任策略。

JSON

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

有关创建角色的说明，请参阅使用自定义信任策略创建角色。

如果您在 S OpenSearch ervice 中启用了细粒度访问控制，则将自动为您的数据源创建一个新的 OpenSearch细粒度访问控制角色。该新的精细访问控制角色名称将为 `Amazon OpenSearchDirectQuery <name of data source>`。

默认情况下，该角色只能访问直接查询数据来源索引。尽管您可以配置该角色以限制或授予对数据来源的访问权限，但建议不要调整该角色的访问权限。如果您删除数据来源，该角色也将被删除。如果任何其他用户被映射到该角色，则将移除其访问权限。

在 OpenSearch 控制面板中配置和查询 S3 数据源

现在，您已创建数据来源，可以配置安全设置、定义 Amazon S3 表或设置加速数据索引。在查询数据之前，本节将引导您了解 OpenSearch 仪表板中数据源的各种用例。

要配置以下部分，必须先在 OpenSearch 仪表板中导航到您的数据源。在左侧导航的管理下，选择数据来源。在管理数据来源下，选择您在控制台中创建的数据来源的名称。

使用 Query Workbench 创建 Spark 表

从 OpenSearch 服务直接查询到 Amazon S3 使用中的 Spark 表 Amazon Glue Data Catalog。您可以从查询工作台中创建表格，而不必离开 OpenSearch 控制面板。

要管理数据源中的现有数据库和表，或者要创建要使用直接查询的新表，请从左侧导航栏中选择 Query Workbench，然后从数据源下拉列表中选择 Amazon S3 数据源。

要设置以 Parquet 格式存储在 S3 中的 VPC 流日志表，请运行以下查询：

```
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"
```

创建该表后，运行以下查询以确保其与直接查询兼容：

```
MSCK REPAIR TABLE  datasourcename.databasename.vpclogstable
```

为常用 Amazon 日志类型设置集成

您可以将存储在 Amazon S3 中的 Amazon 日志类型与 OpenSearch 服务集成。使用 OpenSearch 仪表板安装用于创建 Amazon Glue Data Catalog 表格、保存的查询和仪表板的集成。这些集成使用索引视图来更新仪表板。

有关安装集成的说明，请参阅 OpenSearch文档中的安装集成资产。

选择集成时，请确保其包含`S3 Glue`标签。

设置集成时，请为连接类型指定 S3 连接。然后，选择集成的数据源、数据的 Amazon S3 位置、管理加速索引的检查点以及您的用例所需的资产。

> **Note**
>
> 确保您的检查点的 S3 存储桶具有该检查点位置的写入权限。如果没有这些权限，集成的加速将失败。

设置访问控制

在数据来源的详细信息页面上，找到访问控制部分，然后选择编辑。如果域启用了细粒度访问控制，请选择“受限制”，然后选择要向哪些角色提供对新数据源的访问权限。如果您只想让管理员访问数据来源，也可以选择仅管理员。

> **Important**
>
> 索引用于针对数据来源的任何查询。对给定数据来源的请求索引具有读取权限的用户可以读取针对该数据来源的所有查询。对结果索引具有读取权限的用户可以读取针对该数据来源的所有查询的结果。

在“ OpenSearch发现”中查询 S3 数据

设置表并配置所需的可选查询加速后，就可以开始分析数据了。要查询您的数据，请从下拉菜单中选择您的数据源。如果您使用的是 Amazon S3 和 OpenSearch 控制面板，请前往“发现”并选择数据源名称。

如果您使用的是跳过索引或尚未创建索引，则可以使用 SQL 或 PPL 来查询数据。如果您已配置实体化视图或覆盖索引，则您已有索引，并可在整个控制面板中使用控制面板查询语言（DQL）。您也可以将 PPL 与可观测性插件结合使用，将 SQL 与查询工作台插件结合使用。目前，只有可观测性和查询工作台插件支持 PPL 和 SQL。要使用 OpenSearch 服务 API 查询数据，请参阅异步 API 文档。

> **Note**
>
> 并非所有的 SQL 和 PPL 语句、命令和函数都受支持。有关支持的命令的列表，请参阅the section called “支持的 SQL 和 PPL 命令”。
> 如果您创建了物化视图或覆盖索引，则可以使用 DQL 来查询数据，前提是您已在其中对其进行了索引。

故障排除

在某些情况下，返回的结果可能不合预期。如果您遇到任何问题，请务必遵循以下内容the section called “建议”。

直接查询 Amazon CloudWatch Logs OpenSearch 服务中的数据

本节将引导您完成在 Amazon Ser OpenSearch vice 中创建和配置数据源集成的过程，使您能够高效地查询和分析存储在 CloudWatch 日志中的数据。

在接下来的页面中，您将学习如何设置 CloudWatch Logs 直接查询数据源、浏览必要的先决条件以及如何使用执行 step-by-step程序。 Amazon Web Services Management Console

主题

- 在 OpenSearch 服务中创建 Ama CloudWatch zon Logs 数据源集成
- 在 OpenSearch 仪表板中配置和查询 CloudWatch 日志数据源

在 OpenSearch 服务中创建 Ama CloudWatch zon Logs 数据源集成

如果您使用 Amazon OpenSearch Serverless 来满足您的可观察性需求，那么您现在无需将数据复制或摄取到服务中即可分析您的 Amazon CloudWatch 日志。 OpenSearch 此功能利用直接查询来查询数据，类似于分析来自 OpenSearch 服务的 Amazon S3 中的数据。您可以先从 Amazon 管理控制台创建新的连接数据源。

您可以创建新的数据源来分析 CloudWatch 日志数据，而不必构建 Amazon OpenSearch Serverless 来直接查询日志中的操作日 CloudWatch 志。这使您能够分析在 OpenSearch 服务之外访问的运营数据。通过跨 OpenSearch 服务和 CloudWatch 日志进行查询，您可以开始分析日志中的日 CloudWatch 志，然后返回到监控中的数据源， OpenSearch 而无需切换工具。

要使用此功能，您需要通过 Amazon 管理控制台为 OpenSearch 服务创建 CloudWatch 日志直接查询数据源。

主题

- 先决条件
- 过程
- 后续步骤
- 其他资源

先决条件

在开始之前，请确保您已经阅读了以下文档：

- the section called “Amazon CloudWatch 日志的限制”
- the section called “CloudWatch 日志推荐”
- the section called “ CloudWatch 日志配额”

在创建数据源之前，您必须拥有以下资源 Amazon Web Services 账户：

- 启用 CloudWatch 日志。配置 CloudWatch 日志以收集与您的 OpenSearch 资源 Amazon Web Services 账户 相同的日志。有关说明，请参阅 Amazon Logs 用户指南中的 CloudWatch CloudWatch 日志入门。
- 一个或多个 CloudWatch 日志组。您可以指定包含要查询的数据的日志组。有关创建日志组的说明，请参阅 Amazon Logs 用户指南中的 CloudWatch CloudWatch 日志中的创建日志组。
- （可选）手动创建的 IAM 角色。您可以使用此角色来管理对数据源的访问权限。或者，您可以让 S OpenSearch ervice 自动为您创建具有所需权限的角色。如果您选择使用手动创建的 IAM 角色，请按照中的指导进行操作the section called “手动创建的 IAM 角色所需的权限”。

过程

您可以使用设置集合级别的查询数据源。 Amazon Web Services Management Console

要设置集合级别的数据源，请使用 Amazon Web Services Management Console

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，转到中央管理，然后选择连接的数据源。

3. 选择连接。
4. 选择CloudWatch作为数据源类型。
5. 选择下一步。
6. 在数据连接详细信息下，输入名称和可选描述。
7. 在 IAM 角色下，选择如何管理对日志组的访问权限。
   a. 如果要自动为此数据源创建角色，请执行以下步骤：
      i. 选择 “创建新角色”。
      ii. 输入 IAM 角色的名称。
      iii. 选择一个或多个日志组来定义可以查询哪些数据。
   b. 如果您想使用自己管理的现有角色，请按照以下步骤操作：
      i. 选择使用现有角色。
      ii. 从下拉菜单中选择现有角色。

> **Note**
> 使用自己的角色时，必须通过从 IAM 控制台附加必需的策略来确保其拥有所有必要的权限。有关更多信息，请参阅 the section called “手动创建的 IAM 角色所需的权限”。

8. （可选）在 “标签” 下，向数据源添加标签。
9. 选择下一步。
10. 在 “设置” 下 OpenSearch，选择设置方式 OpenSearch。
    a. 使用默认设置：
       - 查看默认资源名称和数据保留设置。我们建议您使用自定义名称。

         使用默认设置时，将为您创建一个新的 OpenSearch应用程序和 Essentials 工作区，无需支付额外费用。 OpenSearch 使您能够分析多个数据源。它包括工作区，可为热门用例提供量身定制的体验。Workspaces 支持访问控制，使您能够为用例创建私密空间，并且仅与合作者共享。
    b. 使用自定义设置：
       i. 选择自定义。

   ii. 根据需要编辑集合名称和数据保留设置。
   iii. 选择要使用的 OpenSearch 应用程序和工作区。
11. 选择下一步。
12. 查看您的选择，如果需要进行任何更改，请选择 “编辑”。
13. 选择 Connect 以设置数据源。创建数据源后，请停留在此页面上。准备就绪后，您将被带到数据源详细信息页面。

后续步骤

访问 OpenSearch 仪表板

创建数据源后，S OpenSearch ervice 会为您提供 OpenSearch仪表板 URL。您可以使用它来配置访问控制、定义表、为常用日志类型设置基于日志类型的仪表板，以及使用 SQL 或 PPL 查询数据。

有关更多信息，请参阅 the section called “配置 CloudWatch 日志数据源”。

其他资源

手动创建的 IAM 角色所需的权限

创建数据源时，您可以选择一个 IAM 角色来管理对数据的访问权限。您有两种选择：

1. 自动创建新的 IAM 角色
2. 使用您手动创建的现有 IAM 角色

如果您使用手动创建的角色，则需要为该角色附加正确的权限。这些权限必须允许访问特定的数据源，并允许 S OpenSearch ervice 代入该角色。这是必需的，这样 OpenSearch 服务才能安全地访问您的数据并与之交互。

以下示例策略演示了创建和管理数据来源所需的最低权限。如果您拥有更广泛的权限，例如 `logs:*` 或 `AdminstratorAccess` 策略，则这些权限包含示例策略中的最低权限。

在以下示例政策中，*`placeholder text`* 用您自己的信息替换。

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

该角色还必须具有指定目标 ID 的以下信任策略。

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:region:accountId:datasource/rolename"
                }
            }
        }
    ]
}
```

有关创建角色的说明，请参阅使用自定义信任策略创建角色。

默认情况下，该角色只能访问直接查询数据来源索引。尽管您可以配置该角色以限制或授予对数据来源的访问权限，但建议不要调整该角色的访问权限。如果您删除数据来源，该角色也将被删除。如果任何其他用户被映射到该角色，则将移除其访问权限。

在 OpenSearch 仪表板中配置和查询 CloudWatch 日志数据源

现在，您已经创建了数据源，可以开始使用它了 Dashboar OpenSearch ds。本节将引导您了解 OpenSearch 仪表板中数据源的各种用例。

从 “发现” 页面查询日志组

在 OpenSearch 发现页面中，您可以使用配置的新直接查询数据源来查询 CloudWatch 日志组。为此，请选择 “浏览日志”，然后使用搜索栏使用 SQL 或 PPL 生成查询。您可以对日志组返回的数据进行筛选、排序和可视化。要了解 CloudWatch 日志集成支持哪些语句、命令和限制，请参阅the section called “支持的 SQL 和 PPL 命令”。

为您的数据源创建仪表板视图

使用 S OpenSearch ervice 时，您可以使用预先构建的仪表板模板快速分析常用 Amazon 日志类型。对于 CloudWatch 日志，有 VPC CloudTrail、和 WAF 日志的模板。这些模板允许您快速创建针对您的特定数据量身定制的仪表板。它们包括为该特定日志类型量身定制的仪表板。这使您能够快速启动并运行分析这些流行的 Amazon 日志源，而不必从头开始构建所有内容。

> **Note**
>
> 仪表板使用索引视图，该视图使用直接查询 OpenSearch Compute Units (OCUs) 以及无服务器集合索引OCUsOCUs、搜索和存储从 CloudWatch 日志中提取数据。

按照以下步骤使用其中一个预建模板创建仪表板，这样您就可以立即开始探索和分析数据。

创建仪表板视图

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择中央管理，然后选择连接的数据源。
3. 选择数据源以打开详细信息页面。
4. 请选择创建控制面板。
5. 选择要创建的仪表板类型。
6. 输入仪表板的名称。
7. 输入控制面板的可选描述。
8. 选择一个或多个要在控制面板上查看的日志组。
9. 选择刷新仪表板中数据的频率。
10. 选择要使用 OpenSearch 的工作空间。
    a. 要创建新工作区，请选择创建新工作区并输入名称。
    b. 要使用现有工作空间，请选择选择现有工作空间。
11. 请选择创建控制面板。

在 D OpenSearch iscover 中查询 CloudWatch 日志数据

要查询您的数据，请从下拉菜单中选择您的数据源。如果您使用的是 CloudWatch 日志，请从 Essentials 工作区导航到 “发现”，然后开始使用 OpenSearch SQL 或管道处理语言 (PPL) 查询数据。有关支持的命令的列表，请参阅the section called “支持的 SQL 和 PPL 命令”。

> **Note**
>
> 如果您创建了实例化视图，则可以使用 DQL 来查询数据，前提是您已在其中为数据编制了索引。

故障排除

在某些情况下，返回的结果可能不合预期。如果您遇到任何问题，请务必遵循以下内容the section called “建议”。

直接查询 OpenSearch 服务中的 Amazon Security Lake 数据

本节将引导您完成在 Amazon Ser OpenSearch vice 中创建和配置数据源集成的过程，使您能够高效地查询和分析存储在 Security Lake 中的数据。

在接下来的页面中，您将学习如何设置 Security Lake 直接查询数据源、浏览必要的先决条件，以及如何使用执行 step-by-step程序。 Amazon Web Services Management Console

主题

- 在 OpenSearch 服务中创建 Amazon Security Lake 数据源集成
- 在 OpenSearch 控制面板中配置和查询 Security Lake 数据源

在 OpenSearch 服务中创建 Amazon Security Lake 数据源集成

您可以使用 Amazon OpenSearch Serverless 直接查询亚马逊安全湖中的安全数据。为此，您需要创建一个数据源，使您能够在 Security Lake 数据上使用 OpenSearch零 ETL 功能。创建数据源时，您可以直接搜索存储在 Security Lake 中的数据，从中获取见解并分析这些数据。您可以提高查询性能，并使用按需索引对选定的 Security Lake 数据集进行高级 OpenSearch 分析。

主题

- 先决条件
- 过程
- 后续步骤
- 其他资源

先决条件

在开始之前，请确保您已经阅读了以下文档：

- the section called “Amazon 安全湖的限制”
- the section called “安全湖建议”

- the section called “安全湖配额”

在创建数据源之前，请先在 Security Lake 中执行以下操作：

- 启用 Security Lake。将 Security Lake 配置为收集与您的 OpenSearch 资源 Amazon Web Services 区域 相同的日志。有关说明，请参阅亚马逊安全湖用户指南中的亚马逊安全湖入门。
- 设置 Security Lake 权限。确保您已接受资源管理的服务关联角色权限，并且控制台在 “问题” 页面下未显示任何问题。有关更多信息，请参阅 Amazon Sec urity Lake 用户指南中的安全湖服务相关角色。
- 共享安全湖数据源。使用与 Security Lake 相同的账户进行访问时，请确保没有消息要求 OpenSearch 在 Security Lake 控制台中向 Lake Formation 注册您的 Security Lake 存储桶。要进行跨账户 OpenSearch 访问，请在 Security Lake 控制台中设置 Lake Formation 查询订阅者。使用与您的 OpenSearch 资源关联的账户作为订阅者。有关更多信息，请参阅 Amazon Sec urity Lake 用户指南中的安全湖中的订阅者管理。

此外，您还必须拥有以下资源 Amazon Web Services 账户：

- （可选）手动创建的 IAM 角色。您可以使用此角色来管理对数据源的访问权限。或者，您可以让 S OpenSearch ervice 自动为您创建具有所需权限的角色。如果您选择使用手动创建的 IAM 角色，请按照中的指导进行操作the section called “手动创建的 IAM 角色所需的权限”。

过程

您可以将数据源设置为从内部与 Security Lake 数据库连接 Amazon Web Services Management Console。

要使用设置数据源 Amazon Web Services Management Console

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，转到中央管理，然后选择连接的数据源。
3. 选择连接。
4. 选择 S ecurity Lak e 作为数据源类型。
5. 选择下一步。
6. 在数据连接详细信息下，输入名称和可选描述。
7. 在 IAM 权限访问设置下，选择如何管理对您的数据源的访问权限。

a. 如果要自动为此数据源创建角色，请执行以下步骤：

   i. 选择 “创建新角色”。

   ii. 输入 IAM 角色的名称。

   iii. 选择一个或多个 Amazon Glue 表来定义可以查询哪些数据。

b. 如果您想使用自己管理的现有角色，请按照以下步骤操作：

   i. 选择使用现有角色。

   ii. 从下拉菜单中选择现有角色。

> **Note**
>
> 使用自己的角色时，必须通过从 IAM 控制台附加必需的策略来确保其拥有所有必要的权限。有关更多信息，请参阅 the section called “手动创建的 IAM 角色所需的权限”。

8. （可选）在 “标签” 下，向数据源添加标签。
9. 选择下一步。
10. 在 “设置” 下 OpenSearch，选择设置方式 OpenSearch。

    - 查看默认资源名称和数据保留设置。

      使用默认设置时，将为您创建一个新的 OpenSearch应用程序和 Essentials 工作区，无需支付额外费用。 OpenSearch 使您能够分析多个数据源。它包括工作区，可为热门用例提供量身定制的体验。Workspaces 支持访问控制，使您能够为用例创建私密空间，并且仅与合作者共享。

11. 使用自定义设置：

    a. 选择自定义。

    b. 根据需要编辑集合名称和数据保留设置。

    c. 选择要使用的 OpenSearch 应用程序和工作区。

12. 选择下一步。
13. 查看您的选择，如果需要进行任何更改，请选择 “编辑”。
14. 选择 Connect 以设置数据源。创建数据源后，请停留在此页面上。准备就绪后，您将被带到数据源详细信息页面。

后续步骤

访问 OpenSearch 仪表板并创建仪表板

创建数据源后，S OpenSearch ervice 会为您提供 OpenSearch仪表板 URL。您可以使用它通过 SQL 或 PPL 查询数据。Security Lake 集成附带了预打包的 SQL 和 PPL 查询模板，可帮助您开始分析日志。

有关更多信息，请参阅 the section called “配置安全湖数据源”。

其他资源

手动创建的 IAM 角色所需的权限

创建数据源时，您可以选择一个 IAM 角色来管理对数据的访问权限。您有两种选择：

1. 自动创建新的 IAM 角色
2. 使用您手动创建的现有 IAM 角色

如果您使用手动创建的角色，则需要为该角色附加正确的权限。这些权限必须允许访问特定的数据源，并允许 S OpenSearch ervice 代入该角色。这是必需的，这样 OpenSearch 服务才能安全地访问您的数据并与之交互。

以下示例策略演示了创建和管理数据来源所需的最低权限。如果您拥有更广泛的权限（例如`AdminstratorAccess`策略），则这些权限包括示例策略中的最低权限权限。

在以下示例政策中，*`placeholder text`* 用您自己的信息替换。

JSON

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

```
            ],
            "Resource": "arn:aws:aoss:region:account: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:region:account:table/databasename/*",
                "arn:aws:glue:region:account:database/databasename",
                "arn:aws:glue:region:account:catalog",
                "arn:aws:glue:region:account:database/default"
            ]
        },
        {
            "Sid": "AmazonOpenSearchDirectQueryLakeFormationAccess",
            "Effect": "Allow",
            "Action": [
                "lakeformation:GetDataAccess"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

该角色还必须具有指定目标 ID 的以下信任策略。

JSON

```
{
```

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

有关创建角色的说明，请参阅使用自定义信任策略创建角色。

默认情况下，该角色只能访问直接查询数据来源索引。尽管您可以配置该角色以限制或授予对数据来源的访问权限，但建议不要调整该角色的访问权限。如果您删除数据来源，该角色也将被删除。如果任何其他用户被映射到该角色，则将移除其访问权限。

查询使用客户托管密钥加密的 Security Lake 数据

如果与数据连接关联的 Security Lake 存储桶使用服务器端加密并由客户托管 Amazon KMS key，则必须将 LakeFormation 服务角色添加到密钥策略中。这允许服务访问和读取您的查询数据。

在以下示例政策中，*placeholder text* 用您自己的信息替换。

```
{
    "Sid": "Allow LakeFormation to access the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::account:role/aws-service-role/lakeformation.amazonaws.com/
AWSServiceRoleForLakeFormationDataAccess"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

在 OpenSearch 控制面板中配置和查询 Security Lake 数据源

现在，您已经创建了数据源，可以在 OpenSearch 仪表板中对其进行设置。

在查询数据之前，本节将引导您了解 OpenSearch 仪表板中数据源的各种用例。首先，您需要在 OpenSearch 仪表板中导航到您的数据源。在左侧菜单的管理下，选择数据源。然后，选择您之前在 OpenSearch 服务控制台中创建的数据源的名称。

从 Discover 中查询安全湖表

如果您已根据 Security Lake 日志创建了表，则现在可以直接从 D OpenSearch iscover 中查询这些表。这使您能够直接从熟悉的 Discover 界面无缝访问和分析存储在 Security Lake 中的数据。通过直接从 Discover 中查询 Security Lake，您无需手动提取、转换数据并将其加载到单独的搜索索引中。为了快速开始分析您的日志，Discover 包括一组 PPL 和 SQL 保存的查询。

首先选择您配置的数据源。选择要查询的关联数据库和表，然后使用搜索栏针对您的表编写查询。要了解 Security Lake 集成支持哪些语句、命令和限制，请参阅the section called “支持的 SQL 和 PPL 命令”。

要利用可用于 Security Lake 的预建查询，请访问... 在 Discover 的右上角，选择 “打开查询”，然后选择 “模板”。Security Lake 中支持的日志源有许多预先构建的查询可用。搜索与您的用例相匹配的模板，复制要在搜索栏中使用的查询，然后用您自己的信息替换模板字段（例如 “区域” 和 “操作”）。

加速 Discover 中的数据

为了提高性能并在中更快地进行后续查询和分析 OpenSearch，您可以将 Discover 中的查询结果提取到 OpenSearch 索引视图中。

创建索引视图

1. 从 “发现” 中，选择 “创建索引视图”。
2. 在查询编辑器中，输入所需的查询。您可以在此处创建新查询，也可以使用之前搜索的现有查询。
3. 为新的索引视图指定名称。选择一个描述性名称，以帮助您稍后识别视图。
4. 为您的索引视图配置数据保留设置。您可以指定数据应在索引中保留多长时间，从而在性能和存储成本之间取得平衡。
5. 创建索引视图。创建索引视图后，您的索引视图将可用于更快地进行查询和分析。

如果您之前创建过索引视图，则可以从 Discover 中访问这些视图。

使用现有的索引视图

1. 从 “发现” 中，选择 “选择索引视图” 以查看 Security Lake 现有索引视图的列表。
2. 选择要使用的索引视图。这会将视图应用于您当前的查询，从而可能显著加快您的数据检索和分析速度。

为您的数据源创建仪表板视图

使用 S OpenSearch ervice 时，您可以使用预先构建的仪表板模板分析常见的 Amazon 日志类型。对于安全湖，有 VPC CloudTrail、和 WAF 日志的模板。这些模板允许您创建针对您的特定数据量身定制的仪表板。它们包括为该特定日志类型量身定制的预建查询和仪表板。这使您能够快速启动并运行分析这些流行的 Amazon 日志源，而不必从头开始构建所有内容。

> **Note**
>
> 仪表板使用索引视图，这些视图从 Security Lake 提取数据，并有助于直接查询和收集计算。

按照以下步骤使用其中一个预建模板创建仪表板，这样您就可以立即开始探索和分析数据。

创建仪表板视图

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择中央管理，然后选择连接的数据源。
3. 选择数据源以打开详细信息页面。
4. 请选择创建控制面板。
5. 选择要创建的仪表板类型。
6. 输入仪表板的名称。
7. 输入控制面板的可选描述。
8. 选择一个或多个 Amazon Glue 表格以在控制面板上查看。
9. 选择刷新仪表板中数据的频率。
10. 选择要使用 OpenSearch 的工作空间。
    a. 要创建新工作区，请选择创建新工作区。
    b. 要使用现有工作空间，请选择选择现有工作空间。
11. 输入工作空间的名称。

12. 请选择创建控制面板。

故障排除

在某些情况下，返回的结果可能不合预期。如果您遇到任何问题，请务必遵循以下内容the section called “建议”。

在 Amazon OpenSearch 服务中管理数据源

管理数据源是维护直接查询数据源和其他 Amazon 解决方案的可靠性、可用性和性能的重要组成部分。 Amazon 提供了以下工具，用于监控、报告问题并在适当时自动采取措施。

主题

- 使用 CloudWatch 指标数据源进行监控
- 启用和禁用数据来源
- 用 Amazon 预算进行监控
- 删除数据来源

使用 CloudWatch 指标数据源进行监控

您可以使用监控直接查询 CloudWatch。 CloudWatch 收集原始数据并将其处理成可读的、近乎实时的指标。这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解您的 Web 应用程序或服务的执行情况。

您还可以设置警报来监控特定的阈值，并在达到相关阈值时发送通知或执行操作。有关更多信息，请参阅什么是亚马逊 CloudWatch。

亚马逊 S3 报告了以下指标：

指标	描述
AsyncQueryCreateAPI	为了创建异步查询而向该 API 发出的请求总数。 相关统计数据：平均值、最大值、总计 维度：`ClientId`、`DomainName` 频率：60 秒

指标	描述
AsyncQueryGetApiRequestCount	为了检索异步查询结果而向该 API 发出的请求总数。 相关统计数据：平均值、最大值、总计 维度：`ClientId`、`DomainName` 频率：60 秒
AsyncQueryCancelApiRequestCount	为了取消异步查询而向该 API 发出的请求总数。 相关统计数据：平均值、最大值、总计 维度：`ClientId`、`DomainName` 频率：60 秒
AsyncQueryGetApiFailedRequestCusErrCount	检索异步查询结果时由于客户相关错误（例如，查询 ID 无效）而失败的请求数。 相关统计数据：平均值、最大值、总计 维度：`ClientId`、`DomainName` 频率：60 秒
AsyncQueryCancelApiFailedRequestCusErrCount	检索异步查询结果时由于客户相关错误（例如，查询 ID 无效）而失败的请求数。 相关统计数据：平均值、最大值、总计 维度：`ClientId`、`DomainName` 频率：60 秒

指标	描述
AsyncQueryCancelApiFailedRequestSysErrCount	创建异步查询时由于客户相关错误而失败的请求数。 相关统计数据：平均值、最大值、总计 维度：`ClientId`、`DomainName` 频率：60 秒
AsyncQueryGetApiFailedRequestSysErrCount	检索异步查询结果时由于系统相关错误而失败的请求数。 相关统计数据：平均值、最大值、总计 维度：`ClientId`、`DomainName` 频率：60 秒

CloudWatch 日志和安全湖报告以下指标：

指标	描述
DirectQueryRate	对数据源发出的请求率。 相关统计数据：总和、最大值、最小值、平均值 维度：`DataSourceName` 频率：60 秒
DirectQueryLatency	在数据源上运行查询时观察到的延迟。 相关统计数据：平均值、P90、P99、总和、最小值、最大值 维度：`DataSourceName` 频率：60 秒
FailedDirectQueries	在数据源查询中观察到的查询失败总数。

指标	描述
	相关统计数据：总和、最大值、最小值、平均值 维度：`DataSourceName` 频率：60 秒
DirectQueryConsumedOCU	在数据源上运行查询所消耗的数量。 OCUs 相关统计数据：平均值、P90、P99、总和、最小值、最大值 维度：`DataSourceName` 频率：60 秒

启用和禁用数据来源

> **Note**
> 以下信息仅适用于 Amazon S3 数据源。

如果您想停止对数据来源使用直接查询功能，则可以选择禁用该数据来源。禁用数据源将完成现有查询的执行并停止执行所有新查询。

禁用数据源后，用于提高查询性能的加速设置（例如跳过索引、实例化视图、覆盖索引）将设置为手动。禁用数据来源后将其设置为活动状态时，用户查询将按预期运行。先前建立并设置为手动的加速，需要手动配置后才能再次按计划运行。

用 Amazon 预算进行监控

亚马逊 OpenSearch 服务正在将账户级别的 OCU 使用数据填充到账单和成本管理的 Cost Explorer 中。您可以在账户级别考虑 OCU 的使用情况，并在超过阈值时设置阈值和提醒。

在 Cost Explorer 中筛选的使用类型格式看起来像 RegionCode-DirectQuery OCU（OCU-Hours）。如果您想在 DirectQuery OCU（OCU 小时数）使用量达到阈值时收到通知，您可以创建一个 Amazon 预算账户，并根据您设置的阈值配置提醒。或者，对于 Amazon S3，您可以设置一个 Amazon SNS 主题，该主题将在满足阈值标准时关闭数据源。

> **Note**
>
> Amazon 预算中的使用数据不是实时的，最多可能会延迟 8 小时。

删除数据来源

当您删除数据源时，Amazon S OpenSearch ervice 会将其从您的域名或您的集合中删除。OpenSearch 服务还会删除与数据源关联的索引。您的交易数据不会从对方删除 Amazon Web Services 服务，但对方 Amazon Web Services 服务 不会向 OpenSearch 服务发送新数据。

您可以使用 Amazon Web Services Management Console 或 OpenSearch 服务 API 删除数据源集成。

Amazon Web Services Management Console

删除 Amazon S3 数据源

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择域。
3. 选择要删除其数据来源的域。随即打开域详细信息页面。选择常规信息下方的连接选项卡，然后找到直接查询部分。
4. 选择要删除的数据来源，然后选择删除并确认删除。

删除 CloudWatch 日志或安全湖数据源

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择中央管理，然后选择连接的数据源。
3. 选择要删除的数据来源，然后选择删除并确认删除。

OpenSearch 服务 API

要删除 Amazon S3 数据源，请使用 DeleteDataSourceAPI 操作。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource/data-source-name
```

要删除 CloudWatch 日志或安全湖数据源，请使用 DeleteDirectQueryDataSourceAPI 操作。

优化 Amazon OpenSearch 服务数据源的查询性能

当您访问外部数据源时，Amazon Ser OpenSearch vice 中的查询性能可能会降低。这可能是由于网络延迟、数据转换或大数据量等因素造成的。要提高性能，请考虑根据用例为选定数量的数据编制索引：

- 加快在 Amazon S3 上的直接查询（跳过索引）
- 在 Security Lake 上构建仪表板可视化（物化视图）
- 使用索引视图提取查询结果以供离线查看或提高 Security Lake（物化视图）上的性能

有关加速查询的完整文档（包括示例查询），请参阅开源文档中的使用 OpenSearch 索引优化查询性能。

主题

- 跳过索引
- 实体化视图
- 覆盖索引

跳过索引

跳过索引仅提取存储在 Amazon S3 中的数据的元数据。当您查询带有跳过索引的表时，查询计划器会使用该索引重写查询，无需扫描所有分区和文件即可有效地识别数据的位置。这种方法有助于缩小存储数据的确切位置。

有两种方法可以创建跳过索引。第一种方法是从数据源详细信息中自动生成跳过的索引。第二种方法是使用 Query Workbench 使用 SQL 语句手动创建跳过的索引。

要从数据源自动生成跳过的索引，请转到仪表板管理和加速数据，然后选择您的数据库和表（可能需要刷新才能获得最新的数据库和表）。然后，您可以选择 Gen er ate 以自动生成跳过的索引，也可以手动选择要编制索引的每个字段并指定加速（跳过索引类型）。最后，选择 “创建加速” 以创建一个填充新的跳过索引的重复作业。

只有 Amazon S3 数据源支持跳过索引。

有关使用 Query Workbench 设置跳过索引的更多信息，请参阅文档中的跳过索引。 OpenSearch

实体化视图

物化视图使用复杂的查询（例如聚合）来支持 OpenSearch 仪表板可视化。他们根据查询提取您的数据子集并将其存储在 OpenSearch 索引中。然后，您可以使用此索引来创建可视化效果。

Amazon S3 和安全湖数据源支持物化视图。

有关使用 Query Workbench 设置物化视图的更多信息，请参阅文档中的物化视图。 OpenSearch

覆盖索引

覆盖索引从表中的指定列中提取数据，并基于这些数据 OpenSearch创建新索引。您可以将此新索引用于可视化和其他 OpenSearch 功能，例如异常检测或地理空间分析。

只有 Amazon S3 数据源支持覆盖索引。

有关设置覆盖索引的更多信息，请参阅 OpenSearch 文档中的覆盖索引。

支持的 SQL 和 PPL 命令

OpenSearch SQL 和 OpenSearch 管道处理语言 (PPL) 是用于查询、分析和处理 L CloudWatch ogs Insights 和 Secur OpenSearch ity Lake 中数据的语言。您可以在 D OpenSearch iscover 中使用 OpenSearch SQ OpenSearch L 和 PPL 来查询 CloudWatch 日志、Amazon S3 或安全湖中的数据。CloudWatch Logs Insights 还支持 OpenSearch PPL 和 OpenSearch SQL 查询语言，此外还支持 Logs Insights QL（一种专门用于分析日志的查询语言）。 CloudWatch

- OpenSearch SQ OpenSearch L：如果你习惯使用关系数据库，SQL 提供了一个熟悉的选项。OpenSearch SQL 提供了 SQL 功能的子集，使其成为执行临时查询和数据分析任务的理想选择。在 OpenSearch SQL 中，你可以使用 SELECT、FROM、WHERE、GROUP BY、HAVING 等命令以及 SQL 中提供的各种其他 SQL 命令和函数。您可以 JOINs 跨表（或日志组）执行，使用子查询关联表（或日志组）之间的数据，并使用丰富的 JSON、数学、字符串、条件和其他 SQL 函数集对日志和安全数据进行强有力的分析。
- OpenSearch PPL（管道处理语言）：使用 OpenSearch PPL，您可以使用 piped-together 命令检索、查询和分析数据，从而更容易理解和撰写复杂的查询。它的语法基于 Unix 管道，支持串联命令以转换和处理数据。使用 PPL，您可以筛选和聚合数据，并使用子查询 JOINs、LOOKUP 等命令以及一组丰富的数学、字符串、日期、条件和其他函数进行分析。

尽管 OpenSearch PPL 和 OpenSearch SQL 查询语言中的大多数命令在 L CloudWatch ogs 中都很常见 OpenSearch，但这些服务支持哪组命令和函数存在一些差异。有关更多详细信息，请参阅以下页面上的表格。

- the section called “支持的 SQL 命令”
  - the section called “CloudWatch 日志见解”
  - the section called “一般 SQL 限制”
- the section called “支持的 PPL 命令”
  - the section called “为使用 OpenSearch PPL 的 CloudWatch Logs Insights 用户提供的其他信息”

支持 OpenSearch 的 SQL 命令和函数

以下参考表显示了 D OpenSearch iscover 中支持哪些 SQL 命令来查询 Amazon S3、Security Lake 或 CloudWatch 日志中的数据，以及 Logs Insights 中 CloudWatch 支持哪些 SQL 命令。L CloudWatch ogs Insights 中支持的 SQL 语法与 OpenSearch Discover 中支持的 SQL 语法相同，在下表中被称为 CloudWatch 日志。 CloudWatch

> **Note**
>
> OpenSearch 还支持 SQL，用于查询已提取 OpenSearch并存储在索引中的数据。此 SQL 方言不同于直接查询中使用的 SQL，在索引上被称为 OpenSearch SQL。

主题

- 命令
- 函数
- 一般 SQL 限制
- 为使用 OpenSearch SQL 的 “ CloudWatch 日志见解” 用户提供的其他信息

命令

> **Note**
>
> 在示例命令列中，根据需要`<tableName/logGroup>`进行替换，具体取决于您要查询的数据源。

- 命令示例：`SELECT Body , Operation FROM <tableName/logGroup>`
- 如果您要查询 Amazon S3 或安全湖，请使用：`SELECT Body , Operation FROM table_name`
- 如果您要查询 CloudWatch 日志，请使用：`SELECT Body , Operation FROM `LogGroupA``

命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “SELECT 子句”	显示预测值。	支持	支持	支持	`SELECT method, status FROM <tableName/logGroup>`
the section called “WHERE 子句”	根据提供的字段条件筛选日志事件。	支持	支持	支持	`SELECT * FROM <tableName/logGroup> WHERE status = 100`
the section called “GROUP BY 子句”	根据类别对记录事件进行分组，并根据统计数据查找平均值。	支持	支持	支持	`SELECT method, status, COUNT(*) AS request_count, SUM(bytes) AS total_bytes FROM <tableName/logGroup> GROUP BY method, status`
the section called	根据分组条件	支持	支持的	支持的	`SELECT method, status,`

命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令

“HAVING 子句”	筛选结果。				`COUNT(*) AS request_count,`
					`SUM(bytes) AS total_bytes`
					`FROM`
					`<tableName/logGroup>`
					`GROUP BY`
					`method,`
					`status`
					`HAVING`
					`COUNT(*) > 5`

the section called “ORDER BY 子句”	根据顺序子句中的字段对结果进行排序。您可以按降序或升序排序。	支持的	支持的	支持的	`SELECT`
					`*`
					`FROM`
					`<tableName/logGroup>`
					`ORDER BY`
					`status DESC`

the section called “JOIN 子句”
 	
(INNER | CROSS | LEFT OUTER)
	根据常用字段连接两个表的结果。	支持（联接必须使用 Inner和 Outer关键字；SELECT 语句中仅支持一个 JOIN 操作）	支持（必须使用 Inner、Lt Outer 和 Cross 关键字进行联接）	支持（必须使用 Inner、Lt Outer 和 Cross 关键字进行联接）	`SELECT`
					`A.Body,`
					`B.Timestamp`
					`FROM`
					`<tableNameA/logGroupA> AS A`
					`INNER JOIN`
					`<tableNameB/logGroupB> AS B`
					`ON A.`requestId` = B.`requestId` `

命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “LIMIT 子句”	将结果限制在前 N 行。	支持的	支持的	支持的	`SELECT * FROM <tableName/logGroup> LIMIT 10`

命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “案例条款”	评估条件并在满足第一个条件时返回一个值。	支持的	支持的	支持的	SELECT method, status, CASE WHEN status BETWEEN 100 AND 199 THEN 'Informational' WHEN status BETWEEN 200 AND 299 THEN 'Success' WHEN status BETWEEN 300 AND 399 THEN 'Redirection' WHEN status BETWEEN 400 AND 499 THEN 'Client Error' WHEN status BETWEEN 500 AND 599 THEN 'Server Error' ELSE 'Unknown Status' END AS status_category, CASE method WHEN 'GET' THEN 'Read Operation' WHEN 'POST' THEN 'Create Operation' WHEN 'PUT' THEN 'Update Operation' WHEN 'PATCH' THEN 'Partial Update Operation' WHEN 'DELETE' THEN 'Delete Operation' ELSE 'Other Operation' END AS operation_type, bytes, datetime FROM *<tableName/logGroup>*

命令	描述	CloudWa h 日志	Amazon S3	Security Lake	示例命令

the section called “公用表表达式”	在 SELECT SERT、 TE、DEI 或 MERGE 语句中创建命名的临时结果集。	不支持	支持的	支持的	`WITH RequestStats AS (`
					`  SELECT`
					`  method,`
					`  status,`
					`  bytes,`
					`  COUNT(*) AS request_count`
					`  FROM`
					`  tableName`
					`  GROUP BY`
					`  method,`
					`  status,`
					`  bytes`
					`)`
					`SELECT`
					`  method,`
					`  status,`
					`  bytes,`
					`  request_count`
					`FROM`
					`  RequestStats`
					`WHERE`
					`  bytes > 1000`

the section called “EXPLAIN	显示 SQL 语句的执行计划，但未实际执行该语句。	不支持	支持的	支持的	`EXPLAIN`
					`SELECT`
					`  k,`
					`  SUM(v)`
					`FROM`
					`  VALUES`
					`  (1, 2),`
					`  (1, 3) AS t(k, v)`
					`GROUP BY`
					`  k`

命令	描述	CloudWa h 日志	Amazon S3	Security Lake	示例命令
the section called “横向子查询子句”	允许 FROM 子句中的子查询引用同一 FROM 子句中前面各项中的列。	不支持	支持的	支持的	`SELECT * FROM tableName LATERAL ( SELECT * FROM t2 WHERE t1.c1 = t2.c1 )`
the section called “横向视图条款”	通过对基表的每一行应用表生成函数来生成虚拟表。	不支持	支持的	支持的	`SELECT * FROM tableName LATERAL VIEW EXPLODE(ARRAY(30, 60)) tableName AS c_age LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age`
the section called “LIKE 谓词”	使用通配符将字符串与模式进行匹配。	支持的	支持的	支持的	`SELECT method, status, request, host FROM <tableName/logGroup> WHERE method LIKE 'D%'`

命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令

the section called “OFFSET”	指定在开始从查询返回行之前要跳过的行数。	在查询中与子LIMIT结合使用时支持。例如：
		• 支持：SELECT * FROM Table LIMIT 100 OFFSET 10
		• 不支持：SELECT * FROM Table OFFSET 10
			支持的	支持的	`SELECT`
					`  method,`
					`  status,`
					`  bytes,`
					`  datetime`
					`FROM`
					`  <tableName/logGroup>`
					`ORDER BY`
					`  datetime`
					`OFFSET`
					`  10`

命令	描述	CloudWa h 日志	Amazon S3	Security Lake	示例命令
the section called “PIVOT 子句”	将行转换为列，将数据从基于行的格式转换为基于列的格式。	不支持	支持的	支持的	`SELECT * FROM ( SELECT method, status, bytes FROM <tableName/logGroup> ) AS SourceTable PIVOT ( SUM(bytes) FOR method IN ('GET', 'POST', 'PATCH', 'PUT', 'DELETE') ) AS PivotTable`
the section called “集合运算符”	合并两个或多个 SELECT 语句（例如 UNION、ERSECT CEPT）的结果。	支持的	支持的	支持的	`SELECT method, status, bytes FROM <tableName/logGroup> WHERE status = '416' UNION SELECT method, status, bytes FROM <tableName/logGroup> WHERE bytes > 20000`

命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “按子句排序”	指定返回查询结果的顺序。	支持的	支持的	支持的	`SELECT method, status, bytes FROM <tableName/logGroup> SORT BY bytes DESC`
the section called “UNPIVOT”	将列转换为行，将数据从基于列的格式转换为基于行的格式。	不支持	支持的	支持的	`SELECT status, REPLACE(method, '_bytes', '') AS request_method, bytes, datetime FROM PivotedData UNPIVOT ( bytes FOR method IN ( GET_bytes, POST_bytes, PATCH_bytes, PUT_bytes, DELETE_bytes ) ) AS UnpivotedData`

函数

> **Note**
>
> 在示例命令列中，根据需要*<tableName/logGroup>*进行替换，具体取决于您要查询的数据源。

- 命令示例：`SELECT Body , Operation FROM <tableName/logGroup>`
- 如果您要查询 Amazon S3 或安全湖，请使用：`SELECT Body , Operation FROM table_name`
- 如果您要查询 CloudWatch 日志，请使用：`SELECT Body , Operation FROM `LogGroupA``

可用的 SQL 语法	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “字符串函数”	内置函数，可以在 SQL 查询中操作和转换字符串和文本数据。例如，转换大小写、合并字符串、提取部分和清理文本。	支持的	支持的	支持的	`SELECT UPPER(method) AS upper_method, LOWER(host) AS lower_host FROM <tableName/logGroup>`
the section called “日期和时间函数”	用于处理和转换查询中的日期和时间戳数据的内置函数。例如，date_add、date_format、datediff 和当前日期。	支持的	支持的	支持的	`SELECT TO_TIMESTAMP(datetime) AS timestamp, TIMESTAMP_SECONDS(UNIX_TIMESTAMP(datetime)) AS from_seconds, UNIX_TIMESTAMP(datetime) AS to_unix, FROM_UTC_TIMESTAMP(datetime, 'PST') AS to_pst, TO_UTC_TIMESTAMP(datetime, 'EST') AS from_est FROM`

可用的 SQL 语法	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
					`<tableName/logGroup>`
the section called “聚合函数”	内置函数，可对多行执行计算以生成单个汇总值。例如，总和、计数、平均值、最大值和最小值。	支持	支持的	支持的	`SELECT COUNT(*) AS total_records, COUNT(DISTINCT method) AS unique_methods, SUM(bytes) AS total_bytes, AVG(bytes) AS avg_bytes, MIN(bytes) AS min_bytes, MAX(bytes) AS max_bytes FROM <tableName/logGroup>`
the section called “条件函数”	基于指定条件执行操作或有条件地评估表达式的内置函数。例如，大小写和如果。	支持的	支持的	支持的	`SELECT CASE WHEN method = 'GET' AND bytes < 1000 THEN 'Small Read' WHEN method = 'POST' AND bytes > 10000 THEN 'Large Write' WHEN status >= 400 OR bytes = 0 THEN 'Problem' ELSE 'Normal' END AS request_type FROM <tableName/logGroup>`

可用的 SQL 语法	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “JSON 函数”	内置函数，用于在 SQL 查询中解析、提取、修改和查询 JSON 格式的数据（例如，from_json、to_json、get_json_object、json_tuple），允许操作数据集中的 JSON 结构。	支持的	支持的	支持的	`SELECT FROM_JSON( @message, 'STRUCT< host: STRING, user-identifier: STRING, datetime: STRING, method: STRING, status: INT, bytes: INT >' ) AS parsed_json FROM <tableName/logGroup>`
the section called “数组函数”	用于在 SQL 查询中处理数组类型列的内置函数，允许诸如访问、修改和分析数组数据（例如，大小、爆炸、array_contains）之类的操作。	支持的	支持的	支持的	`SELECT scores, size(scores) AS length, array_contains(scores, 90) AS has_90 FROM <tableName/logGroup>`

可用的 SQL 语法	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “窗口函数”	内置函数，可对与当前行（窗口）相关的一组指定行进行计算，支持排名、运行总数和移动平均值（例如 ROW_NUMBER、RANK、LAG、LAG、LEAD）等运算	支持的	支持的	支持的	`SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) AS field2Rank FROM <tableName/logGroup>`
the section called “转换函数”	内置函数，用于在 SQL 查询中将数据从一种类型转换为另一种类型，实现数据类型转换和格式转换（例如，CAST、TO_DATE、TO_TIMESTAMP、BINARY）	支持的	支持的	支持的	`SELECT CAST('123' AS INT) AS converted_number, CAST(123 AS STRING) AS converted_string FROM <tableName/logGroup>`

可用的 SQL 语法	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “谓词函数”	基于指定条件或模式（例如，IN、LIKE、BETWEEN、IS NULL、EXISTS）评估条件并返回布尔值（真/假）的内置函数	支持的	支持的	支持的	`SELECT * FROM <tableName/logGroup> WHERE id BETWEEN 50000 AND 75000`
the section called “地图函数”	将指定的函数应用于集合中的每个元素，将数据转换为一组新的值。	不支持	支持的	支持的	`SELECT MAP_FILTER( MAP( 'method', method, 'status', CAST(status AS STRING), 'bytes', CAST(bytes AS STRING) ), (k, v) -> k IN ('method', 'status') AND v != 'null' ) AS filtered_map FROM <tableName/logGroup> WHERE status = 100`

可用的 SQL 语法	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “数学函数”	对数值数据执行数学运算，例如计算平均值、求和值或三角值。	支持的	支持的	支持的	`SELECT bytes, bytes + 1000 AS added, bytes - 1000 AS subtracted, bytes * 2 AS doubled, bytes / 1024 AS kilobytes, bytes % 1000 AS remainder FROM <tableName/logGroup>`
the section called “多日志组函数”	允许用户在 SQL SELECT 语句中指定多个日志组	支持的	不适用	不适用	`SELECT lg1.Column1, lg1.Column2 FROM `logGroups(logGroupIdentifier: ['LogGroup1', 'LogGroup2'])` AS lg1 WHERE lg1.Column3 = "Success"`
the section called “生成器函数”	创建一个生成值序列的迭代器对象，从而允许在大型数据集中高效使用内存。	不支持	支持的	支持的	`SELECT explode(array(10, 20))`

一般 SQL 限制

将 OpenSearch SQL 与 CloudWatch 日志、Amazon S3 和安全湖一起使用时，必须遵守以下限制。

1. 在 SELECT 语句中只能使用一个 JOIN 操作。
2. 仅支持一个级别的嵌套子查询。
3. 不支持用分号分隔的多个语句查询。
4. 不支持包含相同但仅在大小写上不同的字段名称的查询（例如 field1 和 FIELD1）。

   例如，不支持以下查询：

   ```
   Select AWSAccountId, awsaccountid from LogGroup
   ```

   但是，以下查询是因为两个日志组中的字段名 (@logStream) 相同：

   ```
   Select a.`@logStream`, b.`@logStream` from Table A INNER Join Table B on a.id = b.id
   ```

5. 函数和表达式必须对字段名进行操作，并且是 SELECT 语句的一部分，日志组在 FROM 子句中指定。

   例如，不支持以下查询：

   ```
   SELECT cos(10) FROM LogGroup
   ```

   支持此查询：

   ```
   SELECT cos(field1) FROM LogGroup
   ```

为使用 OpenSearch SQL 的 “ CloudWatch 日志见解” 用户提供的其他信息

CloudWatch 日志支持 OpenSearch Logs Insights 控制台、API 和 CLI 中的 SQL 查询。它支持大多数命令，包括 SELECT、FROM、WHERE、GROUP BY、HAVING、JOINS 和嵌套查询，以及 JSON、数学、字符串和条件函数。但是，L CloudWatch ogs 仅支持读取操作，因此不允许 DDL 或 DML 语句。有关支持的命令和函数的完整列表，请参阅前几节中的表格。

多日志组函数

CloudWatch Logs Insights 支持查询多个日志组的功能。要在 SQL 中解决此用例，可以使用`logGroups`命令。此命令专门用于查询 L CloudWatch ogs Insights 中涉及一个或多个日志组的数据。使用此语法通过在命令中指定多个日志组来查询它们，而不是为每个日志组编写查询并将它们与`UNION`命令组合在一起。

语法：

```
`logGroups(
    logGroupIdentifier: ['LogGroup1','LogGroup2', ...'LogGroupn']
)
```

在此语法中，您最多可以在`logGroupIndentifier`参数中指定 50 个日志组。要引用监控账户中的日志组，请使用 ARNs 而不是`LogGroup`名称。

示例查询：

```
SELECT LG1.Column1, LG1.Column2 from `logGroups(
    logGroupIdentifier: ['LogGroup1', 'LogGroup2']
)` as LG1
WHERE LG1.Column1 = 'ABC'
```

查询日志时，不支持在语`FROM`句后使用涉及多个 CloudWatch 日志组的以下语法：

```
SELECT Column1, Column2 FROM 'LogGroup1', 'LogGroup2', ...'LogGroupn'
WHERE Column1 = 'ABC'
```

限制

使用 SQL 或 PPL 命令时，请将某些字段用反引号括起来以进行查询。带有特殊字符（非字母和非数字）的字段必须使用反引号。例如，用反引号`@message`括`Operation.Export,`起来、和`Test::Field`。您无需用反引号将纯字母名称的列括起来。

包含简单字段的查询示例：

```
SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
LIMIT 1000;
```

同样的查询，但附加了反引号：

```
SELECT `SessionToken`, `Operation`, `StartTime`  FROM `LogGroup-A`
LIMIT 1000;
```

有关不特定于 CloudWatch 日志的其他一般限制，请参阅the section called “一般 SQL 限制”。

示例查询和配额

> **Note**
>
> 以下内容适用于 CloudWatch Logs Insights OpenSearch 用户和查询 CloudWatch 数据的用户。

有关可在 CloudWatch 日志中使用的示例 SQL 查询，请参阅 Amazon CloudWatch Logs Insights 控制台中保存的查询和示例查询以获取示例。

有关查询 OpenSearch 服务 CloudWatch 日志时适用的限制的信息，请参阅 Amazon Logs 用户指南中的CloudWatch CloudWatch 日志配额。限制包括您可以查询的 CloudWatch 日志组数量、可以执行的最大并发查询数、最长查询执行时间以及结果中返回的最大行数。无论您使用哪种语言查询 CloudWatch 日志（即 OpenSearch PPL、SQL 和 Logs Insights），限制都是一样的。

SQL 命令

主题

- 字符串函数
- 日期和时间函数
- 聚合函数
- 条件函数
- JSON 函数
- 数组函数
- 窗口函数
- 转换函数
- 谓词函数
- 地图函数
- 数学函数
- 生成器函数
- SELECT 子句
- WHERE 子句
- GROUP BY 子句
- HAVING 子句

- ORDER BY 子句
- JOIN 子句
- LIMIT 子句
- 案例条款
- 公用表表达式
- EXPLAIN
- 横向子查询子句
- 横向视图条款
- LIKE 谓词
- OFFSET
- PIVOT 子句
- 集合运算符
- 按子句排序
- UNPIVOT

字符串函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
ascii (str)	返回的第一个字符的数值`str`。
base64（垃圾桶）	将参数从二进制`bin`转换为以 64 为基数的字符串。
位长 (expr)	返回字符串数据的位长或二进制数据的位数。
btrim (str)	从中移除前导和尾随空格字符。`str`
btrim（str、trimStr）	从中移除前导和尾随`trimStr`字符。`str`

函数	描述
char (expr)	返回二进制值等同于的 ASCII 字符。`expr`如果 n 大于 256，则结果等同于 chr (n% 256)
char_length (expr)	返回字符串数据的字符长度或二进制数据的字节数。字符串数据的长度包括尾随空格。二进制数据的长度包括二进制零。
字符长度 (expr)	返回字符串数据的字符长度或二进制数据的字节数。字符串数据的长度包括尾随空格。二进制数据的长度包括二进制零。
chr (expr)	返回二进制值等同于的 ASCII 字符。`expr`如果 n 大于 256，则结果等同于 chr (n% 256)
concat_ws (sep [, str | array (str)] +)	返回由分隔的字符串的串联`sep`，跳过空值。
包含（左、右）	返回一个布尔值。如果在左侧找到右侧，则该值为 True。如果任一输入表达式为空，则返回 NULL。否则，返回 False。左边或右边都必须是字符串或二进制类型。
解码（bin、字符集）	使用第二个参数字符集对第一个参数进行解码。
解码（expr、search、result [、search、result]... [，默认]）	按顺序将 expr 与每个搜索值进行比较。如果 expr 等于搜索值，则 decode 将返回相应的结果。如果未找到匹配项，则返回默认值。如果省略默认值，则返回 null。
elt (n、input1、input2、...)	返回`n`第 n 个输入，例如，`input2`当`n`为 2 时返回。
编码（str、字符集）	使用第二个参数字符集对第一个参数进行编码。

函数	描述
结尾为（左、右）	返回一个布尔值。如果左边以右结尾，则该值为 True。如果任一输入表达式为空，则返回 NULL。否则，返回 False。左边或右边都必须是字符串或二进制类型。
find_in_set（str、str_array）	返回逗号分隔列表 () 中给定字符串 (`str`) 的索引（从 1 开始）。`str_array` 如果未找到字符串或给定字符串 (`str`) 包含逗号，则返回 0。
format_number (expr1、expr2)	格式化数字，`expr1`如 '#、###、##.##'，四舍五入到小数位。`expr2`如果`expr2`为 0，则结果没有小数点或小数部分。 `expr2`也接受用户指定的格式。它的功能应该像 MySQL 的格式一样。
format_string（strfmt、obj、...）	从 printf 样式的格式字符串中返回格式化字符串。
initcap (str)	返回`str`时每个单词的首字母均为大写。所有其他字母均为小写。单词由空格分隔。
instr（str、substr）	返回 in 中首次出现的（从 1 开始）`substr`的`str`索引。
lcase (str)	返回`str`时所有字符都改为小写。
左（str、len）	返回字符串中最左边`len`（`len`可以`len`是字符串类型）的字符`str`，如果小于或等于 0，则结果为空字符串。
len (expr)	返回字符串数据的字符长度或二进制数据的字节数。字符串数据的长度包括尾随空格。二进制数据的长度包括二进制零。
长度 (expr)	返回字符串数据的字符长度或二进制数据的字节数。字符串数据的长度包括尾随空格。二进制数据的长度包括二进制零。

函数	描述
levenshtein（str1、str2 [、threshold]）	返回两个给定字符串之间的 Levenshtein 距离。如果设置了阈值且距离大于该阈值，则返回 -1。
定位（substr、str [、pos]）	返回 in `str` 之后第一次`substr`出现的位置`pos`。给定值`pos`和返回值均以 1 为基准。
较低 (str)	返回`str`时所有字符都改为小写。
lpad（str、len [、pad]）	返回`str`，左边填`pad`充长度为。`len`如果`str`长度大于`len`，则返回值将缩短为`len`字符或字节。如果未指定，`str`则如果`pad`是字符串，则在左边填充空格字符；如果是字节序列，则用零填充。
ltrim (str)	从中移除前导空格字符`str`。
luhn_check (str)	根据 Luhn 算法检查数字字符串是否有效。此校验和功能广泛应用于信用卡号和政府识别号，以区分有效号码和输入错误、不正确的数字。
mask（输入 [、upperChar、lowerChar、lower Char、digitChar、O	掩盖给定的字符串值。该函数用 'X' 或 'x' 替换字符，用 'n' 替换数字。这对于创建删除了敏感信息的表副本非常有用。
八位字节长度 (expr)	返回字符串数据的字节长度或二进制数据的字节数。
叠加（输入、替换、pos [、len]）	`input`替换为`replace`，开头为`pos`且长度为`len`。
位置（substr、str [、pos]）	返回 in `str` 之后第一次`substr`出现的位置`pos`。给定值`pos`和返回值均以 1 为基准。
printf（strfmt、obj、...）	从 printf 样式的格式字符串中返回格式化字符串。

函数	描述
regexp_count（str、regexp）	返回字符串中匹配正则表达式模式regexp的次数str。
regexp_extract（str、regexp [、idx]）	提取中与表达式匹配且对应于regexp正则表达式组索引的第一个字符串。str
regexp_extract_all (str、regexp [、idx])	提取中与表达式匹配str且对应于regexp正则表达式组索引的所有字符串。
regexp_instr（str、regexp）	在字符串中搜索正则表达式并返回一个表示匹配子字符串起始位置的整数。位置是从 1 开始的，而不是从 0 开始的。如果未找到匹配项，则返回 0。
regexp_replace（str、regexp、rep [，position]）	将str该匹配项的所有子字符串替换为regexp。rep
regexp_substr (str、regexp)	返回与字符串中的正则表达式regexp相匹配的子字符串str。如果未找到正则表达式，则结果为空。
重复（str，n）	返回重复给定字符串值 n 次的字符串。
替换（str，搜索 [，替换]）	将所有出现的search替换为。replace
右（str、len）	返回字符串中最右边len（len可以len是字符串类型）的字符str，如果小于或等于 0，则结果为空字符串。
rpad（str、len [、pad]）	返回str，右边填pad充长度为。len如果str长度大于len，则返回值将缩短为len个字符。如果未指定，str则如果pad是字符串，则在右边填充空格字符；如果是二进制字符串，则用零填充。
rtrim (str)	从中移除尾随空格字符。str

函数	描述
句子（str [、lang、country]）	拆分str成由单词组成的数组。
soundex (str)	返回字符串的 Soundex 代码。
空格 (n)	返回一个由n空格组成的字符串。
split（str、regex、limit）	str围绕匹配的匹配项进行拆分，regex并返回一个长度最多为的数组 limit
split_part（str、分隔符、partNum）	str按分隔符分割并返回分割中请求的部分（从 1 开始）。如果任何输入为空，则返回 null。如果partNum超出分割部分的范围，则返回空字符串。如果partNum为 0，则引发错误。如果partNum为负数，则从字符串末尾向后计算各个部分。如果delimiter 是空字符串，str则不拆分。
开头为（左、右）	返回一个布尔值。如果左边以右开头，则该值为 True。如果任一输入表达式为空，则返回 NULL。否则，返回 False。左边或右边都必须是字符串或二进制类型。
substr（str、pos [、len]）	返回开头为pos且长度为str的子字符串len，或者字节数组中起始于pos且长度len为的片段。
substr (str FROM pos [FOR len]])	返回开头为pos且长度为str的子字符串len，或者字节数组中起始于pos且长度len为的片段。
子字符串（str、pos [、len]）	返回开头为pos且长度为str的子字符串len，或者字节数组中起始于pos且长度len为的片段。
子字符串（来自 pos 的 str [FOR len]]）	返回开头为pos且长度为str的子字符串len，或者字节数组中起始于pos且长度len为的片段。

函数	描述
substring_index（str、delim、count）	返回分隔符count出现str之前的子字符串。delim如果count为正数，则返回最后一个分隔符左边的所有内容（从左边算起）。如果count为负数，则返回最后一个分隔符右边的所有内容（从右边算起）。substring_index函数在搜索时执行区分大小写的匹配。delim
to_binary (str [, fmt])	根据提供的值将输入str转换为二进制值fmt。fmt可以是 “hex”、“utf-8”、“utf8” 或 “base64” 的不区分大小写的字符串文字。默认情况下，如果fmt省略，则转换的二进制格式为 “十六进制”。如果至少有一个输入参数为 NULL，则该函数返回 NULL。
to_char（numberExpr，formateXPR）	根据. numberExpr 转换为字符串formatExpr 。如果转换失败，则抛出异常。格式可以由以下字符组成，不区分大小写：'0' 或 '9'：指定 0 到 9 之间的预期数字。格式字符串中 0 或 9 的序列与输入值中的数字序列相匹配，生成与格式字符串中相应序列长度相同的结果字符串。如果 0/9 序列包含的数字多于十进制值的匹配部分、以 0 开头且在小数点之前，则结果字符串将用零填充。否则，它会用空格填充。'.' 或 'D'：指定小数点的位置（可选，只允许一次）。'、' 或 'G'：指定分组（千位）分隔符（,）的位置。每个分组分隔符的左侧和右侧必须有 0 或 9。 '

函数	描述
to_number（expr、fmt）	将字符串 “expr” 转换为基于字符串格式 “fmt” 的数字。如果转换失败，则抛出异常。格式可以由以下字符组成，不区分大小写：'0' 或 '9'：指定 0 到 9 之间的预期数字。格式字符串中的 0 或 9 序列与输入字符串中的数字序列相匹配。如果 0/9 序列以 0 开头且在小数点之前，则它只能匹配大小相同的数字序列。否则，如果序列以 9 开头或小数点之后，则它可以匹配大小相同或更小的数字序列。'.' 或 'D'：指定小数点的位置（可选，只允许一次）。'、' 或 'G'：指定分组（千位）分隔符 (,) 的位置。每个分组分隔符的左侧和右侧必须有 0 或 9。 'expr'必须匹配与数字大小相关的分组分隔符。 '
to_varchar（numberExpr，formateXPR）	根据. `numberExpr` 转换为字符串`formatExpr` 。如果转换失败，则抛出异常。格式可以由以下字符组成，不区分大小写：'0' 或 '9'：指定 0 到 9 之间的预期数字。格式字符串中 0 或 9 的序列与输入值中的数字序列相匹配，生成与格式字符串中相应序列长度相同的结果字符串。如果 0/9 序列包含的数字多于十进制值的匹配部分、以 0 开头且在小数点之前，则结果字符串将用零填充。否则，它会用空格填充。'.' 或 'D'：指定小数点的位置（可选，只允许一次）。'、' 或 'G'：指定分组（千位）分隔符（,）的位置。每个分组分隔符的左侧和右侧必须有 0 或 9。 '
翻译（输入、从、到）	通过将`input`字符串中存在的字符替换为`from`字符串中的相应字符来翻译`to`字符串。
修剪 (str)	从中移除前导和尾随空格字符。`str`
修剪（均来自 str）	从中移除前导和尾随空格字符。`str`
修剪（从 str 开始）	从中移除前导空格字符`str`。

函数	描述
修剪（从 str 尾随而来）	从中移除尾随空格字符。`str`
trim（来自 str 的 trimStr）	从中移除前导和尾随`trimStr`字符。`str`
trim（都来自 str 的 trimStr）	从中移除前导和尾随`trimStr`字符。`str`
trim（来自 str 的前导 trimStr）	从中删除前导`trimStr`字符`str`。
trim（来自 str 的尾部 trimStr）	从中删除尾随`trimStr`字符。`str`
try_to_binary (str [, fmt])	这是执行相同操作的`to_binary` 特殊版本，但如果无法执行转换，则返回空值而不是引发错误。
try_to_number（expr、fmt）	根据字符串格式`fmt`将字符串 'expr' 转换为数字。如果字符串 'expr' 与预期格式不匹配，则返回 NULL。该格式遵循与 to_number 函数相同的语义。
ucase（star）	返回`str`时所有字符都改为大写。
unbase64 (str)	将参数从以 64 为基数的字符串`str`转换为二进制。
上限 (str)	返回`str`时所有字符都改为大写。

示例

```
-- ascii
SELECT ascii('222');
+----------+
|ascii(222)|
+----------+
|        50|
+----------+
SELECT ascii(2);
+--------+
|ascii(2)|
```

```
+--------+
|      50|
+--------+
-- base64
SELECT base64('Feathers');
+-----------------+
|base64(Feathers)|
+-----------------+
|     RmVhdGhlcnM=|
+-----------------+
SELECT base64(x'537061726b2053514c');
+-----------------------------+
|base64(X'537061726B2053514C')|
+-----------------------------+
|                 U3BhcmsgU1FM|
+-----------------------------+
-- bit_length
SELECT bit_length('Feathers');
+---------------------+
|bit_length(Feathers)|
+---------------------+
|                   64|
+---------------------+
SELECT bit_length(x'537061726b2053514c');
+---------------------------------+
|bit_length(X'537061726B2053514C')|
+---------------------------------+
|                               72|
+---------------------------------+
-- btrim
SELECT btrim('    Feathers   ');
+----------------------+
|btrim(    Feathers   )|
+----------------------+
|              Feathers|
+----------------------+
SELECT btrim(encode('    Feathers   ', 'utf-8'));
+-------------------------------------+
|btrim(encode(    Feathers   , utf-8))|
+-------------------------------------+
|                             Feathers|
+-------------------------------------+
SELECT btrim('Feathers', 'Fe');
+---------------------+
```

```
|btrim(Alphabet, Al)|
+---------------------+
|               athers|
+---------------------+
SELECT btrim(encode('Feathers', 'utf-8'), encode('Al', 'utf-8'));
+----------------------------------------------------+
|btrim(encode(Feathers, utf-8), encode(Al, utf-8))|
+----------------------------------------------------+
|                                             athers|
+----------------------------------------------------+
-- char
SELECT char(65);
+--------+
|char(65)|
+--------+
|       A|
+--------+
-- char_length
SELECT char_length('Feathers ');
+------------------------+
|char_length(Feathers )|
+------------------------+
|                      9 |
+------------------------+
SELECT char_length(x'537061726b2053514c');
+----------------------------------+
|char_length(X'537061726B2053514C')|
+----------------------------------+
|                                 9|
+----------------------------------+
SELECT CHAR_LENGTH('Feathers ');
+------------------------+
|char_length(Feathers )|
+------------------------+
|                      9|
+------------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+----------------------------+
|character_length(Feathers )|
+----------------------------+
|                           9|
+----------------------------+
-- character_length
SELECT character_length('Feathers ');
```

```
+----------------------------+
|character_length(Feathers )|
+----------------------------+
|                          9|
+----------------------------+
SELECT character_length(x'537061726b2053514c');
+----------------------------------------+
|character_length(X'537061726B2053514C')|
+----------------------------------------+
|                                       9|
+----------------------------------------+
SELECT CHAR_LENGTH('Feathers ');
+-----------------------+
|char_length(Feathers )|
+-----------------------+
|                      9|
+-----------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+----------------------------+
|character_length(Feathers )|
+----------------------------+
|                          9|
+----------------------------+
-- chr
SELECT chr(65);
+-------+
|chr(65)|
+-------+
|      A|
+-------+
-- concat_ws
SELECT concat_ws(' ', 'Fea', 'thers');
+------------------------+
|concat_ws( , Fea, thers)|
+------------------------+
|               Feathers|
+------------------------+
SELECT concat_ws('s');
+------------+
|concat_ws(s)|
+------------+
|            |
+------------+
SELECT concat_ws('/', 'foo', null, 'bar');
```

```
+----------------------------+
|concat_ws(/, foo, NULL, bar)|
+----------------------------+
|                     foo/bar|
+----------------------------+
SELECT concat_ws(null, 'Fea', 'thers');
+---------------------------+
|concat_ws(NULL, Fea, thers)|
+---------------------------+
|                       NULL|
+---------------------------+
-- contains
SELECT contains('Feathers', 'Fea');
+--------------------------+
|contains(Feathers, Fea)|
+--------------------------+
|                      true|
+--------------------------+
SELECT contains('Feathers', 'SQL');
+--------------------------+
|contains(Feathers, SQL)|
+--------------------------+
|                     false|
+--------------------------+
SELECT contains('Feathers', null);
+-------------------------+
|contains(Feathers, NULL)|
+-------------------------+
|                     NULL|
+-------------------------+
SELECT contains(x'537061726b2053514c', x'537061726b');
+-----------------------------------------------+
|contains(X'537061726B2053514C', X'537061726B')|
+-----------------------------------------------+
|                                           true|
+-----------------------------------------------+
-- decode
SELECT decode(encode('abc', 'utf-8'), 'utf-8');
+---------------------------------+
|decode(encode(abc, utf-8), utf-8)|
+---------------------------------+
|                              abc|
+---------------------------------+
```

```
SELECT decode(2, 1, 'Southlake', 2, 'San Francisco', 3, 'New Jersey', 4, 'Seattle',
 'Non domestic');
+-------------------------------------------------------------------------------+
|decode(2, 1, Southlake, 2, San Francisco, 3, New Jersey, 4, Seattle, Non domestic)|
+-------------------------------------------------------------------------------+
|                                                                   San Francisco|
+-------------------------------------------------------------------------------+
SELECT decode(6, 1, 'Southlake', 2, 'San Francisco', 3, 'New Jersey', 4, 'Seattle',
 'Non domestic');
+-------------------------------------------------------------------------------+
|decode(6, 1, Southlake, 2, San Francisco, 3, New Jersey, 4, Seattle, Non domestic)|
+-------------------------------------------------------------------------------+
|                                                                    Non domestic|
+-------------------------------------------------------------------------------+
SELECT decode(6, 1, 'Southlake', 2, 'San Francisco', 3, 'New Jersey', 4, 'Seattle');
+-------------------------------------------------------------------+
|decode(6, 1, Southlake, 2, San Francisco, 3, New Jersey, 4, Seattle)|
+-------------------------------------------------------------------+
|                                                               NULL|
+-------------------------------------------------------------------+
SELECT decode(null, 6, 'Fea', NULL, 'thers', 4, 'rock');
+-------------------------------------------+
|decode(NULL, 6, Fea, NULL, thers, 4, rock)|
+-------------------------------------------+
|                                     thers|
+-------------------------------------------+
-- elt
SELECT elt(1, 'scala', 'java');
+-------------------+
|elt(1, scala, java)|
+-------------------+
|              scala|
+-------------------+
SELECT elt(2, 'a', 1);
+------------+
|elt(2, a, 1)|
+------------+
|           1|
+------------+
-- encode
SELECT encode('abc', 'utf-8');
+------------------+
|encode(abc, utf-8)|
+------------------+
```

```
|        [61 62 63]|
+------------------+
-- endswith
SELECT endswith('Feathers', 'ers');
+------------------------+
|endswith(Feathers, ers)|
+------------------------+
|                    true|
+------------------------+
SELECT endswith('Feathers', 'SQL');
+--------------------------+
|endswith(Feathers, SQL)|
+--------------------------+
|                     false|
+--------------------------+
SELECT endswith('Feathers', null);
+-------------------------+
|endswith(Feathers, NULL)|
+-------------------------+
|                     NULL|
+-------------------------+
SELECT endswith(x'537061726b2053514c', x'537061726b');
+-----------------------------------------------+
|endswith(X'537061726B2053514C', X'537061726B')|
+-----------------------------------------------+
|                                          false|
+-----------------------------------------------+
SELECT endswith(x'537061726b2053514c', x'53514c');
+-------------------------------------------+
|endswith(X'537061726B2053514C', X'53514C')|
+-------------------------------------------+
|                                       true|
+-------------------------------------------+
-- find_in_set
SELECT find_in_set('ab','abc,b,ab,c,def');
+-------------------------------+
|find_in_set(ab, abc,b,ab,c,def)|
+-------------------------------+
|                              3|
+-------------------------------+
-- format_number
SELECT format_number(12332.123456, 4);
+------------------------------+
|format_number(12332.123456, 4)|
```

```
+-----------------------------+
|                   12,332.1235|
+-----------------------------+
SELECT format_number(12332.123456, '##################.###');
+----------------------------------------------------+
|format_number(12332.123456, ##################.###)|
+----------------------------------------------------+
|                                           12332.123|
+----------------------------------------------------+
-- format_string
SELECT format_string("Hello World %d %s", 100, "days");
+--------------------------------------------+
|format_string(Hello World %d %s, 100, days)|
+--------------------------------------------+
|                       Hello World 100 days|
+--------------------------------------------+
-- initcap
SELECT initcap('Feathers');
+------------------+
|initcap(Feathers)|
+------------------+
|         Feathers|
+------------------+
-- instr
SELECT instr('Feathers', 'ers');
+--------------------+
|instr(Feathers, ers)|
+--------------------+
|                   6|
+--------------------+
-- lcase
SELECT lcase('Feathers');
+---------------+
|lcase(Feathers)|
+---------------+
|       feathers|
+---------------+
-- left
SELECT left('Feathers', 3);
+------------------+
|left(Feathers, 3)|
+------------------+
|               Fea|
+------------------+
```

```
SELECT left(encode('Feathers', 'utf-8'), 3);
+---------------------------------+
|left(encode(Feathers, utf-8), 3)|
+---------------------------------+
|                       [RmVh]|
+---------------------------------+
-- len
SELECT len('Feathers ');
+---------------+
|len(Feathers )|
+---------------+
|             9|
+---------------+
SELECT len(x'537061726b2053514c');
+--------------------------+
|len(X'537061726B2053514C')|
+--------------------------+
|                         9|
+--------------------------+
SELECT CHAR_LENGTH('Feathers ');
+-----------------------+
|char_length(Feathers )|
+-----------------------+
|                     9|
+-----------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+----------------------------+
|character_length(Feathers )|
+----------------------------+
|                           9|
+----------------------------+
-- length
SELECT length('Feathers ');
+------------------+
|length(Feathers )|
+------------------+
|                9|
+------------------+
SELECT length(x'537061726b2053514c');
+-----------------------------+
|length(X'537061726B2053514C')|
+-----------------------------+
|                            9|
+-----------------------------+
```

```
SELECT CHAR_LENGTH('Feathers ');
+-----------------------+
|char_length(Feathers )|
+-----------------------+
|                     9|
+-----------------------+
SELECT CHARACTER_LENGTH('Feathers ');
+----------------------------+
|character_length(Feathers )|
+----------------------------+
|                           9|
+----------------------------+
-- levenshtein
SELECT levenshtein('kitten', 'sitting');
+----------------------------+
|levenshtein(kitten, sitting)|
+----------------------------+
|                           3|
+----------------------------+
SELECT levenshtein('kitten', 'sitting', 2);
+-------------------------------+
|levenshtein(kitten, sitting, 2)|
+-------------------------------+
|                             -1|
+-------------------------------+
-- locate
SELECT locate('bar', 'foobarbar');
+-------------------------+
|locate(bar, foobarbar, 1)|
+-------------------------+
|                        4|
+-------------------------+
SELECT locate('bar', 'foobarbar', 5);
+-------------------------+
|locate(bar, foobarbar, 5)|
+-------------------------+
|                        7|
+-------------------------+
SELECT POSITION('bar' IN 'foobarbar');
+-------------------------+
|locate(bar, foobarbar, 1)|
+-------------------------+
|                        4|
+-------------------------+
```

```
-- lower
SELECT lower('Feathers');
+---------------+
|lower(Feathers)|
+---------------+
|       feathers|
+---------------+
-- lpad
SELECT lpad('hi', 5, '??');
+---------------+
|lpad(hi, 5, ??)|
+---------------+
|          ???hi|
+---------------+
SELECT lpad('hi', 1, '??');
+---------------+
|lpad(hi, 1, ??)|
+---------------+
|              h|
+---------------+
SELECT lpad('hi', 5);
+--------------+
|lpad(hi, 5,  )|
+--------------+
|            hi|
+--------------+
SELECT hex(lpad(unhex('aabb'), 5));
+--------------------------------+
|hex(lpad(unhex(aabb), 5, X'00'))|
+--------------------------------+
|                     000000AABB|
+--------------------------------+
SELECT hex(lpad(unhex('aabb'), 5, unhex('1122')));
+--------------------------------------+
|hex(lpad(unhex(aabb), 5, unhex(1122)))|
+--------------------------------------+
|                           112211AABB|
+--------------------------------------+
-- ltrim
SELECT ltrim('    Feathers   ');
+----------------------+
|ltrim(    Feathers   )|
+----------------------+
|           Feathers   |
```

```
+----------------------+
-- luhn_check
SELECT luhn_check('8112189876');
+----------------------+
|luhn_check(8112189876)|
+----------------------+
|                  true|
+----------------------+
SELECT luhn_check('79927398713');
+-----------------------+
|luhn_check(79927398713)|
+-----------------------+
|                   true|
+-----------------------+
SELECT luhn_check('79927398714');
+-----------------------+
|luhn_check(79927398714)|
+-----------------------+
|                  false|
+-----------------------+
-- mask
SELECT mask('abcd-EFGH-8765-4321');
+----------------------------------------+
|mask(abcd-EFGH-8765-4321, X, x, n, NULL)|
+----------------------------------------+
|                     xxxx-XXXX-nnnn-nnnn|
+----------------------------------------+
SELECT mask('abcd-EFGH-8765-4321', 'Q');
+----------------------------------------+
|mask(abcd-EFGH-8765-4321, Q, x, n, NULL)|
+----------------------------------------+
|                     xxxx-QQQQ-nnnn-nnnn|
+----------------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q');
+--------------------------------+
|mask(AbCD123-@$#, Q, q, n, NULL)|
+--------------------------------+
|                     QqQQnnn-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#');
+--------------------------------+
|mask(AbCD123-@$#, X, x, n, NULL)|
+--------------------------------+
|                     XxXXnnn-@$#|
```

```
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q');
+--------------------------------+
|mask(AbCD123-@$#, Q, x, n, NULL)|
+--------------------------------+
|                     QxQQnnn-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q');
+--------------------------------+
|mask(AbCD123-@$#, Q, q, n, NULL)|
+--------------------------------+
|                     QqQQnnn-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q', 'd');
+--------------------------------+
|mask(AbCD123-@$#, Q, q, d, NULL)|
+--------------------------------+
|                     QqQQddd-@$#|
+--------------------------------+
SELECT mask('AbCD123-@$#', 'Q', 'q', 'd', 'o');
+-----------------------------+
|mask(AbCD123-@$#, Q, q, d, o)|
+-----------------------------+
|                  QqQQdddoooo|
+-----------------------------+
SELECT mask('AbCD123-@$#', NULL, 'q', 'd', 'o');
+--------------------------------+
|mask(AbCD123-@$#, NULL, q, d, o)|
+--------------------------------+
|                     AqCDdddoooo|
+--------------------------------+
SELECT mask('AbCD123-@$#', NULL, NULL, 'd', 'o');
+-----------------------------------+
|mask(AbCD123-@$#, NULL, NULL, d, o)|
+-----------------------------------+
|                        AbCDdddoooo|
+-----------------------------------+
SELECT mask('AbCD123-@$#', NULL, NULL, NULL, 'o');
+--------------------------------------+
|mask(AbCD123-@$#, NULL, NULL, NULL, o)|
+--------------------------------------+
|                           AbCD123oooo|
+--------------------------------------+
SELECT mask(NULL, NULL, NULL, NULL, 'o');
```

```
+-------------------------------+
|mask(NULL, NULL, NULL, NULL, o)|
+-------------------------------+
|                           NULL|
+-------------------------------+
SELECT mask(NULL);
+-------------------------+
|mask(NULL, X, x, n, NULL)|
+-------------------------+
|                     NULL|
+-------------------------+
SELECT mask('AbCD123-@$#', NULL, NULL, NULL, NULL);
+----------------------------------------+
|mask(AbCD123-@$#, NULL, NULL, NULL, NULL)|
+----------------------------------------+
|                              AbCD123-@$#|
+----------------------------------------+
-- octet_length
SELECT octet_length('Feathers');
+-----------------------+
|octet_length(Feathers)|
+-----------------------+
|                      8|
+-----------------------+
SELECT octet_length(x'537061726b2053514c');
+-----------------------------------+
|octet_length(X'537061726B2053514C')|
+-----------------------------------+
|                                  9|
+-----------------------------------+
-- overlay
SELECT overlay('Feathers' PLACING '_' FROM 6);
+----------------------------+
|overlay(Feathers, _, 6, -1)|
+----------------------------+
|                  Feathe_ers|
+----------------------------+
SELECT overlay('Feathers' PLACING 'ures' FROM 5);
+------------------------------+
|overlay(Feathers, ures, 5, -1)|
+------------------------------+
|                     Features  |
+------------------------------+
-- position
```

```
SELECT position('bar', 'foobarbar');
+---------------------------+
|position(bar, foobarbar, 1)|
+---------------------------+
|                          4|
+---------------------------+
SELECT position('bar', 'foobarbar', 5);
+---------------------------+
|position(bar, foobarbar, 5)|
+---------------------------+
|                          7|
+---------------------------+
SELECT POSITION('bar' IN 'foobarbar');
+-------------------------+
|locate(bar, foobarbar, 1)|
+-------------------------+
|                        4|
+-------------------------+
-- printf
SELECT printf("Hello World %d %s", 100, "days");
+------------------------------------+
|printf(Hello World %d %s, 100, days)|
+------------------------------------+
|                Hello World 100 days|
+------------------------------------+
-- regexp_count
SELECT regexp_count('Steven Jones and Stephen Smith are the best players', 'Ste(v|
ph)en');
+-------------------------------------------------------------------------------+
|regexp_count(Steven Jones and Stephen Smith are the best players, Ste(v|ph)en)|
+-------------------------------------------------------------------------------+
|                                                                              2|
+-------------------------------------------------------------------------------+
SELECT regexp_count('abcdefghijklmnopqrstuvwxyz', '[a-z]{3}');
+---------------------------------------------------+
|regexp_count(abcdefghijklmnopqrstuvwxyz, [a-z]{3})|
+---------------------------------------------------+
|                                                  8|
+---------------------------------------------------+
-- regexp_extract
SELECT regexp_extract('100-200', '(\\d+)-(\\d+)', 1);
+---------------------------------------+
|regexp_extract(100-200, (\d+)-(\d+), 1)|
+---------------------------------------+
```

```
|                                    100|
+---------------------------------------+
-- regexp_extract_all
SELECT regexp_extract_all('100-200, 300-400', '(\\d+)-(\\d+)', 1);
+-----------------------------------------------------+
|regexp_extract_all(100-200, 300-400, (\d+)-(\d+), 1)|
+-----------------------------------------------------+
|                                           [100, 300]|
+-----------------------------------------------------+
-- regexp_instr
SELECT regexp_instr('user@opensearch.org', '@[^.]*');
+-----------------------------------------------+
|regexp_instr(user@opensearch.org, @[^.]*, 0)|
+-----------------------------------------------+
|                                              5|
+-----------------------------------------------+
-- regexp_replace
SELECT regexp_replace('100-200', '(\\d+)', 'num');
+--------------------------------------+
|regexp_replace(100-200, (\d+), num, 1)|
+--------------------------------------+
|                               num-num|
+--------------------------------------+
-- regexp_substr
SELECT regexp_substr('Steven Jones and Stephen Smith are the best players', 'Ste(v|
ph)en');
+-------------------------------------------------------------------------------+
|regexp_substr(Steven Jones and Stephen Smith are the best players, Ste(v|ph)en)|
+-------------------------------------------------------------------------------+
|                                                                         Steven|
+-------------------------------------------------------------------------------+
SELECT regexp_substr('Steven Jones and Stephen Smith are the best players', 'Jeck');
+-------------------------------------------------------------------------+
|regexp_substr(Steven Jones and Stephen Smith are the best players, Jeck)|
+-------------------------------------------------------------------------+
|                                                                     NULL|
+-------------------------------------------------------------------------+
-- repeat
SELECT repeat('123', 2);
+--------------+
|repeat(123, 2)|
+--------------+
|        123123|
+--------------+
```

```
-- replace
SELECT replace('ABCabc', 'abc', 'DEF');
+-------------------------+
|replace(ABCabc, abc, DEF)|
+-------------------------+
|                   ABCDEF|
+-------------------------+
-- right
SELECT right('Feathers', 3);
+--------------------+
|right(Feathers, 3)|
+--------------------+
|                ers|
+--------------------+
-- rpad
SELECT rpad('hi', 5, '??');
+---------------+
|rpad(hi, 5, ??)|
+---------------+
|          hi???|
+---------------+
SELECT rpad('hi', 1, '??');
+---------------+
|rpad(hi, 1, ??)|
+---------------+
|              h|
+---------------+
SELECT rpad('hi', 5);
+--------------+
|rpad(hi, 5,  )|
+--------------+
|         hi   |
+--------------+
SELECT hex(rpad(unhex('aabb'), 5));
+--------------------------------+
|hex(rpad(unhex(aabb), 5, X'00'))|
+--------------------------------+
|                     AABB000000|
+--------------------------------+
SELECT hex(rpad(unhex('aabb'), 5, unhex('1122')));
+--------------------------------------+
|hex(rpad(unhex(aabb), 5, unhex(1122)))|
+--------------------------------------+
|                           AABB112211|
```

```
+-------------------------------------+
-- rtrim
SELECT rtrim('    Feathers   ');
+----------------------+
|rtrim(    Feathers   )|
+----------------------+
|              Feathers|
+----------------------+
-- sentences
SELECT sentences('Hi there! Good morning.');
+-------------------------------------+
|sentences(Hi there! Good morning., , )|
+-------------------------------------+
|                  [[Hi, there], [Go...|
+-------------------------------------+
-- soundex
SELECT soundex('Miller');
+---------------+
|soundex(Miller)|
+---------------+
|           M460|
+---------------+
-- space
SELECT concat(space(2), '1');
+-------------------+
|concat(space(2), 1)|
+-------------------+
|                  1|
+-------------------+
-- split
SELECT split('oneAtwoBthreeC', '[ABC]');
+--------------------------------+
|split(oneAtwoBthreeC, [ABC], -1)|
+--------------------------------+
|             [one, two, three, ]|
+--------------------------------+
SELECT split('oneAtwoBthreeC', '[ABC]', -1);
+--------------------------------+
|split(oneAtwoBthreeC, [ABC], -1)|
+--------------------------------+
|             [one, two, three, ]|
+--------------------------------+
SELECT split('oneAtwoBthreeC', '[ABC]', 2);
+-------------------------------+
```

```
|split(oneAtwoBthreeC, [ABC], 2)|
+-------------------------------+
|              [one, twoBthreeC]|
+-------------------------------+
-- split_part
SELECT split_part('11.12.13', '.', 3);
+--------------------------+
|split_part(11.12.13, ., 3)|
+--------------------------+
|                        13|
+--------------------------+
-- startswith
SELECT startswith('Feathers', 'Fea');
+----------------------------+
|startswith(Feathers, Fea)|
+----------------------------+
|                        true|
+----------------------------+
SELECT startswith('Feathers', 'SQL');
+--------------------------+
|startswith(Feathers, SQL)|
+--------------------------+
|                     false|
+--------------------------+
SELECT startswith('Feathers', null);
+---------------------------+
|startswith(Feathers, NULL)|
+---------------------------+
|                       NULL|
+---------------------------+
SELECT startswith(x'537061726b2053514c', x'537061726b');
+-------------------------------------------------+
|startswith(X'537061726B2053514C', X'537061726B')|
+-------------------------------------------------+
|                                             true|
+-------------------------------------------------+
SELECT startswith(x'537061726b2053514c', x'53514c');
+---------------------------------------------+
|startswith(X'537061726B2053514C', X'53514C')|
+---------------------------------------------+
|                                        false|
+---------------------------------------------+
-- substr
SELECT substr('Feathers', 5);
```

```
+--------------------------------+
|substr(Feathers, 5, 2147483647)|
+--------------------------------+
|                          hers |
+--------------------------------+
SELECT substr('Feathers', -3);
+---------------------------------+
|substr(Feathers, -3, 2147483647)|
+---------------------------------+
|                             ers|
+---------------------------------+
SELECT substr('Feathers', 5, 1);
+-----------------------+
|substr(Feathers, 5, 1)|
+-----------------------+
|                      h|
+-----------------------+
SELECT substr('Feathers' FROM 5);
+-----------------------------------+
|substring(Feathers, 5, 2147483647)|
+-----------------------------------+
|                             hers |
+-----------------------------------+
SELECT substr('Feathers' FROM -3);
+------------------------------------+
|substring(Feathers, -3, 2147483647)|
+------------------------------------+
|                                ers|
+------------------------------------+
SELECT substr('Feathers' FROM 5 FOR 1);
+--------------------------+
|substring(Feathers, 5, 1)|
+--------------------------+
|                         h|
+--------------------------+
-- substring
SELECT substring('Feathers', 5);
+-----------------------------------+
|substring(Feathers, 5, 2147483647)|
+-----------------------------------+
|                             hers |
+-----------------------------------+
SELECT substring('Feathers', -3);
+------------------------------------+
```

```
|substring(Feathers, -3, 2147483647)|
+-----------------------------------+
|                                ers|
+-----------------------------------+
SELECT substring('Feathers', 5, 1);
+--------------------------+
|substring(Feathers, 5, 1)|
+--------------------------+
|                         h|
+--------------------------+
SELECT substring('Feathers' FROM 5);
+-----------------------------------+
|substring(Feathers, 5, 2147483647)|
+-----------------------------------+
|                              hers |
+-----------------------------------+
SELECT substring('Feathers' FROM -3);
+-----------------------------------+
|substring(Feathers, -3, 2147483647)|
+-----------------------------------+
|                                ers|
+-----------------------------------+
SELECT substring('Feathers' FROM 5 FOR 1);
+--------------------------+
|substring(Feathers, 5, 1)|
+--------------------------+
|                         h|
+--------------------------+
-- substring_index
SELECT substring_index('www.apache.org', '.', 2);
+-------------------------------------+
|substring_index(www.apache.org, ., 2)|
+-------------------------------------+
|                           www.apache|
+-------------------------------------+
-- to_binary
SELECT to_binary('abc', 'utf-8');
+---------------------+
|to_binary(abc, utf-8)|
+---------------------+
|           [61 62 63]|
+---------------------+
-- to_char
SELECT to_char(454, '999');
```

```
+-----------------+
|to_char(454, 999)|
+-----------------+
|              454|
+-----------------+
SELECT to_char(454.00, '000D00');
+-----------------------+
|to_char(454.00, 000D00)|
+-----------------------+
|                 454.00|
+-----------------------+
SELECT to_char(12454, '99G999');
+----------------------+
|to_char(12454, 99G999)|
+----------------------+
|                12,454|
+----------------------+
SELECT to_char(78.12, '$99.99');
+----------------------+
|to_char(78.12, $99.99)|
+----------------------+
|                $78.12|
+----------------------+
SELECT to_char(-12454.8, '99G999D9S');
+----------------------------+
|to_char(-12454.8, 99G999D9S)|
+----------------------------+
|                   12,454.8-|
+----------------------------+
-- to_number
SELECT to_number('454', '999');
+-------------------+
|to_number(454, 999)|
+-------------------+
|                454|
+-------------------+
SELECT to_number('454.00', '000.00');
+-------------------------+
|to_number(454.00, 000.00)|
+-------------------------+
|                   454.00|
+-------------------------+
SELECT to_number('12,454', '99,999');
+-------------------------+
```

```
|to_number(12,454, 99,999)|
+-------------------------+
|                    12454|
+-------------------------+
SELECT to_number('$78.12', '$99.99');
+-------------------------+
|to_number($78.12, $99.99)|
+-------------------------+
|                    78.12|
+-------------------------+
SELECT to_number('12,454.8-', '99,999.9S');
+-------------------------------+
|to_number(12,454.8-, 99,999.9S)|
+-------------------------------+
|                       -12454.8|
+-------------------------------+
-- to_varchar
SELECT to_varchar(454, '999');
+-----------------+
|to_char(454, 999)|
+-----------------+
|              454|
+-----------------+
SELECT to_varchar(454.00, '000D00');
+-----------------------+
|to_char(454.00, 000D00)|
+-----------------------+
|                 454.00|
+-----------------------+
SELECT to_varchar(12454, '99G999');
+----------------------+
|to_char(12454, 99G999)|
+----------------------+
|                12,454|
+----------------------+
SELECT to_varchar(78.12, '$99.99');
+----------------------+
|to_char(78.12, $99.99)|
+----------------------+
|                $78.12|
+----------------------+
SELECT to_varchar(-12454.8, '99G999D9S');
+----------------------------+
|to_char(-12454.8, 99G999D9S)|
```

```
+----------------------------+
|                   12,454.8-|
+----------------------------+
-- translate
SELECT translate('AaBbCc', 'abc', '123');
+---------------------------+
|translate(AaBbCc, abc, 123)|
+---------------------------+
|                     A1B2C3|
+---------------------------+
-- try_to_binary
SELECT try_to_binary('abc', 'utf-8');
+-------------------------+
|try_to_binary(abc, utf-8)|
+-------------------------+
|               [61 62 63]|
+-------------------------+
select try_to_binary('a!', 'base64');
+-------------------------+
|try_to_binary(a!, base64)|
+-------------------------+
|                     NULL|
+-------------------------+
select try_to_binary('abc', 'invalidFormat');
+---------------------------------+
|try_to_binary(abc, invalidFormat)|
+---------------------------------+
|                             NULL|
+---------------------------------+
-- try_to_number
SELECT try_to_number('454', '999');
+-----------------------+
|try_to_number(454, 999)|
+-----------------------+
|                    454|
+-----------------------+
SELECT try_to_number('454.00', '000.00');
+----------------------------+
|try_to_number(454.00, 000.00)|
+----------------------------+
|                      454.00|
+----------------------------+
SELECT try_to_number('12,454', '99,999');
+----------------------------+
```

```
|try_to_number(12,454, 99,999)|
+-----------------------------+
|                        12454|
+-----------------------------+
SELECT try_to_number('$78.12', '$99.99');
+-----------------------------+
|try_to_number($78.12, $99.99)|
+-----------------------------+
|                        78.12|
+-----------------------------+
SELECT try_to_number('12,454.8-', '99,999.9S');
+-----------------------------------+
|try_to_number(12,454.8-, 99,999.9S)|
+-----------------------------------+
|                           -12454.8|
+-----------------------------------+
-- ucase
SELECT ucase('Feathers');
+---------------+
|ucase(Feathers)|
+---------------+
|       FEATHERS|
+---------------+
-- unbase64
SELECT unbase64('U3BhcmsgU1FM');
+----------------------+
|unbase64(U3BhcmsgU1FM)|
+----------------------+
|  [53 70 61 72 6B 2...|
+----------------------+
-- upper
SELECT upper('Feathers');
+---------------+
|upper(Feathers)|
+---------------+
|       FEATHERS|
+---------------+
```

日期和时间函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
添加月（开始日期，数字_月）	返回之`num_months` 后的日期`start_date` 。
转换时区（[SourceTZ、] targetTZ、SourceTS）	将不带时区的时间戳`sourceTs`从`sourceTz`时区转换为。`targetTz`
curdate ()	返回查询评估开始时的当前日期。在同一个查询中对 curdate 的所有调用都返回相同的值。
当前日期 ()	返回查询评估开始时的当前日期。在同一个查询中对 current_date 的所有调用都会返回相同的值。
当前日期	返回查询评估开始时的当前日期。
当前时间戳 ()	返回查询评估开始时的当前时间戳。在同一个查询中对 current_timestamp 的所有调用都会返回相同的值。
当前时间戳	返回查询评估开始时的当前时间戳。
当前时区 ()	返回当前会话的本地时区。
日期_add（开始日期、天数）	返回之`num_days`后的日期`start_date` 。
date_diff（结束日期、开始日期）	返回从`startDate` 到的天数`endDate`。
日期格式（时间戳，fmt）	`timestamp` 以日期格式指定的格式转换为字符串值`fmt`。

函数	描述
date_from_unix_date（天）	根据自 1970 年 1 月 1 日以来的天数创建日期。
日期部分（字段、来源）	提取 date/timestamp 或间隔源的一部分。
date_sub（起始日期，天数）	返回`num_days`之前的日期`start_date` 。
date_trunc (fmt, ts)	返回`ts`截断为格式模型指定的单位的时间戳。`fmt`
dateadd（开始日期、天数）	返回之`num_days`后的日期`start_date` 。
datediff（结束日期、开始日期）	返回从`startDate` 到的天数`endDate`。
日期部分（字段、来源）	提取 date/timestamp 或间隔源的一部分。
日（日期）	返回日期/时间戳的月份中的某一天。
一个月中的某天（日期）	返回日期/时间戳的月份中的某一天。
一周中的某一天（日期）	返回 date/timestamp (1 = 星期日，2 = 星期一，...，7 = 星期六) 的一周中的某一天。
一年中的某一天（日期）	返回日期/时间戳中的某一年。
提取（来自源的字段）	提取 date/timestamp 或间隔源的一部分。
from_unixtime (unix_time [, fmt])	以指定的`unix_time` 形式返回`fmt`。
from_utc_timestamp（时间戳、时区）	给定一个像 '2017-07-14 02:40:00.0 '这样的时间戳，将其解释为UTC时间，并将该时间渲染为给定时区的时间戳。例如，“GMT+1” 将产生 “2017-07-14 03:40:00.0”。
小时（时间戳）	返回字符串/时间戳的小时部分。
最后一天（日期）	返回该日期所属月份的最后一天。

函数	描述
本地时间戳 ()	返回查询评估开始时不带时区的当前时间戳。在同一个查询中对 localtimestamp 的所有调用都会返回相同的值。
本地时间戳	返回查询评估开始时会话时区的当前本地日期时间。
make_date（年、月、日）	根据年、月和日字段创建日期。
make_dt_interval ([天 [, 小时 [, 分钟 [, 秒]]]]])	以天、小时、分钟和秒为单位制作 DayTimeIntervalType 持续时间。
make_interval（[年 [，月 [，周 [，天 [，小时 [，分钟 [，秒]]]]]]]]]]]]）	按年、月、周、天、小时、分钟和秒进行间隔。
make_timestamp（年、月、日、小时、分钟、秒 [，时区]）	根据年、月、日、小时、分钟、秒和时区字段创建时间戳。
make_timestamp_ltz（年、月、日、小时、分钟、秒 [，时区]）	使用年、月、日、小时、分钟、秒和时区字段中的本地时区创建当前时间戳。
make_timestamp_ntz（年、月、日、小时、分钟、秒）	根据年、月、日、小时、分钟、秒字段创建本地日期时间。
make_ym_interval ([年 [，月]])	将年月与年、月进行间隔。
分钟（时间戳）	返回字符串/时间戳的分钟部分。
月（日期）	返回日期/时间戳的月份部分。
months_between（时间戳 1、timestamp2 [，roundOff]）	如果晚`timestamp1` 于`timestamp2` ，则结果为正。如果`timestamp1` 和`timestamp2` 在一个月的同一天，或者两者都是一个月的最后一天，则一天中的时间将被忽略。否则，除非 roundoff=False，否则差额将根据每月 31 天计算，并四舍五入为 8 位数。

函数	描述
下一天（开始日期、一周中的某一天）	返回晚于`start_date` 并按指示命名的第一个日期。如果至少有一个输入参数为 NULL，则该函数返回 NULL。
现在 ()	返回查询评估开始时的当前时间戳。
季度（日期）	返回日期在一年中的季度，范围为 1 到 4。
秒（时间戳）	返回字符串/时间戳的第二个组成部分。
会话窗口（时间列，间隙持续时间）	生成会话窗口，给定时间戳，指定列和间隙持续时间。有关详细说明和示例，请参阅结构化直播指南文档中的 “时间窗类型”。
timestamp_micros（微秒）	根据自 UTC 纪元以来的微秒数创建时间戳。
timestamp_millis（毫秒）	根据自 UTC 纪元以来的毫秒数创建时间戳。
时间戳_秒（秒）	根据自 UTC 时代以来的秒数（可以是小数）创建时间戳。
to_date (date_str [, fmt])	将带有`date_str`表达式的表达`fmt`式解析为日期。如果输入无效，则返回 null。默认情况下，如果省略了，它将遵循投射规则直到`fmt`某个日期。
to_timestamp (timestamp_str [, fmt])	将带有`timestamp_str` 表达式的表达式解析为时间戳。`fmt`如果输入无效，则返回 null。默认情况下，如果省略了，它将遵循时间戳的`fmt`强制转换规则。
to_timestamp_ltz (timestamp_str [, fmt])	将带有`timestamp_str` 表达式的表达`fmt`式解析为具有本地时区的时间戳。如果输入无效，则返回 null。默认情况下，如果省略了，它将遵循时间戳的`fmt`强制转换规则。

函数	描述
to_timestamp_ntz (timestamp_str [, fmt])	将带有`timestamp_str` 表达式的表达式解析为不带时区的时间戳。`fmt`如果输入无效，则返回 null。默认情况下，如果省略了，它将遵循时间戳的`fmt`强制转换规则。
to_unix_timestamp (TimeExp [, fmt])	返回给定时间的 UNIX 时间戳。
to_utc_timestamp（时间戳、时区）	给定一个像 '2017-07-14 02:40:00.0 '这样的时间戳，将其解释为给定时区的时间，并将该时间渲染为UTC的时间戳。例如，“GMT+1” 将产生 “2017-07-14 01:40:00.0”。
trunc（日期，fmt）	返回`date`时将一天中的时间部分截断为格式模型指定的单位。`fmt`
try_to_timestamp (timestamp_str [, fmt])	将带有`timestamp_str` 表达式的表达式解析为时间戳。`fmt`
unix_date（日期）	返回自 1970-01-01 以来的天数。
unix_micros（时间戳）	返回自世界标准时间 1970-01-01 00:00:00 以来的微秒数。
unix_millis（时间戳）	返回自世界标准时间 1970-01-01 00:00:00 以来的毫秒数。截断更高的精度级别。
unix_seconds（时间戳）	返回自世界标准时间 1970-01-01 00:00:00 以来的秒数。截断更高的精度级别。
unix_timestamp ([TimeExp [, fmt]])	返回当前或指定时间的 UNIX 时间戳。
工作日（日期）	返回 date/timestamp (0 = 星期一，1 = 星期二，...，6 = 星期日) 的一周中的某一天。
一年中的一周（日期）	返回给定日期当年中的某周。一周被认为从星期一开始，第 1 周是第一个超过 3 天的周。

函数	描述
窗口（time_column、window_duration [、window_duration [、start_time]]）	给定指定列的时间戳，将行分成一个或多个时间窗口。窗口开头包含在内，但窗口结尾是排他性的，例如 12:05 将在窗口 [12:05,12:10) 中，但不在 [12:00,12:05) 中。Windows 可以支持微秒精度。不支持以月为单位的 Windows。有关详细说明和示例，请参阅结构化流媒体指南文档中的 “事件时间的窗口操作”。
window_time（window_column）	从 time/session 窗口列中提取时间值，该值可用于窗口的事件时间值。提取的时间是 (window.end-1)，这反映了这样一个事实，即聚合窗口具有排他性上限-[开始，结束) 有关详细说明和示例，请参阅结构化流媒体指南文档中的 “事件时间的窗口操作”。
年（日期）	返回日期/时间戳的年份部分。

示例

```
-- add_months
SELECT add_months('2016-08-31', 1);
+-------------------------+
|add_months(2016-08-31, 1)|
+-------------------------+
|               2016-09-30|
+-------------------------+
-- convert_timezone
SELECT convert_timezone('Europe/Brussels', 'America/Los_Angeles',
 timestamp_ntz'2021-12-06 00:00:00');
+-----------------------------------------------------------------------------------------------
+
|convert_timezone(Europe/Brussels, America/Los_Angeles, TIMESTAMP_NTZ '2021-12-06
 00:00:00')|
+-----------------------------------------------------------------------------------------------
+
|                                                                                  2021-12-05
 15:00:00|
```

```
+-----------------------------------------------------------------------------------------------------
+
SELECT convert_timezone('Europe/Brussels', timestamp_ntz'2021-12-05 15:00:00');
+-----------------------------------------------------------------------------------------------------
+
|convert_timezone(current_timezone(), Europe/Brussels, TIMESTAMP_NTZ '2021-12-05
 15:00:00')|
+-----------------------------------------------------------------------------------------------------
+
|                                                                          2021-12-05
 07:00:00|
+-----------------------------------------------------------------------------------------------------
+
-- curdate
SELECT curdate();
+--------------+
|current_date()|
+--------------+
|    2024-02-24|
+--------------+
-- current_date
SELECT current_date();
+--------------+
|current_date()|
+--------------+
|    2024-02-24|
+--------------+
SELECT current_date;
+--------------+
|current_date()|
+--------------+
|    2024-02-24|
+--------------+
-- current_timestamp
SELECT current_timestamp();
+--------------------+
| current_timestamp()|
+--------------------+
|2024-02-24 16:36:...|
+--------------------+
SELECT current_timestamp;
+--------------------+
| current_timestamp()|
+--------------------+
```

```
|2024-02-24 16:36:...|
+--------------------+
-- current_timezone
SELECT current_timezone();
+------------------+
|current_timezone()|
+------------------+
|        Asia/Seoul|
+------------------+
-- date_add
SELECT date_add('2016-07-30', 1);
+------------------------+
|date_add(2016-07-30, 1)|
+------------------------+
|             2016-07-31|
+------------------------+
-- date_diff
SELECT date_diff('2009-07-31', '2009-07-30');
+----------------------------------+
|date_diff(2009-07-31, 2009-07-30)|
+----------------------------------+
|                                 1|
+----------------------------------+
SELECT date_diff('2009-07-30', '2009-07-31');
+----------------------------------+
|date_diff(2009-07-30, 2009-07-31)|
+----------------------------------+
|                                -1|
+----------------------------------+
-- date_format
SELECT date_format('2016-04-08', 'y');
+--------------------------+
|date_format(2016-04-08, y)|
+--------------------------+
|                      2016|
+--------------------------+
-- date_from_unix_date
SELECT date_from_unix_date(1);
+----------------------+
|date_from_unix_date(1)|
+----------------------+
|            1970-01-02|
+----------------------+
-- date_part
```

```
SELECT date_part('YEAR', TIMESTAMP '2019-08-12 01:00:00.123456');
+-------------------------------------------------------+
|date_part(YEAR, TIMESTAMP '2019-08-12 01:00:00.123456')|
+-------------------------------------------------------+
|                                                   2019|
+-------------------------------------------------------+
SELECT date_part('week', timestamp'2019-08-12 01:00:00.123456');
+-------------------------------------------------------+
|date_part(week, TIMESTAMP '2019-08-12 01:00:00.123456')|
+-------------------------------------------------------+
|                                                     33|
+-------------------------------------------------------+
SELECT date_part('doy', DATE'2019-08-12');
+---------------------------------+
|date_part(doy, DATE '2019-08-12')|
+---------------------------------+
|                              224|
+---------------------------------+
SELECT date_part('SECONDS', timestamp'2019-10-01 00:00:01.000001');
+-----------------------------------------------------------+
|date_part(SECONDS, TIMESTAMP '2019-10-01 00:00:01.000001')|
+-----------------------------------------------------------+
|                                                   1.000001|
+-----------------------------------------------------------+
SELECT date_part('days', interval 5 days 3 hours 7 minutes);
+--------------------------------------------------+
|date_part(days, INTERVAL '5 03:07' DAY TO MINUTE)|
+--------------------------------------------------+
|                                                 5|
+--------------------------------------------------+
SELECT date_part('seconds', interval 5 hours 30 seconds 1 milliseconds 1 microseconds);
+-------------------------------------------------------------+
|date_part(seconds, INTERVAL '05:00:30.001001' HOUR TO SECOND)|
+-------------------------------------------------------------+
|                                                    30.001001|
+-------------------------------------------------------------+
SELECT date_part('MONTH', INTERVAL '2021-11' YEAR TO MONTH);
+--------------------------------------------------+
|date_part(MONTH, INTERVAL '2021-11' YEAR TO MONTH)|
+--------------------------------------------------+
|                                                11|
+--------------------------------------------------+
SELECT date_part('MINUTE', INTERVAL '123 23:55:59.002001' DAY TO SECOND);
+-----------------------------------------------------------------+
```

```
|date_part(MINUTE, INTERVAL '123 23:55:59.002001' DAY TO SECOND)|
+----------------------------------------------------------------+
|                                                              55|
+----------------------------------------------------------------+
-- date_sub
SELECT date_sub('2016-07-30', 1);
+-----------------------+
|date_sub(2016-07-30, 1)|
+-----------------------+
|             2016-07-29|
+-----------------------+
-- date_trunc
SELECT date_trunc('YEAR', '2015-03-05T09:32:05.359');
+----------------------------------------+
|date_trunc(YEAR, 2015-03-05T09:32:05.359)|
+----------------------------------------+
|                      2015-01-01 00:00:00|
+----------------------------------------+
SELECT date_trunc('MM', '2015-03-05T09:32:05.359');
+--------------------------------------+
|date_trunc(MM, 2015-03-05T09:32:05.359)|
+--------------------------------------+
|                    2015-03-01 00:00:00|
+--------------------------------------+
SELECT date_trunc('DD', '2015-03-05T09:32:05.359');
+--------------------------------------+
|date_trunc(DD, 2015-03-05T09:32:05.359)|
+--------------------------------------+
|                    2015-03-05 00:00:00|
+--------------------------------------+
SELECT date_trunc('HOUR', '2015-03-05T09:32:05.359');
+----------------------------------------+
|date_trunc(HOUR, 2015-03-05T09:32:05.359)|
+----------------------------------------+
|                      2015-03-05 09:00:00|
+----------------------------------------+
SELECT date_trunc('MILLISECOND', '2015-03-05T09:32:05.123456');
+----------------------------------------------------+
|date_trunc(MILLISECOND, 2015-03-05T09:32:05.123456)|
+----------------------------------------------------+
|                                2015-03-05 09:32:...|
+----------------------------------------------------+
-- dateadd
SELECT dateadd('2016-07-30', 1);
```

```
+-----------------------+
|date_add(2016-07-30, 1)|
+-----------------------+
|             2016-07-31|
+-----------------------+
-- datediff
SELECT datediff('2009-07-31', '2009-07-30');
+--------------------------------+
|datediff(2009-07-31, 2009-07-30)|
+--------------------------------+
|                               1|
+--------------------------------+
SELECT datediff('2009-07-30', '2009-07-31');
+--------------------------------+
|datediff(2009-07-30, 2009-07-31)|
+--------------------------------+
|                              -1|
+--------------------------------+
-- datepart
SELECT datepart('YEAR', TIMESTAMP '2019-08-12 01:00:00.123456');
+-------------------------------------------------------------+
|datepart(YEAR FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-------------------------------------------------------------+
|                                                         2019|
+-------------------------------------------------------------+
SELECT datepart('week', timestamp'2019-08-12 01:00:00.123456');
+-------------------------------------------------------------+
|datepart(week FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-------------------------------------------------------------+
|                                                           33|
+-------------------------------------------------------------+
SELECT datepart('doy', DATE'2019-08-12');
+------------------------------------+
|datepart(doy FROM DATE '2019-08-12')|
+------------------------------------+
|                                 224|
+------------------------------------+
SELECT datepart('SECONDS', timestamp'2019-10-01 00:00:01.000001');
+----------------------------------------------------------------+
|datepart(SECONDS FROM TIMESTAMP '2019-10-01 00:00:01.000001')|
+----------------------------------------------------------------+
|                                                        1.000001|
+----------------------------------------------------------------+
SELECT datepart('days', interval 5 days 3 hours 7 minutes);
```

```
+-----------------------------------------------------+
|datepart(days FROM INTERVAL '5 03:07' DAY TO MINUTE)|
+-----------------------------------------------------+
|                                                    5|
+-----------------------------------------------------+
SELECT datepart('seconds', interval 5 hours 30 seconds 1 milliseconds 1 microseconds);
+-------------------------------------------------------------------+
|datepart(seconds FROM INTERVAL '05:00:30.001001' HOUR TO SECOND)|
+-------------------------------------------------------------------+
|                                                        30.001001|
+-------------------------------------------------------------------+
SELECT datepart('MONTH', INTERVAL '2021-11' YEAR TO MONTH);
+-----------------------------------------------------+
|datepart(MONTH FROM INTERVAL '2021-11' YEAR TO MONTH)|
+-----------------------------------------------------+
|                                                   11|
+-----------------------------------------------------+
SELECT datepart('MINUTE', INTERVAL '123 23:55:59.002001' DAY TO SECOND);
+---------------------------------------------------------------------+
|datepart(MINUTE FROM INTERVAL '123 23:55:59.002001' DAY TO SECOND)|
+---------------------------------------------------------------------+
|                                                                   55|
+---------------------------------------------------------------------+
-- day
SELECT day('2009-07-30');
+---------------+
|day(2009-07-30)|
+---------------+
|             30|
+---------------+
-- dayofmonth
SELECT dayofmonth('2009-07-30');
+----------------------+
|dayofmonth(2009-07-30)|
+----------------------+
|                    30|
+----------------------+
-- dayofweek
SELECT dayofweek('2009-07-30');
+---------------------+
|dayofweek(2009-07-30)|
+---------------------+
|                    5|
+---------------------+
```

```
-- dayofyear
SELECT dayofyear('2016-04-09');
+---------------------+
|dayofyear(2016-04-09)|
+---------------------+
|                  100|
+---------------------+
-- extract
SELECT extract(YEAR FROM TIMESTAMP '2019-08-12 01:00:00.123456');
+-----------------------------------------------------------+
|extract(YEAR FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-----------------------------------------------------------+
|                                                       2019|
+-----------------------------------------------------------+
SELECT extract(week FROM timestamp'2019-08-12 01:00:00.123456');
+-----------------------------------------------------------+
|extract(week FROM TIMESTAMP '2019-08-12 01:00:00.123456')|
+-----------------------------------------------------------+
|                                                         33|
+-----------------------------------------------------------+
SELECT extract(doy FROM DATE'2019-08-12');
+-----------------------------------+
|extract(doy FROM DATE '2019-08-12')|
+-----------------------------------+
|                                224|
+-----------------------------------+
SELECT extract(SECONDS FROM timestamp'2019-10-01 00:00:01.000001');
+-------------------------------------------------------------+
|extract(SECONDS FROM TIMESTAMP '2019-10-01 00:00:01.000001')|
+-------------------------------------------------------------+
|                                                     1.000001|
+-------------------------------------------------------------+
SELECT extract(days FROM interval 5 days 3 hours 7 minutes);
+----------------------------------------------------+
|extract(days FROM INTERVAL '5 03:07' DAY TO MINUTE)|
+----------------------------------------------------+
|                                                   5|
+----------------------------------------------------+
SELECT extract(seconds FROM interval 5 hours 30 seconds 1 milliseconds 1 microseconds);
+-----------------------------------------------------------------+
|extract(seconds FROM INTERVAL '05:00:30.001001' HOUR TO SECOND)|
+-----------------------------------------------------------------+
|                                                        30.001001|
+-----------------------------------------------------------------+
```

```
SELECT extract(MONTH FROM INTERVAL '2021-11' YEAR TO MONTH);
+-----------------------------------------------------+
|extract(MONTH FROM INTERVAL '2021-11' YEAR TO MONTH)|
+-----------------------------------------------------+
|                                                   11|
+-----------------------------------------------------+
SELECT extract(MINUTE FROM INTERVAL '123 23:55:59.002001' DAY TO SECOND);
+-------------------------------------------------------------------+
|extract(MINUTE FROM INTERVAL '123 23:55:59.002001' DAY TO SECOND)|
+-------------------------------------------------------------------+
|                                                                 55|
+-------------------------------------------------------------------+
-- from_unixtime
SELECT from_unixtime(0, 'yyyy-MM-dd HH:mm:ss');
+-------------------------------------+
|from_unixtime(0, yyyy-MM-dd HH:mm:ss)|
+-------------------------------------+
|                  1970-01-01 09:00:00|
+-------------------------------------+
SELECT from_unixtime(0);
+-------------------------------------+
|from_unixtime(0, yyyy-MM-dd HH:mm:ss)|
+-------------------------------------+
|                  1970-01-01 09:00:00|
+-------------------------------------+
-- from_utc_timestamp
SELECT from_utc_timestamp('2016-08-31', 'Asia/Seoul');
+-------------------------------------------+
|from_utc_timestamp(2016-08-31, Asia/Seoul)|
+-------------------------------------------+
|                       2016-08-31 09:00:00|
+-------------------------------------------+
-- hour
SELECT hour('2009-07-30 12:58:59');
+-------------------------+
|hour(2009-07-30 12:58:59)|
+-------------------------+
|                       12|
+-------------------------+
-- last_day
SELECT last_day('2009-01-12');
+--------------------+
|last_day(2009-01-12)|
+--------------------+
```

```
|          2009-01-31|
+--------------------+
-- localtimestamp
SELECT localtimestamp();
+--------------------+
|    localtimestamp()|
+--------------------+
|2024-02-24 16:36:...|
+--------------------+
-- make_date
SELECT make_date(2013, 7, 15);
+----------------------+
|make_date(2013, 7, 15)|
+----------------------+
|            2013-07-15|
+----------------------+
SELECT make_date(2019, 7, NULL);
+------------------------+
|make_date(2019, 7, NULL)|
+------------------------+
|                    NULL|
+------------------------+
-- make_dt_interval
SELECT make_dt_interval(1, 12, 30, 01.001001);
+-------------------------------------+
|make_dt_interval(1, 12, 30, 1.001001)|
+-------------------------------------+
|                 INTERVAL '1 12:30...|
+-------------------------------------+
SELECT make_dt_interval(2);
+----------------------------------+
|make_dt_interval(2, 0, 0, 0.000000)|
+----------------------------------+
|               INTERVAL '2 00:00...|
+----------------------------------+
SELECT make_dt_interval(100, null, 3);
+----------------------------------------+
|make_dt_interval(100, NULL, 3, 0.000000)|
+----------------------------------------+
|                                    NULL|
+----------------------------------------+
-- make_interval
SELECT make_interval(100, 11, 1, 1, 12, 30, 01.001001);
+-----------------------------------------------+
```

```
|make_interval(100, 11, 1, 1, 12, 30, 1.001001)|
+----------------------------------------------+
|                          100 years 11 mont...|
+----------------------------------------------+
SELECT make_interval(100, null, 3);
+----------------------------------------------+
|make_interval(100, NULL, 3, 0, 0, 0, 0.000000)|
+----------------------------------------------+
|                                          NULL|
+----------------------------------------------+
SELECT make_interval(0, 1, 0, 1, 0, 0, 100.000001);
+--------------------------------------------+
|make_interval(0, 1, 0, 1, 0, 0, 100.000001)|
+--------------------------------------------+
|                       1 months 1 days 1...|
+--------------------------------------------+
-- make_timestamp
SELECT make_timestamp(2014, 12, 28, 6, 30, 45.887);
+--------------------------------------------+
|make_timestamp(2014, 12, 28, 6, 30, 45.887)|
+--------------------------------------------+
|                       2014-12-28 06:30:...|
+--------------------------------------------+
SELECT make_timestamp(2014, 12, 28, 6, 30, 45.887, 'CET');
+--------------------------------------------------+
|make_timestamp(2014, 12, 28, 6, 30, 45.887, CET)|
+--------------------------------------------------+
|                            2014-12-28 14:30:...|
+--------------------------------------------------+
SELECT make_timestamp(2019, 6, 30, 23, 59, 60);
+---------------------------------------+
|make_timestamp(2019, 6, 30, 23, 59, 60)|
+---------------------------------------+
|                    2019-07-01 00:00:00|
+---------------------------------------+
SELECT make_timestamp(2019, 6, 30, 23, 59, 1);
+--------------------------------------+
|make_timestamp(2019, 6, 30, 23, 59, 1)|
+--------------------------------------+
|                   2019-06-30 23:59:01|
+--------------------------------------+
SELECT make_timestamp(null, 7, 22, 15, 30, 0);
+-------------------------------------+
|make_timestamp(NULL, 7, 22, 15, 30, 0)|
```

```
+-------------------------------------+
|                                NULL|
+-------------------------------------+
-- make_timestamp_ltz
SELECT make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887);
+-----------------------------------------------+
|make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887)|
+-----------------------------------------------+
|                           2014-12-28 06:30:...|
+-----------------------------------------------+
SELECT make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887, 'CET');
+-------------------------------------------------------+
|make_timestamp_ltz(2014, 12, 28, 6, 30, 45.887, CET)|
+-------------------------------------------------------+
|                                2014-12-28 14:30:...|
+-------------------------------------------------------+
SELECT make_timestamp_ltz(2019, 6, 30, 23, 59, 60);
+-------------------------------------------+
|make_timestamp_ltz(2019, 6, 30, 23, 59, 60)|
+-------------------------------------------+
|                        2019-07-01 00:00:00|
+-------------------------------------------+
SELECT make_timestamp_ltz(null, 7, 22, 15, 30, 0);
+-----------------------------------------+
|make_timestamp_ltz(NULL, 7, 22, 15, 30, 0)|
+-----------------------------------------+
|                                     NULL|
+-----------------------------------------+
-- make_timestamp_ntz
SELECT make_timestamp_ntz(2014, 12, 28, 6, 30, 45.887);
+-----------------------------------------------+
|make_timestamp_ntz(2014, 12, 28, 6, 30, 45.887)|
+-----------------------------------------------+
|                           2014-12-28 06:30:...|
+-----------------------------------------------+
SELECT make_timestamp_ntz(2019, 6, 30, 23, 59, 60);
+-------------------------------------------+
|make_timestamp_ntz(2019, 6, 30, 23, 59, 60)|
+-------------------------------------------+
|                        2019-07-01 00:00:00|
+-------------------------------------------+
SELECT make_timestamp_ntz(null, 7, 22, 15, 30, 0);
+-----------------------------------------+
|make_timestamp_ntz(NULL, 7, 22, 15, 30, 0)|
```

```
+-----------------------------------------+
|                                     NULL|
+-----------------------------------------+
-- make_ym_interval
SELECT make_ym_interval(1, 2);
+----------------------+
|make_ym_interval(1, 2)|
+----------------------+
|  INTERVAL '1-2' YE...|
+----------------------+
SELECT make_ym_interval(1, 0);
+----------------------+
|make_ym_interval(1, 0)|
+----------------------+
|  INTERVAL '1-0' YE...|
+----------------------+
SELECT make_ym_interval(-1, 1);
+-----------------------+
|make_ym_interval(-1, 1)|
+-----------------------+
|   INTERVAL '-0-11' ...|
+-----------------------+
SELECT make_ym_interval(2);
+----------------------+
|make_ym_interval(2, 0)|
+----------------------+
|  INTERVAL '2-0' YE...|
+----------------------+
-- minute
SELECT minute('2009-07-30 12:58:59');
+----------------------------+
|minute(2009-07-30 12:58:59)|
+----------------------------+
|                          58|
+----------------------------+
-- month
SELECT month('2016-07-30');
+-----------------+
|month(2016-07-30)|
+-----------------+
|                7|
+-----------------+
-- months_between
SELECT months_between('1997-02-28 10:30:00', '1996-10-30');
```

```
+-----------------------------------------------------+
|months_between(1997-02-28 10:30:00, 1996-10-30, true)|
+-----------------------------------------------------+
|                                           3.94959677|
+-----------------------------------------------------+
SELECT months_between('1997-02-28 10:30:00', '1996-10-30', false);
+------------------------------------------------------+
|months_between(1997-02-28 10:30:00, 1996-10-30, false)|
+------------------------------------------------------+
|                                  3.9495967741935485|
+------------------------------------------------------+
-- next_day
SELECT next_day('2015-01-14', 'TU');
+------------------------+
|next_day(2015-01-14, TU)|
+------------------------+
|              2015-01-20|
+------------------------+
-- now
SELECT now();
+--------------------+
|               now()|
+--------------------+
|2024-02-24 16:36:...|
+--------------------+
-- quarter
SELECT quarter('2016-08-31');
+-------------------+
|quarter(2016-08-31)|
+-------------------+
|                  3|
+-------------------+
-- second
SELECT second('2009-07-30 12:58:59');
+---------------------------+
|second(2009-07-30 12:58:59)|
+---------------------------+
|                         59|
+---------------------------+
-- session_window
SELECT a, session_window.start, session_window.end, count(*) as cnt FROM VALUES ('A1',
 '2021-01-01 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:10:00'),
 ('A2', '2021-01-01 00:01:00') AS tab(a, b) GROUP by a, session_window(b, '5 minutes')
 ORDER BY a, start;
```

```
+---+-------------------+-------------------+---+
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
| A1|2021-01-01 00:00:00|2021-01-01 00:09:30|  2|
| A1|2021-01-01 00:10:00|2021-01-01 00:15:00|  1|
| A2|2021-01-01 00:01:00|2021-01-01 00:06:00|  1|
+---+-------------------+-------------------+---+
SELECT a, session_window.start, session_window.end, count(*) as cnt FROM VALUES ('A1',
 '2021-01-01 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:10:00'),
 ('A2', '2021-01-01 00:01:00'), ('A2', '2021-01-01 00:04:30') AS tab(a, b) GROUP by a,
 session_window(b, CASE WHEN a = 'A1' THEN '5 minutes' WHEN a = 'A2' THEN '1 minute'
 ELSE '10 minutes' END) ORDER BY a, start;
+---+-------------------+-------------------+---+
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
| A1|2021-01-01 00:00:00|2021-01-01 00:09:30|  2|
| A1|2021-01-01 00:10:00|2021-01-01 00:15:00|  1|
| A2|2021-01-01 00:01:00|2021-01-01 00:02:00|  1|
| A2|2021-01-01 00:04:30|2021-01-01 00:05:30|  1|
+---+-------------------+-------------------+---+
-- timestamp_micros
SELECT timestamp_micros(1230219000123123);
+-----------------------------------+
|timestamp_micros(1230219000123123)|
+-----------------------------------+
|              2008-12-26 00:30:...|
+-----------------------------------+
-- timestamp_millis
SELECT timestamp_millis(1230219000123);
+-------------------------------+
|timestamp_millis(1230219000123)|
+-------------------------------+
|           2008-12-26 00:30:...|
+-------------------------------+
-- timestamp_seconds
SELECT timestamp_seconds(1230219000);
+----------------------------+
|timestamp_seconds(1230219000)|
+----------------------------+
|          2008-12-26 00:30:00|
+----------------------------+
SELECT timestamp_seconds(1230219000.123);
+---------------------------------+
|timestamp_seconds(1230219000.123)|
```

```
+---------------------------------+
|             2008-12-26 00:30:...|
+---------------------------------+
-- to_date
SELECT to_date('2009-07-30 04:17:52');
+----------------------------+
|to_date(2009-07-30 04:17:52)|
+----------------------------+
|                  2009-07-30|
+----------------------------+
SELECT to_date('2016-12-31', 'yyyy-MM-dd');
+------------------------------+
|to_date(2016-12-31, yyyy-MM-dd)|
+------------------------------+
|                     2016-12-31|
+------------------------------+
-- to_timestamp
SELECT to_timestamp('2016-12-31 00:12:00');
+---------------------------------+
|to_timestamp(2016-12-31 00:12:00)|
+---------------------------------+
|              2016-12-31 00:12:00|
+---------------------------------+
SELECT to_timestamp('2016-12-31', 'yyyy-MM-dd');
+-----------------------------------+
|to_timestamp(2016-12-31, yyyy-MM-dd)|
+-----------------------------------+
|                 2016-12-31 00:00:00|
+-----------------------------------+
-- to_timestamp_ltz
SELECT to_timestamp_ltz('2016-12-31 00:12:00');
+------------------------------------+
|to_timestamp_ltz(2016-12-31 00:12:00)|
+------------------------------------+
|                  2016-12-31 00:12:00|
+------------------------------------+
SELECT to_timestamp_ltz('2016-12-31', 'yyyy-MM-dd');
+----------------------------------------+
|to_timestamp_ltz(2016-12-31, yyyy-MM-dd)|
+----------------------------------------+
|                     2016-12-31 00:00:00|
+----------------------------------------+
-- to_timestamp_ntz
SELECT to_timestamp_ntz('2016-12-31 00:12:00');
```

```
+-------------------------------------+
|to_timestamp_ntz(2016-12-31 00:12:00)|
+-------------------------------------+
|                  2016-12-31 00:12:00|
+-------------------------------------+
SELECT to_timestamp_ntz('2016-12-31', 'yyyy-MM-dd');
+----------------------------------------+
|to_timestamp_ntz(2016-12-31, yyyy-MM-dd)|
+----------------------------------------+
|                     2016-12-31 00:00:00|
+----------------------------------------+
-- to_unix_timestamp
SELECT to_unix_timestamp('2016-04-08', 'yyyy-MM-dd');
+-----------------------------------------+
|to_unix_timestamp(2016-04-08, yyyy-MM-dd)|
+-----------------------------------------+
|                               1460041200|
+-----------------------------------------+
-- to_utc_timestamp
SELECT to_utc_timestamp('2016-08-31', 'Asia/Seoul');
+----------------------------------------+
|to_utc_timestamp(2016-08-31, Asia/Seoul)|
+----------------------------------------+
|                     2016-08-30 15:00:00|
+----------------------------------------+
-- trunc
SELECT trunc('2019-08-04', 'week');
+-----------------------+
|trunc(2019-08-04, week)|
+-----------------------+
|             2019-07-29|
+-----------------------+
SELECT trunc('2019-08-04', 'quarter');
+--------------------------+
|trunc(2019-08-04, quarter)|
+--------------------------+
|                2019-07-01|
+--------------------------+
SELECT trunc('2009-02-12', 'MM');
+---------------------+
|trunc(2009-02-12, MM)|
+---------------------+
|           2009-02-01|
+---------------------+
```

```
SELECT trunc('2015-10-27', 'YEAR');
+-----------------------+
|trunc(2015-10-27, YEAR)|
+-----------------------+
|             2015-01-01|
+-----------------------+
-- try_to_timestamp
SELECT try_to_timestamp('2016-12-31 00:12:00');
+-------------------------------------+
|try_to_timestamp(2016-12-31 00:12:00)|
+-------------------------------------+
|                  2016-12-31 00:12:00|
+-------------------------------------+
SELECT try_to_timestamp('2016-12-31', 'yyyy-MM-dd');
+----------------------------------------+
|try_to_timestamp(2016-12-31, yyyy-MM-dd)|
+----------------------------------------+
|                     2016-12-31 00:00:00|
+----------------------------------------+
SELECT try_to_timestamp('foo', 'yyyy-MM-dd');
+---------------------------------+
|try_to_timestamp(foo, yyyy-MM-dd)|
+---------------------------------+
|                             NULL|
+---------------------------------+
-- unix_date
SELECT unix_date(DATE("1970-01-02"));
+---------------------+
|unix_date(1970-01-02)|
+---------------------+
|                    1|
+---------------------+
-- unix_micros
SELECT unix_micros(TIMESTAMP('1970-01-01 00:00:01Z'));
+---------------------------------+
|unix_micros(1970-01-01 00:00:01Z)|
+---------------------------------+
|                          1000000|
+---------------------------------+
-- unix_millis
SELECT unix_millis(TIMESTAMP('1970-01-01 00:00:01Z'));
+---------------------------------+
|unix_millis(1970-01-01 00:00:01Z)|
+---------------------------------+
```

```
|                              1000|
+----------------------------------+
-- unix_seconds
SELECT unix_seconds(TIMESTAMP('1970-01-01 00:00:01Z'));
+----------------------------------+
|unix_seconds(1970-01-01 00:00:01Z)|
+----------------------------------+
|                                 1|
+----------------------------------+
-- unix_timestamp
SELECT unix_timestamp();
+-------------------------------------------------------+
|unix_timestamp(current_timestamp(), yyyy-MM-dd HH:mm:ss)|
+-------------------------------------------------------+
|                                            1708760216|
+-------------------------------------------------------+
SELECT unix_timestamp('2016-04-08', 'yyyy-MM-dd');
+--------------------------------------+
|unix_timestamp(2016-04-08, yyyy-MM-dd)|
+--------------------------------------+
|                            1460041200|
+--------------------------------------+
-- weekday
SELECT weekday('2009-07-30');
+-------------------+
|weekday(2009-07-30)|
+-------------------+
|                  3|
+-------------------+
-- weekofyear
SELECT weekofyear('2008-02-20');
+----------------------+
|weekofyear(2008-02-20)|
+----------------------+
|                     8|
+----------------------+
-- window
SELECT a, window.start, window.end, count(*) as cnt FROM VALUES ('A1', '2021-01-01
 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:06:00'), ('A2',
 '2021-01-01 00:01:00') AS tab(a, b) GROUP by a, window(b, '5 minutes') ORDER BY a,
 start;
+---+-------------------+-------------------+---+
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
```

```
| A1|2021-01-01 00:00:00|2021-01-01 00:05:00|  2|
| A1|2021-01-01 00:05:00|2021-01-01 00:10:00|  1|
| A2|2021-01-01 00:00:00|2021-01-01 00:05:00|  1|
+---+-------------------+-------------------+---+
SELECT a, window.start, window.end, count(*) as cnt FROM VALUES ('A1', '2021-01-01
 00:00:00'), ('A1', '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:06:00'), ('A2',
 '2021-01-01 00:01:00') AS tab(a, b) GROUP by a, window(b, '10 minutes', '5 minutes')
 ORDER BY a, start;
+---+-------------------+-------------------+---+
|  a|              start|                end|cnt|
+---+-------------------+-------------------+---+
| A1|2020-12-31 23:55:00|2021-01-01 00:05:00|  2|
| A1|2021-01-01 00:00:00|2021-01-01 00:10:00|  3|
| A1|2021-01-01 00:05:00|2021-01-01 00:15:00|  1|
| A2|2020-12-31 23:55:00|2021-01-01 00:05:00|  1|
| A2|2021-01-01 00:00:00|2021-01-01 00:10:00|  1|
+---+-------------------+-------------------+---+
-- window_time
SELECT a, window.start as start, window.end as end, window_time(window), cnt FROM
 (SELECT a, window, count(*) as cnt FROM VALUES ('A1', '2021-01-01 00:00:00'), ('A1',
 '2021-01-01 00:04:30'), ('A1', '2021-01-01 00:06:00'), ('A2', '2021-01-01 00:01:00')
 AS tab(a, b) GROUP by a, window(b, '5 minutes') ORDER BY a, window.start);
+---+-------------------+-------------------+--------------------+---+
|  a|              start|                end| window_time(window)|cnt|
+---+-------------------+-------------------+--------------------+---+
| A1|2021-01-01 00:00:00|2021-01-01 00:05:00|2021-01-01 00:04:...|  2|
| A1|2021-01-01 00:05:00|2021-01-01 00:10:00|2021-01-01 00:09:...|  1|
| A2|2021-01-01 00:00:00|2021-01-01 00:05:00|2021-01-01 00:04:...|  1|
+---+-------------------+-------------------+--------------------+---+
-- year
SELECT year('2016-07-30');
+----------------+
|year(2016-07-30)|
+----------------+
|            2016|
+----------------+
```

聚合函数

> **Note**
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

聚合函数对各行的值进行运算以执行数学计算，例如求和、平均 minimum/maximum 值、计数、值、标准差和估计，以及一些非数学运算。

语法

```
aggregate_function(input1 [, input2, ...]) FILTER (WHERE boolean_expression)
```

参数

- `boolean_expression`-指定任何计算结果类型为布尔值的表达式。可以使用逻辑运算符（AND、OR）将两个或多个表达式组合在一起。

有序集合函数

这些聚合函数使用与其他聚合函数不同的语法，因此可以指定对值进行排序的表达式（通常是列名）。

语法

```
{ PERCENTILE_CONT | PERCENTILE_DISC }(percentile) WITHIN GROUP (ORDER BY
 { order_by_expression [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [ , ... ] }) FILTER
 (WHERE boolean_expression)
```

参数

- `percentile`-要查找的值的百分位数。百分位数必须是介于 0.0 和 1.0 之间的常数。
- `order_by_expression`-在聚合值之前对值进行排序的表达式（通常是列名）。
- `boolean_expression`-指定任何计算结果类型为布尔值的表达式。可以使用逻辑运算符（AND、OR）将两个或多个表达式组合在一起。

示例

```
CREATE OR REPLACE TEMPORARY VIEW basic_pays AS SELECT * FROM VALUES
('Jane Doe','Accounting',8435),
('Akua Mansa','Accounting',9998),
('John Doe','Accounting',8992),
('Juan Li','Accounting',8870),
('Carlos Salazar','Accounting',11472),
('Arnav Desai','Accounting',6627),
('Saanvi Sarkar','IT',8113),
('Shirley Rodriguez','IT',5186),
('Nikki Wolf','Sales',9181),
('Alejandro Rosalez','Sales',9441),
('Nikhil Jayashankar','Sales',6660),
('Richard Roe','Sales',10563),
('Pat Candella','SCM',10449),
('Gerard Hernandez','SCM',6949),
('Pamela Castillo','SCM',11303),
('Paulo Santos','SCM',11798),
('Jorge Souza','SCM',10586)
AS basic_pays(employee_name, department, salary);
SELECT * FROM basic_pays;
+-------------------+----------+------+
|    employee_name  |department|salary|
+-------------------+----------+------+
| Arnav Desai       |Accounting|  6627|
| Jorge Souza       |       SCM| 10586|
| Jane Doe          |Accounting|  8435|
| Nikhil Jayashankar|     Sales|  6660|
| Diego Vanauf      |     Sales| 10563|
| Carlos Salazar    |Accounting| 11472|
| Gerard Hernandez  |       SCM|  6949|
| John Doe          |Accounting|  8992|
| Nikki Wolf        |     Sales|  9181|
| Paulo Santos      |       SCM| 11798|
| Saanvi Sarkar     |        IT|  8113|
| Shirley Rodriguez |        IT|  5186|
| Pat Candella      |       SCM| 10449|
| Akua Mansa        |Accounting|  9998|
| Pamela Castillo   |       SCM| 11303|
| Alejandro Rosalez |     Sales|  9441|
| Juan Li           |Accounting|  8870|
+-------------------+----------+------+
SELECT
department,
```

```
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary) AS pc1,
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary) FILTER (WHERE employee_name LIKE
 '%Bo%') AS pc2,
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary DESC) AS pc3,
percentile_cont(0.25) WITHIN GROUP (ORDER BY salary DESC) FILTER (WHERE employee_name
 LIKE '%Bo%') AS pc4,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary) AS pd1,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary) FILTER (WHERE employee_name LIKE
 '%Bo%') AS pd2,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary DESC) AS pd3,
percentile_disc(0.25) WITHIN GROUP (ORDER BY salary DESC) FILTER (WHERE employee_name
 LIKE '%Bo%') AS pd4
FROM basic_pays
GROUP BY department
ORDER BY department;
+----------+-------+--------+-------+--------+-----+-----+-----+-----+
|department|    pc1|     pc2|    pc3|     pc4|  pd1|  pd2|  pd3|  pd4|
+----------+-------+--------+-------+--------+-----+-----+-----+-----+
|Accounting|8543.75| 7838.25| 9746.5|10260.75| 8435| 6627| 9998|11472|
|        IT|5917.75|    NULL|7381.25|    NULL| 5186| NULL| 8113| NULL|
|     Sales|8550.75|    NULL| 9721.5|    NULL| 6660| NULL|10563| NULL|
|       SCM|10449.0|10786.25|11303.0|11460.75|10449|10449|11303|11798|
+----------+-------+--------+-------+--------+-----+-----+-----+-----+
```

条件函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
coalesce (expr1、expr2、...)	如果存在则返回第一个非空参数。否则为 null。
if (expr1、expr2、expr3)	如果`expr1`计算结果为 true，则返回`expr2`；否则返回`expr3`。
如果为空 (expr1、expr2)	`expr2`如果`expr1`为空则返回，`expr1`否则返回。

函数	描述
nanvl (expr1、expr2)	`expr1`如果不是 NaN，则返回，`expr2`否则返回。
nullif (expr1、expr2)	如果`expr1`等于，则返回 null`expr2`，`expr1`否则返回 null。
nvl (expr1、expr2)	`expr2`如果`expr1`为空则返回，`expr1`否则返回。
nvl2（expr1、expr2、expr3）	`expr2`如果不`expr1`为空则返回，`expr3`否则返回。
例子 expr1 然后是 expr2 [当 expr3 然后是 expr4] * [ELSE expr5] END	当 `expr1` = true 时，返回`expr2`；否则 当 `expr3` = true 时，返回`expr4`；否则返回`expr5`。

示例

```
-- coalesce
SELECT coalesce(NULL, 1, NULL);
+-----------------------+
|coalesce(NULL, 1, NULL)|
+-----------------------+
|                      1|
+-----------------------+
-- if
SELECT if(1 < 2, 'a', 'b');
+-------------------+
|(IF((1 < 2), a, b))|
+-------------------+
|                  a|
+-------------------+
-- ifnull
SELECT ifnull(NULL, array('2'));
+----------------------+
|ifnull(NULL, array(2))|
+----------------------+
|                   [2]|
+----------------------+
```

```
-- nanvl
SELECT nanvl(cast('NaN' as double), 123);
+-------------------------------+
|nanvl(CAST(NaN AS DOUBLE), 123)|
+-------------------------------+
|                          123.0|
+-------------------------------+
-- nullif
SELECT nullif(2, 2);
+------------+
|nullif(2, 2)|
+------------+
|        NULL|
+------------+
-- nvl
SELECT nvl(NULL, array('2'));
+--------------------+
|nvl(NULL, array(2))|
+--------------------+
|                [2]|
+--------------------+
-- nvl2
SELECT nvl2(NULL, 2, 1);
+----------------+
|nvl2(NULL, 2, 1)|
+----------------+
|               1|
+----------------+
-- when
SELECT CASE WHEN 1 > 0 THEN 1 WHEN 2 > 0 THEN 2.0 ELSE 1.2 END;
+-------------------------------------------------------------+
|CASE WHEN (1 > 0) THEN 1 WHEN (2 > 0) THEN 2.0 ELSE 1.2 END|
+-------------------------------------------------------------+
|                                                          1.0|
+-------------------------------------------------------------+
SELECT CASE WHEN 1 < 0 THEN 1 WHEN 2 > 0 THEN 2.0 ELSE 1.2 END;
+-------------------------------------------------------------+
|CASE WHEN (1 < 0) THEN 1 WHEN (2 > 0) THEN 2.0 ELSE 1.2 END|
+-------------------------------------------------------------+
|                                                          2.0|
+-------------------------------------------------------------+
SELECT CASE WHEN 1 < 0 THEN 1 WHEN 2 < 0 THEN 2.0 END;
+--------------------------------------------------+
|CASE WHEN (1 < 0) THEN 1 WHEN (2 < 0) THEN 2.0 END|
```

```
+-----------------------------------------------------+
|                                                NULL|
+-----------------------------------------------------+
```

JSON 函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
from_json（jsonStr、架构 [、选项]）	返回具有给定 `jsonStr`和`schema`的结构值。
get_json_object（json_txt，路径）	从 “路径” 中提取一个 json 对象。
json_array_length (jsonArray)	返回最外层的 JSON 数组中元素的数量。
json_object_keys (json_object)	以数组形式返回最外层的 JSON 对象的所有键。
json_tuple（jsonStr、p1、p2、...、pn）	返回一个像函数 get_json_object 这样的元组，但它需要多个名称。所有输入参数和输出列类型均为字符串。

函数	描述
schema_of_json（json [，选项]）	以 JSON 字符串的 DDL 格式返回架构。
to_json（expr [，选项]）	返回具有给定结构值的 JSON 字符串

示例

```
-- from_json
SELECT from_json('{"a":1, "b":0.8}', 'a INT, b DOUBLE');
+----------------------------+
| from_json({"a":1, "b":0.8}) |
+----------------------------+
| {1, 0.8}                   |
+----------------------------+

SELECT from_json('{"time":"26/08/2015"}', 'time Timestamp', map('timestampFormat', 'dd/
MM/yyyy'));
+---------------------------------+
| from_json({"time":"26/08/2015"}) |
+---------------------------------+
| {2015-08-26 00:00...            |
+---------------------------------+

SELECT from_json('{"teacher": "Alice", "student": [{"name": "Bob", "rank": 1}, {"name":
 "Charlie", "rank": 2}]}', 'STRUCT<teacher: STRING, student: ARRAY<STRUCT<name: STRING,
 rank: INT>>>');
+-------------------------------------------------------------------------------------------
+
| from_json({"teacher": "Alice", "student": [{"name": "Bob", "rank": 1}, {"name":
 "Charlie", "rank": 2}]}) |
+-------------------------------------------------------------------------------------------
+
| {Alice, [{Bob, 1}...
                   |
+-------------------------------------------------------------------------------------------
+
```

```
-- get_json_object
SELECT get_json_object('{"a":"b"}', '$.a');
+-------------------------------+
| get_json_object({"a":"b"}, $.a) |
+-------------------------------+
| b                             |
+-------------------------------+

-- json_array_length
SELECT json_array_length('[1,2,3,4]');
+----------------------------+
| json_array_length([1,2,3,4]) |
+----------------------------+
| 4                          |
+----------------------------+

SELECT json_array_length('[1,2,3,{"f1":1,"f2":[5,6]},4]');
+--------------------------------------------------+
| json_array_length([1,2,3,{"f1":1,"f2":[5,6]},4]) |
+--------------------------------------------------+
| 5                                                |
+--------------------------------------------------+

SELECT json_array_length('[1,2');
+------------------------+
| json_array_length([1,2) |
+------------------------+
| NULL                   |
+------------------------+

-- json_object_keys
SELECT json_object_keys('{}');
+--------------------+
| json_object_keys({}) |
+--------------------+
| []                 |
+--------------------+

SELECT json_object_keys('{"key": "value"}');
+----------------------------------+
| json_object_keys({"key": "value"}) |
+----------------------------------+
| [key]                            |
```

```
+----------------------------------+

SELECT json_object_keys('{"f1":"abc","f2":{"f3":"a", "f4":"b"}}');
+-----------------------------------------------------------+
| json_object_keys({"f1":"abc","f2":{"f3":"a", "f4":"b"}}) |
+-----------------------------------------------------------+
| [f1, f2]                                                  |
+-----------------------------------------------------------+

-- json_tuple
SELECT json_tuple('{"a":1, "b":2}', 'a', 'b');
+---+---+
| c0| c1|
+---+---+
|  1|  2|
+---+---+

-- schema_of_json
SELECT schema_of_json('[{"col":0}]');
+----------------------------+
| schema_of_json([{"col":0}]) |
+----------------------------+
| ARRAY<STRUCT<col:...      |
+----------------------------+

SELECT schema_of_json('[{"col":01}]', map('allowNumericLeadingZeros', 'true'));
+-----------------------------+
| schema_of_json([{"col":01}]) |
+-----------------------------+
| ARRAY<STRUCT<col:...       |
+-----------------------------+

-- to_json
SELECT to_json(named_struct('a', 1, 'b', 2));
+----------------------------------+
| to_json(named_struct(a, 1, b, 2)) |
+----------------------------------+
| {"a":1,"b":2}                    |
+----------------------------------+

SELECT to_json(named_struct('time', to_timestamp('2015-08-26', 'yyyy-MM-dd')),
 map('timestampFormat', 'dd/MM/yyyy'));
+-------------------------------------------------------------------+
| to_json(named_struct(time, to_timestamp(2015-08-26, yyyy-MM-dd))) |
```

```
+-----------------------------------------------------------------+
| {"time":"26/08/20...                                            |
+-----------------------------------------------------------------+

SELECT to_json(array(named_struct('a', 1, 'b', 2)));
+----------------------------------------+
| to_json(array(named_struct(a, 1, b, 2))) |
+----------------------------------------+
| [{"a":1,"b":2}]                        |
+----------------------------------------+

SELECT to_json(map('a', named_struct('b', 1)));
+-----------------------------------+
| to_json(map(a, named_struct(b, 1))) |
+-----------------------------------+
| {"a":{"b":1}}                     |
+-----------------------------------+

SELECT to_json(map(named_struct('a', 1),named_struct('b', 2)));
+-------------------------------------------------------+
| to_json(map(named_struct(a, 1), named_struct(b, 2))) |
+-------------------------------------------------------+
| {"[1]":{"b":2}}                                       |
+-------------------------------------------------------+

SELECT to_json(map('a', 1));
+------------------+
| to_json(map(a, 1)) |
+------------------+
| {"a":1}          |
+------------------+

SELECT to_json(array(map('a', 1)));
+-------------------------+
| to_json(array(map(a, 1))) |
+-------------------------+
| [{"a":1}]               |
+-------------------------+
```

数组函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
数组（expr，...）	返回一个包含给定元素的数组。
array_append（数组，元素）	在作为第一个参数传递的数组末尾添加元素。元素的类型应与数组中元素的类型相似。还会将空元素附加到数组中。但是如果数组传递了，则为 NULL 输出为 NULL
紧凑数组（数组）	从数组中移除空值。
array_contains（数组，值）	如果数组包含该值，则返回 true。
array_distinct（数组）	从数组中删除重复的值。
array_except（数组 1、array2）	返回数组 array1 中的元素，但不返回 array2 中的元素，没有重复项。
array_insert（x、pos、val）	将 val 放入数组 x 的索引 pos 中。数组索引从 1 开始。最大负索引为 -1，函数将在当前最后一个元素之后插入新元素。高于数组大小的索引会附加数组，如果索引为负数，则在数组前面加上 “空” 元素。
array_intersect（数组 1、array2）	返回 array1 和 array2 交叉处的元素的数组，没有重复项。
array_join（数组，分隔符 [，nullReplacemplace] ）	使用分隔符和可选字符串连接给定数组的元素以替换空值。如果未为 NullReplacemplace 设置任何值，则会筛选任何空值。

函数	描述
array_max（数组）	返回数组中的最大值。就 double/float 类型而言，NaN 大于任何非 NaN 元素。将跳过空元素。
array_min（数组）	返回数组中的最小值。就 double/float 类型而言，NaN 大于任何非 NaN 元素。将跳过空元素。
数组位置（数组、元素）	返回数组中第一个匹配元素的（从 1 开始）的索引，如果找不到匹配项，则返回 0。
array_prepend（数组、元素）	在作为第一个参数传递的数组开头添加元素。元素的类型应与数组中元素的类型相同。数组前面还会添加空元素。但是，如果传递的数组为 NULL，则输出为 NULL
array_remove（数组、元素）	从数组中移除所有等于元素的元素。
array_repeat（元素，计数）	返回包含元素计数时间的数组。
array_union（array1、array2）	返回 array1 和 array2 并集中的元素的数组，没有重复项。
arrays_overlay (a1, a2)	如果 a1 至少包含一个也存在于 a2 中的非空元素，则返回 true。如果数组没有公共元素并且它们都是非空的，并且其中任何一个都包含空元素，则返回 null，否则返回 false。
arrays_zip (a1, a2,...)	返回一个合并的结构数组，其中第 N 个结构包含输入数组的第 N 个值。
展平 () arrayOfArrays	将数组数组转换为单个数组。
get（数组、索引）	返回给定（基于 0）索引处的数组元素。如果索引指向数组边界之外，则此函数返回 NULL。

函数	描述
顺序（开始、停止、步进）	生成一个从头到尾（含）的元素数组，逐渐递增。返回元素的类型与参数表达式的类型相同。支持的类型有：字节、短整型、整型、长型、日期、时间戳。开始和停止表达式必须解析为相同的类型。如果开始和停止表达式解析为 “日期” 或 “时间戳” 类型，则步骤表达式必须解析为 “间隔”、“年-月间隔” 或 “日时间间隔” 类型，否则解析为与开始和停止表达式相同的类型。
随机播放（数组）	返回给定数组的随机排列。
切片（x、起点、长度）	子集数组 x 从索引开始（数组索引从 1 开始，如果起始为负数，则从结尾开始），长度为指定。
排序数组（数组 [，升序]）	根据数组元素的自然顺序按升序或降序对输入数组进行排序。就 double/float 类型而言，NaN 大于任何非 NaN 元素。Null 元素将按升序放置在返回数组的开头，或者按降序放置在返回数组的末尾。

示例

```
-- array
SELECT array(1, 2, 3);
+--------------+
|array(1, 2, 3)|
+--------------+
|     [1, 2, 3]|
+--------------+
-- array_append
SELECT array_append(array('b', 'd', 'c', 'a'), 'd');
+----------------------------------+
|array_append(array(b, d, c, a), d)|
+----------------------------------+
|                   [b, d, c, a, d]|
```

```
+-----------------------------------+
SELECT array_append(array(1, 2, 3, null), null);
+----------------------------------------+
|array_append(array(1, 2, 3, NULL), NULL)|
+----------------------------------------+
|                    [1, 2, 3, NULL, N...|
+----------------------------------------+
SELECT array_append(CAST(null as Array<Int>), 2);
+---------------------+
|array_append(NULL, 2)|
+---------------------+
|                 NULL|
+---------------------+
-- array_compact
SELECT array_compact(array(1, 2, 3, null));
+-----------------------------------+
|array_compact(array(1, 2, 3, NULL))|
+-----------------------------------+
|                          [1, 2, 3]|
+-----------------------------------+
SELECT array_compact(array("a", "b", "c"));
+----------------------------+
|array_compact(array(a, b, c))|
+----------------------------+
|                    [a, b, c]|
+----------------------------+
-- array_contains
SELECT array_contains(array(1, 2, 3), 2);
+---------------------------------+
|array_contains(array(1, 2, 3), 2)|
+---------------------------------+
|                             true|
+---------------------------------+
-- array_distinct
SELECT array_distinct(array(1, 2, 3, null, 3));
+---------------------------------------+
|array_distinct(array(1, 2, 3, NULL, 3))|
+---------------------------------------+
|                        [1, 2, 3, NULL]|
+---------------------------------------+
-- array_except
SELECT array_except(array(1, 2, 3), array(1, 3, 5));
+---------------------------------------------+
|array_except(array(1, 2, 3), array(1, 3, 5))|
```

```
+-------------------------------------------+
|                                       [2]|
+-------------------------------------------+
-- array_insert
SELECT array_insert(array(1, 2, 3, 4), 5, 5);
+-------------------------------------+
|array_insert(array(1, 2, 3, 4), 5, 5)|
+-------------------------------------+
|                     [1, 2, 3, 4, 5]|
+-------------------------------------+
SELECT array_insert(array(5, 4, 3, 2), -1, 1);
+--------------------------------------+
|array_insert(array(5, 4, 3, 2), -1, 1)|
+--------------------------------------+
|                      [5, 4, 3, 2, 1]|
+--------------------------------------+
SELECT array_insert(array(5, 3, 2, 1), -4, 4);
+--------------------------------------+
|array_insert(array(5, 3, 2, 1), -4, 4)|
+--------------------------------------+
|                      [5, 4, 3, 2, 1]|
+--------------------------------------+
-- array_intersect
SELECT array_intersect(array(1, 2, 3), array(1, 3, 5));
+-------------------------------------------------+
|array_intersect(array(1, 2, 3), array(1, 3, 5))|
+-------------------------------------------------+
|                                          [1, 3]|
+-------------------------------------------------+
-- array_join
SELECT array_join(array('hello', 'world'), ' ');
+----------------------------------+
|array_join(array(hello, world),  )|
+----------------------------------+
|                       hello world|
+----------------------------------+
SELECT array_join(array('hello', null ,'world'), ' ');
+----------------------------------------+
|array_join(array(hello, NULL, world),  )|
+----------------------------------------+
|                             hello world|
+----------------------------------------+
SELECT array_join(array('hello', null ,'world'), ' ', ',');
+-------------------------------------------+
```

```
|array_join(array(hello, NULL, world),  , ,)|
+-------------------------------------------+
|                             hello , world|
+-------------------------------------------+
-- array_max
SELECT array_max(array(1, 20, null, 3));
+--------------------------------+
|array_max(array(1, 20, NULL, 3))|
+--------------------------------+
|                              20|
+--------------------------------+
-- array_min
SELECT array_min(array(1, 20, null, 3));
+--------------------------------+
|array_min(array(1, 20, NULL, 3))|
+--------------------------------+
|                               1|
+--------------------------------+
-- array_position
SELECT array_position(array(312, 773, 708, 708), 708);
+-----------------------------------------------+
|array_position(array(312, 773, 708, 708), 708)|
+-----------------------------------------------+
|                                              3|
+-----------------------------------------------+
SELECT array_position(array(312, 773, 708, 708), 414);
+-----------------------------------------------+
|array_position(array(312, 773, 708, 708), 414)|
+-----------------------------------------------+
|                                              0|
+-----------------------------------------------+
-- array_prepend
SELECT array_prepend(array('b', 'd', 'c', 'a'), 'd');
+-----------------------------------+
|array_prepend(array(b, d, c, a), d)|
+-----------------------------------+
|                    [d, b, d, c, a]|
+-----------------------------------+
SELECT array_prepend(array(1, 2, 3, null), null);
+----------------------------------------+
|array_prepend(array(1, 2, 3, NULL), NULL)|
+----------------------------------------+
|                     [NULL, 1, 2, 3, N...|
+----------------------------------------+
```

```
SELECT array_prepend(CAST(null as Array<Int>), 2);
+----------------------+
|array_prepend(NULL, 2)|
+----------------------+
|                  NULL|
+----------------------+
-- array_remove
SELECT array_remove(array(1, 2, 3, null, 3), 3);
+----------------------------------------+
|array_remove(array(1, 2, 3, NULL, 3), 3)|
+----------------------------------------+
|                            [1, 2, NULL]|
+----------------------------------------+
-- array_repeat
SELECT array_repeat('123', 2);
+--------------------+
|array_repeat(123, 2)|
+--------------------+
|          [123, 123]|
+--------------------+
-- array_union
SELECT array_union(array(1, 2, 3), array(1, 3, 5));
+--------------------------------------------+
|array_union(array(1, 2, 3), array(1, 3, 5))|
+--------------------------------------------+
|                                [1, 2, 3, 5]|
+--------------------------------------------+
-- arrays_overlap
SELECT arrays_overlap(array(1, 2, 3), array(3, 4, 5));
+-----------------------------------------------+
|arrays_overlap(array(1, 2, 3), array(3, 4, 5))|
+-----------------------------------------------+
|                                           true|
+-----------------------------------------------+
-- arrays_zip
SELECT arrays_zip(array(1, 2, 3), array(2, 3, 4));
+-------------------------------------------+
|arrays_zip(array(1, 2, 3), array(2, 3, 4))|
+-------------------------------------------+
|                      [{1, 2}, {2, 3}, ...|
+-------------------------------------------+
SELECT arrays_zip(array(1, 2), array(2, 3), array(3, 4));
+---------------------------------------------------+
|arrays_zip(array(1, 2), array(2, 3), array(3, 4))|
```

```
+-------------------------------------------------+
|                             [{1, 2, 3}, {2, 3...|
+-------------------------------------------------+
-- flatten
SELECT flatten(array(array(1, 2), array(3, 4)));
+----------------------------------------+
|flatten(array(array(1, 2), array(3, 4)))|
+----------------------------------------+
|                            [1, 2, 3, 4]|
+----------------------------------------+
-- get
SELECT get(array(1, 2, 3), 0);
+----------------------+
|get(array(1, 2, 3), 0)|
+----------------------+
|                     1|
+----------------------+
SELECT get(array(1, 2, 3), 3);
+----------------------+
|get(array(1, 2, 3), 3)|
+----------------------+
|                  NULL|
+----------------------+
SELECT get(array(1, 2, 3), -1);
+-----------------------+
|get(array(1, 2, 3), -1)|
+-----------------------+
|                   NULL|
+-----------------------+
-- sequence
SELECT sequence(1, 5);
+---------------+
| sequence(1, 5)|
+---------------+
|[1, 2, 3, 4, 5]|
+---------------+
SELECT sequence(5, 1);
+---------------+
| sequence(5, 1)|
+---------------+
|[5, 4, 3, 2, 1]|
+---------------+
SELECT sequence(to_date('2018-01-01'), to_date('2018-03-01'), interval 1 month);
+----------------------------------------------------------------------+
```

```
|sequence(to_date(2018-01-01), to_date(2018-03-01), INTERVAL '1' MONTH)|
+-----------------------------------------------------------------------+
|                                                   [2018-01-01, 2018...|
+-----------------------------------------------------------------------+
SELECT sequence(to_date('2018-01-01'), to_date('2018-03-01'), interval '0-1' year to
 month);
+-------------------------------------------------------------------------------------+
|sequence(to_date(2018-01-01), to_date(2018-03-01), INTERVAL '0-1' YEAR TO MONTH)|
+-------------------------------------------------------------------------------------+
|                                                                  [2018-01-01, 2018...|
+-------------------------------------------------------------------------------------+
-- shuffle
SELECT shuffle(array(1, 20, 3, 5));
+----------------------------+
|shuffle(array(1, 20, 3, 5))|
+----------------------------+
|              [5, 1, 20, 3]|
+----------------------------+
SELECT shuffle(array(1, 20, null, 3));
+------------------------------+
|shuffle(array(1, 20, NULL, 3))|
+------------------------------+
|              [1, NULL, 20, 3]|
+------------------------------+
-- slice
SELECT slice(array(1, 2, 3, 4), 2, 2);
+------------------------------+
|slice(array(1, 2, 3, 4), 2, 2)|
+------------------------------+
|                       [2, 3]|
+------------------------------+
SELECT slice(array(1, 2, 3, 4), -2, 2);
+-------------------------------+
|slice(array(1, 2, 3, 4), -2, 2)|
+-------------------------------+
|                        [3, 4]|
+-------------------------------+
-- sort_array
SELECT sort_array(array('b', 'd', null, 'c', 'a'), true);
+----------------------------------------+
|sort_array(array(b, d, NULL, c, a), true)|
+----------------------------------------+
|                      [NULL, a, b, c, d]|
```

```
+----------------------------------------+
```

窗口函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

窗口函数对一组行（称为窗口）进行操作，并根据这组行计算每行的返回值。窗口函数对于处理诸如计算移动平均线、计算累积统计数据或根据当前行的相对位置访问行值之类的任务非常有用。

语法

```
window_function [ nulls_option ] OVER ( [ { PARTITION | DISTRIBUTE } BY
 partition_col_name = partition_col_val ( [ , ... ] ) ] { ORDER | SORT } BY expression
 [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [ , ... ] [ window_frame ] )
```

参数

- 排名函数

  语法：`RANK | DENSE_RANK | PERCENT_RANK | NTILE | ROW_NUMBER`

  分析函数

  语法：`CUME_DIST | LAG | LEAD | NTH_VALUE | FIRST_VALUE | LAST_VALUE`

  聚合函数

  语法：`MAX | MIN | COUNT | SUM | AVG | ...`
- `nulls_option`-指定在评估窗口函数时是否跳过空值。尊重 NULLS 意味着不跳过空值，而 IGNORE NULLS 意味着跳过。如果未指定，则默认为 RESPECT NULLS。

  语法：`{ IGNORE | RESPECT } NULLS`

  注意：`Only LAG| LEAD | NTH_VALUE FIRST_VALUE | LAST_VALUE` 可以搭配使用`IGNORE NULLS`。
- `window_frame`-指定窗口的起始位置以及窗口的结束位置。

语法：`{ RANGE | ROWS } { frame_start | BETWEEN frame_start AND frame_end }`

frame_start 和 frame_end 的语法如下：

语法：`UNBOUNDED PRECEDING | offset PRECEDING | CURRENT ROW | offset FOLLOWING | UNBOUNDED FOLLOWING`

offset：指定与当前行位置的偏移量。

注意如果省略 frame_end，则默认为当前行。

示例

```
CREATE TABLE employees (name STRING, dept STRING, salary INT, age INT);
INSERT INTO employees VALUES ("Lisa", "Sales", 10000, 35);
INSERT INTO employees VALUES ("Evan", "Sales", 32000, 38);
INSERT INTO employees VALUES ("Fred", "Engineering", 21000, 28);
INSERT INTO employees VALUES ("Alex", "Sales", 30000, 33);
INSERT INTO employees VALUES ("Tom", "Engineering", 23000, 33);
INSERT INTO employees VALUES ("Jane", "Marketing", 29000, 28);
INSERT INTO employees VALUES ("Jeff", "Marketing", 35000, 38);
INSERT INTO employees VALUES ("Paul", "Engineering", 29000, 23);
INSERT INTO employees VALUES ("Chloe", "Engineering", 23000, 25);
SELECT * FROM employees;
+-----+-----------+------+-----+
| name|       dept|salary|  age|
+-----+-----------+------+-----+
|Chloe|Engineering| 23000|   25|
| Fred|Engineering| 21000|   28|
| Paul|Engineering| 29000|   23|
|Helen|  Marketing| 29000|   40|
|  Tom|Engineering| 23000|   33|
| Jane|  Marketing| 29000|   28|
| Jeff|  Marketing| 35000|   38|
| Evan|      Sales| 32000|   38|
| Lisa|      Sales| 10000|   35|
| Alex|      Sales| 30000|   33|
+-----+-----------+------+-----+
SELECT name, dept, salary, RANK() OVER (PARTITION BY dept ORDER BY salary) AS rank FROM
 employees;
+-----+-----------+------+----+
| name|       dept|salary|rank|
```

```
+-----+-----------+------+----+
| Lisa|      Sales| 10000|   1|
| Alex|      Sales| 30000|   2|
| Evan|      Sales| 32000|   3|
| Fred|Engineering| 21000|   1|
|  Tom|Engineering| 23000|   2|
|Chloe|Engineering| 23000|   2|
| Paul|Engineering| 29000|   4|
|Helen|  Marketing| 29000|   1|
| Jane|  Marketing| 29000|   1|
| Jeff|  Marketing| 35000|   3|
+-----+-----------+------+----+
SELECT name, dept, salary, DENSE_RANK() OVER (PARTITION BY dept ORDER BY salary ROWS
 BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS dense_rank FROM employees;
+-----+-----------+------+----------+
| name|       dept|salary|dense_rank|
+-----+-----------+------+----------+
| Lisa|      Sales| 10000|         1|
| Alex|      Sales| 30000|         2|
| Evan|      Sales| 32000|         3|
| Fred|Engineering| 21000|         1|
|  Tom|Engineering| 23000|         2|
|Chloe|Engineering| 23000|         2|
| Paul|Engineering| 29000|         3|
|Helen|  Marketing| 29000|         1|
| Jane|  Marketing| 29000|         1|
| Jeff|  Marketing| 35000|         2|
+-----+-----------+------+----------+
SELECT name, dept, age, CUME_DIST() OVER (PARTITION BY dept ORDER BY age
RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cume_dist FROM employees;
+-----+-----------+------+------------------+
| name|       dept|age   |         cume_dist|
+-----+-----------+------+------------------+
| Alex|      Sales|    33|0.3333333333333333|
| Lisa|      Sales|    35|0.6666666666666666|
| Evan|      Sales|    38|               1.0|
| Paul|Engineering|    23|              0.25|
|Chloe|Engineering|    25|              0.75|
| Fred|Engineering|    28|              0.25|
|  Tom|Engineering|    33|               1.0|
| Jane|  Marketing|    28|0.3333333333333333|
| Jeff|  Marketing|    38|0.6666666666666666|
|Helen|  Marketing|    40|               1.0|
```

```
+-----+-----------+------+------------------+
SELECT name, dept, salary, MIN(salary) OVER (PARTITION BY dept ORDER BY salary) AS min
FROM employees;
+-----+-----------+------+-----+
| name|       dept|salary|  min|
+-----+-----------+------+-----+
| Lisa|      Sales| 10000|10000|
| Alex|      Sales| 30000|10000|
| Evan|      Sales| 32000|10000|
|Helen|  Marketing| 29000|29000|
| Jane|  Marketing| 29000|29000|
| Jeff|  Marketing| 35000|29000|
| Fred|Engineering| 21000|21000|
|  Tom|Engineering| 23000|21000|
|Chloe|Engineering| 23000|21000|
| Paul|Engineering| 29000|21000|
+-----+-----------+------+-----+
SELECT name, salary,
LAG(salary) OVER (PARTITION BY dept ORDER BY salary) AS lag,
LEAD(salary, 1, 0) OVER (PARTITION BY dept ORDER BY salary) AS lead
FROM employees;
+-----+-----------+------+-----+-----+
| name|       dept|salary|  lag| lead|
+-----+-----------+------+-----+-----+
| Lisa|      Sales| 10000|NULL |30000|
| Alex|      Sales| 30000|10000|32000|
| Evan|      Sales| 32000|30000|    0|
| Fred|Engineering| 21000| NULL|23000|
|Chloe|Engineering| 23000|21000|23000|
|  Tom|Engineering| 23000|23000|29000|
| Paul|Engineering| 29000|23000|    0|
|Helen|  Marketing| 29000| NULL|29000|
| Jane|  Marketing| 29000|29000|35000|
| Jeff|  Marketing| 35000|29000|    0|
+-----+-----------+------+-----+-----+
SELECT id, v,
LEAD(v, 0) IGNORE NULLS OVER w lead,
LAG(v, 0) IGNORE NULLS OVER w lag,
NTH_VALUE(v, 2) IGNORE NULLS OVER w nth_value,
FIRST_VALUE(v) IGNORE NULLS OVER w first_value,
LAST_VALUE(v) IGNORE NULLS OVER w last_value
FROM test_ignore_null
WINDOW w AS (ORDER BY id)
ORDER BY id;
```

```
+--+----+----+----+---------+-----------+----------+
|id|   v|lead| lag|nth_value|first_value|last_value|
+--+----+----+----+---------+-----------+----------+
| 0|NULL|NULL|NULL|     NULL|       NULL|      NULL|
| 1|   x|   x|   x|     NULL|          x|         x|
| 2|NULL|NULL|NULL|     NULL|          x|         x|
| 3|NULL|NULL|NULL|     NULL|          x|         x|
| 4|   y|   y|   y|        y|          x|         y|
| 5|NULL|NULL|NULL|        y|          x|         y|
| 6|   z|   z|   z|        y|          x|         z|
| 7|   v|   v|   v|        y|          x|         v|
| 8|NULL|NULL|NULL|        y|          x|         v|
+--+----+----+----+---------+-----------+----------+
```

转换函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
bigint (expr)	将值 “expr” 转换为目标数据类型 “bigint”。
二进制 (expr)	将值 “expr” 转换为目标数据类型 “二进制”。
布尔值 (expr)	将值 “expr” 转换为目标数据类型 “布尔值”。
cast（expr AS 类型）	将值 `expr`转换为目标数据类型 `type`。
日期（过期）	将值 `expr`转换为目标数据类型`date`。
十进制 (expr)	将值 “expr” 转换为目标数据类型 “十进制”。
双精度（expr）	将值 “expr” 转换为目标数据类型 “double”。
float (expr)	将值 “expr” 转换为目标数据类型 “float”。
int (expr)	将值 `expr`转换为目标数据类型`int`。

函数	描述
smallint (expr)	将值 “expr” 转换为目标数据类型 “smallint”。
字符串 (expr)	将值 “expr” 转换为目标数据类型 “字符串”。
时间戳 (expr)	将值 “expr” 转换为目标数据类型 “时间戳”。
tinyint (expr)	将值 “expr” 转换为目标数据类型 “tinyint”。

示例

```
-- cast
SELECT cast(field as int);
+---------------+
|CAST(field AS INT)|
+---------------+
|             10|
+---------------+
```

谓词函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
！ expr	逻辑非。
expr1 < expr2	如果 “expr1” 小于 “expr2”，则返回 true。
expr1 <= expr2	如果 “expr1” 小于或等于 “expr2”，则返回 true。
expr1 <=> expr2	对于非空操作数，返回与 EQUAL (=) 运算符相同的结果，但是如果两个操作数都为空，则返回 true；如果其中一个为空，则返回 false。

函数	描述
expr1 = expr2	如果 `expr1` 等于 `expr2`，则返回真，否则返回假。
expr1 == expr2	如果 `expr1` 等于 `expr2`，则返回真，否则返回假。
expr1 > expr2	如果 “expr1” 大于 “expr2”，则返回 true。
expr1 >= expr2	如果 “expr1” 大于或等于 “expr2”，则返回 true。
expr1 和 expr2	合乎逻辑和.
str ilike pattern [逃生逃生]	如果 str 不区分大小写与 “escape” 与 “pattern” 匹配，则返回 true；如果有任何参数为空，则返回 null，否则返回 false。
expr1 in (expr2、expr3、...)	如果 “expr” 等于任何 valN，则返回 true。
isnan (expr)	如果 “expr” 为 NaN，则返回真，否则返回假。
isnotnull (expr)	如果 `expr`不为空则返回真，否则返回假。
为空 (expr)	如果 `expr`为空则返回真，否则返回假。
str like pattern [逃生逃生]	如果 str 将 “模式” 与 “escape” 匹配，则返回 true；如果有任何参数为空，则返回 null，否则返回 false。
不是 expr	逻辑非。
expr1 或 expr2	逻辑或。
regexp（str、regexp）	如果 “str” 与 “regexp” 匹配，则返回 true，否则返回 false。
regexp_like（str、regexp）	如果 “str” 与 “regexp” 匹配，则返回 true，否则返回 false。
rlike（str，regexp）	如果 “str” 与 “regexp” 匹配，则返回 true，否则返回 false。

示例

```
-- !
```

```
SELECT ! true;
+----------+
|(NOT true)|
+----------+
|     false|
+----------+
SELECT ! false;
+-----------+
|(NOT false)|
+-----------+
|       true|
+-----------+
SELECT ! NULL;
+----------+
|(NOT NULL)|
+----------+
|      NULL|
+----------+
-- <
SELECT to_date('2009-07-30 04:17:52') < to_date('2009-07-30 04:17:52');
+-----------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) < to_date(2009-07-30 04:17:52))|
+-----------------------------------------------------------------+
|                                                            false|
+-----------------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') < to_date('2009-08-01 04:17:52');
+-----------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) < to_date(2009-08-01 04:17:52))|
+-----------------------------------------------------------------+
|                                                             true|
+-----------------------------------------------------------------+
SELECT 1 < NULL;
+----------+
|(1 < NULL)|
+----------+
|      NULL|
+----------+
-- <=
SELECT 2 <= 2;
+--------+
|(2 <= 2)|
+--------+
|    true|
+--------+
```

```
SELECT 1.0 <= '1';
+----------+
|(1.0 <= 1)|
+----------+
|      true|
+----------+
SELECT to_date('2009-07-30 04:17:52') <= to_date('2009-07-30 04:17:52');
+-----------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) <= to_date(2009-07-30 04:17:52))|
+-----------------------------------------------------------------+
|                                                             true|
+-----------------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') <= to_date('2009-08-01 04:17:52');
+-----------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) <= to_date(2009-08-01 04:17:52))|
+-----------------------------------------------------------------+
|                                                             true|
+-----------------------------------------------------------------+
SELECT 1 <= NULL;
+-----------+
|(1 <= NULL)|
+-----------+
|       NULL|
+-----------+
-- <=>
SELECT 2 <=> 2;
+---------+
|(2 <=> 2)|
+---------+
|     true|
+---------+
SELECT 1 <=> '1';
+---------+
|(1 <=> 1)|
+---------+
|     true|
+---------+
SELECT true <=> NULL;
+---------------+
|(true <=> NULL)|
+---------------+
|          false|
+---------------+
SELECT NULL <=> NULL;
```

```
+---------------+
|(NULL <=> NULL)|
+---------------+
|           true|
+---------------+
-- =
SELECT 2 = 2;
+-------+
|(2 = 2)|
+-------+
|   true|
+-------+
SELECT 1 = '1';
+-------+
|(1 = 1)|
+-------+
|   true|
+-------+
SELECT true = NULL;
+-------------+
|(true = NULL)|
+-------------+
|         NULL|
+-------------+
SELECT NULL = NULL;
+-------------+
|(NULL = NULL)|
+-------------+
|         NULL|
+-------------+
-- ==
SELECT 2 == 2;
+-------+
|(2 = 2)|
+-------+
|   true|
+-------+
SELECT 1 == '1';
+-------+
|(1 = 1)|
+-------+
|   true|
+-------+
SELECT true == NULL;
```

```
+-------------+
|(true = NULL)|
+-------------+
|         NULL|
+-------------+
SELECT NULL == NULL;
+-------------+
|(NULL = NULL)|
+-------------+
|         NULL|
+-------------+
-- >
SELECT 2 > 1;
+-------+
|(2 > 1)|
+-------+
|   true|
+-------+
SELECT 2 > 1.1;
+-------+
|(2 > 1)|
+-------+
|   true|
+-------+
SELECT to_date('2009-07-30 04:17:52') > to_date('2009-07-30 04:17:52');
+-------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) > to_date(2009-07-30 04:17:52))|
+-------------------------------------------------------------+
|                                                        false|
+-------------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') > to_date('2009-08-01 04:17:52');
+-------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) > to_date(2009-08-01 04:17:52))|
+-------------------------------------------------------------+
|                                                        false|
+-------------------------------------------------------------+
SELECT 1 > NULL;
+----------+
|(1 > NULL)|
+----------+
|      NULL|
+----------+
-- >=
SELECT 2 >= 1;
```

```
+--------+
|(2 >= 1)|
+--------+
|    true|
+--------+
SELECT 2.0 >= '2.1';
+------------+
|(2.0 >= 2.1)|
+------------+
|       false|
+------------+
SELECT to_date('2009-07-30 04:17:52') >= to_date('2009-07-30 04:17:52');
+----------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) >= to_date(2009-07-30 04:17:52))|
+----------------------------------------------------------------+
|                                                            true|
+----------------------------------------------------------------+
SELECT to_date('2009-07-30 04:17:52') >= to_date('2009-08-01 04:17:52');
+----------------------------------------------------------------+
|(to_date(2009-07-30 04:17:52) >= to_date(2009-08-01 04:17:52))|
+----------------------------------------------------------------+
|                                                           false|
+----------------------------------------------------------------+
SELECT 1 >= NULL;
+-----------+
|(1 >= NULL)|
+-----------+
|       NULL|
+-----------+
-- and
SELECT true and true;
+---------------+
|(true AND true)|
+---------------+
|           true|
+---------------+
SELECT true and false;
+----------------+
|(true AND false)|
+----------------+
|           false|
+----------------+
SELECT true and NULL;
+---------------+
```

```
|(true AND NULL)|
+---------------+
|           NULL|
+---------------+
SELECT false and NULL;
+----------------+
|(false AND NULL)|
+----------------+
|           false|
+----------------+
-- ilike
SELECT ilike('Wagon', '_Agon');
+-------------------+
|ilike(Wagon, _Agon)|
+-------------------+
|               true|
+-------------------+
SELECT '%SystemDrive%\Users\John' ilike '\%SystemDrive\%\\users%';
+----------------------------------------------------------+
|ilike(%SystemDrive%\Users\John, \%SystemDrive\%\\users%)|
+----------------------------------------------------------+
|                                                      true|
+----------------------------------------------------------+
SELECT '%SystemDrive%\\USERS\\John' ilike '\%SystemDrive\%\\\\Users%';
+----------------------------------------------------------+
|ilike(%SystemDrive%\USERS\John, \%SystemDrive\%\\Users%)|
+----------------------------------------------------------+
|                                                      true|
+----------------------------------------------------------+
SELECT '%SystemDrive%/Users/John' ilike '/%SYSTEMDrive/%//Users%' ESCAPE '/';
+----------------------------------------------------------+
|ilike(%SystemDrive%/Users/John, /%SYSTEMDrive/%//Users%)|
+----------------------------------------------------------+
|                                                      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|
+---------------+
```

地图函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
元素_at（数组，索引）	返回给定（从 1 开始）索引处的数组元素。
元素_at（地图，键）	返回给定密钥的值。如果密钥不包含在地图中，则该函数返回 NULL。
地图（key0、value0、key1、value1...）	用给定的配 key/value 对创建地图。
map_concat（地图，...）	返回所有给定地图的并集
map_contains_key（地图、密钥）	如果地图包含密钥，则返回 true。
map_entris（地图）	返回给定地图中所有条目的无序数组。
map_from_arrays（键、值）	使用一对给定 key/value 数组创建地图。keys 中的所有元素都不应为空
map_from_entries () arrayOfEntries	返回根据给定条目数组创建的地图。
map_keys（地图）	返回一个包含地图键的无序数组。
map_values（地图）	返回包含地图值的无序数组。
str_to_map（文本 [，pairDelim [，]] ） keyValueDelim	使用分隔符将文本分成键/值对后创建地图。“pairDelim` 的默认分隔符是 '、' 和 ':' 代表 `。keyValueDelim`pairDelim`和``都被视为正则表达式。keyValueDelim

函数	描述
try_element_at（数组，索引）	返回给定（从 1 开始）索引处的数组元素。如果 Index 为 0，则系统将抛出错误。如果索引 < 0，则访问从最后一个到第一个的元素。如果索引超过数组的长度，则该函数始终返回 NULL。
try_element_at（地图，键）	返回给定密钥的值。如果密钥不包含在地图中，则该函数始终返回 NULL。

示例

```
-- element_at
SELECT element_at(array(1, 2, 3), 2);
+-----------------------------+
|element_at(array(1, 2, 3), 2)|
+-----------------------------+
|                            2|
+-----------------------------+
SELECT element_at(map(1, 'a', 2, 'b'), 2);
+------------------------------+
|element_at(map(1, a, 2, b), 2)|
+------------------------------+
|                             b|
+------------------------------+
-- map
SELECT map(1.0, '2', 3.0, '4');
+--------------------+
| map(1.0, 2, 3.0, 4)|
+--------------------+
|{1.0 -> 2, 3.0 -> 4}|
+--------------------+
-- map_concat
SELECT map_concat(map(1, 'a', 2, 'b'), map(3, 'c'));
+--------------------------------------+
|map_concat(map(1, a, 2, b), map(3, c))|
+--------------------------------------+
|                  {1 -> a, 2 -> b, ...|
+--------------------------------------+
-- map_contains_key
SELECT map_contains_key(map(1, 'a', 2, 'b'), 1);
```

```
+-----------------------------------+
|map_contains_key(map(1, a, 2, b), 1)|
+-----------------------------------+
|                               true|
+-----------------------------------+
SELECT map_contains_key(map(1, 'a', 2, 'b'), 3);
+-----------------------------------+
|map_contains_key(map(1, a, 2, b), 3)|
+-----------------------------------+
|                              false|
+-----------------------------------+
-- map_entries
SELECT map_entries(map(1, 'a', 2, 'b'));
+----------------------------+
|map_entries(map(1, a, 2, b))|
+----------------------------+
|            [{1, a}, {2, b}]|
+----------------------------+
-- map_from_arrays
SELECT map_from_arrays(array(1.0, 3.0), array('2', '4'));
+----------------------------------------------+
|map_from_arrays(array(1.0, 3.0), array(2, 4))|
+----------------------------------------------+
|                         {1.0 -> 2, 3.0 -> 4}|
+----------------------------------------------+
-- map_from_entries
SELECT map_from_entries(array(struct(1, 'a'), struct(2, 'b')));
+----------------------------------------------------+
|map_from_entries(array(struct(1, a), struct(2, b)))|
+----------------------------------------------------+
|                                   {1 -> a, 2 -> b}|
+----------------------------------------------------+
-- map_keys
SELECT map_keys(map(1, 'a', 2, 'b'));
+--------------------------+
|map_keys(map(1, a, 2, b))|
+--------------------------+
|                   [1, 2]|
+--------------------------+
-- map_values
SELECT map_values(map(1, 'a', 2, 'b'));
+----------------------------+
|map_values(map(1, a, 2, b))|
+----------------------------+
```

```
|                     [a, b]|
+---------------------------+
-- str_to_map
SELECT str_to_map('a:1,b:2,c:3', ',', ':');
+-----------------------------+
|str_to_map(a:1,b:2,c:3, ,, :)|
+-----------------------------+
|         {a -> 1, b -> 2, ...|
+-----------------------------+
SELECT str_to_map('a');
+-------------------+
|str_to_map(a, ,, :)|
+-------------------+
|        {a -> NULL}|
+-------------------+
-- try_element_at
SELECT try_element_at(array(1, 2, 3), 2);
+---------------------------------+
|try_element_at(array(1, 2, 3), 2)|
+---------------------------------+
|                                2|
+---------------------------------+
SELECT try_element_at(map(1, 'a', 2, 'b'), 2);
+----------------------------------+
|try_element_at(map(1, a, 2, b), 2)|
+----------------------------------+
|                                 b|
+----------------------------------+
```

数学函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

函数	描述
expr1% expr2	返回 `expr1`/`expr2` 之后的余数。

函数	描述
expr1 * expr2	返回 `expr1`*`expr2`。
expr1 + expr2	返回 `expr1`+`expr2`。
expr1-expr2	返回 `expr1`-`expr2`。
expr1/expr2	返回 `expr1`/`expr2`。它总是执行浮点除法。
abs (expr)	返回数值或间隔值的绝对值。
acos (expr)	返回 `expr`的反余弦值（又名反余弦值），就像由 `java.lang.math.acos`计算一样。
acosh (expr)	返回 `expr`的反双曲余弦值。
亚洲（expr）	返回 `expr` 的反正弦值（又名反正弦值），就像由 `java.lang.math.asin`计算得出的一样。
asing (expr)	返回 `expr`的反双曲正弦值。
atan (expr)	返回 `expr` 的反正切（又名反正切），就像由 `java.lang.math.atan` 计算得出的一样
atan2（expry、exprX）	返回平面 x 轴的正 x 轴与坐标（`exprx`，`expr y`）给出的点之间的角度（以弧度为单位），就像由 `java.lang.math.atan2` 计算得出的一样。
atanh (expr)	返回 `expr`的反双曲正切值。
bin (expr)	返回以二进制表示的长整值 “expr” 的字符串表示形式。
bround（expr、d）	使用 HALF_EVEN 四舍五入模式返回 “expr” 四舍五入到 “d” 小数位数。
cbrt (expr)	返回 `expr`的立方根。

函数	描述
ceil（expr [，scale]）	返回四舍五入后不小于 “expr” 的最小数字。可以指定一个可选的 “scale” 参数来控制舍入行为。
上限（expr [，比例]）	返回四舍五入后不小于 “expr” 的最小数字。可以指定一个可选的 “scale” 参数来控制舍入行为。
conv（num、from_base、to_base）	将 “数字” 从 “from_base” 转换为 “to_base”。
cos (expr)	返回 “expr” 的余弦值，就像由 `java.lang.math.cos` 计算得出的一样。
cosh (expr)	返回 `expr`的双曲余弦值，就像由 `java.lang.math.cosh`计算得出的一样。
成本（expr）	返回 `expr` 的余切值，就像由 `1/java.lang.math.tan` 计算得出的一样。
csc (expr)	返回 “expr” 的余割值，就像由 `1/java.lang.math.sin` 计算得出的一样。
度（expr）	将弧度转换为度。
expr1 div expr2	将 “expr1” 除以 “expr2”。如果操作数为空或 “expr2” 为 0，则返回 NULL。结果被抛得太长了。
e ()	返回欧拉的数字 e。
exp (expr)	将 e 返回到 “expr” 的次方。
expm1 (expr)-返回 exp (`expr`)	1
阶乘 (expr)	返回 “expr” 的阶乘。 `expr`是 [0.. 20]。否则为 null。

函数	描述
下限（expr [，缩放]）	返回四舍五入后不大于 “expr” 的最大数字。可以指定一个可选的 “scale” 参数来控制舍入行为。
最棒的（expr，...）	返回所有参数中的最大值，跳过空值。
十六进制 (expr)	将 `expr`转换为十六进制。
hypot（expr1、expr2）	返回 sqrt (`expr1`**2 + `expr2`**2)。
最少（expr，...）	返回所有参数中的最小值，跳过空值。
ln (expr)	返回 “expr” 的自然对数（以 e 为底）。
日志（基本、expr）	返回 “expr” 与 “base” 的对数。
log10 (expr)	返回以 10 为底的 “expr” 的对数。
log1p (expr)	返回日志 (1 + `expr`)。
log2 (expr)	返回以 2 为底的 “expr” 的对数。
expr1 模组 expr2	返回 `expr1`/`expr2` 之后的余数。
负数（expr）	返回 `expr`的负值。
pi ()	返回 pi。
pmod (expr1、expr2)	返回 `expr1` mod `expr2` 的正值。
阳性（expr）	返回 “expr” 的值。
pow (expr1、expr2)	将 “expr1” 提升到 “expr2” 的次方。
功率 (expr1、expr2)	将 “expr1” 提升到 “expr2” 的次方。
弧度 (expr)	将度数转换为弧度。
兰德 ([种子])	返回一个随机值，该值在 [0, 1) 中具有独立且同分布 (i.i.d.) 的均匀分布。

函数	描述
randn ([种子])	返回一个随机值，该值具有从标准正态分布中提取的独立且同分布 (i.i.d.) 的值。
随机（[种子]）	返回一个随机值，该值在 [0, 1) 中具有独立且同分布 (i.i.d.) 的均匀分布。
rint (expr)	返回值最接近参数且等于数学整数的双精度值。
回合（expr，d）	使用 HALF_UP 四舍五入模式返回 “expr” 四舍五入到 “d” 小数位数。
秒（expr）	返回 “expr” 的正割值，就像由 `1/java.lang.math.cos` 计算得出的一样。
向左移动（基本、expr）	按位左移。
标志 (expr)	返回 -1.0、0.0 或 1.0，因为 “expr” 是负数、0 或正数。
signum (expr)	返回 -1.0、0.0 或 1.0，因为 “expr” 是负数、0 或正数。
sin (expr)	返回 “expr” 的正弦值，就像由 `java.lang.math.sin` 计算得出的一样。
sing (expr)	返回 `expr`的双曲正弦值，就像由 `java.lang.math.sinh`计算出来一样。
sqrt (expr)	返回 `expr`的平方根。
棕褐色（expr）	返回 “expr” 的正切值，就像由 `java.lang.math.tan` 计算得出的一样。
tanh (expr)	返回 `expr` 的双曲正切值，就像由 `java.lang.math.tanh` 计算得出的一样。

函数	描述
try_add (expr1、expr2)	返回 `expr1`和`expr2`的总和，溢出时结果为空。可接受的输入类型与 `+` 运算符相同。
try_divide（股息，除数）	返回 “股息”/“除数”。它总是执行浮点除法。如果 `expr2` 为 0，则其结果始终为空。 “分红” 必须是数字或间隔。 “除数” 必须是数字。
try_multiply (expr1、expr2)	返回 `expr1`*`expr2`，溢出时结果为空。可接受的输入类型与 `*` 运算符相同。
try_subtract (expr1、expr2)	返回 `expr1`-`expr2`，溢出时结果为空。可接受的输入类型与 `-` 运算符相同。
unhex (expr)	将十六进制 `expr`转换为二进制。
width_bucket（值、最小值、最大值、num_bucket）	在带有 “num_bucket” 存储桶的等宽直方图中返回 “值” 的存储桶编号，范围为 “min_value” 到 “max_value”。”

示例

```
-- %
SELECT 2 % 1.8;
+---------+
|(2 % 1.8)|
+---------+
|      0.2|
+---------+
SELECT MOD(2, 1.8);
+-----------+
|mod(2, 1.8)|
+-----------+
|        0.2|
+-----------+
-- *
SELECT 2 * 3;
+-------+
|(2 * 3)|
```

```
+-------+
|      6|
+-------+
-- +
SELECT 1 + 2;
+-------+
|(1 + 2)|
+-------+
|      3|
+-------+
-- -
SELECT 2 - 1;
+-------+
|(2 - 1)|
+-------+
|      1|
+-------+
-- /
SELECT 3 / 2;
+-------+
|(3 / 2)|
+-------+
|    1.5|
+-------+
SELECT 2L / 2L;
+-------+
|(2 / 2)|
+-------+
|    1.0|
+-------+
-- abs
SELECT abs(-1);
+-------+
|abs(-1)|
+-------+
|      1|
+-------+
SELECT abs(INTERVAL -'1-1' YEAR TO MONTH);
+----------------------------------+
|abs(INTERVAL '-1-1' YEAR TO MONTH)|
+----------------------------------+
|              INTERVAL '1-1' YE...|
+----------------------------------+
-- acos
```

```
SELECT acos(1);
+-------+
|ACOS(1)|
+-------+
|    0.0|
+-------+
SELECT acos(2);
+-------+
|ACOS(2)|
+-------+
|    NaN|
+-------+
-- acosh
SELECT acosh(1);
+--------+
|ACOSH(1)|
+--------+
|     0.0|
+--------+
SELECT acosh(0);
+--------+
|ACOSH(0)|
+--------+
|     NaN|
+--------+
-- asin
SELECT asin(0);
+-------+
|ASIN(0)|
+-------+
|    0.0|
+-------+
SELECT asin(2);
+-------+
|ASIN(2)|
+-------+
|    NaN|
+-------+
-- asinh
SELECT asinh(0);
+--------+
|ASINH(0)|
+--------+
|     0.0|
```

```
+--------+
-- atan
SELECT atan(0);
+-------+
|ATAN(0)|
+-------+
|    0.0|
+-------+
-- atan2
SELECT atan2(0, 0);
+-----------+
|ATAN2(0, 0)|
+-----------+
|        0.0|
+-----------+
-- atanh
SELECT atanh(0);
+--------+
|ATANH(0)|
+--------+
|     0.0|
+--------+
SELECT atanh(2);
+--------+
|ATANH(2)|
+--------+
|     NaN|
+--------+
-- bin
SELECT bin(13);
+-------+
|bin(13)|
+-------+
|   1101|
+-------+
SELECT bin(-13);
+--------------------+
|            bin(-13)|
+--------------------+
|1111111111111111...|
+--------------------+
SELECT bin(13.3);
+---------+
|bin(13.3)|
```

```
+---------+
|     1101|
+---------+
-- bround
SELECT bround(2.5, 0);
+--------------+
|bround(2.5, 0)|
+--------------+
|             2|
+--------------+
SELECT bround(25, -1);
+--------------+
|bround(25, -1)|
+--------------+
|            20|
+--------------+
-- cbrt
SELECT cbrt(27.0);
+----------+
|CBRT(27.0)|
+----------+
|       3.0|
+----------+
-- ceil
SELECT ceil(-0.1);
+----------+
|CEIL(-0.1)|
+----------+
|         0|
+----------+
SELECT ceil(5);
+-------+
|CEIL(5)|
+-------+
|      5|
+-------+
SELECT ceil(3.1411, 3);
+---------------+
|ceil(3.1411, 3)|
+---------------+
|          3.142|
+---------------+
SELECT ceil(3.1411, -3);
+----------------+
```

```
|ceil(3.1411, -3)|
+----------------+
|            1000|
+----------------+
-- ceiling
SELECT ceiling(-0.1);
+-------------+
|ceiling(-0.1)|
+-------------+
|            0|
+-------------+
SELECT ceiling(5);
+----------+
|ceiling(5)|
+----------+
|         5|
+----------+
SELECT ceiling(3.1411, 3);
+------------------+
|ceiling(3.1411, 3)|
+------------------+
|             3.142|
+------------------+
SELECT ceiling(3.1411, -3);
+-------------------+
|ceiling(3.1411, -3)|
+-------------------+
|               1000|
+-------------------+
-- conv
SELECT conv('100', 2, 10);
+----------------+
|conv(100, 2, 10)|
+----------------+
|               4|
+----------------+
SELECT conv(-10, 16, -10);
+------------------+
|conv(-10, 16, -10)|
+------------------+
|               -16|
+------------------+
-- cos
SELECT cos(0);
```

```
+------+
|COS(0)|
+------+
|   1.0|
+------+
-- cosh
SELECT cosh(0);
+-------+
|COSH(0)|
+-------+
|    1.0|
+-------+
-- cot
SELECT cot(1);
+------------------+
|            COT(1)|
+------------------+
|0.6420926159343306|
+------------------+
-- csc
SELECT csc(1);
+------------------+
|            CSC(1)|
+------------------+
|1.1883951057781212|
+------------------+
-- degrees
SELECT degrees(3.141592653589793);
+--------------------------+
|DEGREES(3.141592653589793)|
+--------------------------+
|                     180.0|
+--------------------------+
-- div
SELECT 3 div 2;
+---------+
|(3 div 2)|
+---------+
|        1|
+---------+
SELECT INTERVAL '1-1' YEAR TO MONTH div INTERVAL '-1' MONTH;
+-------------------------------------------------------+
|(INTERVAL '1-1' YEAR TO MONTH div INTERVAL '-1' MONTH)|
+-------------------------------------------------------+
```

```
|                                                        -13|
+-----------------------------------------------------------+
-- e
SELECT e();
+-----------------+
|              E()|
+-----------------+
|2.718281828459045|
+-----------------+
-- exp
SELECT exp(0);
+------+
|EXP(0)|
+------+
|   1.0|
+------+
-- expm1
SELECT expm1(0);
+--------+
|EXPM1(0)|
+--------+
|     0.0|
+--------+
-- factorial
SELECT factorial(5);
+------------+
|factorial(5)|
+------------+
|         120|
+------------+
-- floor
SELECT floor(-0.1);
+-----------+
|FLOOR(-0.1)|
+-----------+
|         -1|
+-----------+
SELECT floor(5);
+--------+
|FLOOR(5)|
+--------+
|       5|
+--------+
SELECT floor(3.1411, 3);
```

```
+----------------+
|floor(3.1411, 3)|
+----------------+
|           3.141|
+----------------+
SELECT floor(3.1411, -3);
+-----------------+
|floor(3.1411, -3)|
+-----------------+
|                0|
+-----------------+
-- greatest
SELECT greatest(10, 9, 2, 4, 3);
+------------------------+
|greatest(10, 9, 2, 4, 3)|
+------------------------+
|                      10|
+------------------------+
-- hex
SELECT hex(17);
+-------+
|hex(17)|
+-------+
|     11|
+-------+
SELECT hex('SQL');
+------------------+
|    hex(SQL)|
+------------------+
|53514C|
+------------------+
-- hypot
SELECT hypot(3, 4);
+-----------+
|HYPOT(3, 4)|
+-----------+
|        5.0|
+-----------+
-- least
SELECT least(10, 9, 2, 4, 3);
+---------------------+
|least(10, 9, 2, 4, 3)|
+---------------------+
|                    2|
```

```
+---------------------+
-- ln
SELECT ln(1);
+-----+
|ln(1)|
+-----+
|  0.0|
+-----+
-- log
SELECT log(10, 100);
+------------+
|LOG(10, 100)|
+------------+
|         2.0|
+------------+
-- log10
SELECT log10(10);
+---------+
|LOG10(10)|
+---------+
|      1.0|
+---------+
-- log1p
SELECT log1p(0);
+--------+
|LOG1P(0)|
+--------+
|     0.0|
+--------+
-- log2
SELECT log2(2);
+-------+
|LOG2(2)|
+-------+
|    1.0|
+-------+
-- mod
SELECT 2 % 1.8;
+---------+
|(2 % 1.8)|
+---------+
|      0.2|
+---------+
SELECT MOD(2, 1.8);
```

```
+-----------+
|mod(2, 1.8)|
+-----------+
|        0.2|
+-----------+
-- negative
SELECT negative(1);
+-----------+
|negative(1)|
+-----------+
|         -1|
+-----------+
-- pi
SELECT pi();
+-----------------+
|             PI()|
+-----------------+
|3.141592653589793|
+-----------------+
-- pmod
SELECT pmod(10, 3);
+-----------+
|pmod(10, 3)|
+-----------+
|          1|
+-----------+
SELECT pmod(-10, 3);
+------------+
|pmod(-10, 3)|
+------------+
|           2|
+------------+
-- positive
SELECT positive(1);
+-----+
|(+ 1)|
+-----+
|    1|
+-----+
-- pow
SELECT pow(2, 3);
+---------+
|pow(2, 3)|
+---------+
```

```
|      8.0|
+---------+
-- power
SELECT power(2, 3);
+-----------+
|POWER(2, 3)|
+-----------+
|        8.0|
+-----------+
-- radians
SELECT radians(180);
+-----------------+
|     RADIANS(180)|
+-----------------+
|3.141592653589793|
+-----------------+
-- rand
SELECT rand();
+------------------+
|            rand()|
+------------------+
|0.7211420708112387|
+------------------+
SELECT rand(0);
+------------------+
|           rand(0)|
+------------------+
|0.7604953758285915|
+------------------+
SELECT rand(null);
+------------------+
|        rand(NULL)|
+------------------+
|0.7604953758285915|
+------------------+
-- randn
SELECT randn();
+-------------------+
|            randn()|
+-------------------+
|-0.8175603217732732|
+-------------------+
SELECT randn(0);
+------------------+
```

```
|          randn(0)|
+------------------+
|1.6034991609278433|
+------------------+
SELECT randn(null);
+------------------+
|       randn(NULL)|
+------------------+
|1.6034991609278433|
+------------------+
-- random
SELECT random();
+-----------------+
|           rand()|
+-----------------+
|0.394205008255365|
+-----------------+
SELECT random(0);
+------------------+
|           rand(0)|
+------------------+
|0.7604953758285915|
+------------------+
SELECT random(null);
+------------------+
|        rand(NULL)|
+------------------+
|0.7604953758285915|
+------------------+
-- rint
SELECT rint(12.3456);
+-------------+
|rint(12.3456)|
+-------------+
|         12.0|
+-------------+
-- round
SELECT round(2.5, 0);
+-------------+
|round(2.5, 0)|
+-------------+
|            3|
+-------------+
-- sec
```

```
SELECT sec(0);
+------+
|SEC(0)|
+------+
|   1.0|
+------+
-- shiftleft
SELECT shiftleft(2, 1);
+---------------+
|shiftleft(2, 1)|
+---------------+
|              4|
+---------------+
-- sign
SELECT sign(40);
+--------+
|sign(40)|
+--------+
|     1.0|
+--------+
SELECT sign(INTERVAL -'100' YEAR);
+--------------------------+
|sign(INTERVAL '-100' YEAR)|
+--------------------------+
|                      -1.0|
+--------------------------+
-- signum
SELECT signum(40);
+----------+
|SIGNUM(40)|
+----------+
|       1.0|
+----------+
SELECT signum(INTERVAL -'100' YEAR);
+----------------------------+
|SIGNUM(INTERVAL '-100' YEAR)|
+----------------------------+
|                        -1.0|
+----------------------------+
-- sin
SELECT sin(0);
+------+
|SIN(0)|
+------+
```

```
|   0.0|
+------+
-- sinh
SELECT sinh(0);
+-------+
|SINH(0)|
+-------+
|    0.0|
+-------+
-- sqrt
SELECT sqrt(4);
+-------+
|SQRT(4)|
+-------+
|    2.0|
+-------+
-- tan
SELECT tan(0);
+------+
|TAN(0)|
+------+
|   0.0|
+------+
-- tanh
SELECT tanh(0);
+-------+
|TANH(0)|
+-------+
|    0.0|
+-------+
-- try_add
SELECT try_add(1, 2);
+-------------+
|try_add(1, 2)|
+-------------+
|            3|
+-------------+
SELECT try_add(2147483647, 1);
+----------------------+
|try_add(2147483647, 1)|
+----------------------+
|                  NULL|
+----------------------+
SELECT try_add(date'2021-01-01', 1);
```

```
+-----------------------------+
|try_add(DATE '2021-01-01', 1)|
+-----------------------------+
|                   2021-01-02|
+-----------------------------+
SELECT try_add(date'2021-01-01', interval 1 year);
+----------------------------------------------+
|try_add(DATE '2021-01-01', INTERVAL '1' YEAR)|
+----------------------------------------------+
|                                  2022-01-01|
+----------------------------------------------+
SELECT try_add(timestamp'2021-01-01 00:00:00', interval 1 day);
+------------------------------------------------------------+
|try_add(TIMESTAMP '2021-01-01 00:00:00', INTERVAL '1' DAY)|
+------------------------------------------------------------+
|                                      2021-01-02 00:00:00|
+------------------------------------------------------------+
SELECT try_add(interval 1 year, interval 2 year);
+----------------------------------------------+
|try_add(INTERVAL '1' YEAR, INTERVAL '2' YEAR)|
+----------------------------------------------+
|                            INTERVAL '3' YEAR|
+----------------------------------------------+
-- try_divide
SELECT try_divide(3, 2);
+----------------+
|try_divide(3, 2)|
+----------------+
|             1.5|
+----------------+
SELECT try_divide(2L, 2L);
+----------------+
|try_divide(2, 2)|
+----------------+
|             1.0|
+----------------+
SELECT try_divide(1, 0);
+----------------+
|try_divide(1, 0)|
+----------------+
|            NULL|
+----------------+
SELECT try_divide(interval 2 month, 2);
+---------------------------------+
```

```
|try_divide(INTERVAL '2' MONTH, 2)|
+---------------------------------+
|             INTERVAL '0-1' YE...|
+---------------------------------+
SELECT try_divide(interval 2 month, 0);
+---------------------------------+
|try_divide(INTERVAL '2' MONTH, 0)|
+---------------------------------+
|                             NULL|
+---------------------------------+
-- try_multiply
SELECT try_multiply(2, 3);
+------------------+
|try_multiply(2, 3)|
+------------------+
|                 6|
+------------------+
SELECT try_multiply(-2147483648, 10);
+-----------------------------+
|try_multiply(-2147483648, 10)|
+-----------------------------+
|                         NULL|
+-----------------------------+
SELECT try_multiply(interval 2 year, 3);
+----------------------------------+
|try_multiply(INTERVAL '2' YEAR, 3)|
+----------------------------------+
|              INTERVAL '6-0' YE...|
+----------------------------------+
-- try_subtract
SELECT try_subtract(2, 1);
+------------------+
|try_subtract(2, 1)|
+------------------+
|                 1|
+------------------+
SELECT try_subtract(-2147483648, 1);
+----------------------------+
|try_subtract(-2147483648, 1)|
+----------------------------+
|                        NULL|
+----------------------------+
SELECT try_subtract(date'2021-01-02', 1);
+----------------------------------+
```

```
|try_subtract(DATE '2021-01-02', 1)|
+----------------------------------+
|                        2021-01-01|
+----------------------------------+
SELECT try_subtract(date'2021-01-01', interval 1 year);
+---------------------------------------------------+
|try_subtract(DATE '2021-01-01', INTERVAL '1' YEAR)|
+---------------------------------------------------+
|                                         2020-01-01|
+---------------------------------------------------+
SELECT try_subtract(timestamp'2021-01-02 00:00:00', interval 1 day);
+------------------------------------------------------------------+
|try_subtract(TIMESTAMP '2021-01-02 00:00:00', INTERVAL '1' DAY)|
+------------------------------------------------------------------+
|                                              2021-01-01 00:00:00|
+------------------------------------------------------------------+
SELECT try_subtract(interval 2 year, interval 1 year);
+---------------------------------------------------+
|try_subtract(INTERVAL '2' YEAR, INTERVAL '1' YEAR)|
+---------------------------------------------------+
|                                 INTERVAL '1' YEAR|
+---------------------------------------------------+
-- unhex
SELECT decode(unhex('53514C'), 'UTF-8');
+----------------------------------------+
|decode(unhex(53514C), UTF-8)|
+----------------------------------------+
|                                     SQL|
+----------------------------------------+
-- width_bucket
SELECT width_bucket(5.3, 0.2, 10.6, 5);
+--------------------------------+
|width_bucket(5.3, 0.2, 10.6, 5)|
+--------------------------------+
|                               3|
+--------------------------------+
SELECT width_bucket(-2.1, 1.3, 3.4, 3);
+--------------------------------+
|width_bucket(-2.1, 1.3, 3.4, 3)|
+--------------------------------+
|                               0|
+--------------------------------+
SELECT width_bucket(8.1, 0.0, 5.7, 4);
+-------------------------------+
```

```
|width_bucket(8.1, 0.0, 5.7, 4)|
+------------------------------+
|                             5|
+------------------------------+
SELECT width_bucket(-0.9, 5.2, 0.5, 2);
+-------------------------------+
|width_bucket(-0.9, 5.2, 0.5, 2)|
+-------------------------------+
|                              3|
+-------------------------------+
SELECT width_bucket(INTERVAL '0' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10);
+-------------------------------------------------------------------------------+
|width_bucket(INTERVAL '0' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10)|
+-------------------------------------------------------------------------------+
|                                                                              1|
+-------------------------------------------------------------------------------+
SELECT width_bucket(INTERVAL '1' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10);
+-------------------------------------------------------------------------------+
|width_bucket(INTERVAL '1' YEAR, INTERVAL '0' YEAR, INTERVAL '10' YEAR, 10)|
+-------------------------------------------------------------------------------+
|                                                                              2|
+-------------------------------------------------------------------------------+
SELECT width_bucket(INTERVAL '0' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10);
+-----------------------------------------------------------------------------+
|width_bucket(INTERVAL '0' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10)|
+-----------------------------------------------------------------------------+
|                                                                            1|
+-----------------------------------------------------------------------------+
SELECT width_bucket(INTERVAL '1' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10);
+-----------------------------------------------------------------------------+
|width_bucket(INTERVAL '1' DAY, INTERVAL '0' DAY, INTERVAL '10' DAY, 10)|
+-----------------------------------------------------------------------------+
|                                                                            2|
+-----------------------------------------------------------------------------+
```

生成器函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持这些 SQL 函数，请参阅the section called “支持的 SQL 命令”。

函数	描述
爆炸 (expr)	将数组 `expr`的元素分成多行，或者将映射 `expr`的元素分成多行和多列。除非另有说明，否则对数组的元素使用默认列名 “col”，对映射的元素使用默认的列名 “key” 和 “value”。
explode_outer (expr)	将数组 `expr`的元素分成多行，或者将映射 `expr`的元素分成多行和多列。除非另有说明，否则对数组的元素使用默认列名 “col”，对映射的元素使用默认的列名 “key” 和 “value”。
内联 (expr)	将结构数组分解成一个表。除非另有说明，否则默认使用列名 col1、col2 等。
inline_outer (expr)	将结构数组分解成一个表。除非另有说明，否则默认使用列名 col1、col2 等。
posexplode (expr)	将数组 `expr`的元素分成具有位置的多行，或者将映射 `expr`的元素分成具有位置的多行和列。除非另有说明，否则使用列名 “pos” 作为位置，使用 “col” 表示数组的元素，使用 “key” 和 “value” 表示地图的元素。
posexplode_outer (expr)	将数组 `expr`的元素分成具有位置的多行，或者将映射 `expr`的元素分成具有位置的多行和列。除非另有说明，否则使用列名 “pos” 作为位置，使用 “col” 表示数组的元素，使用 “key” 和 “value” 表示地图的元素。
stack (n、e xpr1、...、exprk)	将 “expr1”、...、“exprk” 分为 “n” 行。除非另有说明，否则默认使用列名 col0、col1 等。

示例

```
-- explode
SELECT explode(array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT explode(collection => array(10, 20));
```

```
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT * FROM explode(collection => array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

-- explode_outer
SELECT explode_outer(array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT explode_outer(collection => array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

SELECT * FROM explode_outer(collection => array(10, 20));
+---+
|col|
+---+
| 10|
| 20|
+---+

-- inline
SELECT inline(array(struct(1, 'a'), struct(2, 'b')));
+----+----+
|col1|col2|
```

```
+----+----+
|   1|   a|
|   2|   b|
+----+----+

-- inline_outer
SELECT inline_outer(array(struct(1, 'a'), struct(2, 'b')));
+----+----+
|col1|col2|
+----+----+
|   1|   a|
|   2|   b|
+----+----+

-- posexplode
SELECT posexplode(array(10,20));
+---+---+
|pos|col|
+---+---+
|  0| 10|
|  1| 20|
+---+---+

SELECT * FROM posexplode(array(10,20));
+---+---+
|pos|col|
+---+---+
|  0| 10|
|  1| 20|
+---+---+

-- posexplode_outer
SELECT posexplode_outer(array(10,20));
+---+---+
|pos|col|
+---+---+
|  0| 10|
|  1| 20|
+---+---+

SELECT * FROM posexplode_outer(array(10,20));
+---+---+
|pos|col|
+---+---+
```

```
|  0| 10|
|  1| 20|
+---+---+

-- stack
SELECT stack(2, 1, 2, 3);
+----+----+
|col0|col1|
+----+----+
|   1|   2|
|   3|NULL|
+----+----+
```

SELECT 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

OpenSearch SQL 支持用于从一个或多个表中检索结果集的`SELECT`语句。下一节描述了查询的整体语法和不同的查询结构。

语法

```
select_statement
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select_statement, ... ]
[ ORDER BY
    { expression [ ASC | DESC ] [ NULLS { FIRST | LAST } ]
    [ , ... ]
    }
]
[ SORT BY
    { expression [ ASC | DESC ] [ NULLS { FIRST | LAST } ]
    [ , ... ]
    }
]
[ WINDOW { named_window [ , WINDOW named_window, ... ] } ]
[ LIMIT { ALL | expression } ]
```

Wh `select_statement` ile 被定义为：

```
SELECT [ ALL | DISTINCT ] { [ [ named_expression ] [ , ... ] ] }
FROM { from_item [ , ... ] }
[ PIVOT clause ]
[ UNPIVOT clause ]
[ LATERAL VIEW clause ] [ ... ]
[ WHERE boolean_expression ]
[ GROUP BY expression [ , ... ] ]
[ HAVING boolean_expression ]
```

参数

- 全部

  从关系中选择所有匹配的行，默认情况下处于启用状态。
- 与众不同

  删除结果中的重复项后，从关系中选择所有匹配的行。
- 命名表达式

  具有指定名称的表达式。通常，它表示列表达式。

  语法：`expression [[AS] alias]`
- from_item

  表格关系

  加入关系

  枢轴关系

  取消透视关系

  表值函数

  内联表

  `[ LATERAL ] ( Subquery )`
- PIVOT

  该`PIVOT`子句用于数据透视。您可以根据特定的列值获取聚合值。
- UNPIVOT

该UNPIVOT子句将列转换为行。与之相反PIVOT，但值的聚合除外。

- 横向视图

  该LATERAL VIEW子句与生成器函数（例如）结合使用EXPLODE，后者将生成包含一行或多行的虚拟表。

  LATERAL VIEW会将这些行应用于每个原始输出行。

- WHERE

  根据提供的谓词筛选FROM子句的结果。

- 分组依据

  指定用于对行进行分组的表达式。

  它与聚合函数（MIN、、、、MAXCOUNTSUMAVG、等）结合使用，根据分组表达式对行进行分组，并聚合每个组中的值。

  将子FILTER句附加到聚合函数时，仅将匹配的行传递给该函数。

- 有

  指定筛选生成的行的GROUP BY谓词。

  该HAVING子句用于在执行分组后筛选行。

  如果HAVING不指定GROUP BY，则表示GROUP BY不带分组表达式（全局聚合）。

- 排序依据

  指定查询完整结果集的行顺序。

  输出行按分区排序。

  此参数与SORT BY和互斥DISTRIBUTE BY且不能同时指定。

- 排序依据

  指定每个分区中行的排序顺序。

  此参数与互斥ORDER BY且不能同时指定。

- LIMIT

指定语句或子查询可以返回的最大行数。

此子句主要与结合使用ORDER BY，以产生确定性结果。

- 布尔表达式

  指定任何计算结果类型为 boolean 的表达式。

  可以使用逻辑运算符 (AND,OR) 将两个或多个表达式组合在一起。

- expression

  指定一个或多个值、运算符和计算结果为一个值的 SQL 函数的组合。

- named_window

  为一个或多个源窗口规范指定别名。

  可以在查询的窗口定义中引用源窗口规范。

WHERE 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该WHERE子句用于根据指定条件限制查询或子查询的子FROM句的结果。

语法

```
WHERE boolean_expression
```

参数

- 布尔表达式

  指定任何计算结果类型为 boolean 的表达式。

  可以使用逻辑运算符 (AND,OR) 将两个或多个表达式组合在一起。

示例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
(300, 'Mike', 80),
(400, 'Dan',  50);

-- Comparison operator in `WHERE` clause.
SELECT * FROM person WHERE id > 200 ORDER BY id;
+---+----+---+
| id|name|age|
+---+----+---+
|300|Mike| 80|
|400| Dan| 50|
+---+----+---+

-- Comparison and logical operators in `WHERE` clause.
SELECT * FROM person WHERE id = 200 OR id = 300 ORDER BY id;
+---+----+----+
| id|name| age|
+---+----+----+
|200|Mary|null|
|300|Mike|  80|
+---+----+----+

-- IS NULL expression in `WHERE` clause.
SELECT * FROM person WHERE id > 300 OR age IS NULL ORDER BY id;
+---+----+----+
| id|name| age|
+---+----+----+
|200|Mary|null|
|400| Dan|  50|
+---+----+----+

-- Function expression in `WHERE` clause.
SELECT * FROM person WHERE length(name) > 3 ORDER BY id;
+---+----+----+
| id|name| age|
+---+----+----+
|100|John|  30|
|200|Mary|null|
|300|Mike|  80|
```

```
+---+----+----+

-- `BETWEEN` expression in `WHERE` clause.
SELECT * FROM person WHERE id BETWEEN 200 AND 300 ORDER BY id;
+---+----+----+
| id|name| age|
+---+----+----+
|200|Mary|null|
|300|Mike|  80|
+---+----+----+

-- Scalar Subquery in `WHERE` clause.
SELECT * FROM person WHERE age > (SELECT avg(age) FROM person);
+---+----+---+
| id|name|age|
+---+----+---+
|300|Mike| 80|
+---+----+---+

-- Correlated Subquery in `WHERE` clause.
SELECT id FROM person
WHERE exists (SELECT id FROM person where id = 200);
+---+----+----+
|id |name|age |
+---+----+----+
|200|Mary|null|
+---+----+----+
```

GROUP BY 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该GROUP BY子句用于根据一组指定的分组表达式对行进行分组，并根据一个或多个指定的聚合函数计算行组的聚合。

系统还通过GROUPING SETS、、CUBEROLLUP子句对同一个输入记录集进行多次聚合。分组表达式和高级聚合可以混合在GROUP BY子句中，也可以嵌套在子GROUPING SETS 句中。更多详情请见本Mixed/Nested Grouping Analytics 节。

将子FILTER句附加到聚合函数时，仅将匹配的行传递给该函数。

语法

```
GROUP BY group_expression [ , group_expression [ , ... ] ] [ WITH { ROLLUP | CUBE } ]
GROUP BY { group_expression | { ROLLUP | CUBE | GROUPING SETS } (grouping_set
 [ , ...]) } [ , ... ]
```

而聚合函数定义为：

```
aggregate_name ( [ DISTINCT ] expression [ , ... ] ) [ FILTER ( WHERE
 boolean_expression ) ]
```

参数

- 群组表达式

  指定将行分组在一起所依据的标准。根据分组表达式的结果值对行进行分组。

  分组表达式可以是类似的列名GROUP BY a、类似的列位置或类似GROUP BY 0的表达式GROUP BY a + b。

- 分组集

  分组集由括号中的零个或多个逗号分隔的表达式指定。当分组集只有一个元素时，可以省略括号。

  例如，GROUPING SETS ((a), (b)) 与 GROUPING SETS (a, b) 相同。

  语法：{ ([expression [, ...]]) | expression }

- 分组集

  对之后指定的每个分组集的行进行分组GROUPING SETS。

  例如，GROUP BY GROUPING SETS ((warehouse), (product))在语义上等同于GROUP BY warehouse和的结果并集。GROUP BY product此子句是 UNION ALL 的简写，其中UNION ALL运算符的每个分支对子句中指定的每个分组集进行聚合。GROUPING SETS

  同样，GROUP BY GROUPING SETS ((warehouse, product), (product), ())在语义上等同于全局聚合的结果GROUP BY warehouse, product, GROUP BY product并集。

- ROLLUP

在单个语句中指定多个聚合级别。此子句用于计算基于多个分组集的聚合。 ROLLUP是. 的简写。GROUPING SETS

例如，GROUP BY warehouse, product WITH ROLLUP or GROUP BY ROLLUP(warehouse, product) 等同于 GROUP BY GROUPING SETS((warehouse, product), (warehouse), ())。

GROUP BY ROLLUP(warehouse, product, (warehouse, location)) 等同于 GROUP BY GROUPING SETS((warehouse, product, location), (warehouse, product), (warehouse), ())。

ROLLUP 规范的 N 个元素会产生 N+1 个分组集。

- CUBE

  CUBE 子句用于根据 GROUP BY 子句中指定的分组列组合执行聚合。CUBE 是分组集合的简写。

  例如，GROUP BY warehouse, product WITH CUBE or GROUP BY CUBE(warehouse, product) 等同于 GROUP BY GROUPING SETS((warehouse, product), (warehouse), (product), ())。

  GROUP BY CUBE(warehouse, product, (warehouse, location)) 等同于 GROUP BY GROUPING SETS((warehouse, product, location), (warehouse, product), (warehouse, location), (product, warehouse, location), (warehouse), (product), (warehouse, product), ())。CUBE规范的 N 个元素的结果是 2^ GROUPING SETS N。

- 混合/嵌套分组分析

  一个GROUP BY子句可以包含多个组表达式和多个。CUBE|ROLLUP|GROUPING SETS GROUPING SETS也可以有嵌套CUBE|ROLLUP|GROUPING SETS子句，例如GROUPING SETS(ROLLUP(warehouse, location)、CUBE(warehouse, location))、GROUPING SETS(warehouse, GROUPING SETS(location, GROUPING SETS(ROLLUP(warehouse, location),CUBE(warehouse, location))))。

  CUBE|ROLLUP只是一个语法糖果GROUPING SETS。有关如何转换CUBE|ROLLUP为，请参阅上面的部分GROUPING SETS。 group_expression在这种情况下，可以被视为一个单一群体GROUPING SETS。

对于GROUP BY子句GROUPING SETS中的多个子句，我们GROUPING SETS通过对原始GROUPING SETS子句进行交叉乘积来生成单个。对于嵌套GROUPING SETS在GROUPING SETS子句中，我们只需取其分组集并将其去除即可。

例如，GROUP BY warehouse, GROUPING SETS((product), ()), GROUPING SETS((location, size), (location), (size), ()) and GROUP BY warehouse, ROLLUP(product), CUBE(location, size) 等同于 GROUP BY GROUPING SETS((warehouse, product, location, size), (warehouse, product, location), (warehouse, product, size), (warehouse, product), (warehouse, location, size), (warehouse, location), (warehouse, size), (warehouse))。

GROUP BY GROUPING SETS(GROUPING SETS(warehouse), GROUPING SETS((warehouse, product))) 等同于 GROUP BY GROUPING SETS((warehouse), (warehouse, product))。

- 聚合名称

  指定聚合函数名称（MINMAXCOUNT、SUM、AVG、、等）。

- 与众不同

  在将输入行中的重复项传递给聚合函数之前，将其删除。

- 过滤器

  筛选将WHERE子句boolean_expression中的计算结果为 true 的输入行传递给聚合函数；其他行将被丢弃。

示例

```
CREATE TABLE dealer (id INT, city STRING, car_model STRING, quantity INT);
INSERT INTO dealer VALUES
(100, 'Fremont', 'Honda Civic', 10),
(100, 'Fremont', 'Honda Accord', 15),
(100, 'Fremont', 'Honda CRV', 7),
(200, 'Dublin', 'Honda Civic', 20),
(200, 'Dublin', 'Honda Accord', 10),
(200, 'Dublin', 'Honda CRV', 3),
(300, 'San Jose', 'Honda Civic', 5),
(300, 'San Jose', 'Honda Accord', 8);

-- Sum of quantity per dealership. Group by `id`.
```

```
SELECT id, sum(quantity) FROM dealer GROUP BY id ORDER BY id;
+---+-------------+
| id|sum(quantity)|
+---+-------------+
|100|           32|
|200|           33|
|300|           13|
+---+-------------+

-- Use column position in GROUP by clause.
SELECT id, sum(quantity) FROM dealer GROUP BY 1 ORDER BY 1;
+---+-------------+
| id|sum(quantity)|
+---+-------------+
|100|           32|
|200|           33|
|300|           13|
+---+-------------+

-- Multiple aggregations.
-- 1. Sum of quantity per dealership.
-- 2. Max quantity per dealership.
SELECT id, sum(quantity) AS sum, max(quantity) AS max FROM dealer GROUP BY id ORDER BY
 id;
+---+---+---+
| id|sum|max|
+---+---+---+
|100| 32| 15|
|200| 33| 20|
|300| 13|  8|
+---+---+---+

-- Count the number of distinct dealer cities per car_model.
SELECT car_model, count(DISTINCT city) AS count FROM dealer GROUP BY car_model;
+------------+-----+
|   car_model|count|
+------------+-----+
| Honda Civic|    3|
|   Honda CRV|    2|
|Honda Accord|    3|
+------------+-----+

-- Sum of only 'Honda Civic' and 'Honda CRV' quantities per dealership.
SELECT id, sum(quantity) FILTER (
```

```
WHERE car_model IN ('Honda Civic', 'Honda CRV')
) AS `sum(quantity)` FROM dealer
GROUP BY id ORDER BY id;
+---+-------------+
| id|sum(quantity)|
+---+-------------+
|100|           17|
|200|           23|
|300|            5|
+---+-------------+

-- Aggregations using multiple sets of grouping columns in a single statement.
-- Following performs aggregations based on four sets of grouping columns.
-- 1. city, car_model
-- 2. city
-- 3. car_model
-- 4. Empty grouping set. Returns quantities for all city and car models.
SELECT city, car_model, sum(quantity) AS sum FROM dealer
GROUP BY GROUPING SETS ((city, car_model), (city), (car_model), ())
ORDER BY city;
+---------+------------+---+
|     city|   car_model|sum|
+---------+------------+---+
|     null|        null| 78|
|     null| HondaAccord| 33|
|     null|    HondaCRV| 10|
|     null|  HondaCivic| 35|
|   Dublin|        null| 33|
|   Dublin| HondaAccord| 10|
|   Dublin|    HondaCRV|  3|
|   Dublin|  HondaCivic| 20|
|  Fremont|        null| 32|
|  Fremont| HondaAccord| 15|
|  Fremont|    HondaCRV|  7|
|  Fremont|  HondaCivic| 10|
| San Jose|        null| 13|
| San Jose| HondaAccord|  8|
| San Jose|  HondaCivic|  5|
+---------+------------+---+

-- Group by processing with `ROLLUP` clause.
-- Equivalent GROUP BY GROUPING SETS ((city, car_model), (city), ())
SELECT city, car_model, sum(quantity) AS sum FROM dealer
GROUP BY city, car_model WITH ROLLUP
```

```
ORDER BY city, car_model;
+---------+------------+---+
|     city|   car_model|sum|
+---------+------------+---+
|     null|        null| 78|
|   Dublin|        null| 33|
|   Dublin| HondaAccord| 10|
|   Dublin|    HondaCRV|  3|
|   Dublin|  HondaCivic| 20|
|  Fremont|        null| 32|
|  Fremont| HondaAccord| 15|
|  Fremont|    HondaCRV|  7|
|  Fremont|  HondaCivic| 10|
| San Jose|        null| 13|
| San Jose| HondaAccord|  8|
| San Jose|  HondaCivic|  5|
+---------+------------+---+

-- Group by processing with `CUBE` clause.
-- Equivalent GROUP BY GROUPING SETS ((city, car_model), (city), (car_model), ())
SELECT city, car_model, sum(quantity) AS sum FROM dealer
GROUP BY city, car_model WITH CUBE
ORDER BY city, car_model;
+---------+------------+---+
|     city|   car_model|sum|
+---------+------------+---+
|     null|        null| 78|
|     null| HondaAccord| 33|
|     null|    HondaCRV| 10|
|     null|  HondaCivic| 35|
|   Dublin|        null| 33|
|   Dublin| HondaAccord| 10|
|   Dublin|    HondaCRV|  3|
|   Dublin|  HondaCivic| 20|
|  Fremont|        null| 32|
|  Fremont| HondaAccord| 15|
|  Fremont|    HondaCRV|  7|
|  Fremont|  HondaCivic| 10|
| San Jose|        null| 13|
| San Jose| HondaAccord|  8|
| San Jose|  HondaCivic|  5|
+---------+------------+---+

--Prepare data for ignore nulls example
```

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'Mary', NULL),
(200, 'John', 30),
(300, 'Mike', 80),
(400, 'Dan', 50);

--Select the first row in column age
SELECT FIRST(age) FROM person;
+--------------------+
| first(age, false)  |
+--------------------+
| NULL               |
+--------------------+

--Get the first row in column `age` ignore nulls,last row in column `id` and sum of
 column `id`.
SELECT FIRST(age IGNORE NULLS), LAST(id), SUM(id) FROM person;
+-------------------+------------------+----------+
| first(age, true)  | last(id, false)  | sum(id)  |
+-------------------+------------------+----------+
| 30                | 400              | 1000     |
+-------------------+------------------+----------+
```

HAVING 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该HAVING子句用于GROUP BY根据指定条件筛选生成的结果。它通常与GROUP BY子句一起使用。

语法

```
HAVING boolean_expression
```

参数

- 布尔表达式

指定任何计算结果类型为 boolean 的表达式。可以使用逻辑运算符 (AND,OR) 将两个或多个表达式组合在一起。

注意子HAVING句中指定的表达式只能指：

1. 常量
2. 出现在中的表达式 GROUP BY
3. 聚合函数

示例

```
CREATE TABLE dealer (id INT, city STRING, car_model STRING, quantity INT);
INSERT INTO dealer VALUES
(100, 'Fremont', 'Honda Civic', 10),
(100, 'Fremont', 'Honda Accord', 15),
(100, 'Fremont', 'Honda CRV', 7),
(200, 'Dublin', 'Honda Civic', 20),
(200, 'Dublin', 'Honda Accord', 10),
(200, 'Dublin', 'Honda CRV', 3),
(300, 'San Jose', 'Honda Civic', 5),
(300, 'San Jose', 'Honda Accord', 8);

-- `HAVING` clause referring to column in `GROUP BY`.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING city = 'Fremont';
+-------+---+
|   city|sum|
+-------+---+
|Fremont| 32|
+-------+---+

-- `HAVING` clause referring to aggregate function.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING sum(quantity) > 15;
+-------+---+
|   city|sum|
+-------+---+
| Dublin| 33|
|Fremont| 32|
+-------+---+

-- `HAVING` clause referring to aggregate function by its alias.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING sum > 15;
```

```
+-------+---+
|   city|sum|
+-------+---+
| Dublin| 33|
|Fremont| 32|
+-------+---+

-- `HAVING` clause referring to a different aggregate function than what is present in
-- `SELECT` list.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING max(quantity) > 15;
+------+---+
|  city|sum|
+------+---+
|Dublin| 33|
+------+---+

-- `HAVING` clause referring to constant expression.
SELECT city, sum(quantity) AS sum FROM dealer GROUP BY city HAVING 1 > 0 ORDER BY city;
+--------+---+
|    city|sum|
+--------+---+
|  Dublin| 33|
| Fremont| 32|
|San Jose| 13|
+--------+---+

-- `HAVING` clause without a `GROUP BY` clause.
SELECT sum(quantity) AS sum FROM dealer HAVING sum(quantity) > 10;
+---+
|sum|
+---+
| 78|
+---+
```

ORDER BY 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该ORDER BY子句用于按用户指定的顺序按排序方式返回结果行。与 SORT BY 子句不同，此子句保证输出中的总顺序。

语法

```
ORDER BY { expression [ sort_direction | nulls_sort_order ] [ , ... ] }
```

参数

- 排序依据

  指定以逗号分隔的表达式列表以及用于对行进行排序的可选参数sort_direction。nulls_sort_order

- 排序方向

  （可选）指定是按升序还是降序对行进行排序。

  排序方向的有效值ASC为升序和降序DESC。

  如果未明确指定排序方向，则默认情况下，行按升序排序。

  语法：[ASC | DESC]

- nulls_sort_order

  （可选）指定NULL值是否返回 before/after 非 NULL 值。

  如果未指定 null_sort_order，则如果排序顺序为，则先NULLs排序，如果排序顺序为 ASC NULLS 则排在最后。DESC

  1. 如果NULLS FIRST已指定，则无论排序顺序如何，都将首先返回 NULL 值。

  2. 如果NULLS LAST已指定，则无论排序顺序如何，最后都会返回 NULL 值。

  语法：[NULLS { FIRST | LAST }]

示例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
```

```
(300, 'Mike', 80),
(400, 'Jerry', NULL),
(500, 'Dan',  50);

-- Sort rows by age. By default rows are sorted in ascending manner with NULL FIRST.
SELECT name, age FROM person ORDER BY age;
+-----+----+
| name| age|
+-----+----+
|Jerry|null|
| Mary|null|
| John|  30|
|  Dan|  50|
| Mike|  80|
+-----+----+

-- Sort rows in ascending manner keeping null values to be last.
SELECT name, age FROM person ORDER BY age NULLS LAST;
+-----+----+
| name| age|
+-----+----+
| John|  30|
|  Dan|  50|
| Mike|  80|
| Mary|null|
|Jerry|null|
+-----+----+

-- Sort rows by age in descending manner, which defaults to NULL LAST.
SELECT name, age FROM person ORDER BY age DESC;
+-----+----+
| name| age|
+-----+----+
| Mike|  80|
|  Dan|  50|
| John|  30|
|Jerry|null|
| Mary|null|
+-----+----+

-- Sort rows in ascending manner keeping null values to be first.
SELECT name, age FROM person ORDER BY age DESC NULLS FIRST;
+-----+----+
| name| age|
```

```
+-----+----+
|Jerry|null|
| Mary|null|
| Mike|  80|
|  Dan|  50|
| John|  30|
+-----+----+

-- Sort rows based on more than one column with each column having different
-- sort direction.
SELECT * FROM person ORDER BY name ASC, age DESC;
+---+-----+----+
| id| name| age|
+---+-----+----+
|500|  Dan|  50|
|400|Jerry|null|
|100| John|  30|
|200| Mary|null|
|300| Mike|  80|
+---+-----+----+
```

JOIN 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

SQL 联接用于根据联接标准合并两个关系中的行。以下部分介绍整体联接语法和不同类型的联接以及示例。

语法

```
relation INNER JOIN relation [ join_criteria ]
```

参数

- 关系

  指定要连接的关系。

- 加入类型

指定联接类型。

语法：`INNER | CROSS | LEFT OUTER`

- 加入标准

  指定如何将一个关系中的行与另一个关系的行合并。

  语法：`ON boolean_expression | USING ( column_name [ , ... ] )`

- 布尔表达式

  指定返回类型为布尔值的表达式。

联接类型

- 内部联接

  需要明确指定内部联接。它选择在两个关系中具有匹配值的行。

  语法：`relation INNER JOIN relation [ join_criteria ]`

- 左键加入

  左联接返回左侧关系中的所有值和右侧关系中的匹配值，如果不存在匹配项，则附加 NULL。它也被称为左外连接。

  语法：`relation LEFT OUTER JOIN relation [ join_criteria ]`

- 交叉连接

  交叉连接返回两个关系的笛卡尔乘积。

  语法：`relation CROSS JOIN relation [ join_criteria ]`

示例

```
-- Use employee and department tables to demonstrate different type of joins.
SELECT * FROM employee;
+---+-----+------+
| id| name|deptno|
+---+-----+------+
|105|Chloe|     5|
```

```
|103| Paul|     3|
|101| John|     1|
|102| Lisa|     2|
|104| Evan|     4|
|106|  Amy|     6|
+---+-----+------+
SELECT * FROM department;
+------+-----------+
|deptno|   deptname|
+------+-----------+
|     3|Engineering|
|     2|      Sales|
|     1|  Marketing|
+------+-----------+

-- Use employee and department tables to demonstrate inner join.
SELECT id, name, employee.deptno, deptname
FROM employee INNER JOIN department ON employee.deptno = department.deptno;
+---+-----+------+-----------|
| id| name|deptno|   deptname|
+---+-----+------+-----------|
|103| Paul|     3|Engineering|
|101| John|     1|  Marketing|
|102| Lisa|     2|      Sales|
+---+-----+------+-----------|

-- Use employee and department tables to demonstrate left join.
SELECT id, name, employee.deptno, deptname
FROM employee LEFT JOIN department ON employee.deptno = department.deptno;
+---+-----+------+-----------|
| id| name|deptno|   deptname|
+---+-----+------+-----------|
|105|Chloe|     5|       NULL|
|103| Paul|     3|Engineering|
|101| John|     1|  Marketing|
|102| Lisa|     2|      Sales|
|104| Evan|     4|       NULL|
|106|  Amy|     6|       NULL|
+---+-----+------+-----------|

-- Use employee and department tables to demonstrate cross join.
SELECT id, name, employee.deptno, deptname FROM employee CROSS JOIN department;
+---+-----+------+-----------|
| id| name|deptno|   deptname|
```

```
+---+-----+------+-----------|
|105|Chloe|     5|Engineering|
|105|Chloe|     5|  Marketing|
|105|Chloe|     5|      Sales|
|103| Paul|     3|Engineering|
|103| Paul|     3|  Marketing|
|103| Paul|     3|      Sales|
|101| John|     1|Engineering|
|101| John|     1|  Marketing|
|101| John|     1|      Sales|
|102| Lisa|     2|Engineering|
|102| Lisa|     2|  Marketing|
|102| Lisa|     2|      Sales|
|104| Evan|     4|Engineering|
|104| Evan|     4|  Marketing|
|104| Evan|     4|      Sales|
|106|  Amy|     4|Engineering|
|106|  Amy|     4|  Marketing|
|106|  Amy|     4|      Sales|
+---+-----+------+-----------|
```

LIMIT 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该`LIMIT`子句用于限制该语`SELECT`句返回的行数。通常，此子句与一起使用是`ORDER BY`为了确保结果具有确定性。

语法

```
LIMIT { ALL | integer_expression }
```

参数

- 全部

  如果指定，则查询将返回所有行。换句话说，如果指定了此选项，则不应用任何限制。

- 整数表达式

  指定返回整数的可折叠表达式。

示例

```
CREATE TABLE person (name STRING, age INT);
INSERT INTO person VALUES
('Jane Doe', 25),
('Pat C', 18),
('Nikki W', 16),
('John D', 25),
('Juan L', 18),
('Jorge S', 16);

-- Select the first two rows.
SELECT name, age FROM person ORDER BY name LIMIT 2;
+-------+---+
|   name|age|
+-------+---+
|  Pat C| 18|
|Jorge S| 16|
+------+---+

-- Specifying ALL option on LIMIT returns all the rows.
SELECT name, age FROM person ORDER BY name LIMIT ALL;
+--------+---+
|    name|age|
+--------+---+
|   Pat C| 18|
| Jorge S| 16|
|  Juan L| 18|
|  John D| 25|
| Nikki W| 16|
|Jane Doe| 25|
+--------+---+

-- A function expression as an input to LIMIT.
SELECT name, age FROM person ORDER BY name LIMIT length('OPENSEARCH');
+-------+---+
|   name|age|
+-------+---+
|  Pat C| 18|
```

```
|Jorge S| 16|
| Juan L| 18|
| John D| 25|
|Nikki W| 16|
+-------+---+
```

案例条款

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该CASE子句使用规则根据指定条件返回特定的结果，类似于其他编程语言中的 if/else 语句。

语法

```
CASE [ expression ] { WHEN boolean_expression THEN then_expression } [ ... ]
[ ELSE else_expression ]
END
```

参数

- 布尔表达式

  指定任何计算结果类型为 boolean 的表达式。

  可以使用逻辑运算符 (AND,OR) 将两个或多个表达式组合在一起。
- 然后_表达式

  根据布尔表达式条件指定 then 表达式。

  `then_expression`并且都`else_expression`应该是相同类型或可强制转换为普通类型。
- else 表达式

  指定默认表达式。

  `then_expression`并且都 `else_expression`应该是相同类型或可强制转换为普通类型。

示例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
(300, 'Mike', 80),
(400, 'Dan', 50);
SELECT id, CASE WHEN id > 200 THEN 'bigger' ELSE 'small' END FROM person;
+------+--------------------------------------------------+
|  id  | CASE WHEN (id > 200) THEN bigger ELSE small END  |
+------+--------------------------------------------------+
| 100  | small                                            |
| 200  | small                                            |
| 300  | bigger                                           |
| 400  | bigger                                           |
+------+--------------------------------------------------+
SELECT id, CASE id WHEN 100 then 'bigger' WHEN  id > 300 THEN '300' ELSE 'small' END
 FROM person;
+------
+------------------------------------------------------------------------------------------------------
+
|  id  | CASE WHEN (id = 100) THEN bigger WHEN (id = CAST((id > 300) AS INT)) THEN 300
 ELSE small END  |
+------
+------------------------------------------------------------------------------------------------------
+
| 100  | bigger
                |
| 200  | small
                |
| 300  | small
                |
| 400  | small
                |
+------
+------------------------------------------------------------------------------------------------------
+
```

公用表表达式

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

公用表表达式 (CTE) 定义了一个临时结果集，用户可以在 SQL 语句的范围内多次引用该结果集。CTE 主要用于SELECT语句中。

语法

```
WITH common_table_expression [ , ... ]
```

Wh `common_table_expression` ile 被定义为：

```
Syntexpression_name [ ( column_name [ , ... ] ) ] [ AS ] ( query )
```

参数

- 表达式名称

  指定公用表表达式的名称。
- query

  一份SELECT声明。

示例

```
-- CTE with multiple column aliases
WITH t(x, y) AS (SELECT 1, 2)
SELECT * FROM t WHERE x = 1 AND y = 2;
+---+---+
|  x|  y|
+---+---+
|  1|  2|
+---+---+

-- CTE in CTE definition
```

```
WITH t AS (
WITH t2 AS (SELECT 1)
SELECT * FROM t2
)
SELECT * FROM t;
+---+
|  1|
+---+
|  1|
+---+

-- CTE in subquery
SELECT max(c) FROM (
WITH t(c) AS (SELECT 1)
SELECT * FROM t
);
+------+
|max(c)|
+------+
|     1|
+------+

-- CTE in subquery expression
SELECT (
WITH t AS (SELECT 1)
SELECT * FROM t
);
+----------------+
|scalarsubquery()|
+----------------+
|               1|
+----------------+

-- CTE in CREATE VIEW statement
CREATE VIEW v AS
WITH t(a, b, c, d) AS (SELECT 1, 2, 3, 4)
SELECT * FROM t;
SELECT * FROM v;
+---+---+---+---+
|  a|  b|  c|  d|
+---+---+---+---+
|  1|  2|  3|  4|
+---+---+---+---+
```

EXPLAIN

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该EXPLAIN语句用于为输入语句提供逻辑/物理计划。默认情况下，此条款仅提供有关实体计划的信息。

语法

```
EXPLAIN [ EXTENDED | CODEGEN | COST | FORMATTED ] statement
```

参数

- 扩展

  生成已解析的逻辑计划、分析的逻辑计划、优化的逻辑计划和物理计划。

  已解析的逻辑计划是从查询中提取的未解析计划。

  分析后的逻辑计划会unresolvedRelation将unresolvedAttribute和转换为完全类型的对象。

  优化的逻辑计划通过一组优化规则进行转换，从而生成物理计划。
- 代码生成

  为报表（如果有）和实体计划生成代码。
- 成本

  如果计划节点统计数据可用，则生成逻辑计划和统计信息。
- 格式化

  生成两个部分：物理计划大纲和节点详细信息。
- 语句

  指定要解释的 SQL 语句。

示例

```
-- Default Output
EXPLAIN select k, sum(v) from values (1, 2), (1, 3) t(k, v) group by k;
+----------------------------------------------------+
|                                                plan|
+----------------------------------------------------+
| == Physical Plan ==
*(2) HashAggregate(keys=[k#33], functions=[sum(cast(v#34 as bigint))])
+- Exchange hashpartitioning(k#33, 200), true, [id=#59]
+- *(1) HashAggregate(keys=[k#33], functions=[partial_sum(cast(v#34 as bigint))])
+- *(1) LocalTableScan [k#33, v#34]
|
+----------------------------------------------------

-- Using Extended
EXPLAIN EXTENDED select k, sum(v) from values (1, 2), (1, 3) t(k, v) group by k;
+----------------------------------------------------+
|                                                plan|
+----------------------------------------------------+
| == Parsed Logical Plan ==
'Aggregate ['k], ['k, unresolvedalias('sum('v), None)]
 +- 'SubqueryAlias `t`
+- 'UnresolvedInlineTable [k, v], [List(1, 2), List(1, 3)]

 == Analyzed Logical Plan ==
 k: int, sum(v): bigint
 Aggregate [k#47], [k#47, sum(cast(v#48 as bigint)) AS sum(v)#50L]
 +- SubqueryAlias `t`
    +- LocalRelation [k#47, v#48]

 == Optimized Logical Plan ==
 Aggregate [k#47], [k#47, sum(cast(v#48 as bigint)) AS sum(v)#50L]
 +- LocalRelation [k#47, v#48]

 == Physical Plan ==
 *(2) HashAggregate(keys=[k#47], functions=[sum(cast(v#48 as bigint))], output=[k#47,
 sum(v)#50L])
+- Exchange hashpartitioning(k#47, 200), true, [id=#79]
   +- *(1) HashAggregate(keys=[k#47], functions=[partial_sum(cast(v#48 as bigint))],
 output=[k#47, sum#52L])
    +- *(1) LocalTableScan [k#47, v#48]
|
+----------------------------------------------------+
```

```
-- Using Formatted
EXPLAIN FORMATTED select k, sum(v) from values (1, 2), (1, 3) t(k, v) group by k;
+----------------------------------------------------+
|                                                plan|
+----------------------------------------------------+
| == Physical Plan ==
 * HashAggregate (4)
 +- Exchange (3)
    +- * HashAggregate (2)
       +- * LocalTableScan (1)


 (1) LocalTableScan [codegen id : 1]
 Output: [k#19, v#20]

 (2) HashAggregate [codegen id : 1]
 Input: [k#19, v#20]

 (3) Exchange
 Input: [k#19, sum#24L]

 (4) HashAggregate [codegen id : 2]
 Input: [k#19, sum#24L]
|
+----------------------------------------------------+
```

横向子查询子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

LATERAL SUBQUERY是前面有关键字的子查询。LATERAL它提供了一种引用前面FROM子句中的列的方法。如果没有LATERAL关键字，子查询只能引用外部查询中的列，而不能引用FROM子句中的列。LATERAL SUBQUERY使复杂的查询更简单、更高效。

语法

```
[ LATERAL ] primary_relation [ join_relation ]
```

参数

- 主要关系

  指定主要关系。它可以是下列项之一：

  1. 表格关系
  2. 别名查询

     语法：`( query ) [ [ AS ] alias ]`
  3. 别名关系

     `Syntax: ( relation ) [ [ AS ] alias ]`

示例

```
CREATE TABLE t1 (c1 INT, c2 INT);
INSERT INTO t1 VALUES (0, 1), (1, 2);
CREATE TABLE t2 (c1 INT, c2 INT);
INSERT INTO t2 VALUES (0, 2), (0, 3);
SELECT * FROM t1,
LATERAL (SELECT * FROM t2 WHERE t1.c1 = t2.c1);
+--------+-------+--------+-------+
|  t1.c1 | t1.c2 |  t2.c1 | t2.c2 |
+-------+--------+--------+-------+
|    0   |   1   |    0   |   3   |
|    0   |   1   |    0   |   2   |
+-------+--------+--------+-------+
SELECT a, b, c FROM t1,
LATERAL (SELECT c1 + c2 AS a),
LATERAL (SELECT c1 - c2 AS b),
LATERAL (SELECT a * b AS c);
+--------+-------+--------+
|    a   |   b   |    c   |
+-------+--------+--------+
|    3   |  -1   |   -3   |
|    1   |  -1   |   -1   |
+-------+--------+--------+
```

横向视图条款

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该LATERAL VIEW子句与生成器函数（例如）结合使用EXPLODE，后者将生成包含一行或多行的虚拟表。 LATERAL VIEW会将这些行应用于每个原始输出行。

语法

```
LATERAL VIEW [ OUTER ] generator_function ( expression [ , ... ] ) [ table_alias ] AS
 column_alias [ , ... ]
```

参数

- 外部

  如果OUTER指定，则如果输入为空 array/map 则返回 null，或者返回 null。
- 生成器函数

  指定生成器函数（EXPLODEINLINE、等。 ）。
- 表别名

  的别名generator_function，这是可选的。
- 列别名

  列出了的列别名generator_function，这些别名可以在输出行中使用。

  如果有多个输出列，则可以generator_function有多个别名。

示例

```
CREATE TABLE person (id INT, name STRING, age INT, class INT, address STRING);
INSERT INTO person VALUES
(100, 'John', 30, 1, 'Street 1'),
(200, 'Mary', NULL, 1, 'Street 2'),
(300, 'Mike', 80, 3, 'Street 3'),
```

```
(400, 'Dan', 50, 4, 'Street 4');
SELECT * FROM person
LATERAL VIEW EXPLODE(ARRAY(30, 60)) tableName AS c_age
LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age;
+------+-------+-------+--------+-----------+--------+--------+
|  id  | name  |  age  | class  |  address  | c_age  | d_age  |
+------+-------+-------+--------+-----------+--------+--------+
| 100  | John  | 30    | 1      | Street 1  | 30     | 40     |
| 100  | John  | 30    | 1      | Street 1  | 30     | 80     |
| 100  | John  | 30    | 1      | Street 1  | 60     | 40     |
| 100  | John  | 30    | 1      | Street 1  | 60     | 80     |
| 200  | Mary  | NULL  | 1      | Street 2  | 30     | 40     |
| 200  | Mary  | NULL  | 1      | Street 2  | 30     | 80     |
| 200  | Mary  | NULL  | 1      | Street 2  | 60     | 40     |
| 200  | Mary  | NULL  | 1      | Street 2  | 60     | 80     |
| 300  | Mike  | 80    | 3      | Street 3  | 30     | 40     |
| 300  | Mike  | 80    | 3      | Street 3  | 30     | 80     |
| 300  | Mike  | 80    | 3      | Street 3  | 60     | 40     |
| 300  | Mike  | 80    | 3      | Street 3  | 60     | 80     |
| 400  | Dan   | 50    | 4      | Street 4  | 30     | 40     |
| 400  | Dan   | 50    | 4      | Street 4  | 30     | 80     |
| 400  | Dan   | 50    | 4      | Street 4  | 60     | 40     |
| 400  | Dan   | 50    | 4      | Street 4  | 60     | 80     |
+------+-------+-------+--------+-----------+--------+--------+
SELECT c_age, COUNT(1) FROM person
LATERAL VIEW EXPLODE(ARRAY(30, 60)) AS c_age
LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age
GROUP BY c_age;
+--------+-----------+
| c_age  | count(1)  |
+--------+-----------+
| 60     | 8         |
| 30     | 8         |
+--------+-----------+
SELECT * FROM person
LATERAL VIEW EXPLODE(ARRAY()) tableName AS c_age;
+-----+-------+------+--------+----------+--------+
| id  | name  | age  | class  | address  | c_age  |
+-----+-------+------+--------+----------+--------+
+-----+-------+------+--------+----------+--------+
SELECT * FROM person
LATERAL VIEW OUTER EXPLODE(ARRAY()) tableName AS c_age;
+------+-------+-------+--------+-----------+--------+
|  id  | name  |  age  | class  |  address  | c_age  |
```

```
+------+-------+-------+--------+-----------+--------+
| 100  | John  | 30    | 1      | Street 1  | NULL   |
| 200  | Mary  | NULL  | 1      | Street 2  | NULL   |
| 300  | Mike  | 80    | 3      | Street 3  | NULL   |
| 400  | Dan   | 50    | 4      | Street 4  | NULL   |
+------+-------+-------+--------+-----------+--------+
```

LIKE 谓词

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

LIKE谓词用于搜索特定的模式。此谓词还支持多种模式，其量词包括ANYSOME、和。ALL

语法

```
[ NOT ] { LIKE search_pattern [ ESCAPE esc_char ] | [ RLIKE | REGEXP ] regex_pattern }
[ NOT ] { LIKE quantifiers ( search_pattern [ , ... ]) }
```

参数

- 搜索模式

  指定 LIKE 子句要搜索的字符串模式。它可以包含特殊的模式匹配字符：

  - %匹配零个或多个字符。
  - _恰好匹配一个字符。
- esc_char

  指定转义字符。默认的转义字符是\。
- regex_pattern

  指定要由 o RLIKE r REGEXP 子句搜索的正则表达式搜索模式。
- 量词

  指定谓词量词包括ANY、SOME和。ALL

  ANY或SOME表示如果其中一个模式与输入匹配，则返回 true。

ALL表示如果所有模式都与输入相匹配，则返回 true。

示例

```
CREATE TABLE person (id INT, name STRING, age INT);
INSERT INTO person VALUES
(100, 'John', 30),
(200, 'Mary', NULL),
(300, 'Mike', 80),
(400, 'Dan',  50),
(500, 'Evan_w', 16);
SELECT * FROM person WHERE name LIKE 'M%';
+---+----+----+
| id|name| age|
+---+----+----+
|300|Mike|  80|
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name LIKE 'M_ry';
+---+----+----+
| id|name| age|
+---+----+----+
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name NOT LIKE 'M_ry';
+---+------+---+
| id|  name|age|
+---+------+---+
|500|Evan_W| 16|
|300|  Mike| 80|
|100|  John| 30|
|400|   Dan| 50|
+---+------+---+
SELECT * FROM person WHERE name RLIKE 'M+';
+---+----+----+
| id|name| age|
+---+----+----+
|300|Mike|  80|
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name REGEXP 'M+';
+---+----+----+
```

```
| id|name| age|
+---+----+----+
|300|Mike|  80|
|200|Mary|null|
+---+----+----+
SELECT * FROM person WHERE name LIKE '%\_%';
+---+------+---+
| id|  name|age|
+---+------+---+
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name LIKE '%$_%' ESCAPE '$';
+---+------+---+
| id|  name|age|
+---+------+---+
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name LIKE ALL ('%an%', '%an');
+---+----+----+
| id|name| age|
+---+----+----+
|400| Dan|  50|
+---+----+----+
SELECT * FROM person WHERE name LIKE ANY ('%an%', '%an');
+---+------+---+
| id|  name|age|
+---+------+---+
|400|   Dan| 50|
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name LIKE SOME ('%an%', '%an');
+---+------+---+
| id|  name|age|
+---+------+---+
|400|   Dan| 50|
|500|Evan_W| 16|
+---+------+---+
SELECT * FROM person WHERE name NOT LIKE ALL ('%an%', '%an');
+---+----+----+
| id|name| age|
+---+----+----+
|100|John|  30|
|200|Mary|null|
|300|Mike|  80|
```

```
+---+----+----+
SELECT * FROM person WHERE name NOT LIKE ANY ('%an%', '%an');
+---+------+----+
| id|  name| age|
+---+------+----+
|100|  John|  30|
|200|  Mary|null|
|300|  Mike|  80|
|500|Evan_W|  16|
+---+------+----+
SELECT * FROM person WHERE name NOT LIKE SOME ('%an%', '%an');
+---+------+----+
| id|  name| age|
+---+------+----+
|100|  John|  30|
|200|  Mary|null|
|300|  Mike|  80|
|500|Evan_W|  16|
+---+------+----+
```

OFFSET

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该`OFFSET`子句用于指定在开始返回`SELECT`语句返回的行之前要跳过的行数。通常，此子句与一起使用是`ORDER BY`为了确保结果具有确定性。

语法

```
OFFSET integer_expression
```

参数

- 整数表达式

  指定返回整数的可折叠表达式。

示例

```
CREATE TABLE person (name STRING, age INT);
INSERT INTO person VALUES
('Jane Doe', 25),
('Pat C', 18),
('Nikki W', 16),
('Juan L', 25),
('John D', 18),
('Jorge S', 16);

-- Skip the first two rows.
SELECT name, age FROM person ORDER BY name OFFSET 2;
+-------+---+
|   name|age|
+-------+---+
| John D| 18|
| Juan L| 25|
|Nikki W| 16|
|Jane Doe| 25|
+-------+---+

-- Skip the first two rows and returns the next three rows.
SELECT name, age FROM person ORDER BY name LIMIT 3 OFFSET 2;
+-------+---+
|   name|age|
+-------+---+
| John D| 18|
| Juan L| 25|
|Nikki W| 16|
+-------+---+

-- A function expression as an input to OFFSET.
SELECT name, age FROM person ORDER BY name OFFSET length('WAGON');
+-------+---+
|   name|age|
+-------+---+
|Jane Doe| 25|
+-------+---+
```

PIVOT 子句

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该PIVOT子句用于数据透视。我们可以根据特定的列值获取聚合值，这些值将转换为SELECT子句中使用的多列。该PIVOT子句可以在表名或子查询之后指定。

语法

```
PIVOT ( { aggregate_expression [ AS aggregate_expression_alias ] } [ , ... ] FOR
 column_list IN ( expression_list ) )
```

参数

- aggregate_expression

  指定聚合表达式(SUM(a)COUNT(DISTINCT b)、等。)。
- 聚合表达式别名

  为聚合表达式指定别名。
- column_list

  包含FROM子句中的列，该子句指定要用新列替换的列。您可以使用方括号将各列括起来，例如(c1, c2)。
- expression_list

  指定新列，这些列用于匹配中的值column_list作为聚合条件。您也可以为它们添加别名。

示例

```
CREATE TABLE person (id INT, name STRING, age INT, class INT, address STRING);
INSERT INTO person VALUES
(100, 'John', 30, 1, 'Street 1'),
(200, 'Mary', NULL, 1, 'Street 2'),
(300, 'Mike', 80, 3, 'Street 3'),
(400, 'Dan', 50, 4, 'Street 4');
```

```
SELECT * FROM person
PIVOT (
SUM(age) AS a, AVG(class) AS c
FOR name IN ('John' AS john, 'Mike' AS mike)
);
+------+-----------+---------+---------+---------+---------+
|  id  |  address  | john_a  | john_c  | mike_a  | mike_c  |
+------+-----------+---------+---------+---------+---------+
| 200  | Street 2  | NULL    | NULL    | NULL    | NULL    |
| 100  | Street 1  | 30      | 1.0     | NULL    | NULL    |
| 300  | Street 3  | NULL    | NULL    | 80      | 3.0     |
| 400  | Street 4  | NULL    | NULL    | NULL    | NULL    |
+------+-----------+---------+---------+---------+---------+
SELECT * FROM person
PIVOT (
SUM(age) AS a, AVG(class) AS c
FOR (name, age) IN (('John', 30) AS c1, ('Mike', 40) AS c2)
);
+------+-----------+-------+-------+-------+-------+
|  id  |  address  | c1_a  | c1_c  | c2_a  | c2_c  |
+------+-----------+-------+-------+-------+-------+
| 200  | Street 2  | NULL  | NULL  | NULL  | NULL  |
| 100  | Street 1  | 30    | 1.0   | NULL  | NULL  |
| 300  | Street 3  | NULL  | NULL  | NULL  | NULL  |
| 400  | Street 4  | NULL  | NULL  | NULL  | NULL  |
+------+-----------+-------+-------+-------+-------+
```

集合运算符

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

集合运算符用于将两个输入关系合并为一个输入关系。 OpenSearch SQL 支持三种类型的集合运算符：

- EXCEPT 或 MINUS
- INTERSECT
- UNION

输入关系的列数必须相同，且相应列的数据类型必须兼容。

除了

EXCEPT并EXCEPT ALL返回在一个关系中找到但未在另一个关系中找到的行。 EXCEPT（或者EXCEPT DISTINCT）只获取不同的行，而EXCEPT ALL不会从结果行中删除重复的行。请注意，MINUS这是的别名EXCEPT。

语法

```
[ ( ] relation [ ) ] EXCEPT | MINUS [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

示例

```
-- Use table1 and table2 tables to demonstrate set operators in this page.
SELECT * FROM table1;
+---+
|  c|
+---+
|  3|
|  1|
|  2|
|  2|
|  3|
|  4|
+---+
SELECT * FROM table2;
+---+
|  c|
+---+
|  5|
|  1|
|  2|
|  2|
+---+
SELECT c FROM table1 EXCEPT SELECT c FROM table2;
+---+
|  c|
+---+
|  3|
|  4|
+---+
SELECT c FROM table1 MINUS SELECT c FROM table2;
```

```
+---+
|  c|
+---+
|  3|
|  4|
+---+
SELECT c FROM table1 EXCEPT ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  3|
|  3|
|  4|
+---+
SELECT c FROM table1 MINUS ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  3|
|  3|
|  4|
+---+
```

相交

INTERSECT并INTERSECT ALL返回在两个关系中找到的行。 INTERSECT（或者INTERSECT DISTINCT）只获取不同的行，而INTERSECT ALL不会从结果行中删除重复的行。

语法

```
[ ( ] relation [ ) ] INTERSECT [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

示例

```
(SELECT c FROM table1) INTERSECT (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  2|
+---+
(SELECT c FROM table1) INTERSECT DISTINCT (SELECT c FROM table2);
```

```
+---+
|  c|
+---+
|  1|
|  2|
+---+
(SELECT c FROM table1) INTERSECT ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  2|
|  2|
+---+
```

联盟

UNION并UNION ALL返回在任一关系中找到的行。 UNION（或者UNION DISTINCT）只获取不同的行，而UNION ALL不会从结果行中删除重复的行。

语法

```
[ ( ] relation [ ) ] UNION [ ALL | DISTINCT ] [ ( ] relation [ ) ]
```

示例

```
(SELECT c FROM table1) UNION (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  3|
|  5|
|  4|
|  2|
+---+
(SELECT c FROM table1) UNION DISTINCT (SELECT c FROM table2);
+---+
|  c|
+---+
|  1|
|  3|
```

```
|  5|
|  4|
|  2|
+---+
SELECT c FROM table1 UNION ALL (SELECT c FROM table2);
+---+
|  c|
+---+
|  3|
|  1|
|  2|
|  2|
|  3|
|  4|
|  5|
|  1|
|  2|
|  2|
+---+
```

按子句排序

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该`SORT BY`子句用于返回按用户指定顺序在每个分区内排序的结果行。当有多个分区时，`SORT BY`可能会返回部分排序的结果。这与保证输出总顺序的`ORDER BY`子句不同。

语法

```
SORT BY { expression [ sort_direction | nulls_sort_order ] [ , ... ] }
```

参数

- 排序依据

  指定以逗号分隔的表达式列表以及用于对每个分区内的行进行排序的可选参数 sort_direction 和 nulls_sort_order。

- 排序方向

  （可选）指定是按升序还是降序对行进行排序。

  排序方向的有效值ASC为升序和降序DESC。

  如果未明确指定排序方向，则默认情况下，行按升序排序。

  语法：`[ ASC | DESC ]`

- nulls_sort_order

  （可选）指定是否在非 NULL 值之前/之后返回 NULL 值。

  如果`null_sort_order`未指定，则如果 NULLs 排序顺序为，则先排序ASC；如果排序顺序为DESC，则最后排序 NULLS。

  1. 如果`NULLS FIRST`已指定，则无论排序顺序如何，都将首先返回 NULL 值。

  2. 如果`NULLS LAST`已指定，则无论排序顺序如何，最后都会返回 NULL 值。

  语法：`[ NULLS { FIRST | LAST } ]`

示例

```
CREATE TABLE person (zip_code INT, name STRING, age INT);
INSERT INTO person VALUES
(94588, 'Shirley Rodriguez', 50),
(94588, 'Juan Li', 18),
(94588, 'Anil K', 27),
(94588, 'John D', NULL),
(94511, 'David K', 42),
(94511, 'Aryan B.', 18),
(94511, 'Lalit B.', NULL);
-- Sort rows by `name` within each partition in ascending manner
SELECT name, age, zip_code FROM person SORT BY name;
+------------------+----+--------+
|              name| age|zip_code|
+------------------+----+--------+
|            Anil K|  27|   94588|
|           Juan Li|  18|   94588|
|            John D|null|   94588|
| Shirley Rodriguez|  50|   94588|
```

```
|          Aryan B.|  18|   94511|
|           David K|  42|   94511|
|          Lalit B.|null|   94511|
+------------------+----+--------+
-- Sort rows within each partition using column position.
SELECT name, age, zip_code FROM person SORT BY 1;
+----------------+----+----------+
|              name| age|zip_code|
+------------------+----+--------+
|            Anil K|  27|   94588|
|           Juan Li|  18|   94588|
|            John D|null|   94588|
| Shirley Rodriguez|  50|   94588|
|          Aryan B.|  18|   94511|
|           David K|  42|   94511|
|          Lalit B.|null|   94511|
+------------------+----+--------+

-- Sort rows within partition in ascending manner keeping null values to be last.
SELECT age, name, zip_code FROM person SORT BY age NULLS LAST;
+----+------------------+--------+
| age|              name|zip_code|
+----+------------------+--------+
|  18|           Juan Li|   94588|
|  27|            Anil K|   94588|
|  50| Shirley Rodriguez|   94588|
|null|            John D|   94588|
|  18|          Aryan B.|   94511|
|  42|           David K|   94511|
|null|          Lalit B.|   94511|
+--------------+--------+--------+

-- Sort rows by age within each partition in descending manner, which defaults to NULL
 LAST.
SELECT age, name, zip_code FROM person SORT BY age DESC;
+----+------------------+--------+
| age|              name|zip_code|
+----+------------------+--------+
|  50|          Shirley Rodriguez|   94588|
|  27|            Anil K|   94588|
|  18|           Juan Li|   94588|
|null|            John D|   94588|
|  42|           David K|   94511|
|  18|          Aryan B.|   94511|
```

```
|null|          Lalit B.|   94511|
+----+------------------+--------+

-- Sort rows by age within each partition in descending manner keeping null values to
 be first.
SELECT age, name, zip_code FROM person SORT BY age DESC NULLS FIRST;
+----+------------------+--------+
| age|              name|zip_code|
+----+------------------+--------+
|null|            John D|   94588|
|  50| Shirley Rodriguez|   94588|
|  27|            Anil K|   94588|
|  18|           Juan Li|   94588|
|null|          Lalit B.|   94511|
|  42|           David K|   94511|
|  18|          Aryan B.|   94511|
+--------------+--------+--------+

-- Sort rows within each partition based on more than one column with each column
 having
-- different sort direction.
SELECT name, age, zip_code FROM person
SORT BY name ASC, age DESC;
+------------------+----+--------+
|              name| age|zip_code|
+------------------+----+--------+
|            Anil K|  27|   94588|
|           Juan Li|  18|   94588|
|            John D|null|   94588|
| Shirley Rodriguez|  50|   94588|
|          Aryan B.|  18|   94511|
|           David K|  42|   94511|
|          Lalit B.|null|   94511|
+------------------+----+--------+
```

UNPIVOT

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 SQL 命令，请参阅the section called “支持的 SQL 命令”。

该UNPIVOT子句将多列转换为子SELECT句中使用的多行。该UNPIVOT子句可以在表名或子查询之后指定。

语法

```
UNPIVOT [ { INCLUDE | EXCLUDE } NULLS ] (
    { single_value_column_unpivot | multi_value_column_unpivot }
) [[AS] alias]

single_value_column_unpivot:
    values_column
    FOR name_column
    IN (unpivot_column [[AS] alias] [, ...])

multi_value_column_unpivot:
    (values_column [, ...])
    FOR name_column
    IN ((unpivot_column [, ...]) [[AS] alias] [, ...])
```

参数

- 取消透视列

  包含子句中的列，该FROM子句指定了我们要取消透视的列。
- 名称_列

  包含未转置列名称的列的名称。
- 值列

  保存未旋转列值的列的名称。

示例

```
CREATE TABLE sales_quarterly (year INT, q1 INT, q2 INT, q3 INT, q4 INT);
INSERT INTO sales_quarterly VALUES
(2020, null, 1000, 2000, 2500),
(2021, 2250, 3200, 4200, 5900),
(2022, 4200, 3100, null, null);
-- column names are used as unpivot columns
SELECT * FROM sales_quarterly
UNPIVOT (
```

```
sales FOR quarter IN (q1, q2, q3, q4)
);
+------+---------+-------+
| year | quarter | sales |
+------+---------+-------+
| 2020 | q2      | 1000  |
| 2020 | q3      | 2000  |
| 2020 | q4      | 2500  |
| 2021 | q1      | 2250  |
| 2021 | q2      | 3200  |
| 2021 | q3      | 4200  |
| 2021 | q4      | 5900  |
| 2022 | q1      | 4200  |
| 2022 | q2      | 3100  |
+------+---------+-------+
-- NULL values are excluded by default, they can be included
-- unpivot columns can be alias
-- unpivot result can be referenced via its alias
SELECT up.* FROM sales_quarterly
UNPIVOT INCLUDE NULLS (
sales FOR quarter IN (q1 AS Q1, q2 AS Q2, q3 AS Q3, q4 AS Q4)
) AS up;
+------+---------+-------+
| year | quarter | sales |
+------+---------+-------+
| 2020 | Q1      | NULL  |
| 2020 | Q2      | 1000  |
| 2020 | Q3      | 2000  |
| 2020 | Q4      | 2500  |
| 2021 | Q1      | 2250  |
| 2021 | Q2      | 3200  |
| 2021 | Q3      | 4200  |
| 2021 | Q4      | 5900  |
| 2022 | Q1      | 4200  |
| 2022 | Q2      | 3100  |
| 2022 | Q3      | NULL  |
| 2022 | Q4      | NULL  |
+------+---------+-------+
-- multiple value columns can be unpivoted per row
SELECT * FROM sales_quarterly
UNPIVOT EXCLUDE NULLS (
(first_quarter, second_quarter)
FOR half_of_the_year IN (
(q1, q2) AS H1,
```

```
(q3, q4) AS H2
)
);
+------+------------------+---------------+----------------+
|  id  | half_of_the_year | first_quarter | second_quarter |
+------+------------------+---------------+----------------+
| 2020 | H1               | NULL          | 1000           |
| 2020 | H2               | 2000          | 2500           |
| 2021 | H1               | 2250          | 3200           |
| 2021 | H2               | 4200          | 5900           |
| 2022 | H1               | 4200          | 3100           |
+------+------------------+---------------+----------------+
```

支持的 PPL 命令

下表显示了 OpenSearch 控制面板支持哪些用于查询 CloudWatch 日志、Amazon S3 或安全湖的 PPL 命令，以及 L CloudWatch ogs Insights 支持哪些命令。 CloudWatch Logs Insights 在查询 CloudWatch 日志时使用与 OpenSearch 仪表板相同的 PPL 语法，并且这些表将两者都称为 CloudWatch 日志。

> **Note**
>
> 在 S OpenSearch ervice 之外分析数据时，命令的执行方式可能与对 OpenSearch 索引执行的命令不同。

主题

- 命令
- 函数
- 为使用 OpenSearch PPL 的 CloudWatch Logs Insights 用户提供的其他信息

命令

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “fields”	显示一组需要投影的字段。	支持的	支持的	支持的	`fields field1, field2`
the section called “其中”	根据您指定的条件筛选数据。	支持的	支持的	支持的	`where field1="success" | where field2 != "i-023fe0a90929d8822" | fields field3, col4, col5, col6 | head 1000`
the section called “stats”	执行聚合和计算。	支持的	支持的	支持的	`stats count(), count(`field1`), min(`field1`), max(`field1`), avg(`field1`) by field2`

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
					`| head 1000`
the section called “parse”	从字符串中提取正则表达式（regex）模式并显示提取的模式。提取的模式可以进一步用于创建新字段或筛选数据。	支持的	支持的	支持的	`parse `field1` ".*/(?<field2>[^/]+$)" | where field2 = "requestId" | fields field2, `field2` | head 1000`
the section called “模式”	从文本字段中提取日志模式并将结果附加到搜索结果中。按模式对日志进行分组可以更轻松地从大量日志数据中汇总统计数据，以进行分析和故障排除。	不支持	支持的	支持的	`patterns new_field='no_numbers' pattern='[0-9]' message | fields message, no_numbers`

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “排序”	按字段名对显示的结果进行排序。使用 sor FieldNamet- 按降序排序。	支持的	支持的	支持的	`stats count(), count(`field1`), min(`field1`) as field1Alias, max(`field1`), avg(`field1`) by field2 | sort -field1Alias | head 1000`
the section called “eval”	修改或处理字段的值并将其存储在不同的字段中。这对于以数学方式修改列、将字符串函数应用于列或将日期函数应用于列非常有用。	支持的	支持的	支持的	`eval field2 = `field1` * 2 | fields field1, field2 | head 20`

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “rename”	重命名搜索结果中的一个或多个字段。	支持的	支持的	支持的	`rename field2 as field1 | fields field1`
the section called “head”	将显示的查询结果限制为前 N 行。	支持的	支持的	支持的	`fields `@message` | head 20`
the section called “grok”	使用基于正则表达式的 grok 模式解析文本字段，并将结果附加到搜索结果中。	支持的	支持的	支持的	`grok email '.+@%{HOSTNAME:host}' | fields email`
the section called “top”	查找字段中最常用的值。	支持的	支持的	支持的	`top 2 Field1 by Field2`
the section called “dedup”	根据您指定的字段删除重复的条目。	支持的	支持的	支持的	`dedup field1 | fields field1, field2, field3`

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “join”	将两个数据集连接在一起。	支持的	支持的	支持的	`source=customer | join ON c_custkey = o_custkey orders | head 10`
the section called “查找”	通过添加或替换查询索引（维度表）中的数据来丰富您的搜索数据。您可以使用维度表中的值扩展索引的字段，在匹配查找条件时追加或替换值	不支持	支持的	支持的	`where orderType = 'Cancelled' | lookup account_list, mkt_id AS mkt_code replace amount, account_name as name | stats count(mkt_code), avg(amount) by name`

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “subquery”	在管道处理语言 (PPL) 语句中执行复杂的嵌套查询。	支持的	支持的	支持的	`where id in [ subquery source=users | where user in [ subquery source=actions | where action="login" | fields user ] | fields uid ]`
the section called “罕见的”	查找字段列表中所有字段中频率最低的值。	支持的	支持的	支持的	`rare Field1 by Field2`
the section called “趋势线”	计算字段的移动平均值。	支持的	支持的	支持	`trendline sma(2, field1) as field1Alias`

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “赛事统计”	使用计算出的汇总统计数据丰富您的事件数据。它分析事件中的指定字段，计算各种统计度量，然后将这些结果作为新字段附加到每个原始事件中。	支持（除外count()）	支持	支持	`eventstats sum(field1) by field2`
the section called “flatten”	展平字段，该字段必须是以下类型：`struct<?,?> or array<struct<?,?>>`	支持	支持	支持	`source=table | flatten field1`
the section called “字段摘要”	计算每个字段的基本统计数据（计数、非重复计数、最小值、最大值、平均值、stddev 和平均值）。	支持（每个查询一个字段）	支持	支持	`where field1 != 200 | fieldsummary includefields=field1 nulls=true`

PPL 命令	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “填空”	使用您提供的值填充空字段。它可以用于一个或多个字段。	支持	支持	支持	`fields field1 | eval field2=field1 | fillnull value=0 field1`
the section called “扩展”	将包含多个值的字段分解为单独的行，为指定字段中的每个值创建一个新行。	支持	支持	支持	`expand employee | stats max(salary) as max by state, company`
the section called “描述”	获取有关表、架构和目录的结构和元数据的详细信息	不支持	支持	支持	`describe schema.table`

函数

PPL 功能	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “字符串” (CONCAT, CONCAT_WS ，LENGTH, LOWER, LTRIM, POSITION,	PPL 中的内置函数，可以在 PPL 查询中操作和转换字符串和文	支持	支持	支持	`eval col1Len = LENGTH(col1) | fields col1Len`

PPL 功能	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
REVERSE, RIGHT, RTRIM, SUBSTRING , TRIM, UPPER)	本数据。例如，转换大小写、合并字符串、提取部分和清理文本。				

PPL 功能	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “日期和时间” (DAY, DAYOFMONTH , DAY_OF_MONTH ,DAYOFWEEK , DAY_OF_WEEK , DAYOFYEAR , DAY_OF_YEAR , DAYNAME, FROM_UNIXTIME , HOUR, HOUR_OF_DAY , LAST_DAY, LOCALTIMESTAMP , LOCALTIME , MAKE_DATE , MINUTE, MINUTE_OF_HOUR , MONTH, MONTHNAME , MONTH_OF_YEAR , NOW, QUARTER, SECOND, SECOND_OF_MINUTE , SUBDATE, SYSDATE, TIMESTAMP , UNIX_TIMESTAMP , WEEK, WEEKDAY, WEEK_OF_YEAR , DATE_ADD,	用于在 PPL 查询中处理和转换日期和时间戳数据的内置函数。例如，date_ad date_format、datediff 和当前日期。	支持	支持	支持	`eval newDate = ADDDATE(DATE('2020-08-26'), 1) | fields newDate`

PPL 功能	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
DATE_SUB, TIMESTAMP ADD ，TIMESTAMP DIFF ，UTC_TIMESTAMP ，CURRENT_TIMEZONE)					
the section called “状况” (EXISTS, IF, IFNULL, ISNOTNULL ， ISNULL, NULLIF)	内置函数，可对多行执行计算以生成单个汇总值。例如，总和、计数、平均值、最大值和最小值。	支持	支持	支持	`eval field2 = isnull(col1) | fields field2, col1, field3`

PPL 功能	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “数学” (ABS, ACOS, ASIN, ATAN, ATAN2, CEIL, CEILING, CONV, COS, COT, CRC32, DEGREES, E, EXP, FLOOR, LN, LOG, LOG2, LOG10, MOD, PI. POW, POWER, RADIANS, RAND, ROUND, SIGN, SIN, SQRT, CBRT)	用于在 PPL 查询中执行数学计算和转换的内置函数。例如：abs（绝对值）、四舍五入（四舍五入数字）、sqrt（方根）、pow（方计算）和 ceil（向上舍入到最接近的整数）。	支持	支持	支持	`eval field2 = ACOS(col1) | fields col1`
the section called “Expressions” （算术运算符 (+、、*) -、谓词运算符 (>. <、) IN)	表达式的内置函数，尤其是值表达式，返回标量值。表达式有不同的类型和形式。	支持	支持	支持	`where age > (25 + 5) | fields age`

PPL 功能	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “IP 地址” (CIDRMATCH)	用于处理 IP 地址的内置功能，例如 CIDR。	支持	支持	支持	`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)	用于处理 JSON 的内置函数，包括数组、提取和验证。	支持	支持	支持	`eval `json_extract('{"a":"b"}', '$.a')` = json_extract('{"a":"b"}', '$a')`
the section called “Lambda” (EXISTS, FILTER, REDUCE, TRANSFORM)	用于处理 JSON 的内置函数，包括数组、提取和验证。	不支持	支持	支持	`eval array = json_array(1, -1, 2), result = filter(array, x -> x > 0) | fields result`

PPL 功能	描述	CloudWatch 日志	Amazon S3	Security Lake	示例命令
the section called “密码学” (MD5, SHA1, SHA2)	内置函数允许您生成唯一的数据指纹，这些指纹可用于验证、比较或作为更复杂的安全协议的一部分。	支持	支持	支持	`eval `MD5('hello')` = MD5('hello') | fields `MD5('hello')`

为使用 OpenSearch PPL 的 CloudWatch Logs Insights 用户提供的其他信息

尽管 CloudWatch Logs Insights 支持大多数 OpenSearch PPL 命令和功能，但目前不支持某些命令和功能。例如，它目前不支持 PPL 中的查找命令。自 2025 年 6 月 2 日起，L CloudWatch ogs Insights 现在支持 PPL 中的 JOIN、子查询、Flatten、Fillnull、Expand、Cidrmatch 和 JSON 函数。有关支持的查询命令和函数的完整列表，请参阅上表中的 Ama CloudWatch zon Logs 列。

示例查询和配额

以下内容适用于 CloudWatch Logs Insights OpenSearch 用户和查询 CloudWatch 数据的用户。

有关查询 OpenSearch 服务 CloudWatch 日志时适用的限制的信息，请参阅 Amazon Logs 用户指南中的CloudWatch CloudWatch 日志配额。限制包括您可以查询的 CloudWatch 日志组数量、可以执行的最大并发查询数、最长查询执行时间以及结果中返回的最大行数。无论您使用哪种语言查询 CloudWatch 日志（即 OpenSearch PPL、SQL 和 Logs Insights QL），限制都是一样的。

PPL 命令

主题

- comment
- 关联命令

- 重复数据删除命令
- 描述命令
- eval 命令
- eventstats 命令
- 扩展命令
- 解释命令
- fillnull 命令
- 字段命令
- 扁平化命令
- grok 命令
- head 命令
- 加入命令
- 查找命令
- 解析命令
- 模式命令
- 罕见的命令
- 重命名命令
- 搜索命令
- 排序命令
- stats 命令
- 子查询命令
- top 命令
- 趋势线命令
- 命令在哪里
- 字段摘要
- 扩展命令
- PPL 函数

comment

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

PPL 同时支持行注释和区块注释。系统不评估评论文本。

台词注释

行注释以两个斜线//开头，以新行结尾。

示例：

```
os> source=accounts | top gender // finds most common gender of all the accounts
fetched rows / total rows = 2/2
+----------+
| gender   |
|----------|
| M        |
| F        |
+----------+
```

屏蔽评论

区块评论以斜杠开头，后跟星号\ *，以星号结尾，后跟斜线 */。

示例：

```
os> source=accounts | dedup 2 gender /* dedup the document with gender field keep 2
 duplication */ | fields account_number, gender
fetched rows / total rows = 3/3
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 6                | M        |
| 13               | F        |
+------------------+----------+
```

关联命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

您可以根据共同的维度和时间范围关联不同的数据源。

当你处理来自不同垂直行业的大量数据时，这种关联性至关重要，这些数据共享相同的时间段，但没有正式同步。

通过根据时间范围和相似维度关联这些不同的数据源，您可以丰富数据并发现有价值的见解。

示例

可观测性域有三个不同的数据源：

- 日志
- Metrics
- 跟踪

这些数据源可能共享共同的维度。要从一个数据源过渡到另一个数据源，您需要正确关联它们。使用语义命名约定，您可以识别日志、跟踪和指标中的共享元素。

示例：

```
{
  "@timestamp": "2018-07-02T22:23:00.186Z",
  "aws": {
    "elb": {
      "backend": {
        "http": {
          "response": {
            "status_code": 500
          }
        },
        "ip": "********",
        "port": "80"
      },
      ...
     "target_port": [
```

```
        "10.0.0.1:80"
      ],
      "target_status_code": [
        "500"
      ],
      "traceId": "Root=1-58337262-36d228ad5d99923122bbe354",
      "type": "http"
    }
  },
  "cloud": {
    "provider": "aws"
  },
  "http": {
    "request": {
    ...
  },
  "communication": {
    "source": {
      "address": "**************",
      "ip": "**************",
      "port": 2817
    }
  },
  "traceId": "Root=1-58337262-36d228ad5d99923122bbe354"
}
```

此示例显示了来自驻留在其上的 Amazon服务的 Amazon ELB 日志。它显示状态码为 500 的后端 HTTP 响应，表示存在错误。这可能会触发警报或成为常规监控过程的一部分。下一步是收集有关此事件的相关数据，以便进行彻底调查。

虽然您可能会想查询与时间范围相关的所有数据，但这种方法可能会让人不知所措。你最终可能会得到太多信息，花更多的时间筛选出不相关的数据，而不是找出根本原因。

相反，您可以通过关联来自不同来源的数据来使用更具针对性的方法。您可以使用以下维度进行关联：

- 知识产权-`"ip": "10.0.0.1" | "ip": "**************"`
- 端口-`"port": 2817 | "target_port": "10.0.0.1:80"`

假设您可以访问其他跟踪和指标索引，并且熟悉架构结构，则可以创建更精确的关联查询。

以下是包含您可能想要关联的 HTTP 信息的跟踪索引文档的示例：

```
{
  "traceId": "c1d985bd02e1dbb85b444011f19a1ecc",
  "spanId": "55a698828fe06a42",
  "traceState": [],
  "parentSpanId": "",
  "name": "mysql",
  "kind": "CLIENT",
  "@timestamp": "2021-11-13T20:20:39+00:00",
  "events": [
    {
      "@timestamp": "2021-03-25T17:21:03+00:00",
       ...
    }
  ],
  "links": [
    {
      "traceId": "c1d985bd02e1dbb85b444011f19a1ecc",
      "spanId": "55a698828fe06a42w2",
      },
      "droppedAttributesCount": 0
    }
  ],
  "resource": {
    "service@name": "database",
    "telemetry@sdk@name": "opentelemetry",
    "host@hostname": "ip-172-31-10-8.us-west-2.compute.internal"
  },
  "status": {
    ...
  },
  "attributes": {
    "http": {
      "user_agent": {
        "original": "Mozilla/5.0"
      },
      "network": {
          ...
        }
      },
      "request": {
          ...
        }
      },
```

```
      "response": {
        "status_code": "200",
        "body": {
          "size": 500
        }
      },
      "client": {
        "server": {
          "socket": {
            "address": "***********",
            "domain": "example.com",
            "port": 80
          },
          "address": "***********",
          "port": 80
        },
        "resend_count": 0,
        "url": {
          "full": "http://example.com"
        }
      },
      "server": {
        "route": "/index",
        "address": "***********",
        "port": 8080,
        "socket": {
         ...
        },
        "client": {
         ...
         }
        },
        "url": {
         ...
        }
      }
    }
  }
}
```

在这种方法中，您可以看到可以与 elb 日志关联 client/server `ip`的 http `traceId` 和 http，以便更好地了解系统的行为和状况。

新的关联查询命令

以下是允许进行此类调查的新命令：

```
source alb_logs, traces | where alb_logs.ip="10.0.0.1" AND
 alb_logs.cloud.provider="aws"|
correlate exact fields(traceId, ip) scope(@timestamp, 1D) mapping(alb_logs.ip =
 traces.attributes.http.server.address, alb_logs.traceId = traces.traceId )
```

以下是命令的每个部分的用途：

1. `source alb_logs, traces`-这将选择要关联的数据源。
2. `where ip="10.0.0.1" AND cloud.provider="aws"`-这缩小了您的搜索范围。
3. `correlate exact fields(traceId, ip)`-这告诉系统根据以下字段的精确匹配来关联数据：
   - 该`ip`字段具有明确的筛选条件，因此它将用于所有数据源的关联中。
   - 该`traceId`字段没有显式筛选器，因此它将匹配所有数据源中相同的 TraceID。

字段名称表示关联命令中函数的逻辑含义。实际的连接条件取决于您提供的映射语句。

该术语`exact`意味着关联语句将要求所有字段都匹配才能完成查询语句。

该术语`approximate`将尝试在最佳情况下进行匹配，并且不会拒绝部分匹配的行。

解决不同的字段映射问题

如果同一逻辑字段（例如`ip`）在数据源中具有不同的名称，则需要提供路径字段的显式映射。为了解决这个问题，您可以扩展关联条件以匹配具有相似逻辑含义的不同字段名称。你可以这样做：

```
alb_logs.ip = traces.attributes.http.server.address, alb_logs.traceId = traces.traceId
```

对于参与关联联接的每个字段，您都应提供相关的映射语句，其中包括要通过此关联命令连接的所有表。

示例

在此示例中，有 2 个来源：`alb_logs, traces`

有 2 个字段：`traceId, ip`

有 2 个映射语句：`alb_logs.ip = traces.attributes.http.server.address, alb_logs.traceId = traces.traceId`

确定关联时间范围的范围

要简化执行引擎（驱动程序）完成的工作，可以添加 scope 语句。这会明确指示联接查询确定搜索范围的时间。

`scope(@timestamp, 1D)`我

在此示例中，搜索范围以每日为重点，因此将同一天出现的相关性分组在一起。这种范围界定机制简化了搜索结果并允许更好地控制结果，从而可以根据您的需求进行渐进式搜索解决方案。

支持司机

新的关联命令实际上是一个 “隐藏” 的联接命令。因此，只有以下 PPL 驱动程序支持此命令。在这些驱动程序中，关联命令将直接转换为相应的 Catalyst Join 逻辑计划。

示例

`source alb_logs, traces, metrics | where ip="10.0.0.1" AND cloud.provider="aws"| correlate exact on (ip, port) scope(@timestamp, 2018-07-02T22:23:00, 1 D)`

逻辑计划：

```
'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]
```

Catalyst 引擎根据最高效的联接顺序优化此查询。

重复数据删除命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用dedup命令根据指定字段从搜索结果中移除相同的文档。

语法

使用以下语法：

```
dedup [int] <field-list> [keepempty=<bool>] [consecutive=<bool>]
```

int

- 可选。
- 当您指定时，该dedup命令会为每个组合保留多个事件<int>。的数字<int>必须大于 0。如果不指定数字，则只保留第一个发生的事件。所有其他重复项都将从结果中删除。
- 默认值：1

keepempty

- 可选。
- 如果为 true，则保留字段列表中任何字段都为 NULL 值或缺失的文档。
- 默认：false

consecutive

- 可选。
- 如果为 true，则仅删除具有连续重复值组合的事件。
- 默认：false

field-list

- 必需。

- 以逗号分隔的字段列表。至少有一个字段是必填字段。

示例 1：按一个字段进行重复数据删除

此示例说明如何使用性别字段对文档进行重复数据删除。

PPL 查询：

```
os> source=accounts | dedup gender | fields account_number, gender;
fetched rows / total rows = 2/2
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 13               | F        |
+------------------+----------+
```

示例 2：保留 2 份重复的文档

该示例显示了如何删除带有性别字段的重复文档，同时保留两个重复项。

PPL 查询：

```
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        |
+------------------+----------+
```

示例 3：默认情况下保留或忽略空字段

该示例说明如何通过保留空值字段对文档进行重复数据删除。

PPL 查询：

```
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   |
+------------------+-----------------------+
```

该示例说明如何通过忽略空值字段来对文档进行重复数据删除。

PPL 查询：

```
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   |
+------------------+-----------------------+
```

示例 4：连续文档中的重复数据删除

该示例显示了如何在连续的文档中进行重复数据删除。

PPL 查询：

```
os> source=accounts | dedup gender consecutive=true | fields account_number, gender;
fetched rows / total rows = 3/3
+------------------+----------+
| account_number   | gender   |
+------------------+----------+
| 1                | M        |
| 13               | F        |
| 18               | M        |
+------------------+----------+
```

其他示例

- `source = table | dedup a | fields a,b,c`

- source = table | dedup a,b | fields a,b,c
- source = table | dedup a keepempty=true | fields a,b,c
- source = table | dedup a,b keepempty=true | fields a,b,c
- source = table | dedup 1 a | fields a,b,c
- source = table | dedup 1 a,b | fields a,b,c
- source = table | dedup 1 a keepempty=true | fields a,b,c
- source = table | dedup 1 a,b keepempty=true | fields a,b,c
- source = table | dedup 2 a | fields a,b,c
- source = table | dedup 2 a,b | fields a,b,c
- source = table | dedup 2 a keepempty=true | fields a,b,c
- source = table | dedup 2 a,b keepempty=true | fields a,b,c
- source = table | dedup 1 a consecutive=true| fields a,b,c（不支持连续重复数据删除）

限制

- 对于 | dedup 2 a, b keepempty=false

```
DataFrameDropColumns('_row_number_)
+- Filter ('_row_number_ <= 2) // allowed duplication = 2
   +- Window [row_number() windowspecdefinition('a, 'b, 'a ASC NULLS FIRST, 'b ASC
 NULLS FIRST, specifiedwindowframe(RowFrame, unboundedpreceding$(), currentrow$()))
 AS _row_number_], ['a, 'b], ['a ASC NULLS FIRST, 'b ASC NULLS FIRST]
       +- Filter (isnotnull('a) AND isnotnull('b)) // keepempty=false
          +- Project
             +- UnresolvedRelation
```

- 对于 | dedup 2 a, b keepempty=true

```
Union
:- DataFrameDropColumns('_row_number_)
:  +- Filter ('_row_number_ <= 2)
:     +- Window [row_number() windowspecdefinition('a, 'b, 'a ASC NULLS FIRST, 'b ASC
 NULLS FIRST, specifiedwindowframe(RowFrame, unboundedpreceding$(), currentrow$()))
 AS _row_number_], ['a, 'b], ['a ASC NULLS FIRST, 'b ASC NULLS FIRST]
:        +- Filter (isnotnull('a) AND isnotnull('b))
:           +- Project
```

```
:              +- UnresolvedRelation
+- Filter (isnull('a) OR isnull('b))
   +- Project
      +- UnresolvedRelation
```

描述命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用`describe`命令获取有关表、架构和目录的结构和元数据的详细信息。以下是该`describe`命令的各种示例和用例。

描述

- `describe table`此命令等同于 S `DESCRIBE EXTENDED table` QL 命令
- `describe schema.table`
- `describe schema.`table``
- `describe catalog.schema.table`
- `describe catalog.schema.`table``
- `describe `catalog`.`schema`.`table``

eval 命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

该`eval`命令计算表达式并将结果附加到搜索结果中。

语法

使用以下语法：

```
eval <field>=<expression> [","  <field>=<expression> ]...
```

- `field`: 必填项。如果字段名称不存在，则会添加一个新字段。如果该字段名称已经存在，则它将被覆盖。
- `expression`: 必填项。系统支持的任何表达式。

示例 1：创建新字段

此示例说明如何为每个文档创建一个新`doubleAge`字段。新的`doubleAge`是年龄乘以2的评估结果。

PPL 查询：

```
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          |
+-------+-------------+
```

示例 2：覆盖现有字段

此示例说明如何使用年龄加 1 覆盖现有年龄字段。

PPL 查询：

```
os> source=accounts | eval age = age + 1 | fields age ;
fetched rows / total rows = 4/4
+-------+
| age   |
|-------|
| 33    |
| 37    |
| 29    |
| 34    |
+-------+
```

示例 3：使用在 eval 中定义的字段创建新字段

此示例说明如何使用在 eval 命令中定义的ddAge字段创建新字段。新字段ddAge是doubleAge乘以 2 的计算结果，其中doubleAge在 eval 命令中定义。

PPL 查询：

```
os> source=accounts | eval doubleAge = age * 2, ddAge = doubleAge * 2 | fields age,
 doubleAge, ddAge ;
fetched rows / total rows = 4/4
+-------+-------------+---------+
| age   | doubleAge   | ddAge   |
|-------+-------------+---------|
| 32    | 64          | 128     |
| 36    | 72          | 144     |
| 28    | 56          | 112     |
| 33    | 66          | 132     |
+-------+-------------+---------+
```

假设：a、b、中c是否存在字段 table

其他示例

- source = table | eval f = 1 | fields a,b,c,f
- source = table | eval f = 1（输出 a、b、c、f 字段）
- 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

用案例示例进行评估：

```
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')
```

用另一个案例示例进行评估：

假设：a、b、中c是否存在字段 table

其他示例

- source = table | eval f = 1 | fields a,b,c,f
- source = table | eval f = 1（输出 a、b、c、f 字段）
- 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`

用案例示例进行评估：

```
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')
```

用另一个案例示例进行评估：

```
source = table |  where ispresent(a) |
eval status_category =
 case(a >= 200 AND a < 300, 'Success',
  a >= 300 AND a < 400, 'Redirection',
  a >= 400 AND a < 500, 'Client Error',
  a >= 500, 'Server Error'
  else 'Incorrect HTTP status code'
 )
 | stats count() by status_category
```

限制

- 不支持覆盖现有字段。尝试这样做的查询会抛出异常，并显示消息 “引用'a'不明确”。

```
- `source = table | eval a = 10 | fields a,b,c`
- `source = table | eval a = a * 2 | stats avg(a)`
- `source = table | eval a = abs(a) | where a > 0`
```

```
- `source = table | eval a = signum(a) | where a < 0`
```

eventstats 命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用该`eventstats`命令使用计算出的汇总统计数据来丰富您的事件数据。它的工作原理是分析事件中的指定字段，计算各种统计度量，然后将这些结果作为新字段附加到每个原始事件。

赛事统计的关键方面

1. 它对整个结果集或在定义的组内执行计算。
2. 原始事件保持不变，并添加了包含统计结果的新字段。
3. 该命令对于比较分析、识别异常值或为单个事件提供更多背景信息特别有用。

统计数据和赛事统计数据之间的区别

`stats`和`eventstats`命令都用于计算统计数据，但它们在操作方式和生成内容方面有一些关键区别。

输出格式

- `stats`：生成仅包含计算统计数据的汇总表。
- `eventstats`：将计算的统计数据作为新字段添加到现有事件中，同时保留原始数据。

活动保留

- `stats`：将结果集简化为仅统计摘要，丢弃单个事件。
- `eventstats`：保留所有原始事件，并使用计算出的统计数据添加新字段。

使用案例

- `stats`：最适合创建摘要报告或仪表板。通常用作汇总结果的最终命令。
- `eventstats`：当您需要使用统计背景丰富事件以进行进一步分析或筛选时，这很有用。可以在搜索中使用来添加可在后续命令中使用的统计信息。

语法

使用以下语法：

```
eventstats <aggregation>... [by-clause]
```

聚合

- 必需。
- 聚合函数。
- 聚合的参数必须是字段。

附带条款

- 可选。
- 语法：`by [span-expression,] [field,]...`
- by 子句可以包括字段和表达式，例如标量函数和聚合函数。您也可以使用 span 子句将特定字段拆分为间隔相等的存储桶。然后，eventstats 命令根据这些跨度存储桶执行聚合。
- 默认：如果未指定 by 子句，则 eventstats 命令会聚合整个结果集。

跨度表达式

- 可选，最多一个。
- 语法：`span(field_expr, interval_expr)`
- 默认情况下，间隔表达式的单位是自然单位。但是，对于日期和时间类型字段，使用日期/时间单位时，需要在间隔表达式中指定单位。

  例如，要按照 10 年`age`将字段拆分为存储桶，请使用`span(age, 10)`。对于基于时间的字段，您可以使用将`timestamp`字段拆分为每小时间隔。`span(timestamp, 1h)`

可用时间单位

跨度间隔单位
毫秒 (ms)
秒

跨度间隔单位
分钟（m，区分大小写）
小时 (h)
日 (d)
周 (w)
月（M，区分大小写）
季度 (q)
年 (y)

聚合函数

COUNT

COUNT返回 SELECT 语句检索到的行中 expr 数量的计数。

对于 CloudWatch 日志，COUNT不支持使用查询。

示例：

```
os> source=accounts | eventstats count();
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------+--------------------
+------------+-------------------------+--------+-------+---------+
| account_number | balance  | firstname | lastname | age | gender | address
 | employer   | email                    | city   | state | count() |
+----------------+----------+-----------+----------+-----+--------+--------------------
+------------+-------------------------+--------+-------+---------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
 | AnyCorp    | janedoe@anycorp.com      | Brogan | IL    | 4       |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
 | AnyCompany | marymajor@anycompany.com | Dante  | TN    | 4       |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
 | AnyOrg     |                          | Nogal  | VA    | 4       |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
 |            | juanli@exampleorg.com    | Orick  | MD    | 4       |
```

```
+----------------+----------+-----------+----------+-----+--------+--------------------
+------------+-------------------------+--------+-------+---------+
```

SUM

SUM(expr)返回 expr 的总和。

示例：

```
os> source=accounts | eventstats sum(age) by gender;
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer   | email                     | city   | state | sum(age) by gender |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp    | janedoe@anycorp.com      | Brogan | IL    | 101                |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | AnyCompany | marymajor@anycompany.com | Dante  | TN    | 101                |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg     |                          | Nogal  | VA    | 28                 |
| 18             | 4180     | Juan      | Li       | 33  | M      | 467 Example Court
   |            | juanli@exampleorg.com    | Orick  | MD    | 101                |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
```

AVG

AVG(expr)返回 expr 的平均值。

示例：

```
os> source=accounts | eventstats avg(age) by gender;
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+--------------------------+--------+-------
+--------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                     | city   | state | avg(age) by gender |
```

```
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+---------------------------+--------+-------
+--------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 33.67             |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 33.67             |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 28.00             |
| 18             | 4180     | Juan      | Li       | 33  | M      | 467 Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 33.67             |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+---------------------------+--------+-------
+--------------------+
```

MAX

`MAX(expr)`返回 expr 的最大值。

示例

```
os> source=accounts | eventstats max(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+---------------------------+--------+-------
+-----------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                    | city   | state | max(age)  |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+---------------------------+--------+-------
+-----------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 36        |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 36        |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 36        |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 36        |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+------------+---------------------------+--------+-------
+-----------+
```

MIN

`MIN(expr)`返回 expr 的最小值。

示例

```
os> source=accounts | eventstats min(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+-----------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                     | city   | state | min(age)  |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+-----------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 28        |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 28        |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | *** Any Street
   | AnyOrg      |                          | Nogal  | VA    | 28        |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 28        |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+-----------+
```

STDDEV_SAMP

`STDDEV_SAMP(expr)`返回 expr 的样本标准差。

示例

```
os> source=accounts | eventstats stddev_samp(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+------------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                     | city   | state | stddev_samp(age)       |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+------------------------+
```

```
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 3.304037933599835      |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 3.304037933599835      |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 3.304037933599835      |
| 18             | 4180     | Juan      | Li       | 33  | M      | 467 Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 3.304037933599835      |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+------------------------+
```

STDDEV_POP

`STDDEV_POP(expr)`返回 expr 的总体标准差。

示例

```
os> source=accounts | eventstats stddev_pop(age);
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+------------------------+
| account_number | balance  | firstname | lastname | age | gender | address
   | employer    | email                    | city   | state | stddev_pop(age)        |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+------------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | 880 Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 2.****************     |
| 6              | 5686     | Mary      | Major    | 36  | M      | *** Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 2.****************     |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | *** Any Street
   | AnyOrg      |                          | Nogal  | VA    | 2.****************     |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 2.****************     |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+------------------------+
```

百分位数或百分位_大约

`PERCENTILE(expr, percent)`或者按指定百分比`PERCENTILE_APPROX(expr, percent)`返回 expr 的近似百分位数值。

百分比

- 该数字必须是介于 0 和 100 之间的常数。

示例

```
os> source=accounts | eventstats percentile(age, 90) by gender;
fetched rows / total rows = 4/4
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+--------------------------------+
| account_number | balance  | firstname | lastname | age | gender | address
    | employer    | email                    | city   | state | percentile(age, 90) by
 gender  |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+--------------------------------+
| 1              | 39225    | Jane      | Doe      | 32  | M      | *** Any Lane
   | AnyCorp     | janedoe@anycorp.com      | Brogan | IL    | 36
       |
| 6              | 5686     | Mary      | Major    | 36  | M      | 671 Example Street
   | Any Company | marymajor@anycompany.com | Dante  | TN    | 36
       |
| 13             | 32838    | Nikki     | Wolf     | 28  | F      | 789 Any Street
   | AnyOrg      |                          | Nogal  | VA    | 28
       |
| 18             | 4180     | Juan      | Li       | 33  | M      | *** Example Court
   |             | juanli@exampleorg.com    | Orick  | MD    | 36
       |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+--------------------------------+
```

示例 1：按组计算字段的平均值、总和和和计数

该示例显示了按性别计算所有账户组的平均年龄、总年龄和事件计数。

```
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     |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+-----------+-----------+-------+
```

示例 2：按跨度计算计数

该示例以 10 年为间隔获取年龄计数。

```
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        |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+----------+
```

示例 3：按性别和跨度计算计数

该示例以 5 年为间隔获取年龄计数，并按性别进行分组。

```
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   |
+----------------+----------+-----------+----------+-----+--------
+-----------------------+-------------+--------------------------+--------+-------
+-----+
```

使用量

- 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)`

带跨度的聚合

- `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`

具有时间窗跨度的聚合（翻转窗口功能）

- `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`

按多个级别对聚合进行分组

- `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`

扩展命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用expand命令展平以下类型的字段：

- `Array<Any>`
- `Map<Any>`

语法

使用以下语法：

```
expand <field> [As alias]
```

字段

- 要扩展（分解）的字段。必须是支持的类型。

别名

- 可选。要使用的名称而不是原始字段名称。

使用量

该expand命令为指定数组或映射字段中的每个元素生成一行，其中：

- 数组元素变成单独的行。
- 地图键值对被分成不同的行，每个键值都表示为一行。

- 提供别名时，分解值将以别名而不是原始字段名表示。
- 它可以与其他命令（例如、和）结合使用 `statseval`，`parse`以便在扩展后操作或提取数据。

示例

- `source = table | expand employee | stats max(salary) as max by state, company`
- `source = table | expand employee as worker | stats max(salary) as max by state, company`
- `source = table | expand employee as worker | eval bonus = salary * 3 | fields worker, bonus`
- `source = table | expand employee | parse description '(?<email>.+@.+)' | fields employee, email`
- `source = table | eval array=json_array(1, 2, 3) | expand array as uid | fields name, occupation, uid`
- `source = table | expand multi_valueA as multiA | expand multi_valueB as multiB`

解释命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

该`explain`命令可帮助您了解查询执行计划，使您能够分析和优化查询以提高性能。本简介简要概述了 explain 命令的用途及其在查询优化中的重要性。

注释

- `source=accounts | top gender // finds most common gender of all the accounts`（行注释）
- `source=accounts | dedup 2 gender /* dedup the document with gender field keep 2 duplication */ | fields account_number, gender`（屏蔽评论）

描述

- `describe table`此命令等同于 S `DESCRIBE EXTENDED table` QL 命令
- `describe schema.table`

- `describe schema.`table``
- `describe catalog.schema.table`
- `describe catalog.schema.`table``
- `describe `catalog`.`schema`.`table``

解释

- `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`

字段

- `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`

字段摘要

- `source = t | fieldsummary includefields=status_code nulls=false`
- `source = t | fieldsummary includefields= id, status_code, request_path nulls=true`
- `source = t | where status_code != 200 | fieldsummary includefields= status_code nulls=true`

嵌套字段

- `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`

筛选条件

- `source = table | where a = 1 | fields a,b,c`
- `source = table | where a >= 1 | fields a,b,c`
- `source = table | where a < 1 | fields a,b,c`
- `source = table | where b != 'test' | fields a,b,c`
- `source = table | where c = 'test' | fields a,b,c | head 3`
- `source = table | where ispresent(b)`
- `source = table | where isnull(coalesce(a, b)) | fields a,b,c | head 3`
- `source = table | where isempty(a)`
- `source = table | where isblank(a)`
- `source = table | where case(length(a) > 6, 'True' else 'False') = 'True'`
- `source = table | where a not in (1, 2, 3) | fields a,b,c`
- `source = table | where a between 1 and 4`-注意：这将返回 >= 1 和 <= 4，即 [1, 4]
- `source = table | where b not between '2024-09-10' and '2025-09-10'`-注意：这将返回 b >= '********** '和 b <=' 2025-09-10 '
- `source = table | where cidrmatch(ip, '************/24')`
- `source = table | where cidrmatch(ipv6, '2003:db8::/32')`
- `source = table | trendline sma(2, temperature) as temp_trend`

与知识产权相关的查询

- `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`

复杂的过滤器

```
source = table | eval status_category =
case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
else 'Incorrect HTTP status code')
| where case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
else 'Incorrect HTTP status code'
) = 'Incorrect HTTP status code'
```

```
source = table
| eval factor = case(a > 15, a - 14, isnull(b), a - 7, a < 3, a + 1 else 1)
| where case(factor = 2, 'even', factor = 4, 'even', factor = 6, 'even', factor = 8,
 'even' else 'odd') = 'even'
| stats count() by factor
```

具有逻辑条件的过滤器

- `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`

Eval

假设：a、b、中c是否存在字段 table

- source = table | eval f = 1 | fields a,b,c,f
- source = table | eval f = 1（输出 a、b、c、f 字段）
- source = table | eval n = now() | eval t = unix_timestamp(a) | fields n,t
- source = table | eval f = a | where f > 1 | sort f | fields a,b,c | head 5
- source = table | eval f = a * 2 | eval h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = f * 2 | fields a,f,h
- source = table | eval f = a * 2, h = b | stats avg(f) by h
- source = table | eval f = ispresent(a)
- source = table | eval r = coalesce(a, b, c) | fields r
- source = table | eval e = isempty(a) | fields e
- source = table | eval e = isblank(a) | fields e
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one', a = 2, 'two', a = 3, 'three', a = 4, 'four', a = 5, 'five', a = 6, 'six', a = 7, 'se7en', a = 8, 'eight', a = 9, 'nine')
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else 'unknown')
- source = table | eval f = case(a = 0, 'zero', a = 1, 'one' else concat(a, ' is an incorrect binary digit'))
- source = table | eval digest = md5(fieldName) | fields digest
- source = table | eval digest = sha1(fieldName) | fields digest
- source = table | eval digest = sha2(fieldName,256) | fields digest
- source = table | eval digest = sha2(fieldName,512) | fields digest

fillnull 命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

描述

使用`fillnull`命令将搜索结果的一个或多个字段中的空值替换为指定值。

语法

使用以下语法：

```
fillnull [with <null-replacement> in <nullable-field>["," <nullable-field>]] | [using
 <source-field> = <null-replacement> ["," <source-field> = <null-replacement>]]
```

- 空值替换：必须。用于替换空值的值。
- 可空字段：必填字段。字段参考。此字段中的空值将替换为空值替换中指定的值。

示例 1：填空一个字段

该示例说明了如何在单个字段上使用 fillnull：

```
os> source=logs | fields status_code | eval input=status_code | fillnull with 0 in
 status_code;
| input | status_code |
|-------|-------------|
| 403   | 403         |
| 403   | 403         |
| NULL  | 0           |
| NULL  | 0           |
| 200   | 200         |
| 404   | 404         |
| 500   | 500         |
| NULL  | 0           |
| 500   | 500         |
| 404   | 404         |
| 200   | 200         |
| 500   | 500         |
| NULL  | 0           |
| NULL  | 0           |
| 404   | 404         |
```

示例 2：将 Fillnull 应用于多个字段

该示例显示了应用于多个字段的 fillnull。

```
os> source=logs | fields request_path, timestamp | eval
 input_request_path=request_path, input_timestamp = timestamp | fillnull with '???' in
 request_path, timestamp;
| input_request_path | input_timestamp       | request_path | timestamp              |
|--------------------------------------------------------------------------------------|
| /contact           | NULL                  | /contact     | ???                    |
| /home              | NULL                  | /home        | ???                    |
| /about             | 2023-10-01 10:30:00   | /about       | 2023-10-01 10:30:00    |
| /home              | 2023-10-01 10:15:00   | /home        | 2023-10-01 10:15:00    |
| NULL               | 2023-10-01 10:20:00   | ???          | 2023-10-01 10:20:00    |
| NULL               | 2023-10-01 11:05:00   | ???          | 2023-10-01 11:05:00    |
| /about             | NULL                  | /about       | ???                    |
| /home              | 2023-10-01 10:00:00   | /home        | 2023-10-01 10:00:00    |
| /contact           | NULL                  | /contact     | ???                    |
| NULL               | 2023-10-01 10:05:00   | ???          | 2023-10-01 10:05:00    |
| NULL               | 2023-10-01 10:50:00   | ???          | 2023-10-01 10:50:00    |
| /services          | NULL                  | /services    | ???                    |
| /home              | 2023-10-01 10:45:00   | /home        | 2023-10-01 10:45:00    |
| /services          | 2023-10-01 11:00:00   | /services    | 2023-10-01 11:00:00    |
| NULL               | 2023-10-01 10:35:00   | ???          | 2023-10-01 10:35:00    |
```

示例 3：Fillnull 应用于具有各种空替换值的多个字段。

该示例显示了 fillnull，其中包含用于替换空值的各种值。

- `/error`在`request_path`田野里
- `1970-01-01 00:00:00`在`timestamp`田野里

```
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    |
```

字段命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用`fields`命令保留或删除搜索结果中的字段。

语法

使用以下语法：

```
field [+|-] <field-list>
```

- `index`: 可选。

  如果使用加号 (+)，则仅保留字段列表中指定的字段。

  如果使用减号 (-)，则字段列表中指定的所有字段都将被删除。

  默认：+
- `field list`: 必填项。要保留或删除的以逗号分隔的字段列表。

示例 1：从结果中选择指定字段

此示例说明如何从搜索结果中提取`account_number``firstname`、和`lastname`字段。

PPL 查询：

```
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         |
+------------------+-------------+------------+
```

示例 2：从结果中移除指定字段

此示例说明如何从搜索结果中移除该`account_number`字段。

PPL 查询：

```
os> source=accounts | fields account_number, firstname, lastname | fields -
 account_number ;
fetched rows / total rows = 4/4
+-------------+------------+
| firstname   | lastname   |
|-------------+------------|
| Jane        | Doe        |
| John        | Doe        |
| Jorge       | Souza      |
| Juan        | Li         |
+-------------+------------+
```

其他示例

- `source = table`
- `source = table | fields a,b,c`
- `source = table | fields + a,b,c`
- `source = table | fields - b,c`
- `source = table | eval b1 = b | fields - b1,c`

嵌套字段示例：

```
`source = catalog.schema.table1, catalog.schema.table2 | fields A.nested1, B.nested1`
`source = catalog.table | where struct_col2.field1.subfield > 'valueA' | sort int_col |
 fields  int_col, struct_col.field1.subfield, struct_col2.field1.subfield`
```

```
`source = catalog.schema.table | where struct_col2.field1.subfield > 'valueA' | sort
 int_col | fields  int_col, struct_col.field1.subfield, struct_col2.field1.subfield`
```

扁平化命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用 flatten 命令展开以下类型的字段：

- `struct<?,?>`
- `array<struct<?,?>>`

语法

使用以下语法：

```
flatten <field>
```

- 字段：要展平的字段。该字段必须是支持的类型。

架构

col_name	data_type
_time	字符串
桥梁	<length:bigint, name:string>数组 <struct>
city	字符串
coor	结构<alt:bigint, lat:double, long:double>
country	字符串

数据

_time	桥梁	city	coor	country
2024-09-13T 12:00:00	[{801，塔桥}，{928，伦敦桥}]	伦敦	{35、51.5074、-0.1278}	England
2024-09-13T 12:00:00	[{232，Pont Neuf}，{160，Pont Alexandre III}]	巴黎	{35、48.8566、2.3522}	法国
2024-09-13T 12:00:00	[{48，里亚托桥}，{11，叹息桥}]	威尼斯	{2、45.4408、12.3155}	意大利
2024-09-13T 12:00:00	[{***，查理大桥}，{343，军团桥}]	布拉格	{200、50.0755、14.4378}	捷克共和国
2024-09-13T 12:00:00	[{375，Chain Bridge}，{333，Liberty Bridge}]	布达佩斯	{96、47.4979、19.0402}	匈牙利
1990-09-13T 12:00:00	NULL	华沙	NULL	波兰

示例 1：扁平化结构

此示例说明如何扁平化结构字段。

PPL 查询：

```
source=table | flatten coor
```

_time	桥梁	city	country	alt	lat	长整数
2024-09-13T 12:00:00	[{801，塔桥}，{928，伦敦桥}]	伦敦	England	35	51.5074	-0.1278
2024-09-13T 12:00:00	[{232，Pc t Neuf}，{16 0，Pont Alexandre III}]	巴黎	法国	35	48.8566	2.3522
2024-09-13T 12:00:00	[{48，里亚托桥}，{11，叹息桥}]	威尼斯	意大利	2	45.4408	12.3155
2024-09-13T 12:00:00	[{516，查理大桥}，{343，军团桥}]	布拉格	捷克共和国	200	50.0755	14.4378
2024-09-13T 12:00:00	[{375，Ch in Bridge}，{ 333，Libe ty Bridge}]	布达佩斯	匈牙利	96	47.4979	19.0402

_time	桥梁	city	country	alt	lat	长整数
1990-09-13T 12:00:00	NULL	华沙	波兰	NULL	NULL	NULL

示例 2：扁平化数组

该示例显示了如何扁平化结构字段数组。

PPL 查询：

```
source=table | flatten bridges
```

_time	city	coor	country	length	名称
2024-09-13T 12:00:00	伦敦	{35、51.5074、-0.1278}	England	801	塔桥
2024-09-13T 12:00:00	伦敦	{35、51.5074、-0.1278}	England	928	伦敦桥
2024-09-13T 12:00:00	巴黎	{35、48.8566、2.3522}	法国	232	Pont Neuf
2024-09-13T 12:00:00	巴黎	{35、48.8566、2.3522}	法国	160	亚历山大三世桥
2024-09-13T 12:00:00	威尼斯	{2、45.4408、12.3155}	意大利	48	里亚托桥
2024-09-13T 12:00:00	威尼斯	{2、45.4408、12.3155}	意大利	11	叹息之桥
2024-09-13T 12:00:00	布拉格	{200、50.0755、14.4378}	捷克共和国	516	查理大桥

_time	city	coor	country	length	名称
2024-09-13T 12:00:00	布拉格	{200、50.0755、14.4378}	捷克共和国	343	军团桥
2024-09-13T 12:00:00	布达佩斯	{96、47.4979、19.0402}	匈牙利	375	链桥
2024-09-13T 12:00:00	布达佩斯	{96、47.4979、19.0402}	匈牙利	333	自由桥
1990-09-13T 12:00:00	华沙	NULL	波兰	NULL	NULL

示例 3：扁平化数组和结构

此示例说明如何扁平化多个字段。

PPL 查询：

```
source=table | flatten bridges | flatten coor
```

_time	city	country	length	名称	alt	lat	长整数
2024-09-13T 12:00:00	伦敦	England	801	塔桥	35	51.5074	-0.1278
2024-09-13T 12:00:00	伦敦	England	928	伦敦桥	35	51.5074	-0.1278
2024-09-13T 12:00:00	巴黎	法国	232	Pont Neuf	35	48.8566	2.3522

_time	city	country	length	名称	alt	lat	长整数
2024-09-13T 12:00:00	巴黎	法国	160	亚历山大三世桥	35	48.8566	2.3522
2024-09-13T 12:00:00	威尼斯	意大利	48	里亚托桥	2	45.4408	12.3155
2024-09-13T 12:00:00	威尼斯	意大利	11	叹息之桥	2	45.4408	12.3155
2024-09-13T 12:00:00	布拉格	捷克共和国	516	查理大桥	200	50.0755	14.4378
2024-09-13T 12:00:00	布拉格	捷克共和国	343	军团桥	200	50.0755	14.4378
2024-09-13T 12:00:00	布达佩斯	匈牙利	375	链桥	96	47.4979	19.0402
2024-09-13T 12:00:00	布达佩斯	匈牙利	333	自由桥	96	47.4979	19.0402
1990-09-13T 12:00:00	华沙	波兰	NULL	NULL	NULL	NULL	NULL

grok 命令

> **Note**
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

该grok命令使用 grok 模式解析文本字段，并将结果附加到搜索结果中。

语法

使用以下语法：

```
grok <field> <pattern>
```

字段

- 必需。
- 该字段必须是文本字段。

模式

- 必需。
- 用于从给定文本字段中提取新字段的 grok 模式。
- 如果新的字段名称已经存在，它将替换原始字段。

Grok 模式

grok 模式用于匹配每个文档的文本字段以提取新字段。

示例 1：创建新字段

此示例说明如何host为每个文档创建一个新字段。 host将是该email字段后@面的主机名。解析空字段将返回一个空字符串。

```
os> source=accounts | grok email '.+@%{HOSTNAME:host}' | fields email, host ;
fetched rows / total rows = 4/4
+-------------------------+-------------+
```

```
| email                    | host        |
|--------------------------+-------------|
| jane_doe@example.com     | example.com |
| arnav_desai@example.net  | example.net |
| null                     |             |
| juan_li@example.org      | example.org |
+--------------------------+-------------+
```

示例 2：覆盖现有字段

此示例说明如何在移除门牌号的情况下覆盖现有address字段。

```
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    |
+------------------+
```

示例 3：使用 grok 解析日志

此示例说明如何使用 grok 解析原始日志。

```
os> source=apache | grok message '%{COMMONAPACHELOG}' | fields COMMONAPACHELOG,
 timestamp, response, bytes ;
fetched rows / total rows = 4/4
+-----------------------------------------------------------------------------------------------
+----------------------------+------------+---------+
| COMMONAPACHELOG
                                          | timestamp                  | response   |
 bytes   |
|-----------------------------------------------------------------------------------------------
+----------------------------+------------+---------|
| 177.95.8.74 - upton5450 [28/Sep/2022:10:15:57 -0700] "HEAD /e-business/mindshare
 HTTP/1.0" 404 19927                        | 28/Sep/2022:10:15:57 -0700 | 404        |
 19927   |
| 127.45.152.6 - pouros8756 [28/Sep/2022:10:15:57 -0700] "GET /architectures/
convergence/niches/mindshare HTTP/1.0" 100 28722 | 28/Sep/2022:10:15:57 -0700 | 100
    | 28722   |
```

```
| *************** - - [28/Sep/2022:10:15:57 -0700] "PATCH /strategize/out-of-the-box
 HTTP/1.0" 401 27439                      | 28/Sep/2022:10:15:57 -0700 | 401        |
 27439   |
| ************** - - [28/Sep/2022:10:15:57 -0700] "POST /users HTTP/1.1" 301 9481
                                        | 28/Sep/2022:10:15:57 -0700 | 301        | 9481
    |
+-----------------------------------------------------------------------------------------------------
+---------------------------+------------+---------+
```

限制

grok 命令与解析命令具有相同的限制。

head 命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用head命令返回搜索顺序中可选偏移量之后的前 N 个指定结果。

语法

使用以下语法：

```
head [<size>] [from <offset>]
```

<size>

- 可选整数。
- 要返回的结果数量。
- 默认值：10

<offset>

- 可选之后的整数from。
- 要跳过的结果数。
- 默认：0

示例 1：获取前 10 个结果

此示例说明如何从账户索引中检索最多 10 个结果。

PPL 查询：

```
os> source=accounts | fields firstname, age | head;
fetched rows / total rows = 4/4
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| Jane        | 32    |
| John        | 36    |
| Jorge       | 28    |
| Juan        | 33    |
+-------------+-------+
```

示例 2：获取前 N 个结果

该示例显示了账户索引的前 N 个结果。

PPL 查询：

```
os> source=accounts | fields firstname, age | head 3;
fetched rows / total rows = 3/3
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| Jane        | 32    |
| John        | 36    |
| Jorge       | 28    |
+-------------+-------+
```

示例 3：在偏移量 M 之后获取前 N 个结果

此示例说明如何在从账户索引中跳过 M 个结果后检索前 N 个结果。

PPL 查询：

```
os> source=accounts | fields firstname, age | head 3 from 1;
fetched rows / total rows = 3/3
```

```
+-------------+-------+
| firstname   | age   |
|-------------+-------|
| John        | 36    |
| Jorge       | 28    |
| Juan        | 33    |
+-------------+-------+
```

加入命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

join 命令允许您根据公共字段合并来自多个来源的数据，从而使您能够执行复杂的分析并从分布式数据集中获得更深入的见解

架构

至少有两个指数，`otel-v1-apm-span-*`（大）和`otel-v1-apm-service-map`（小）。

来自索引的相关字段：

`otel-v1-apm-span-*`

- traceID-跟踪的唯一标识符。来自同一条轨迹的所有跨度共享相同的 traceID。
- SpanID-追踪中跨度的唯一标识符，在创建跨度时分配。
- parentSpanId -此跨度的父跨度的 spanID。如果这是根跨度，则此字段必须为空。
- durationInNanos -开始时间和结束时间之间的差异（以纳秒为单位）。（这是`latency`在用户界面中）
- serviceName-跨度来源的资源。
- TraceGroup-跟踪的根跨度的名称。

`otel-v1-apm-service-map`

- serviceName-发出跨度的服务的名称。
- destination.domain-此客户端正在调用的服务的服务名称。

- destination.resource-此客户端正在调用的跨度名称（API、操作等）。
- target.domain-客户端正在调用的服务的服务名称。
- target.resource-客户端正在调用的跨度名称（API、操作等）。
- traceGroupName -启动请求链的顶级 span 名称。

要求

Supp join ort 可以计算以下内容：

对于每项服务，在服务地图索引上连接跨度索引，以计算不同类型过滤器下的指标。

此示例查询计算按`order`服务的跟踪组`client_cancel_order`筛选时的延迟。

```
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`)
```

迁移到 PPL

join 命令的语法

```
SEARCH source=<left-table>
| <other piped command>
| [joinType] JOIN
    [leftAlias]
    ON joinCriteria
    <right-table>
| <other piped command>
```

重写

```
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
```

加入类型

- 语法：`INNER | LEFT OUTER | CROSS`
- 可选
- 要执行的联接类型。`INNER`如果未指定，则默认值为。

leftAlias

- 语法：`left = <leftAlias>`
- 可选
- 用于左联接侧的子查询别名，以避免命名模棱两可。

加入标准

- 语法：`<expression>`
- 必需
- 语法以开头`ON`。它可以是任何比较表达式。通常，加入条件如下所示`<leftAlias>.<leftField>=<rightAlias>.<rightField>`。

例如：`l.id = r.id`。如果联接条件包含多个条件，则可以在每个比较表达式AND之间指定 and OR 运算符。例如 `l.id = r.id AND l.email = r.email AND (r.age > 65 OR r.age < 18)`。

更多示例

从 SQL 查询迁移（TPC-H 第 13 季度）：

```
SELECT c_count, COUNT(*) AS custdist
FROM
  ( SELECT c_custkey, COUNT(o_orderkey) c_count
    FROM customer LEFT OUTER JOIN orders ON c_custkey = o_custkey
        AND o_comment NOT LIKE '%unusual%packages%'
    GROUP BY c_custkey
  ) AS c_orders
GROUP BY c_count
ORDER BY custdist DESC, c_count DESC;
```

由 PPL 联接查询重写：

```
SEARCH source=customer
| FIELDS c_custkey
| LEFT OUTER JOIN
    ON c_custkey = o_custkey AND o_comment NOT LIKE '%unusual%packages%'
    orders
| STATS count(o_orderkey) AS c_count BY c_custkey
| STATS count() AS custdist BY c_count
| SORT - custdist, - c_count
```

限制：右侧加入不支持子搜索。

如果支持子搜索，则可以按如下方式重写上面的 PPL 查询：

```
SEARCH source=customer
| FIELDS c_custkey
| LEFT OUTER JOIN
   ON c_custkey = o_custkey
   [
      SEARCH source=orders
      | WHERE o_comment NOT LIKE '%unusual%packages%'
      | FIELDS o_orderkey, o_custkey
```

```
  ]
| STATS count(o_orderkey) AS c_count BY c_custkey
| STATS count() AS custdist BY c_count
| SORT - custdist, - c_count
```

查找命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用该`lookup`命令通过添加或替换查找索引（维度表）中的数据来丰富您的搜索数据。此命令允许您使用维度表中的值扩展索引的字段。您还可以在满足查找条件时使用它来追加或替换值。该`lookup`命令比使用静态数据集丰富源数据的`Join`命令更合适。

语法

使用以下语法：

```
SEARCH source=<sourceIndex>
| <other piped command>
| LOOKUP <lookupIndex> (<lookupMappingField> [AS <sourceMappingField>])...
    [(REPLACE | APPEND) (<inputField> [AS <outputField>])...]
| <other piped command>
```

查找索引

- 必需。
- 查询索引（维度表）的名称。

lookupMappingField

- 必需。
- 查找索引中的映射键，类似于右表中的联接键。您可以指定多个字段，用逗号分隔。

sourceMappingField

- 可选。

- 默认：< lookupMappingField >。
- 源查询中的映射密钥，类似于左侧的联接键。

输入字段

- 可选。
- 默认：查找索引中找到匹配值的所有字段。
- 查找索引中的一个字段，将匹配的值应用于结果输出。您可以指定多个字段，用逗号分隔。

输出字段

- 可选。
- 默认值：`<inputField>`。
- 输出中的一个字段。您可以指定多个输出字段。如果您在源查询中指定现有字段名称，则其值将替换或附加为 InputField 中的匹配值。如果您指定了新的字段名称，则该名称将被添加到结果中。

替换 | 追加

- 可选。
- 默认：替换
- 指定如何处理匹配的值。如果指定 REPLACE，则<lookupIndex>字段中匹配的值会覆盖结果中的值。如果指定APPEND，则<lookupIndex>字段中的匹配值只会追加到结果中的缺失值上。

使用量

- 查找 <lookupIndex>ID 为 cid 将邮件替换为电子邮件
- 查询<lookupIndex>名称将邮件替换为电子邮件
- 查找 <lookupIndex>ID 为 cid，姓名追加地址，邮件为电子邮件
- 查找 <lookupIndex>ID

示例

请见以下 示例。

```
SEARCH source=<sourceIndex>
```

```
| WHERE orderType = 'Cancelled'
| LOOKUP account_list, mkt_id AS mkt_code REPLACE amount, account_name AS name
| STATS count(mkt_code), avg(amount) BY name
```

```
SEARCH source=<sourceIndex>
| DEDUP market_id
| EVAL category=replace(category, "-", ".")
| EVAL category=ltrim(category, "dvp.")
| LOOKUP bounce_category category AS category APPEND classification
```

```
SEARCH source=<sourceIndex>
| LOOKUP bounce_category category
```

解析命令

该parse命令使用正则表达式解析文本字段，并将结果附加到搜索结果中。

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

语法

使用以下语法：

```
parse <field> <pattern>
```

field

- 必需。
- 该字段必须是文本字段。

pattern

- 必填字符串。
- 这是用于从给定文本字段中提取新字段的正则表达式模式。
- 如果新的字段名称已经存在，它将替换原始字段。

正则表达式

正则表达式模式用于将每个文档的整个文本字段与 Java 正则表达式引擎进行匹配。表达式中每个命名的捕获组都将成为一个新STRING字段。

示例 1：创建新字段

该示例说明如何host为每个文档创建一个新字段。 host将是该email字段后@面的主机名。解析空字段将返回一个空字符串。

PPL 查询：

```
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 |
+-----------------------+-------------+
```

示例 2：覆盖现有字段

该示例说明如何在移除门牌号的情况下覆盖现有address字段。

PPL 查询：

```
os> source=accounts | parse address '\d+ (?<address>.+)' | fields address ;
fetched rows / total rows = 4/4
+------------------+
| address          |
|------------------|
| Example Lane     |
| Example Street   |
| Example Avenue   |
| Example Court    |
+------------------+
```

示例 3：按转换后的解析字段进行筛选和排序

该示例说明如何对address字段中大于 500 的街道号码进行排序。

PPL 查询：

```
os> source=accounts | parse address '(?<streetNumber>\d+) (?<street>.+)' | where
 cast(streetNumber as int) > 500 | sort num(streetNumber) | fields streetNumber,
 street ;
fetched rows / total rows = 3/3
+----------------+----------------+
| streetNumber   | street         |
|----------------+----------------|
| ***            | Example Street |
| ***            | Example Avenue |
| 880            | Example Lane   |
+----------------+----------------+
```

限制

parse 命令有一些限制：

- 无法再次解析由 parse 定义的字段。

  以下命令不起作用：

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | parse street '\w+ (?<road>\w+)'
  ```

- parse 定义的字段不能被其他命令覆盖。

  where不会匹配任何文档，因为street无法覆盖：

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | eval street='1' | where
   street='1' ;
  ```

- parse 使用的文本字段不能被覆盖。

  street由于已被覆盖，address因此无法成功解析：

  ```
  source=accounts | parse address '\d+ (?<street>.+)' | eval address='1' ;
  ```

- 在stats命令中使用 parse 定义的字段后，无法对其进行筛选或排序。

  where在以下命令中将不起作用：

```
source=accounts | parse email '.+@(?<host>.+)' | stats avg(age) by host | where
 host=pyrami.com ;
```

模式命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

该patterns命令从文本字段中提取日志模式并将结果附加到搜索结果中。按模式对日志进行分组可以更轻松地从大量日志数据中汇总统计数据，以进行分析和故障排除。

语法

使用以下语法：

```
patterns [new_field=<new-field-name>] [pattern=<pattern>] <field>
```

new-field-name

- 可选字符串。
- 这是用于提取模式的新字段的名称。
- 默认值为 `patterns_field`。
- 如果名称已经存在，它将替换原始字段。

模式

- 可选字符串。
- 这是应该从文本字段中过滤掉的字符的正则表达式模式。
- 如果不存在，则默认模式为字母数字字符 (`[a-zA-Z\d]`)。

字段

- 必需。
- 该字段必须是文本字段。

示例 1：创建新字段

该示例说明如何在每个文档中email使用提取标点符号。解析空字段将返回一个空字符串。

PPL 查询：

```
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   | @.               |
+-----------------------+------------------+
```

示例 2：提取日志模式

该示例说明如何使用默认模式从原始日志字段中提取标点符号。

PPL 查询：

```
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
                                        | ... - - [//::: -] " / /."       |
+-----------------------------------------------------------------------------------------------------------
+---------------------------------+
```

示例 3：使用自定义正则表达式模式提取日志模式

该示例说明如何使用用户定义的模式从原始日志字段中提取标点符号。

PPL 查询：

```
os> source=apache | patterns new_field='no_numbers' pattern='[0-9]' message | fields
 message, no_numbers ;
fetched rows / total rows = 4/4
+-----------------------
```

使用`rare`命令查找字段列表中所有字段中最不常见的值元组。

> **Note**
>
> 对于分组依据字段的每个不同值元组，最多返回 10 个结果。

语法

使用以下语法：

```
rare [N] <field-list> [by-clause] rare_approx [N] <field-list> [by-clause]
```

字段清单

- 必需。
- 以逗号分隔的字段名称列表。

附带条款

- 可选。
- 用于对结果进行分组的一个或多个字段。

N

- 要返回的结果数量。
- 默认值：10

稀有度_大约

- 使用 HyperLogLog++ 算法估计的基数计算稀有 (n) 个字段的近似计数。

示例 1：在字段中查找最不常见的值

该示例找出了所有账户中最不常见的性别。

PPL 查询：

```
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        |
+----------+
```

示例 2：找出按性别组织的最不常见的价值观

该示例按性别找出所有账户组中最不常见的年龄。

PPL 查询：

```
os> source=accounts | rare 5 age by gender;
os> source=accounts | rare_approx 5 age by gender;
fetched rows / total rows = 4/4
+----------+-------+
| gender   | age   |
|----------+-------|
| F        | 28    |
| M        | 32    |
| M        | 33    |
| M        | 36    |
+----------+-------+
```

重命名命令

使用`rename`命令更改搜索结果中一个或多个字段的名称。

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

语法

使用以下语法：

```
rename <source-field> AS <target-field>["," <source-field> AS <target-field>]...
```

源字段

- 必需。
- 这是您要重命名的字段的名称。

目标字段

- 必需。
- 这是您要重命名的名称。

示例 1：重命名一个字段

此示例说明如何重命名单个字段。

PPL 查询：

```
os> source=accounts | rename account_number as an | fields an;
fetched rows / total rows = 4/4
+------+
| an   |
|------|
| 1    |
| 6    |
| 13   |
| 18   |
+------+
```

示例 2：重命名多个字段

此示例说明如何重命名多个字段。

PPL 查询：

```
os> source=accounts | rename account_number as an, employer as emp | fields an, emp;
fetched rows / total rows = 4/4
+------+---------+
| an   | emp     |
```

```
|------+---------|
| 1    | Pyrami  |
| 6    | Netagy  |
| 13   | Quility |
| 18   | null    |
+------+---------+
```

限制

- 不支持覆盖现有字段：

```
source=accounts | grok address '%{NUMBER} %{GREEDYDATA:address}' | fields address
```

搜索命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用`search`命令从索引中检索文档。该`search`命令只能用作 PPL 查询中的第一个命令。

语法

使用以下语法：

```
search source=[<remote-cluster>:]<index> [boolean-expression]
```

search

- 可选。
- 搜索关键词，可以省略。

索引

- 必需。
- 搜索命令必须指定从哪个索引进行查询。

- `<cluster name>`:对于跨集群搜索，索引名称可以加上前缀。

布尔表达式

- 可选。
- 任何计算结果为布尔值的表达式。

示例 1：获取所有数据

该示例显示了从账户索引中获取所有文档。

PPL 查询：

```
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         |
+------------------+-------------+----------------------+-----------+----------
+--------+----------------+---------+-------+-----------------------+------------+
```

示例 2：使用条件获取数据

该示例显示了使用从账户索引中获取所有文档。

PPL 查询：

```
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        |
+------------------+-------------+--------------------+-----------+----------+--------
+-----------------+---------+-------+-----------------------+------------+
```

排序命令

使用`sort`命令按指定字段对搜索结果进行排序。

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

语法

使用以下语法：

```
sort <[+|-] sort-field>...
```

[+|-]

- 可选。
- 加号 [+] 代表升序， NULL/MISSING 值在前。
- 减号 [-] 代表降序， NULL/MISSING 值排在最后。
- 默认：升序排序， NULL/MISSING 值在前。

排序字段

- 必需。
- 用于排序的字段。

示例 1：按一个字段排序

该示例说明如何使用年龄字段按升序对文档进行排序。

PPL 查询：

```
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    |
+------------------+-------+
```

示例 2：按一个字段排序并返回所有结果

该示例说明如何使用年龄字段按升序对文档进行排序。

PPL 查询：

```
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    |
+------------------+-------+
```

示例 3：按一个字段降序排序

该示例说明如何按降序对带有年龄字段的文档进行排序。

PPL 查询：

```
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    |
+-------------------+-------+
```

示例 4：按多个字段排序

该示例说明如何对文档进行排序，性别字段按升序排序，年龄字段按降序排序。

PPL 查询：

```
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    |
+------------------+----------+-------+
```

示例 5：按包含空值的字段排序

该示例说明如何按默认选项（升序排序，先为空）对雇主字段进行排序。结果显示空值位于第一行。

PPL 查询：

```
os> source=accounts | sort employer | fields employer;
fetched rows / total rows = 4/4
+------------+
| employer   |
|------------|
| null       |
| AnyCompany |
| AnyCorp    |
| AnyOrgty   |
+------------+
```

stats 命令

使用`stats`命令根据搜索结果计算聚合。

> **Note**
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

空值/缺失值处理

空值/缺失值处理

函数	NULL	缺失
COUNT	不算在内	不算在内
SUM	忽略	忽略
AVG	忽略	忽略
MAX	忽略	忽略
MIN	忽略	忽略

语法

使用以下语法：

```
stats <aggregation>... [by-clause]
```

聚合

- 必需。
- 应用于字段的聚合函数。

附带条款

- 可选。
- 语法：`by [span-expression,] [field,]...`
- 指定用于对聚合结果进行分组的字段和表达式。by-clause 允许您使用字段和表达式对聚合结果进行分组。您可以使用标量函数、聚合函数甚至跨度表达式将特定字段拆分为间隔相等的存储桶。

- 默认：如果<by-clause>未指定，则 stats 命令返回一行，表示整个结果集的聚合。

跨度表达式

- 可选，最多一个。
- 语法：`span(field_expr, interval_expr)`
- 默认情况下，间隔表达式的单位是自然单位。如果该字段是日期和时间类型字段，并且间隔以 date/time 单位表示，则可以在间隔表达式中指定单位。
- 例如，按照 10 年将`age`字段拆分为存储桶。`span(age, 10)`要将时间戳字段拆分为每小时间隔，请使用`span(timestamp, 1h)`。

可用时间单位

跨度间隔单位
毫秒 (ms)
秒
分钟（m，区分大小写）
小时 (h)
日 (d)
周 (w)
月（M，区分大小写）
季度 (q)
年 (y)

聚合函数

COUNT

返回 SELECT 语句检索到的行中 expr 数量的计数。

示例：

```
os> source=accounts | stats count();
fetched rows / total rows = 1/1
+-----------+
| count()   |
|-----------|
| 4         |
+-----------+
```

SUM

用于`SUM(expr)`返回 expr 的总和。

示例

```
os> source=accounts | stats sum(age) by gender;
fetched rows / total rows = 2/2
+------------+----------+
| sum(age)   | gender   |
|------------+----------|
| 28         | F        |
| 101        | M        |
+------------+----------+
```

AVG

用于`AVG(expr)`返回 expr 的平均值。

示例

```
os> source=accounts | stats avg(age) by gender;
fetched rows / total rows = 2/2
+--------------------+----------+
| avg(age)           | gender   |
|--------------------+----------|
| 28.0               | F        |
| 33.666666666666664 | M        |
+--------------------+----------+
```

MAX

用于`MAX(expr)`返回 expr 的最大值。

示例

```
os> source=accounts | stats max(age);
fetched rows / total rows = 1/1
+------------+
| max(age)   |
|------------|
| 36         |
+------------+
```

MIN

用于`MIN(expr)`返回 expr 的最小值。

示例

```
os> source=accounts | stats min(age);
fetched rows / total rows = 1/1
+------------+
| min(age)   |
|------------|
| 28         |
+------------+
```

STDDEV_SAMP

用于`STDDEV_SAMP(expr)`返回 expr 的样本标准差。

示例：

```
os> source=accounts | stats stddev_samp(age);
fetched rows / total rows = 1/1
+--------------------+
| stddev_samp(age)   |
|--------------------|
| 3.304037933599835  |
+--------------------+
```

STDDEV_POP

用于`STDDEV_POP(expr)`返回 expr 的总体标准差。

示例：

```
os> source=accounts | stats stddev_pop(age);
fetched rows / total rows = 1/1
+--------------------+
| stddev_pop(age)    |
|--------------------|
| 2.**************** |
+--------------------+
```

拿

用于`TAKE(field [, size])`返回字段的原始值。它不能保证值的顺序。

字段

- 必需。
- 该字段必须是文本字段。

size

- 可选整数。
- 应返回值的数量。
- 默认值为 10。

示例

```
os> source=accounts | stats take(firstname);
fetched rows / total rows = 1/1
+-----------------------------+
| take(firstname)             |
|-----------------------------|
| [Jane, Mary, Nikki, Juan    |
+-----------------------------+
```

百分位数或百分位_大约

使用`PERCENTILE(expr, percent)`或`PERCENTILE_APPROX(expr, percent)`以指定百分比返回 expr 的近似百分位数值。

百分比

- 该数字必须是介于 0 和 100 之间的常数。

示例

```
os> source=accounts | stats percentile(age, 90) by gender;
fetched rows / total rows = 2/2
+-----------------------+----------+
| percentile(age, 90)   | gender   |
|-----------------------+----------|
| 28                    | F        |
| 36                    | M        |
+-----------------------+----------+
```

示例 1：计算事件数量

该示例说明如何计算账户中的事件数量。

```
os> source=accounts | stats count();
fetched rows / total rows = 1/1
+-----------+
| count()   |
|-----------|
| 4         |
+-----------+
```

示例 2：计算字段的平均值

该示例说明如何计算所有账户的平均年龄。

```
os> source=accounts | stats avg(age);
fetched rows / total rows = 1/1
+------------+
| avg(age)   |
|------------|
| 32.25      |
+------------+
```

示例 3：按组计算字段的平均值

该示例说明如何计算按性别分组的所有账户的平均年龄。

```
os> source=accounts | stats avg(age) by gender;
fetched rows / total rows = 2/2
+--------------------+----------+
| avg(age)           | gender   |
|--------------------+----------|
| 28.0               | F        |
| 33.666666666666664 | M        |
+--------------------+----------+
```

示例 4：按组计算字段的平均值、总和和和计数

该示例说明如何计算按性别分组的所有账户的平均年龄、总年龄和事件计数。

```
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        |
+--------------------+------------+-----------+----------+
```

示例 5：计算字段的最大值

该示例计算了所有账户的最大年龄。

```
os> source=accounts | stats max(age);
fetched rows / total rows = 1/1
+------------+
| max(age)   |
|------------|
| 36         |
+------------+
```

示例 6：按组计算字段的最大值和最小值

该示例计算了按性别分组的所有账户的最大和最小年龄值。

```
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        |
+------------+------------+----------+
```

示例 7：计算字段的不同计数

要获取字段中不同值的计数，可以使用DISTINCT_COUNT（或DC）函数代替COUNT。该示例计算了所有帐户的计数和不同性别计数字段。

```
os> source=accounts | stats count(gender), distinct_count(gender);
fetched rows / total rows = 1/1
+-----------------+-------------------------+
| count(gender)   | distinct_count(gender)  |
|-----------------+-------------------------|
| 4               | 2                       |
+-----------------+-------------------------+
```

示例 8：按跨度计算计数

该示例以 10 年为间隔获取年龄计数。

```
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         |
+--------------+------------+
```

示例 9：按性别和跨度计算计数

此示例统计按性别和年龄跨度为 5 年的记录进行分组。

```
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        |
+-------+------------+----------+
```

无论命令中指定的顺序如何，span 表达式始终作为第一个分组键出现。

```
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        |
+-------+------------+----------+
```

示例 10：计算计数并按性别和跨度获取电子邮件列表

该示例按10年间隔获取年龄计数，并按性别分组，此外，每行最多可获得5封电子邮件的列表。

```
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        |
+-------+-----------------------------------------------------+------------+----------+
```

示例 11：计算字段的百分位数

该示例说明如何计算所有账户的第 90 个百分位年龄。

```
os> source=accounts | stats percentile(age, 90);
fetched rows / total rows = 1/1
+-----------------------+
| percentile(age, 90)   |
|-----------------------|
```

```
| 36                     |
+------------------------+
```

示例 12：按组计算字段的百分位数

该示例说明如何按性别计算所有账户组的第 90 个百分位年龄。

```
os> source=accounts | stats percentile(age, 90) by gender;
fetched rows / total rows = 2/2
+-----------------------+----------+
| percentile(age, 90)   | gender   |
|-----------------------+----------|
| 28                    | F        |
| 36                    | M        |
+-----------------------+----------+
```

示例 13：按性别和跨度计算百分位数

该示例按照 10 年间隔获得第 90 个百分位的年龄，并按性别分组。

```
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)`
```

带跨度的聚合

```
- `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`
```

具有时间窗口跨度的聚合（翻转窗口功能）

```
- `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`
```

按多个级别对聚合进行分组

```
- `source = table | stats avg(age) as avg_state_age by country, state | stats
 avg(avg_state_age) as avg_country_age by country`
- `source = table | stats avg(age) as avg_city_age by country, state, city | eval
 new_avg_city_age = avg_city_age - 1 | stats avg(new_avg_city_age) as avg_state_age
 by country, state | where avg_state_age > 18 | stats avg(avg_state_age) as
 avg_adult_country_age by country`
```

子查询命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用该`subquery`命令在管道处理语言 (PPL) 语句中执行复杂的嵌套查询。

```
source=logs | where field in [ subquery source=events | where condition | fields
 field ]
```

在此示例中，主搜索 (`source=logs`) 按子查询 (`source=events`) 的结果进行筛选。

subquery 命令支持用于复杂数据分析的多级嵌套。

嵌套子查询示例

```
source=logs | where id in [ subquery source=users | where user in [ subquery
 source=actions | where action="login" | fields user] | fields uid ]
```

InSubquery 用法

- `source = outer | where a in [ source = inner | fields b ]`
- `source = outer | where (a) in [ source = inner | fields b ]`
- `source = outer | where (a,b,c) in [ source = inner | fields d,e,f ]`
- `source = outer | where a not in [ source = inner | fields b ]`
- `source = outer | where (a) not in [ source = inner | fields b ]`
- `source = outer | where (a,b,c) not in [ source = inner | fields d,e,f ]`
- `source = outer a in [ source = inner | fields b ]`（使用子查询进行搜索筛选）
- `source = outer a not in [ source = inner | fields b ]`（使用子查询进行搜索筛选）
- `source = outer | where a in [ source = inner1 | where b not in [ source = inner2 | fields c ] | fields b ]`（嵌套）
- `source = table1 | inner join left = l right = r on l.a = r.a AND r.a in [ source = inner | fields d ] | fields l.a, r.a, b, c`（作为联接过滤器）

使用子查询内 PPL 的 SQL 迁移示例

TPC-H 第 4 季度（带聚合功能的子查询内）

```
select
  o_orderpriority,
  count(*) as order_count
from
  orders
where
  o_orderdate >= date '1993-07-01'
  and o_orderdate < date '1993-07-01' + interval '3' month
  and o_orderkey in (
    select
      l_orderkey
    from
      lineitem
    where l_commitdate < l_receiptdate
  )
```

```
group by
  o_orderpriority
order by
  o_orderpriority
```

由 PPL InSubquery 查询重写：

```
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（嵌套在子查询内）

```
select
  s_name,
  s_address
from
  supplier,
  nation
where
  s_suppkey in (
    select
      ps_suppkey
    from
      partsupp
    where
      ps_partkey in (
        select
          p_partkey
        from
          part
        where
          p_name like 'forest%'
      )
  )
  and s_nationkey = n_nationkey
  and n_name = 'CANADA'
```

```
order by
  s_name
```

由 PPL InSubquery 查询重写：

```
source = supplier
| where s_suppkey IN [
    source = partsupp
    | where ps_partkey IN [
        source = part
        | where like(p_name, "forest%")
        | fields p_partkey
      ]
    | fields ps_suppkey
  ]
| inner join left=l right=r on s_nationkey = n_nationkey and n_name = 'CANADA'
  nation
| sort s_name
```

ExistsSubquery 用法

假设：a，b是表外部的字段，是表内部的字段c，d是表内侧的字段e，f是表inner2的字段。

- `source = outer | where exists [ source = inner | where a = c ]`
- `source = outer | where not exists [ source = inner | where a = c ]`
- `source = outer | where exists [ source = inner | where a = c and b = d ]`
- `source = outer | where not exists [ source = inner | where a = c and b = d ]`
- `source = outer exists [ source = inner | where a = c ]`（使用子查询进行搜索筛选）
- `source = outer not exists [ source = inner | where a = c ]`（使用子查询进行搜索筛选）
- `source = table as t1 exists [ source = table as t2 | where t1.a = t2.a ]`（表别名在 exists 子查询中很有用）
- `source = outer | where exists [ source = inner1 | where a = c and exists [ source = inner2 | where c = e ] ]`（嵌套）
- `source = outer | where exists [ source = inner1 | where a = c | where exists [ source = inner2 | where c = e ] ]`（嵌套）

- `source = outer | where exists [ source = inner | where c > 10 ]`（存在不相关性）
- `source = outer | where not exists [ source = inner | where c > 10 ]`（存在不相关性）
- `source = outer | where exists [ source = inner ] | eval l = "nonEmpty" | fields l`（存在特殊的不相关性）

ScalarSubquery 用法

假设：a，b是表外部的字段c，d是表内部的字段e，表的字段f是嵌套的

不相关的标量子查询

在 “选择” 中：

- `source = outer | eval m = [ source = inner | stats max(c) ] | fields m, a`
- `source = outer | eval m = [ source = inner | stats max(c) ] + b | fields m, a`

在哪里：

- `source = outer | where a > [ source = inner | stats min(c) ] | fields a`

在 “搜索” 筛选器中：

- `source = outer a > [ source = inner | stats min(c) ] | fields a`

相关标量子查询

在 “选择” 中：

- `source = outer | eval m = [ source = inner | where outer.b = inner.d | stats max(c) ] | fields m, a`
- `source = outer | eval m = [ source = inner | where b = d | stats max(c) ] | fields m, a`
- `source = outer | eval m = [ source = inner | where outer.b > inner.d | stats max(c) ] | fields m, a`

在哪里：

- source = outer | where a = [source = inner | where outer.b = inner.d | stats max(c)]
- source = outer | where a = [source = inner | where b = d | stats max(c)]
- source = outer | where [source = inner | where outer.b = inner.d OR inner.d = 1 | stats count()] > 0 | fields a

在 “搜索” 筛选器中：

- source = outer a = [source = inner | where b = d | stats max(c)]
- source = outer [source = inner | where outer.b = inner.d OR inner.d = 1 | stats count()] > 0 | fields a

嵌套标量子查询

- 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]

（关系）子查询

InSubquery，ExistsSubquery并且ScalarSubquery都是子查询表达式。但它RelationSubquery不是子查询表达式，而是一个在 Join 或 From 子句中常用的子查询计划。

- source = table1 | join left = l right = r [source = table2 | where d > 10 | head 5]（右侧连接中的子查询）
- source = [source = table1 | join left = l right = r [source = table2 | where d > 10 | head 5] | stats count(a) by b] as outer | head 1

其他背景

InSubqueryExistsSubquery、和ScalarSubquery是子查询表达式中常用在子where句和搜索过滤器中。

其中命令：

```
| where <boolean expression> | ...
```

搜索过滤器：

```
search source=* <boolean expression> | ...
```

子查询表达式可以在布尔表达式中使用：

```
| where orders.order_id in [ source=returns | where return_reason="damaged" | field
 order_id ]
```

这`orders.order_id in [ source=... ]`是`<boolean expression>`.

通常，我们将这种子查询子句命名为`InSubquery`表达式。它是一个`<boolean expression>`.

具有不同联接类型的子查询

使用 `ScalarSubquery` 的示例：

```
source=employees
| join source=sales on employees.employee_id = sales.employee_id
| where sales.sale_amount > [ source=targets | where target_met="true" | fields
 target_value ]
```

与 InSubquery ExistsSubquery、和 ScalarSubquery，a RelationSubquery 不是子查询表达式。相反，它是一个子查询计划。

```
SEARCH source=customer
| FIELDS c_custkey
| LEFT OUTER JOIN left = c, right = o ON c.c_custkey = o.o_custkey
   [
      SEARCH source=orders
      | WHERE o_comment NOT LIKE '%unusual%packages%'
      | FIELDS o_orderkey, o_custkey
   ]
| STATS ...
```

top 命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用top命令查找字段列表中所有字段中最常见的值元组。

语法

使用以下语法：

```
top [N] <field-list> [by-clause] top_approx [N] <field-list> [by-clause]
```

N

- 要返回的结果数量。
- 默认值：10

字段清单

- 必需。
- 以逗号分隔的字段名称列表。

附带条款

- 可选。
- 用于对结果进行分组的一个或多个字段。

顶部_大约

- 使用 HyperLogLog++ 估计的基数算法对 (n) 个顶部字段的近似计数。

示例 1：查找字段中最常见的值

该示例找出了所有账户中最常见的性别。

PPL 查询：

```
os> source=accounts | top gender;
os> source=accounts | top_approx gender;
fetched rows / total rows = 2/2
+----------+
| gender   |
|----------|
| M        |
| F        |
+----------+
```

示例 2：查找字段中最常见的值（限制为 1）

该示例找到了所有账户中最常见的单一性别。

PPL 查询：

```
os> source=accounts | top_approx 1 gender;
fetched rows / total rows = 1/1
+----------+
| gender   |
|----------|
| M        |
+----------+
```

示例 3：找出最常见的值，按性别分组

该示例找出所有账户最常见的年龄，按性别分组。

PPL 查询：

```
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    |
+----------+-------+
```

趋势线命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用`trendline`命令计算字段的移动平均值。

语法

使用以下语法

```
TRENDLINE [sort <[+|-] sort-field>] SMA(number-of-datapoints, field) [AS alias]
 [SMA(number-of-datapoints, field) [AS alias]]...
```

[+|-]

- 可选。
- 加号 [+] 代表升序， NULL/MISSING 值在前。
- 减号 [-] 代表降序， NULL/MISSING 值排在最后。
- 默认：升序排序， NULL/MISSING 值在前。

排序字段

- 使用排序时为必填项。
- 用于排序的字段。

number-of-datapoints

- 必需。
- 计算移动平均线的数据点的数量。
- 必须大于零。

字段

- 必需。

- 应计算移动平均线的字段的名称。

别名

- 可选。
- 包含移动平均线的结果列的名称。

仅支持简单移动平均线 (SMA) 类型。它的计算方式是这样的：

```
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
```

示例 1：计算温度时间序列的简单移动平均线

该示例使用两个数据点计算温度的简单移动平均值。

PPL 查询：

```
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|
+-----------+---------+--------------------+----------+
```

示例 2：通过排序计算温度时间序列的简单移动平均值

该示例使用按设备 ID 降序排序的两个和三个数据点来计算两个简单的温度移动平均线。

PPL 查询：

```
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|
+-----------+---------+--------------------+------------+------------------+
```

命令在哪里

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

该`where`命令使用布尔表达式来筛选搜索结果。只有当布尔表达式的计算结果为 true 时，它才会返回结果。

语法

使用以下语法：

```
where <boolean-expression>
```

布尔表达式

- 可选。
- 任何可以计算为布尔值的表达式。

示例 1：使用条件筛选结果集

该示例说明如何从账户索引中提取符合特定条件的文档。

PPL 查询：

```
os> source=accounts | where account_number=1 or gender="F" | fields account_number,
 gender;
```

```
fetched rows / total rows = 2/2
+------------------+----------+
| account_number   | gender   |
|------------------+----------|
| 1                | M        |
| 13               | F        |
+------------------+----------+
```

其他示例

具有逻辑条件的过滤器

- `source = table | where c = 'test' AND a = 1 | fields a,b,c`
- `source = table | where c != 'test' OR a > 1 | fields a,b,c | head 1`
- `source = table | where c = 'test' NOT a > 1 | fields a,b,c`
- `source = table | where a = 1 | fields a,b,c`
- `source = table | where a >= 1 | fields a,b,c`
- `source = table | where a < 1 | fields a,b,c`
- `source = table | where b != 'test' | fields a,b,c`
- `source = table | where c = 'test' | fields a,b,c | head 3`
- `source = table | where ispresent(b)`
- `source = table | where isnull(coalesce(a, b)) | fields a,b,c | head 3`
- `source = table | where isempty(a)`
- `source = table | where isblank(a)`
- `source = table | where case(length(a) > 6, 'True' else 'False') = 'True'`
- `source = table | where a between 1 and 4`-注意：这将返回 >= 1 和 <= 4，即 [1, 4]
- `source = table | where b not between '2024-09-10' and '2025-09-10'`-注意：这将返回 b >= '********** '和 b <=' 2025-09-10 '
- `source = table | where cidrmatch(ip, '***********/24')`
- `source = table | where cidrmatch(ipv6, '2003:db8::/32')`

```
source = table | eval status_category =
    case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
```

```
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
    else 'Incorrect HTTP status code')
    | where case(a >= 200 AND a < 300, 'Success',
    a >= 300 AND a < 400, 'Redirection',
    a >= 400 AND a < 500, 'Client Error',
    a >= 500, 'Server Error'
    else 'Incorrect HTTP status code'
    ) = 'Incorrect HTTP status code'
```

```
source = table
    | eval factor = case(a > 15, a - 14, isnull(b), a - 7, a < 3, a + 1 else 1)
    | where case(factor = 2, 'even', factor = 4, 'even', factor = 6, 'even', factor =
 8, 'even' else 'odd') = 'even'
    |  stats count() by factor
```

字段摘要

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 命令，请参阅。the section called “命令”

使用该`fieldsummary`命令计算每个字段的基本统计数据（计数、不同计数、最小值、最大值、平均值、标准差、平均值），并确定每个字段的数据类型。此命令可以与之前的任何管道一起使用，并将它们考虑在内。

语法

使用下面的语法。对于 CloudWatch 日志用例，查询中仅支持一个字段。

```
... | fieldsummary <field-list> (nulls=true/false)
```

包含字段

- 列出要与统计数据一起收集到统一结果集中的所有列。

null

- 可选。

- 如果设置为 true，则在聚合计算中包含空值（对于数值，将空值替换为零）。

示例 1

PPL 查询：

```
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"  |
+------------------+-------------+------------+------------+------------+------------
+------------+------------+----------------|
```

示例 2

PPL 查询：

```
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"|
+------------------+-------------+------------+------------+------------+------------
+------------+------------+----------------|
```

扩展命令

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

使用expand命令展平 Array <Any>或 Map 类型的字段<Any>，为每个元素或键值对生成单独的行。

语法

使用以下语法：

```
expand <field> [As alias]
```

字段

- 要扩展（分解）的字段。
- 该字段必须是支持的类型。

别名

- 可选。
- 要使用的名称而不是原始字段名称。

使用指南

expand 命令为指定数组或映射字段中的每个元素生成一行，其中：

- 数组元素变成单独的行。
- 地图键值对被分成不同的行，每个键值都表示为一行。
- 提供别名时，分解值将以别名而不是原始字段名表示。

您可以将此命令与其他命令（例如 stats、eval 和 parse）结合使用，在扩展后操作或提取数据。

示例

- `source = table | expand employee | stats max(salary) as max by state, company`

- `source = table | expand employee as worker | stats max(salary) as max by state, company`
- `source = table | expand employee as worker | eval bonus = salary * 3 | fields worker, bonus`
- `source = table | expand employee | parse description '(?<email>.+@.+)' | fields employee, email`
- `source = table | eval array=json_array(1, 2, 3) | expand array as uid | fields name, occupation, uid`
- `source = table | expand multi_valueA as multiA | expand multi_valueB as multiB`

您可以将 expand 命令与其他命令（例如 eval、stats 等）结合使用。使用多个 expand 命令将创建每个复合数组或地图中所有内部元素的笛卡尔乘积。

有效的 SQL 下推查询

扩展命令使用横向视图爆炸转换为等效的 SQL 操作，从而允许在 SQL 查询级别高效地分解数组或地图。

```
SELECT customer exploded_productId
FROM table
LATERAL VIEW explode(productId) AS exploded_productId
```

explode 命令提供以下功能：

- 这是一种返回新列的列操作。
- 它会为分解列中的每个元素创建一个新行。
- 作为分解字段的一部分，内部空值将被忽略（没有一行代表 created/exploded 空值）。

PPL 函数

主题

- PPL 条件函数
- PPL 加密哈希函数
- PPL 日期和时间函数

- PPL 表达式
- PPL IP 地址函数
- PPL JSON 函数
- PPL Lambda 函数
- PPL 数学函数
- PPL 字符串函数
- PPL 类型转换函数

PPL 条件函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

ISNULL

描述：如果该字段为空，则`isnull(field)`返回 true。

参数类型：

- 所有支持的数据类型。

返回类型：

- BOOLEAN

示例：

```
os> source=accounts | eval result = isnull(employer) | fields result, employer,
 firstname
fetched rows / total rows = 4/4
+----------+-------------+-------------+
| result   | employer    | firstname   |
|----------+-------------+-------------|
| False    | AnyCompany  | Mary        |
| False    | ExampleCorp | Jane        |
```

```
| False    | ExampleOrg  | Nikki       |
| True     | null        | Juan        |
+----------+-------------+-------------+
```

ISNOTNULL

描述：如果该字段不为空，则`isnotnull(field)`返回 true。

参数类型：

- 所有支持的数据类型。

返回类型：

- BOOLEAN

示例：

```
os> source=accounts | where not isnotnull(employer) | fields account_number, employer
fetched rows / total rows = 1/1
+------------------+------------+
| account_number   | employer   |
|------------------+------------|
| 18               | null       |
+------------------+------------+
```

EXISTS

示例：

```
os> source=accounts | where exists(email) | fields account_number, email
fetched rows / total rows = 1/1
```

如果为空

描述：`field2`如果`field1`为空则`ifnull(field1, field2)`返回。

参数类型：

- 所有支持的数据类型。

- 如果两个参数的类型不同，则该函数将无法通过语义检查。

返回类型：

- 任何

示例：

```
os> source=accounts | eval result = ifnull(employer, 'default') | fields result,
 employer, firstname
fetched rows / total rows = 4/4
+------------+------------+-------------+
| result     | employer   | firstname   |
|------------+------------+-------------|
| AnyCompany | AnyCompany | Mary        |
| ExampleCorp| ExampleCorp| Jane        |
| ExampleOrg | ExampleOrg | Nikki       |
| default    | null       | Juan        |
+------------+------------+-------------+
```

NULLIF

描述：如果两个参数相同，则`nullif(field1, field2)`返回 null，否则返回 field1。

参数类型：

- 所有支持的数据类型。
- 如果两个参数的类型不同，则该函数将无法通过语义检查。

返回类型：

- 任何

示例：

```
os> source=accounts | eval result = nullif(employer, 'AnyCompany') | fields result,
 employer, firstname
fetched rows / total rows = 4/4
+----------------+----------------+-------------+
| result         | employer       | firstname   |
```

```
|----------------+----------------+-------------|
| null           | AnyCompany     | Mary        |
| ExampleCorp    | ExampleCorp    | Jane        |
| ExampleOrg     | ExampleOrg     | Nikki       |
| null           | null           | Juan        |
+----------------+----------------+-------------+
```

IF

描述：`expr1`如果条件为真则`if(condition, expr1, expr2)`返回，否则返回`expr2`。

参数类型：

- 所有支持的数据类型。
- 如果两个参数的类型不同，则该函数将无法通过语义检查。

返回类型：

- 任何

示例：

```
os> source=accounts | eval result = if(true, firstname, lastname) | fields result,
 firstname, lastname
fetched rows / total rows = 4/4
+----------+-------------+----------+
| result   | firstname | lastname   |
|----------+-------------+----------|
| Jane     | Jane      | Doe        |
| Mary     | Mary      | Major      |
| Pat      | Pat       | Candella   |
| Dale     | Jorge     | Souza      |
+----------+-----------+------------+

os> source=accounts | eval result = if(false, firstname, lastname) | fields result,
 firstname, lastname
fetched rows / total rows = 4/4
+----------+-------------+------------+
| result   | firstname   | lastname   |
|----------+-------------+------------|
| Doe      | Jane        | Doe        |
```

```
| Major    | Mary        | Major      |
| Candella | Pat         | Candella   |
| Souza    | Jorge       | Souza      |
+----------+-------------+------------+

os> source=accounts | eval is_vip = if(age > 30 AND isnotnull(employer), true, false) |
 fields is_vip, firstname, lastname
fetched rows / total rows = 4/4
+----------+-------------+------------+
| is_vip   | firstname   | lastname   |
|----------+-------------+------------|
| True     | Jane        | Doe        |
| True     | Mary        | Major      |
| False    | Pat         | Candella   |
| False    | Jorge       | Souza      |
+----------+-------------+------------+
```

PPL 加密哈希函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

MD5

MD5 计算 MD5 摘要并以 32 个字符的十六进制字符串形式返回该值。

用法：`md5('hello')`

参数类型：

- string

返回类型：

- string

示例：

```
os> source=people | eval `MD5('hello')` = MD5('hello') | fields `MD5('hello')`
```

```
fetched rows / total rows = 1/1
+----------------------------------+
| MD5('hello')                     |
|----------------------------------|
| <32 character hex string>        |
+----------------------------------+
```

SHA1

SHA1 返回 SHA-1 的十六进制字符串结果。

用法：`sha1('hello')`

参数类型：

- string

返回类型：

- string

示例：

```
os> source=people | eval `SHA1('hello')` = SHA1('hello') | fields `SHA1('hello')`
fetched rows / total rows = 1/1
+-------------------------------------------+
| SHA1('hello')                             |
|-------------------------------------------|
| <40-character SHA-1 hash result>          |
+-------------------------------------------+
```

SHA2

SHA2 返回 SHA-2 系列哈希函数（SHA-224、SHA-256、SHA-384 和 SHA-512）的十六进制字符串结果。Numbits 表示结果所需的位长，其值必须为 224、256、384、512

用法：

- `sha2('hello',256)`
- `sha2('hello',512)`

参数类型：

- 字符串，整数

返回类型：

- string

示例：

```
os> source=people | eval `SHA2('hello',256)` = SHA2('hello',256) | fields
 `SHA2('hello',256)`
fetched rows / total rows = 1/1
+------------------------------------------------------------------+
| SHA2('hello',256)                                                |
|------------------------------------------------------------------|
| <64-character SHA-256 hash result>                               |
+------------------------------------------------------------------+

os> source=people | eval `SHA2('hello',512)` = SHA2('hello',512) | fields
 `SHA2('hello',512)`
fetched rows / total rows = 1/1
+------------------------------------------------------------------+
| SHA2('hello',512)                                                |
                                                |
|------------------------------------------------------------------|
| <128-character SHA-512 hash result>                              |
+------------------------------------------------------------------+
```

PPL 日期和时间函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

DAY

用法：`DAY(date)`提取日期的月份中的某一天，范围为 1 到 31。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词:DAYOFMONTH, DAY_OF_MONTH

示例：

```
os> source=people | eval `DAY(DATE('2020-08-26'))` = DAY(DATE('2020-08-26')) | fields
 `DAY(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------+
| DAY(DATE('2020-08-26'))   |
|---------------------------|
| 26                        |
+---------------------------+
```

DAYOFMONTH

用法：DAYOFMONTH(date)提取日期的月份中的某一天，范围为 1 到 31。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词:DAY, DAY_OF_MONTH

示例：

```
os> source=people | eval `DAYOFMONTH(DATE('2020-08-26'))` =
 DAYOFMONTH(DATE('2020-08-26')) | fields `DAYOFMONTH(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+----------------------------------+
| DAYOFMONTH(DATE('2020-08-26'))   |
|----------------------------------|
| 26                               |
+----------------------------------+
```

DAY_OF_MONTH

用法：DAY_OF_MONTH(DATE)提取日期的月份中的某一天，范围为 1 到 31。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词:DAY, DAYOFMONTH

示例：

```
os> source=people | eval `DAY_OF_MONTH(DATE('2020-08-26'))` =
 DAY_OF_MONTH(DATE('2020-08-26')) | fields `DAY_OF_MONTH(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| DAY_OF_MONTH(DATE('2020-08-26'))   |
|-----------------------------------|
| 26                                |
+-----------------------------------+
```

DAYOFWEEK

用法：DAYOFWEEK(DATE)返回某一日期的工作日索引（1 = 星期日，2 = 星期一，...，7 = 星期六）。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词：DAY_OF_WEEK

示例：

```
os> source=people | eval `DAYOFWEEK(DATE('2020-08-26'))` =
 DAYOFWEEK(DATE('2020-08-26')) | fields `DAYOFWEEK(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------------+
| DAYOFWEEK(DATE('2020-08-26'))   |
|---------------------------------|
| 4                               |
+---------------------------------+
```

DAY_OF_WEEK

用法：DAY_OF_WEEK(DATE)返回某一日期的工作日索引（1 = 星期日，2 = 星期一，...，7 = 星期六）。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词：DAYOFWEEK

示例：

```
os> source=people | eval `DAY_OF_WEEK(DATE('2020-08-26'))` =
 DAY_OF_WEEK(DATE('2020-08-26')) | fields `DAY_OF_WEEK(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| DAY_OF_WEEK(DATE('2020-08-26'))   |
|-----------------------------------|
| 4                                 |
+-----------------------------------+
```

DAYOFYEAR

用法：DAYOFYEAR(DATE)返回日期的一年中的某一天，范围为 1 到 366。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词：DAY_OF_YEAR

示例：

```
os> source=people | eval `DAYOFYEAR(DATE('2020-08-26'))` =
 DAYOFYEAR(DATE('2020-08-26')) | fields `DAYOFYEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------------+
| DAYOFYEAR(DATE('2020-08-26'))   |
|---------------------------------|
| 239                             |
+---------------------------------+
```

DAY_OF_YEAR

用法：DAY_OF_YEAR(DATE)返回日期的一年中的某一天，范围为 1 到 366。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词：DAYOFYEAR

示例：

```
os> source=people | eval `DAY_OF_YEAR(DATE('2020-08-26'))` =
 DAY_OF_YEAR(DATE('2020-08-26')) | fields `DAY_OF_YEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| DAY_OF_YEAR(DATE('2020-08-26'))   |
|-----------------------------------|
| 239                               |
+-----------------------------------+
```

DAYNAME

用法：DAYNAME(DATE)返回某一日期的工作日名称，包括星期一、星期二、星期三、星期四、星期五、星期六和星期日。

参数类型：STRING/DATE/TIMESTAMP

返回类型：字符串

示例：

```
os> source=people | eval `DAYNAME(DATE('2020-08-26'))` = DAYNAME(DATE('2020-08-26')) |
 fields `DAYNAME(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-------------------------------+
| DAYNAME(DATE('2020-08-26'))   |
|-------------------------------|
| Wednesday                     |
+-------------------------------+
```

FROM_UNIXTIME

用法：FROM_UNIXTIME返回以时间戳或字符串值形式给出的参数的表示形式。此函数执行函数的反向转换。UNIX_TIMESTAMP

如果您提供第二个参数，则FROM_UNIXTIME使用它来格式化与DATE_FORMAT函数类似的结果。

如果时间戳超出 1970-01-01 00:00:00 到 3001-01-18 23:59:59.999 999（0 到 32536771199.99999 纪元时间）的范围，则该函数返回。NULL

参数类型：双精度、字符串

返回类型地图：

双精度->时间戳

双精度，字符串-> 字符串

示例：

```
os> source=people | eval `FROM_UNIXTIME(1220249547)` = FROM_UNIXTIME(1220249547) |
 fields `FROM_UNIXTIME(1220249547)`
fetched rows / total rows = 1/1
+-----------------------------+
| FROM_UNIXTIME(1220249547)   |
|-----------------------------|
| 2008-09-01 06:12:27         |
+-----------------------------+

os> source=people | eval `FROM_UNIXTIME(1220249547, 'HH:mm:ss')` =
 FROM_UNIXTIME(1220249547, 'HH:mm:ss') | fields `FROM_UNIXTIME(1220249547, 'HH:mm:ss')`
fetched rows / total rows = 1/1
+-----------------------------------------+
| FROM_UNIXTIME(1220249547, 'HH:mm:ss')   |
|-----------------------------------------|
| 06:12:27                                |
+-----------------------------------------+
```

HOUR

用法：HOUR(TIME)提取时间的小时值。

与一天中的标准时间不同，此函数中的时间值的范围可以大于 23。因此，的返回值HOUR(TIME)可以大于 23。

参数类型：STRING/TIME/TIMESTAMP

返回类型：整数

同义词：HOUR_OF_DAY

示例：

```
os> source=people | eval `HOUR(TIME('01:02:03'))` = HOUR(TIME('01:02:03')) | fields
 `HOUR(TIME('01:02:03'))`
```

```
fetched rows / total rows = 1/1
+--------------------------+
| HOUR(TIME('01:02:03'))   |
|--------------------------|
| 1                        |
+--------------------------+
```

HOUR_OF_DAY

用法：从给定时间HOUR_OF_DAY(TIME)提取小时值。

与一天中的标准时间不同，此函数中的时间值的范围可以大于 23。因此，的返回值HOUR_OF_DAY(TIME)可以大于 23。

参数类型：STRING/TIME/TIMESTAMP

返回类型：整数

同义词：HOUR

示例：

```
os> source=people | eval `HOUR_OF_DAY(TIME('01:02:03'))` =
 HOUR_OF_DAY(TIME('01:02:03')) | fields `HOUR_OF_DAY(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+---------------------------------+
| HOUR_OF_DAY(TIME('01:02:03'))   |
|---------------------------------|
| 1                               |
+---------------------------------+
```

LAST_DAY

用法：将该月的最后一天作为给定日期参数的 DATE 值LAST_DAY返回。

参数类型：DATE/STRING/TIMESTAMP/TIME

返回类型：日期

示例：

```
os> source=people | eval `last_day('2023-02-06')` = last_day('2023-02-06') | fields
 `last_day('2023-02-06')`
```

```
fetched rows / total rows = 1/1
+--------------------------+
| last_day('2023-02-06')   |
|--------------------------|
| 2023-02-28               |
+--------------------------+
```

LOCALTIMESTAMP

用法:LOCALTIMESTAMP()是的同义词NOW()。

示例：

```
> source=people | eval `LOCALTIMESTAMP()` = LOCALTIMESTAMP() | fields
 `LOCALTIMESTAMP()`
fetched rows / total rows = 1/1
+---------------------+
| LOCALTIMESTAMP()    |
|---------------------|
| 2022-08-02 15:54:19 |
+---------------------+
```

LOCALTIME

用法:LOCALTIME()是的同义词。NOW()

示例：

```
> source=people | eval `LOCALTIME()` = LOCALTIME() | fields `LOCALTIME()`
fetched rows / total rows = 1/1
+---------------------+
| LOCALTIME()         |
|---------------------|
| 2022-08-02 15:54:19 |
+---------------------+
```

MAKE_DATE

用法：根据给定的年、月和日值MAKE_DATE返回日期值。所有参数均四舍五入为整数。

规格：1. MAKE_DATE（整数、整数、整数）-> 日期

参数类型：整数、整数、整数

返回类型：日期

示例：

```
os> source=people | eval `MAKE_DATE(1945, 5, 9)` = MAKEDATE(1945, 5, 9) | fields
 `MAKEDATE(1945, 5, 9)`
fetched rows / total rows = 1/1
+------------------------+
| MAKEDATE(1945, 5, 9)   |
|------------------------|
| 1945-05-09             |
+------------------------+
```

MINUTE

用法：以 0 到 59 范围内的整数形式`MINUTE(TIME)`返回给定时间的分钟部分。

参数类型：STRING/TIME/TIMESTAMP

返回类型：整数

同义词：`MINUTE_OF_HOUR`

示例：

```
os> source=people | eval `MINUTE(TIME('01:02:03'))` =  MINUTE(TIME('01:02:03')) |
 fields `MINUTE(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+----------------------------+
| MINUTE(TIME('01:02:03'))   |
|----------------------------|
| 2                          |
+----------------------------+
```

MINUTE_OF_HOUR

用法：以 0 到 59 范围内的整数形式`MINUTE_OF_HOUR(TIME)`返回给定时间的分钟部分。

参数类型：STRING/TIME/TIMESTAMP

返回类型：整数

同义词：MINUTE

示例：

```
os> source=people | eval `MINUTE_OF_HOUR(TIME('01:02:03'))` =
 MINUTE_OF_HOUR(TIME('01:02:03')) | fields `MINUTE_OF_HOUR(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| MINUTE_OF_HOUR(TIME('01:02:03'))   |
|-----------------------------------|
| 2                                 |
+-----------------------------------+
```

MONTH

用法：以整数形式MONTH(DATE)返回给定日期的月份，范围为 1 到 12（其中 1 代表一月，12 代表十二月）。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词：MONTH_OF_YEAR

示例：

```
os> source=people | eval `MONTH(DATE('2020-08-26'))` =  MONTH(DATE('2020-08-26')) |
 fields `MONTH(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+----------------------------+
| MONTH(DATE('2020-08-26'))   |
|----------------------------|
| 8                          |
+----------------------------+
```

MONTHNAME

用法：以整数形式MONTHNAME(DATE)返回给定日期的月份，范围为 1 到 12（其中 1 代表一月，12 代表十二月）。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词：MONTH_OF_YEAR

示例：

```
os> source=people | eval `MONTHNAME(DATE('2020-08-26'))` =
 MONTHNAME(DATE('2020-08-26')) | fields `MONTHNAME(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+---------------------------------+
| MONTHNAME(DATE('2020-08-26'))   |
|---------------------------------|
| August                          |
+---------------------------------+
```

MONTH_OF_YEAR

用法：以整数形式MONTH_OF_YEAR(DATE)返回给定日期的月份，范围为 1 到 12（其中 1 代表一月，12 代表十二月）。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

同义词：MONTH

示例：

```
os> source=people | eval `MONTH_OF_YEAR(DATE('2020-08-26'))` =
 MONTH_OF_YEAR(DATE('2020-08-26')) | fields `MONTH_OF_YEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-------------------------------------+
| MONTH_OF_YEAR(DATE('2020-08-26'))   |
|-------------------------------------|
| 8                                   |
+-------------------------------------+
```

NOW

用法：NOW以 'YYYY-MM-DD hh: mm: s TIMESTAMP s' 格式返回当前日期和时间。该值以集群时区表示。

> **Note**
>
> NOW()返回一个表示语句何时开始执行的常量时间。这不同于SYSDATE()，后者返回确切的执行时间。

返回类型：时间戳

规范：NOW ()-> 时间戳

示例：

```
os> source=people | eval `value_1` = NOW(), `value_2` = NOW() | fields `value_1`,
 `value_2`
fetched rows / total rows = 1/1
+---------------------+---------------------+
| value_1             | value_2             |
|---------------------+---------------------|
| 2022-08-02 15:39:05 | 2022-08-02 15:39:05 |
+---------------------+---------------------+
```

QUARTER

用法：以整数形式QUARTER(DATE)返回给定日期的一年中的季度，范围为 1 到 4。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

示例：

```
os> source=people | eval `QUARTER(DATE('2020-08-26'))` = QUARTER(DATE('2020-08-26')) |
 fields `QUARTER(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+-------------------------------+
| QUARTER(DATE('2020-08-26'))   |
|-------------------------------|
| 3                             |
+-------------------------------+
```

SECOND

用法：以整数形式SECOND(TIME)返回给定时间的第二个分量，范围为 0 到 59。

参数类型：STRING/TIME/TIMESTAMP

返回类型：整数

同义词：SECOND_OF_MINUTE

示例：

```
os> source=people | eval `SECOND(TIME('01:02:03'))` = SECOND(TIME('01:02:03')) | fields
 `SECOND(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+----------------------------+
| SECOND(TIME('01:02:03'))   |
|----------------------------|
| 3                          |
+----------------------------+
```

SECOND_OF_MINUTE

用法：以整数形式SECOND_OF_MINUTE(TIME)返回给定时间的第二个分量，范围为 0 到 59。

参数类型：STRING/TIME/TIMESTAMP

返回类型：整数

同义词：SECOND

示例：

```
os> source=people | eval `SECOND_OF_MINUTE(TIME('01:02:03'))` =
 SECOND_OF_MINUTE(TIME('01:02:03')) | fields `SECOND_OF_MINUTE(TIME('01:02:03'))`
fetched rows / total rows = 1/1
+--------------------------------------+
| SECOND_OF_MINUTE(TIME('01:02:03'))   |
|--------------------------------------|
| 3                                    |
+--------------------------------------+
```

SUBDATE

用法：从给定日期中SUBDATE(DATE, DAYS)减去第二个参数（例如DATE或DAYS）。

参数类型：日期/时间戳，长整型

返回类型地图：（日期，长）-> 日期

反义词：ADDDATE

示例：

```
os> source=people | eval `'2008-01-02' - 31d` = SUBDATE(DATE('2008-01-02'), 31),
 `'2020-08-26' - 1` = SUBDATE(DATE('2020-08-26'), 1), `ts '2020-08-26 01:01:01' -
 1` = SUBDATE(TIMESTAMP('2020-08-26 01:01:01'), 1) | fields `'2008-01-02' - 31d`,
 `'2020-08-26' - 1`, `ts '2020-08-26 01:01:01' - 1`
fetched rows / total rows = 1/1
+----------------------+--------------------+--------------------------------+
| '2008-01-02' - 31d   | '2020-08-26' - 1   | ts '2020-08-26 01:01:01' - 1   |
|----------------------+--------------------+--------------------------------|
| 2007-12-02 00:00:00  | 2020-08-25         | 2020-08-25 01:01:01            |
+----------------------+--------------------+--------------------------------+
```

SYSDATE

用法：以 'YYYY-MM-DD hh: mm: ss.nnnnnn' 格式的TIMESTAMP值SYSDATE()返回当前日期和时间。

SYSDATE()返回其执行的确切时间。这与 NOW () 不同，后者返回一个恒定时间，指示语句何时开始执行。

可选参数类型：INTEGER（0 到 6）-指定返回值中小数秒的位数。

返回类型：时间戳

示例：

```
os> source=people | eval `SYSDATE()` = SYSDATE() | fields `SYSDATE()`
fetched rows / total rows = 1/1
+----------------------------+
| SYSDATE()                  |
|----------------------------|
| 2022-08-02 15:39:05.123456 |
+----------------------------+
```

TIMESTAMP

用法：TIMESTAMP(EXPR)构造一个以输入字符串expr作为时间戳的时间戳类型。

使用单个参数，从输入中TIMESTAMP(expr)构造一个时间戳。如果expr是字符串，则将其解释为时间戳。对于非字符串参数，该函数使用 UTC 时区expr转换为时间戳。当expr是TIME值时，该函数在转换之前应用今天的日期。

与两个参数一起使用时，TIMESTAMP(expr1, expr2)将时间表达式 (expr2) 添加到日期或时间戳表达式 (expr1)，并将结果作为时间戳值返回。

参数类型：STRING/DATE/TIME/TIMESTAMP

返回类型地图：

(STRING/DATE/TIME/TIMESTAMP)-> 时间戳

(STRING/DATE/TIME/TIMESTAMP, STRING/DATE/TIME/TIMESTAMP)-> 时间戳

示例：

```
os> source=people | eval `TIMESTAMP('2020-08-26 13:49:00')` = TIMESTAMP('2020-08-26
 13:49:00'), `TIMESTAMP('2020-08-26 13:49:00', TIME('12:15:42'))` =
 TIMESTAMP('2020-08-26 13:49:00', TIME('12:15:42')) | fields `TIMESTAMP('2020-08-26
 13:49:00')`, `TIMESTAMP('2020-08-26 13:49:00', TIME('12:15:42'))`
fetched rows / total rows = 1/1
+------------------------------------
+-------------------------------------------------------+
| TIMESTAMP('2020-08-26 13:49:00')   | TIMESTAMP('2020-08-26 13:49:00',
 TIME('12:15:42'))   |
|------------------------------------
+-------------------------------------------------------|
| 2020-08-26 13:49:00                | 2020-08-27 02:04:42
     |
+------------------------------------
+-------------------------------------------------------+
```

UNIX_TIMESTAMP

用法：UNIX_TIMESTAMP将给定的日期参数转换为 Unix 时间（自 1970 年初开始的纪元以来的秒数）。如果未提供任何参数，则返回当前 Unix 时间。

日期参数可以是DATE、TIMESTAMP字符串或以下格式之一的数字：YYMMDD、YYMMDDhhmmssYYYYMMDD、或YYYYMMDDhhmmss。如果参数包含时间分量，则可以选择包含小数秒。

如果参数的格式无效或超出了 1970-01-01 00:00:00 到 3001-01-18 23:59:59.999 999（纪元时间为 0 到 32536771199.999999）的范围，则该函数返回。NULL

该函数接受DATE、TIMESTAMP、或DOUBLE作为参数类型，或者不接受任何参数。它总是返回一个代表 Unix 时间戳的DOUBLE值。

要进行反向转换，可以使用 FROM_UNIXTIME 函数。

参数类型：<NONE>/DOUBLE/DATE/TIMESTAMP

返回类型：DOUBLE

示例：

```
os> source=people | eval `UNIX_TIMESTAMP(double)` = UNIX_TIMESTAMP(20771122143845),
 `UNIX_TIMESTAMP(timestamp)` = UNIX_TIMESTAMP(TIMESTAMP('1996-11-15 17:05:42')) |
 fields `UNIX_TIMESTAMP(double)`, `UNIX_TIMESTAMP(timestamp)`
fetched rows / total rows = 1/1
+--------------------------+----------------------------+
| UNIX_TIMESTAMP(double)   | UNIX_TIMESTAMP(timestamp)   |
|--------------------------+----------------------------|
| 3404817525.0             | 848077542.0                 |
+--------------------------+----------------------------+
```

WEEK

用法：WEEK(DATE)返回给定日期的星期数。

参数类型：DATE/TIMESTAMP/STRING

返回类型：整数

同义词：WEEK_OF_YEAR

示例：

```
os> source=people | eval `WEEK(DATE('2008-02-20'))` = WEEK(DATE('2008-02-20')) | fields
 `WEEK(DATE('2008-02-20'))`
fetched rows / total rows = 1/1
+----------------------------+
| WEEK(DATE('2008-02-20'))   |
|----------------------------|
| 8                          |
```

```
+----------------------------+
```

WEEKDAY

用法：WEEKDAY(DATE)返回日期的工作日索引（0 = 星期一，1 = 星期二，...，6 = 星期日）。

它与dayofweek函数类似，但每天返回不同的索引。

参数类型：STRING/DATE/TIME/TIMESTAMP

返回类型：整数

示例：

```
os> source=people | eval `weekday(DATE('2020-08-26'))` = weekday(DATE('2020-08-26'))
 | eval `weekday(DATE('2020-08-27'))` = weekday(DATE('2020-08-27')) | fields
 `weekday(DATE('2020-08-26'))`, `weekday(DATE('2020-08-27'))`
fetched rows / total rows = 1/1
+-------------------------------+-------------------------------+
| weekday(DATE('2020-08-26'))   | weekday(DATE('2020-08-27'))   |
|-------------------------------+-------------------------------|
| 2                             | 3                             |
+-------------------------------+-------------------------------+
```

WEEK_OF_YEAR

用法：WEEK_OF_YEAR(DATE)返回给定日期的星期数。

参数类型：DATE/TIMESTAMP/STRING

返回类型：整数

同义词：WEEK

示例：

```
os> source=people | eval `WEEK_OF_YEAR(DATE('2008-02-20'))` = WEEK(DATE('2008-02-20'))|
 fields `WEEK_OF_YEAR(DATE('2008-02-20'))`
fetched rows / total rows = 1/1
+-----------------------------------+
| WEEK_OF_YEAR(DATE('2008-02-20'))   |
|-----------------------------------|
```

```
| 8                                   |
+-------------------------------------+
```

YEAR

用法：YEAR(DATE)返回日期的年份，范围在 1000 到 9999 之间，或者返回 0 表示 “零” 日期。

参数类型：STRING/DATE/TIMESTAMP

返回类型：整数

示例：

```
os> source=people | eval `YEAR(DATE('2020-08-26'))` = YEAR(DATE('2020-08-26')) | fields
 `YEAR(DATE('2020-08-26'))`
fetched rows / total rows = 1/1
+----------------------------+
| YEAR(DATE('2020-08-26'))   |
|----------------------------|
| 2020                       |
+----------------------------+
```

DATE_ADD

用法：DATE_ADD(date, INTERVAL expr unit)将指定的间隔与给定日期相加。

参数类型：日期、间隔

返回类型：日期

反义词：DATE_SUB

示例：

```
os> source=people | eval `'2020-08-26' + 1d` = DATE_ADD(DATE('2020-08-26'), INTERVAL 1
 DAY) | fields `'2020-08-26' + 1d`
fetched rows / total rows = 1/1
+---------------------+
| '2020-08-26' + 1d   |
|---------------------|
| 2020-08-27          |
+---------------------+
```

DATE_SUB

用法：DATE_SUB(date, INTERVAL expr unit)从日期中减去间隔 expr。

参数类型：日期、间隔

返回类型：日期

反义词：DATE_ADD

示例：

```
os> source=people | eval `'2008-01-02' - 31d` = DATE_SUB(DATE('2008-01-02'), INTERVAL
 31 DAY) | fields `'2008-01-02' - 31d`
fetched rows / total rows = 1/1
+---------------------+
| '2008-01-02' - 31d  |
|---------------------|
| 2007-12-02          |
+---------------------+
```

TIMESTAMPADD

用法：将指定的时间间隔与给定日期相加后返回一个TIMESTAMP值。

论点：

- 间隔：间隔（秒、分钟、小时、日、周、月、季度、年）
- 整数：整数
- 日期：日期、时间戳或字符串

如果您提供STRING作为日期参数，请将其格式化为有效的参数TIMESTAMP。该函数会自动将DATE参数转换为 a TIMESTAMP。

示例：

```
os> source=people | eval `TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00')` =
 TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00') | eval `TIMESTAMPADD(QUARTER, -1,
```

```
 '2000-01-01 00:00:00')` = TIMESTAMPADD(QUARTER, -1, '2000-01-01 00:00:00') | fields
 `TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00')`, `TIMESTAMPADD(QUARTER, -1, '2000-01-01
 00:00:00')`
fetched rows / total rows = 1/1
+-----------------------------------------------
+---------------------------------------------------+
| TIMESTAMPADD(DAY, 17, '2000-01-01 00:00:00') | TIMESTAMPADD(QUARTER, -1, '2000-01-01
 00:00:00') |
|-----------------------------------------------
+---------------------------------------------------|
| 2000-01-18 00:00:00                          | 1999-10-01 00:00:00
           |
+-----------------------------------------------
+---------------------------------------------------+
```

TIMESTAMPDIFF

用法：以指定的间隔单位`TIMESTAMPDIFF(interval, start, end)`返回开始和结束 date/times 之间的差值。

论点：

- 间隔：间隔（秒、分钟、小时、日、周、月、季度、年）
- 开始：日期、时间戳或字符串
- 结束：日期、时间戳或字符串

该函数会在适当`TIMESTAMP`时自动将参数转换为。将`STRING`参数格式化为有效的 `TIMESTAMP` s。

示例：

```
os> source=people | eval `TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00', '2001-03-06
 00:00:00')` = TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00', '2001-03-06 00:00:00') |
 eval `TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'), timestamp('1997-01-01
 00:00:00'))` = TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'),
 timestamp('1997-01-01 00:00:00')) | fields `TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00',
 '2001-03-06 00:00:00')`, `TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'),
 timestamp('1997-01-01 00:00:00'))`
fetched rows / total rows = 1/1
+-----------------------------------------------------------------
+-------------------------------------------------------------------------------------------
+
```

```
| TIMESTAMPDIFF(YEAR, '1997-01-01 00:00:00', '2001-03-06 00:00:00') |
 TIMESTAMPDIFF(SECOND, timestamp('1997-01-01 00:00:23'), timestamp('1997-01-01
 00:00:00')) |
|-----------------------------------------------------------------
+-------------------------------------------------------------------------------------------------|
| 4                                                                  | -23
                                                                        |
+-----------------------------------------------------------------
+-------------------------------------------------------------------------------------------------
+
```

UTC_TIMESTAMP

用法：以 “YYYY-MM-DD hh: mm: ss” 中的值形式`UTC_TIMESTAMP`返回当前 UTC 时间戳。

返回类型：时间戳

规范：UTC_TIMESTAMP ()-> TIMESTAMP

示例：

```
> source=people | eval `UTC_TIMESTAMP()` = UTC_TIMESTAMP() | fields `UTC_TIMESTAMP()`
fetched rows / total rows = 1/1
+---------------------+
| UTC_TIMESTAMP()     |
|---------------------|
| 2022-10-03 17:54:28 |
+---------------------+
```

CURRENT_TIMEZONE

用法：`CURRENT_TIMEZONE`返回当前的本地时区。

返回类型：字符串

示例：

```
> source=people | eval `CURRENT_TIMEZONE()` = CURRENT_TIMEZONE() | fields
 `CURRENT_TIMEZONE()`
fetched rows / total rows = 1/1
+------------------------+
| CURRENT_TIMEZONE()     |
|------------------------|
```

```
| America/Chicago        |
+------------------------+
```

PPL 表达式

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

表达式，尤其是值表达式，返回标量值。表达式有不同的类型和形式。例如，有诸如原子表达式的字面值以及基于它们的算术、谓词和函数表达式。可以在不同的子句中使用表达式，例如在`Filter`和`Stats`命令中使用算术表达式。

运算符

算术表达式是由数字文字和二进制算术运算符组成的表达式，如下所示：

1. `+`: 添加。
2. `-`: 减去。
3. `*`: 相乘。
4. `/`: 除法（对于整数，结果是一个整数，其中小数部分被丢弃）
5. `%`: Modulo（仅适用于整数；结果是除法的余数）

优先级

使用圆括号控制算术运算符的优先级。否则，优先级较高的运算符将首先执行。

类型转换

在查找运算符签名时会执行隐式类型转换。例如，一个整数`+`是一个实数，与生成实数`+(double,double)`的签名相匹配。此规则也适用于函数调用。

不同类型的算术表达式的示例：

```
os> source=accounts | where age > (25 + 5) | fields age ;
fetched rows / total rows = 3/3
+-------+
| age   |
```

```
|-------|
| 32    |
| 36    |
| 33    |
+-------+
```

谓词运算符

谓词运算符是计算结果为真的表达式。MISSING和NULL值比较遵循以下规则：

- 一个MISSING值只等于一个MISSING值并且小于其他值。
- 一个NULL值等于一个NULL值，大于一个MISSING值，但小于所有其他值。

运算符

谓词运算符

名称	描述
>	大于运算符
>=	大于或等于运算符
<	小于运算符
!=	不等于运算符
<=	小于或等于运算符
=	等于运算符
LIKE	简单的图案匹配
IN	空值测试
AND	AND 运算符
OR	OR 运算符
XOR	异或运算符
NOT	非空值测试

您可以比较日期时间。比较不同的日期时间类型（例如DATE和TIME）时，两者都将转换为。DATETIME以下规则适用于转换：

- TIME适用于今天的日期。
- DATE在午夜解释。

基本谓词运算符

比较运算符的示例：

```
os> source=accounts | where age > 33 | fields age ;
fetched rows / total rows = 1/1
+-------+
| age   |
|-------|
| 36    |
+-------+
```

IN

值列表中IN运算符测试字段的示例：

```
os> source=accounts | where age in (32, 33) | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 32    |
| 33    |
+-------+
```

OR

OR运算符示例：

```
os> source=accounts | where age = 32 OR age = 33 | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 32    |
| 33    |
```

```
+-------+
```

NOT

NOT运算符示例：

```
os> source=accounts | where age not in (32, 33) | fields age ;
fetched rows / total rows = 2/2
+-------+
| age   |
|-------|
| 36    |
| 28    |
+-------+
```

PPL IP 地址函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

CIDRMATCH

用法：`CIDRMATCH(ip, cidr)`检查指定的 IP 地址是否在给定的 cidr 范围内。

参数类型：

- 字符串，字符串
- 返回类型：布尔值

示例：

```
os> source=ips | where cidrmatch(ip, '***********/24') | fields ip
fetched rows / total rows = 1/1
+--------------+
| ip           |
|--------------|
| ***********  |
+--------------+
```

```
os> source=ipsv6 | where cidrmatch(ip, '2003:db8::/32') | fields ip
fetched rows / total rows = 1/1
+-----------------------------------------+
| ip                                      |
|-----------------------------------------|
| 2003:0db8:****:****:****:****:****:0000 |
+-----------------------------------------+
```

> **Note**
>
> - `ip`可以是 IPv4 或 IPv6地址。
> - `cidr`可以是一个 IPv4 或一个 IPv6方块。
> - `ip`并且`cidr`必须两者兼 IPv4 而有之 IPv6。
> - `ip`并且`cidr`必须同时有效且为非空/非空。

PPL JSON 函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

JSON

用法：`json(value)`评估字符串是否可以解析为 JSON 格式。如果是有效的 JSON，则该函数返回原始字符串；如果无效，则返回 null。

参数类型：字符串

返回类型：字符串/空。有效 JSON 对象格式的字符串表达式。

示例：

```
os> source=people | eval `valid_json()` = json('[1,2,3,{"f1":1,"f2":[5,6]},4]') |
 fields valid_json
fetched rows / total rows = 1/1
+---------------------------------+
| valid_json                      |
+---------------------------------+
```

```
| [1,2,3,{"f1":1,"f2":[5,6]},4]   |
+---------------------------------+

os> source=people | eval `invalid_json()` = json('{"invalid": "json"') | fields
 invalid_json
fetched rows / total rows = 1/1
+----------------+
| invalid_json   |
+----------------+
| null           |
+----------------+
```

JSON_OBJECT

用法：从键值对的成员`json_object(<key>, <value>[, <key>, <value>]...)`返回 JSON 对象。

参数类型：

- A <key>必须是字符串。
- A <value>可以是任何数据类型。

返回类型：JSON_OBJECT。有效 JSON 对象的 StructType 表达式。

示例：

```
os> source=people | eval result = json_object('key', 123.45) | fields result
fetched rows / total rows = 1/1
+------------------+
| result           |
+------------------+
| {"key":123.45}   |
+------------------+

os> source=people | eval result = json_object('outer', json_object('inner', 123.45)) |
 fields result
fetched rows / total rows = 1/1
+------------------------------+
| result                       |
+------------------------------+
| {"outer":{"inner":123.45}}   |
+------------------------------+
```

JSON_ARRAY

用法：使用值列表`json_array(<value>...)`创建 JSON 数组。

参数类型：A `<value>` 可以是任何类型的值，例如字符串、数字或布尔值。

返回类型：数组。有效 JSON 数组所支持的任何数据类型的数组。

示例：

```
os> source=people | eval `json_array` = json_array(1, 2, 0, -1, 1.1, -0.11)
fetched rows / total rows = 1/1
+------------------------------+
| json_array                   |
+------------------------------+
| [1.0,2.0,0.0,-1.0,1.1,-0.11] |
+------------------------------+

os> source=people | eval `json_array_object` = json_object("array", json_array(1, 2, 0,
 -1, 1.1, -0.11))
fetched rows / total rows = 1/1
+----------------------------------------+
| json_array_object                      |
+----------------------------------------+
| {"array":[1.0,2.0,0.0,-1.0,1.1,-0.11]} |
+----------------------------------------+
```

TO_JSON_STRING

用法：`to_json_string(jsonObject)`返回具有给定 json 对象值的 JSON 字符串。

参数类型：JSON_OBJECT

返回类型：字符串

示例：

```
os> source=people | eval `json_string` = to_json_string(json_array(1, 2, 0, -1, 1.1,
 -0.11)) | fields json_string
fetched rows / total rows = 1/1
+--------------------------------+
| json_string                    |
+--------------------------------+
| [1.0,2.0,0.0,-1.0,1.1,-0.11]   |
```

```
+--------------------------------+

os> source=people | eval `json_string` = to_json_string(json_object('key', 123.45)) |
 fields json_string
fetched rows / total rows = 1/1
+-----------------+
| json_string     |
+-----------------+
| {'key', 123.45} |
+-----------------+
```

ARRAY_LENGTH

用法：`array_length(jsonArray)`返回最外层数组中元素的数量。

参数类型：数组。一个数组或 JSON_ARRAY 对象。

返回类型：整数

示例：

```
os> source=people | eval `json_array` = json_array_length(json_array(1,2,3,4)),
 `empty_array` = json_array_length(json_array())
fetched rows / total rows = 1/1
+--------------+---------------+
| json_array   | empty_array   |
+--------------+---------------+
| 4            | 0             |
+--------------+---------------+
```

JSON_EXTRACT

用法：根据指定的 JSON 路径从 JSON 字符串中`json_extract(jsonStr, path)`提取 JSON 对象。如果输入 JSON 字符串无效，则该函数返回 null。

参数类型：字符串、字符串

返回类型：字符串

- 有效 JSON 对象格式的字符串表达式。
- NULL如果 JSON 无效，则会返回。

示例：

```
os> source=people | eval `json_extract('{"a":"b"}', '$.a')` = json_extract('{"a":"b"}',
 '$a')
fetched rows / total rows = 1/1
+----------------------------------+
| json_extract('{"a":"b"}', 'a')   |
+----------------------------------+
| b                                |
+----------------------------------+

os> source=people | eval `json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[1].b')` =
 json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[1].b')
fetched rows / total rows = 1/1
+-------------------------------------------------------------+
| json_extract('{"a":[{"b":1.0},{"b":2.0}]}', '$.a[1].b')   |
+-------------------------------------------------------------+
| 2.0                                                         |
+-------------------------------------------------------------+

os> source=people | eval `json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[*].b')` =
 json_extract('{"a":[{"b":1},{"b":2}]}', '$.a[*].b')
fetched rows / total rows = 1/1
+-------------------------------------------------------------+
| json_extract('{"a":[{"b":1.0},{"b":2.0}]}', '$.a[*].b')   |
+-------------------------------------------------------------+
| [1.0,2.0]                                                   |
+-------------------------------------------------------------+

os> source=people | eval `invalid_json` = json_extract('{"invalid": "json"')
fetched rows / total rows = 1/1
+----------------+
| invalid_json   |
+----------------+
| null           |
+----------------+
```

JSON_KEYS

用法：以数组`json_keys(jsonStr)`形式返回最外层的 JSON 对象的所有键。

参数类型：字符串。有效 JSON 对象格式的字符串表达式。

返回类型：数组 [字符串]。NULL对于任何其他有效的 JSON 字符串、空字符串或无效的 JSON，该函数都会返回。

示例：

```
os> source=people | eval `keys` = json_keys('{"f1":"abc","f2":{"f3":"a","f4":"b"}}')
fetched rows / total rows = 1/1
+------------+
| keus       |
+------------+
| [f1, f2]   |
+------------+

os> source=people | eval `keys` = json_keys('[1,2,3,{"f1":1,"f2":[5,6]},4]')
fetched rows / total rows = 1/1
+--------+
| keys   |
+--------+
| null   |
+--------+
```

JSON_VALID

用法：`json_valid(jsonStr)`评估 JSON 字符串是否使用有效的 JSON 语法并返回 TRUE 或 FALSE。

参数类型：字符串

返回类型：布尔值

示例：

```
os> source=people | eval `valid_json` = json_valid('[1,2,3,4]'), `invalid_json` =
 json_valid('{"invalid": "json"') | feilds `valid_json`, `invalid_json`
fetched rows / total rows = 1/1
+--------------+----------------+
| valid_json   | invalid_json   |
+--------------+----------------+
| True         | False          |
+--------------+----------------+

os> source=accounts | where json_valid('[1,2,3,4]') and isnull(email) | fields
 account_number, email
fetched rows / total rows = 1/1
+------------------+---------+
| account_number   | email   |
```

```
|------------------+---------|
| 13               | null    |
+------------------+---------+
```

PPL Lambda 函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

EXISTS

用法：`exists(array, lambda)`评估 Lambda 谓词是否适用于数组中的一个或多个元素。

参数类型：数组、LAMBDA

返回类型：布尔值。`TRUE`如果数组中至少有一个元素满足 Lambda 谓词，则返回，否则返回。`FALSE`

示例：

```
 os> source=people | eval array = json_array(1, -1, 2), result = exists(array, x -> x >
 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| true      |
+-----------+

os> source=people | eval array = json_array(-1, -3, -2), result = exists(array, x -> x
 > 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| false     |
+-----------+
```

FILTER

用法：使用给定的 Lambda 函数`filter(array, lambda)`筛选输入数组。

参数类型：数组、LAMBDA

返回类型：数组。一个包含输入数组中满足 lambda 谓词的所有元素的数组。

示例：

```
 os> source=people | eval array = json_array(1, -1, 2), result = filter(array, x -> x >
 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| [1, 2]    |
+-----------+

os> source=people | eval array = json_array(-1, -3, -2), result = filter(array, x -> x
 > 0) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| []        |
+-----------+
```

TRANSFORM

用法：使用 Lambda `transform(array, lambda)` 变换函数转换数组中的元素。如果使用二进制 Lambda 函数，则第二个参数表示元素的索引。这与函数式编程`map`中的 a 类似。

参数类型：数组、LAMBDA

返回类型：数组。一个数组，其中包含将 lambda 转换函数应用于输入数组中每个元素的结果。

示例：

```
os> source=people | eval array = json_array(1, 2, 3), result = transform(array, x -> x
 + 1) | fields result
fetched rows / total rows = 1/1
+--------------+
| result       |
+--------------+
| [2, 3, 4]    |
```

```
+--------------+

os> source=people | eval array = json_array(1, 2, 3), result = transform(array, (x, i)
 -> x + i) | fields result
fetched rows / total rows = 1/1
+--------------+
| result       |
+--------------+
| [1, 3, 5]    |
+--------------+
```

REDUCE

用法：`reduce(array, start, merge_lambda, finish_lambda)`通过应用 lambda 函数将数组简化为单个值。该函数将 merge_lambda 应用于起始值和所有数组元素，然后将应用于`finish_lambda`结果。

参数类型：数组、任意、LAMBDA、LAMBDA

返回类型：任意。将 Lambda 函数应用于起始值和输入数组的最终结果。

示例：

```
 os> source=people | eval array = json_array(1, 2, 3), result = reduce(array, 0, (acc,
 x) -> acc + x) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| 6         |
+-----------+

os> source=people | eval array = json_array(1, 2, 3), result = reduce(array, 10, (acc,
 x) -> acc + x) | fields result
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| 16        |
+-----------+

os> source=people | eval array = json_array(1, 2, 3), result = reduce(array, 0, (acc,
 x) -> acc + x, acc -> acc * 10) | fields result
```

```
fetched rows / total rows = 1/1
+-----------+
| result    |
+-----------+
| 60        |
+-----------+
```

PPL 数学函数

> **Note**
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

ABS

用法：`ABS(x)` 计算 x 的绝对值。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：INTEGER/LONG/FLOAT/DOUBLE

示例：

```
os> source=people | eval `ABS(-1)` = ABS(-1) | fields `ABS(-1)`
fetched rows / total rows = 1/1
+-----------+
| ABS(-1)   |
|-----------|
| 1         |
+-----------+
```

ACOS

用法：`ACOS(x)`计算 x 的反余弦值。`NULL`如果 x 不在 -1 到 1 的范围内，则返回。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `ACOS(0)` = ACOS(0) | fields `ACOS(0)`
fetched rows / total rows = 1/1
+--------------------+
| ACOS(0)            |
|--------------------|
| 1.5707963267948966 |
+--------------------+
```

ASIN

用法：`asin(x)`计算 x 的反正弦值。`NULL`如果 x 不在 -1 到 1 的范围内，则返回。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `ASIN(0)` = ASIN(0) | fields `ASIN(0)`
fetched rows / total rows = 1/1
+-----------+
| ASIN(0)   |
|-----------|
| 0.0       |
+-----------+
```

ATAN

用法：`ATAN(x)`计算 x 的反正切。`atan(y, x)`计算 y/x 的反正切，唯一的不同是两个参数的符号决定结果的象限。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `ATAN(2)` = ATAN(2), `ATAN(2, 3)` = ATAN(2, 3) | fields
 `ATAN(2)`, `ATAN(2, 3)`
fetched rows / total rows = 1/1
+--------------------+--------------------+
```

```
| ATAN(2)            | ATAN(2, 3)         |
|--------------------+--------------------|
| 1.1071487177940904 | 0.5880026035475675 |
+--------------------+--------------------+
```

ATAN2

用法：ATAN2(y, x)计算 y/x 的反正切，唯一的不同是两个参数的符号决定结果的象限。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `ATAN2(2, 3)` = ATAN2(2, 3) | fields `ATAN2(2, 3)`
fetched rows / total rows = 1/1
+--------------------+
| ATAN2(2, 3)        |
|--------------------|
| 0.5880026035475675 |
+--------------------+
```

CBRT

用法：CBRT计算数字的立方根。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：双精度：

INTEGER/LONG/FLOAT/DOUBLE-> 双打

示例：

```
opensearchsql> source=location | eval `CBRT(8)` = CBRT(8), `CBRT(9.261)` = CBRT(9.261),
 `CBRT(-27)` = CBRT(-27) | fields `CBRT(8)`, `CBRT(9.261)`, `CBRT(-27)`;
fetched rows / total rows = 2/2
+-----------+---------------+-------------+
| CBRT(8)   | CBRT(9.261)   | CBRT(-27)   |
|-----------+---------------+-------------|
```

```
| 2.0       | 2.1           | -3.0        |
| 2.0       | 2.1           | -3.0        |
+-----------+---------------+-------------+
```

CEIL

用法：CEILING函数的别名。 CEILING(T)取值 T 的上限

限制：CEILING仅当 IEEE 754 双精度类型在存储时显示十进制时，才能按预期工作。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：长整型

示例：

```
os> source=people | eval `CEILING(0)` = CEILING(0), `CEILING(50.00005)` =
 CEILING(50.00005), `CEILING(-50.00005)` = CEILING(-50.00005) | fields `CEILING(0)`,
 `CEILING(50.00005)`, `CEILING(-50.00005)`
fetched rows / total rows = 1/1
+--------------+---------------------+----------------------+
| CEILING(0)   | CEILING(50.00005)   | CEILING(-50.00005)   |
|--------------+---------------------+----------------------|
| 0            | 51                  | -50                  |
+--------------+---------------------+----------------------+

os> source=people | eval `CEILING(3147483647.12345)` = CEILING(3147483647.12345),
 `CEILING(113147483647.12345)` = CEILING(113147483647.12345),
 `CEILING(3147483647.00001)` = CEILING(3147483647.00001) | fields
 `CEILING(3147483647.12345)`, `CEILING(113147483647.12345)`,
 `CEILING(3147483647.00001)`
fetched rows / total rows = 1/1
+-----------------------------+------------------------------
+-----------------------------+
| CEILING(3147483647.12345)   | CEILING(113147483647.12345)   |
 CEILING(3147483647.00001)   |
|-----------------------------+------------------------------
+-----------------------------|
| 3147483648                  | 113147483648                  | 3147483648
     |
+-----------------------------+------------------------------
+-----------------------------+
```

CONV

用法：CONV(x, a, b)将数字 x 从基数转换为 b 基数。

参数类型：x：字符串，a：整数，b：整数

返回类型：字符串

示例：

```
os> source=people | eval `CONV('12', 10, 16)` = CONV('12', 10, 16), `CONV('2C', 16,
 10)` = CONV('2C', 16, 10), `CONV(12, 10, 2)` = CONV(12, 10, 2), `CONV(1111, 2, 10)` =
 CONV(1111, 2, 10) | fields `CONV('12', 10, 16)`, `CONV('2C', 16, 10)`, `CONV(12, 10,
 2)`, `CONV(1111, 2, 10)`
fetched rows / total rows = 1/1
+----------------------+----------------------+-------------------
+----------------------+
| CONV('12', 10, 16)   | CONV('2C', 16, 10)   | CONV(12, 10, 2)   | CONV(1111, 2, 10)
 |
|----------------------+----------------------+-------------------
+----------------------|
| c                    | 44                   | 1100              | 15
 |
+----------------------+----------------------+-------------------
+----------------------+
```

COS

用法：COS(x)计算 x 的余弦值，其中 x 以弧度给出。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `COS(0)` = COS(0) | fields `COS(0)`
fetched rows / total rows = 1/1
+----------+
| COS(0)   |
|----------|
| 1.0      |
+----------+
```

COT

用法：`COT(x)`计算 x 的余切。如果 x 等于 0，则返回 out-of-range错误。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `COT(1)` = COT(1) | fields `COT(1)`
fetched rows / total rows = 1/1
+--------------------+
| COT(1)             |
|--------------------|
| 0.6420926159343306 |
+--------------------+
```

CRC32

用法：`CRC32`计算循环冗余校验值并返回 32 位无符号值。

参数类型：字符串

返回类型：长整型

示例：

```
os> source=people | eval `CRC32('MySQL')` = CRC32('MySQL') | fields `CRC32('MySQL')`
fetched rows / total rows = 1/1
+------------------+
| CRC32('MySQL')   |
|------------------|
| 3259397556       |
+------------------+
```

DEGREES

用法：`DEGREES(x)`将 x 从弧度转换为度。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `DEGREES(1.57)` = DEGREES(1.57) | fields `DEGREES(1.57)`
fetched rows / total rows  = 1/1
+-------------------+
| DEGREES(1.57)     |
|-------------------|
| 89.95437383553924 |
+-------------------+
```

E

用法：E()返回欧拉数字。

返回类型：DOUBLE

示例：

```
os> source=people | eval `E()` = E() | fields `E()`
fetched rows / total rows = 1/1
+-------------------+
| E()               |
|-------------------|
| 2.718281828459045 |
+-------------------+
```

EXP

用法EXP(x)：将 e 提高到 x 的次方。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `EXP(2)` = EXP(2) | fields `EXP(2)`
fetched rows / total rows = 1/1
+------------------+
| EXP(2)           |
|------------------|
| 7.38905609893065 |
```

```
+------------------+
```

FLOOR

用法：FLOOR(T)取值 T 的下限。

限制：FLOOR仅当 IEEE 754 双精度类型在存储时显示十进制时，才能按预期工作。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：长整型

示例：

```
os> source=people | eval `FLOOR(0)` = FLOOR(0), `FLOOR(50.00005)` = FLOOR(50.00005),
 `FLOOR(-50.00005)` = FLOOR(-50.00005) | fields `FLOOR(0)`, `FLOOR(50.00005)`,
 `FLOOR(-50.00005)`
fetched rows / total rows = 1/1
+------------+-------------------+--------------------+
| FLOOR(0)   | FLOOR(50.00005)   | FLOOR(-50.00005)   |
|------------+-------------------+--------------------|
| 0          | 50                | -51                |
+------------+-------------------+--------------------+

os> source=people | eval `FLOOR(3147483647.12345)` = FLOOR(3147483647.12345),
 `FLOOR(113147483647.12345)` = FLOOR(113147483647.12345), `FLOOR(3147483647.00001)`
 = FLOOR(3147483647.00001) | fields `FLOOR(3147483647.12345)`,
 `FLOOR(113147483647.12345)`, `FLOOR(3147483647.00001)`
fetched rows / total rows = 1/1
+---------------------------+----------------------------+---------------------------+
| FLOOR(3147483647.12345)   | FLOOR(113147483647.12345)  | FLOOR(3147483647.00001)   |
|---------------------------+----------------------------+---------------------------|
| 3147483647                | 113147483647               | 3147483647                |
+---------------------------+----------------------------+---------------------------+

os> source=people | eval `FLOOR(282474973688888.022)` = FLOOR(282474973688888.022),
 `FLOOR(9223372036854775807.022)` = FLOOR(9223372036854775807.022),
 `FLOOR(9223372036854775807.0000001)` = FLOOR(9223372036854775807.0000001)
 | fields `FLOOR(282474973688888.022)`, `FLOOR(9223372036854775807.022)`,
 `FLOOR(9223372036854775807.0000001)`
fetched rows / total rows = 1/1
+------------------------------+-----------------------------------
+--------------------------------------+
```

```
| FLOOR(282474973688888.022)   | FLOOR(9223372036854775807.022)   |
 FLOOR(9223372036854775807.0000001)   |
|------------------------------+----------------------------------
+--------------------------------------|
| 282474973688888              | 9223372036854775807              | 9223372036854775807
                  |
+------------------------------+----------------------------------
+--------------------------------------+
```

LN

用法：LN(x)返回 x 的自然对数。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `LN(2)` = LN(2) | fields `LN(2)`
fetched rows / total rows = 1/1
+--------------------+
| LN(2)              |
|--------------------|
| 0.6931471805599453 |
+--------------------+
```

LOG

用法：LOG(x)返回 x 的自然对数，即 x 的以 e 为底的对数。log (B, x) 等同于 log (x) /log (B)。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `LOG(2)` = LOG(2), `LOG(2, 8)` = LOG(2, 8) | fields `LOG(2)`,
 `LOG(2, 8)`
fetched rows / total rows = 1/1
+--------------------+-------------+
| LOG(2)             | LOG(2, 8)   |
|--------------------+-------------|
```

```
| 0.6931471805599453 | 3.0         |
+--------------------+-------------+
```

LOG2

用法：等同`LOG2(x)`于`log(x)/log(2)`。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `LOG2(8)` = LOG2(8) | fields `LOG2(8)`
fetched rows / total rows = 1/1
+-----------+
| LOG2(8)   |
|-----------|
| 3.0       |
+-----------+
```

LOG10

用法：等同`LOG10(x)`于`log(x)/log(10)`。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `LOG10(100)` = LOG10(100) | fields `LOG10(100)`
fetched rows / total rows = 1/1
+--------------+
| LOG10(100)   |
|--------------|
| 2.0          |
+--------------+
```

MOD

用法：`MOD(n, m)`计算数字 n 除以 m 的余数。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：如果 m 为非零值，则在 n 和 m 的类型之间使用更宽的类型。如果 m 等于 0，则返回 NULL。

示例：

```
os> source=people | eval `MOD(3, 2)` = MOD(3, 2), `MOD(3.1, 2)` = MOD(3.1, 2) | fields
 `MOD(3, 2)`, `MOD(3.1, 2)`
fetched rows / total rows = 1/1
+-------------+---------------+
| MOD(3, 2)   | MOD(3.1, 2)   |
|-------------+---------------|
| 1           | 1.1           |
+-------------+---------------+
```

PI

用法：`PI()` 返回常量 pi。

返回类型：DOUBLE

示例：

```
os> source=people | eval `PI()` = PI() | fields `PI()`
fetched rows / total rows = 1/1
+-------------------+
| PI()              |
|-------------------|
| 3.141592653589793 |
+-------------------+
```

POW

用法：`POW(x, y)`计算 x 的值提高到 y 的次方。错误的输入会返回NULL结果。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

同义词：`POWER(_, _)`

示例：

```
os> source=people | eval `POW(3, 2)` = POW(3, 2), `POW(-3, 2)` = POW(-3, 2), `POW(3,
 -2)` = POW(3, -2) | fields `POW(3, 2)`, `POW(-3, 2)`, `POW(3, -2)`
fetched rows / total rows = 1/1
+-------------+--------------+--------------------+
| POW(3, 2)   | POW(-3, 2)   | POW(3, -2)         |
|-------------+--------------+--------------------|
| 9.0         | 9.0          | 0.1111111111111111 |
+-------------+--------------+--------------------+
```

POWER

用法：POWER(x, y)计算 x 的值提高到 y 的次方。错误的输入会返回NULL结果。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

同义词：POW(_, _)

示例：

```
os> source=people | eval `POWER(3, 2)` = POWER(3, 2), `POWER(-3, 2)` = POWER(-3, 2),
 `POWER(3, -2)` = POWER(3, -2) | fields `POWER(3, 2)`, `POWER(-3, 2)`, `POWER(3, -2)`
fetched rows / total rows = 1/1
+---------------+----------------+--------------------+
| POWER(3, 2)   | POWER(-3, 2)   | POWER(3, -2)       |
|---------------+----------------+--------------------|
| 9.0           | 9.0            | 0.1111111111111111 |
+---------------+----------------+--------------------+
```

RADIANS

用法：RADIANS(x)将 x 从度数转换为弧度。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `RADIANS(90)` = RADIANS(90) | fields `RADIANS(90)`
fetched rows / total rows  = 1/1
+--------------------+
```

```
| RADIANS(90)        |
|--------------------|
| 1.5707963267948966 |
+--------------------+
```

RAND

用法：RAND( )/RAND(N)返回 0 <= 值 < 1.0 范围内的随机浮点值。如果指定整数 N，则该函数将在执行前初始化种子。这种行为的一个含义是，使用相同的参数 N，每次都rand(N)返回相同的值，从而生成可重复的列值序列。

参数类型：整数

返回类型：FLOAT

示例：

```
os> source=people | eval `RAND(3)` = RAND(3) | fields `RAND(3)`
fetched rows / total rows = 1/1
+------------+
| RAND(3)    |
|------------|
| 0.73105735 |
+------------+
```

ROUND

用法：将参数 x ROUND(x, d) 四舍五入到 d 个小数位。如果未指定 d，则默认为 0。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型地图：

- （整数/长 [，整数]）-> 长
- （浮点数/双精度 [，整数]）-> 长

示例：

```
os> source=people | eval `ROUND(12.34)` = ROUND(12.34), `ROUND(12.34, 1)` =
 ROUND(12.34, 1), `ROUND(12.34, -1)` = ROUND(12.34, -1), `ROUND(12, 1)` = ROUND(12, 1)
 | fields `ROUND(12.34)`, `ROUND(12.34, 1)`, `ROUND(12.34, -1)`, `ROUND(12, 1)`
```

```
fetched rows / total rows = 1/1
+----------------+-------------------+--------------------+----------------+
| ROUND(12.34)   | ROUND(12.34, 1)   | ROUND(12.34, -1)   | ROUND(12, 1)   |
|----------------+-------------------+--------------------+----------------|
| 12.0           | 12.3              | 10.0               | 12             |
+----------------+-------------------+--------------------+----------------+
```

SIGN

用法：将参数的符号SIGN返回为 -1、0 或 1，具体取决于数字是负数、零还是正数。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：整数

示例：

```
os> source=people | eval `SIGN(1)` = SIGN(1), `SIGN(0)` = SIGN(0), `SIGN(-1.1)` =
 SIGN(-1.1) | fields `SIGN(1)`, `SIGN(0)`, `SIGN(-1.1)`
fetched rows / total rows = 1/1
+-----------+-----------+--------------+
| SIGN(1)   | SIGN(0)   | SIGN(-1.1)   |
|-----------+-----------+--------------|
| 1         | 0         | -1           |
+-----------+-----------+--------------+
```

SIN

用法：sin(x)计算 x 的正弦值，其中 x 以弧度给出。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型：DOUBLE

示例：

```
os> source=people | eval `SIN(0)` = SIN(0) | fields `SIN(0)`
fetched rows / total rows = 1/1
+----------+
| SIN(0)   |
|----------|
| 0.0      |
+----------+
```

SQRT

用法：SQRT计算非负数的平方根。

参数类型：INTEGER/LONG/FLOAT/DOUBLE

返回类型地图：

- （非负数）INTEGER/LONG/FLOAT/DOUBLE-> 双精度
- （负数）INTEGER/LONG/FLOAT/DOUBLE-> 空

示例：

```
os> source=people | eval `SQRT(4)` = SQRT(4), `SQRT(4.41)` = SQRT(4.41) | fields
 `SQRT(4)`, `SQRT(4.41)`
fetched rows / total rows = 1/1
+-----------+--------------+
| SQRT(4)   | SQRT(4.41)   |
|-----------+--------------|
| 2.0       | 2.1          |
+-----------+--------------+
```

PPL 字符串函数

> **Note**
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

CONCAT

用法：`CONCAT(str1, str2, ...., str_9)`将最多 9 个字符串相加。

参数类型：

- 字符串、字符串、...、字符串
- 返回类型：字符串

示例：

```
os> source=people | eval `CONCAT('hello', 'world')` = CONCAT('hello', 'world'),
 `CONCAT('hello ', 'whole ', 'world', '!')` = CONCAT('hello ', 'whole ', 'world', '!')
 | fields `CONCAT('hello', 'world')`, `CONCAT('hello ', 'whole ', 'world', '!')`
fetched rows / total rows = 1/1
+----------------------------+-------------------------------------------+
| CONCAT('hello', 'world')   | CONCAT('hello ', 'whole ', 'world', '!')   |
|----------------------------+-------------------------------------------|
| helloworld                 | hello whole world!                         |
+----------------------------+-------------------------------------------+
```

CONCAT_WS

用法：使用指定的分隔符`CONCAT_WS(sep, str1, str2)`连接两个或多个字符串。

参数类型：

- 字符串、字符串、...、字符串
- 返回类型：字符串

示例：

```
os> source=people | eval `CONCAT_WS(',', 'hello', 'world')` = CONCAT_WS(',', 'hello',
 'world') | fields `CONCAT_WS(',', 'hello', 'world')`
fetched rows / total rows = 1/1
+------------------------------------+
| CONCAT_WS(',', 'hello', 'world')   |
|------------------------------------|
| hello,world                        |
+------------------------------------+
```

LENGTH

用法：`length(str)`返回以字节为单位的输入字符串的长度。

参数类型：

- string
- 返回类型：整数

示例：

```
os> source=people | eval `LENGTH('helloworld')` = LENGTH('helloworld') | fields
 `LENGTH('helloworld')`
fetched rows / total rows = 1/1
+------------------------+
| LENGTH('helloworld')   |
|------------------------|
| 10                     |
+------------------------+
```

LOWER

用法：`lower(string)`将输入字符串转换为小写。

参数类型：

- string
- 返回类型：字符串

示例：

```
os> source=people | eval `LOWER('helloworld')` = LOWER('helloworld'),
 `LOWER('HELLOWORLD')` = LOWER('HELLOWORLD') | fields `LOWER('helloworld')`,
 `LOWER('HELLOWORLD')`
fetched rows / total rows = 1/1
+------------------------+------------------------+
| LOWER('helloworld')    | LOWER('HELLOWORLD')    |
|------------------------+------------------------|
| helloworld             | helloworld             |
+------------------------+------------------------+
```

LTRIM

用法：从输入字符串中`ltrim(str)`删除前导空格字符。

参数类型：

- string
- 返回类型：字符串

示例：

```
os> source=people | eval `LTRIM('   hello')` = LTRIM('   hello'), `LTRIM('hello   ')` =
 LTRIM('hello   ') | fields `LTRIM('   hello')`, `LTRIM('hello   ')`
fetched rows / total rows = 1/1
+---------------------+---------------------+
| LTRIM('   hello')   | LTRIM('hello   ')   |
|---------------------+---------------------|
| hello               | hello               |
+---------------------+---------------------+
```

POSITION

用法：POSITION(substr IN str)返回字符串中子字符串首次出现的位置。如果子字符串不在字符串中，则返回 0。如果有任何参数为空，则返回 NULL。

参数类型：

- 字符串，字符串
- 返回类型整数

示例：

```
os> source=people | eval `POSITION('world' IN 'helloworld')` = POSITION('world'
 IN 'helloworld'), `POSITION('invalid' IN 'helloworld')`= POSITION('invalid' IN
 'helloworld')  | fields `POSITION('world' IN 'helloworld')`, `POSITION('invalid' IN
 'helloworld')`
fetched rows / total rows = 1/1
+-------------------------------------+---------------------------------------+
| POSITION('world' IN 'helloworld')   | POSITION('invalid' IN 'helloworld')   |
|-------------------------------------+---------------------------------------|
| 6                                   | 0                                     |
+-------------------------------------+---------------------------------------+
```

REVERSE

用法：REVERSE(str)返回输入字符串的反向字符串。

参数类型：

- string
- 返回类型：字符串

示例：

```
os> source=people | eval `REVERSE('abcde')` = REVERSE('abcde') | fields
 `REVERSE('abcde')`
fetched rows / total rows = 1/1
+--------------------+
| REVERSE('abcde')   |
|--------------------|
| edcba              |
+--------------------+
```

RIGHT

用法：`right(str, len)`返回输入字符串中最右边的字符。如果子字符串不在字符串中，则返回 0。如果有任何参数为空，则返回 NULL。

参数类型：

- 字符串，整数
- 返回类型：字符串

示例：

```
os> source=people | eval `RIGHT('helloworld', 5)` = RIGHT('helloworld', 5),
 `RIGHT('HELLOWORLD', 0)` = RIGHT('HELLOWORLD', 0) | fields `RIGHT('helloworld', 5)`,
 `RIGHT('HELLOWORLD', 0)`
fetched rows / total rows = 1/1
+--------------------------+--------------------------+
| RIGHT('helloworld', 5)   | RIGHT('HELLOWORLD', 0)   |
|--------------------------+--------------------------|
| world                    |                          |
+--------------------------+--------------------------+
```

RTRIM

用法：`rtrim(str)`从输入字符串中删除尾随空格字符。

参数类型：

- string
- 返回类型：字符串

示例：

```
os> source=people | eval `RTRIM('   hello')` = RTRIM('   hello'), `RTRIM('hello   ')` =
 RTRIM('hello   ') | fields `RTRIM('   hello')`, `RTRIM('hello   ')`
fetched rows / total rows = 1/1
+---------------------+---------------------+
| RTRIM('   hello')   | RTRIM('hello   ')   |
|---------------------+---------------------|
|    hello            | hello               |
+---------------------+---------------------+
```

SUBSTRING

用法：`substring(str, start)`或`substring(str, start, length)`返回输入字符串的子字符串。如果未指定长度，它将从起始位置返回整个字符串。

参数类型：

- 字符串、整数、整数
- 返回类型：字符串

示例：

```
os> source=people | eval `SUBSTRING('helloworld', 5)` = SUBSTRING('helloworld',
 5), `SUBSTRING('helloworld', 5, 3)` = SUBSTRING('helloworld', 5, 3) | fields
 `SUBSTRING('helloworld', 5)`, `SUBSTRING('helloworld', 5, 3)`
fetched rows / total rows = 1/1
+------------------------------+---------------------------------+
| SUBSTRING('helloworld', 5)   | SUBSTRING('helloworld', 5, 3)   |
|------------------------------+---------------------------------|
| oworld                       | owo                             |
+------------------------------+---------------------------------+
```

TRIM

用法：`trim(string)`从输入字符串中删除前导和尾随空格。

参数类型：

- string

- 返回类型：字符串

示例：

```
os> source=people | eval `TRIM('   hello')` = TRIM('   hello'), `TRIM('hello   ')` =
 TRIM('hello   ') | fields `TRIM('   hello')`, `TRIM('hello   ')`
fetched rows / total rows = 1/1
+--------------------+--------------------+
| TRIM('   hello')   | TRIM('hello   ')   |
|--------------------+--------------------|
| hello              | hello              |
+--------------------+--------------------+
```

UPPER

用法：`upper(string)`将输入字符串转换为大写。

参数类型：

- string
- 返回类型：字符串

示例：

```
os> source=people | eval `UPPER('helloworld')` = UPPER('helloworld'),
 `UPPER('HELLOWORLD')` = UPPER('HELLOWORLD') | fields `UPPER('helloworld')`,
 `UPPER('HELLOWORLD')`
fetched rows / total rows = 1/1
+-----------------------+-----------------------+
| UPPER('helloworld')   | UPPER('HELLOWORLD')   |
|-----------------------+-----------------------|
| HELLOWORLD            | HELLOWORLD            |
+-----------------------+-----------------------+
```

PPL 类型转换函数

> **Note**
>
> 要查看哪些 Amazon 数据源集成支持此 PPL 功能，请参阅。the section called “函数”

TRIM

用法`cast(expr as dateType)`：`expr`将转换为，`dataType`并返回的值`dataType`。

以下转换规则适用：

类型转换规则

源/目标	string	NUMBER	BOOLEAN	TIMESTAMP	DATE	TIME
string		注意事项1	注意事项1	TIMESTAMP ()	DATE()	TIME()
NUMBER	注意事项1		v! =0	不适用	不适用	不适用
BOOLEAN	注意事项1	v？ 1:0		不适用	不适用	不适用
TIMESTAMP	注意事项1	不适用	不适用		DATE()	TIME()
DATE	注意事项1	不适用	不适用	不适用		不适用
TIME	注意事项1	不适用	不适用	不适用	不适用	

转换为字符串示例：

```
os> source=people | eval `cbool` = CAST(true as string), `cint` = CAST(1 as string),
 `cdate` = CAST(CAST('2012-08-07' as date) as string) | fields `cbool`, `cint`, `cdate`
fetched rows / total rows = 1/1
+---------+--------+------------+
| cbool   | cint   | cdate      |
|---------+--------+------------|
| true    | 1      | 2012-08-07 |
+---------+--------+------------+
```

强制转换成数字示例：

```
os> source=people | eval `cbool` = CAST(true as int), `cstring` = CAST('1' as int) |
 fields `cbool`, `cstring`
fetched rows / total rows = 1/1
+---------+-----------+
| cbool   | cstring   |
```

```
|---------+-----------|
| 1       | 1         |
+---------+-----------+
```

迄今为止的演员阵容示例：

```
os> source=people | eval `cdate` = CAST('2012-08-07' as date), `ctime` =
 CAST('01:01:01' as time), `ctimestamp` = CAST('2012-08-07 01:01:01' as timestamp) |
 fields `cdate`, `ctime`, `ctimestamp`
fetched rows / total rows = 1/1
+------------+----------+---------------------+
| cdate      | ctime    | ctimestamp          |
|------------+----------+---------------------|
| 2012-08-07 | 01:01:01 | 2012-08-07 01:01:01 |
+------------+----------+---------------------+
```

连锁演员阵容示例：

```
os> source=people | eval `cbool` = CAST(CAST(true as string) as boolean) | fields
 `cbool`
fetched rows / total rows = 1/1
+---------+
| cbool   |
|---------|
| True    |
+---------+
```

监控 Amazon OpenSearch 服务域名

监控是维护 Amazon S OpenSearch ervice 和其他 Amazon 解决方案的可靠性、可用性和性能的重要组成部分。 Amazon 提供以下工具来监控您的 OpenSearch 服务资源、报告问题并在适当时自动采取措施：

Amazon CloudWatch

Amazon 会实时 CloudWatch 监控您的 OpenSearch 服务资源。您可以收集和跟踪指标，创建自定义的控制面板，以及设置警报以在指标达到特定阈值时通知您或采取措施。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

Amazon CloudWatch 日志

Amazon L CloudWatch ogs 允许您监控、存储和访问您的 OpenSearch 日志文件。 CloudWatch 日志会监控日志文件中的信息，并且可以在达到特定阈值时通知您。有关更多信息，请参阅 Amazon CloudWatch 日志用户指南。

Amazon EventBridge

Amazon EventBridge 提供近乎实时的系统事件流，用于描述您的 OpenSearch 服务域中的变化。您可以创建规则来监视某些事件，并在这些事件发生时在其他 Amazon 服务中触发自动操作。有关更多信息，请参阅 Amazon EventBridge 用户指南。

Amazon CloudTrail

Amazon CloudTrail 捕获对 OpenSearch 服务的配置 API 调用作为事件。然后它将这些事件传送到您指定的 Amazon S3 存储桶。通过使用此信息，您可以标识哪些用户和账户发出请求、发出请求的源 IP 地址以及请求的发生时间。有关更多信息，请参阅 Amazon 用户指南 CloudTrail 。

主题

- 使用 Amazon 监控 OpenSearch 集群指标 CloudWatch
- 使用 Amazon OpenSearch 日志监控 CloudWatch 日志
- 在 Amazon OpenSearch 服务中监控审计日志
- 使用 Amazon 监控 OpenSearch 服务事件 EventBridge
- 使用监控亚马逊 OpenSearch 服务 API 调用 Amazon CloudTrail

使用 Amazon 监控 OpenSearch 集群指标 CloudWatch

Amazon S OpenSearch ervice 会将数据从您的域发布到 Amazon CloudWatch。 CloudWatch 允许您按一组有序的时间序列数据（称为指标）来检索关于这些数据点的统计数据。 OpenSearch 服务每隔 60 秒向发送大部分指标。 CloudWatch 如果您使用通用型 EBS 卷或磁性 EBS 卷，则 EBS 卷指标将仅每五分钟更新一次。所有累积性指标（例如 `ThreadpoolWriteRejected`、`ThreadpoolSearchRejected`）都在内存中，并且会丢失状态。在节点丢弃、节点反弹、节点更换和蓝绿部署期间，指标将会重置。有关 Amazon 的更多信息 CloudWatch，请参阅 Amazon CloudWatch 用户指南。

S OpenSearch ervice 控制台将根据来自的原始数据显示一系列图表 CloudWatch。根据您的需求，您可能更愿意查看集群数据， CloudWatch 而不是控制台中的图表。该服务会将指标存档两周，然后再丢弃。这些指标均是免费提供的，但 CloudWatch 仍会收取创建控制面板和警报的费用。有关更多信息，请参阅 Amazon CloudWatch 定价。

OpenSearch 服务将以下指标发布到 CloudWatch：

- the section called “集群指标”
- the section called “专用主节点指标”
- the section called “EBS 卷指标”
- the section called “实例指标”
- the section called “UltraWarm 指标”
- the section called “专用协调器节点指标”
- the section called “冷存储指标”
- the section called “提醒指标”
- the section called “异常检测指标”
- the section called “异步搜索指标”
- the section called “SQL 指标”
- the section called “k-NN 指标”
- the section called “跨集群搜索指标”
- the section called “跨集群复制指标”
- the section called “学习排名指标”
- the section called “管道处理语言指标”

在中查看指标 CloudWatch

CloudWatch 指标的分组首先依据服务命名空间，然后依据每个命名空间内的各种维度组合。

使用 CloudWatch 控制台查看指标

1. 打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 在左侧导航窗格中，找到 Metrics（指标），然后选择 All metrics（所有指标）。选择 ES/OpenSearchService 命名空间。
3. 选择维度以查看相应指标。单个节点的指标位于 `ClientId, DomainName, NodeId` 维度中。集群指标位于 `Per-Domain, Per-Client Metrics` 维度中。某些节点指标在集群级别进行聚合，因此包含在这两个维度中。分区指标位于 `ClientId, DomainName, NodeId, ShardRole` 维度中。

要查看指标列表，请使用 Amazon CLI

运行以下命令：

```
aws cloudwatch list-metrics --namespace "AWS/ES"
```

解释S OpenSearch ervice 中的运行状况图表

要查看 OpenSearch 服务中的指标，请使用集群运行状况和实例运行状况选项卡。实例运行状况选项卡使用框图让您 at-a-glance了解每个 OpenSearch节点的运行状况：

- 每个彩色框显示指定时间段内节点的值范围。
- 蓝框表示与其他节点一致的值。红框表示异常值。
- 每个框中的白线显示节点的当前值。
- 每个框两侧的“细线”显示该时间段内所有节点的最小值和最大值。

如果对您的域进行配置更改，则 Cluster health (集群运行状况) 和 Instance health (实例运行状况) 选项卡中各个实例的列表的大小通常会在短时间内增长一倍，然后再恢复为正确数量。有关此行为的说明，请参阅the section called “配置更改”。

集群指标

Amazon S OpenSearch ervice 为集群提供以下指标。

指标	描述
`ClusterStatus.green`	值为 1 指示将所有索引分片分配给集群中的节点。 相关统计数据：Maximum
`ClusterStatus.yellow`	值为 1 指示将所有索引的主要分片分配给集群中的节点，但是至少有一个索引的分片副本不是如此。有关更多信息，请参阅 the section called “黄色集群状态”。 相关统计数据：Maximum
`ClusterStatus.red`	值为 1 指示至少一个索引的主分片和副本分片未分配给集群中的节点。有关更多信息，请参阅 the section called “红色集群状态”。 相关统计数据：Maximum
`Shards.active`	活动主分区和副本分区的总数。 相关统计数据：最大值、总计
`Shards.unassigned`	未分配给集群中节点的分区数。 相关统计数据：最大值、总计
`Shards.delayedUnassigned`	其节点分配因超时设置已延迟的分区数。 相关统计数据：最大值、总计
`Shards.activePrimary`	活动主分区数。 相关统计数据：最大值、总计

指标	描述
Shards.initializing	正在初始化的分区数。 相关统计数据：总计
Shards.relocating	正在重新定位的分区数。 相关统计数据：总计
Nodes	S OpenSearch ervice 集群中的节点数，包括专用主 UltraWarm 节点和节点。有关更多信息，请参阅 the section called “配置更改”。 相关统计数据：Maximum
SearchableDocuments	跨集群中所有数据节点的可搜索文档的总数。 相关统计数据：最小值、最大值、平均值
DeletedDocuments	跨集群的所有数据节点已标记为删除的文档总数。这些文档不会再出现在搜索结果中， OpenSearch 只会在分段合并期间将已删除的文档从磁盘中移除。此指标在提出删除请求后会增加，在分段合并后会减少。 相关统计数据：最小值、最大值、平均值
CPUUtilization	集群中数据节点的 CPU 利用率百分比。最大值显示 CPU 利用率最高的节点。平均值表示集群中的所有节点。此指标也可用于单独的节点。 相关统计数据：Maximum、Average

指标	描述
FreeStorageSpace	集群中各数据节点的可用空间。Sum 显示集群的总可用空间，但您必须保留一分钟的时间来获取准确值。Minimum 和 Maximum 分别显示具有最小和最大可用空间的节点。此指标也适用于单个节点。OpenSearch ClusterBlockException 当该指标达到0时，服务会抛出。要恢复，您必须删除索引，添加更大的实例，或向现有实例添加基于 EBS 的存储。要了解更多信息，请参阅the section called “缺少可用存储空间”。 OpenSearch 服务控制台以 GiB 为单位显示此值。Amazon CloudWatch 控制台以 MiB 为单位显示此值。 Note FreeStorageSpace 将始终低于 OpenSearch _cluster/stats 和_cat/allocation APIs 提供的值。 OpenSearch Service 在每个实例上预留一定的存储空间百分比以执行内部操作。有关更多信息，请参阅计算存储要求。 相关统计数据：Minimum、Maximum、Average、Sum
ClusterUsedSpace	集群的已使用空间总量。您必须保留一分钟的时间来获取准确值。 OpenSearch 服务控制台以 GiB 为单位显示此值。Amazon CloudWatch 控制台以 MiB 为单位显示此值。 相关统计数据：Minimum、Maximum
ClusterIndexWrites Blocked	指示您的集群是接受还是阻止传入的写入请求。值为 0 表示集群接受请求。值为 1 表示阻止请求。 一些常见的因素包括：FreeStorageSpace 过低或 JVMMemory Pressure 过高。为了缓解这一问题，可以考虑增加磁盘空间或扩展集群。 相关统计数据：Maximum

指标	描述
JVMMemoryPressure	用于集群中所有数据节点的 Java 堆的最大百分比。 OpenSearch Service 将实例的一半内存用于 Java 堆，最大堆大小为 32 GiB。您最多可以将实例的 RAM 垂直扩展至 64GiB，此时可以通过添加实例水平扩展。请参阅the section called “推荐的 CloudWatch 警报”。 相关统计数据：Maximum Note 在服务软件 R20220323 中更改了此指标的逻辑。有关更多信息，请参阅版本注释。
OldGenJVMMemoryPressure	集群中所有数据节点上用于“上一代”的 Java 堆的最大百分比。此指标也在节点级别获取。 相关统计数据：Maximum
AutomatedSnapshotFailure	集群的失败的自动快照的数量。值 1 指示在过去的 36 个小时内未为域拍摄自动快照。 相关统计数据：Minimum、Maximum
CPUCreditBalance	集群中的数据节点可用的剩余 CPU 积分。一个 CPU 信用提供一个完整 CPU 核心性能一分钟。有关更多信息，请参阅《Amazon EC2 开发者指南》中的 CPU 积分。此指标仅对 T2 实例类型有效。 相关统计数据：Minimum
OpenSearchDashboardsHealthyNodes	OpenSearch 控制面板的运行状况检查。如果最小值、最大值和平均值都等于 1，则控制面板运行正常。如果您有 10 个节点，最大值为 1，最小值为 0，平均值为 0.7，则意味着 7 个节点 (70%) 运行正常，3 个节点 (30%) 运行状况不佳。 相关统计数据：最小值、最大值、平均值

指标	描述
OpensearchDashboardsReportingFailedRequestSysErrCount	由于服务器问题或功能限制而生成失败的 OpenSearch 控制面板报告的请求数。 相关统计数据：总计
OpensearchDashboardsReportingFailedRequestUserErrCount	由于客户端问题而生成失败的 OpenSearch 控制面板报告的请求数。 相关统计数据：总计
OpensearchDashboardsReportingRequestCount	生成 OpenSearch 控制面板报告的请求总数。 相关统计数据：总计
OpensearchDashboardsReportingSuccessCount	生成 OpenSearch 控制面板报告的成功请求数。 相关统计数据：总计
KMSKeyError	值 1 表示已禁用用于加密静态数据的密 Amazon KMS 钥。要将域还原为正常操作，请重新启用该密钥。控制台仅对该加密静态数据的域显示此指标。 相关统计数据：Minimum、Maximum
KMSKeyInaccessible	值 1 表示已删除用于加密静态数据的 Amazon KMS 密钥或已撤销其对 Serv OpenSearch ice 的授权。您无法恢复处于此状态的域。但如果您具有手动快照，则可以使用它将该域的数据迁移到新域。控制台仅对该加密静态数据的域显示此指标。 相关统计数据：Minimum、Maximum

指标	描述
InvalidHostHeaderRequests	对 OpenSearch 集群的包含无效（或缺少）主机标头的 HTTP 请求数。有效请求包括作为主机标头值的域主机名。 OpenSearch Service 拒绝对没有限制性访问策略的公共访问域进行无效请求。我们建议对所有域应用限制性访问策略。 如果您看到此指标的较大值，请确认您的 OpenSearch 客户端在其请求中包含域主机名（例如，而不是其 IP 地址）。 相关统计数据：总计
OpenSearchRequests (previously ElasticsearchRequests)	向 OpenSearch 集群发出的请求数。 相关统计数据：总计
2xx, 3xx, 4xx, 5xx	导致指定的 HTTP 响应代码（2xx、3xx、4xx、5xx）的对域的请求数。 相关统计数据：总计
ThroughputThrottle	指示磁盘是否受到节流。当 ReadThroughputMicroBursting 和 WriteThroughputMicroBursting 的总吞吐量高于最大吞吐量 MaxProvisionedThroughput 时，就会发生节流。MaxProvisionedThroughput 是实例吞吐量或预调配卷吞吐量的较低值。值为 1 表示磁盘受到节流。值为 0 表示行为正常。 有关实例吞吐量的更多信息，请参阅 Amazon EBS 优化的实例。有关卷吞吐量的信息，请参阅 Amazon EBS 卷类型。 相关统计数据：Minimum、Maximum

指标	描述
IopsThrottle	指示该域每秒进行读写操作的次数（IOPS）是否已被节流。当数据节点的 IOPS 违反数据节点上 EBS 卷或 EC2 实例的最大允许限制时，就会发生节流。 有关实例 IOPS 的信息，请参阅 Amazon EBS 优化的实例。有关卷 IOPS 的信息，请参阅 Amazon EBS 卷类型。 相关统计数据：Minimum、Maximum
HighSwapUsage	值为 1 表示页面错误引起的交换可能会在特定时间段内导致底层磁盘使用量激增。 相关统计数据：Maximum

专用主节点指标

Amazon S OpenSearch ervice 为专用主节点提供了以下指标。

指标	描述
MasterCPUUtilization	专用主节点使用的 CPU 资源的最大百分比。建议在此指标达到 60% 时增加实例类型的大小。 相关统计数据：Maximum
MasterFreeStorageSpace	此指标不相关，可以被忽略。该服务不使用主节点作为数据节点。
MasterJVMMemoryPressure	用于集群中所有专用主节点的 Java 堆的最大百分比。建议在此指标达到 85% 时迁移到更大的实例类型。 相关统计数据：Maximum

指标	描述
	Note 在服务软件 R20220323 中更改了此指标的逻辑。有关更多信息，请参阅版本注释。
`MasterOldGenJVMMemoryPressure`	每个主节点上用于“上一代”的 Java 堆的最大百分比。 相关统计数据：Maximum
`MasterCPUCreditBalance`	集群中专用主节点可用的剩余 CPU 积分。一个 CPU 信用提供一个完整 CPU 核心性能一分钟。有关更多信息，请参阅《Amazon EC2 开发者指南》中的 CPU 积分。此指标仅对 T2 实例类型有效。 相关统计数据：Minimum
`MasterReachableFromNode`	`MasterNotDiscovered` 运行状况检查异常。值为 1 表示行为正常。值为 0 表示 `/_cluster/health/` 失败。 失败意味着无法从源节点访问主节点。这通常是由网络连接问题或 Amazon 依赖问题导致的。 相关统计数据：Maximum
`MasterSysMemoryUtilization`	使用中的主节点内存的百分比。 相关统计数据：Maximum

专用协调器节点指标

Amazon S OpenSearch ervice 为专用协调器节点提供了以下指标。

指标	描述
`CoordinatorCPUUtilization`	专用协调器节点使用的 CPU 资源的最大百分比。建议在此指标达到 80% 时增加实例类型的大小。 相关统计数据：Maximum

指标	描述
CoordinatorJVMMemoryPressure	用于集群中所有专用协调器节点的 Java 堆的最大百分比。建议在此指标达到 85% 时迁移到更大的实例类型。 相关统计数据：Maximum
CoordinatorOldGenJVMMemoryPressure	每个主节点上用于“上一代”的 Java 堆的最大百分比。 相关统计数据：Maximum
CoordinatorSysMemoryUtilization	使用中的协调器节点内存的百分比。 相关统计数据：Maximum
CoordinatorFreeStorageSpace	该指标指示服务未将协调器节点作为数据节点使用。

EBS 卷指标

Amazon S OpenSearch ervice 为 EBS 卷提供以下指标。

指标	描述
ReadLatency	EBS 卷上读取操作的延迟（以秒为单位）。此指标也可用于单独的节点。 相关统计数据：最小值、最大值、平均值
WriteLatency	EBS 卷上写入操作的延迟（以秒为单位）。此指标也可用于单独的节点。 相关统计数据：最小值、最大值、平均值
ReadThroughput	EBS 卷上读取操作的吞吐量（以字节/秒为单位）。此指标也可用于单独的节点。 相关统计数据：最小值、最大值、平均值
ReadThroughputMicroBursting	考虑微爆时，EBS 卷上读取操作的吞吐量（以每秒字节数为单位）。此指标也可用于单独的节点。当 EBS 卷在显著缩短的时间段（少于一分钟）内突破高 IOPS 或吞吐量时，就会发生微爆。

指标	描述
	相关统计数据：最小值、最大值、平均值
WriteThroughput	EBS 卷上写入操作的吞吐量（以字节/秒为单位）。此指标也可用于单独的节点。 相关统计数据：最小值、最大值、平均值
WriteThroughputMicroBursting	考虑微爆时，针对 EBS 卷上写入操作的吞吐量（以每秒字节数为单位）。此指标也可用于单独的节点。当 EBS 卷在显著缩短的时间段（少于一分钟）内突破高 IOPS 或吞吐量时，就会发生微爆。 相关统计数据：最小值、最大值、平均值
DiskQueueDepth	针对 EBS 卷的待处理输入和输出 (I/O) 请求的数量。 相关统计数据：最小值、最大值、平均值
ReadIOPS	针对 EBS 卷上的读取操作的每秒输入和输出 (I/O) 操作数。此指标也可用于单独的节点。 相关统计数据：最小值、最大值、平均值
ReadIOPSMicroBursting	考虑到微爆时，针对 EBS 卷上的读取操作的每秒输入和输出 (I/O) 操作数。此指标也可用于单独的节点。当 EBS 卷在显著缩短的时间段（少于一分钟）内突破高 IOPS 或吞吐量时，就会发生微爆。 相关统计数据：最小值、最大值、平均值
WriteIOPS	针对 EBS 卷上的写入操作的每秒输入和输出 (I/O) 操作数。此指标也可用于单独的节点。 相关统计数据：最小值、最大值、平均值
WriteIOPS MicroBursting	考虑微爆时，针对 EBS 卷上的写入操作的每秒输入和输出 (I/O) 操作数。此指标也可用于单独的节点。当 EBS 卷在显著缩短的时间段（少于一分钟）内突破高 IOPS 或吞吐量时，就会发生微爆。 相关统计数据：最小值、最大值、平均值

指标	描述
BurstBalance	一个 EBS 卷的可爆发存储桶中剩余输入和输出（I/O）积分的百分比。值为 100 表示该卷积累的积分数量已达最大数量。如果此百分比低于 70%，请参阅 the section called “EBS 可爆发容量余额低”。对于具有 gp3 卷类型的域以及具有卷大小超过 1000 GiB 的 gp2 卷的域，突增余额保持在 0。 相关统计数据：最小值、最大值、平均值
VolumeStalledIOcheck	EBS 卷的状态以确定何时受损。指标是一个二进制值，将根据 EBS 卷能否完成输入和输出操作返回 0（通过）或 1（失败）状态。 VolumeStalledIOcheck 也适用于单个节点。 相关统计数据：最小值、最大值、平均值

实例指标

Amazon S OpenSearch ervice 为域中的每个实例提供以下指标。 OpenSearch Service 还聚合这些实例指标以便您了解总体集群运行状况。您可以使用控制台中的 Sample Count（样本数）统计数据验证此行为。请注意，下表中的每个指标对于节点 和 集群都有相关的统计数据。

> **Important**
>
> Elasticsearch 的不同版本使用不同的线程池来处理对 _index API 的调用。Elasticsearch 1.5 和 2.3 使用索引线程池。Elasticsearch x、6.0 和 6.2 使用批量线程池。 OpenSearch Elasticsearch 6.3 及更高版本使用写入线程池。目前，S OpenSearch ervice 控制台不包含批量线程池的图形。
> 使用 GET _cluster/settings?include_defaults=true 来检查集群的线程池和队列大小。

指标	描述
FetchLatency	节点中所有分片提取操作所用的总时间差（以毫秒为单位），介于 N 分钟和 (N-1) 分钟之间。

指标	描述
	相关节点统计数据：Average 相关集群统计数据：Average、Maximum
FetchRate	数据节点上所有分片每分钟执行的分片提取操作总数。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum、Sum
ScrollTotal	数据节点上所有分片每分钟执行的分片滚动操作总数。 相关节点统计数据：平均值、Maximum 相关集群统计数据：Average、Maximum、Sum
ScrollCurrent	当前处于运行状态的分片滚动操作数。 相关节点统计数据：平均值、Maximum 相关集群统计数据：Average、Maximum、Sum
OpenContexts	打开的搜索上下文的数量。 相关节点统计数据：平均值、Maximum 相关集群统计数据：Average、Maximum、Sum
ThreadCount	该 OpenSearch 进程当前正在使用的线程总数。 相关节点统计数据：平均值、Maximum 相关集群统计数据：Average、Maximum、Sum
ShardReactivateCount	从空闲状态下激活所有分片的总次数。 相关节点统计数据：总计、Maximum 相关集群统计数据：总计、Maximum

指标	描述
ConcurrentSearchRate	数据节点上所有分片每分钟使用并发分段搜索的搜索请求总数。对 _search API 的单次调用可能会从许多不同的分片返回结果。如果这些分片中有 5 个位于一个节点上，则节点会为此指标报告 5 次，即使客户只发出一次请求也是如此。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum、Sum
ConcurrentSearchLatency	在 N 分钟和 (N-1) 分钟之间，节点中使用并发分段搜索的所有搜索的总时间差（以毫秒为单位）。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum
IndexingLatency	节点中所有索引操作所用的总时间差（以毫秒为单位），介于 N 分钟和 (N-1) 分钟之间。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum
IndexingRate	每分钟的索引操作数。对 _bulk API 的单次调用，该 API 添加两个文档并将两个计数更新为四个操作，这可在一个或多个节点中扩散。如果该索引有一个或多个副本，并且该索引有一个优化型实例的 OpenSearch域中，集群中的其他节点也会记录总计四个索引编制操作。对于具有优化型实例的 OpenSearch 域，其他有副本的节点不会记录任何操作。文档删除不计入此指标。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum、Sum

指标	描述
SearchLatency	节点中所有搜索的总时间差（以毫秒为单位），介于 N 分钟和 (N-1) 分钟之间。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum
SearchRate	数据节点上所有分片的每分钟搜索请求总数。对 _search API 的单次调用可能会从许多不同的分片返回结果。如果这些分片中有 5 个位于一个节点上，则节点会为此指标报告 5 次，即使客户只发出一次请求也是如此。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum、Sum
SegmentCount	数据节点上的分段数。您拥有的分段越多，每次搜索所花费的时间就越长。 OpenSearch 偶尔会将较小的分段合并为较大的分段。 相关节点统计数据：最大值、平均值 相关集群统计数据：Sum、Maximum、Average
SysMemoryUtilization	使用中的实例内存的百分比。此指标的值较高是正常的，通常不表示集群存在问题。有关潜在性能和稳定性问题的更好指示，请参阅 JVMMemoryPressure 指标。 相关节点统计数据：Minimum、Maximum、Average 相关集群统计数据：Minimum、Maximum、Average
JVMGCYoungCollectionCount	“年轻代”垃圾回收的运行次数。大量不断增长的运行数对于集群操作来说是正常的。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average

指标	描述
JVMGCYoungCollectionTime	集群执行“年轻代”垃圾回收所花费的时间，以毫秒为单位。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
JVMGCOldCollectionCount	“年老代”垃圾回收的运行次数。在具有足够资源的集群中，此数字应保持很小并且不会频繁增长。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
JVMGCOldCollectionTime	集群执行“年老代”垃圾回收所花费的时间，以毫秒为单位。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
OpenSearchDashboardsConcurrentConnections	OpenSearch控制面板的活动并发连接数。如果此数字始终很高，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
OpenSearchDashboardsHealthyNode	单个 OpenSearch 控制面板节点的运行状况检查。值为 1 表示行为正常。值为 0 表示无法访问控制面板。 相关节点统计数据：最小值 相关集群统计数据：Minimum、Maximum、Average
OpenSearchDashboardsHeapTotal	分配给 OpenSearch 控制面板的堆内存量，以 MiB 为单位分配此值。不同的 EC2 实例类型可能会影响精确的内存分配。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average

指标	描述
OpenSearchDashboardsHeapUsed	OpenSearch 控制面板使用的绝对堆内存数，以 MiB 为单位。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
OpenSearchDashboardsHeapUtilization	OpenSearch控制面板使用的可用堆内存的最大百分比。如果此值超过 80%，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Minimum、Maximum、Average
OpenSearchDashboardsOS1MinuteLoad	OpenSearch 控制面板的一分钟 CPU 负载平均值。理想情况下，CPU 负载应保持在 1.00 以下。虽然临时峰值很好，但如果此指标始终高于 1.00，我们建议增加实例类型的大小。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum
OpenSearchDashboardsRequestTotal	向 OpenSearch 控制面板发出的 HTTP 请求的总计数。如果您的系统速度较慢，或者您看到大量的控制面板请求，请考虑增加实例类型的大小。 相关节点统计数据：总计 相关集群统计数据：Sum
OpenSearchDashboardsResponseTimesMaxInMillis	OpenSearch 控制面板响应请求的最大时间量（以毫秒为单位）。如果请求一直花费很长时间才能返回结果，请考虑增加实例类型的大小。 相关节点统计数据：Maximum 相关集群统计数据：最大值、平均值

指标	描述
SearchTaskCancelled	协调器节点取消的次数。 相关节点统计数据：总计 相关集群统计数据：Sum
SearchShardTaskCancelled	数据节点取消的次数。 相关节点统计数据：总计 相关集群统计数据：Sum
ThreadpoolForce_mergeQueue	强制合并线程池中的排队任务数。如果队列大小一直很大，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
ThreadpoolForce_mergeRejected	强制合并线程池中的已拒绝任务数。如果此数字持续增长，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum
ThreadpoolForce_mergeThreads	强制合并线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
ThreadpoolIndexQueue	索引线程池中的排队任务数。如果队列大小一直很大，请考虑扩展您的集群。索引队列的最大大小为 200。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average

指标	描述
ThreadpoolIndexRejected	索引线程池中的已拒绝任务数。如果此数字持续增长，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum
ThreadpoolIndexThreads	索引线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
ThreadpoolSearchQueue	搜索线程池中的排队任务数。如果队列大小一直很大，请考虑扩展您的集群。搜索队列的最大大小为 1000。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
ThreadpoolSearchRejected	搜索线程池中的已拒绝任务数。如果此数字持续增长，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum
ThreadpoolSearchThreads	搜索线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
Threadpoolsql-workerQueue	SQL 搜索线程池中的排队任务数。如果队列大小一直很大，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average

指标	描述
Threadpoolsql-workerRejected	SQL 搜索线程池中的已拒绝任务数。如果此数字持续增长，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum
Threadpoolsql-workerThreads	SQL 搜索线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
ThreadpoolBulkQueue	批量线程池中的排队任务数。如果队列大小一直很大，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
ThreadpoolBulkRejected	批量线程池中的已拒绝任务数。如果此数字持续增长，请考虑扩展您的集群。 相关节点统计数据：Maximum 相关集群统计数据：Sum
ThreadpoolBulkThreads	批量线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
ThreadpoolIndexSearcherQueue	索引搜索器线程池中已排队的任务数。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average

指标	描述
ThreadpoolIndexSearcherRejected	索引搜索器线程池中已拒绝的任务数。 相关节点统计数据：Maximum 相关集群统计数据：Sum
ThreadpoolIndexSearcherThreads	索引搜索器线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
ThreadpoolWriteThreads	写入线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
ThreadpoolWriteQueue	写入线程池中的排队任务数。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum
ThreadpoolWriteRejected	写入线程池中的已拒绝任务数。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum **Note** 由于版本 7.1 中的默认写入队列大小从 200 增加到 10000，因此此指标不再是从 OpenSearch Service 拒绝的唯一指示符。使用 `CoordinatingWriteRejected` 、`PrimaryWriteRejected` 和 `ReplicaWriteRejected` 指标来监控版本 7.1 及更高版本中的拒绝。

指标	描述
`CoordinatingWriteRejected`	自上次 S OpenSearch ervice 进程启动以来，由于索引压力而在协调节点上发生的拒绝总数。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum 此指标在版本 7.1 及更高版本中可用。
`PrimaryWriteRejected`	自上次 S OpenSearch ervice 进程启动以来，由于索引压力而在主分区上发生的拒绝总数。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum 此指标在版本 7.1 及更高版本中可用。
`ReplicaWriteRejected`	自上次 S OpenSearch ervice 进程启动以来，由于索引压力而在副本分区上发生的拒绝总数。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum 此指标在版本 7.1 及更高版本中可用。
`WorkloadManagementEnabled`	指示是否启用工作负载管理功能。值为 1 表示已启用，值为 0 表示已禁monitor_only 用。 相关节点统计数据：最大值、最小值 相关集群统计数据：Average、Sum 此指标在版本 7.1 及更高版本中可用。

指标	描述
SoftQueryGroupCount	域中软模式下的查询组数量。 相关节点统计数据：平均值、Maximum 相关集群统计数据：Average、Maximum、Sum 此指标在版本 7.1 及更高版本中可用。
EnforcedQueryGroupCount	域中处于强制模式的查询组的数量。 相关节点统计数据：平均值、Maximum 相关集群统计数据：Average、Maximum、Sum 此指标在版本 7.1 及更高版本中可用。

UltraWarm 指标

Amazon OpenSearch 服务为UltraWarm节点提供以下指标。

指标	描述
WarmCPUUtilization	集群中 UltraWarm 节点的 CPU 使用率百分比。最大值显示 CPU 利用率最高的节点。Average 代表集群中的所有 UltraWarm 节点。此指标也适用于单个 UltraWarm 节点。 相关统计数据：Maximum、Average
WarmFreeStorageSpace	以 MiB 为单位的可用温存储空间量。由于 UltraWarm使用 Amazon S3 而非附加磁盘，Sum因此这是唯一相关的统计数据。您必须保留一分钟的时间来获取准确值。 相关统计数据：总计
WarmSearchableDocuments	跨集群中所有温索引的可搜索文档总数。您必须保留一分钟的时间来获取准确值。 相关统计数据：总计

指标	描述
WarmSearchLatency	所有搜索的总时间差（以毫秒为单位），介于 N 分钟和 (N-1) 分钟 UltraWarm 之间。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum
WarmSearchRate	每分钟针对 UltraWarm 节点上所有分片的搜索请求总数。对 _search API 的单次调用可能会从许多不同的分片返回结果。如果这些分片中有 5 个位于一个节点上，则节点会为此指标报告 5 次，即使客户只发出一次请求也是如此。 相关节点统计数据：Average 相关集群统计数据：Average、Maximum、Sum
WarmStorageSpaceUtilization	集群使用的温存储空间总量。 相关统计数据：Maximum
HotStorageSpaceUtilization	集群使用的热存储空间总量。 相关统计数据：Maximum
WarmSysMemoryUtilization	使用中的温节点内存的百分比。 相关统计数据：Maximum
HotToWarmMigrationQueueSize	当前等待从热存储迁移到温存储的索引数。 相关统计数据：Maximum
WarmToHotMigrationQueueSize	当前等待从温存储迁移到热存储的索引数。 相关统计数据：Maximum

指标	描述
HotToWarmMigrationFailureCount	从热迁移到温迁移失败的总数。 相关统计数据：总计
HotToWarmMigrationForceMergeLatency	迁移过程的强制合并阶段的平均延迟时间。如果这个阶段始终需要太长时间，请考虑增加 index.ultrawarm.migration.force_merge.max_num_segments 。 相关统计数据：Average
HotToWarmMigrationSnapshotLatency	迁移过程快照阶段的平均延迟时间。如果此阶段始终花费太长时间，请确保分区的大小适当，并在整个集群中分布。 相关统计数据：Average
HotToWarmMigrationProcessingLatency	成功从热迁移到温迁移的平均延迟时间，不包括队列中花费的时间。此值是完成迁移过程的强制合并、快照和分区重新定位阶段所需的时间总和。 相关统计数据：Average
HotToWarmMigrationSuccessCount	成功从热迁移到温迁移的总数。 相关统计数据：总计
HotToWarmMigrationSuccessLatency	成功从热迁移到温迁移的平均延迟时间，包括在队列中花费的时间。 相关统计数据：Average
WarmThreadpoolSearchThreads	UltraWarm 搜索线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Average、Sum

指标	描述
`WarmThreadpoolSearchRejected`	UltraWarm 搜索线程池中的已拒绝任务数。如果此数量持续增长，请考虑增加更多 UltraWarm节点。 相关节点统计数据：Maximum 相关集群统计数据：Sum
`WarmThreadpoolSearchQueue`	UltraWarm 搜索线程池中的排队任务数。如果队列大小一直很大，请考虑增加更多 UltraWarm节点。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
`WarmJVMMemoryPressure`	用于 UltraWarm节点的 Java 堆的最大百分比。 相关统计数据：Maximum Note 在服务软件 R20220323 中更改了此指标的逻辑。有关更多信息，请参阅版本注释。
`WarmOldGenJVMMemoryPressure`	每个 UltraWarm 节点上用于 “上一代” 的 Java 堆的最大百分比。 相关统计数据：Maximum
`WarmJVMGCYoungCollectionCount`	“年轻代” 垃圾回收在 UltraWarm 节点上的运行次数。大量不断增长的运行数对于集群操作来说是正常的。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average

指标	描述
`WarmJVMGCYoungCollectionTime`	集群在节点上执行 “年轻代” 垃圾回收所花费的时间，以毫秒为单位。UltraWarm 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
`WarmJVMGCOldCollectionCount`	“年老代” 垃圾回收在 UltraWarm 节点上的运行次数。在具有足够资源的集群中，此数字应保持很小并且不会频繁增长。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
`WarmConcurrentSearchRate`	UltraWarm 节点上所有分片每分钟使用并发分段搜索的搜索请求总数。对 `_search` API 的单次调用可能会从许多不同的分片返回结果。如果这些分片中有 5 个位于一个节点上，则节点会为此指标报告 5 次，即使客户只发出一次请求也是如此。 相关节点统计数据：Average 相关集群统计数据：Sum、Maximum、Average
`WarmConcurrentSearchLatency`	在 N 分钟和 (N-1) 分钟之间， UltraWarm 节点中使用并发分段搜索的所有搜索的总时间差（以毫秒为单位）。 相关节点统计数据：Average 相关集群统计数据：最大值、平均值
`WarmThreadpoolIndexSearcherQueue`	UltraWarm 索引搜索器线程池中的排队任务数。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average

指标	描述
WarmThreadpoolIndexSearcherRejected	UltraWarm 索引搜索器线程池中的已拒绝任务数。 相关节点统计数据：Maximum 相关集群统计数据：Sum
WarmThreadpoolIndexSearcherThreads	UltraWarm 索引搜索器线程池的大小。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Average

冷存储指标

Amazon S OpenSearch ervice 为冷存储提供以下指标。

指标	描述
ColdStorageSpaceUtilization	集群使用的冷存储空间总量，以 MiB 为单位。 相关统计数据：最大值
ColdToWarmMigrationFailureCount	从冷到温迁移失败的总数。 相关统计数据：总计
ColdToWarmMigrationLatency	成功完成冷到温迁移所需的时间量。 相关统计数据：Average
ColdToWarmMigrationQueueSize	当前等待从冷存储迁移到温存储的索引数。 相关统计数据：Maximum
ColdToWarmMigrationSuccessCount	成功从冷到温迁移的总数。 相关统计数据：总计

指标	描述
WarmToColdMigrationFailureCount	从温到冷迁移失败的总数。 相关统计数据：总计
WarmToColdMigrationLatency	成功完成温到冷迁移的时间量。 相关统计数据：Average
WarmToColdMigrationQueueSize	当前等待从温存储迁移到冷存储的索引数。 相关统计数据：Maximum
WarmToColdMigrationSuccessCount	成功从温到冷迁移的总数。 相关统计数据：总计

OR1 指标

Amazon OpenSearch 服务为OR1实例提供以下指标。

指标	描述
RemoteStorageUsedSpace	集群使用的 Amazon S3 空间总量（单位为 MiB）。 相关统计数据：总计
RemoteStorageWriteRejected	由于远程存储和复制压力而在主分片上被拒绝的请求总数。此项从上次 S OpenSearch ervice 进程启动时开始计算。 相关统计数据：总计
ReplicationLagMaxTime	副本分片相比主分片的滞后时间大小（以毫秒为单位）。 相关统计数据：Maximum

提醒指标

Amazon Ser OpenSearch vice 为提醒提供以下指标。

指标	描述
AlertingDegraded	值为 1 表示警报索引为红色，或一个或多个节点未按计划运行。值为 0 表示行为正常。 相关统计数据：Maximum
AlertingIndexExists	值为 1 表示 .opensearch-alerting-config 索引存在。值为 0 表示该索引不存在。在您首次使用警报功能之前，此值将保持为 0。 相关统计数据：Maximum
AlertingIndexStatus.green	索引的运行状况。值为 1 表示绿色。值为 0 表示索引不存在或不是绿色。 相关统计数据：Maximum
AlertingIndexStatus.red	索引的运行状况。值为 1 表示红色。值为 0 表示索引不存在或不是红色。 相关统计数据：Maximum
AlertingIndexStatus.yellow	索引的运行状况。值为 1 表示黄色。值为 0 表示索引不存在或不是黄色。 相关统计数据：Maximum
AlertingNodesNotOnSchedule	值为 1 表示某些作业未按计划运行。值为 0 表示所有警报作业都按计划运行（或警报作业不存在）。检查S OpenSearch ervice 控制台或_nodes/stats 请求以查看是否有节点显示高资源使用率。 相关统计数据：Maximum
AlertingNodesOnSchedule	值为 1 表示所有警报作业都按计划运行（或警报作业不存在）。值为 0 表示某些作业未按计划运行。 相关统计数据：Maximum

指标	描述
AlertingScheduledJobEnabled	值为 1 表示 opensearch.scheduled_jobs.enabled 集群设置为 true。值为 0 表示该设置为 false，并且计划的作业已禁用。 相关统计数据：Maximum

异常检测指标

Amazon S OpenSearch ervice 为异常检测提供以下指标。

指标	描述
ADPluginUnhealthy	值为 1 表示异常检测插件无法正常工作，或者因为故障次数太多，或者因为它使用了一个红色的索引。值为 0 表示插件正按预期工作。 相关统计数据：Maximum
ADExecuteRequestCount	检测异常的请求数。 相关统计数据：总计
ADExecuteFailureCount	检测异常的失败请求数。 相关统计数据：总计
ADHCExecuteFailureCount	检测高基数探测器异常的失败请求数。 相关统计数据：总计
ADHCExecuteRequestCount	检测高基数探测器异常的请求数。 相关统计数据：总计
ADAnomalyResultsIndexStatusIndexExists	值为 1 表示 .opensearch-anomaly-results 别名指向的索引存在。在首次使用异常检测之前，此值将保持为 0。 相关统计数据：Maximum

指标	描述
ADAnomalyResultsIndexStatus.red	值为 1 表示 .opensearch-anomaly-results 别名指向的索引为红色。值为 0 表示不是。在首次使用异常检测之前，此值将保持为 0。 相关统计数据：Maximum
ADAnomalyDetectorsIndexStatusIndexExists	值为 1 表示 .opensearch-anomaly-detectors 索引存在。值为 0 表示该索引不存在。在首次使用异常检测之前，此值将保持为 0。 相关统计数据：Maximum
ADAnomalyDetectorsIndexStatus.red	值为 1 表示 .opensearch-anomaly-detectors 索引为红色。值为 0 表示不是。在首次使用异常检测之前，此值将保持为 0。 相关统计数据：Maximum
ADModelsCheckpointIndexStatusIndexExists	值为 1 表示 .opensearch-anomaly-checkpoints 索引存在。值为 0 表示该索引不存在。在首次使用异常检测之前，此值将保持为 0。 相关统计数据：Maximum
ADModelsCheckpointIndexStatus.red	值为 1 表示 .opensearch-anomaly-checkpoints 索引为红色。值为 0 表示不是。在首次使用异常检测之前，此值将保持为 0。 相关统计数据：Maximum

异步搜索指标

Amazon S OpenSearch ervice 为异步搜索提供以下指标。

异步搜索协调器节点统计数据（每个协调器节点）

指标	描述
AsynchronousSearchSubmissionRate	过去 1 分钟内提交的异步搜索数。

指标	描述
AsynchronousSearchInitializedRate	过去 1 分钟内初始化的异步搜索数。
AsynchronousSearchRunningCurrent	当前正在运行的异步搜索数。
AsynchronousSearchCompletionRate	过去 1 分钟内成功完成的异步搜索数。
AsynchronousSearchFailureRate	最后一分钟内完成和失败的异步搜索数。
AsynchronousSearchPersistRate	过去 1 分钟内持续存在的异步搜索数。
AsynchronousSearchPersistFailedRate	最后一分钟内失败的异步搜索数。
AsynchronousSearchRejected	自节点启动时间以来拒绝的异步搜索总数。
AsynchronousSearchCancelled	自节点启动时间以来取消的异步搜索总数。

指标	描述
AsynchronousSearchMaxRunningTime	最后一分钟内节点上运行时间最长的异步搜索的持续时间。

异步搜索集群统计数据

指标	描述
AsynchronousSearchStoreHealth	最后一分钟内持久索引（红色/非红色）中的存储运行状况。
AsynchronousSearchStoreSize	过去 1 分钟内跨所有分区的系统索引大小。
AsynchronousSearchStoredResponseCount	过去 1 分钟内系统索引中存储的响应数。

自动调整指标

Amazon S OpenSearch ervice 为自动调整提供以下指标。

指标	描述
AutoTuneChangesHistoryHeapSize	堆大小调整值的更改历史记录（以 MiB 为单位）。
AutoTuneChangesHis	JVM YongGen 参数的更改历史记录。

指标	描述
toryJVMYoungGenArgs	
AutoTuneFailed	用于指示自动调整更改失败的布尔值。
AutoTuneSucceeded	用于指示自动调整更改成功的布尔值。
AutoTuneValue	无中断更改的队列更改历史记录（计数）和缓存调整更改历史记录（以 MiB 为单位）。

带待机功能的多可用区指标

Amazon S OpenSearch ervice 为带待机功能的多可用区提供以下指标。

活动的可用区中数据节点的节点-级别指标

指标	描述
CPUUtilization	集群中数据节点的 CPU 利用率百分比。最大值显示 CPU 利用率最高的节点。平均值表示集群中的所有节点。此指标也可用于单独的节点。
FreeStorageSpace	集群中各数据节点的可用空间。Sum 显示集群的总可用空间，但您必须保留一分钟的时间来获取准确值。Minimum 和 Maximum 分别显示具有最小和最大可用空间的节点。此指标也适用于单个节点。 OpenSearch ClusterBlockException 当该指标达到0时，服务会抛出。要恢复，您必须删除索引，添加更大的实例，或向现有实例添加基于 EBS 的存储。要了解更多信息，请参阅the section called “缺少可用存储空间”。 OpenSearch 服务控制台以 GiB 为单位显示此值。Amazon CloudWatch 控制台以 MiB 为单位显示此值。
JVMMemoryPressure	用于集群中所有数据节点的 Java 堆的最大百分比。 OpenSearch Service 将实例的一半内存用于 Java 堆，最大堆大小为 32 GiB。您最多可以将实例的 RAM 垂直扩展至 64GiB，此时可以通过添加实例水平扩展。请参阅the section called “推荐的 CloudWatch 警报”。

指标	描述
SysMemory Utilization	使用中的实例内存的百分比。此指标的值较高是正常的，通常不表示集群存在问题。有关潜在性能和稳定性问题的更好指示，请参阅 JVMMemory Pressure 指标。
IndexingLatency	节点中所有索引操作所用的总时间差（以毫秒为单位），介于 N 分钟和 (N-1) 分钟之间。
IndexingRate	每分钟的索引操作数。
SearchLatency	节点中所有搜索的总时间差（以毫秒为单位），介于 N 分钟和 (N-1) 分钟之间。
SearchRate	数据节点上所有分片的每分钟搜索请求总数。
ThreadpoolSearchQueue	搜索线程池中的排队任务数。如果队列大小一直很大，请考虑扩展您的集群。搜索队列的最大大小为 1000。
ThreadpoolWriteQueue	写入线程池中的排队任务数。
ThreadpoolSearchRejected	搜索线程池中的已拒绝任务数。如果此数字持续增长，请考虑扩展您的集群。
ThreadpoolWriteRejected	写入线程池中的已拒绝任务数。

活动的可用区中集群-级别指标

指标	描述
DataNodes	活动分片和备用分片的总数。
DataNodes Shards.active	活动主分区和副本分区的总数。

指标	描述
DataNodes Shards.unassigned	未分配给集群中节点的分区数。
DataNodes Shards.initializing	正在初始化的分区数。
DataNodes Shards.relocating	正在重新定位的分区数。

可用区轮换指标

如果是 `ActiveReads.`*`Availability-Zone`* `= 1`，则该区处于活动状态。如果是 `ActiveReads.`*`Availability-Zone`* `= 0`，则该区处于待机状态。

时间点指标

Amazon S OpenSearch ervice 为时间点 (PIT) 搜索提供以下指标。

PIT 协调器节点统计数据（每个协调器节点）

指标	描述
CurrentPointInTime	节点中活动 PIT 搜索上下文的数量。
TotalPointInTime	自节点启动时间以来过期的 PIT 搜索上下文数量。
AvgPointInTimeAliveTime	自节点启动时间以来 PIT 搜索上下文保持活动状态的平均时间。
HasActivePointInTime	值为 1 表示自节点启动时间以来节点上活动的 PIT 上下文。值 0 表示没有。

指标	描述
HasUsedPointInTime	值为 1 表示自节点启动时间以来节点上过期的 PIT 上下文。值 0 表示没有。

SQL 指标

Amazon Serv OpenSearch ice 为 SQL 支持提供以下指标。

指标	描述
SQLFailedRequestCountByCusErr	由于客户端问题而失败的对 _sql API 的请求数。例如，请求可能会因 IndexNotFoundException 返回 HTTP 状态代码 400。 相关统计数据：总计
SQLFailedRequestCountBySysErr	由于服务器问题或功能限制而失败的对 _sql API 的请求数。例如，请求可能会因 VerificationException 返回 HTTP 状态代码 503。 相关统计数据：总计
SQLRequestCount	对 _sql API 的请求数。 相关统计数据：总计
SQLDefaultCursorRequestCount	类似于 SQLRequestCount ，但仅统计分页请求。 相关统计数据：总计
SQLUnhealthy	值为 1 表示 SQL 插件将返回 5xx 响应代码或将无效的查询 DSL 传递到 OpenSearch 来响应特定请求。其他请求将继续成功。值为 0 表示最近未失败。如果您看到持续值为 1，请排查您的客户端对插件发出的请求的问题。 相关统计数据：Maximum

k-NN 指标

Amazon S OpenSearch ervice 包括 k 最近邻 (k-NN) 插件的以下指标。

指标	描述
KNNCacheCapacityReached	每节点指标，用于是否已达到缓存容量。此指标仅与近似 k-NN 搜索相关。 相关统计数据：Maximum
KNNCircuitBreakerTriggered	每个集群指标，用于是否触发断路器。如果任何节点返回 KNNCacheCapacityReached 值为 1，则此值也将返回 1。此指标仅与近似 k-NN 搜索相关。 相关统计数据：Maximum
KNNEvictionCount	由于内存限制或空闲时间而从缓存中移出的图形数的每节点指标。不计入由于索引删除而发生的显式移出。此指标仅与近似 k-NN 搜索相关。 相关统计数据：总计
KNNGraphIndexErrors	每节点指标，用于将文档的 knn_vector 字段添加到产生错误的图形的请求数。 相关统计数据：总计
KNNGraphIndexRequests	每节点指标，用于将文档的 knn_vector 字段添加到图形的请求数。 相关统计数据：总计
KNNGraphMemoryUsage	当前缓存大小（内存中所有图形的总大小）的每节点指标（以千字节为单位）。此指标仅与近似 k-NN 搜索相关。 相关统计数据：Average
KNNGraphQueryErrors	产生错误的图形查询数的每节点指标。 相关统计数据：总计

指标	描述
KNNGraphQueryRequests	图形查询次数的每节点指标。 相关统计数据：总计
KNNHitCount	缓存命中次数的每节点指标。当用户查询已加载到内存中的图形时，会发生缓存命中。此指标仅与近似 k-NN 搜索相关。 相关统计数据：总计
KNNLoadExceptionCount	尝试将图形加载到缓存时发生异常次数的每节点指标。此指标仅与近似 k-NN 搜索相关。 相关统计数据：总计
KNNLoadSuccessCount	每节点指标，用于插件将图形成功加载到缓存中的次数。此指标仅与近似 k-NN 搜索相关。 相关统计数据：总计
KNNMissCount	缓存未命中次数的每节点指标。当用户查询尚未加载到内存中的图形时，会发生缓存未命中。此指标仅与近似 k-NN 搜索相关。 相关统计数据：总计
KNNQueryRequests	k-NN 插件收到的查询请求数的每节点指标。 相关统计数据：总计
KNNScriptCompilationErrors	每节点指标，用于脚本编译过程中的错误数。此统计数据仅与 k-NN 分数脚本搜索相关。 相关统计数据：总计
KNNScriptCompilations	每节点指标，用于编译 k-NN 脚本的次数。此值通常应为 1 或 0，但是如果包含已编译脚本的缓存已填充，k-NN 脚本可能会重新编译。此统计数据仅与 k-NN 分数脚本搜索相关。 相关统计数据：总计

指标	描述
KNNScriptQueryErrors	脚本查询过程中错误数的每节点指标。此统计数据仅与 k-NN 分数脚本搜索相关。 相关统计数据：总计
KNNScriptQueryRequests	脚本查询总数的每节点指标。此统计数据仅与 k-NN 分数脚本搜索相关。 相关统计数据：总计
KNNTotalLoadTime	k-NN 将图形加载到缓存中所花费的时间（以纳秒为单位）。此指标仅与近似 k-NN 搜索相关。 相关统计数据：总计

跨集群搜索指标

Amazon S OpenSearch ervice 为跨集群搜索提供以下指标。

源域指标

指标	维度	描述
CrossClusterOutboundConnections	ConnectionId	连接的节点数。如果响应中包含一个或多个跳过的域，则可使用此指标跟踪任何运行状况不佳的连接。如果此数值降至 0，则连接运行状况不佳。
CrossClusterOutboundRequests	ConnectionId	发送到目标域的搜索请求数。用于检查跨集群搜索请求的负载是否使域不堪重负，将此指标的任何峰值与任何 JVM/CPU 峰值相关联。

目标域指标

指标	维度	描述
CrossClusterInboundRequests	ConnectionId	从源域接收的传入连接请求数。

添加 CloudWatch 警报，以防您意外断开连接。有关创建警报的步骤，请参阅基于静态阈值创建CloudWatch警报。

跨集群复制指标

Amazon S OpenSearch ervice 为跨集群复制提供以下指标。

指标	描述
ReplicationRate	每秒复制操作的平均速率。该指标与 IndexingRate 指标类似。
LeaderCheckPoint	对于某个特定连接，涵盖所有复制索引的领导者索引检查点值的和。您可以使用此指标来度量复制延迟。
FollowerCheckPoint	对于某个特定连接，涵盖所有复制索引的跟随者索引检查点值的和。您可以使用此指标来度量复制延迟。
ReplicationNumSyncingIndices	复制状态为 SYNCING 的索引数。
ReplicationNumBootstrappingIndices	复制状态为 BOOTSTRAPPING 的索引数。
ReplicationNumPausedIndices	复制状态为 PAUSED 的索引数。

指标	描述
ReplicationNumFailedIndices	复制状态为 FAILED 的索引数。
CrossClusterOutboundReplicationRequests	关注者域上的复制传输请求数。传输请求是内部请求，每次调用复制 API 操作时都会发生。当关注者域轮询领导者域的变更时，也会发生这些请求。
CrossClusterInboundReplicationRequests	领导者域上的复制传输请求数。传输请求是内部请求，每次调用复制 API 操作时都会发生。
AutoFollowNumSuccessStartReplication	特定连接的复制规则已成功创建的跟随者索引的数量。
AutoFollowNumFailedStartReplication	存在匹配模式时，复制规则未能创建的跟随者索引的数量。出现此问题的原因可能是远程集群上的网络问题或安全问题（即关联的角色不具有启动复制的权限）。
AutoFollowLeaderCallFailure	从跟随者索引到领导者索引的提取新数据的查询是否有任何失败。值为 1 表示在最近一分钟内有 1 个或更多失败的调用。

学习排名指标

Amazon S OpenSearch ervice 为学习排名提供以下指标。

指标	描述
LTRRequestTotalCount	排名请求的总计数。
LTRRequestErrorCount	不成功请求的总计数。
LTRStatus.red	跟踪运行插件所需的索引之一是否为红色。
LTRMemoryUsage	插件使用的总内存。
LTRFeatureMemoryUsageInBytes	学习排名功能字段使用的内存量（以字节为单位）。
LTRFeatresetMemoryUsageInBytes	所有学习排名功能集使用的内存量（以字节为单位）。
LTRModelMemoryUsageInBytes	所有学习排名模型使用的内存量（以字节为单位）。

管道处理语言指标

Amazon S OpenSearch ervice 为管道处理语言提供以下指标。

指标	描述
PPLFailedRequestCountByCusErr	由于客户端问题而失败的对 _ppl API 的请求数。例如，请求可能会因 IndexNotFoundException 返回 HTTP 状态代码 400。
PPLFailedRequestCountBySysErr	由于服务器问题或功能限制而失败的对 _ppl API 的请求数。例如，请求可能会因 VerificationException 返回 HTTP 状态代码 503。

指标	描述
PPLRequestCount	对 _ppl API 的请求数。

使用 Amazon OpenSearch 日志监控 CloudWatch 日志

亚马逊 OpenSearch 服务通过亚马逊 OpenSearch 日志公开以下日 CloudWatch 志：

- 错误日志
- 搜索请求慢速日志
- 分片慢速日志
- 审核日志

搜索分片慢速日志、索引分片慢速日志和错误日志，有助于排查性能和稳定性问题。审核日志跟踪用户活动，以达到合规性目的。所有日志默认情况下已禁用。如果启用，则适用标准 CloudWatch 定价。

> **Note**
>
> 错误日志仅适用于 OpenSearch Elasticsearch 版本 5.1 及更高版本。慢速日志适用于所有版本 OpenSearch 和 Elasticsearch 版本。

对于其日志， OpenSearch 使用 Apache Log4j 2 及其内置日志级别（从最低到最严重）TRACE，、DEBUG、INFOWARN、ERROR和。FATAL

如果您启用错误日志， OpenSearch 服务会将WARNERROR、和的日志行发布FATAL到 CloudWatch。OpenSearch Service 还发布了该DEBUG关卡中的几个例外情况，包括：

- org.opensearch.index.mapper.MapperParsingException
- org.opensearch.index.query.QueryShardException
- org.opensearch.action.search.SearchPhaseExecutionException
- org.opensearch.common.util.concurrent.OpenSearchRejectedExecutionException
- java.lang.IllegalArgumentException

错误日志在许多情况下有助于诊断，包括：

- Painless 脚本编译问题
- 无效的查询
- 索引问题
- 快照失败
- 索引状态管理迁移失败

> **Note**
> 并非所有错误都在错误日志中报告。

> **Note**
> OpenSearch 服务不会记录所有发生的错误。

主题

- 启用日志发布（控制台）
- 启用日志发布 (Amazon CLI)
- 启用日志发布 (Amazon SDKs)
- 启用日志发布 (CloudFormation)
- 设置搜索请求慢速日志阈值
- 设置分片慢速日志阈值
- 测试慢速日志
- 查看日志

启用日志发布（控制台）

OpenSearch 服务控制台是允许向发布日志的最简单方法 CloudWatch。

启用向 CloudWatch （控制台）发布日志

1. 前往 aws.amazon.com，然后选择 “登录” 并提供您的凭证。
2. 在 “分析” 下，选择 “亚马逊 OpenSearch 服务”。

3. 选择要更新的域。
4. 在 Logs（日志）选项卡上，选择一种日志类型，然后选择 Enable（启用）。
5. 创建新的 CloudWatch 日志组或选择现有的日志组。

> **Note**
> 如果计划启用多个日志，建议将它们发布到相应的日志组。这样分开更便于进行日志扫描。

6. 选择一个包含适当权限的访问策略，或使用控制台提供的 JSON 创建策略：

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "cw_log_group_arn:*"
    }
  ]
}
```

   建议您在策略中使用 `aws:SourceAccount` 和 `aws:SourceArn` 条件键来防止出现混淆代理人问题。源帐户是域的所有者，并且源 ARN 是域的 ARN。您的域必须在服务软件 R20211203 或更高版本上才能添加这些条件键。

   例如，您可以将以下条件块添加到策略：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
```

```
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

> **Important**
>
> CloudWatch 日志支持每个区域 10 个资源策略。如果您计划为多个 OpenSearch 服务域启用日志，则应创建并重复使用包含多个日志组的更广泛的策略，以避免达到此限制。有关更新您的策略的步骤，请参阅the section called “启用日志发布 (Amazon CLI)”。

7. 请选择启用。

   您的域状态将从 Active 更改为 Processing。在启用日志发布之前，必须重新回到 Active 状态。此项更改通常需要 30 分钟，但可能需要更长时间，具体取决于您的域配置。

如果您已启用其中的一种分片慢速日志，请参阅the section called “设置分片慢速日志阈值”。如果启用了审核日志，请参阅 the section called “步骤 2：在 OpenSearch 控制面板中打开审核日志”。如果您仅启用了错误日志，您不需要执行任何其他配置步骤。

启用日志发布 (Amazon CLI)

在启用日志发布之前，您需要一个 CloudWatch 日志组。如果还没有日志组，可以使用以下命令创建一个：

```
aws logs create-log-group --log-group-name my-log-group
```

输入下一命令查找日志组的 ARN，然后将它记下来：

```
aws logs describe-log-groups --log-group-name my-log-group
```

现在，您可以向 OpenSearch 服务授予写入日志组的权限。您必须在命令结尾处提供日志组的 ARN：

```
aws logs put-resource-policy \
  --policy-name my-policy \
  --policy-document '{ "Version": "2012-10-17", "Statement": [{ "Sid": "",
 "Effect": "Allow", "Principal": { "Service": "es.amazonaws.com"}, "Action":
[ "logs:PutLogEvents","logs:CreateLogStream"],"Resource": "cw_log_group_arn:*"}]}'
```

> **Important**
>
> CloudWatch 日志支持每个区域 10 个资源策略。如果您计划为多个 OpenSearch 服务域启用分片慢日志，则应创建并重复使用包含多个日志组的更广泛的策略，以避免达到此限制。

如果您以后需要查看此策略，请使用 `aws logs describe-resource-policies` 命令。要更新策略，请发出带有新策略文档的相同 `aws logs put-resource-policy` 命令。

最后，可以使用 `--log-publishing-options` 选项来启用发布。该选项的语法与 `create-domain` 和 `update-domain-config` 命令相同。

参数	有效值
`--log-publishing-options`	`SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn=` *`cw_log_group_arn`* `,Enabled=true|false}`
	`INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn=` *`cw_log_group_arn`* `,Enabled=true|false}`
	`ES_APPLICATION_LOGS={CloudWatchLogsLogGroupArn=` *`cw_log_group_arn`* `,Enabled=true|false}`
	`AUDIT_LOGS={CloudWatchLogsLogGroupArn=` *`cw_log_group_arn`* `,Enabled=true|false}`

> **Note**
>
> 如果计划启用多个日志，建议将它们发布到相应的日志组。这样分开更便于进行日志扫描。

示例

下面的示例为指定域启用了发布搜索和索引分片慢速日志：

```
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}"
```

要禁用发布到 CloudWatch，请使用运行相同的命令`Enabled=false`。

如果您已启用其中的一种分片慢速日志，请参阅the section called “设置分片慢速日志阈值”。如果启用了审核日志，请参阅 the section called “步骤 2：在 OpenSearch 控制面板中打开审核日志”。如果您仅启用了错误日志，您不需要执行任何其他配置步骤。

启用日志发布 (Amazon SDKs)

在启用日志发布之前，必须先创建一个 CloudWatch 日志组，获取其 ARN，并授予 OpenSearch 服务写入该组的权限。相关操作记录在《亚马逊 CloudWatch 日志 API 参考》中：

- `CreateLogGroup`
- `DescribeLogGroup`
- `PutResourcePolicy`

您可以使用访问这些操作Amazon SDKs。

Amazon SDKs（Android 和 iOS 除外 SDKs）支持《亚马逊 OpenSearch 服务 API 参考》中定义的所有操作，包括`CreateDomain`和`--log-publishing-options`选项`UpdateDomainConfig`。

如果您已启用其中的一种分片慢速日志，请参阅the section called “设置分片慢速日志阈值”。如果您仅启用了错误日志，您不需要执行任何其他配置步骤。

启用日志发布 (CloudFormation)

在此示例中，我们使用创建名 CloudFormation 为的日志组`opensearch-logs`，分配相应的权限，然后创建一个为应用程序日志、搜索分片慢日志和索引慢日志启用日志发布功能的域。

在启用日志发布之前，您需要创建一个 CloudWatch 日志组：

```
Resources:
  OpenSearchLogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      LogGroupName: opensearch-logs
Outputs:
  Arn:
```

```
    Value:
      'Fn::GetAtt':
        - OpenSearchLogGroup
        - Arn
```

模板将输出日志组的 ARN。在本例中，ARN 为 `arn:aws:logs:us-east-1:123456789012:log-group:opensearch-logs`。

使用 ARN 创建资源策略，授予 OpenSearch 服务写入日志组的权限：

```
Resources:
 OpenSearchLogPolicy:
   Type: AWS::Logs::ResourcePolicy
   Properties:
     PolicyName: my-policy
     PolicyDocument: "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Sid\": \"\",
 \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"es.amazonaws.com\"}, \"Action
\":[ \"logs:PutLogEvents\",\"logs:CreateLogStream\"],\"Resource\": \"arn:aws:logs:us-
east-1:123456789012:log-group:opensearch-logs:*\"}]}"
```

最后，创建以下 CloudFormation 堆栈，该堆栈生成带有日志发布功能的 OpenSearch 服务域。访问策略允许用户 Amazon Web Services 账户 向域发出所有 HTTP 请求。

```
Resources:
  OpenSearchServiceDomain:
    Type: "AWS::OpenSearchService::Domain"
    Properties:
      DomainName: my-domain
      EngineVersion: "OpenSearch_1.0"
      ClusterConfig:
        InstanceCount: 2
        InstanceType: "r6g.xlarge.search"
        DedicatedMasterEnabled: true
        DedicatedMasterCount: 3
        DedicatedMasterType: "r6g.xlarge.search"
      EBSOptions:
        EBSEnabled: true
        VolumeSize: 10
        VolumeType: "gp2"
      AccessPolicies:
        Version: "2012-10-17"
        Statement:
```

```
            Effect: "Allow"
            Principal:
                AWS: "arn:aws:iam::123456789012:user/es-user"
            Action: "es:*"
            Resource: "arn:aws:es:us-east-1:123456789012:domain/my-domain/*"
      LogPublishingOptions:
        ES_APPLICATION_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        SEARCH_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        INDEX_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
```

有关详细的语法信息，请参阅 Amazon CloudFormation 用户指南中的日志发布选项。

设置搜索请求慢速日志阈值

搜索请求慢速日志可用于在版本 2.13 及更高版本上运行的 OpenSearch 服务域上进行搜索。搜索请求慢速日志阈值是针对请求总用时配置的。这与分片请求慢速日志不同，后者是针对单个分片用时配置的。

您可以使用集群设置来指定搜索请求慢速日志。这与分片慢速日志不同，后者是使用索引设置启用的。例如，您可以通过 OpenSearch REST API 指定以下设置：

```
PUT domain-endpoint/_cluster/settings
{
  "transient": {
    "cluster.search.request.slowlog.threshold.warn": "5s",
    "cluster.search.request.slowlog.threshold.info": "2s"
  }
}
```

设置分片慢速日志阈值

OpenSearch 默认情况下禁用分片慢日志。启用向发布分片慢速日志后 CloudWatch，仍必须为每个 OpenSearch 索引指定日志阈值。这些阈值精确定义应在哪个日志级别记录哪些内容。

例如，您可以通过 OpenSearch REST API 指定以下设置：

```
PUT domain-endpoint/index/_settings
{
  "index.search.slowlog.threshold.query.warn": "5s",
  "index.search.slowlog.threshold.query.info": "2s"
}
```

测试慢速日志

要测试搜索请求和分片慢速日志是否都成功发布，请考虑从非常低的值开始以验证日志是否出现在其中 CloudWatch，然后将阈值提高到更有用的级别。

如果日志没有显示，请检查以下内容：

- CloudWatch 日志组存在吗？ 检查控制 CloudWatch 台。
- S OpenSearch ervice 是否有权写入日志组？ 检查 OpenSearch 服务控制台。
- OpenSearch 服务域是否配置为发布到日志组？ 检查 OpenSearch 服务控制台，使用 Amazon CLI `describe-domain-config`选项，或`DescribeDomainConfig`使用其中一个来电 SDKs。
- OpenSearch 日志阈值是否足够低，以至于您的请求超出了这些阈值？

  要检查某个域的搜索请求慢速日志阈值，请使用以下命令：

  ```
  GET domain-endpoint/_cluster/settings?flat_settings
  ```

  要检查某个索引的分片慢速日志阈值，请使用以下命令：

  ```
  GET domain-endpoint/index/_settings?pretty
  ```

如果要禁用索引的慢速日志，请将您更改的任何阈值恢复为其默认值 `-1`。

禁用发布到 CloudWatch 使用 OpenSearch 服务控制台或者 Amazon CLI 不会停止生成 OpenSearch 日志；它只会停止发布这些日志。如果不再需要分片慢速日志，请务必检查索引设置；如果不再需要搜索请求慢速日志，请务必检查域设置。

查看日志

查看应用程序和慢速登录 CloudWatch 就像查看任何其他 CloudWatch 日志一样。有关更多信息，请参阅 Amazon 日志用户指南中的查看 CloudWatch 日志数据。

下面是查看日志时的一些注意事项：

- OpenSearch 服务仅向发布每行的前 255,000 个字符。 CloudWatch其余任何内容都将被截断。对于审核日志，每条消息为 10,000 个字符。
- 在中 CloudWatch，日志流名称的后缀为`-index-slow-logs`、`-search-slow-logs`-`application-logs`、和，`-audit-logs`以帮助识别其内容。

在 Amazon OpenSearch 服务中监控审计日志

如果您的 Amazon S OpenSearch ervice 域使用精细的访问控制，则可以为数据启用审核日志。审计日志是高度可定制的，允许您跟踪 OpenSearch集群上的用户活动，包括身份验证成功和失败、对身份验证的请求 OpenSearch、索引更改以及传入的搜索查询。默认配置会跟踪一组常用的用户操作，但建议根据您的具体需求定制设置。

就像OpenSearch 应用程序日志和慢速日志一样，S OpenSearch ervice 会将审核日志发布到 CloudWatch 日志。如果启用，则适用标准 CloudWatch 定价。

> **Note**
>
> 要启用审核日志，必须将您的用户角色映射到该`security_manager`角色，这样您就可以访问 OpenSearch `plugins/_security` REST API。要了解更多信息，请参阅the section called “修改主用户”。

主题

- 限制
- 启用审计日志
- 使用启用审核日志 Amazon CLI
- 使用配置 API 启用审计日志记录
- 审计日志图层和类别
- 审计日志设置
- 审计日志示例
- 使用 REST API 配置审计日志

限制

审计日志具有以下限制：

- 审计日志不包括被目标域访问策略拒绝的跨集群搜索请求。
- 每个审计日志消息的最大大小为 10,000 个字符。如果审计日志消息超出此限制，则会截断该消息。

启用审计日志

启用审结日志是一个两步流程。首先，将您的域配置为将审核日志发布到 CloudWatch 日志。然后，您可以在 OpenSearch 控制面板中启用审核日志，并对其进行配置以满足您的需求。

> **Important**
>
> 如果您在执行以下步骤时遇到错误，请参阅 the section called “无法启用审核日志” 了解故障排除信息。

步骤 1：启用审计日志并配置访问策略

这些步骤将介绍如何使用控制台启用审计日志。您也可以使用或OpenSearch 服务 API 启用它们。Amazon CLI

为 OpenSearch 服务域启用审核日志（控制台）

1. 选择要打开其配置的域，然后转到 Logs（日志）选项卡。
2. 选择 Audit logs（审计日志），然后再选择 Enable（启用）。
3. 创建 CloudWatch 日志组或选择现有日志组。
4. 选择一个包含适当权限的访问策略，或使用控制台提供的 JSON 创建策略：

   JSON

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

```
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "cw_log_group_arn"
    }
  ]
}
```

建议您在策略中使用 `aws:SourceAccount` 和 `aws:SourceArn` 条件键来防止出现混淆代理人问题。源帐户是域的所有者，并且源 ARN 是域的 ARN。您的域必须在服务软件 R20211203 或更高版本上才能添加这些条件键。

例如，您可以将以下条件块添加到策略：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

5. 请选择启用。

步骤 2：在 OpenSearch 控制面板中打开审核日志

在 OpenSearch 服务控制台中启用审计日志后，还必须在 Dashboards 中启用 OpenSearch 审核日志并对其进行配置以满足您的需求。

1. 打开 “ OpenSearch 控制面板”，然后从左侧菜单中选择 “安全”。
2. 选择审计日志。
3. 选择启用审计日志录入。

控制面板 UI 提供完全控制常规设置和合规性设置。有关所有配置选项的描述，请参阅审计日志设置。

使用启用审核日志 Amazon CLI

以下 Amazon CLI 命令在现有域上启用审核日志：

```
aws opensearch update-domain-config --domain-name my-domain --log-publishing-options
 "AUDIT_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-
group:my-log-group,Enabled=true}"
```

您还可以在创建域时启用审计日志。有关详细信息，请参阅 Amazon CLI 命令参考。

使用配置 API 启用审计日志记录

对配置 API 的以下请求将启用现有域上的审计日志：

```
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
    }
  }
}
```

有关更多信息，请参阅亚马逊 OpenSearch服务 API 参考。

审计日志图层和类别

集群通信发生在两个单独的层：REST 层和传输层。

- REST 层涵盖了与 HTTP 客户端（例如 curl、Logstash、Dashboards、 OpenSearch Java 高级 REST 客户端、Python 请求库）的通信，所有这些都是到达集群的 HTTP 请求。
- 传输层覆盖节点之间的通信。例如，在搜索请求到达集群后（通过 REST 层），服务该请求的协调节点将查询发送到其他节点、接收它们的响应、收集必要的文档，并将它们整理到最终响应中。分片分配和重新平衡等操作也会在传输层上进行。

您可以启用或禁用整个图层的审计日志以及图层的各个审计类别。下表包含审计类别及其可用图层的汇总。

类别	描述	REST 可用	可用于传输
FAILED_LOGIN	请求包含无效凭据，并且身份验证失败。	支持	是
MISSING_PRIVILEGES	用户没有发出请求的权限。	支持	是
GRANTED_PRIVILEGES	用户具有发出请求的权限。	支持	是
OPENSEARCH_SECURITY_INDEX_ATTEMPT	一个请求试图修改 `.opendistro_security` 索引。	否	是
已身份验证	请求包含有效凭据，并且身份验证成功。	支持	是
INDEX_EVENT	请求对索引执行管理操作，例如创建索引、设置别名或执行强制合并。该类别包含的`indices:admin/` 操作的完整列表可在OpenSearch 文档中找到。	否	是

除了这些标准类别之外，精细的访问控制还提供了几个额外的类别，旨在满足数据合规性要求。

类别	描述
COMPLIANCE_DOC_READ	请求对索引中的文档执行读取事件。
COMPLIANCE_DOC_WRITE	请求对索引中的文档执行了写入事件。

类别	描述
COMPLIANCE_INTERNAL_CONFIG_READ	在 `.opendistro_security` 索引中执行读取事件的请求。
COMPLIANCE_INTERNAL_CONFIG_WRITE	在 `.opendistro_security` 索引中执行写入事件的请求。

您可以使用任何类别和消息属性组合。例如，如果您发送 REST 请求为文档编制索引，您可能会在审计日志中看到以下行：

- 在 REST 层上进行身份验证（身份验证）
- 传输层（授权）上的 GRANTED_PRIVILEGE
- COMPLIANCE_DOC_WRITE（文档写入索引）

审计日志设置

审计日志有许多配置选项。

常规设置

常规设置允许您启用或禁用单个类别或整个图层。强烈建议将授予的权限和身份验证保留为排除的类别。否则，对集群的每个有效请求都会记录这些类别。

名称	后端设置	描述
REST 层	enable_rest	启用或禁用 REST 层上发生的事件。
REST 禁用类别	disabled_rest_categories	指定要在 REST 层忽略的审计类别。修改这些类别可以大大增加审计日志的大小。
传输层	enable_transport	启用或禁用传输层上发生的事件。

名称	后端设置	描述
传输禁用类别	disabled_transport_categories	指定传输层上必须忽略的审计类别。修改这些类别可以大大增加审计日志的大小。

属性设置允许您自定义每个日志行中的详细信息量。

名称	后端设置	描述
批量请求	resolve_bulk_requests	启用此设置为批量请求中的每个文档生成一个日志，这可以大大增加审计日志的大小。
请求正文	log_request_body	包括请求的请求正文。
解决索引	resolve_indices	将别名解析为索引。

使用忽略设置排除一组用户或 API 路径：

名称	后端设置	描述
忽略的用户	ignore_users	指定要排除的用户。
忽略的请求	ignore_requests	指定要排除的请求模式。

合规性设置

合规性设置允许您优化索引、文档或字段级别的访问权限。

名称	后端设置	描述
合规性日志	enable_compliance	启用或禁用符合性日志记录。

您可以为读取和写入事件日志记录指定以下设置。

名称	后端设置	描述
内部配置日志记录	internal_config	启用或禁用 `.opendistro_security` 索引中的事件记录。

您可以为读取事件指定以下设置。

名称	后端设置	描述
读取元数据	read_metadata_only	仅包含读取事件的元数据。请勿包含任何文档字段。
忽略的用户	read_ignore_users	请勿为读取事件包含特定用户。
观看的字段	read_watched_fields	指定要监视读取事件的索引和字段。添加受监视字段会为每个文档访问生成一个日志，这可能会大大增加审计日志的大小。监视字段支持索引模式和字段模式： `{ "index-name-pattern": [ "field-name-pattern" ], "logs*": [ "message" ], "twitter": [ "id", "user*" ] }`

您可以为写入事件指定以下设置。

名称	后端设置	描述
写入元数据	write_metadata_only	仅包含写入事件的元数据。请勿包含任何文档字段。
日志差异	write_log_diffs	如果仅写入元数据为假，则仅包括写入事件之间的差异。
忽略的用户	write_ignore_users	请勿为写入事件包括某些用户。
观看索引	write_watched_indices	指定要监视写入事件的索引或索引模式。添加受监视字段会为每个文档访问生成一个日志，这可能会大大增加审计日志的大小。

审计日志示例

本节包括一个示例配置、搜索请求以及索引的所有读取和写入事件的结果审计日志。

步骤 1：配置审计日志

启用将审计日志发布到 CloudWatch 日志组后，导航到 OpenSearch 仪表板审核日志页面并选择启用审核日志。

1. 在常规设置中，选择配置并确保 REST 层处于启用状态。
2. 在合规性设置中，选择配置。
3. 在写入的下面，在观看的字段中将添加所有的写入事件的 `accounts` 添加添加到此索引。
4. 在 Read（读取）下的 Watched Fields（监控字段）中，添加 `accounts` 索引的 `ssn` 和 `id-` 字段：

```
{
  "accounts-": [
    "ssn",
    "id-"
  ]
}
```

步骤 2：执行读取和写入事件

1. 导航到 “ OpenSearch 控制面板”，选择 “开发工具”，然后为示例文档编制索引：

```
PUT accounts/_doc/0
{
  "ssn": "123",
  "id-": "456"
}
```

2. 要测试读取事件，请发送以下请求：

```
GET accounts/_search
{
  "query": {
    "match_all": {}
  }
}
```

步骤 3：观察日志

1. 打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择 日志组。
3. 选择您在启用审计日志时指定的日志组。在日志组中，S OpenSearch ervice 会为您域中的每个节点创建一个日志流。
4. 在日志流中，选择搜索全部。
5. 有关读取和写入事件，请参阅相应的日志。在日志出现之前，您可能会有 5 秒的延迟。

   示例写入审计日志

```
{
  "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"
  ]
}
```

示例读取审计日志

```
{
  "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"
  ]
}
```

要包含请求正文，请返回 OpenSearch 控制面板中的合规性设置并禁用 “写入元数据”。要排除某个具体用户的事件，请将该用户添加到忽略的用户。

有关每个审计日志字段的描述，请参阅审计日志字段参考。有关搜索和分析审核日志数据的信息，请参阅 Amazon Logs 用户指南中的使用 CloudWatch 日志见解分析 CloudWatch 日志数据。

使用 REST API 配置审计日志

我们建议使用 OpenSearch 仪表板来配置审计日志，但您也可以使用精细的访问控制 REST API。本节包含一个示例请求。有关 REST API 的完整文档可在OpenSearch 文档中找到。

```
PUT _opendistro/_security/api/audit/config
{
  "enabled": true,
  "audit": {
    "enable_rest": true,
    "disabled_rest_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "enable_transport": true,
    "disabled_transport_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "resolve_bulk_requests": true,
    "log_request_body": true,
    "resolve_indices": true,
    "exclude_sensitive_headers": true,
    "ignore_users": [
      "kibanaserver"
    ],
    "ignore_requests": [
      "SearchRequest",
      "indices:data/read/*",
      "/_cluster/health"
    ]
  },
  "compliance": {
    "enabled": true,
    "internal_config": true,
    "external_config": false,
    "read_metadata_only": true,
    "read_watched_fields": {
      "read-index-1": [
        "field-1",
        "field-2"
      ],
      "read-index-2": [
```

```
        "field-3"
      ]
    },
    "read_ignore_users": [
      "read-ignore-1"
    ],
    "write_metadata_only": true,
    "write_log_diffs": false,
    "write_watched_indices": [
      "write-index-1",
      "write-index-2",
      "log-*",
      "*"
    ],
    "write_ignore_users": [
      "write-ignore-1"
    ]
  }
}
```

使用 Amazon 监控 OpenSearch 服务事件 EventBridge

Amazon S OpenSearch ervice 与 Amazon 集成， EventBridge 以通知您影响您的域的特定事件。Amazon 服务 EventBridge 中的事件将近乎实时传输到。同样的事件也会发送到 Amazon 的前身 Amazon CloudWatch Events EventBridge。您可以编写简单规则来指示您关注的事件，并指示要在事件匹配规则时执行的自动化操作。可自动触发的操作包括：

- 调用函数 Amazon Lambda
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams
- 激活 Amazon Step Functions 状态机
- 通知 Amazon SNS 主题或 Amazon SQS 队列

有关更多信息，请参阅《亚马逊 EventBridge 用户指南》 EventBridge中的 “亚马逊入门”。

主题

- 服务软件更新事件
- 自动调整事件

- 集群运行状况事件
- VPC 端点事件
- 节点停用事件
- 性能降级节点停用事件
- 域错误事件
- 教程：侦听 Amazon S OpenSearch ervice EventBridge 事件
- 教程：发送有关可用软件更新的 Amazon SNS 警报

服务软件更新事件

OpenSearch 当以下任一服务软件更新事件发生 EventBridge 时，服务会将事件发送到。

可用服务软件更新

OpenSearch 服务软件更新可用。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
    "description": "Service software update R20220928 available. Service Software
 Deployment Mechanism:
                    Blue/Green. For more information on deployment configuration,
 please
                    see: https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
```

```
  }
}
```

已安排服务软件更新

OpenSearch 安排了服务软件更新后，服务会发送此事件。对于可选的更新，您会在计划日期收到通知，并且可以随时选择重新安排。对于必需的更新，您会在计划日期前的三天收到通知，并且可以在强制窗口中选择重新安排。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Scheduled",
    "severity": "High",
    "description": "A new service software update [R20200330-p1] has been scheduled at
 [21st May 2023 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
  }
}
```

已重新安排服务软件更新

OpenSearch 重新安排可选服务软件更新后，服务会发送此事件。有关更多信息，请参阅 the section called “可选更新与必需更新”。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Rescheduled",
    "severity": "High",
    "description": "The service software update [R20200330-p1], which was originally
 scheduled for
                    [21st May 2023 12:40 GMT], has been rescheduled to [23rd May 2023
 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
  }
}
```

服务软件更新已启动

OpenSearch 服务软件更新已启动。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
```

```
  "detail": {
    "event": "Service Software Update",
    "status": "Started",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] started.
  }
}
```

服务软件更新已完成

OpenSearch 服务软件更新已完成。

示例

以下是该类型的示例事件：

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

已取消服务软件更新

OpenSearch 取消了服务软件更新后，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
```

```
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Cancelled",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] has been
 cancelled as a
                    newer update is available. Please schedule the latest update."
  }
}
```

已取消计划服务软件更新

OpenSearch 先前为域安排的服务软件更新已取消后，服务会发送此事件。

示例

以下是该类型的示例事件：

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

未执行服务软件更新

OpenSearch 服务无法启动服务软件更新时发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Unexecuted",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] cannot be
 started. Reason: [reason]"
  }
}
```

服务软件更新失败

OpenSearch 服务软件更新失败时发送此事件。

示例

以下是该类型的示例事件：

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

已要求服务软件更新

OpenSearch 当需要更新服务软件时，服务会发送此事件。有关更多信息，请参阅 the section called “可选更新与必需更新”。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Required",
    "severity": "High",
    "description": "Service software update [R20200330-p1] available. Update
                    will be automatically installed after [21st May 2023] if no
                    action is taken. Service Software Deployment Mechanism: Blue/Green.
                    For more information on deployment configuration, please see:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
  }
}
```

自动调整事件

OpenSearch 当以下任一自动调整事件发生 EventBridge 时，服务会将事件发送到。

自动调整挂起

OpenSearch Service 发送此事件时，自动调整已确定改进集群性能和可用性的优化建议。您只能看到已禁用自动调整的域的此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Pending",
    "description": "Auto-Tune recommends the following new settings for your
 domain: { JVM Heap size : 60%}. Enable Auto-Tune to improve cluster stability and
 performance.",
    "scheduleTime": "{iso8601-timestamp}"
  }
}
```

自动调整已开始

OpenSearch Service 会在自动调整开始将新设置应用到您的域时发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
```

```
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Started",
    "scheduleTime": "{iso8601-timestamp}",
    "startTime": "{iso8601-timestamp}",
    "description" : "Auto-Tune is applying the following settings to your domain: { JVM
 Heap size : 60%}."
  }
}
```

自动调节需要计划蓝绿部署

OpenSearch Service 发送此事件时，自动调整已确定需要计划的蓝绿部署的优化建议。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
    "status": "Pending",
    "startTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has identified the following settings for your domain
 that require a blue/green deployment: { JVM Heap size : 60%}.
                    You can schedule the deployment for your preferred time."
  }
}
```

自动调整已取消

OpenSearch Service 会在自动调整计划被取消时发送此事件，因为没有待处理的优化建议。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
    "status": "Cancelled",
    "scheduleTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has cancelled the upcoming blue/green deployment."
  }
}
```

自动调整已完成

OpenSearch Service 发送此事件时，自动调整已完成蓝绿部署，并且集群在新 JVM 设置的情况下运行。

示例

以下是该类型的示例事件：

```
{
  "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%}."
  }
}
```

已禁用自动调整并恢复更改

OpenSearch Service 会在禁用自动调整并回滚应用的更改时发送此事件。

示例

以下是该类型的示例事件：

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

已禁用自动调整并保留更改

OpenSearch Service 会在禁用自动调整并保留应用的更改时发送此事件。

示例

以下是该类型的示例事件：

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

集群运行状况事件

OpenSearch Service 会在集群的运行状况受到影响 EventBridge 时发送某些事件。

红色集群恢复已开始

OpenSearch Service 会在集群状态持续变为红色超过一个小时后发送此事件。它会尝试从快照中自动还原一个或多个红色索引，以修复集群状态。

示例

以下是该类型的示例事件：

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

红色集群恢复部分完成

OpenSearch 在尝试修复红色集群状态期间，Service 仅在其只能从快照还原部分红色索引时，才会发送此事件。

示例

以下是该类型的示例事件：

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

红色集群恢复失败

OpenSearch Service 会在尝试修复红色集群状态时无法恢复任何索引，则会发送此事件。

示例

以下是该类型的示例事件：

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

要删除的分片

OpenSearch 当红色集群持续为红色 14 天后尝试自动修复红色集群状态时，如果仍有一个或多个索引为红色，则会发送此事件。再过 7 天（总共 21 天持续变为红色）之后，S OpenSearch ervice 将在所有红色索引上删除未分配的分片。

示例

以下是该类型的示例事件：

```
{
   "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"
   }
}
```

分片已删除

OpenSearch Service 会在集群状态持续变为红色 21 天之后发送此事件。它会删除所有红色索引上未分配的分片（存储和计算）。有关更多信息，请参阅 the section called “自动修复红色集群”。

示例

以下是该类型的示例事件：

```
{
   "version":"0",
```

```
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2022-04-09T10:54:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "severity":"High",
      "description":"We have deleted unassinged shards, the unit of storage and
 compute, in
                     red indices: index-1, index-2 because these indices were red for
 more than
                     21 days and could not be restored with the automated restore
 process.
                     Please refer to https://docs.aws.amazon.com/opensearch-service/
latest/developerguide/handling-errors.html#handling-errors-red-cluster-status for
 troubleshooting steps.",
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Shard(s) deleted"
   }
}
```

高分片数警告

OpenSearch 当热数据节点中的平均分片数超过建议的默认限制 1,000 的 90% 时，Service 将会发送此事件。尽管较新版本的 Elasticsearch OpenSearch 支持可配置最大分片数单个节点限制，但我们还是建议每个节点上拥有不超过 1,000 个分片。请参阅选择分片数。

示例

以下是该类型的示例事件：

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

超出分片数限制

OpenSearch 当热数据节点中的平均分片数超过建议的默认限制 1,000 时，Service 将会发送此事件。尽管较新版本的 Elasticsearch OpenSearch 支持可配置最大分片数单个节点限制，但我们还是建议每个节点上拥有不超过 1,000 个分片。请参阅选择分片数。

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High Shard Count",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more data nodes have more than 1000 shards. To ensure
 optimum performance and stability of your
                    cluster, please refer to the best practice guidelines - https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/sizing-domains.html#bp-
sharding."
  }
}
```

磁盘空间不足

OpenSearch Service 会在集群中有一个或多个节点的可用存储空间低于 25%（或少于 25GB）时发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Low Disk Space",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more data nodes in your cluster has less than 25% of storage
 space or less than 25GB.
                   Your cluster will be blocked for writes at 20% or 20GB. Please refer
 to the documentation for more information - https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/handling-errors.html#troubleshooting-cluster-block"
  }
}
```

低磁盘水位违例

OpenSearch Service 会在集群中所有节点的可用存储空间低于 10%（或少于 10 GB）时发送此事件。当所有节点都发生低磁盘水位违例时，任何新的索引都会生成一个黄色的集群，而当所有节点都低于高磁盘水位时，它将产生一个红色集群。

示例

以下是该类型的示例事件：

```
{
  "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 可爆发容量余额低于 70%

OpenSearch 当一个或多个数据节点上的 EBS 可爆发容量余额低于 70% 时发送此事件。如果 EBS 可爆发容量余额耗尽，会导致集群完全不可用和 I/O 请求节流，从而导致索引和搜索请求出现高延迟和超时。有关修复此问题的步骤，请参阅the section called “EBS 可爆发容量余额低”。

示例

以下是该类型的示例事件：

```
{
  "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 可爆发容量余额低于 20%

OpenSearch 当一个或多个数据节点上的 EBS 可爆发容量余额低于 20% 时发送此事件。如果 EBS 可爆发容量余额耗尽，会导致集群完全不可用和 I/O 请求节流，从而导致索引和搜索请求出现高延迟和超时。有关修复此问题的步骤，请参阅the section called “EBS 可爆发容量余额低”。

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"EBS Burst Balance",
     "status":"Warning",
     "severity":"High",
     "description":"EBS burst balance on one or more data nodes is below 20%.
                    Follow https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#handling-errors-low-ebs-burst
                    to fix this issue.
  }
}
```

磁盘吞吐量节流

OpenSearch Service 会在 EBS 卷或实例的吞吐量限制导致对域的读取和写入请求被节流时发送此事件。 EC2 如果您收到此通知，请考虑根据 Amazon 建议的最佳实践纵向扩展卷或实例。如果您的卷类型是 gp2，请增加卷大小。如果您的卷类型是 gp3，请预调配更多吞吐量。您还可以检查您的实例基础和最大 EBS 吞吐量是否大于或等于预调配的卷吞吐量，并可以相应扩展。

示例

以下是该类型的示例事件：

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

分片大小较大

OpenSearch 当集群中的一个或多个分片超过 50GiB 或 65 GiB 时，服务会发送此事件。为确保最佳集群性能和稳定性，请减少分片大小。

有关更多信息，请参阅分片最佳实践。

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
```

```
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Large Shard Size",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more shards are larger than 65GiB. To ensure optimum cluster
 performance and stability, reduce shard sizes.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-large-shard-
size."
  }
}
```

高 JVM 使用率

OpenSearch Service 会在域的`JVMMemoryPressure`指标超过 80% 时发送此事件。如果在 30 分钟内超过 92%，则对集群的所有写入操作都将被阻止。为确保最佳的集群稳定性，请减少集群流量或扩展域，以便为工作负载提供足够的内存。

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High JVM Usage",
     "status":"Warning",
     "severity":"High",
     "description":"JVM memory pressure has exceeded 80%. If it exceeds 92% for 30
 minutes, all write operations to your cluster
                    will be blocked. To ensure optimum cluster stability, reduce
 traffic to the cluster or use larger instance types.
```

```
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-high-jvm."
  }
}
```

GC 不足

OpenSearch Service 会在最大 JVM 大于 70%，且最大值和最小值之差小于 30% 时发送此事件。这可能表明 JVM 无法在垃圾回收周期中回收足够的内存来处理您的工作负载。这可能会导致响应速度越来越慢和延迟增加；在某些情况下，甚至会因为运行状况检查超时而导致节点掉线。为确保最佳的集群稳定性，请减少集群流量或扩展域，以便为工作负载提供足够的内存。

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Insufficient GC",
     "status":"Warning",
     "severity":"Medium",
     "description":"Maximum JVM is above 70% and JVM range is less than 30%. This may
 indicate insufficient garbage collection for your workload.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-insufficient-
gc."
  }
}
```

自定义索引路由警告

OpenSearch 当您的域处于处理状态时，Service 会发送此事件，并且包含带有自定义 index.routing.allocation 设置的索引，这可能会导致蓝绿部署停滞。验证设置是否正确应用。

示例

以下是该类型的示例事件：

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

分片锁定失败

OpenSearch Service 会在由于未分配的分片导致您的域无法正常运行时发送此事件。[ShardLockObtainFailedException]有关更多信息，请参阅如何解决 Ama OpenSearch zon Service 中内存分片锁定异常？

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
```

```
  "detail":{
     "event":"Failed Shard Lock",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is unhealthy due to unassigned shards with
 [ShardLockObtainFailedException]. For more information,
                    see https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/monitoring-events.html#monitoring-events-failed-shard-lock."
}
```

VPC 端点事件

OpenSearch 服务将某些事件发送到 EventBridge 与Amazon PrivateLink 接口端点相关的。

VPC 端点创建失败

OpenSearch 服务无法创建请求的 VPC 端点时，它将发送此事件。发生此错误可能是因为您已达到针对某一区域内允许的 VPC 端点数量的限制。如果指定子网或安全组不存在，您也会看到此错误。

示例

以下是该类型的示例事件：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Create Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to create VPC endpoint aos-0d4c74c0342343 for domain
                    arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: You've reached the limit on the
                    number of VPC endpoints that you can create in the AWS Region."
   }
```

```
}
```

VPC 端点更新失败

OpenSearch 服务无法删除请求的 VPC 端点时，它将发送此事件。

示例

以下是该类型的示例事件：

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Update Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to update VPC endpoint aos-0d4c74c0342343 for domain
                     arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: <failure message>."
   }
}
```

VPC 端点删除失败

OpenSearch 服务无法删除请求的 VPC 端点时，它将发送此事件。

示例

以下是该类型的示例事件：

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

节点停用事件

OpenSearch 当以下任一节点停用事件发生 EventBridge 时，服务会将事件发送到。

已计划停用节点

OpenSearch Service 会在计划节点停用时发送此事件。

示例

以下是该类型的示例事件：

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

已完成节点停用

OpenSearch Service 会在完成节点停用时发送此事件。

示例

以下是该类型的示例事件：

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

节点停用失败

OpenSearch Service 会在节点停用失败时发送此事件。

示例

以下是该类型的示例事件：

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

性能降级节点停用事件

OpenSearch 由于节点上的硬件性能降级而需要更换节点时，Service 会发送这些事件。

性能降级节点停用通知

OpenSearch 为您的域计划了停用并更换性能降级节点的自动化操作后，Service 会发送此事件。

示例

以下是该类型的示例事件：

```
{
   "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"
   }
}
```

性能降级节点停用完成

OpenSearch 性能降级的节点已经停用并用新的节点更换后，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
   "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"
   }
}
```

性能降级节点停用失败

OpenSearch 性能降级节点停用失败时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
   "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"
   }
}
```

域错误事件

OpenSearch 当以下任一域错误发生 EventBridge 时，服务会将事件发送到。

域更新验证故障

OpenSearch 在尝试更新或执行域配置更改时，如果遇到一个或多个验证故障，则会发送此事件。要获取解决这些故障的步骤，请参阅the section called “对验证错误进行故障排除”。

示例

以下是该类型的示例事件：

```
{
   "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 密钥不可访问

OpenSearch Service 会在无法访问您的 Amazon KMS 密钥时发送此事件。

示例

以下是该类型的示例事件：

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

域隔离

OpenSearch 当您的域被隔离并且由于网络无法访问而无法接收、读取或写入请求时，Service 将会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2023-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Domain Isolation Notification",
     "status":"Error",
     "severity":"High",
     "description":"Your OpenSearch Service domain has been isolated. An isolated
 domain is unreachable by network and cannot receive, read, or write requests. For more
 information and assistance, please contact AWS Support at https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/encryption-at-rest.html#disabled-key."
  }
}
```

教程：侦听 Amazon S OpenSearch ervice EventBridge 事件

在本教程中，您将设置一个简单的函数，该 Amazon Lambda 函数会侦听 Amazon S OpenSearch ervice 事件并将这些事件写出到 Logs CloudWatch 日志流。

先决条件

本教程假定您已经有一个现有 OpenSearch 服务域。如果您尚未创建域，请按照 创建和管理域 中的步骤创建一个。

第 1 步：创建 Lambda 函数

在此过程中，您将创建一个简单的 Lambda 函数来充当 Serv OpenSearch ice 事件流消息的目标。

创建目标 Lambda 函数

1. 打开 Amazon Lambda 控制台，网址为https://console.aws.amazon.com/lambda/。

2. 依次选择 Create function 和 Author from scratch。
3. 对于函数名称，请输入 event-handler。
4. 对于运行时系统，选择 Python 3.8。
5. 选择创建函数。
6. 在 Function code 部分中，编辑示例代码以匹配以下示例：

```
import json

def lambda_handler(event, context):
    if event["source"] != "aws.es":
        raise ValueError("Function only supports input from events with a source
 type of: aws.es")

    print(json.dumps(event))
```

   这是一个简单的 Python 3.8 函数，可输出 OpenSearch 服务发送的事件。如果一切配置正确，则在本教程结束时，您将在与此 Lambda 函数关联的 CloudWatch 日志流中看到事件详细信息。
7. 选择部署。

步骤 2：注册事件规则

在此步骤中，您将创建一个捕获 OpenSearch 服务域中的事件的 EventBridge 规则。该规则捕获来自定义该规则的账户中的所有事件。事件消息本身包含有关事件源的信息（包括事件源所在的域）。您可以使用此信息以编程方式过滤和排序事件。

创建 EventBridge 规则

1. 打开 EventBridge 控制台，网址为https://console.aws.amazon.com/events/。
2. 选择创建规则。
3. 将规则命名为 event-rule。
4. 选择下一步。
5. 对于事件模式，请选择Amazon 服务、Amazon OpenSearch 服务和所有事件。此模式适用于您的所有 OpenSearch 服务域以及每个 OpenSearch 服务事件。或者，您也可以创建一个更具体的模式来过滤掉一些结果。
6. 按 Next（下一步）。
7. 对于目标，选择 Lambda function（Lambda 函数）。在函数下拉菜单中，选择 event-handler。

8. 按 Next（下一步）。
9. 跳过标签，然后再次按 Next（下一步）。
10. 检查配置并选择 Create rule（创建规则）。

第 3 步：测试您的配置

下次您在通知部分，如果所有配置都正确完成，您的 Lambda 函数将触发并将事件数据写入该函数的 Log CloudWatch s 日志流。 OpenSearch

测试配置

1. 打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格上，选择日志，然后为您的 Lambda 函数选择日志组（例如，/aws/lambda/event-handler）。
3. 选择日志流以查看事件数据。

教程：发送有关可用软件更新的 Amazon SNS 警报

在本教程中，您将配置一个 Amazon EventBridge 事件规则，该规则将捕获 Amazon SNS 中可用 OpenSearch 服务软件更新的通知，并通过 Amazon Simple Notification Service (Amazon SNS) 向您发送电子邮件通知。

先决条件

本教程假定您已经有一个现有 OpenSearch 服务域。如果您尚未创建域，请按照 创建和管理域 中的步骤创建一个。

步骤 1：创建并订阅 Amazon SNS 主题

在本教程中，您配置一个 Amazon SNS 主题来充当新事件规则的事件目标。

创建 Amazon SNS 目标

1. 从 v3/Home 打开 Amazon SNS 控制台。https://console.aws.amazon.com/sns/
2. 选择主题和创建主题。
3. 对于作业类型，请选择标准，并命名作业为 software-update。
4. 选择 创建主题。

5. 创建主题后，选择创建订阅。
6. 对于协议，选择电子邮件。对于端点，输入您当前有权访问的电子邮件地址，然后选择 创建订阅。
7. 检查您的电子邮件账户，并等待接收订阅确认电子邮件。在收到此电子邮件后，选择 确认订阅。

步骤 2：注册事件规则

接下来，注册一个仅捕获服务软件更新事件的事件规则。

创建事件规则

1. 打开 EventBridge 控制台，网址为https://console.aws.amazon.com/events/。
2. 选择创建规则。
3. 将规则命名为 softwareupdate-rule。
4. 选择下一步。
5. 对于事件模式，请选择Amazon 服务、亚马逊 OpenSearch 服务和亚马逊 OpenSearch 服务软件更新通知。此模式匹配 Service 中的任何 OpenSearch 服务软件更新事件。有关事件模式的更多信息，请参阅《亚马逊 EventBridge用户指南》中的亚马逊 EventBridge 事件模式。
6. 您还可以进行筛选以仅针对特定的严重性。有关每个事件的严重性，请参阅the section called “服务软件更新事件”。
7. 选择下一步。
8. 对于目标，请选择 SNS topic（SNS 主题），然后选择 software-update。
9. 选择下一步。
10. 跳过标签，然后选择 Next（下一步）。
11. 检查规则配置并选择 Create rule（创建规则）。

下次您从 Service 收到有关可用 OpenSearch 服务软件更新的通知时，如果所有内容都配置正确，Amazon SNS 应向您发送有关此更新的电子邮件警报。

使用监控亚马逊 OpenSearch 服务 API 调用 Amazon CloudTrail

Amazon Service 与 Amazon CloudTrail一项 OpenSearch 服务集成，该服务可记录用户、角色或 Amazon 服务在 OpenSearch 服务中采取的操作。 CloudTrail 将 OpenSearch 服务的所有配置 API 调用捕获为事件。

> **Note**
>
> CloudTrail 仅捕获对配置 API 的调用，例如`CreateDomain`和`GetUpgradeStatus`。CloudTrail 不会捕获对的呼叫 OpenSearch APIs，例如`_search`和`_bulk`。有关这些调用，请参阅 the section called “监控审计日志”。

捕获的呼叫包括来自 OpenSearch 服务控制台或 Amazon SDK 的呼叫。 Amazon CLI如果您创建跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 OpenSearch 服务事件。如果您未配置跟踪，您仍然可以在 CloudTrail 主机上的事件历史记录中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向 OpenSearch 服务部门发出的请求、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅《Amazon CloudTrail 用户指南》。

中的亚马逊 OpenSearch 服务信息 CloudTrail

CloudTrail 在您创建账户 Amazon Web Services 账户 时已在您的账户上启用。当 OpenSearch 服务中发生活动时，该活动会与其他 Amazon 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在 Amazon Web Services 账户 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用事件历史查看 CloudTrail 事件。

要持续记录您的 Amazon Web Services 账户 账户中的事件，包括 OpenSearch 服务事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 Amazon Web Services 区域。跟踪记录 Amazon 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 Amazon 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 为您创建路线 Amazon Web Services 账户
- Amazon 与日志的服务集成 CloudTrail
- 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个区域的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail日志文件

所有 OpenSearch 服务配置 API 操作均由《亚马逊 OpenSearch 服务 API 参考》记录 CloudTrail 并记录在案。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户证书还是 Amazon Identity and Access Management (IAM) 用户凭证发出
- 请求是使用角色还是联合用户的临时安全凭证发出的
- 请求是否由其他 Amazon 服务发出

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解亚马逊 OpenSearch 服务日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序出现。

以下示例显示了演示该`CreateDomain`操作的 CloudTrail 日志条目：

```
{
  "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 服务中的安全

云安全 Amazon 是重中之重。作为 Amazon 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方 Amazon 的共同责任。责任共担模式将其描述为云的 安全性和云中 的安全性：

- 云安全 — Amazon 负责保护在 Amazon 云中运行 Amazon 服务的基础架构。 Amazon 还为您提供可以安全使用的服务。作为 Amazon 合规性计划的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用于亚马逊 OpenSearch 服务的合规计划，请参阅合规计划范围内的Amazon 服务。
- 云端安全-您的责任由您使用的 Amazon 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 OpenSearch 服务时如何应用分担责任模型。以下主题向您展示如何配置 OpenSearch 服务以满足您的安全和合规性目标。您还将学习如何使用其他 Amazon 服务来帮助您监控和保护您的 OpenSearch 服务资源。

主题

- 亚马逊 OpenSearch 服务中的数据保护
- 亚马逊 OpenSearch 服务中的身份和访问管理
- 防止跨服务混淆代理
- Amazon 服务中的精细访问控制 OpenSearch
- Amazon OpenSearch 服务的合规性验证
- Amazon S OpenSearch ervice 中的能力
- 亚马逊 OpenSearch 服务的 JWT 身份验证和授权
- Amazon S OpenSearch ervice 中的基础设施安全性
- 仪表板的 SAML 身份验证 OpenSearch
- 为控制面板配置 Amazon Cognito 身份验证 OpenSearch
- 使用适用于 Amazon Service OpenSearch 的服务相关角色

亚马逊 OpenSearch 服务中的数据保护

分 Amazon 分担责任模型适用于亚马逊 OpenSearch 服务中的数据保护。如本模型所述 Amazon ，负责保护运行所有内容的全球基础架构 Amazon Web Services 云。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 Amazon Web Services 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。

出于数据保护目的，我们建议您保护 Amazon Web Services 账户 凭证并使用 Amazon IAM Identity Center 或 Amazon Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 用于 SSL/TLS 与 Amazon 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 Amazon CloudTrail。有关使用 CloudTrail 跟踪捕获 Amazon 活动的信息，请参阅《Amazon CloudTrail 用户指南》中的使用跟 CloudTrail 踪。
- 使用 Amazon 加密解决方案以及其中的所有默认安全控件 Amazon Web Services 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 Amazon 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、 Amazon CLI API 或 Amazon Web Services 服务 使用 OpenSearch 服务或其他服务时 Amazon SDKs。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

为 Amazon OpenSearch 服务加密静态数据

OpenSearch 服务域提供静态数据加密，这是一项有助于防止未经授权访问您的数据的安全功能。该功能使用 Amazon Key Management Service (Amazon KMS) 来存储和管理您的加密密钥，并使用 256 位密钥的高级加密标准算法 (AES-256) 来执行加密。如果启用，该功能会对域的以下方面进行加密：

- 所有索引（包括 UltraWarm 存储中的索引）
- OpenSearch 日志

- 交换文件
- 应用程序目录中的所有其他数据
- 自动快照

以下内容在您启用静态数据加密时不会 加密，但您可以执行额外的步骤来保护它们：

- 手动快照：您目前无法使用 Amazon KMS 密钥加密手动快照。但是，您可以使用 S3 托管密钥的服务器端加密或 KMS 密钥对您用作快照存储库的存储桶进行加密。有关说明，请参阅the section called “注册手动快照存储库”。
- 慢日志和错误日志：如果您发布日志并想要对其进行加密，则可以使用与 OpenSearch 服务域相同的 Amazon KMS 密钥加密其 CloudWatch 日志组。有关更多信息，请参阅 Amazon CloudWatch 日志用户指南 Amazon Key Management Service中的使用加密 CloudWatch 日志中的日志数据。

> **Note**
>
> 如果已在现有域上启用冷存储， UltraWarm 则无法在该域上启用静态加密。您必须先禁用 UltraWarm 或冷存储，启用静态加密，然后重新启用 UltraWarm 或冷存储。如果要将索引保留在 UltraWarm 或冷存储中，则必须在禁用索引 UltraWarm 或冷存储之前将其移至热存储。

OpenSearch 服务仅支持对称加密 KMS 密钥，不支持非对称密钥。要了解如何创建对称密钥，请参阅Amazon Key Management Service 开发人员指南中的创建 KMS 密钥。

无论是否启用静态加密，所有域都会使用 AES-256 和 OpenSearch 服务管理的密钥自动加密自定义软件包。

权限

要使用 OpenSearch 服务控制台配置静态数据的加密，您必须具有读取权限 Amazon KMS，例如以下基于身份的策略：

JSON

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

```
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

如果要使用 Amazon 自有密钥以外的密钥，则还必须有权为该密钥创建授权。这些权限通常采用基于资源的策略的形式，您在创建密钥时会指定该策略。

如果您想将密钥保留为 OpenSearch 服务专用，则可以在该密钥策略中添加 kms: ViaService 条件：

```
"Condition": {
  "StringEquals": {
    "kms:ViaService": "es.us-west-1.amazonaws.com"
  },
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
```

有关更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的在 Amazon KMS中使用密钥策略。

启用静态数据加密

对新域名上的静态数据进行加密需要 OpenSearch 或 Elasticsearch 5.1 或更高版本。在现有域名上启用它需要或 Elasticsearch 6.7 OpenSearch 或更高版本。

启用静态数据控制台加密

1. 在 Amazon 控制台中打开该域，然后选择操作和编辑安全配置。
2. 在 Encryption（加密）下，选择 Enable encryption of data at rest（启用静态数据加密）。
3. 选择要使用的 Amazon KMS 密钥，然后选择保存更改。

此外，您还可以通过配置 API 启用加密。以下请求允许对现有域中的静态数据进行加密：

```
{
   "ClusterConfig":{
      "EncryptionAtRestOptions":{
         "Enabled": true,
         "KmsKeyId":"arn:aws:kms:us-east-1:123456789012:alias/my-key"
      }
   }
}
```

已禁用或已删除 KMS 密钥

如果您禁用或删除用于加密域名的密钥，则该域将无法访问。 OpenSearch 服务会向您发送通知，通知您它无法访问 KMS 密钥。立即重新启用密钥以访问您的域。

如果您的密钥被删除， OpenSearch 服务团队将无法帮助您恢复数据。 Amazon KMS 只有在等待至少七天后才会删除密钥。如果您的密钥正在等待删除，请取消删除或拍摄手动快照，以防止数据丢失。

禁用静态数据加密

在您配置一个域以对静态数据进行加密后，您无法禁用该设置。相反，您可以随时拍摄现有域的手动快照，创建另一个域，迁移您的数据和删除旧域。

监控对静态数据进行加密的域

对静态数据进行加密的域有两个额外指标：`KMSKeyError` 和 `KMSKeyInaccessible`。仅当域发现您的加密密钥存在问题时，才会显示这些指标。有关这些指标的完整说明，请参阅the section called “集群指标”。您可以使用 OpenSearch 服务控制台或 Amazon CloudWatch 控制台查看它们。

> **Tip**
>
> 每个指标都代表一个域的重大问题，因此我们建议您为这两个指标创建 CloudWatch 警报。有关更多信息，请参阅 the section called “推荐的 CloudWatch 警报”。

其他考虑因素

- 自动密钥轮换会保留 Amazon KMS 密钥的属性，因此轮换不会影响您访问 OpenSearch 数据的能力。加密 OpenSearch 服务域不支持手动密钥轮换，这包括创建新密钥和更新对旧密钥的所有引用。要了解更多信息，请参阅《Amazon Key Management Service 开发者指南》中的轮换 Amazon KMS 密钥。

- 某些实例类型不支持静态数据加密。有关更多信息，请参阅 the section called “支持的实例类型”。
- 对静态数据进行加密的域对其自动快照使用了不同的存储库名称。有关更多信息，请参阅 the section called “还原快照”。
- 虽然我们强烈建议启用静态加密，但它可能会增加额外的 CPU 开销和几毫秒的延迟。但是，大多数使用案例对这些差异并不敏感，影响的程度取决于集群、客户端和使用情况配置文件的配置。

Node-to-node 亚马逊 OpenSearch 服务加密

Node-to-node 加密在 Amazon OpenSearch 服务的默认功能之外还提供了一层额外的安全保护。

每个 OpenSearch 服务域（无论该域是否使用 VPC 访问权限）都位于自己的专用 VPC 中。这种架构可以防止潜在的攻击者拦截 OpenSearch 节点之间的流量，并确保集群的安全。但是，默认情况下，VPC 内的流量是未加密的。 Node-to-node加密为 VPC 内的所有通信启用 TLS 1.2 加密。

如果您通过 HTTPS 将数据发送到 S OpenSearch ervice，则 node-to-node加密有助于确保您的数据在整个集群中 OpenSearch 分发（和重新分发）时保持加密状态。如果数据通过 HTTP 未加密到达，则 OpenSearch 服务会在数据到达集群后对其进行加密。您可以使用控制台、 Amazon CLI或配置 API 要求所有进入该域的流量都通过 HTTPS 到达。

Node-to-node 如果启用精细访问控制，则需要加密。

启用 node-to-node加密

Node-to-node 对新域名进行加密需要任何版本或 Elasticsearch 6.0 或更高版本。 OpenSearch在现有域上启用 node-to-node加密需要任何版本的 Elasticsearch 6.7 或更高版本。 OpenSearch在 Amazon 控制台中选择现有域、Actions（操作）和 Edit security configuration（编辑安全配置）。

或者，您可以使用 Amazon CLI 或配置 API。有关更多信息，请参阅Amazon CLI 命令参考和OpenSearch 服务 API 参考。

禁用 node-to-node加密

将域配置为使用 node-to-node加密后，您无法禁用该设置。相反，您可以为加密域拍摄手动快照，创建另一个域，迁移您的数据和删除旧域。

亚马逊 OpenSearch 服务中的身份和访问管理

Amazon OpenSearch 服务提供了多种方法来控制对您的域名的访问权限。本主题介绍了各种策略类型，其彼此交互的方式，以及如何创建您自己的自定义策略。

> **Important**
>
> VPC 支持为 OpenSearch 服务访问控制引入了一些额外的注意事项。有关更多信息，请参阅 the section called “关于 VPC 域的访问策略”。

策略的类型

OpenSearch 服务支持三种类型的访问策略：

- the section called “基于资源的策略”
- the section called “基于身份的策略”
- the section called “基于 IP 的策略”

基于资源的策略

您可以在创建域时添加基于资源的策略（通常称为域访问策略）。这些策略指定主体可以对域的子资源执行哪些操作（跨集群搜索除外）。子资源包括 OpenSearch 索引和。 APIsIAM 中的委托人 JSON 策略元素指定了允许访问的账户、用户或角色。资源 JSON 策略元素指定这些委托人可以访问哪些子资源。

例如，以下基于资源的策略将授予 `test-user` 针对 `test-domain` 上的子资源的完全访问权限 (`es:*`)：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:*"
```

```
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

两个重要注意事项适用于此策略：

- 这些权限仅适用于此域。除非您在其他域上创建类似的策略，否则 `test-user` 只能访问 `test-domain`。
- `Resource` 元素中的尾随 `/*` 非常重要，并表示基于资源的策略仅适用于域的子资源，而不适用于域本身。在基于资源的策略中，`es:*` 操作等同于 `es:ESHttp*`。

  例如，`test-user` 可以向索引 (`GET https://search-test-domain.us-west-1.es.amazonaws.com/test-index`) 发送请求，但不可以更新域的配置 (`POST https://es.us-west-1.amazonaws.com/2021-01-01/opensearch/domain/test-domain/config`)。注意两个终端节点之间的差异。访问配置 API 需要一个基于身份的策略。

您可以通过添加通配符来指定部分索引名称。此示例将识别任何以 `commerce` 开头的索引：

```
arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce*
```

在这种情况下，该通配符意味着 `test-user` 可以请求 `test-domain` 中名称以 `commerce` 开头的索引。

要进一步限制 `test-user`，您可以应用以下策略：

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"
    }
  ]
}
```

现在，`test-user` 只能执行一项操作：搜索 `commerce-data` 索引。域中的所有其他索引均不可访问，且如果没有使用 `es:ESHttpPut` 或 `es:ESHttpPost` 操作的权限，`test-user` 将无法添加或修改文档。

接下来，您可以决定为高级用户配置角色。此策略允许索引 URIs中的所有人`power-user-role`访问 HTTP GET 和 PUT 方法：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/power-user-role"
        ]
      },
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPut"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce-
data/*"
    }
  ]
}
```

如果您的域位于 VPC 中或使用精细访问控制，您可以使用开放域访问策略。否则，您的域访问策略必须包含一些限制，无论是按委托人还是按 IP 地址。

有关所有可用操作的信息，请参阅the section called “策略元素参考”。要对数据进行更精细的控制，请使用具有精细访问控制权限的开放域访问策略。

基于身份的策略

与作为每个 OpenSearch 服务域一部分的基于资源的策略不同，您可以使用 Amazon Identity and Access Management (IAM) 服务将基于身份的策略附加到用户或角色。与基于资源的策略一样，基于身份的策略指定谁可以访问服务，他们可以执行哪些操作，以及他们可以对哪些资源执行这些操作（如果适用）。

虽然不一定，但基于身份的策略往往更通用。它们通常仅控制用户可执行的配置 API 操作。制定这些策略后，您可以在 Service 中使用基于资源的策略（或精细的访问控制）为用户 OpenSearch 提供对索引和的访问 OpenSearch 权限。 APIs

> **Note**
>
> 使用 Amazon 托管`AmazonOpenSearchServiceReadOnlyAccess`策略的用户无法在控制台上看到集群运行状况。要允许他们查看集群运行状况（和其他 OpenSearch 数据），请将`es:ESHttpGet`操作添加到访问策略并将其附加到他们的账户或角色。

由于基于身份的策略附加到用户或角色 (委托人)，JSON 不会指定委托人。以下策略授予对以 `Describe` 和 `List` 开头的操作的访问权限。这种操作组合提供对域配置的只读访问权限，但不提供对域本身中存储的数据的访问权限：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:Describe*",
        "es:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

管理员可能拥有对 OpenSearch 服务以及存储在所有域中的所有数据的完全访问权限：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

借助基于身份的策略，您可以使用标签来控制对配置 API 的访问。例如，如果域具有 `team:devops` 标签，以下策略允许附加的委托人查看和更新域的配置：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:UpdateDomainConfig",
      "es:DescribeDomain",
      "es:DescribeDomainConfig"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/team": [
          "devops"
        ]
```

```
      }
    }
  }]
}
```

您还可以使用标签来控制对 OpenSearch API 的访问权限。 OpenSearch API 基于标签的策略仅适用于 HTTP 方法。例如，如果域名带有`environment:production`标签，则以下策略允许附加的委托人向 OpenSearch API 发送 GET 和 PUT 请求：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:ESHttpGet",
      "es:ESHttpPut"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

要对 OpenSearch API 进行更精细的控制，可以考虑使用精细的访问控制。

> **Note**
>
> OpenSearch APIs 向任何基于标签的策略添加一个或多个标签后，必须执行单个标签操作（例如添加、删除或修改标签），更改才能在域上生效。您必须使用服务软件 R20211203 或更高版本才能在基于标签的策略中包含 OpenSearch API 操作。

OpenSearch 服务支持配置 API 的`RequestTag`和`TagKeys`全局条件密钥，而不是 OpenSearch API。这些条件仅适用于在请求中包含标签的 API 调用，例如 `CreateDomain`、`AddTags` 和`RemoveTags`。以下策略允许附加的委托人创建域，但仅当它们在请求中包含`team:it` 标签时：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "es:CreateDomain",
      "es:AddTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/team": [
          "it"
        ]
      }
    }
  }
}
```

有关使用标签进行访问控制的更多信息以及基于资源的策略和基于身份的策略之间的区别，请参阅 IAM 用户指南中的基于属性使用 ABAC 授权定义权限。

基于 IP 的策略

基于 IP 的策略将对域的访问限制在一个或多个 IP 地址或 CIDR 块。从技术上讲，基于 IP 的策略不是一种不同类型的策略。相反，它们只是基于资源的策略，它们指定匿名委托人并包含特殊条件。有关信息，请参阅 IAM 用户指南中的 IAM JSON 策略元素：条件。

基于 IP 的策略的主要吸引力在于它们允许向 OpenSearch 服务域发出未签名的请求，从而允许您使用 curl 和 OpenSearch Dashboards 等客户端，或者通过代理服务器访问该域。要了解更多信息，请参阅the section called “使用代理从仪表板访问 OpenSearch 服务”。

> **Note**
>
> 如果您为您的域启用了 VPC 访问，则无法配置基于 IP 的策略。相反，您可以使用`security groups`来控制哪些 IP 地址可以访问该域。有关更多信息，请参阅以下主题：
>
> - the section called “关于 VPC 域的访问策略”
> - 使用 Amazon VPC 用户指南中的安全组控制您的 Amazon 资源的流量

以下策略向源自指定 IP 范围的所有 HTTP 请求授予对 `test-domain` 的访问权限：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

如果您的域具有公共终端节点，并且不使用精细访问控制，我们建议您将 IAM 委托人和 IP 地址结合使用。此策略仅在请求源自指定的 IP 范围时向 `test-user` 授予 HTTP 访问权限：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::987654321098:user/test-user"
      ]
    },
    "Action": [
      "es:ESHttp*"
    ],
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24"
        ]
      }
    },
    "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
  }]
}
```

提出和签署 OpenSearch 服务请求

即使您配置了完全开放的基于资源的访问策略，对 OpenSearch 服务配置 API 的所有请求也必须经过签名。如果您的策略指定了 IAM 角色或用户，则向的请求 OpenSearch APIs 也必须使用签 Amazon 名版本 4 进行签名。签名方法因 API 而异：

- 要调用 OpenSearch 服务配置 API，我们建议您使用其中一个Amazon SDKs。与创建和签署自己的请求相比，这 SDKs极大地简化了流程，可以为您节省大量时间。配置 API 终端节点采用以下格式：

  ```
  es.region.amazonaws.com/2021-01-01/
  ```

  例如，以下请求将对 `movies` 域进行配置更改，但您必须自行签名（不推荐）：

  ```
  POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/movies/config
  {
  ```

```
  "ClusterConfig": {
    "InstanceType": "c5.xlarge.search"
  }
}
```

如果您使用其中一个 SDKs，例如 Boto 3，SDK 会自动处理请求签名：

```
import boto3

client = boto3.client(es)
response = client.update_domain_config(
  DomainName='movies',
  ClusterConfig={
    'InstanceType': 'c5.xlarge.search'
  }
)
```

有关 Java 代码示例，请参阅the section called “使用 Amazon SDKs”。

- 要拨打电话 OpenSearch APIs，您必须签署自己的请求。它们 OpenSearch APIs 使用以下格式：

```
domain-id.region.es.amazonaws.com
```

例如，以下请求将在 `movies` 索引中搜索 thor：

```
GET https://my-domain.us-east-1.es.amazonaws.com/movies/_search?q=thor
```

> **Note**
>
> 该服务会忽略 URLs 为使用签名版本 4 签名的 HTTP POST 请求传入的参数。

当策略发生冲突时

当策略不一致或没有明确提到某个用户时，复杂性将提高。IAM 用户指南中的 IAM 工作原理提供了策略评估逻辑的简要摘要：

- 默认情况下，所有请求都将被拒绝。
- 显式允许将取代此默认设置。

- 显式拒绝将覆盖任何允许。

例如，如果基于资源的策略授予您对域子资源（ OpenSearch 索引或 API）的访问权限，但基于身份的策略拒绝您访问，则您的访问将被拒绝。如果某个基于身份的策略授予访问权限，基于资源的策略不指定您是否应具有访问权限，则您将被允许访问。请参阅以下交叉策略表，了解域子资源结果的完整摘要。

	在基于资源的策略中允许	在基于资源的策略中拒绝	在基于资源的策略中既不允许也不拒绝
Allowed in identity-based policy	允许	拒绝	允许
Denied in identity-based policy	拒绝	拒绝	拒绝
Neither allowed nor denied in identity-based policy	允许	拒绝	拒绝

策略元素参考

OpenSearch 服务支持 IAM 策略元素参考中的大多数策略元素，但除外`NotPrincipal`。下表显示了最常用的元素。

JSON 策略元素	摘要
`Version`	当前版本的策略语言为 `2012-10-17` 。所有访问策略均应指定该值。
`Effect`	此元素指定该语句是允许还是拒绝对特定操作的访问。有效值为 `Allow` 或 `Deny`。
`Principal`	此元素指定允许 Amazon Web Services 账户 或拒绝访问资源的或 IAM 角色或用户，可以采用多种形式：

JSON 策略元素	摘要
	• Amazon 账户：`"Principal":{"AWS": ["123456789012"]}` 或 `"Principal":{"AWS": ["arn:aws:iam::123456789012:root"]}` • IAM 用户：`"Principal":{"AWS": ["arn:aws:iam::123456789012:user/test-user"]}` • IAM 角色：`"Principal":{"AWS": ["arn:aws:iam::123456789012:role/test-role"]}` **Important** 指定 * 通配符会支持匿名访问域，我们不建议这样做，除非您添加一个基于 IP 的条件，使用 VPC 支持，或启用细粒度的访问控制。此外，还要仔细检查以下策略，来确认它们不会授予广泛访问权限： • 附加到关联 Amazon 主体（例如 IAM 角色）的基于身份的策略 • 附加到关联资源的基于 Amazon 资源的策略（例如 Amazon Key Management Service KMS 密钥）

JSON 策略元素	摘要

Action	OpenSearch 服务对 OpenSearch HTTP 方法使用ESHttp*操作。其余操作适用于配置 API。
	
	某些 es: 操作支持资源级权限。例如，您可以向用户授予删除一个特定域的权限，而无需向该用户授予删除任何 域的权限。其他操作仅适用于服务本身。例如，es:ListDomainNames 在单个域范围内没有意义，因此需要一个通配符。
	
	有关所有可用操作的列表以及它们是适用于域子资源 (test-domain/*)、域配置 () 还是仅适用于服务 (test-domain)，请参阅《服务授权参考》中的 Amazon S OpenSearch ervice 操作、资源和条件密钥 *
	基于资源 的策略与资源级 权限不同。基于资源的策略是附加到域的完整 JSON 策略。资源级权限可让您将操作限制于特定域或子资源。在实际工作中，您可以将资源级权限视为基于资源的策略或基于身份的策略的可选部分。
	
	虽然 es:CreateDomain 的资源级权限可能看起来不直观—究竟为何要向用户授予创建已存在的域的权限？但通配符的使用可让您强制实施一个简单的域命名架构，例如 "Resource": "arn:aws:es:us-west-1:987654321098:domain/my-team-name-*" 。
	
	当然，您完全可以在限制性较低的资源元素旁包括如下操作：
	
	JSON
	
	{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"es:ESHttpGet",
	"es:DescribeDomain"
	],
	"Resource": "*"
	}
	]

JSON 策略元素	摘要
	`}` 要了解有关配对操作和资源的更多信息，请参阅此表中的 `Resource` 元素。
`Condition`	OpenSearch 服务支持 IAM 用户指南中Amazon 全局条件上下文密钥中描述的大多数条件。值得注意的例外包括 OpenSearch 服务不支持的`aws:PrincipalTag` 密钥。 配置基于 IP 的策略时，您指定 IP 地址或 CIDR 块作为条件，如下所示： `"Condition": { "IpAddress": { "aws:SourceIp": [ "192.0.2.0/32" ] } }` 如中所the section called “基于身份的策略”述`aws:ResourceTag` ，`aws:RequestTag` 、和`aws:TagKeys` 条件键适用于配置 API 以及 OpenSearch APIs。

JSON 策略元素	摘要

`Resource`	OpenSearch 服务以三种基本方式使用`Resource`元素：
	• 对于应用于 S OpenSearch ervice 本身的操作（例如`es:ListDomainNames` 或允许完全访问权限），请使用以下语法：
	`"Resource": "*"`
	• 对于涉及域的配置的操作 (例如 `es:DescribeDomain`)，您可以使用以下语法：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name "`
	• 对于应用于域的子资源的操作 (例如 `es:ESHttpGet`)，您可以使用以下语法：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /*"`
	您不必使用通配符。 OpenSearch 服务允许您为每个 OpenSearch 索引或 API 定义不同的访问策略。例如，您可以限制用户对 `test-index` 索引的权限：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index"`
	与对 `test-index` 的完全访问权限相比，您可能更愿意将策略限制在仅搜索 API：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index/_search"`
	您甚至可以控制对单个文档的访问：
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index/test-type/1"`

JSON 策略元素	摘要
	本质上，如果将子资源 OpenSearch 表示为 URI，则可以使用访问策略控制对其的访问。有关对用户可访问的资源的更多控制，请参阅the section called “精细访问控制”。 有关哪些操作支持资源级权限的详细信息，请参阅此表中的 `Action` 元素。

高级选项和 API 注意事项

OpenSearch 服务有几个高级选项，其中一个涉及访问控制：`rest.action.multi.allow_explicit_index`。在其默认设置为 true 时，它允许用户在某些情况下绕过子资源权限。

例如，请考虑以下基于资源的策略：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/test-index/*",
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/_bulk"
      ]
    },
    {
      "Effect": "Allow",
```

```
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/
restricted-index/*"
    }
  ]
}
```

此政策授予对 OpenSearch 批量 API `test-index` 的`test-user`完全访问权限。它还允许对 GET 发送 `restricted-index` 请求。

正如您可能预料的，以下索引请求将因权限错误而失败：

```
PUT https://search-test-domain.us-west-1.es.amazonaws.com/restricted-index/movie/1
{
  "title": "Your Name",
  "director": "Makoto Shinkai",
  "year": "2016"
}
```

与索引 API 不同，批量 API 可让您在一次调用中创建、更新和删除许多文档。但是，您通常在请求正文中指定这些操作，而不是在请求 URL 中。由于 S OpenSearch ervice 用于 URLs 控制对域子资源的访问权限，因此实际上可以使用批量 API 对域子资源进行`restricted-index`更改。`test-user`即使该用户缺乏对索引的 POST 权限，以下请求也将成功：

```
POST https://search-test-domain.us-west-1.es.amazonaws.com/_bulk
{ "index" : { "_index": "restricted-index", "_type" : "movie", "_id" : "1" } }
{ "title": "Your Name", "director": "Makoto Shinkai", "year": "2016" }
```

在这种情况下，访问策略无法实现其意图。要防止用户绕过这些类型的限制，您可以将 `rest.action.multi.allow_explicit_index` 更改为 false。如果此值为 false，则对在请求正文中指定索引名称的批量、mget 和 msearch APIs 的所有调用都将停止工作。换句话说，对 `_bulk` 的调用不再有效，但对 `test-index/_bulk` 的调用仍然有效。这第二个终端节点包含一个索引名称，因此您无需在请求正文中指定一个索引名称。

OpenSearch 仪表板严重依赖于 mget 和 msearch，因此更改后不太可能正常运行。对于部分补救，您可以保留`rest.action.multi.allow_explicit_index`为真，拒绝某些用户访问其中一个或多个 APIs。

有关更改此设置的信息，请参阅the section called “高级集群设置”。

同样，以下基于资源的策略包含两个小问题：

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/*"
    }
  ]
}
```

- 即使显式拒绝，`test-user` 仍然可以进行 `GET https://search-test-domain.us-west-1.es.amazonaws.com/_all/_search` 和 `GET https://search-test-domain.us-west-1.es.amazonaws.com/*/_search` 等调用，以访问 `restricted-index` 中的文档。
- 由于 `Resource` 元素引用 `restricted-index/*`，`test-user` 无权直接访问索引的文档。但是，该用户有权删除整个索引。要防止访问和删除，策略必须改为指定 `restricted-index*`。

与混合广泛的允许和集中的拒绝相比，最安全的方法是遵循最小特权原则，仅授予执行任务所需的权限。有关控制对单个索引或 OpenSearch 操作的访问权限的更多信息，请参见the section called “精细访问控制”。

> **Important**
>
> 指定 * 通配符将启用对域的匿名访问。不建议使用通配符。此外，还要仔细检查以下策略，以确认不会授予广泛的访问权限：
>
> - 附加到关联 Amazon 委托人（例如，IAM 角色）的基于身份的策略
> - 附加到关联资源的基于 Amazon 资源的策略（例如 Amazon Key Management Service KMS 密钥）

配置访问策略

- 有关在 S OpenSearch ervice 中创建或修改基于资源和 IP 的策略的说明，请参阅the section called “配置访问策略”。
- 有关在 IAM 中创建或修改基于身份的策略的说明，请参阅 IA M 用户指南中的使用客户托管策略定义自定义 IAM 权限。

其他示例策略

尽管本章包含许多示例策略，但 Amazon 访问控制是一个复杂的主题，最好通过示例来理解。有关更多信息，请参阅 IAM 用户指南中 IAM 基于身份的策略示例。

亚马逊 OpenSearch 服务 API 权限参考

在设置访问控制时，您编写可附加到 IAM 身份的权限策略（基于身份的策略）。有关详细参考信息，请参阅《服务授权参考》中的以下主题：

- OpenSearch 服务的操作、资源和条件键。
- OpenSearch Ingestion 的操作、资源和条件键。

此参考包含有关可在 IAM policy 中使用哪些 API 操作的信息。它还包括您可以为其授予权限的 Amazon 资源，以及可用于精细访问控制的条件密钥。

您需要在策略的 `Action` 字段中指定操作、在策略的 `Resource` 字段中指定资源值、在策略的 `Condition` 字段中指定条件。要为 OpenSearch 服务指定操作，请使用`es:`前缀和 API 操作名称（例如`es:CreateDomain`）。要为 In OpenSearch gestion 指定操作，请使用`osis:`前缀和 API 操作（例如）。`osis:CreatePipeline`

Amazon 亚马逊 OpenSearch 服务的托管政策

Amazon 托管策略是由创建和管理的独立策略 Amazon。 Amazon 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， Amazon 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 Amazon 客户使用。我们建议通过定义特定于您的使用场景的客户管理型策略来进一步减少权限。

您无法更改 Amazon 托管策略中定义的权限。如果 Amazon 更新 Amazon 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 Amazon 最有可能在启动新的 API 或现有服务可以使用新 Amazon Web Services 服务 的 API 操作时更新 Amazon 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 Amazon 托管策略。

AmazonOpenSearchDirectQueryGlueCreateAccess

授予亚马逊 OpenSearch 服务直接查询服务访问`CreateDatabaseCreatePartition`、`CreateTable`、和`BatchCreatePartition` Amazon Glue API。

您可以在 IAM 控制台中找到该AmazonOpenSearchDirectQueryGlueCreateAccess策略。

AmazonOpenSearchServiceFullAccess

授予对的 OpenSearch 服务配置 API 操作和资源的完全访问权限 Amazon Web Services 账户。

您可以在 IAM 控制台中找到该AmazonOpenSearchServiceFullAccess策略。

AmazonOpenSearchServiceReadOnlyAccess

授予对所有 OpenSearch 服务资源的只读访问权限 Amazon Web Services 账户。

您可以在 IAM 控制台中找到该AmazonOpenSearchServiceReadOnlyAccess策略。

AmazonOpenSearchServiceRolePolicy

您不能将 `AmazonOpenSearchServiceRolePolicy` 附加到自己的 IAM 实体。此策略附加到允许服务访问账户资源的 OpenSearch 服务相关角色。有关更多信息，请参阅 the section called “权限”。

您可以在 IAM 控制台中找到该AmazonOpenSearchServiceRolePolicy策略。

AmazonOpenSearchServiceCognitoAccess

提供必要的最低 Amazon Cognito 权限，以便启用 Cognito 身份验证。

您可以在 IAM 控制台中找到该AmazonOpenSearchServiceCognitoAccess策略。

AmazonOpenSearchIngestionServiceRolePolicy

您不能将 `AmazonOpenSearchIngestionServiceRolePolicy` 附加到自己的 IAM 实体。此策略附加到服务相关角色，该角色允许 OpenSearch Ingestion 为摄取管道启用 VPC 访问权限、创建标签以及向您的账户发布与摄取 CloudWatch 相关的指标。有关更多信息，请参阅 the section called “使用服务相关角色”。

您可以在 IAM 控制台中找到该AmazonOpenSearchIngestionServiceRolePolicy策略。

OpenSearchIngestionSelfManagedVpcePolicy

您不能将 `OpenSearchIngestionSelfManagedVpcePolicy` 附加到自己的 IAM 实体。此策略附加到服务相关角色，该角色允许 OpenSearch Ingestion 为摄取管道启用自我管理 VPC 访问权限、创建标签以及向您的账户发布与摄取相关的指标。 CloudWatch 有关更多信息，请参阅 the section called “使用服务相关角色”。

您可以在 IAM 控制台中找到该OpenSearchIngestionSelfManagedVpcePolicy策略。

AmazonOpenSearchIngestionFullAccess

授予对 OpenSearch Ingestion API 操作和资源的完全访问权限。 Amazon Web Services 账户

您可以在 IAM 控制台中找到该AmazonOpenSearchIngestionFullAccess策略。

AmazonOpenSearchIngestionReadOnlyAccess

授予对所有 OpenSearch Ingestion 资源的只读访问权限。 Amazon Web Services 账户

您可以在 IAM 控制台中找到该AmazonOpenSearchIngestionReadOnlyAccess策略。

AmazonOpenSearchServerlessServiceRolePolicy

提供向其发送 OpenSearch 无服务器指标数据所需的最低 Amazon CloudWatch 权限。 CloudWatch

您可以在 IAM 控制台中找到该AmazonOpenSearchServerlessServiceRolePolicy策略。

OpenSearch Amazon 托管策略的服务更新

查看自该服务开始跟踪变更以来 OpenSearch 服务的 Amazon 托管策略更新的详细信息。

更改	描述	日期
更新了 `AmazonOpenSearchServiceRolePolicy`	在政策中添加了以下声明。当 Amazon S OpenSearch ervice 担任`AWSServiceRoleForAmazonOpenSearchService` 服务相关角色时，策略中的这一新声明 OpenSearch 允许更新仅由 OpenSearch管理的任何 Amazon IAM Identity Center 应用程序的访问范围。 `{ "Effect": "Allow", "Action": "sso:PutApplicationAccessScope", "Resource": "arn:aws:sso::*:application/*/*", "Condition": { "StringEquals": { "aws:ResourceOrgID": "${aws:PrincipalOrgID}" } } }`	2025 年 3 月 31 日
更新了 `AmazonOpenSearchServerlessServiceRolePolicy`	在策略 `AmazonOpenSearchServerlessServiceRolePolicy` 中添加了 Sid `AllowAOSS`	2024 年 7 月 12 日

更改	描述	日期
	`Cloudwatc hMetrics` 。Sid 是一种语句 ID，作为策略语句的可选标识符。	
新增了 `OpenSearchIngestio nSelfManagedVpcePo licy`	一项新政策，允许 OpenSearch Ingestion 为摄取管道启用自我管理 VPC 访问权限、创建标签以及向您的账户发布与 CloudWatch 摄取相关的指标。 有关策略 JSON，请参阅 IAM 控制台。	2024 年 6 月 12 日
新增了 `AmazonOpenSearchDi rectQueryGlueCreat eAccess`	授予亚马逊 OpenSearc h 服务直接查询服务访问`CreateDat abase` `CreatePar tition` 、`CreateTab le` 、和`BatchCrea tePartition` Amazon Glue API。	2024 年 5 月 6 日
更新了`AmazonOpenSearchSe rviceRolePolicy` 和 `AmazonElasticsearc hServiceRolePolicy`	添加了服务相关角色分配和取消分配 IPv6 地址所需的权限。 已弃用的 Elasticsearch 策略也已更新，以确保向后兼容。	2023 年 10 月 18 日

更改	描述	日期
新增了 `AmazonOpenSearchIngestionServiceRolePolicy`	一项新政策，允许 OpenSearch Ingestion 为摄取管道启用 VPC 访问权限、创建标签以及向您的账户发布与摄取相关的指标 CloudWatch 。 有关策略 JSON，请参阅 IAM 控制台。	2023 年 4 月 26 日
新增了 `AmazonOpenSearchIngestionFullAccess`	一项新政策，授予对 OpenSearch Ingestion API 操作和资源的完全访问权限。 Amazon Web Services 账户 有关策略 JSON，请参阅 IAM 控制台。	2023 年 4 月 26 日
新增了 `AmazonOpenSearchIngestionReadOnlyAccess`	一项新策略，授予对所有 OpenSearch Ingestion 资源的只读访问权限。 Amazon Web Services 账户 有关策略 JSON，请参阅 IAM 控制台。	2023 年 4 月 26 日
新增了 `AmazonOpenSearchServerlessServiceRolePolicy`	一项新策略，提供向其发送 OpenSearch 无服务器指标数据所需的最低权限。 Amazon CloudWatch 有关策略 JSON，请参阅 IAM 控制台。	2022 年 11 月 29 日

更改	描述	日期
更新了`AmazonOpenSearchServiceRolePolicy` 和 `AmazonElasticsearchServiceRolePolicy`	增加了服务相关角色创建OpenSearch 服务托管 VPC 终端节点所需的权限。某些操作只能在请求包含标签 `OpenSearchManaged=true` 时执行。 已弃用的 Elasticsearch 策略也已更新，以确保向后兼容。	2022 年 11 月 7 日
更新了`AmazonOpenSearchServiceRolePolicy` 和 `AmazonElasticsearchServiceRolePolicy`	增加了对该操作的支持，该`PutMetricData` 操作是向 Amazon 发布 OpenSearch 集群指标所必需的 CloudWatch。 已弃用的 Elasticsearch 策略也已更新，以确保向后兼容。 有关策略 JSON，请参阅 IAM 控制台。	2022 年 9 月 12 日
更新了`AmazonOpenSearchServiceRolePolicy` 和 `AmazonElasticsearchServiceRolePolicy`	增加了对 `acm` 资源类型的支持。该策略为服务相关角色提供了验证和验证 ACM 资源所需的最低 Amazon Certificate Manager (ACM) 只读权限，以便创建和更新启用了自定义终端节点的域。 已弃用的 Elasticsearch 策略也已更新，以确保向后兼容。	2022 年 7 月 28 日

更改	描述	日期
更新了AmazonOpenSearchServiceCognitoAccess 和 AmazonESCognitoAccess	增加了对该操作的支持，该UpdateUserPoolClient 操作是从 Elasticsearch 升级到期间设置 Cognito 用户池配置所必需的。OpenSearch 纠正了 SetIdentityPoolRoles 操作的权限，以允许访问所有资源。 已弃用的 Elasticsearch 策略也已更新，以确保向后兼容。	2021 年 12 月 20 日
更新了 AmazonOpenSearchServiceRolePolicy	增加了对 security-group 资源类型的支持。该策略提供了服务相关角色启用 VPC 访问所需的最低亚马逊 EC2 和 Elastic Load Balancing 权限。	2021 年 9 月 9 日
• 新增了 AmazonOpenSearchServiceFullAccess • 已弃用 AmazonESFullAccess	此新策略旨在取代旧策略。这两个策略都提供对 OpenSearch 服务配置 API 和所有 HTTP 方法的完全访问权限 OpenSearch APIs。精细访问权限和基于资源的策略仍然可以限制访问。	2021 年 9 月 7 日

更改	描述	日期
• 新增了 `AmazonOpenSearchServiceReadOnlyAccess` • 已弃用 `AmazonESReadOnlyAccess`	此新策略旨在取代旧策略。这两个策略都提供对 OpenSearch 服务配置 API（`es:Describe* es:List*`、和`es:Get*`）的只读访问权限，并且不提供对 HTTP 方法的访问权限 OpenSearch APIs。	2021 年 9 月 7 日
• 新增了 `AmazonOpenSearchServiceCognitoAccess` • 已弃用 `AmazonESCognitoAccess`	此新策略旨在取代旧策略。两个策略都提供了最低 Amazon Cognito 权限，以便启用 Cognito 身份验证。	2021 年 9 月 7 日
• 新增了 AmazonOpenSearchServiceRolePolicy • 已弃用 `AmazonElasticsearchServiceRolePolicy`	此新策略旨在取代旧策略。这两个策略都提供了服务相关角色启用 VPC 访问所需的最低 Amazon EC2 和 Elastic Load Balancing 权限。	2021 年 9 月 7 日
已开启跟踪更改	Amazon OpenSearch 服务现在可以跟踪 Amazon托管策略的更改。	2021 年 9 月 7 日

防止跨服务混淆代理

混淆代理问题是一个安全性问题，即不具有某操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 Amazon，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况， Amazon 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

我们建议在资源策略中使用aws:SourceArn和aws:SourceAccount全局条件上下文密钥来限制 Amazon S OpenSearch ervice 向该资源提供的其他服务的权限。如果 aws:SourceArn 值不包含账户 ID，例如 Amazon S3 存储桶 ARN，您必须使用两个全局条件上下文密钥来限制权限。如果同时使用全局条件上下文密钥和包含账户 ID 的 aws:SourceArn 值，则 aws:SourceAccount 值和 aws:SourceArn 值中的账户在同一策略语句中使用时，必须使用相同的账户 ID。如果您只希望将一个资源与跨服务访问相关联，请使用 aws:SourceArn。如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 aws:SourceAccount。

的值aws:SourceArn必须是 OpenSearch 服务域的 ARN。

防范混淆代理问题最有效的方法是使用 aws:SourceArn 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符 (*) 的 aws:SourceArn 全局上下文条件键。例如 arn:aws:*es*:*:*123456789012*:*。

以下示例显示了如何在 S OpenSearch ervice 中使用aws:SourceArn和aws:SourceAccount全局条件上下文键来防止出现混淆的副手问题。

JSON

```
{
   "Version":"2012-10-17",
   "Statement":{
      "Sid":"ConfusedDeputyPreventionExamplePolicy",
      "Effect":"Allow",
      "Principal":{
         "Service":"es.amazonaws.com"
      },
      "Action":"sts:AssumeRole",
      "Condition":{
         "StringEquals":{
            "aws:SourceAccount":"123456789012"
         },
         "ArnLike":{
            "aws:SourceArn":"arn:aws:es:region:123456789012:domain/my-domain"
         }
      }
   }
}
```

Amazon 服务中的精细访问控制 OpenSearch

精细的访问控制提供了更多方法来控制对您在 Ama OpenSearch zon Service 上的数据的访问权限。例如，根据请求发出者，您可能希望搜索仅返回一个索引中的结果。您可能想隐藏文档中的某些字段或完全排除某些文档。

精细访问控制提供了以下功能：

- 基于角色的访问控制
- 索引、文档和字段级别的安全性
- OpenSearch 仪表板多租户
- OpenSearch 和 OpenSearch 仪表板的 HTTP 基本身份验证

主题

- 大局：精细的访问控制和服务安全 OpenSearch
- 重要概念
- 关于主用户
- 启用精细访问控制
- 以主用户身份访问 OpenSearch 仪表板
- 管理权限
- 推荐配置
- 限制
- 修改主用户
- 其他主用户
- 手动快照
- 集成
- REST API 差异
- 教程：使用 IAM 主用户和 Amazon Cognito 身份验证配置域
- 教程：使用内部用户数据库和 HTTP 基本身份验证配置域

大局：精细的访问控制和服务安全 OpenSearch

Amazon OpenSearch 服务安全有三个主要层：

Network

第一个安全层是网络，它决定请求是否到达 OpenSearch 服务域。如果您在创建域时选择 Public access (公共访问)，则来自任何连接到 Internet 的客户端的请求都能到达域终端节点。如果您选择 VPC access (VPC 访问)，则客户端必须连接到 VPC（并且关联的安全组必须允许它）才能使请求到达终端节点。有关更多信息，请参阅 the section called “VPC 支持”。

域访问策略

第二个安全层是域访问策略。在请求到达域终端节点后，基于资源的访问策略允许或拒绝请求访问给定 URI。访问策略在请求本身到达 OpenSearch 前在域“边缘”接受或拒绝请求。

精细访问控制

第三个也是最后一个安全层是精细访问控制。在基于资源的访问策略允许请求到达域终端节点后，精细访问控制对用户凭证进行评估，并对用户进行身份验证或拒绝请求。如果精细访问控制对用户进行身份验证，它将获取映射到该用户的所有角色，并使用完整的权限集来确定如何处理请求。

> **Note**
>
> 如果基于资源的访问策略包含 IAM 角色或用户，则客户端必须使用签 Amazon 名版本 4 发送已签名的请求。因此，访问策略可能会与精细访问控制发生冲突，特别是在您使用内部用户数据库和 HTTP 基本身份验证时。您无法使用用户名和密码以及 IAM 凭证对请求进行签名。通常，如果您启用精细访问控制，我们建议您使用不需要已签名请求的域访问策略。

下列示意图说明了一个常见配置：启用了精细访问控制的 VPC 访问域、基于 IAM 的访问策略和 IAM 主用户。

下列示意图说明了另一个常见配置：启用了精细访问控制的公共访问域、不使用 IAM 委托人的访问策略以及内部用户数据库中的主用户。

示例

考虑向 `movies/_search?q=thor` 发出 GET 请求。用户是否有权搜索 `movies` 索引？ 如果是这样，用户是否有权查看该索引中的所有文档？ 响应是否应忽略或匿名化任何字段？ 对于主用户，响应可能如下所示：

```
{
  "hits": {
    "total": 7,
    "max_score": 8.772789,
    "hits": [{
        "_index": "movies",
        "_type": "_doc",
        "_id": "tt0800369",
        "_score": 8.772789,
        "_source": {
          "directors": [
            "Kenneth Branagh",
            "Joss Whedon"
          ],
          "release_date": "2011-04-21T00:00:00Z",
          "genres": [
            "Action",
            "Adventure",
            "Fantasy"
          ],
          "plot": "The powerful but arrogant god Thor is cast out of Asgard to
 live amongst humans in Midgard (Earth), where he soon becomes one of their finest
 defenders.",
          "title": "Thor",
          "actors": [
            "Chris Hemsworth",
            "Anthony Hopkins",
            "Natalie Portman"
          ],
```

```
          "year": 2011
        }
      },
      ...
    ]
  }
}
```

如果具有更有限的权限的用户发出完全相同的请求，则响应可能如下所示：

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

响应的命中次数较少，并且每次命中的字段较少。此外，`release_date` 字段是匿名的。如果不具有权限的用户发出相同的请求，集群将返回错误：

```
{
  "error": {
    "root_cause": [{
      "type": "security_exception",
      "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
```

```
    }],
    "type": "security_exception",
    "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
  },
  "status": 403
}
```

如果用户提供的凭证无效，则集群将返回 `Unauthorized` 异常。

重要概念

在开始使用精细访问控制时，请注意以下概念：

- 角色 – 使用精细访问控制的核心方式。在此情况下，角色与 IAM 角色不同。角色包含任意权限组合：集群范围的、特定于索引的、文档级别的和字段级别的。
- 映射 – 配置角色后，您可以将该角色映射到一个或多个用户。例如，您可以将三个角色映射到单个用户：一个角色提供对控制面板的访问权限，一个角色提供对 `index1` 的只读访问权限，还有一个角色提供对 `index2` 的写入访问权限。您也可以将所有这些权限包含在单个角色中。
- 用户-向 OpenSearch 集群发出请求的人员或应用程序。用户具有他们在发出请求时指定的凭据（IAM 访问密钥或用户名和密码）。

关于主用户

Ser OpenSearch vice 中的主用户可以是用户名和密码组合，也可以是对底层 OpenSearch集群拥有完全权限的 IAM 委托人。如果用户拥有 OpenSearch 集群的所有访问权限，并且能够在 OpenSearch 仪表板中创建内部用户、角色和角色映射，则该用户被视为主用户。

在 OpenSearch 服务控制台或通过 CLI 创建的主用户会自动映射到两个预定义的角色：

- `all_access` – 提供对所有集群范围操作的完全访问权限、写入所有集群索引的权限以及向所有租户写入的权限。
- `security_manager` – 提供对安全插件的访问权限以及对用户和权限进行管理的访问权限。

使用这两个角色，用户可以访问 OpenSearch 仪表板中的 “安全” 选项卡，在那里他们可以管理用户和权限。如果您创建了其他内部用户并且仅将其映射到 `all_access` 角色，则该用户无权访问安全选项卡。您可以通过将用户显式映射到 `all_access` 和 `security_manager` 角色来创建其他的主用户。有关说明，请参阅the section called “其他主用户”。

在为域创建主用户时，您可以指定现有的 IAM 主体，也可以在内部用户数据库中创建主用户。在决定要使用哪种方式时，请注意以下因素：

- IAM 委托人 — 如果您为主用户选择 IAM 委托人，则必须使用签 Amazon 名版本 4 对集群的所有请求进行签名。

  OpenSearch 服务不考虑任何 IAM 委托人的权限。IAM 用户或角色仅用于身份验证的目的。该用户或角色拥有的策略与主用户获得的授权无关。授权是通过 OpenSearch 安全插件中的各种权限进行的。

  例如，您可以向 IAM 委托人分配零 IAM 权限，只要计算机或个人可以向该用户或角色进行身份验证，他们就拥有 OpenSearch 服务中主用户的权力。

  如果您想在多个集群上使用相同的用户，如果您想使用 Amazon Cognito 访问控制面板，或者您的 OpenSearch 客户端支持签名版本 4 签名，我们建议您使用 IAM。
- 内部用户数据库 – 如果您在内部用户数据库中创建主用户（使用用户名和密码组合），则可以使用 HTTP 基本身份验证（以及 IAM 凭证）向集群发出请求。大多数客户端都支持基本身份验证，包括 curl，它还支持带有 --aw s-s Amazon igv4 选项的签名版本 4。内部用户数据库存储在 OpenSearch 索引中，因此您无法与其他集群共享。

  在以下情况下，我们建议您使用内部用户数据库：如果不需要跨多个集群重用用户，如果您想要使用 HTTP 基本身份验证访问控制面板（而不是 Amazon Cognito）或您的客户端仅支持基本身份验证。内部用户数据库是开始使用 S OpenSearch ervice 的最简单方法。

启用精细访问控制

使用控制台、 Amazon CLI或配置 API 启用精细的访问控制。要查看步骤，请参阅 创建和管理域。

精细的访问控制需要使用 Elasticsearch OpenSearch 6.7 或更高版本。它还要求所有到该域的流量都使用 HTTPS，对静态数据进行node-to-node 加密和加密。对请求进行额外处理可能需要占用各数据节点的计算和内存资源，具体取决于配置精细访问控制高级功能的方式。一旦启用精细访问控制，便无法将其禁用。

在现有域上启用精细访问控制

您可以对正在运行 OpenSearch 或 Elasticsearch 6.7 或更高版本的现有域名启用精细访问控制。

在现有域上（控制台）上启用精细访问控制

1. 选择域、Actions（操作）和 Edit security configuration（编辑安全配置）。

2. 选择 Enable fine-grained access control（启用精细访问控制）。
3. 选择如何创建主用户：
   - 如果要使用 IAM 进行用户管理，请选择 Set IAM ARN as master user (将 IAM ARN 设置为主用户)，然后为 IAM 角色指定 ARN。
   - 如果要使用内部用户数据库，请选择创建主用户，并指定用户名和密码。
4. （可选）选择 Enable migration period for open/IP-based access policy（为开放/基于 IP 的访问策略启用迁移期）。此设置启用了 30 天过渡期，在此期间，现有用户可以继续访问域而不发生中断，并且现有开放和基于 IP 的访问策略将继续适用于您的域。在此迁移期间，我们建议管理员为该域创建必要的角色并将其映射到用户。如果您使用基于身份的策略，而不是开放或基于 IP 的访问策略，则可以禁用此设置。

   此外，您还需要更新客户端，以便在迁移期间处理精细访问控制。例如，如果您将 IAM 角色映射为精细的访问控制，则必须更新您的客户端，才能开始使用签 Amazon 名版本 4 对请求进行签名。如果您使用精细访问控制配置 HTTP 基本身份验证，则必须更新客户端才能在请求中提供相应的基本身份验证凭证。

   在迁移期间，访问该域的 Das OpenSearch hboards 端点的用户将直接登陆 “发现” 页面，而不是登录页面。管理员和主用户可以选择 Login（登录），以使用管理员凭据登录并配置角色映射。

   > **Important**
   >
   > OpenSearch 服务会在 30 天后自动禁用迁移期。我们建议您在创建必要的角色并将其映射到用户之后立即终止它。迁移期结束后，您无法重新启用它。

5. 选择保存更改。

更改将会触发蓝/绿部署，在此期间，集群运行状况变为红色，但所有集群操作不会受到影响。

在现有域 (CLI) 上启用精细访问控制

将 `AnonymousAuthEnabled` 设置为 `true` 以通过精细访问控制启用迁移期：

```
aws opensearch update-domain-config --domain-name test-domain --region us-east-1 \
      --advanced-security-options '{ "Enabled": true,
 "InternalUserDatabaseEnabled":true, "MasterUserOptions": {"MasterUserName":"master-
username","MasterUserPassword":"master-password"},"AnonymousAuthEnabled": true}'
```

关于 default_role

精细访问控制需要角色映射。如果您的域使用基于身份的访问策略，S OpenSearch ervice 会自动将您的用户映射到名为 default_role 的新角色，以帮助您正确迁移现有用户。此临时映射可确保在您创建自己的角色映射之前，您的用户仍然可以成功发送 IAM 签名的 GET 和 PUT 请求。

该角色不会向您的 OpenSearch 服务域添加任何安全漏洞或缺陷。我们建议您在设置自己的角色并相应地映射它们之后删除默认角色。

迁移场景

下表介绍了在现有域上启用精细访问控制之前和之后每种身份验证方法的行为，以及管理员为将用户正确映射到角色所必须采取的步骤：

身份验证方法	启用精细访问控制之前	启用精细访问控制之后	管理员任务
基于身份的策略	满足 IAM policy 的所有用户都可以访问该域。	您无需启用迁移期。 OpenSearch 服务会自动将所有满足 IAM 策略的用户映射到 default_ role，以便他们可以继续访问该域。	1. 在域上创建自定义角色映射。 2. 删除 default_role。
基于 IP 的策略	来自允许的 IP 地址或 CIDR 块的所有用户均可访问该域。	在 30 天的迁移期间，来自允许的 IP 地址或 CIDR 块的所有用户均可继续访问该域。	1. 在域上创建自定义角色映射。 2. 根据角色映射配置，更新客户端，以提供基本身份验证凭证或 IAM 凭证。 3. 禁用迁移期。来自允许的 IP 地址或 CIDR 块的发送请求且无基本身份验证或 IAM 凭证的用户将失去对该域的访问权限。
开放访问策略	互联网上的所有用户均可访问该域。	在 30 天的迁移期间，互联网上的所有用户均可继续访问该域。	1. 在域上创建角色映射。 2. 根据角色映射配置，更新客户端，以提供基本身份验证凭证或 IAM 凭证。

身份验证方法	启用精细访问控制之前	启用精细访问控制之后	管理员任务
			3. 禁用迁移期。发送请求且无基本身份验证或 IAM 凭证的用户将失去对域的访问权限。

以主用户身份访问 OpenSearch 仪表板

精细访问控制具有 OpenSearch 仪表板插件，可简化管理任务。可以使用控制面板管理用户、角色、映射、操作组和租户。但是， OpenSearch Dashboards 登录页面和基础身份验证方法会有所不同，具体取决于您管理用户和配置域名的方式。

- 如果要使用 IAM 进行用户管理，请使用 the section called “控制面板的 Amazon Cognito 身份验证 OpenSearch ” 访问控制面板。否则，控制面板将显示一个不起作用的登录页面。请参阅the section called “限制”。

  采用 Amazon Cognito 身份验证，身份池中的某个代入角色必须与您为主用户指定的 IAM 角色匹配。有关此配置的更多信息，请参阅the section called “（可选）配置精细访问”和the section called “教程：使用 Cognito 身份验证的精细访问控制”。

- 如果您选择使用内部用户数据库，则可以使用您的主用户名和密码登录控制面板。您必须通过 HTTPS 访问控制面板。面向控制面板的 Amazon Cognito 和 SAML 身份验证都取代了此登录屏幕。

  有关此配置的更多信息，请参阅the section called “教程：使用基本身份验证的内部用户数据库”。

- 如果选择使用 SAML 身份验证，则可以使用来自外部身份提供程序的凭据登录。有关更多信息，请参阅 the section called “仪表板的 SAML 身份验证 OpenSearch ”。

管理权限

如the section called “重要概念”中所述，您可以使用角色、用户和映射管理精细访问控制权限。此部分介绍如何创建和应用这些资源。我们建议您以主用户身份登录控制面板来执行这些操作。

> **Note**
>
> 您选择授予用户的权限因使用案例而有很大差异。我们无法在本文档中涵盖所有场景。在决定向用户授予哪些权限时，请务必参考以下各节中提到的 OpenSearch 集群和索引权限，并始终遵循最低权限原则。

创建角色

您可以使用 OpenSearch 仪表板或 REST API 中的`_plugins/_security`操作来创建用于精细访问控制的新角色。有关更多信息，请参阅创建角色。

精细访问控制还包含大量预定义角色。Dashb OpenSearch oards 和 Logstash 等客户端会向其发出各种各样的请求 OpenSearch，这使得手动创建具有最低权限集的角色变得困难。例如，`opensearch_dashboards_user` 角色包括用户处理索引模式、可视化内容、控制面板和租户所需的权限。我们建议将它映射到任何访问仪表板的用户或后端角色，以及其他允许访问其他索引的角色。

亚马逊 OpenSearch 服务不提供以下 OpenSearch 角色：

- observability_full_access
- observability_read_access
- reports_read_access
- reports_full_access

Amazon Ser OpenSearch vice 提供了几个不适用于以下角色的角色 OpenSearch：

- ultrawarm_manager
- ml_full_access
- cold_manager
- notifications_full_access
- notifications_read_access

集群级安全性

集群级权限包括用于发出广泛请求（例如 _mget、_msearch 和 _bulk）、监控运行状况、拍摄快照等操作的功能。在创建角色时，使用 Cluster Permissions (集群权限)部分管理这些权限。有关集群级权限的完整列表，请参阅集群权限。

您通常可以使用默认操作组的组合来实现所需的安全状况，而不是个人权限。有关集群级操作组的列表，请参阅集群级。

索引级安全性

索引级权限包括用于执行创建新索引、搜索索引、读取和写入文档、删除文档、管理别名等操作的功能。在创建角色时，使用 Index Permissions (索引权限)部分管理这些权限。有关索引级权限的完整列表，请参阅索引权限。

您通常可以使用默认操作组的组合来实现所需的安全状况，而不是个人权限。有关索引级操作组的列表，请参阅索引级。

文档级安全性

文档级安全性允许您限制用户可在索引中查看的文档。创建角色时，请指定索引模式和 OpenSearch 查询。映射到该角色的任何用户只能查看与查询匹配的文档。文档级安全性将影响搜索时收到的命中数。

有关更多信息，请参阅文档级安全性。

字段级安全性

字段级安全性允许您控制用户可查看的文档字段。创建角色时，添加要包含或排除的字段的列表。如果包含字段，则映射到该角色的任何用户都只能查看这些字段。如果排除字段，则他们可以查看除 排除的字段之外的所有字段。字段级安全性将影响搜索时包含在命中内的字段数。

有关更多信息，请参阅字段级安全性。

字段掩码

字段遮罩是字段级安全性的替代方法，它允许您匿名化字段中的数据，而不是将其完全删除。创建角色时，添加要遮罩的字段的列表。字段遮罩将影响搜索时是否能查看字段内容。

> **Tip**
> 如果您对字段应用标准掩码，S OpenSearch ervice 会使用安全的随机哈希，这可能会导致聚合结果不准确。要对掩码字段执行聚合，请改用基于模式的掩码。

创建用户

如果您启用了内部用户数据库，则可以使用 OpenSearch仪表板或 REST API 中的`_plugins/_security`操作来创建用户。有关更多信息，请参阅创建用户。

如果您为主用户选择了 IAM，请忽略控制面板的此部分。改为创建 IAM 角色。有关更多信息，请参阅 IAM 用户指南。

将角色映射到用户

角色映射是精细访问控制的最重要的方面。精细访问控制具有一些预定义的角色来帮助您入门，但除非您将角色映射到用户，否则，向集群发出的每个请求都会以权限错误结束。

后端角色有助于简化角色映射过程。您可以将角色映射到全部 100 个用户共享的单个后端角色，而不是将同一角色映射到 100 个单独的用户。后端角色可以为 IAM 角色或任意字符串。

- 在 “用户” 部分中指定用户 ARNs、用户和 Amazon Cognito 用户字符串。Cognito 用户字符串采用的形式为 `Cognito/`*`user-pool-id`*`/`*`username`*。
- ARNs 在后端角色部分指定后端角色和 IAM 角色。

您可以使用 OpenSearch 控制面板或 REST API 中的`_plugins/_security`操作将角色映射到用户。有关更多信息，请参阅将角色映射到用户。

创建操作组

操作组是可以在不同的资源中重用的权限集。您可以使用 OpenSearch 仪表板或 REST API 中的`_plugins/_security`操作创建新的操作组，尽管默认操作组足以满足大多数用例的需求。有关原定设置操作组的更多信息，请参阅原定设置操作组。

OpenSearch 仪表板多租户

租户是用于保存索引模式、可视化内容、控制面板和其他控制面板对象的空间。控制面板多租户可让您安全地与其他控制面板用户共享您的工作（或保持您工作的私密性）并动态配置租户。您可以控制哪些角色有权访问租户，以及这些角色是否具有读取或写入访问权限。全局租户是默认租户。要了解更多信息，请参阅OpenSearch 仪表板多租户。

查看当前租户或更改租户

1. 导航到 OpenSearch 仪表板并登录。
2. 选择右上角的用户图标，然后选择切换租户。
3. 在创建可视化内容或控制面板之前验证租户。如果想与所有其他控制面板用户共享您的工作，请选择 Global (全球)。要与一部分控制面板用户共享您的工作，请选择其他共享租户。否则，请选择 Private (私有)。

> **Note**
>
> OpenSearch 仪表板为每个租户维护一个单独的索引，并创建一个名为的索引模板`tenant_template`。请勿删除或修改`tenant_template`索引，因为如果租户索引映射配置错误，可能会导致 OpenSearch 仪表板出现故障。

推荐配置

由于精细访问控制与其他安全功能的交互方式，我们建议您使用适用于大多数使用案例的多种精细访问控制配置。

描述	主用户	域访问策略
使用 IAM 凭证调用 OpenSearch APIs，并使用 SAML 身份验证访问控制面板。使用控制面板 REST API 管理精细访问控制角色。	IAM 角色或用户	JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": {`

描述	主用户	域访问策略

		``
		"Amazon": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": " domain-arn /*"
		}
		]
		}
		``

使用 IAM 凭证或基本身份验证调用 OpenSearch APIs。使用控制面板 REST API 管理精细访问控制角色。
 	
此配置提供了很大的灵活性，尤其是在您的 OpenSearch客户端仅支持基本身份验证的情况下。
 	
如果您有现有身份提供商，则使用 SAML 身份验证访问控制面板。否则，请管理内部用户数据库中的控制面板用户。
	用户名和密码	JSON
		
		``
		{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"Amazon": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": " domain-arn /*"
		}
		]
		}
		``

描述	主用户	域访问策略
使用 IAM 凭证调用 OpenSearch APIs，并使用 Amazon Cognito 访问控制面板。使用控制面板 REST API 管理精细访问控制角色。	IAM 角色或用户	JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Amazon": "*" }, "Action": "es:ESHttp*", "Resource": " domain-arn /*" } ] }`

描述	主用户	域访问策略
使用 IAM 凭证调用 OpenSearch APIs，并阻止对控制面板的大部分访问权限。使用 REST API 管理精细访问控制角色。	IAM 角色或用户	JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Amazon": "*" }, "Action": "es:ESHttp*", "Resource": " domain-arn /*" }, { "Effect": "Deny", "Principal": { "Amazon": "*" }, "Action": "es:ESHttp*", "Resource": " domain-arn /_dashboards*" } ] }`

限制

精细访问控制具有几个重要限制：

- 如果域位于 VPC 中，则角色映射的 `hosts` 部分（将角色映射映射到主机名或 IP 地址）会不起作用。您仍可以将角色映射到用户和后端角色。
- 如果您为主用户选择 IAM，但不启用 Amazon Cognito 或 SAML 身份验证，控制面板将显示一个不起作用的登录页面。
- 如果您为主用户选择 IAM，则仍可以在内部用户数据库中创建用户。但是，由于此配置下未启用 HTTP 基本身份验证，因此，使用这些用户凭证签名的任何请求都将被拒绝。

- 如果您使用 SQL 查询您无权访问的索引，则您会收到“no permissions (无权限)”错误。如果索引不存在，则会收到“no such index (无此类索引)”错误。错误消息中的此类差异意味着，如果您碰巧猜到其名称，则可以确认索引的存在。

  要最大程度地减小问题，请不要在索引名称中包含敏感信息。要拒绝所有对 SQL 的访问，请将以下元素添加到您的域访问策略：

```
{
  "Effect": "Deny",
  "Principal": {
    "Amazon": [
      "*"
    ]
  },
  "Action": [
    "es:*"
  ],
  "Resource": "arn:aws:es:us-east-1:123456789012:domain/my-domain/_plugins/_sql"
}
```

- 如果您的域版本为 2.3 或更高版本且启用了精细访问控制，则将 `max_clause_count` 设置为 1 会导致域出现问题。我们建议将此账户设置为更高的数值。
- 如果您在未设置精细访问控制的域中启用精细访问控制，则对于为直接查询而创建的数据来源，需要自行设置精细访问控制角色。有关如何设置精细访问角色的更多信息，请参阅创建与 Amazon S3 的亚马逊 OpenSearch 服务数据源集成。

修改主用户

如果您忘记了主用户的详细信息，则可以使用控制台、 Amazon CLI或配置 API 重新配置它。

修改主用户（控制台）

1. 在家中导航至 Amazon OpenSearch 服务控制台 https://console.aws.amazon.com/aos//。
2. 选择域，然后选择 Actions（操作）、Edit security configuration（编辑安全配置）。
3. 选择 Set IAM ARN as master user (将 IAM ARN 设置为主用户) 或 Create new master user (创建新的主用户)。
   - 如果您之前使用了 IAM 主用户，则精细访问控制会将 `all_access` 角色重新映射到您指定的新 IAM ARN。

- 如果您之前使用了内部用户数据库，则精细访问控制会创建一个新的主用户。您可以使用新的主用户删除旧的主用户。
- 从内部用户数据库切换到 IAM 主用户不从内部用户数据库中删除任何用户。相反，它只是禁用 HTTP 基本身份验证。手动从内部用户数据库中删除用户，或保留这些用户，以防您需要重新启用 HTTP 基本身份验证。

4. 选择保存更改。

其他主用户

在创建域时指定主用户，但如果需要，可以使用此主用户创建其他主用户。你有两个选择：OpenSearch 仪表板或 REST API。

- 在控制面板中，选择 Security (安全性) 和 Roles (角色)，然后将新主用户映射到 `all_access` 和 `security_manager` 角色。

- 要使用 REST API，请发送以下请求：

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

```
PUT _plugins/_security/api/rolesmapping/security_manager
{
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

这些请求将替换 当前角色映射，因此，首先执行 GET 请求，以便您可以在 PUT 请求中包含所有当前角色。如果您无法访问控制面板，并且希望将 IAM 角色从 Amazon Cognito 映射到 `all_access` 角色，则 REST API 会特别有用。

手动快照

精细访问控制会给拍摄手动快照带来一些额外的复杂性。要注册快照存储库——即使您使用 HTTP 基本身份验证以实现所有其他目的——您必须将 `manage_snapshots` 角色映射到具有 `iam:PassRole` 权限的 IAM 角色来代入 `TheSnapshotRole`，如 the section called “先决条件” 中所定义。

然后，使用该 IAM 角色向域发送已签名请求，如the section called “注册手动快照存储库”中所述。

集成

如果您在服务中使用其他 Amazon OpenSearch 服务，则必须为这些服务提供具有适当权限的 IAM 角色。例如，Data Firehose 传输流通常使用名为 `firehose_delivery_role` 的 IAM 角色。在控制面板中，创建一个用于精细访问控制的角色，并将 IAM 角色映射到该角色。在此情况下，新角色需要以下权限：

```
{
  "cluster_permissions": [
    "cluster_composite_ops",
    "cluster_monitor"
  ],
  "index_permissions": [{
    "index_patterns": [
      "firehose-index*"
    ],
    "allowed_actions": [
      "create_index",
      "manage",
      "crud"
    ]
  }]
}
```

权限将因每个服务执行的操作而异。为数据编制索引的 Amazon IoT 规则或 Amazon Lambda 函数可能需要与 Firehose 类似的权限，而仅执行搜索的 Lambda 函数可以使用更有限的权限集。

REST API 差异

精细访问控制 REST API 会因您的 OpenSearch/Elasticsearch 版本而略为不同。在发出 PUT 请求之前，请发出 GET 请求以验证预期请求正文。例如，对 `_plugins/_security/api/user` 的 GET 请求将返回所有用户，然后您可以修改并使用这些用户来发出有效的 PUT 请求。

在 Elasticsearch 6. x，用于创建用户的请求与以下内容类似：

```
PUT _opendistro/_security/api/user/new-user
{
  "password": "some-password",
  "roles": ["new-backend-role"]
}
```

在 E OpenSearch lasticsearch 7.x 上，请求如下所示（`_opendistro`如果使用 Elasticsearch，则改`_plugins`为）：

```
PUT _plugins/_security/api/user/new-user
{
  "password": "some-password",
  "backend_roles": ["new-backend-role"]
}
```

此外，在 Elasticsearch 6. x 中，租户是角色的属性：

```
GET _opendistro/_security/api/roles/all_access

{
  "all_access": {
    "cluster": ["UNLIMITED"],
    "tenants": {
      "admin_tenant": "RW"
    },
    "indices": {
      "*": {
        "*": ["UNLIMITED"]
      }
    },
    "readonly": "true"
  }
}
```

在 OpenSearch Elasticsearch 7.x 中，它们是具有自己的 URI 的对象（`_opendistro`如果使用 Elasticsearch，则改`_plugins`为）：

```
GET _plugins/_security/api/tenants

{
  "global_tenant": {
    "reserved": true,
    "hidden": false,
    "description": "Global tenant",
    "static": false
  }
}
```

有关 OpenSearch REST API 的文档，请参阅安全插件 API 参考。

> **Tip**
>
> 如果您使用内部用户数据库，则可以使用 curl 发出请求并测试您的域。尝试以下示例命令：
>
> ```
> curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_search'
> curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_plugins/
> _security/api/user'
> ```

教程：使用 IAM 主用户和 Amazon Cognito 身份验证配置域

本教程介绍了一个用于精细访问控制的常见亚马逊 OpenSearch 服务用例：对控制面板采用 Amazon Cognito 身份验证的 IAM 主用户。 OpenSearch

在本教程中，我们将配置一个主 IAM 角色和一个受限 IAM 角色，然后将其与 Amazon Cognito 中的用户关联。然后，主用户可以登录控制 OpenSearch 面板，将受限用户映射到角色，并使用精细的访问控制来限制用户的权限。

尽管这些步骤使用 Amazon Cognito 用户池进行身份验证，但相同的基本流程适用于任何 Cognito 身份验证提供商，从而允许您将不同的 IAM 角色分配给不同的用户。

在本教程中，您将完成以下步骤：

1. 创建主 IAM 角色和受限 IAM 角色
2. 使用 Cognito 身份验证创建域
3. 配置 Cognito 用户池和身份池
4. 在 OpenSearch 仪表板中映射角色

5. 测试权限

步骤 1：创建主 IAM 角色和受限 IAM 角色

导航到 Amazon Identity and Access Management (IAM) 控制台并创建两个单独的角色：

- `MasterUserRole`：主用户，它将拥有针对集群的完全权限，并可管理角色和角色映射。
- `LimitedUserRole`：更受限制的角色，您将以主用户身份为其授予有限访问权限。

有关创建角色的说明，请参阅 IAM 用户指南中的使用自定义信任策略创建角色。

这两个角色都必须具有以下信任策略，允许您的 Cognito 身份池代入这些角色：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "cognito-identity.amazonaws.com:aud": "{identity-pool-id}"
      },
      "ForAnyValue:StringLike": {
        "cognito-identity.amazonaws.com:amr": "authenticated"
      }
    }
  }]
}
```

> **Note**
>
> 将 `identity-pool-id` 替换为您的 Amazon Cognito 身份池的唯一标识符。例如 `us-east-1:0c6cdba7-3c3c-443b-a958-fb9feb207aa6`。

步骤 2：使用 Cognito 身份验证创建域

在https://console.aws.amazon.com/aos/家中/ OpenSearch 进入亚马逊服务控制台，然后使用以下设置创建一个域名：

- OpenSearch 1.0 或更高版本，或 Elasticsearch 7.8 或更高版本
- 公有访问权限
- 以主用户（在上一步中创建）身份借助 `MasterUserRole` 启用的精细访问控制
- 控制面板启用了 Amazon Cognito 身份验证。 OpenSearch 有关启用 Cognito 身份验证以及选择用户和身份池的说明，请参阅 the section called “将域配置为使用 Amazon Cognito 身份验证”。
- 以下域访问策略：

  JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Amazon": "arn:aws:iam::{account-id}:root"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:{region}:{account-id}:domain/{domain-name}/*"
    }
  ]
}
```

- 所有到达域的流量所需的 HTTPS
- Node-to-node 加密
- 静态数据加密

步骤 3：配置 Cognito 用户

创建域时，请按照 Amazon Cognito开发人员指南中的创建用户池配置 Amazon Cognito 的主用户和受限用户。最后，按照在 Amazon Cognito 中创建身份池中的步骤配置您的身份池。用户池和身份池必须在同一个 Amazon Web Services 区域中。

步骤 4：在 OpenSearch 仪表板中映射角色

现在，您的用户已配置完毕，您可以以主用户身份登录 OpenSearch 仪表板并将用户映射到角色。

1. 返回 OpenSearch 服务控制台并导航到您创建的域名的控制 OpenSearch 面板 URL。URL 遵循此格式：*domain-endpoint*/_dashboards/。
2. 使用 master-user 凭证登录。
3. 选择 Add sample data（添加示例数据），然后添加示例飞行数据。
4. 在左侧导航窗格中，选择 Security（安全性）、Roles（角色）、Create role（创建角色）。
5. 将角色命名为 new-role。
6. 对于 Index（索引），指定 opensearch_dashboards_sample_data_fli*（Elasticsearch 域上的 kibana_sample_data_fli*）。
7. 对于 Index permissions（索引权限），选择 read（读取）。
8. 对于 Document Level Security Query (文档级安全查询)，请指定以下查询：

```
{
  "match": {
    "FlightDelay": true
  }
}
```

9. 对于字段级安全性，请选择 Exclude（不包括）并指定 FlightNum。
10. 对于匿名，指定 Dest。
11. 选择创建。
12. 选择映射的用户、管理映射。为 LimitedUserRole 添加 Amazon 资源名称（ARN）作为外部标识，然后选择 Map（映射）。
13. 返回角色列表，然后选择 opensearch_dashboards_user。选择映射的用户、管理映射。为 LimitedUserRole 添加 ARN 作为后端角色，然后选择映射。

步骤 5：测试权限

角色正确映射后，您可以以受限用户身份登录并测试权限。

1. 在新的私有浏览器窗口中，导航到该域的 OpenSearch 仪表板网址，使用limited-user凭据登录，然后选择自己浏览。
2. 转到 Dev Tools (开发人员工具)，然后运行原定设置搜索：

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

请注意权限错误。`limited-user` 没有运行集群范围内搜索的权限。

3. 运行另一个搜索：

```
GET opensearch_dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

请注意，在所有匹配的文档中，有一个为 `true` 的 `FlightDelay` 字段，一个匿名化的 `Dest` 字段，但没有 `FlightNum` 字段。

4. 在原始浏览器窗口中，以 `master-user` 身份登录，选择 Dev Tools (开发人员工具)，然后执行相同的搜索。请注意权限、命中数、匹配文档和包含字段的差异。

教程：使用内部用户数据库和 HTTP 基本身份验证配置域

本教程涵盖了另一个流行的细粒度访问控制用例：内部用户数据库中的主用户和仪表板的 HTTP 基本身份验证。 OpenSearch 然后，主用户可以登录 OpenSearch 控制面板，创建内部用户，将用户映射到角色，并使用精细的访问控制来限制用户的权限。

在本教程中，您将完成以下步骤：

1. 使用主用户创建域
2. 在 OpenSearch 仪表板中配置内部用户
3. 在 OpenSearch 仪表板中映射角色
4. 测试权限

步骤 1：创建域

在https://console.aws.amazon.com/aos/家中/ OpenSearch 进入亚马逊服务控制台，然后使用以下设置创建一个域名：

- OpenSearch 1.0 或更高版本，或 Elasticsearch 7.9 或更高版本
- 公有访问权限
- 对内部用户数据库中的主用户的精细访问控制（对于本教程的其余部分，为 `TheMasterUser`）
- 控制面板的 Amazon Cognito 身份验证已禁用
- 以下访问策略：

  JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Amazon": "arn:aws:iam::{account-id}:root"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:{region}:{account-id}:domain/{domain-name}/*"
    }
  ]
}
```

- 所有到达域的流量所需的 HTTPS
- Node-to-node 加密
- 静态数据加密

步骤 2：在 OpenSearch 控制面板中创建内部用户

现在您已经有了域名，可以登录 OpenSearch 控制面板并创建内部用户。

1. 返回 OpenSearch 服务控制台并导航到您创建的域名的控制 OpenSearch 面板 URL。URL 遵循此格式：*domain-endpoint*/_dashboards/。

2. 使用 `TheMasterUser` 登录。
3. 选择 Add sample data（添加示例数据），然后添加示例飞行数据。
4. 在左侧导航窗格中，选择安全、内部用户、创建内部用户。
5. 命名用户 `new-user` 并指定密码。然后选择创建。

步骤 3：在 OpenSearch 仪表板中映射角色

现在，用户配置完毕，您可以将用户映射到角色。

1. 在 “控制面 OpenSearch 板” 的 “安全” 部分，选择 “角色”、“创建角色”。
2. 将角色命名为 `new-role`。
3. 对于索引，为索引模式指定 `opensearch_dashboards_sample_data_fli*`（Elasticsearch 域上的 `kibana_sample_data_fli*`）。
4. 对于操作组，请选择读取。
5. 对于 Document Level Security Query (文档级安全查询)，请指定以下查询：

```
{
  "match": {
    "FlightDelay": true
  }
}
```

6. 对于字段级安全性，请选择 Exclude（不包括）并指定 `FlightNum`。
7. 对于匿名，指定 `Dest`。
8. 选择创建。
9. 选择映射的用户、管理映射。然后添加 `new-user` 到用户，选择映射。
10. 返回角色列表，然后选择 opensearch_dashboards_user。选择映射的用户、管理映射。然后添加 `new-user` 到用户，选择映射。

步骤 4：测试权限

角色正确映射后，您可以以受限用户身份登录并测试权限。

1. 在新的私有浏览器窗口中，导航到该域的 OpenSearch 仪表板网址，使用`new-user`凭据登录，然后选择自己浏览。

2. 转到 Dev Tools (开发人员工具)，然后运行原定设置搜索：

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

请注意权限错误。`new-user` 没有运行集群范围内搜索的权限。

3. 运行另一个搜索：

```
GET dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

请注意，在所有匹配的文档中，有一个为 `true` 的 `FlightDelay` 字段，一个匿名化的 `Dest` 字段，但没有 `FlightNum` 字段。

4. 在原始浏览器窗口中，以 `TheMasterUser` 身份登录，选择 Dev Tools (开发人员工具)，然后执行相同的搜索。请注意权限、命中数、匹配文档和包含字段的差异。

Amazon OpenSearch 服务的合规性验证

作为多项合规计划的一部分，第三方审计师评估亚马逊 OpenSearch 服务的安全与 Amazon 合规性。这些计划包括 SOC、PCI 和 HIPAA。

如果您有合规性要求，可以考虑使用 OpenSearch 或 Elasticsearch 6.0 或更高版本的任何版本。早期版本的 Elasticsearch 不提供静态数据加密和node-to-node 加密的组合，因此不太可能满足您的需求。如果精细的访问控制对您的用例很重要，您也可以考虑使用任何版本的 OpenSearch 或 Elasticsearch 6.7 或更高版本。无论如何，在创建域名时选择特定版本 OpenSearch 或 Elasticsearch 版本并不能保证合规性。

要了解是否属于特定合规计划的范围，请参阅Amazon Web Services 服务 “” Amazon Web Services 服务 中的 “按合规计划划分的范围”，然后选择您感兴趣的合规计划。 Amazon Web Services 服务 有关一般信息，请参阅合规计划。

您可以使用下载第三方审计报告 Amazon Artifact。有关更多信息，请参阅中的 “下载报告” Amazon Artifact。

您在使用 Amazon Web Services 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 Amazon 提供了以下资源来帮助实现合规性：

- Security & Compliance：这些解决方案实施指南讨论了架构考虑因素，并提供了部署安全性和合规性功能的步骤。
- 合规资源 — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- 使用Amazon Config 开发人员指南中的规则评估资源 — 该 Amazon Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- Amazon Security Hub— 这 Amazon Web Services 服务 提供了您内部安全状态的全面视图 Amazon。Security Hub 通过安全控制措施评估您的 Amazon 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控制措施的列表，请参阅 Security Hub 控制措施参考。
- Amazon GuardDuty — 它通过监控您的 Amazon Web Services 账户环境中是否存在可疑和恶意活动，来 Amazon Web Services 服务 检测您的工作负载、容器和数据面临的潜在威胁。 GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。

Amazon S OpenSearch ervice 中的能力

Amazon 全球基础设施围绕 Amazon Web Services 区域 和可用区构建。 Amazon Web Services 区域 提供多个在物理上独立且隔离的可用区，这些可用区与延迟率低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础架构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 Amazon Web Services 区域 和可用区的更多信息，请参阅Amazon 全球基础设施。

除了 Amazon 全球基础设施之外，Ser OpenSearch vice 还提供多种功能，以帮助支持您的数据恢复能力和备份需求：

- 多可用区域分片和副本分片
- 自动和手动快照

亚马逊 OpenSearch 服务的 JWT 身份验证和授权

亚马逊 OpenSearch 服务现在允许您使用 JSON 网络令牌 (JWTs) 进行身份验证和授权。 JWTs 是基于 JSON 的访问令牌，用于授予单点登录 (SSO) 访问权限。您可以在 S OpenSearch ervice JWTs 中

使用创建单点登录令牌来验证对您的 OpenSearch 服务域的请求。要使用 JWTs，必须启用精细访问控制，并且必须提供有效的 RSA 或 ECDSA PEM 格式的公钥。有关精细访问控制的更多信息，请参阅 Amazon Service 中的精细访问控制。 OpenSearch

您可以使用 OpenSearch 服务控制台、 Amazon Command Line Interface (Amazon CLI) 或配置 JSON Web 令牌 Amazon SDKs。

注意事项

在 JWTs 使用 Amazon OpenSearch 服务之前，您必须考虑以下几点：

- 由于 PEM 格式的 RSA 公有密钥很大，我们建议使用 Amazon 控制台配置 JWT 身份验证和授权。
- 在为您指定主题和角色字段时，您必须提供有效的用户和角色 JWTs，否则，请求将被拒绝。
- OpenSearch 2.11 是可用于 JWT 身份验证的最早兼容版本。

修改域访问策略

在将域配置为使用 JWT 身份验证和授权之前，必须更新域访问策略，以允许 JWT 用户访问该域。否则，所有传入的 JWT 授权请求都将被拒绝。为子资源 (/*) 提供完全访问权限的建议域访问策略是：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "domain-arn/*"
    }
  ]
}
```

配置 JWT 身份验证和授权

您可以在域创建过程中或通过更新现有域来启用 JWT 身份验证和授权。根据您选择的具体选项，设置步骤会略有差异。

以下步骤说明了如何在 OpenSearch 服务控制台中为 JWT 身份验证和授权配置现有域：

1. 在 “域配置” 下，导航到 “JWT 身份验证和授权” OpenSearch，选择 “启用 JWT 身份验证和授权”。
2. 配置要用于域的公有密钥。要执行此操作，您可以上传包含公有密钥的 PEM 文件，也可以手动输入。

   > **Note**
   > 如果上传或输入的密钥无效，则会在文本框上方显示一条警告，说明具体的问题。

3. （可选）您可以在“其他设置”下配置以下可选字段

   - 主题密钥 —您可以将此字段留空以使用您的默认`sub`密钥 JWTs.
   - 角色密钥 —您可以将此字段留空以使用您的默认`roles`密钥 JWTs.

   完成更改后，请保存域。

使用 JWT 发送测试请求

创建具有指定主体和角色对的新 JWT 后，您可以发送测试请求。为此，请使用私钥通过创建 JWT 的工具签署您的请求。 OpenSearch 服务能够通过验证此签名来验证传入的请求。

> **Note**
> 如果您为 JWT 指定了自定义的主体密钥或角色密钥，则必须使用 JWT 的正确声明名称。

以下是如何使用 JWT 令牌通过域名的搜索端点访问 OpenSearch 服务的示例：

```
curl -XGET "$search_endpoint" -H "Authorization: Bearer <JWT>"
```

配置 JWT 身份验证和授权（Amazon CLI）

如果域存在，以下 Amazon CLI 命令将启用 JWT 身份验证和授权： OpenSearch

```
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>"}}'
```

配置 JWT 身份验证和授权（通过 API 配置）

以下对配置 API 的请求在现有域 OpenSearch 上启用 JWT 身份验证和授权：

```
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"
   }
  }
}
```

生成密钥对

为了 JWTs 为您的 OpenSearch 域进行配置，您需要提供隐私增强邮件 (PEM) 格式的公钥。亚马逊 OpenSearch 服务目前在使用时支持两种非对称加密算法 JWTs：RSA 和 ECDSA。

要使用通用的 openssl 库创建 RSA 密钥对，请执行以下步骤：

1. `openssl genrsa -out privatekey.pem 2048`
2. `openssl rsa -in privatekey.pem -pubout -out publickey.pem`

在此示例中，`publickey.pem`文件包含用于 Amazon OpenSearch 服务的公钥，而`privatekey.pem`包含用于对 JWTs 发送到服务的文件进行签名的私有密钥。此外，如果您需要将私钥转换为常用`pkcs8`格式来生成，则可以选择将其转换为常用格式 JWTs。

如果您使用上传按钮将 PEM 文件直接添加到控制台，该文件的扩展名必须为 `.pem`，目前不支持其他文件扩展名，例如 `.crt`、`.cert` 或 `.key`。

Amazon S OpenSearch ervice 中的基础设施安全性

作为一项托管式服务，Amazon S OpenSearch ervice 受 Amazon 全球网络安全保护。有关 Amazon 安全服务以及如何 Amazon 保护基础架构的信息，请参阅Amazon 云安全。要按照基础设施安全最佳实践设计您的 Amazon 环境，请参阅《安全性支柱 Amazon Well-Arch itected Framework》中的基础设施保护。

您可以使用 Amazon 发布的 API 调用通过网络访问 S OpenSearch ervice。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用 Amazon Security Token Service（Amazon STS）生成临时安全凭证来对请求进行签名。

您可以使用 Amazon 发布的 API 调用通过网络访问 S OpenSearch ervice 配置 API。要配置要接受的最低所需 TLS 版本，请指定域端点选项中的 `TLSSecurityPolicy` 值：

```
aws opensearch update-domain-config --domain-name my-domain --domain-endpoint-options
 '{"TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"}'
```

有关详细信息，请参阅 Amazon CLI 命令参考。

根据您的域配置，您可能还需要签署发往的请求 OpenSearch APIs。有关更多信息，请参阅 the section called “提出和签署 OpenSearch 服务请求”。

OpenSearch Service 支持公共访问域，它可以接收来自任何连接 Internet 的设备的请求，并且支持 VP C 访问域, 它与公共 Internet 隔离开来。

使用 OpenSearch OpenSearch 服务管理的 VPC 终端节点访问亚马逊服务 ()Amazon PrivateLink

您可以通过设置 OpenSearch OpenSearch 服务托管的 VPC 端点（由提供支持 Amazon PrivateLink）来访问 Amazon Service 域。这些端点可在您的 VPC 和 Amazon Serv OpenSearch ice 之间创建私有连接。您可以像在 VPC 中一样访问 OpenSearch 服务 VPC 域，而无需使用互联网网关、NAT 设备、VPN 连接或 Amazon Direct Connect 连接。VPC 中的实例不需要公有 IP 地址即可访问 OpenSearch 服务。

您可以配置 OpenSearch 服务域，以公开在同一 VPC、不同或不同内的公有或私有子网上运行的其他端点。 Amazon Web Services 账户这使您能为访问您的域（无论这些域在何处运行）增加一层额外的安全保护，而无需管理基础架构。下图阐明了位于同一 VPC 内的 OpenSearch Service 托管的 VPC 端点：

您可以通过创建由提供支持的 OpenSearch 服务托管的接口 VPC 端点来 Amazon PrivateLink建立此私有连接。我们将在您为接口 VPC 端点启用的每个子网中创建一个端点网络接口。这些是服务托管式网络接口，用作发往 Serv OpenSearch ice 的流量的入口点。标准Amazon PrivateLink 接口终端节点定价适用于计费的 OpenSearch 服务托管 VPC 终端节点。 Amazon PrivateLink

您可以为运行所有版本 OpenSearch 和旧式 Elasticsearch 的域创建 VPC 端点。有关更多信息，请参阅《Amazon PrivateLink 指南》中的通过 Amazon PrivateLink访问 Amazon Web Services 服务。

S OpenSearch ervice 注意事项和限制

在为 S OpenSearch ervice 设置接口 VPC 端点之前，请先查看Amazon PrivateLink 《指南》中的使用接口 VPC 端点访问 Amazon 服务。

在使用 OpenSearch 服务托管的 VPC 端点时，请考虑以下事项：

- 您只能使用接口 VPC 端点连接到 VPC 域。不支持公共域。
- VPC 端点只能连接到同一 Amazon Web Services 区域内的域。
- HTTPS 是 VPC 端点唯一支持的协议。不允许使用 HTTP。
- OpenSearch Service 支持通过接口 VPC 端点调用所有受支持的 OpenSearch API 操作。
- 每个账户最多可以配置 50 个端点，每个域最多可以配置 10 个端点。一个域最多可以有 10 个授权主体。
- 目前无法使用 Amazon CloudFormation 创建接口 VPC 端点。
- 您只能通过 OpenSearch 服务控制台或使用OpenSearch 服务 API 创建接口 VPC 端点。无法使用 Amazon VPC 控制台为 OpenSearch 服务创建接口 VPC 端点。
- OpenSearch 无法从互联网访问 Service 托管的 VPC 端点。在路由表和 OpenSearch 安全组允许的情况下与预调配端点的 VPC 对等连接的任何 V VPCs PC 中访问，或者在路由表和安全组允许的情况下与预调配端点的 VPC 对等连接的任何 VPC 中访问。
- Serv OpenSearch ice 不支持 VPC 端点策略。您可以将安全组与端点网络接口关联起来，以控制通过接口 VPC 端点流向 Serv OpenSearch ice 的流量。
- 您的服务相关角色必须与用于创建 VPC 端点的 Amazon 账户相同。
- 要创建、更新和删除 OpenSearch 服务 VPC 端点，除 Amazon Service EC2 权限之外，还必须拥有以下 Amazon OpenSearch 权限：
  - `ec2:CreateVpcEndpoint`
  - `ec2:DescribeVpcEndpoints`
  - `ec2:ModifyVpcEndpoint`
  - `ec2:DeleteVpcEndpoints`
  - `ec2:CreateTags`
  - `ec2:DescribeTags`
  - `ec2:DescribeSubnets`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeVpcs`

> **Note**
>
> 目前，无法仅限在 OpenSearch 服务中创建 VPC 端点。我们正在努力，希望在未来的更新中做到这一点。

提供针对域的访问权限

如果您要访问域的 VPC 位于另一个中 Amazon Web Services 账户，您需要从所有者账户中为其授权，然后才能创建接口 VPC 端点。

允许另一个中的 VPC Amazon Web Services 账户 访问您的域

1. 在家中打开亚马逊 OpenSearch 服务控制台 https://console.aws.amazon.com/aos//。
2. 在导航窗格中，选择 Domains（域），然后打开您要为其提供访问权限的域。
3. 转到 VPC 端点选项卡，其中显示有权访问您的域 VPCs 的账户和相应的。
4. 选择 Authorize principal（为主体授权）。
5. 输入要访问您的域的账户的 Amazon Web Services 账户 ID。此步骤将为指定账户授权，以针对域创建 VPC 端点。
6. 选择授权。

为 VPC 域创建接口 VPC 端点

您可以使用服务控制台或 Amazon Command Line Interface (Amazon CLI) 为 OpenSearch 服务创建接口 VPC 终端节点。 OpenSearch

为 OpenSearch 服务域创建接口 VPC 端点

1. 在家中打开亚马逊 OpenSearch 服务控制台 https://console.aws.amazon.com/aos//。
2. 在左侧导航窗格中，选择 VPC endpoints（VPC 端点）。
3. 选择创建端点。
4. 选择是连接位于当前 Amazon Web Services 账户 中的域，还是连接位于其他中的域 Amazon Web Services 账户。
5. 选择您使用此端点连接的域。如果域位于当前中 Amazon Web Services 账户，请使用下拉列表选择该域。如果域位于另一个账户中，请输入要连接的域的 Amazon 资源名称（ARN）。要在其他账户中选择域名，所有者需要向您提供该域名的访问权限。要选择位于另一个账户中的域，所有者需要为您提供访问该域的权限。
6. 对于 VPC，选择您要从中访问 OpenSearch 服务的 VPC。
7. 对于 Subnets（子网），选择您要从中访问 OpenSearch Service 的一个或多个子网。
8. 对于 Security groups（安全组），选择要与端点网络接口关联的安全组。这是一个关键步骤，您可以在该步骤中限制您授权进入端点的入站流量的端口、协议和源。安全组规则必须允许将使用 VPC 端点与 Serv OpenSearch ice 通信的资源与端点网络接口进行通信。

9. 选择创建端点。该端点应在 2 至 5 分钟内处于活动状态。

使用配置 AP OpenSearch I 使用服务托管的 VPC 端点

使用以下 API 操作来创建和管理 OpenSearch 服务托管的 VPC 端点。

- CreateVpcEndpoint
- ListVpcEndpoints
- UpdateVpcEndpoint
- DeleteVpcEndpoint

使用以下 API 操作来管理针对 VPC 域的端点访问：

- AuthorizeVpcEndpointAccess
- ListVpcEndpointAccess
- ListVpcEndpointsForDomain
- RevokeVpcEndpointAccess

仪表板的 SAML 身份验证 OpenSearch

OpenSearch 控制面板的 SAML 身份验证允许您使用现有的身份提供商为运行 OpenSearch 或 Elasticsearch 6.7 或更高版本的亚马逊 OpenSearch 服务域上的控制面板提供单点登录 (SSO)。要使用 SAML 身份验证，必须启用访问权限的精细控制。

控制面板的 S AM L 身份验证允许您使用第三方身份 OpenSearch 提供商登录控制面板、管理精细访问控制、搜索数据和构建可视化效果，而不是通过 Amazon Cognito 或内部用户数据库进行身份验证。OpenSearch 服务支持使用 SAML 2.0 标准的提供商，例如 Okta、Keycloak、Active Directory 联合身份验证服务 (ADFS)、Auth0 和。 Amazon IAM Identity Center

仪表板的 SAML 身份验证仅适用于通过 Web 浏览器访问 OpenSearch 仪表板。您的 SAML 凭据不允许您直接向 OpenSearch 或控制面板 APIs发出 HTTP 请求。

SAML 配置概述

本文档假定您有现有的身份提供程序并且熟悉它。我们无法为您的具体提供商提供详细的配置步骤，只能为您的 OpenSearch 服务域提供详细的配置步骤。

OpenSearch 仪表板登录流程可以采用以下两种形式之一：

- 已启动服务提供程序 (SP)：导航到控制面板（例如 `https://my-domain.us-east-1.es.amazonaws.com/_dashboards`），它会将您重定向到登录屏幕。登录后，身份提供程序会将您重定向到控制面板。
- 身份提供者 (IdP) 已启动：您导航到您的身份提供商，登录，然后从应用程序目录中选择 Dash OpenSearch boards。

OpenSearch 服务提供两种单点登录 URLs，SP 启动和 IDP 启动，但您只需要与所需控制面板登录流程相匹配的单点登录。 OpenSearch

无论使用哪种身份验证类型，目标都是通过身份提供程序登录并接收包含您的用户名（必需）和任何后端角色（可选，但推荐执行）。此信息允许访问权限的精细控制向 SAML 用户分配权限。在外部身份提供程序中，后端角色通常称为“角色”或“组”。

注意事项

在配置 SAML 身份验证时，请考虑以下事项：

- 由于 IdP 元数据文件的大小，我们强烈建议使用 Amazon 控制台配置 SAML 身份验证。
- 域一次只支持一种控制面板身份验证方法。如果您启用了 OpenSearch 控制面板的 Amazon Cognito 身份验证，则必须先将其禁用，然后才能启用 SAML 身份验证。
- 如果将网络负载均衡器与 SAML 一起使用，则必须先创建自定义端点。有关更多信息，请参阅 ???。
- 对于非 IAM 身份（例如 Amazon S OpenSearch erverless 和 SAML 中的 SAML 以及亚马逊服务的基本内部用户授权），服务控制策略 (SCP) 将不适用或不进行评估。 OpenSearch

用于 VPC 域的 SAML 身份验证

SAML 不要求身份提供程序和服务提供程序之间直接进行通信。因此，即使您的 OpenSearch 域托管在私有 VPC 中，只要您的浏览器可以与 OpenSearch 集群和身份提供商通信，您仍然可以使用 SAML。您的浏览器基本上充当身份提供商和服务提供商之间的中间人。有关解释 SAML 身份验证流程的有用图表，请参阅 Okta 文档。

修改域访问策略

在配置 SAML 身份验证之前，必须更新域访问策略，以允许 SAML 用户访问该域。否则，会显示拒绝访问的错误。

我们建议使用以下域访问策略，它将提供针对域上子资源 (/*) 的完全访问权限：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "domain-arn/*"
    }
  ]
}
```

要使策略的限制更为严格，可以在策略中添加 IP 地址条件。此条件将仅允许访问指定的 IP 地址范围或子网。例如，以下策略仅允许从 192.0.2.0/24 子网进行访问：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
```

```
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "domain-arn/*"
    }
  ]
}
```

> **Note**
>
> 开放域访问策略要求在域上启用精细访问控制，否则您会看到以下错误：
> `To protect domains with public access, a restrictive policy or fine-grained access control is required.`
> 如果您的主用户或内部用户配置了强健密码，则从安全角度来看，在使用精细访问控制的同时保持策略开放也许是可以接受的。有关更多信息，请参阅 ???。

配置 SP 或 IdP 发起的身份验证

这些步骤说明了如何通过 SP 启动或 IDP 启动的身份验证为仪表板启用 SAML 身份验证。OpenSearch 有关同时启用这两种身份验证所需的额外步骤，请参阅配置 SP 和 IdP 发起的身份验证。

步骤 1：启用 SAML 身份验证

您可以在域创建期间启用 SAML 身份验证，也可以在现有域上选择 Actions（操作），Edit security configuration（编辑安全配置）。根据您的选择，以下步骤略有不同。

在域配置中，在 “ OpenSearch 仪表板/Kibana 的 SAML 身份验证” 下，选择 “启用 SAML 身份验证”。

步骤 2：配置身份提供程序

根据配置 SAML 身份验证的时间，执行以下步骤。

如果正在创建新域

如果您正在创建新域，则服务尚无法生成 OpenSearch 服务提供商实体 ID 或 SSO URLs。身份提供程序需要这些值才能正确启用 SAML 身份验证，但它们只能在创建域后生成。要在域创建期间解决这种

相互依赖关系，您可以在 IdP 配置中提供临时值以生成所需的元数据，然后在域处于活动状态后对其进行更新。

如果您使用的是自定义终端节点，则可以推断出 URLs 将是什么。例如，如果您的自定义端点是 `www.custom-endpoint.com`，则服务提供程序实体 ID 将是 `www.custom-endpoint.com`，IdP 发起的 SSO URL 将是 `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs/idpinitiated`，SP 发起的 SSO URL 将是 `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs`。在创建域之前，您可以使用这些值来配置身份提供程序。有关示例，请参阅下一节。

> **Note**
>
> 您不能使用双栈端点登录，因为 HTTP 请求的 FQDN 与 SAML 请求的 FQDN 不同。如果您想使用双堆栈终端节点登录，则 OpenSearch管理员需要设置自定义终端节点并将别名记录值设置为双堆栈端点。

如果您没有使用自定义端点，则可以在 IdP 中输入临时值以生成所需的元数据，然后在域处于活动状态后对其进行更新。

例如，在 Okta 中，您可以在 Single sign on URL（单一登录 URL）和 Audience URI (SP Entity ID)（受众 URI（SP 实体 ID））字段输入 `https://temp-endpoint.amazonaws.com`，以生成元数据。然后，在域名处于活动状态后，您可以从 S OpenSearch ervice 中检索正确的值并在 Okta 中对其进行更新。有关说明，请参阅the section called “第 6 步：更新你的 IdP URLs”。

如果您正在编辑现有域

如果您要在现有域上启用 SAML 身份验证，请复制服务提供商实体 ID 和其中一个 SSO URLs。有关使用哪个 URL 的指南，请参阅 the section called “SAML 配置概述”。

Service provider entity ID

https://search-my-saml-domain-ob5t7vqdask2pav3r5pjjtvrxy.us-east-1.es.amazonaws.com

IdP-initiated SSO URL

https://search-my-saml-domain-ob5t7vqdask2pav3r5pjjtvrxy.us-east-1.es.amazonaws.com/_dashboards/_opendistro/_security/saml/acs/idpinitiated

SP-initiated SSO URL

https://search-my-saml-domain-ob5t7vqdask2pav3r5pjjtvrxy.us-east-1.es.amazonaws.com/_dashboards/_opendistro/_security/saml/acs

使用这些值配置您的身份提供程序。这是过程中最复杂的部分，不幸的是，术语和步骤因提供程序而异。请参阅提供程序的文档。

例如，在 Okta 中，您创建一个 SAML 2.0 Web 应用程序。为 Single sign on URL（单一登录 URL）指定 SSO URL。对于受众 URI（SP 实体 ID）中，指定 SP 实体 ID。

Okta 拥有用户和组，而不是用户和后端角色。对于 Group Attribute Statements（组属性语句），建议将 `role` 添加到 Name（名称）字段，并将正则表达式 `.+` 添加到 Filter（筛选条件）字段。此语句告诉 Okta 身份提供程序在用户进行身份验证包含 SAML 断言的字段 `role` 下面的所有用户组。

在 IAM Identity Center 中，您可以将 SP 实体 ID 指定为应用程序 SAML 受众。您还需要指定以下属性映射：`Subject=${user:subject}:format=unspecified` 和 `Role=${user:groups}:format=uri`。

在 Auth0 中，创建常规 Web 应用程序，然后启用 SAML 2.0 加载项。在 Keycloak 中，创建客户端。

步骤 3：导入 IdP 元数据

配置了身份提供程序后，它会生成 IdP 元数据文件。此 XML 文件包含有关提供者的信息，如 TLS 证书、单点登录端点和身份提供者的实体 ID。

复制 IdP 元数据文件的内容并将其粘贴到服务控制台的 “来自 IdP 的元数据” 字段中。 OpenSearch 也可以选择从 XML 文件导入并上载文件。元数据文件应如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
```

```
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
 Location="idp-sso-url"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

步骤 4：配置 SAML 字段

输入 IdP 元数据后，请在 OpenSearch 服务控制台中配置以下其他字段：

- IdP entity ID（IdP 实体 ID）：从元数据文件中复制 `entityID` 属性值粘贴到此字段。许多身份提供程序还将此值显示为配置后摘要的一部分。有些提供程序称之为 “发行人”。
- SAML 主用户名和 SAML 主后 and/or 端角色-您指定的用户后端角色获得集群的完全权限，相当于新的主用户，但只能在控制面板中 OpenSearch 使用这些权限。

  例如，在 Okta 中，您可能有属于群组 `admins` 的用户 `jdoe`。如果将 `jdoe` 添加到 SAML 主用户名字段中，只有该用户才会获得完全权限。如果将 `admins` 添加到 SAML 主后端角色字段中，任何属于 `admins` 组的用户将获得完全权限。

> **Note**
>
> SAML 断言的内容必须与用于 SAML 主用户名和 SAML 主角色的字符串完全匹配。一些身份提供商在其用户名前添加前缀，这可能会导致 hard-to-diagnose不匹配。在身份提供程序用户界面中，您可能会看到 `jdoe`，但 SAML 断言可能包含 `auth0|jdoe`。始终使用 SAML 断言中的字符串。

许多身份提供程序让您可以在配置过程中查看示例断言，以及 SAML 跟踪可以帮助您检查和排除真实断言的内容。断言如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion ID="id67229299299259351343340162"
 IssueInstant="2020-09-22T22:03:08.633Z" Version="2.0"
  xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">idp-issuer</
saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">username</saml2:NameID>
    <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <saml2:SubjectConfirmationData NotOnOrAfter="2020-09-22T22:08:08.816Z"
 Recipient="domain-endpoint/_dashboards/_opendistro/_security/saml/acs"/>
    </saml2:SubjectConfirmation>
  </saml2:Subject>
  <saml2:Conditions NotBefore="2020-09-22T21:58:08.816Z"
 NotOnOrAfter="2020-09-22T22:08:08.816Z">
    <saml2:AudienceRestriction>
      <saml2:Audience>domain-endpoint</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml2:Conditions>
  <saml2:AuthnStatement AuthnInstant="2020-09-22T19:54:37.274Z">
    <saml2:AuthnContext>

 <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport<
saml2:AuthnContextClassRef>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
    <saml2:Attribute Name="role" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
      <saml2:AttributeValue
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xs:string">GroupName Match Matches regex ".+" (case-sensitive)
      </saml2:AttributeValue>
    </saml2:Attribute>
  </saml2:AttributeStatement>
</saml2:Assertion>
```

步骤 5：（可选）配置其他设置

在 Additional settings（其他设置）下，配置以下可选字段：

- Subject key（使用者密钥）：您可以将此字段留空，以便将 SAML 断言的 `NameID` 元素用于用户名。如果您的断言不使用此标准元素，而是将用户名作为自定义属性，请在此处指定该属性。
- Roles key（角色密钥）：如果要使用后端角色（推荐），请在此字段指定断言的属性，例如 `role` 或 `group`。这是另一种情况，其中像 SAML 跟踪可以提供帮助。
- 会话上线时间-默认情况下， OpenSearch 控制面板会在 24 小时后将用户注销。通过指定新值，可以将此值配置为 60 到 1440（24 小时）之间的任何数字。

配置好后，请保存域。

第 6 步：更新你的 IdP URLs

如果您在创建域时启用了 SAML 身份验证，则必须在 IdP 中指定临时 URLs 身份才能生成 XML 元数据文件。在域名状态更改为后`Active`，您可以获取正确的域名 URLs 并修改您的 IdP。

要检索 URLs，请选择域并选择操作、编辑安全配置。在 OpenSearch Dashboards/Kibana 的 SAML 身份验证下，您可以找到正确的服务提供商实体 ID 和 SSO。 URLs复制这些值并使用它们来配置您的身份提供商，替换您在步骤 2 中提供的临时 URLs 身份提供商。

步骤 7：将 SAML 用户映射到角色

当您的域名状态为 “激活” 且您的 IdP 配置正确后，请导航至 OpenSearch 控制面板。

- 如果您选择了 SP 启动的 URL，请导航到 *`domain-endpoint`*`/_dashboards`。若要直接登录到特定租户，可将 `?security_tenant=`*`tenant-name`* 附加到 URL。
- 如果选择了 IdP 启动的 URL，请导航到身份提供程序的应用程序目录。

在这两种情况下，请以 SAML 主用户或属于 SAML 主后端角色的用户身份录入。要继续执行步骤 7 中的示例，请以 `jdoe` 或 `admins` 组中 成员录入)。

OpenSearch 仪表板加载后，选择安全、角色。然后，映射角色以允许其他用户访问 OpenSearch 仪表板。

例如，您可能将受信任的同事 `jroe` 映射添加到 `all_access` 和 `security_manager` 角色。您还可以将后端角色 `analysts` 映射添加到 `readall` 和 `opensearch_dashboards_user` 角色。

如果您更喜欢使用 API 而不是 OpenSearch 控制面板，请参阅以下示例请求：

```
PATCH _plugins/_security/api/rolesmapping
[
  {
    "op": "add", "path": "/security_manager", "value": { "users": ["master-user",
 "jdoe", "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/all_access", "value": { "users": ["master-user", "jdoe",
 "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/readall", "value": { "backend_roles": ["analysts"] }
  },
  {
    "op": "add", "path": "/opensearch_dashboards_user", "value": { "backend_roles":
 ["analysts"] }
  }
]
```

配置 SP 和 IdP 发起的身份验证

如果要配置 SP 和 IDP 启动的身份验证，则必须通过身份提供程序执行此操作。例如，在 Okta 中，您可以执行以下步骤：

1. 在您的 SAML 应用程序中，转到 General（通用）、SAML settings（SAML 设置）。
2. 对于 Single sign on URL（单点登录 URL），提供您的 IdP 发起的 SSO URL。例如 `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs/idpinitiated`。
3. 启用 “允许此应用请求其他 SSO URLs”。
4. 在 “可请求的 SSO” 下 URLs，添加一个或多个 SP 发起的 SSO。 URLs例如 `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs`。

配置 SAML 身份验证 (Amazon CLI)

以下 Amazon CLI 命令为现有域上的 OpenSearch 仪表板启用 SAML 身份验证：

```
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"}}'
```

必须转义元数据 XML 中的所有引号和换行符。例如，使用 `<KeyDescriptor use=\"signing\">\n` 而不是 `<KeyDescriptor use="signing">` 和换行符。有关使用 Amazon CLI，请参阅 Amazon CLI 命令参考。

配置 SAML 身份验证（配置 API）

以下对配置 API 的请求为现有域上的 OpenSearch 仪表板启用 SAML 身份验证：

```
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"
    }
  }
}
```

必须转义元数据 XML 中的所有引号和换行符。例如，使用 `<KeyDescriptor use=\"signing\">\n` 而不是 `<KeyDescriptor use="signing">` 和换行符。有关使用配置 API 的详细信息，请参阅OpenSearch 服务 API 参考。

SAML 故障排除

错误	详细信息
不允许您的请求：*/some/path* “”。	验证您提供了正确的 SSO URL （步骤 3）发送给您的身份提供程序。

错误	详细信息
请提供有效的身份提供者元数据文档以启用 SAML。	您的 IdP 元数据文件不符合 SAML 2.0 标准。使用验证工具检查错误。
SAML 配置选项不显示在控制台中。	更新到最新服务软件。
SAML 配置错误：检索 SAML 配置时出现问题，请检查您的设置。	出现这种通用错误的原因很多。 • 检查是否为身份提供商提供了正确的 SP 实体 ID 和 SSO URL。 • 重新生成 IdP 元数据文件，并验证 IdP 实体 ID。将任何更新的元数据添加到 Amazon 控制台。 • 验证您的域访问策略是否允许访问 OpenSearch 仪表板和`_plugins/_security/*` 。通常，我们建议对使用精细访问控制的域采用开放访问策略。 • 有关配置 SAML 的步骤，请咨询身份提供程序的文档。
缺少角色：此用户没有可用的角色，请与您的系统管理员联系。	您已成功进行身份验证，但 SAML 断言中的用户名和任何后端角色未映射到任何角色，因此没有权限。这些映射区分大小写。 系统管理员可以使用类似于 SAML-tracer 的工具来验证 SAML 断言的内容，然后使用以下请求检查角色映射： `GET _plugins/_security/api/rolesmapping`
尝试访问 OpenSearch 控制面板时，您的浏览器会持续重定向或收到 HTTP 500 错误。	如果 SAML 断言包含大量角色，总共约 1,500 个字符，则可能会发生这些错误。例如，如果您传递 80 个角色（其平均长度为 20 个字符），则可能会超过 Web 浏览器中 Cookie 的大小限制。从 2.7 OpenSearch 版开始，SAML 断言支持最多 5000 个字符的角色。

错误	详细信息
您无法注销 ADFS。	ADFS 要求对所有注销请求进行签名，但 OpenSearch 服务不支持。`<SingleLogoutService />` 从 IdP 元数据文件中移除以强制 OpenSearch 服务使用自己的内部注销机制。
`Could not find entity descriptor for __PATH__.`	提供给 OpenSearch 服务的元数据 XML 中提供的 IdP 的实体 ID 与 SAML 响应中提供的实体 ID 不同。要解决这个问题，请确保二者匹配。在您的域上启用 CW 应用程序错误日志，查找错误消息，调试 SAML 集成问题。
`Signature validation failed. SAML response rejected.`	OpenSearch 服务无法使用元数据 XML 中提供的 IdP 证书验证 SAML 响应中的签名。可能是手动错误，也可能是 IdP 已轮换证书。在通过提供给 OpenSearch 服务的元数据 XML 中更新来自您的 IdP 的最新证书。Amazon Web Services Management Console
`__PATH__ is not a valid audience for this response.`	SAML 响应的受众字段与域端点不匹配。要修复此错误，请更新 SP 受众字段，使其与域端点相符。如果已启用自定义端点，则受众字段应与自定义端点匹配。在您的域上启用 CW 应用程序错误日志，查找错误消息，调试 SAML 集成问题。
浏览器响应收到代码为 `Invalid Request Id` 的 HTTP 400 错误。	如果将 IdP 启动的 URL 配置为 *`<DashboardsURL>`* `/_opendistro/_security/saml/acs` 格式，通常会发生此错误。相反，改用 *`<DashboardsURL>`* `/_opendistro/_security/saml/acs/idpinitiated` 格式配置 URL。

错误	详细信息
通过 __PATH__（而不是 __PATH__）接收响应。	SAML 响应的目标字段与以下 URL 格式之一不匹配： • `<DashboardsURL> /_opendistro/_security/saml/acs` • `<DashboardsURL> /_opendistro/_security/saml/acs/idpinitiated` . 根据您使用的登录流程（SP 启动或 IDP 发起的），在与其中一个匹配的目标字段中输入。 OpenSearch URLs
响应具有 `InResponseTo` 属性，但没有 `InResponseTo` 。	您正在使用 IDP 启动的 URL 执行 SP 启动的登录流。改用 SP 启动的 URL。

禁用 SAML 身份验证

禁用 OpenSearch 仪表板的 SAML 身份验证（控制台）

1. 选择域、Actions（操作）和 Edit security configuration（编辑安全配置）。
2. 取消选中启用 SAML 身份验证。
3. 选择保存更改。
4. 域完成处理后，请使用以下请求验证精细访问控制角色映射：

```
GET _plugins/_security/api/rolesmapping
```

禁用仪表板的 SAML 身份验证不会删除 SAML 主用户名（即 SAML 主后端角色） and/or 的映射。如果要删除这些映射，请使用内部用户数据库（如果已启用）登录到控制面板，或使用 API 删除它们：

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "users": [
    "master-user"
  ]
```

```
}
```

IAM Identity Center 可信身份传播支持 Amazon OpenSearch 服务

现在，您可以通过可信身份传播使用集中配置 Amazon 的 IAM Identity Center 委托人（用户和群组）通过 OpenSearch 服务应用程序访问 OpenSearch 域。要启用 Amazon S OpenSearch ervice 的 IAM 身份中心支持，您需要启用 IAM 身份中心。要详细了解如何执行此操作，请参阅什么是 IAM 身份中心？ 。请参阅如何在应用程序中 OpenSearch 将 OpenSearch域关联为数据源？ 了解详情。

您可以使用 OpenSearch 服务控制台、 Amazon Command Line Interface (Amazon CLI) 或配置 IAM 身份中心 Amazon SDKs。

> **Note**
>
> 控制面板（与集群同置）不支持 IAM 身份中心委托人。仅通过集中式 OpenSearch 用户界面（仪表板）支持它们。

注意事项

在将 IAM 身份中心与 Amazon OpenSearch 服务配合使用之前，您必须考虑以下几点：

- IAM 身份中心已在账户中启用。
- OpenSearch 域名版本为 1.3 或更高版本。
- 已在域上启用@@ 细粒度访问控制。
- 域应与 IAM 身份中心实例位于同一区域。
- 域和OpenSearch 应用程序应属于同一个 Amazon 账户。

修改域访问策略

在配置 IAM Identity Center 之前，您必须更新域访问策略或在 OpenSearch 应用程序中为可信身份传播配置的 IAM 角色的权限。

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "IAM Role configured in OpenSearch application"
            },
            "Action": "es:ESHttp*",
            "Resource": "domain-arn/*"
        },
        {
        ... // Any other permissions
        }
    ]
}
```

配置 IAM 身份中心身份验证和授权（控制台）

您可以在域名创建过程中或通过更新现有域来启用 IAM Identity Center 身份验证和授权。根据您选择的具体选项，设置步骤会略有差异。

以下步骤说明了如何在 Amazon S OpenSearch ervice 控制台中为 IAM 身份中心身份验证和授权配置现有域：

1. 在 “域配置” 下，导航到 “安全配置”，选择 “编辑”，然后导航到 “IAM 身份中心身份验证” 部分，然后选择 “启用通过 IAM Identity Center 进行身份验证的 API 访问”。
2. 按如下方式选择 SubjectKey 和角色键。
   - 主题密钥-选择一个 UserId （默认），UserName 然后选择电子邮件以使用相应的属性作为访问域名的委托人。
   - 角色密钥-选择一个 GroupId（默认），然后 GroupName 使用相应的属性值作为与 IdC 委托人关联的所有群组的后端角色。fine-grained-access-control

完成更改后，请保存域。

配置细粒度访问控制

在您的 OpenSearch 域上启用 IAM Identity Center 选项后，您可以通过创建到后端角色的角色映射来配置对 IAM Identity Center 委托人的访问权限。委托人的后端角色值基于 IdC 委托人的组成员资格和 GroupId 或 GroupName的 RolesKey配置。

> **Note**
>
> Amazon OpenSearch 服务可以为单个用户最多支持 100 个群组。如果您尝试使用的实例数量超过允许的数量，则会遇到与 fine-grained-access-control授权处理不一致的问题，并会收到 403error 消息。

配置 IAM 身份中心身份验证和授权 (CLI)

```
  aws opensearch update-domain-config \
      --domain-name my-domain \
      --identity-center-options '{"EnabledAPIAccess": true,
 "IdentityCenterInstanceARN": "instance arn",  "SubjectKey": "UserId/UserName/
UserEmail" , "RolesKey": "GroupId/GroupName"}'
```

在域上禁用 IAM 身份中心身份验证

要在您的 OpenSearch 域上禁用 IAM 身份中心，请执行以下操作：

1. 选择域、Actions（操作）和 Edit security configuration（编辑安全配置）。
2. 取消选中启用通过 IAM 身份中心进行身份验证的 API 访问权限。
3. 选择保存更改。
4. 域完成处理后，移除为 IdC 委托人添加的角色映射

要通过 CLI 禁用 IAM 身份中心，您可以使用以下方法

```
  aws opensearch update-domain-config \
      --domain-name my-domain \
      --identity-center-options '{"EnabledAPIAccess": false}'
```

为控制面板配置 Amazon Cognito 身份验证 OpenSearch

您可以使用 Amazon Cognito 对您在亚马逊 OpenSearch 服务中默认安装的 OpenSearch控制面板进行身份验证和保护。Amazon Cognito 身份验证是可选的，仅适用于使用 OpenSearch 或 Elasticsearch

5.1 或更高版本的域名。如果不配置 Amazon Cognito 身份验证，您仍可使用 基于 IP 的访问策略和代理服务器、HTTP 基本身份验证或 SAML。

大部分身份验证过程都在 Amazon Cognito 中进行，但本节提供了有关配置 Amazon Cognito 资源以使用 OpenSearch 服务域的指南和要求。标准定价适用于所有 Amazon Cognito 资源。

> **Tip**
>
> 首次将域配置为使用 OpenSearch 控制面板的 Amazon Cognito 身份验证时，我们建议您使用控制台。Amazon Cognito 资源的可定制程度极高，并且控制台可以帮助您确定和理解对您重要的功能。

主题

- 先决条件
- 将域配置为使用 Amazon Cognito 身份验证
- 允许经过身份验证的角色
- 配置身份提供商
- （可选）配置精细访问
- （可选）自定义登录页面
- （可选）配置高级安全
- 测试
- 限额
- 常见配置问题
- 禁用控制面板的 Amazon Cognito 身份验证 OpenSearch
- 删除在控制面板中使用 Amazon Cognito 身份验证的域名 OpenSearch

先决条件

在为 OpenSearch 控制面板配置 Amazon Cognito 身份验证之前，您必须满足几个先决条件。OpenSearch 服务控制台有助于简化这些资源的创建，但了解每种资源的用途有助于配置和故障排除。用于仪表板的 Amazon Cognito 身份验证需要以下资源：

- Amazon Cognito 用户池
- Amazon Cognito 身份池

- IAM 角色已附加了 `AmazonOpenSearchServiceCognitoAccess` 策略 (`CognitoAccessForAmazonOpenSearch`)

> **Note**
>
> 用户池和身份池必须在同一个 Amazon Web Services 区域中。您可以使用相同的用户池、身份池和 IAM 角色将控制面板的 Amazon Cognito 身份验证添加到多个 OpenSearch 服务域。要了解更多信息，请参阅the section called “限额”。

关于用户池

用户池有两个主要功能：创建和管理用户目录、让用户注册和登录。有关创建用户池的说明，请参阅 Amazon Cognito 开发者指南中的用户池入门。

在创建要与 S OpenSearch ervice 配合使用的用户池时，请考虑以下几点：

- 您的 Amazon Cognito 用户池必须有一个域名。 OpenSearch 服务使用此域名将用户重定向到登录页面以访问仪表板。除了域名外，用户池不需要任何非默认配置。
- 您必须指定池的必需标准属性——名称、出生日期、电子邮件地址和电话号码等属性。创建用户池之后您将不能更改这些属性，因此此时请选择对您重要的属性。
- 在创建用户池时，请选择用户能否创建自己的账户、账户的最小密码强度以及是否启用多因素身份验证。如果您计划使用外部身份提供商，这些设置无关紧要。从技术上说，您可以启用用户池作为身份提供商，同时启用外部身份提供商，但大多数人更愿意选其中一项。

用户池的 IDs 形式为*`region_ID`*。如果您计划使用 Amazon CLI 或 S Amazon DK 来配置 OpenSearch 服务，请记下 ID。

关于身份池

借助身份池，您可以在用户登录后向用户分配临时性的受限权限角色。有关创建身份池的说明，请参阅 Amazon Cognito 开发者指南中的身份池控制台概述。创建用于 OpenSearch 服务的身份池时，请考虑以下几点：

- 如果您使用 Amazon Cognito 控制台，则必须选中 Enable access to unauthenticated identities (启用未经验证的身份的访问权限) 复选框来创建身份池。在您创建身份池并配置 OpenSearch 服务域之后，Amazon Cognito 会禁用此设置。

- 您无需向身份池添加外部身份提供商。当您将 OpenSearch 服务配置为使用 Amazon Cognito 身份验证时，它会将身份池配置为使用您刚刚创建的用户池。
- 创建身份池后，您必须选择未经身份验证和经过身份验证的 IAM 角色。这些角色会指定用户在登录之前和之后拥有的访问策略。如果您使用 Amazon Cognito 控制台，它可以为您创建这些角色。在创建经过身份验证的角色后，请记下 ARN，采用以下格式：`arn:aws:iam::123456789012:role/Cognito_identitypoolnameAuth_Role`。

身份池的 IDs 形式为`region:ID-ID-ID-ID-ID`。如果您计划使用 Amazon CLI 或 S Amazon DK 来配置 OpenSearch 服务，请记下 ID。

关于 CognitoAccessForAmazonOpenSearch 角色

OpenSearch 服务需要权限才能配置 Amazon Cognito 用户和身份池并使用它们进行身份验证。为此`AmazonOpenSearchServiceCognitoAccess`，您可以使用 Amazon 托管策略。 `AmazonESCognitoAccess`是一项旧政策，在服务重命名为 Amazon OpenSearch 服务`AmazonOpenSearchServiceCognitoAccess`时已被取代。这两项策略都提供了启用亚马逊 Cognito 身份验证所需的最低亚马逊 Cognito 权限。有关策略的详细信息，请参阅AmazonOpenSearchServiceCognitoAccess《Amazon 托管策略参考指南》。

如果您使用控制台创建或配置您的 OpenSearch 服务域，它会为您创建一个 IAM 角色并将`AmazonOpenSearchServiceCognitoAccess`策略（如果是 Elasticsearch 域，则为该`AmazonESCognitoAccess`策略）附加到该角色。此角色的默认名称为`CognitoAccessForAmazonOpenSearch`。

角色权限策略`AmazonOpenSearchServiceCognitoAccess`以及`AmazonESCognitoAccess`两者都允许 S OpenSearch ervice 在所有身份和用户池上完成以下操作：

- 操作：`cognito-idp:DescribeUserPool`
- 操作：`cognito-idp:CreateUserPoolClient`
- 操作：`cognito-idp:DeleteUserPoolClient`
- 操作：`cognito-idp:UpdateUserPoolClient`
- 操作：`cognito-idp:DescribeUserPoolClient`
- 操作：`cognito-idp:AdminInitiateAuth`
- 操作：`cognito-idp:AdminUserGlobalSignOut`
- 操作：`cognito-idp:ListUserPoolClients`
- 操作：`cognito-identity:DescribeIdentityPool`

- 操作：`cognito-identity:SetIdentityPoolRoles`
- 操作：`cognito-identity:GetIdentityPoolRoles`

如果您使用 Amazon CLI 或其中一个 Amazon SDKs，则必须在配置 OpenSearch 服务域时创建自己的角色、附加策略并为此角色指定 ARN。角色必须拥有以下信任关系：

JSON

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

有关说明，请参阅 I AM 用户指南中的创建角色以向 Amazon 服务委派权限和添加和删除 IAM 身份权限。

将域配置为使用 Amazon Cognito 身份验证

完成先决条件后，您可以将 OpenSearch 服务域配置为使用 Amazon Cognito 控制面板。

> **Note**
>
> 亚马逊 Cognito 并非全部可用。 Amazon Web Services 区域有关支持的区域列表；，请参阅 Amazon Cognito 的服务终端节点。在 Amazon Cognito 中，您无需使用与服务相同的区域。OpenSearch

配置 Amazon Cognito 身份验证（控制台）

由于它为您创建`CognitoAccessForAmazonOpenSearch`角色，因此控制台提供了最简单的配置体验。除了标准的 OpenSearch 服务权限外，您还需要以下一组权限才能使用控制台创建使用 Amazon Cognito 身份验证控制面板的 OpenSearch 域。

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"
    }
  ]
}
```

有关向身份（用户、用户组或角色）添加权限的说明，请参阅添加 IAM 身份权限（控制台）。

如果 `CognitoAccessForAmazonOpenSearch` 已存在，则您需要较少的权限：

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"
    }
  ]
}
```

要为控制面板配置 Amazon Cognito 身份验证（控制台）

1. 在家中打开亚马逊 OpenSearch 服务控制台 https://console.aws.amazon.com/aos//。
2. 在 Domains（域）下，选择要配置的域。
3. 选择 Actions（操作）、Edit security configuration（编辑安全配置）。
4. 选择启用 Amazon Cognito 身份验证。
5. 对于区域，选择 Amazon Web Services 区域 包含您的 Amazon Cognito 用户池和身份池的。
6. 对于 Cognito user pool（Cognito 用户池），选择一个用户池或创建一个。有关更多信息，请参阅 the section called “关于用户池”。
7. 对于 Cognito identity pool（Cognito 身份池），选择一个身份池或创建一个。有关更多信息，请参阅 the section called “关于身份池”。

> **Note**
>
> Create user pool（创建用户池）和 Create identity pool（创建身份池）链接会将您定向到 Amazon Cognito 控制台，并需要您手动创建这些资源。此过程不是自动的。有关更多信息，请参阅 the section called “先决条件”。

8. 于 IAM role name（IAM 角色名称），请使用默认值 `CognitoAccessForAmazonOpenSearch`（推荐）或输入新名称。有关更多信息，请参阅 the section called “关于 CognitoAccessForAmazonOpenSearch 角色”。
9. 选择保存更改。

在您的域完成处理后，请参阅the section called “允许经过身份验证的角色”和the section called “配置身份提供商”，了解更多配置步骤。

配置 Amazon Cognito 身份验证 (Amazon CLI)

使用`--cognito-options`参数配置您的 OpenSearch 服务域。`create-domain` 和 `update-domain-config` 命令均使用以下语法：

```
--cognito-options Enabled=true,UserPoolId="user-pool-id",IdentityPoolId="identity-pool-id",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonOpenSearch"
```

示例

以下示例使用 `CognitoAccessForAmazonOpenSearch` 角色在启用用于 Amazon Cognito 的身份验证的 `us-east-1` 区域中创建一个域并向该域提供对 `Cognito_Auth_Role` 的访问权限：

```
aws opensearch create-domain --domain-name my-domain --region us-east-1 --access-policies '{ "Version":"2012-10-17", "Statement":[{"Effect":"Allow","Principal":{"AWS": ["arn:aws:iam::123456789012:role/Cognito_Auth_Role"]},"Action":"es:ESHttp*","Resource":"arn:aws:es:us-east-1:123456789012:domain/*" }]}' --engine-version "OpenSearch_1.0" --cluster-config InstanceType=m4.xlarge.search,InstanceCount=1 --ebs-options EBSEnabled=true,VolumeSize=10 --cognito-options Enabled=true,UserPoolId="us-east-1_123456789",IdentityPoolId="us-east-1:12345678-1234-1234-1234-123456789012",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonOpenSearch"
```

在您的域完成处理后，请参阅the section called “允许经过身份验证的角色”和the section called “配置身份提供商”，了解更多配置步骤。

配置亚马逊 Cognito 身份验证 ()Amazon SDKs

Amazon SDKs （Android 和 iOS 除外 SDKs）支持《亚马逊 OpenSearch 服务 API 参考》中定义的所有`UpdateDomainConfig`操作，包括`CreateDomain`和操作的`CognitoOptions`参数。有关安装和使用的更多信息 Amazon SDKs，请参阅Amazon 软件开发套件。

在您的域完成处理后，请参阅the section called “允许经过身份验证的角色”和the section called “配置身份提供商”，了解更多配置步骤。

允许经过身份验证的角色

默认情况下，您按照中的指南配置的经过身份验证的 IAM 角色the section called “关于身份池”没有访问 OpenSearch 控制面板所需的权限。您必须为该角色提供额外权限。

> **Note**
>
> 如果您配置了精细访问控制，并使用开放的或基于 IP 的访问策略，则可以跳过此步骤。

您可以将这些权限包含在基于身份的策略中，但是除非您希望经过身份验证的用户有权访问所有 OpenSearch 服务域，否则将基于资源的策略附加到单个域是更好的方法。

对于 `Principal`，请为您按照 the section called “关于身份池” 中的指导原则配置的、经过 Cognito 身份验证的角色指定 ARN。

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":[
               "arn:aws:iam::123456789012:role/Cognito_identitypoolnameAuth_Role"
            ]
         },
```

```
            "Action":[
               "es:ESHttp*"
            ],
            "Resource":"arn:aws:es:region:123456789012:domain/domain-name/*"
         }
      ]
   }
```

有关向 OpenSearch 服务域添加基于资源的策略的说明，请参阅the section called “配置访问策略”。

配置身份提供商

当您将域配置为使用控制面板的 Amazon Cognito 身份验证时， OpenSearch 服务会向用户池添加应用程序客户端，并将用户池作为身份验证提供者添加到身份池中。

> **Warning**
>
> 请勿重命名或删除应用程序客户端。

根据用户池的配置情况，您可能需要手动创建用户账户，或者用户可能能够自行创建。如果这些设置可接受，您无需再采取进一步操作。但是，很多人都倾向于使用外部身份提供商。

要启用 SAML 2.0 身份提供商，您必须提供 SAML 元数据文档。要启用 Login with Amazon、Facebook 和 Google 等社交身份提供商，您必须拥有一个应用程序 ID 和来自提供商的应用程序密钥。您可以启用身份提供商的任意组合。

配置用户池的最简单方法是使用 Amazon Cognito 控制台。有关说明，请参阅 Amazon Cognito 开发者指南中的使用第三方身份提供商登录用户池和使用应用程序客户端进行应用程序特定设置。

（可选）配置精细访问

您可能已经注意到，默认身份池设置为每位登录同一 IAM 角色的用户分配了同一 IAM 角色 (`Cognito_identitypoolAuth_Role`)，这意味着每个用户都可以访问相同的 Amazon 资源。如果要将精细访问控制与 Amazon Cognito 结合使用，例如，如果您希望组织的分析人员具有对多个索引的只读访问权限，同时希望开发人员具有对所有索引的写入访问权限您有两种选择：

- 创建用户组并配置您的身份提供商，以根据用户的身份验证令牌选择 IAM 角色（推荐）。
- 配置身份提供商，以根据一个或多个规则选择 IAM 角色。

有关包含精细访问控制的演练，请参阅the section called “教程：使用 Cognito 身份验证的精细访问控制”。

> **Important**
>
> 与默认角色一样，Amazon Cognito 必须是每个附加角色的信任关系的一部分。有关详细信息，请参阅 Amazon Cognito 开发者指南中的为角色映射创建角色。

用户组和令牌

当您创建用户组时，您会为该组的成员选择 IAM 角色。有关创建群组的信息，请参阅 Amazon Cognito 开发者指南中的向用户池中添加群组。

创建一个或多个用户组后，您可以对您的身份验证提供商进行配置，以便为用户分配其组的角色，而不是身份池的默认角色。选择从令牌选择角色，然后选择使用默认身份验证角色或拒绝以指定身份池如何处理不属于组的用户。

规则

规则实质上是由 Amazon Cognito 按顺序评估的一系列 `if` 语句。例如，如果一个用户的电子邮件地址包含 `@corporate`，Amazon Cognito 则为该用户分配 `Role_A`。如果一个用户的电子邮件地址包含 `@subsidiary`，则为该用户分配 `Role_B`。否则，它会为该用户分配默认的经过身份验证的角色。

要了解更多信息，请参阅 Amazon Cognito 开发者指南中的使用基于规则的映射为用户分配角色。

（可选）自定义登录页面

您可以使用 Amazon Cognito 控制台上传自定义徽标并对登录页进行 CSS 更改。有关说明和 CSS 属性的完整列表，请参阅 Amazon Cognito 开发者指南中的自定义托管用户界面（经典）品牌。

（可选）配置高级安全

Amazon Cognito 用户池支持高级安全功能，如多重验证、盗用凭证检查和自适应身份验证。要了解更多信息，请参阅《亚马逊 Cognito 开发者指南》中的使用 Ama zon Cognito 用户池安全功能。

测试

对配置感到满意后，请验证用户体验是否符合的预期。

访问 OpenSearch 控制面板

1. 在 Web 浏览器中导航到 `https://opensearch-domain/_dashboards`。若要直接登录到特定租户，请将 `?security_tenant=tenant-name` 附加到 URL。
2. 使用首选凭证进行登录。
3. OpenSearch 仪表板加载后，至少配置一种索引模式。控制面板使用这些模式来标识要分析的索引。输入 `*`，选择 Next step (下一步)，然后选择 Create index pattern (创建索引模式)。
4. 要搜索或查看您的数据，请选择 Discover (发现)。

如果此过程的任何步骤失败，请参阅the section called “常见配置问题”了解故障排除信息。

限额

Amazon Cognito 对于它的许多资源都有软限制。如果您想为大量 OpenSearch 服务域启用控制面板身份验证，请查看 Amazon Cognito 中的配额，并根据需要请求提高限制。

每个 OpenSearch 服务域都会向用户池添加一个应用程序客户端，从而向身份池中添加一个身份验证提供商。如果您为超过 10 个域启用 OpenSearch 控制面板身份验证，则可能会遇到 “每个身份池的最大 Amazon Cognito 用户池提供者数量” 限制。如果您超过限制，则您尝试配置为使用控制面板的 Amazon Cognito 身份验证的任何 OpenSearch 服务域都可能陷入正在处理的配置状态。

常见配置问题

下表列出了常见的配置问题和解决方案。

配置 OpenSearch 服务

事务	解决方案
`OpenSearch Service can't create the role` (console)	您没有正确的 IAM 权限。请添加the section called “配置 Amazon Cognito 身份验证（控制台）”中指定的权限。
`User is not authorized to perform: iam:PassRole on resource CognitoAccessForAmazonOpenSearch` (console)	您没有该CognitoAccessForAmazonOpenSearch角色的`iam:PassRole` 权限。将以下策略附加到您的账户中： JSON {

事务	解决方案

	``
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"iam:PassRole"
	],
	"Resource": "arn:aws:
	iam:: 123456789012:role/service-role/
	CognitoAccessForAmazonOpenSearch "
	}
	]
	}
	``
	您也可以附加 `IAMFullAccess` 策略。

`User is not authorized to perform: cognito-identity:ListIdentityPools on resource`	您没有 Amazon Cognito 的读取权限。将 `AmazonCognitoReadOnly` 策略附加到您的账户。
`An error occurred (ValidationException) when calling the CreateDomain operation: OpenSearch Service must be allowed to use the passed role`	OpenSearch `CognitoAccessForAmazonOpenSearch` 角色的信任关系中未指定服务。检查您的角色是否使用了the section called “关于 CognitoAccessForAmazonOpenSearch 角色”指定的信任关系。也可以使用控制台来配置 Amazon Cognito 身份验证。控制台将为您创建一个角色。
`An error occurred (ValidationException) when calling the CreateDomain operation: User is not authorized to perform: cognito-idp: action on resource: user pool`	`--cognito-options` 中指定的角色无权访问 Amazon Cognito。检查该角色是否已附加 Amazon 托管`AmazonOpenSearchServiceCognitoAccess` 策略。也可以使用控制台来配置 Amazon Cognito 身份验证。控制台将为您创建一个角色。

事务	解决方案
`An error occurred (ValidationException) when calling the CreateDomain operation : User pool does not exist`	OpenSearch 服务找不到用户池。确认您已创建一个用户池并具有正确的 ID。要查找 ID，你可以使用 Amazon Cognito 控制台或以下命令： Amazon CLI `aws cognito-idp list-user-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateDomain operation : IdentityPool not found`	OpenSearch 服务找不到身份池。确认您已创建一个用户池并具有正确的 ID。要查找 ID，你可以使用 Amazon Cognito 控制台或以下命令： Amazon CLI `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`	用户池没有域名。您可以使用 Amazon Cognito 控制台或以下 Amazon CLI 命令配置一个： `aws cognito-idp create-user-pool-domain --domain name --user-pool-id id`

访问 OpenSearch 仪表板

事务	解决方案
登录页不显示我的首选身份提供商。	检查您是否按照中的指定为 OpenSearch 服务应用程序客户端启用了身份提供商the section called “配置身份提供商”。
登录页看上去与我的组织无关。	请参阅the section called “（可选）自定义登录页面”。
我的登录凭证不起作用。	检查您是否按照the section called “配置身份提供商”中指定的方式配置了身份提供商。 如果您使用用户池作为身份提供商，请检查 Amazon Cognito 控制台上是否存在账户。

事务	解决方案
OpenSearch 仪表板要么根本无法加载，要么无法正常工作。	Amazon Cognito 身份验证角色需要阈 (/*) 的 `es:ESHttp*` 权限才能访问和使用控制面板。检查您是否按照the section called “允许经过身份验证的角色”中指定的方式添加了访问策略。
当我从一个选项卡注销 OpenSearch 仪表板时，其余选项卡会显示一条消息，指出刷新令牌已被撤销。	当您在使用 Amazon Cognito 身份验证时退出 OpenSearch 控制面板会话时， OpenSearch 服务会运行一项AdminUserGlobalSignOut操作，使您退出所有活动的 OpenSearch 控制面板会话。
`Invalid identity pool configuration. Check assigned IAM roles for this pool.`	Amazon Cognito 不具有代表经过身份验证的用户代入 IAM 角色的权限。修改角色的信任关系以包括： 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" } } }] }`

事务	解决方案
`Token is not from a supported provider of this identity pool.`	在从用户池删除应用程序客户端时，会出现这个不常见的错误。尝试在新的浏览器会话中打开控制面板。

禁用控制面板的 Amazon Cognito 身份验证 OpenSearch

使用以下过程可对控制面板禁用 Amazon Cognito 身份验证。

要为控制面板（控制台）禁用 Amazon Cognito 身份验证

1. 打开亚马逊 OpenSearch 服务控制台。
2. 在 Domains（域）下，选择要配置的域。
3. 选择 Actions（操作）、Edit security configuration（编辑安全配置）。
4. 取消选择启用 Amazon Cognito 身份验证。
5. 选择保存更改。

> **Important**
> 如果您不再需要 Amazon Cognito 用户池和身份池，请将其删除。否则，您将继续产生费用。

删除在控制面板中使用 Amazon Cognito 身份验证的域名 OpenSearch

为防止使用控制面板的 Amazon Cognito 身份验证的域名停留在 “处理中” 的配置状态，请在删除其关联的 Amazon Cognito 用户和身份池之前删除 OpenSearch 服务域。

使用适用于 Amazon Service OpenSearch 的服务相关角色

亚马逊 OpenSearch 服务使用 Amazon Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 S OpenSearch ervice 直接相关。服务相关角色是 Service 预定义的，拥有该 OpenSearch 服务代表您调用其他服务代表您调用其他服务代表您调用其他 Amazon 服务权限。

服务相关角色可让您更轻松地设置 OpenSearch 服务，因为您不必手动添加必要的权限。OpenSearch Service 定义其服务相关角色的权限，除非另外定义，否则只有 S OpenSearch ervice 可以代入该角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其他 IAM 实体。有关服务相关角色和权限策略的更新，请参阅 Amazon S OpenSearch ervice 文档历史记录。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 配合使用的Amazon 服务并查找 Service - Linked Rol es 列为 Yes 的服务。选择是和链接，查看该服务的服务相关角色文档。

主题

- 使用服务相关角色创建 VPC 域和直接查询数据源
- 使用服务相关角色创建 OpenSearch 无服务器集合
- 使用服务相关角色创建 OpenSearch 摄取管道

使用服务相关角色创建 VPC 域和直接查询数据源

亚马逊 OpenSearch 服务使用 Amazon Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 S OpenSearch ervice 直接相关。服务相关角色是 Service 预定义的，拥有该 OpenSearch 服务代表您调用其他服务代表您调用其他服务代表您调用其他 Amazon 服务权限。

OpenSearch Service 使用名为的服务相关角色 AWSServiceRoleForAmazonOpenSearchService，它将提供该角色为域或直接查询数据源启用 VPC 访问权限所需的最低 Amazon EC2 和 Elastic Load Balancing 权限所需的最低 Amazon 和 Elastic Load Balancing 权限。

旧式 Elasticsearch 角色

Amazon OpenSearch 服务使用名`AWSServiceRoleForAmazonOpenSearchService`为的服务相关角色。您的账户还可以包含一个名为 `AWSServiceRoleForAmazonElasticsearchService` 的旧式服务相关角色，它与已经弃用的 Elasticsearch API 终端节点一起使用。

如果您的账户中不存在此旧式 Elasticsearch 角色，S OpenSearch ervice 会在您首次创建域时自动创建一个新的 OpenSearch 服务相关角色。 OpenSearch 否则，您的账户将继续使用此 Elasticsearch 角色。为使这种自动创建成功，您必须具有 `iam:CreateServiceLinkedRole` 操作的权限。

权限

`AWSServiceRoleForAmazonOpenSearchService` 服务相关角色信任以下服务代入该角色：

- `opensearchservice.amazonaws.com`

名为 S AmazonOpenSearchServiceRolePolicyer OpenSearch vice 的角色权限策略允许 Service 对指定资源完成以下操作：

- 操作：* 上的 acm:DescribeCertificate
- 操作：cloudwatch:PutMetricData 上的 *
- 操作：ec2:CreateNetworkInterface 上的 *
- 操作：ec2:DeleteNetworkInterface 上的 *
- 操作：ec2:DescribeNetworkInterfaces 上的 *
- 操作：ec2:ModifyNetworkInterfaceAttribute 上的 *
- 操作：ec2:DescribeSecurityGroups 上的 *
- 操作：ec2:DescribeSubnets 上的 *
- 操作：ec2:DescribeVpcs 上的 *
- 操作：针对所有网络接口和 VPC 端点执行 ec2:CreateTags
- 操作：* 上的 ec2:DescribeTags
- 操作：针对ec2:CreateVpcEndpoint所有安全组 VPCs、子网和路由表以及所有 VPC 端点执行（当请求包含标签）时，针对所有 VPC 端点执行 OpenSearchManaged=true
- 操作：针对ec2:ModifyVpcEndpoint所有安全组 VPCs、子网和路由表以及所有 VPC 端点执行（当请求包含标签）时，针对所有 VPC 端点执行 OpenSearchManaged=true
- 操作：当请求包含标签 OpenSearchManaged=true 时，针对所有端点执行 ec2:DeleteVpcEndpoints
- 操作：* 上的 ec2:AssignIpv6Addresses
- 操作：ec2:UnAssignIpv6Addresses 上的 *
- 操作：elasticloadbalancing:AddListenerCertificates 上的 *
- 操作：* 上的 elasticloadbalancing:RemoveListenerCertificates

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

创建 服务相关角色

您无需手动创建服务相关角色。当您使用创建已启用 VPC 的域或直接查询数据源时 Amazon Web Services Management Console，Service 将为您创建 OpenSearch 服务相关角色。为使这种自动创建成功，您必须具有 iam:CreateServiceLinkedRole 操作的权限。

您还可以使用 IAM 控制台、IAM CLI 或 IAM; API 来手动创建服务相关角色。有关更多信息，请参阅 IAM 用户指南 中的创建服务相关角色。

编辑服务相关角色

OpenSearch Service 不允许您编辑`AWSServiceRoleForAmazonOpenSearchService`服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 服务相关角色

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样就没有未被主动监控或维护的未使用实体。但是，您必须先清除您的服务相关角色，然后才能手动删除它。

清除 服务相关角色

必须先确认服务相关角色没有活动会话并删除该角色使用的任何资源，然后才能使用 IAM 删除服务相关角色。

在 IAM 控制台中检查服务相关角色是否具有活动会话

1. 的服务 Amazon Web Services Management Console 相关角色权限https://console.aws.amazon.com/iam/。
2. 在 IAM 控制台的导航窗格中，选择角色。然后选择 `AWSServiceRoleForAmazonOpenSearchService` 角色的名称（不是复选框）。
3. 在所选角色的 Summary 页面上，选择 Access Advisor 选项卡。
4. 在访问顾问选项卡查看服务相关角色的近期活动。

> **Note**
>
> 如果您不确定 S OpenSearch ervice 是否正在使用`AWSServiceRoleForAmazonOpenSearchService`角色，可以尝试删除该角色。如果服务正在使用该角色，则删除操作会失败，并且您可以查看正在使用该角色的资源。如果该角色正在使用，则您必须等待会话结束，然后才能删除该角色，并且/或者删除使用该角色的资源。您无法撤销服务相关角色对会话的权限。

手动删除服务相关角色

从 IAM 控制台、AP Amazon I 或 CLI中删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的删除服务相关角色。

使用服务相关角色创建 OpenSearch 无服务器集合

OpenSearch 无服务器使用 Amazon Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 S OpenSearch ervice 直接相关。服务相关角色是Service 预定义的，拥有该 OpenSearch 服务代表您调用其他服务代表您调用其他服务代表您调用其他 Amazon 服务权限。

OpenSearch Serverless 使用名为 Serverless 的服务相关角色 AWSServiceRoleForAmazonOpenSearchServerless，它将提供该角色向您的账户发布与无服务器相关的权限 CloudWatch。与之关联的角色权限策略名 AWSServiceRoleForAmazonOpenSearchServerless 为`AmazonOpenSearchServerlessServiceRolePolicy`。有关该策略的更多信息，请参阅AmazonOpenSearchServerlessServiceRolePolicy《Amazon 托管策略参考指南》。

Serverless 的服务相关角色权限 OpenSearch

OpenSearch Serverless 使用名为的服务相关角色 AWSServiceRoleForAmazonOpenSearchServerless， OpenSearch Serverless 可以代表您调用 Amazon 服务。

AWSServiceRoleForAmazonOpenSearchServerless 服务相关角色信任以下服务代入该角色：

- `observability.aoss.amazonaws.com`

名为 Serverl OpenSearch ess 的角色权限策略`AmazonOpenSearchServerlessServiceRolePolicy`允许 Serverless 对指定资源完成以下操作：

- 操作：针对所有 Amazon 资源执行 `cloudwatch:PutMetricData`

> **Note**
>
> 该策略包含条件键`{"StringEquals": {"cloudwatch:namespace": "AWS/AOSS"}}`，这意味着服务相关角色只能向`AWS/AOSS` CloudWatch命名空间发送指标数据。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为 OpenSearch Serverless 创建服务相关角色

您无需手动创建服务相关角色。当您在 Amazon Web Services Management Console、或 Amazon API 中创建 OpenSearch 无服务器集合时 Amazon CLI， OpenSearch Serverless 会为您创建与服务相关的角色。

> **Note**
>
> 当您首次创建集合时，必须在基于身份的策略中为您分配 `iam:CreateServiceLinkedRole`。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您创建 OpenSearch 无服务器集合时，Ser OpenSearch verless 会再次为您创建服务相关角色。

您也可以使用 IAM 控制台为 Amazon Serverless（A mazon OpenSearch Serverless）使用案例创建服务相关角色。在 Amazon CLI 或 Amazon API 中，使用服务名称创建服务相关角色：`observability.aoss.amazonaws.com`

```
aws iam create-service-linked-role --aws-service-name
 "observability.aoss.amazonaws.com"
```

有关更多信息，请参阅 IAM 用户指南 中的创建服务相关角色。如果您删除了此服务相关角色，可以使用同样的过程再次创建角色。

编辑 Serverless 的 OpenSearch 服务相关角色

OpenSearch Serverless 不允许您编辑 AWSServiceRoleForAmazonOpenSearchServerless 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 Serverless 的 OpenSearch 服务相关角色

如果不再需要使用某个需要服务相关角色的特征或服务，我们建议您删除该角色。这将防止您拥有未被主动监控或维护的未使用实体。但是，您必须先清除服务相关角色的资源，然后才能手动删除它。

要删除 AWSServiceRoleForAmazonOpenSearchServerless，必须先删除您的 Amazon Web Services 账户所有 OpenSearch 无服务器集合。

> **Note**
>
> 在您尝试删除资源时，如果 OpenSearch Serverless 正在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟后重试。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台 Amazon CLI、或 Amazon API 删除 AWSServiceRoleForAmazonOpenSearchServerless服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

OpenSearch Serverless 服务相关角色的支持区域

OpenSearch Serverless 支持在每个提供 Ser OpenSearch verless 的区域中使用 AWSServiceRoleForAmazonOpenSearchServerless 服务相关角色。有关支持的区域列表，请参阅中的 Amazon OpenSearch 无服务器终端节点和配额。Amazon Web Services 一般参考

使用服务相关角色创建 OpenSearch 摄取管道

Amazon OpenSearch Ingestion 使用 Amazon Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种与 OpenSearch Ingestion 直接关联的独特类型的 IAM 角色。服务相关角色是由 OpenSearch Ingestion 预定义的，拥有该服务代表您调用其他服务代表您调用其他服务代表您调用其他 Amazon 服务权限。

OpenSearch Ingestion 使用名为的服务相关角色 AWSServiceRoleForAmazonOpenSearchIngestionService，除非您使用自管理 VPC，在这种情况下，Ingestion 将使用名为的服务相关角色。AWSServiceRoleForOpensearchIngestionSelfManagedVpce附加策略提供该角色在您的账户与 OpenSearch Ingestion 之间创建虚拟私有云 (VPC) 以及向您的账户发布 CloudWatch指标所需的权限。

权限

`AWSServiceRoleForAmazonOpenSearchIngestionService` 服务相关角色信任以下服务代入该角色：

- osis.amazon.com

名为 Ingestion 的角色权限策略AmazonOpenSearchIngestionServiceRolePolicy允 OpenSearch 许 Ingestion 对指定资源完成以下操作：

- 操作：* 上的 ec2:DescribeSubnets
- 操作：ec2:DescribeSecurityGroups 上的 *
- 操作：ec2:DeleteVpcEndpoints 上的 *
- 操作：ec2:CreateVpcEndpoint 上的 *
- 操作：ec2:DescribeVpcEndpoints 上的 *
- 操作：ec2:CreateTags 上的 arn:aws:ec2:*:*:network-interface/*
- 操作：cloudwatch:PutMetricData 上的 cloudwatch:namespace": "AWS/OSIS"

AWSServiceRoleForOpensearchIngestionSelfManagedVpce 服务相关角色信任以下服务代入该角色：

- self-managed-vpce.osis.amazon.com

名为 Ingestion 的角色权限策略OpenSearchIngestionSelfManagedVpcePolicy允 OpenSearch 许 Ingestion 对指定资源完成以下操作：

- 操作：* 上的 ec2:DescribeSubnets
- 操作：ec2:DescribeSecurityGroups 上的 *
- 操作：ec2:DescribeVpcEndpoints 上的 *
- 操作：cloudwatch:namespace": "AWS/OSIS" 上的 cloudwatch:PutMetricData

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为 OpenSearch Ingestion 创建服务相关角色

您无需手动创建服务相关角色。当您使用、或 Amazon API 中创建 OpenSearch 摄取管道时 Amazon Web Services Management Console， OpenSearch Ingestion 将为您创建服务相关角色。 Amazon CLI

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您创建 OpenSearch 摄取管道时，In OpenSearch gestion 将再次为您创建服务相关角色。

编辑 Ingestion 服务相关角色 OpenSearch

OpenSearch Ingestion 不允许您编辑`AWSServiceRoleForAmazonOpenSearchIngestionService`服务相关角色。创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 Ingestion 服务相关角色 OpenSearch

如果您不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。但是，您必须先清除服务相关角色的资源，然后才能手动删除它。

清除服务相关角色

必须先删除服务相关角色使用的所有资源，然后才能使用 IAM 删除该角色。

> **Note**
>
> 在您尝试删除资源时，如果 OpenSearch Ingestion 正在使用该角色，删除操作可能会失败。如果发生这种情况，请等待几分钟后重试。

删除 OpenSearch 或角色使用的摄取**`AWSServiceRoleForAmazonOpenSearchIngestionService`**资源**`AWSServiceRoleForOpensearchIngestionSelfManagedVpce`**

1. 导航到 Amazon S OpenSearch ervice 控制台，选择摄取。
2. 删除所有管道。有关说明，请参阅the section called “删除管道”。

删除 Ingestion 服务相关角色 OpenSearch

您可以使用 OpenSearch Ingestion 控制台删除服务相关角色。

删除服务相关角色 (控制台)

1. 导航到 IAM 控制台。

2. 选择角色并搜索AWSServiceRoleForAmazonOpenSearchIngestionService或AWSServiceRoleForOpensearchIngestio色。
3. 选择角色，选择删除。

Amazon S OpenSearch ervice 示例代码

本章包含关于使用 Amazon S OpenSearch ervice 的常见示例代码：使用各种编程语言对 HTTP 请求进行签名，压缩 HTTP 请求正文，以及使用 Amazon Service 创建域。 Amazon SDKs

主题

- Elasticsearch 客户端兼容性
- 在 Amazon Service OpenSearch 中压缩 HTTP 请求
- 使用 Amazon SDKs 与 Amazon OpenSearch 服务互动

Elasticsearch 客户端兼容性

最新版本的 Elasticsearch 客户端可能包括许可证或版本检查，这些检查会人为破坏兼容性。下表包括有关使用这些客户端的哪些版本与 S OpenSearch ervice 实现最佳兼容的建议。

> **Important**
>
> 这些客户端版本已过时，且不会使用最新的依赖项（包括 log4J）进行更新。我们强烈建议尽可能使用客户端的 OpenSearch 版本。

客户端	建议的版本
Java 低级别 REST 客户端	7.13.4
Java 高级别 REST 客户端	7.13.4
Python Elasticsearch 客户端	7.13.4
RuUT Elasticsearch 客户端	7.13.3
Node.js Elasticsearch 客户端	7.13.0

在 Amazon Service OpenSearch 中压缩 HTTP 请求

您可以使用 gzip 压缩压缩在 Amazon S OpenSearch ervice 域中压缩 HTTP 请求和响应。Gzip 压缩可帮助您缩小文档的大小，降低带宽利用率和延迟时间，从而提高传输速度。

所有运行 OpenSearch 或 Elasticsearch 6.0 或更高版本的域都支持 Gzip 压缩。一些 OpenSearch 客户端具有对 gzip 压缩的内置支持，许多编程语言都具有简化过程的库。

启用 gzip 压缩

不要与类似的 OpenSearch 设置混淆，`http_compression.enabled`它特定于 S OpenSearch ervice 并启用或禁用域上的 gzip 压缩。域名正在运行 OpenSearch 或 Elasticsearch 7. x 默认启用了 gzip 压缩，而运行 Elasticsearch 的域名则为 6。 x 默认将其禁用。

要启用 gzip 压缩，请发送以下请求：

```
PUT _cluster/settings
{
  "persistent" : {
    "http_compression.enabled": true
  }
}
```

必须解压缩对 `_cluster/settings` 的请求，因此您可能需要使用单独的客户端或标准 HTTP 请求来更新群集设置。

要确认您是否成功启用了 gzip 压缩，请发送以下请求：

```
GET _cluster/settings?include_defaults=true
```

确保在响应中看到以下设置：

```
...
"http_compression": {
  "enabled": "true"
}
...
```

必需的标头

当包含 gzip 压缩的请求主体时，请保留标准的 `Content-Type: application/json` 标头，然后添加 `Content-Encoding: gzip` 标头。要接受 gzip 压缩的响应，也请添加 `Accept-Encoding: gzip` 标头。如果 OpenSearch 客户端支持 gzip 压缩，它可能会自动包含这些标头。

示例代码 (Python 3)

以下示例使用 opensearch-py 执行压缩并发送请求。此代码使用您的 IAM 证书对请求进行签名。

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3

host = '' # e.g. my-test-domain.us-east-1.es.amazonaws.com
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Create the client.
search = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    http_compress = True, # enables gzip compression for request bodies
    connection_class = RequestsHttpConnection
)

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

# Send the request.
print(search.index(index='movies', id='1', body=document, refresh=True))

# print(search.index(index='movies', doc_type='_doc', id='1', body=document,
 refresh=True))
```

或者，您可以指定适当的头，自己压缩请求正文，并使用标准的 HTTP 库，比如请求。此代码使用 HTTP 基本凭据对请求进行签名，如果您使用访问权限的精细控制，则域会提供支持。

```
import requests
import gzip
import json

base_url = '' # The domain with https:// and a trailing slash. For example, https://my-test-domain.us-east-1.es.amazonaws.com/
auth = ('master-user', 'master-user-password') # For testing only. Don't store
 credentials in code.

headers = {'Accept-Encoding': 'gzip', 'Content-Type': 'application/json',
           'Content-Encoding': 'gzip'}

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

# Compress the document.
compressed_document = gzip.compress(json.dumps(document).encode())

# Send the request.
path = 'movies/_doc?refresh=true'
url = base_url + path
response = requests.post(url, auth=auth, headers=headers, data=compressed_document)
print(response.status_code)
print(response.text)
```

使用 Amazon SDKs 与 Amazon OpenSearch 服务互动

此部分包含关于如何使用的示例，与 Amazon S OpenSearch ervice 配置 API 进行交互的示例。Amazon SDKs 这些代码示例演示了如何创建、更新和删除S OpenSearch ervice 域。

Java

此部分包含的版本 1 和版本 2 适用于 Java 的 Amazon SDK。

Version 2

此示例使用版本 2 中的OpenSearchClientBuilder构造函数 适用于 Java 的 Amazon SDK 来创建 OpenSearch 域、更新其配置并将其删除。取消注释对 `waitForDomainProcessing` 的调用（并注释对 `deleteDomain` 的调用）以允许域上线并可供使用。

```
package com.example.samples;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.opensearch.OpenSearchClient;
import software.amazon.awssdk.services.opensearch.model.ClusterConfig;
import software.amazon.awssdk.services.opensearch.model.EBSOptions;
import software.amazon.awssdk.services.opensearch.model.CognitoOptions;
import software.amazon.awssdk.services.opensearch.model.NodeToNodeEncryptionOptions;
import software.amazon.awssdk.services.opensearch.model.CreateDomainRequest;
import software.amazon.awssdk.services.opensearch.model.CreateDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainRequest;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainResponse;
import software.amazon.awssdk.services.opensearch.model.OpenSearchException;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */

public class OpenSearchSample {

    public static void main(String[] args) {

     String domainName = "my-test-domain";

     // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.

        OpenSearchClient client = OpenSearchClient.builder()
```

```
          // Unnecessary, but lets you use a region different than your default.
          .region(Region.US_EAST_1)
          // Unnecessary, but if desired, you can use a different provider chain.
          .credentialsProvider(DefaultCredentialsProvider.create())
                    .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        //waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        //waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon OpenSearch Service domain with the specified options.
     * Some options require other Amazon Web Services resources, such as an Amazon
Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain you want to create
     */

    public static void createDomain(OpenSearchClient client, String domainName) {

     // Create the request and set the desired configuration options

        try {

            ClusterConfig clusterConfig = ClusterConfig.builder()
                    .dedicatedMasterEnabled(true)
                    .dedicatedMasterCount(3)
                    // Small, inexpensive instance types for testing. Not
recommended for production.
                    .dedicatedMasterType("t2.small.search")
                    .instanceType("t2.small.search")
                    .instanceCount(5)
                    .build();

            // Many instance types require EBS storage.
```

```
            EBSOptions ebsOptions = EBSOptions.builder()
                    .ebsEnabled(true)
                    .volumeSize(10)
                    .volumeType("gp2")
                    .build();

            NodeToNodeEncryptionOptions encryptionOptions =
 NodeToNodeEncryptionOptions.builder()
                    .enabled(true)
                    .build();

            CreateDomainRequest createRequest = CreateDomainRequest.builder()
                    .domainName(domainName)
                    .engineVersion("OpenSearch_1.0")
                    .clusterConfig(clusterConfig)
                    .ebsOptions(ebsOptions)
                    .nodeToNodeEncryptionOptions(encryptionOptions)
                    // You can uncomment this line and add your account ID, a
 username, and the
                    // domain name to add an access policy.
                    // .accessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")
                    .build();

            // Make the request.
            System.out.println("Sending domain creation request...");
            CreateDomainResponse createResponse =
 client.createDomain(createRequest);
            System.out.println("Domain status:
 "+createResponse.domainStatus().toString());
            System.out.println("Domain ID:
 "+createResponse.domainStatus().domainId());


        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Updates the configuration of an Amazon OpenSearch Service domain with the
```

```
    * specified options. Some options require other Amazon Web Services resources,
such as an
    * Amazon Cognito user pool and identity pool, whereas others require just an
    * instance type or instance count.
    *
    * @param client
    *            The client to use for the requests to Amazon OpenSearch Service
    * @param domainName
    *            The name of the domain to update
    */

   public static void updateDomain(OpenSearchClient client, String domainName) {

    // Updates the domain to use three data instances instead of five.
       // You can uncomment the Cognito line and fill in the strings to enable
Cognito
       // authentication for OpenSearch Dashboards.

       try {

           ClusterConfig clusterConfig = ClusterConfig.builder()
                   .instanceCount(5)
                   .build();

           CognitoOptions cognitoOptions = CognitoOptions.builder()
                   .enabled(true)
                   .userPoolId("user-pool-id")
                   .identityPoolId("identity-pool-id")
                   .roleArn("role-arn")
                   .build();

           UpdateDomainConfigRequest updateRequest =
UpdateDomainConfigRequest.builder()
                   .domainName(domainName)
                   .clusterConfig(clusterConfig)
                   //.cognitoOptions(cognitoOptions)
                   .build();

           System.out.println("Sending domain update request...");
           UpdateDomainConfigResponse updateResponse =
client.updateDomainConfig(updateRequest);
           System.out.println("Domain config:
"+updateResponse.domainConfig().toString());
```

```
        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
     * several minutes.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to delete
     */

    public static void deleteDomain(OpenSearchClient client, String domainName) {

        try {

            DeleteDomainRequest deleteRequest = DeleteDomainRequest.builder()
                    .domainName(domainName)
                    .build();

            System.out.println("Sending domain deletion request...");
            DeleteDomainResponse deleteResponse =
client.deleteDomain(deleteRequest);
            System.out.println("Domain status: "+deleteResponse.toString());


        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Waits for the domain to finish processing changes. New domains typically take
15-30 minutes
     * to initialize, but can take longer depending on the configuration. Most
updates to existing domains
     * take a similar amount of time. This method checks every 15 seconds and
finishes only when
```

```
     * the domain's processing status changes to false.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to check
     */

    public static void waitForDomainProcessing(OpenSearchClient client, String
 domainName) {
        // Create a new request to check the domain status.
        DescribeDomainRequest describeRequest = DescribeDomainRequest.builder()
                .domainName(domainName)
                .build();

        // Every 15 seconds, check whether the domain is processing.
        DescribeDomainResponse describeResponse =
 client.describeDomain(describeRequest);
        while (describeResponse.domainStatus().processing()) {
            try {
                System.out.println("Domain still processing...");
                TimeUnit.SECONDS.sleep(15);
                describeResponse = client.describeDomain(describeRequest);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        // Once we exit that loop, the domain is available
        System.out.println("Amazon OpenSearch Service has finished processing
 changes for your domain.");
        System.out.println("Domain description: "+describeResponse.toString());
    }
}
```

Version 1

此示例使用的版本 1 中的AWSElasticsearchClientBuilder构造函数创建旧式 Elasticsearch 域，更新其配置和对其进行删除。 适用于 Java 的 Amazon SDK 取消注释对 `waitForDomainProcessing` 的调用（并注释对 `deleteDomain` 的调用）以允许域上线并可供使用。

```
package com.amazonaws.samples;
```

```
import java.util.concurrent.TimeUnit;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticsearch.AWSElasticsearch;
import com.amazonaws.services.elasticsearch.AWSElasticsearchClientBuilder;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainResult;
import
 com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.EBSOptions;
import com.amazonaws.services.elasticsearch.model.ElasticsearchClusterConfig;
import com.amazonaws.services.elasticsearch.model.ResourceNotFoundException;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigRequest;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigResult;
import com.amazonaws.services.elasticsearch.model.VolumeType;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */

public class OpenSearchSample {

    public static void main(String[] args) {

        final String domainName = "my-test-domain";

        // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.
        final AWSElasticsearch client = AWSElasticsearchClientBuilder
                .standard()
                // Unnecessary, but lets you use a region different than your
 default.
                .withRegion(Regions.US_WEST_2)
                // Unnecessary, but if desired, you can use a different provider
 chain.
```

```
                .withCredentials(new DefaultAWSCredentialsProviderChain())
                .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon OpenSearch Service domain with the specified options.
     * Some options require other Amazon Web Services resources, such as an Amazon
Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain you want to create
     */
    private static void createDomain(final AWSElasticsearch client, final String
domainName) {

        // Create the request and set the desired configuration options
        CreateElasticsearchDomainRequest createRequest = new
CreateElasticsearchDomainRequest()
                .withDomainName(domainName)
                .withElasticsearchVersion("7.10")
                .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                        .withDedicatedMasterEnabled(true)
                        .withDedicatedMasterCount(3)
                        // Small, inexpensive instance types for testing. Not
recommended for production
                        // domains.
                        .withDedicatedMasterType("t2.small.elasticsearch")
                        .withInstanceType("t2.small.elasticsearch")
                        .withInstanceCount(5))
                // Many instance types require EBS storage.
                .withEBSOptions(new EBSOptions()
                        .withEBSEnabled(true)
                        .withVolumeSize(10)
```

```
                        .withVolumeType(VolumeType.Gp2));
                // You can uncomment this line and add your account ID, a username,
 and the
                // domain name to add an access policy.
                // .withAccessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")


        // Make the request.
        System.out.println("Sending domain creation request...");
        CreateElasticsearchDomainResult createResponse =
 client.createElasticsearchDomain(createRequest);
        System.out.println("Domain creation response from Amazon OpenSearch
 Service:");
        System.out.println(createResponse.getDomainStatus().toString());
    }

    /**
     * Updates the configuration of an Amazon OpenSearch Service domain with the
     * specified options. Some options require other Amazon Web Services resources,
 such as an
     * Amazon Cognito user pool and identity pool, whereas others require just an
     * instance type or instance count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain to update
     */
    private static void updateDomain(final AWSElasticsearch client, final String
 domainName) {
        try {
            // Updates the domain to use three data instances instead of five.
            // You can uncomment the Cognito lines and fill in the strings to enable
 Cognito
            // authentication for OpenSearch Dashboards.
            final UpdateElasticsearchDomainConfigRequest updateRequest = new
 UpdateElasticsearchDomainConfigRequest()
                    .withDomainName(domainName)
                    // .withCognitoOptions(new CognitoOptions()
                            // .withEnabled(true)
                            // .withUserPoolId("user-pool-id")
```

```
                            // .withIdentityPoolId("identity-pool-id")
                            // .withRoleArn("role-arn")
                    .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                            .withInstanceCount(3));

            System.out.println("Sending domain update request...");
            final UpdateElasticsearchDomainConfigResult updateResponse = client
                    .updateElasticsearchDomainConfig(updateRequest);
            System.out.println("Domain update response from Amazon OpenSearch
Service:");
            System.out.println(updateResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }

    /**
     * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
     * several minutes.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to delete
     */
    private static void deleteDomain(final AWSElasticsearch client, final String
domainName) {
        try {
            final DeleteElasticsearchDomainRequest deleteRequest = new
DeleteElasticsearchDomainRequest()
                    .withDomainName(domainName);

            System.out.println("Sending domain deletion request...");
            final DeleteElasticsearchDomainResult deleteResponse =
client.deleteElasticsearchDomain(deleteRequest);
            System.out.println("Domain deletion response from Amazon OpenSearch
Service:");
            System.out.println(deleteResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }

    /**
```

```
     * Waits for the domain to finish processing changes. New domains typically take
 15-30 minutes
     * to initialize, but can take longer depending on the configuration. Most
 updates to existing domains
     * take a similar amount of time. This method checks every 15 seconds and
 finishes only when
     * the domain's processing status changes to false.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to check
     */
    private static void waitForDomainProcessing(final AWSElasticsearch client, final
 String domainName) {
        // Create a new request to check the domain status.
        final DescribeElasticsearchDomainRequest describeRequest = new
 DescribeElasticsearchDomainRequest()
                .withDomainName(domainName);

        // Every 15 seconds, check whether the domain is processing.
        DescribeElasticsearchDomainResult describeResponse =
 client.describeElasticsearchDomain(describeRequest);
        while (describeResponse.getDomainStatus().isProcessing()) {
            try {
                System.out.println("Domain still processing...");
                TimeUnit.SECONDS.sleep(15);
                describeResponse =
 client.describeElasticsearchDomain(describeRequest);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        // Once we exit that loop, the domain is available
        System.out.println("Amazon OpenSearch Service has finished processing
 changes for your domain.");
        System.out.println("Domain description response from Amazon OpenSearch
 Service:");
        System.out.println(describeResponse.toString());
    }
}
```

Python

此示例使用中的OpenSearchService低级 Python 客户端创建域、更新其配置和对其进行删除。Amazon SDK for Python (Boto)

```
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)
```

节点

此示例使用适用于 JavaScript SDK 版本 3 客户OpenSearch 端创建域、更新其配置和对其进行删除。Node.js

```
var {
    OpenSearchClient,
    CreateDomainCommand,
    DescribeDomainCommand,
    UpdateDomainConfigCommand,
    DeleteDomainCommand
} = require("@aws-sdk/client-opensearch");
var sleep = require('sleep');

var client = new OpenSearchClient();

var domainName = 'my-test-domain'

// Create a new domain, update its configuration, and delete it.
createDomain(client, domainName)
waitForDomainProcessing(client, domainName)
updateDomain(client, domainName)
waitForDomainProcessing(client, domainName)
deleteDomain(client, domainName)

async function createDomain(client, domainName) {
    // Creates an Amazon OpenSearch Service domain with the specified options.
    var command = new CreateDomainCommand({
        DomainName: domainName,
        EngineVersion: 'OpenSearch_1.0',
        ClusterConfig: {
        'InstanceType': 't2.small.search',
        'InstanceCount': 5,
        'DedicatedMasterEnabled': 'True',
        'DedicatedMasterType': 't2.small.search',
        'DedicatedMasterCount': 3
        },
        EBSOptions:{
            'EBSEnabled': 'True',
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies: "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/my-test-domain/*
\"}]}",
         NodeToNodeEncryptionOptions:{
            'Enabled': 'True'
```

```
        }
    });
    const response = await client.send(command);
    console.log("Creating domain...");
    console.log(response);
}

async function updateDomain(client, domainName) {
    // Updates the domain to use three data nodes instead of five.
    var command = new UpdateDomainConfigCommand({
        DomainName: domainName,
        ClusterConfig: {
        'InstanceCount': 3
        }
    });
    const response = await client.send(command);
    console.log('Sending domain update request...');
    console.log(response);
}

async function deleteDomain(client, domainName) {
    // Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes.
    var command = new DeleteDomainCommand({
        DomainName: domainName
    });
    const response = await client.send(command);
    console.log('Sending domain deletion request...');
    console.log(response);
}

async function waitForDomainProcessing(client, domainName) {
    // Waits for the domain to finish processing changes.
    try {
        var command = new DescribeDomainCommand({
            DomainName: domainName
        });
        var response = await client.send(command);

        while (response.DomainStatus.Processing == true) {
            console.log('Domain still processing...')
            await sleep(15000) // Wait for 15 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
```

```
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        // Once we exit the loop, the domain is available.
        console.log('Amazon OpenSearch Service has finished processing changes for your
 domain.');
        console.log('Domain description:');
        console.log(response);

    } catch (error) {
        if (error.name === 'ResourceNotFoundException') {
            console.log('Domain not found. Please check the domain name.');
            }
    };
}
```

在 Amazon OpenSearch 服务中为数据编制索引

由于亚马逊 OpenSearch 服务使用 REST API，因此有许多方法可以为文档编制索引。您可以使用标准客户端 (如 curl) 或可发送 HTTP 请求的任何编程语言。为了进一步简化与之交互的过程，S OpenSearch ervice 提供了多种编程语言的客户端。高级用户可以直接跳至the section called “将流数据加载到 OpenSearch 服务中”。

我们强烈建议您使用 Amazon OpenSearch Ingestion 来摄取数据，这是在服务中内置的完全托管的数据收集器。 OpenSearch 有关更多信息，请参阅 Amazon OpenSearch Ingesti on。

有关索引的简介，请参阅OpenSearch文档。

索引的命名限制

OpenSearch 服务索引有以下命名限制：

- 所有字母必须为小写形式。
- 索引名称不能以 _ 或 - 开头。
- 索引名称不能包含空格、逗号、:、"、*、+、/、\、|、?、#、> 或 <。

不要在索引、类型或文档 ID 名称中包含敏感信息。 OpenSearch 服务在其统一资源标识符 (URIs) 中使用这些名称。服务器和应用程序经常记录 HTTP 请求，如果 URIs 包含敏感信息，可能会导致不必要的数据泄露：

```
2018-10-03T23:39:43 198.51.100.14 200 "GET https://opensearch-domain/dr-jane-doe/flu-
patients-2018/202-555-0100/ HTTP/1.1"
```

即使您没有查看关联的 JSON 文档的权限，您可以从这条假的日志线推断，Doe 医生的一个电话号码为 202-555-0100 的患者在 2018 年患了流感。

如果 Ser OpenSearch vice 在索引名称中检测到真实或感知的 IP 地址（例如`my-index-12.34.56.78.91`），则会屏蔽该 IP 地址。调用 `_cat/indices` 会产生以下响应：

```
green open my-index-x.x.x.x.91    soY19tBERoKo71WcEScidw 5 1 0 0   2kb  1kb
```

为避免不必要的混淆，请避免在索引名称中包括 IP 地址。

减小响应大小

来自_index和的回复_bulk APIs 包含大量信息。此信息可用于对请求进行问题排查或实施重试逻辑，但可使用大量带宽。在本示例中，对 32 字节文档编制索引会生成 339 字节响应（包括标头）：

```
PUT opensearch-domain/more-movies/_doc/1
{"title": "Back to the Future"}
```

响应

```
{
  "_index": "more-movies",
  "_type": "_doc",
  "_id": "1",
  "_version": 4,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 2,
    "failed": 0
  },
  "_seq_no": 3,
  "_primary_term": 1
}
```

此响应大小可能看起来很小，但如果您每天索引 1,000,000 个文档（大约每秒 11.5 个文档），则每个响应 339 字节的下载流量为每月 10.17 GB。

如果担心数据传输成本，请使用filter_path参数来减小 OpenSearch 服务响应的大小，但要注意不要筛选出识别或重试失败请求所需的字段。这些字段会因客户端而异。该filter_path参数适用于所有 Service REST APIs，但对于经常调用的 OpenSearch 服务 REST 特别有用，例如_index和_bulk APIs： APIs

```
PUT opensearch-domain/more-movies/_doc/1?filter_path=result,_shards.total
{"title": "Back to the Future"}
```

响应

```
{
```

```
  "result": "updated",
  "_shards": {
    "total": 2
  }
}
```

不同于包括字段，您可以使用 - 前缀排除字段。filter_path 还支持通配符：

```
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"}
```

响应

```
{
  "errors": false,
  "items": [
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    },
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    }
  ]
}
```

索引编解码器

索引编解码器决定如何压缩索引上存储的字段并将其存储在磁盘上。索引编解码器由静态 index.codec 设置控制，静态设置指定压缩算法。此设置会影响索引分片大小和操作性能。

有关支持的编解码器及其性能特性的列表，请参阅文档中的支持的编解码器。 OpenSearch

在选择索引编解码器时，请注意以下事项：

- 为避免更改现有索引的编解码器设置所带来的难题，请在使用新的编解码器设置之前，在非生产环境中测试具有代表性的工作负载。有关更多信息，请参阅更改索引编解码器。
- 不能将 Zstandard 压缩编解码器（`"index.codec": "zstd"` 或 `"index.codec": "zstd_no_dict"`）用于 k-NN 或安全分析索引。

将流数据加载到 Amazon OpenSearch 服务

您可以使用 OpenSearch Ingestion 将流数据直接加载到您的亚马逊 OpenSearch 服务域中，无需使用第三方解决方案。要将数据发送到 OpenSearch Ingestion，您需要配置数据生成器，该服务会自动将数据传输到您指定的域或集合。要开始使用 OpenSearch Ingestion，请参阅。the section called “教程：将数据摄取到集合”

您仍然可以使用其他来源加载流数据，例如 Amazon Data Firehose 和 Amazon CloudWatch Logs，它们内置了对 OpenSearch 服务的支持。其他用户（如 Amazon S3、Amazon Kinesis Data Streams 和 Amazon DynamoDB）使用 Amazon Lambda 函数作为事件处理程序。Lambda 函数响应新数据的方式是处理数据并将其流式传输到域。

> **Note**
>
> Lambda 支持多种常用编程语言，并且在大多数 Amazon Web Services 区域中都可用。有关更多信息，请参阅 Amazon Lambda 开发人员指南中的 Lambda 入门和 Amazon Web Services 一般参考 中的 Amazon 服务端点。

从 OpenSearch Ingestion 加载流数据

您可以使用 Amazon OpenSearch Ingestion 将数据加载到 OpenSearch 服务域中。您可以将数据生成器配置为将数据发送到 OpenSearch Ingestion，它会自动将数据传送到您指定的集合。您还可以将 OpenSearch Ingestion 配置为在交付数据之前对其进行转换。有关更多信息，请参阅 Amazon OpenSearch Ingestion。

从 Amazon S3 表中加载流数据

您可以使用 Lambda 将数据从 Amazon S3 发送到您的 OpenSearch 服务域。到达 S3 存储桶的新数据将触发事件通知到 Lambda，这将运行自定义代码以执行编制索引。

这种流式传输数据的方式极其灵活。可以为对象元数据编制索引，或者如果对象是纯文本，则对对象正文的部分元素进行解析和编制索引。此节包含一些简单的 Python 示例代码，这些代码使用正则表达式解析日志文件并为匹配项编制索引。

先决条件

继续操作之前，必须具有以下资源。

先决条件	描述
Amazon S3 存储桶	有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的创建您的第一个 S3 存储桶。存储桶必须与您的 OpenSearch 服务域位于同一区域。
OpenSearch 服务域	Lambda 函数处理数据之后数据的目的地。有关更多信息，请参阅 the section called “ 创建 OpenSearch 服务域”。

创建 Lambda 部署程序包

部署程序包为 ZIP 或 JAR 文件，其中包含代码及其依赖项。此节包括 Python 示例代码。对于其他编程语言，请参阅 Amazon Lambda 开发人员指南中的 Lambda 部署程序包。

1. 创建目录。在此示例中，我们使用名称 `s3-to-opensearch`。
2. 在名为 `sample.py` 的目录中创建一个文件：

```
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)
```

编辑 `region` 和 `host` 的变量。

3. 安装 pip（如果您尚未安装，则将依赖项安装到 `package` 目录：

```
cd s3-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

所有 Lambda 执行环境都已安装 Boto3，因此无需将其包含在部署程序包中。

4. 打包应用程序代码和依赖项：

```
cd package
zip -r ../lambda.zip .

cd ..
zip -g lambda.zip sample.py
```

创建 Lambda 函数

创建部署程序包之后，可以创建 Lambda 函数。创建函数时，选择名称、运行时 (例如，Python 3.8) 和 IAM 角色。IAM 角色定义对函数的权限。有关详细说明，请参阅 Amazon Lambda 开发人员指南中的通过控制台创建 Lambda 函数。

此示例假定使用的是控制台。选择 Python 3.9 以及具有 S3 读取权限和 OpenSearch 服务写入权限的角色，如以下屏幕截图所示：

在创建此函数后，必须添加一个触发器。在此示例中，我们希望代码在日志文件到达 S3 存储桶中时执行：

1. 选择添加触发器并选择 S3。
2. 选择存储桶。
3. 对于 Event type (事件类型)，选择 PUT。
4. 对于 Prefix (前缀)，键入 `logs/`。
5. 对于后缀，键入 `.log`。
6. 确认递归调用警告，然后选择添加。

最后，可以上传部署程序包：

1. 选择上载自和 .zip 文件，然后按照提示上传部署程序包。
2. 上载完成后，编辑 Runtime 设置并更改处理程序为 `sample.handler`。此设置告知 Lambda 在触发之后应执行的文件 (`sample.py`) 和方法 (`handler`)。

此时，您拥有一整套资源：一个用于存储日志文件的存储桶、一个在向存储桶中添加日志文件时运行的函数、执行解析和索引的代码，以及一个用于搜索和可视化的 OpenSearch 服务域。

测试 Lambda 函数

在创建此函数之后，可以通过将文件上传到 Amazon S3 存储桶来测试此函数。使用以下示例日志行创建一个名为 `sample.log` 的文件：

```
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"
```

将文件上传到 S3 存储桶的 `logs` 文件夹。有关说明，请参阅 Amazon Simple Storage Service 用户指南中的将对象上传到存储桶。

然后使用 OpenSearch 服务控制台或 OpenSearch 仪表板验证`lambda-s3-index`索引是否包含两个文档。还可以发出标准搜索请求：

```
GET https://domain-name/lambda-s3-index/_search?pretty
{
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vTYXaWIBJWV_TTkEuSDg",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.91",
          "message" : "GET /some-file.jpg",
          "timestamp" : "10/Oct/2000:14:56:14 -0700"
        }
      },
      {
```

```
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vjYmaWIBJWV_TTkEuCAB",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.90",
          "message" : "PUT /some-file.jpg",
          "timestamp" : "10/Oct/2000:13:55:36 -0700"
        }
      }
    ]
  }
}
```

从 Amazon Kinesis Data Streams 加载流数据

您可以将流数据从 Kinesis Data Streams 加载 OpenSearch 到服务。到达此数据流的新数据将向 Lambda 触发事件通知，这将运行自定义代码以执行索引编制。此节包括一些简单的 Python 示例代码。

先决条件

继续操作之前，必须具有以下资源。

先决条件	描述
Amazon Kinesis Data Stream	Lambda 函数的事件源。要了解更多信息，请参阅 Kinesis Data Streams。
OpenSearch 服务域	Lambda 函数处理数据之后数据的目的地。有关更多信息，请参阅 the section called “ 创建 OpenSearch 服务域”

IAM 角色	此角色必须具有基本的 OpenSearch 服务、Kinesis 和 Lambda 权限，例如：
	
	JSON
	
	`{`
	`"Version": "2012-10-17",`
	`"Statement": [`

先决条件	描述

	``
	    {
	      "Effect": "Allow",
	      "Action": [
	        "es:ESHttpPost",
	        "es:ESHttpPut",
	        "logs:CreateLogGroup",
	        "logs:CreateLogStream",
	        "logs:PutLogEvents",
	        "kinesis:GetShardIterator",
	        "kinesis:GetRecords",
	        "kinesis:DescribeStream",
	        "kinesis:ListStreams"
	      ],
	      "Resource": "*"
	    }
	  ]
	}
	``
	
	角色必须拥有以下信任关系：
	
	JSON
	
	``
	{
	  "Version": "2012-10-17",
	  "Statement": [
	    {
	      "Effect": "Allow",
	      "Principal": {
	        "Service": "lambda.amazonaws.com"
	      },
	      "Action": "sts:AssumeRole"
	    }
	  ]
	}
	``
	
	要了解更多信息，请参阅 IAM 用户手册中的创建 IAM 角色。

创建 Lambda 函数

按照the section called “创建 Lambda 部署程序包”中的说明操作，但创建一个名为 `kinesis-to-opensearch` 的目录并对 `sample.py` 使用以下代码：

```
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.'
```

编辑 `region` 和 `host` 的变量。

安装 pip——如果您尚未安装，则使用以下命令安装依赖项：

```
cd kinesis-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

然后按照the section called “创建 Lambda 函数”中的说明操作，但指定the section called “先决条件”中的 IAM 角色和以下触发器设置：

- Kinesis stream：您的 Kinesis stream
- 批处理大小：100
- 起始位置：时间范围

有关更多信息，请参阅 Amazon Kinesis Data Streams 开发人员指南中的什么是 Amazon Kinesis Data Streams？。

此时，您已拥有一整套资源：Kinesis 数据流、在流接收新数据并索引该数据之后运行的函数，以及用于搜索和可视化的 OpenSearch 服务域。

测试 Lambda 函数

创建此函数后，可以通过使用 Amazon CLI将新记录添加到数据流来测试它：

```
aws kinesis put-record --stream-name test --data "My test data." --partition-key
 partitionKey1 --region us-west-1
```

然后使用 OpenSearch 服务控制台或 OpenSearch 仪表板验证是否`lambda-kine-index`包含文档。还可使用以下请求：

```
GET https://domain-name/lambda-kine-index/_search
{
  "hits" : [
    {
      "_index": "lambda-kine-index",
      "_type": "_doc",
      "_id":
 "shardId-000000000000:49583511615762699495012960821421456686529436680496087042",
      "_score": 1,
      "_source": {
```

```
        "timestamp": 1523648740.051,
        "message": "My test data.",
        "id":
 "shardId-000000000000:49583511615762699495012960821421456686529436680496087042"
      }
    }
  ]
}
```

从 Amazon DynamoDB 表中加载流数据

您可以使用 Amazon Lambda 将数据从亚马逊 DynamoDB 发送到您的 OpenSearch 服务域。到达数据库表的新数据将触发事件通知到 Lambda，这将运行自定义代码以执行编制索引。

先决条件

继续操作之前，必须具有以下资源。

先决条件	描述
DynamoDB 表	此表包含源数据。有关更多信息，请参阅 Amazon DynamoDB 开发人员指南中的 DynamoDB Tables 中的基本操作。 该表必须与您的 OpenSearch 服务域位于同一区域，并且必须将直播设置为 “新图像”。要了解更多信息，请参阅启用流。
OpenSearch 服务域	Lambda 函数处理数据之后数据的目的地。有关更多信息，请参阅 the section called “ 创建 OpenSearch 服务域”。
IAM 角色	此角色必须具有基本的 OpenSearch 服务、DynamoDB 和 Lambda 执行权限，例如： JSON `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "es:ESHttpPost",`

先决条件	描述

	"es:ESHttpPut",
	"dynamodb:DescribeStream",
	"dynamodb:GetRecords",
	"dynamodb:GetShardIterator",
	"dynamodb:ListStreams",
	"logs:CreateLogGroup",
	"logs:CreateLogStream",
	"logs:PutLogEvents"
	],
	"Resource": "*"
	}
	]
	}
	
	角色必须拥有以下信任关系：
	
	JSON
	
	{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Principal": {
	"Service": "lambda.amazonaws.com"
	},
	"Action": "sts:AssumeRole"
	}
	]
	}
	
	要了解更多信息，请参阅 IAM 用户手册中的创建 IAM 角色。

创建 Lambda 函数

按照the section called “创建 Lambda 部署程序包”中的说明操作，但创建一个名为 `ddb-to-opensearch` 的目录并对 `sample.py` 使用以下代码：

```
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.'
```

编辑 `region` 和 `host` 的变量。

安装 pip——如果您尚未安装，则使用以下命令安装依赖项：

```
cd ddb-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

然后按照the section called “创建 Lambda 函数”中的说明操作，但指定the section called “先决条件”中的 IAM 角色和以下触发器设置：

- 表：DynamoDB 表
- 批处理大小：100
- 起始位置：时间范围

要了解更多信息，请参阅 Amazon DynamoDB 开发人员指南中使用 DynamoDB Streams 和 Lambda 处理新项目。

此时，您已拥有一整套资源：用于存放源数据的 DynamoDB 表、表更改的 DynamoDB 流、在源数据更改并索引这些更改后运行的函数，以及用于搜索和可视化的服务域。 OpenSearch

测试 Lambda 函数

创建此函数后，可以通过使用 Amazon CLI将新项目添加到 DynamoDB 表来测试它：

```
aws dynamodb put-item --table-name test --item '{"director": {"S": "Kevin
 Costner"},"id": {"S": "00001"},"title": {"S": "The Postman"}}' --region us-west-1
```

然后使用 OpenSearch 服务控制台或 OpenSearch 仪表板验证是否`lambda-index`包含文档。还可使用以下请求：

```
GET https://domain-name/lambda-index/_doc/00001
{
    "_index": "lambda-index",
    "_type": "_doc",
    "_id": "00001",
    "_version": 1,
    "found": true,
    "_source": {
        "director": {
            "S": "Kevin Costner"
        },
        "id": {
            "S": "00001"
        },
        "title": {
            "S": "The Postman"
        }
    }
```

```
}
```

从 Amazon Data Firehose 加载流数据

Firehose 支持将 OpenSearch 服务作为送货目的地。有关如何将流数据加载到 OpenSearch 服务的说明，请参阅《亚马逊数据 Fireh ose 开发者指南》中的创建 Kinesis Data Firehos e 传送流 OpenSearch 和为目的地选择服务。

在将数据加载到 S OpenSearch ervice 之前，可能需要对数据执行转换。要了解有关使用 Lambda 函数执行此任务的更多信息，请参阅此同一指南中的 Amazon Kinesis Data Firehose 数据转换。

在您配置传输流时，Firehose 具有 “一键式” IAM 角色，该角色为其提供向 OpenSearch 服务发送数据、在 Amazon S3 上备份数据以及使用 Lambda 转换数据所需的资源访问权限。由于手动创建此类角色的过程非常复杂，我们建议使用提供的角色。

正在加载来自亚马逊的流媒体数据 CloudWatch

您可以使用 CloudWatch CloudWatch 日志订阅将流数据从 Logs 加载到您的 OpenSearch 服务域。有关 Amazon CloudWatch 订阅的信息，请参阅通过订阅实时处理日志数据。有关配置信息，请参阅《亚马逊 CloudWatch开发者指南》中的将 CloudWatch 日志数据流式传输到亚马逊 OpenSearch 服务。

加载来自的流数据 Amazon IoT

Amazon IoT 您可以使用规则发送数据。要了解更多信息，请参阅《Amazon IoT 开发者指南》中的OpenSearch操作。

使用 Logstash 将数据加载到亚马逊 OpenSearch 服务

Logstash（Logstash OSS）的开源版本为使用批量 API 将数据上传到您的亚马逊服务域提供了一种便捷的方式。 OpenSearch 该服务支持所有标准的 Logstash 输入插件，包括 Amazon S3 输入插件。OpenSearch 服务支持logstash-output-opensearch输出插件，该插件同时支持基本身份验证和 IAM 证书。该插件适用于 Logstash OSS 版本 8.1 及更低版本。

配置

Logstash 配置因域使用的身份验证类型不同而不同。

无论使用何种身份验证方法，都必须在配置文件的输出部分中将 `ecs_compatibility` 设置为 `disabled`。Logstash 8.0 引入了一项突破性更改，其中所有插件都默认以 ECS 兼容模式运行。必须覆盖此默认值才能保持旧有的行为。

精细访问控制配置

如果您的 OpenSearch 服务域使用带有 HTTP 基本身份验证的精细访问控制，则配置与任何其他 OpenSearch集群类似。此示例配置文件从开源版本的 Filebeat (Filebeat OSS) 获取其输入。

```
input {
  beats  {
    port => 5044
  }
}

output {
  opensearch {
    hosts       => "https://domain-endpoint:443"
    user        => "my-username"
    password    => "my-password"
    index       => "logstash-logs-%{+YYYY.MM.dd}"
    ecs_compatibility => disabled
    ssl_certificate_verification => false
  }
}
```

配置因 Beats 应用程序和用例而异，但您的 Filebeat OSS 配置可能如下所示：

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /path/to/logs/dir/*.log
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
setup.ilm.enabled: false
setup.ilm.check_exists: false
setup.template.settings:
  index.number_of_shards: 1
output.logstash:
  hosts: ["logstash-host:5044"]
```

IAM 配置

如果您的网域使用基于 IAM 的域访问策略或主用户的精细访问控制，则必须使用 IAM 凭证签署所有向 Ser OpenSearch vice 发出的请求。以下基于身份的策略将所有 HTTP 请求授予域的子资源。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:aws-account-id:domain/domain-name/*"
    }
  ]
}
```

要设置 Logstash 配置，请将您的配置文件更改为使用插件进行输出。此示例配置文件从 S3 存储桶中的文件获取其输入：

```
input {
  s3 {
    bucket => "amzn-s3-demo-"
    region => "us-east-1"
  }
}

output {
  opensearch {
    hosts => ["domain-endpoint:443"]
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      }
      index  => "logstash-logs-%{+YYYY.MM.dd}"
      ecs_compatibility => disabled
```

```
  }
}
```

如果您不想在配置文件中提供 IAM 凭证，则可以将其导出（或运行 `aws configure`）：

```
export AWS_ACCESS_KEY_ID="your-access-key"
export AWS_SECRET_ACCESS_KEY="your-secret-key"
export AWS_SESSION_TOKEN="your-session-token"
```

如果您的 OpenSearch 服务域位于 VPC 中，则 Logstash OSS 计算机必须能够连接到 VPC 并能够通过 VPC 安全组访问该域。有关更多信息，请参阅 the section called “关于 VPC 域的访问策略”。

在亚马逊 OpenSearch 服务中搜索数据

在 Amazon S OpenSearch ervice 中搜索文档有几种常用的方法，包括 URI 搜索和请求正文搜索。OpenSearch 服务提供了可改善搜索体验的其他功能，例如自定义软件包、SQL 支持和异步搜索。如需全面的 OpenSearch 搜索 API 参考，请参阅OpenSearch文档。

> **Note**
>
> 以下示例请求适用于 OpenSearch APIs。某些请求可能不适用于旧 Elasticsearch 版本。

主题

- URI 搜索
- 请求正文搜索
- 对搜索结果进行分页
- 控制面板查询语言
- 在 Amazon OpenSearch 服务中导入和管理包裹
- 使用 SQL 查询您的 Amazon Serv OpenSearch ice 数据
- 在 Amazon Service 中搜索 k 最近邻 (k-NN)。 OpenSearch
- 在 Amazon OpenSearch 服务中进行跨集群搜索
- 学习 Amazon S OpenSearch ervice 的排名
- 在 Amazon S OpenSearch ervice 异步搜索
- Amazon S OpenSearch ervice 中的时间点搜索
- 在 Amazon Service OpenSearch 中执行语义搜索
- Amazon S OpenSearch ervice 中的并发分段搜索
- Amazon Ser OpenSearch vice 生成自然语言查询

URI 搜索

统一资源标识符 (URI) 搜索是最简单的搜索形式。在 URI 搜索中，可以指定查询作为 HTTP 请求参数：

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/_search?q=house
```

示例响应可能与以下内容下类似：

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

默认情况下，此查询在所有索引的所有字段中搜索 house 一词。要缩小搜索范围，请在 URI 中指定索引 (`movies`) 和文档字段 (`title`)：

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?q=title:house
```

您可以在请求中包含其他参数，但支持的参数仅提供一小部分 OpenSearch 搜索选项。以下请求将返回 20 个（而不是默认的 10 个）结果并按年（而不是按 `_score`）对这些结果进行排序：

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?
q=title:house&size=20&sort=year:desc
```

请求正文搜索

要执行更复杂的搜索，请对查询使用 HTTP 请求正文和特定于 OpenSearch 域的语言 (DSL)。利用查询 DSL，您可以指定所有的 OpenSearch 搜索选项。

> **Note**
>
> 不能在文本字段值中包括 Unicode 特殊字符，否则该值将被解析为由特殊字符分隔的多个值。这种错误的解析可能会导致无意中筛选文档，并可能影响对其访问的控制。有关更多信息，请参阅 OpenSearch文档中关于文本字段中 Unicode 特殊字符的注释。

以下 `match` 查询类似于最终 URI 搜索示例：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "sort": {
    "year": {
```

```
      "order": "desc"
    }
  },
  "query": {
    "query_string": {
      "default_field": "title",
      "query": "house"
    }
  }
}
```

> **Note**
>
> 对于请求正文搜索，_search API 接受 HTTP GET 和 POST，但并非所有 HTTP 客户端都支持将请求正文添加到 GET 请求。POST 是更普遍的选择。

在许多情况下，您可能想搜索多个字段，而不是全部字段。使用 multi_match 查询：

```
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"]
    }
  }
}
```

提升字段

可以通过“提升”某些字段来增强搜索相关性。提升是一种倍增器，它使一个字段中的匹配项的权重高于其他字段中的匹配项的权重。在以下示例中，john 在 plot 字段中的匹配项的影响是 title 字段中匹配项的影响的_score两倍，并且是 actors 或 directors 字段中匹配项的影响的四倍。其结果是，John Wick 和 John Carter 等电影在搜索结果中接近榜首，而由 John Travolta 主演的电影则接近底部。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
```

```
  "query": {
    "multi_match": {
      "query": "john",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

搜索结果突出显示

该highlight选项告诉 OpenSearch 如果查询匹配一个或多个字段，则在hits数组内返回一个额外的对象：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    }
  }
}
```

如果查询与 plot 字段的内容匹配，则命中的内容可能与以下类似：

```
{
  "_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>."
    ]
  }
}
```

默认情况下，将匹配的字符串 OpenSearch 包装在标签中，为匹配项提供最多 100 个字符的上下文，并通过识别标点符号、空格、制表符和换行符将内容分成句子。所有这些设置均可自定义：

```
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": ".,!? "
  }
}
```

计数 API

如果您对文档内容不感兴趣，只是想知道匹配项的数量，则可使用 `_count` API 而非 `_search` API。以下请求使用 `query_string` 查询来标识浪漫喜剧：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_count
{
  "query": {
    "query_string": {
      "default_field": "genres",
      "query": "romance AND comedy"
    }
  }
}
```

示例响应可能与以下内容下类似：

```
{
  "count": 564,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  }
}
```

对搜索结果进行分页

如果您需要显示大量搜索结果，可使用几种不同的方法对结果进行分页。

时间点

时间点 (PIT) 功能是一种搜索类型，可让您对固定时间的数据集运行不同的查询。这是中首选的分页方法 OpenSearch，特别是对于深度分页。您可以将 PIT 与 OpenSearch 服务版本 2.5 及更高版本一起使用。有关 PIT 的更多信息，请参阅 ???。

from 和 size 参数

最简单的分页方法是使用 from 和 size 参数。以下请求返回从零开始编制索引的搜索结果列表中的 20-39 个结果：

```
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"]
    }
  }
}
```

有关搜索分页的更多信息，请参阅文档中的分页结果。 OpenSearch

控制面板查询语言

您可以使用仪表板查询语言 (DQL) 在仪表板中 OpenSearch 搜索数据和可视化效果。DQL 使用四种主要查询类型：术语、布尔、日期和范围和嵌套字段。

术语查询

术语查询要求您指定要搜索的术语。

要执行术语查询，请输入以下内容：

```
host:www.example.com
```

布尔查询

您可以使用布尔运算符 AND、or 和 not 组合多个查询。

要执行布尔查询，请粘贴以下内容：

```
host.keyword:www.example.com and response.keyword:200
```

日期和范围查询

您可以使用日期和范围查询来查找查询之前或之后的日期。

- > 表示搜索指定日期之后的日期。
- < 表示搜索指定日期之前的日期。

```
@timestamp > "2020-12-14T09:35:33"
```

嵌套字段查询

如果您具有包含嵌套字段的文档，则必须指定要检索文档的哪些部分。以下是一个包含嵌套字段的示例文档：

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

要使用 DQL 检索特定字段，请粘贴以下内容：

```
NBA players: {player-name: Lebron James}
```

要检索嵌套文档中的多个对象，请粘贴以下内容：

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo}
```

要在某个范围内搜索，请粘贴以下内容：

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo and < 30}
```

如果您的文档中有一个对象嵌套在另一个对象中，您仍然可以通过指定所有级别来检索数据。要进行此检索，请粘贴以下内容：

```
Top-Power-forwards.NBA players: {player-name:Lebron James}
```

在 Amazon OpenSearch 服务中导入和管理包裹

Amazon Ser OpenSearch vice 可让您上传自定义字典文件（例如，停用词和同义词），并将插件与域关联起来。这些插件可以是预先打包的、自定义的，也可以是第三方的，这使您可以灵活地扩展域的功能。所有这些类型文件的通用术语都是程序包。

- 字典文件通过指示忽略常见的高频字词或 OpenSearch 将类似的字词（例如 “软冻蛋奶”、” 冰淇淋” 和 “冰激凌”）视为等效，从而帮助优化搜索结果。它们还可以改善词干提取，如日语（kuromoji）分析插件所示。
- 预先打包的插件提供内置功能，例如用于个性化搜索结果的 Amazon Personalize 插件。这些插件使用`ZIP-PLUGIN`软件包类型。有关更多信息，请参阅 the section called “插件（按引擎版本）”。
- 自定义插件和第三方插件允许您添加量身定制的功能或与外部系统集成，从而为您的域名提供更大的灵活性。与预先打包的插件一样，您可以将自定义插件作为`ZIP-PLUGIN`包上传。对于第三方插件，您还必须将插件许可证和配置文件作为单独的软件包导入，然后将它们全部与域相关联。

  有关更多信息，请参阅以下主题：

  - the section called “自定义插件”
  - the section called “第三方插”

> **Note**
>
> 一个域最多可与 20 个插 此限制包括所有插件类型，包括可选插件、第三方插件和自定义插件。

主题

- 所需的权限
- 将程序包上传到 Amazon S3
- 导入和关联程序包
- 将包与配合使用 OpenSearch
- 更新程序包
- 使用新词典手动更新索引
- 取消程序包关联并移除程序包
- 在 Amazon OpenSearch 服务中管理自定义插件
- 在 Amazon OpenSearch 服务中安装第三方插件

所需的权限

没有管理员权限的用户需要某些 Amazon Identity and Access Management (IAM) 操作来管理软件包：

- `es:CreatePackage`— 创建软件包
- `es:DeletePackage`— 删除包
- `es:AssociatePackage`— 将软件包与域关联
- `es:DissociatePackage`— 取消程序包与域的关联

您还需要对自定义软件包所在的 Amazon S3 存储桶路径或对象的权限。

授予 IAM 中的所有权限，而不是域访问策略中的权限。有关更多信息，请参阅 the section called “身份和访问管理”。

将程序包上传到 Amazon S3

因为已安装了预打包的插件包，所以本节介绍如何上传自定义字典程序包。在您将自定义字典与域关联之前，您必须将其上传到 Amazon S3 存储桶。有关说明，请参阅 Amazon Simple Storage Service 用户指南中的上传对象。受支持的插件无需上传。

如果字典包含敏感信息，请在上传时指定使用 S3 托管密钥的服务器端加密。 OpenSearch Service 无法访问您使用 Amazon KMS 密钥保护的 S3 中的文件。

上传文件后，记下其 S3 路径。路径格式为 `s3://amzn-s3-demo-bucket/file-path/file-name`。

您可以使用以下同义词文件进行测试。将其保存为 `synonyms.txt`。

```
danish, croissant, pastry
ice cream, gelato, frozen custard
sneaker, tennis shoe, running shoe
basketball shoe, hightop
```

某些字典（如 Hunspell 字典）使用多个文件，并且在文件系统上需要自己的目录。目前，S OpenSearch ervice 仅支持单文件字典。

导入和关联程序包

控制台是将自定义字典导入 S OpenSearch ervice 的最简单方法。从 Amazon S3 导入字典时，S OpenSearch ervice 存储自己的程序包副本，并使用 AES-256 以及 OpenSearch 服务托管密钥自动加密该副本。

S OpenSearch ervice 中已预安装了可选插件，因此您无需上传插件，但需要将其与域关联起来。控制台中的程序包屏幕上列出了可用的插件。

导入软件包并将其与域关联

1. 在亚马逊 OpenSearch 服务控制台中，选择套餐。
2. 选择 Import package（导入软件包）。
3. 为程序包提供描述性名称。
4. 提供文件的 S3 路径，然后选择 Import (导入)。
5. 返回到 Packages (程序包) 屏幕。
6. 当程序包状态为 Available (可用) 时，选择它。
7. 选择关联到域。
8. 选择一个域，然后选择 “下一步”。查看软件包并选择 “关联”。
9. 在导航窗格中，选择您的域，然后选择 程序包 选项卡。
10. 如果程序包是自定义字典，当程序包可用时记下 ID。在对的请求中`analyzers/id`用作文件路径 OpenSearch。

将包与配合使用 OpenSearch

本节介绍如何使用这两种类型的软件包：自定义字典和预打包插件。

使用自定义字典

将文件与域关联后，可以在创建分词器和令牌过滤器user_dictionary时在synonyms_pathstopwords_path、和等参数中使用该文件。确切的参数因对象而异。多个对象支持 synonyms_path 和 stopwords_path，但 user_dictionary 专用于 kuromoji 插件。

对于 IK（中文）分析插件，您可以将自定义字典文件作为自定义软件包上载并将其关联到某个域，插件会自动选择该字典，无需使用 user_dictionary 参数。如果您的文件是同义词文件，请使用 synonyms_path 参数。

以下示例将同义词文件添加到新索引中：

```
PUT my-index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "my_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["my_filter"]
          }
        },
        "filter": {
          "my_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F111111111",
            "updateable": true
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "standard",
        "search_analyzer": "my_analyzer"
      }
    }
  }
```

```
}
```

此请求为使用标准标记器和同义词标记筛选条件的索引创建自定义分析器。

- 标记器根据一组规则将字符流分解为标记（通常是字词）。最简单的例子是空白标记器，它在每次遇到空白字符时都会将前面的字符分解为标记。一个更复杂的例子是标准标记器，它使用一组基于语法的规则来跨多种语言工作。
- 标记筛选条件可添加、修改或删除标记。例如，同义词标记筛选条件在同义词列表中找到字词时添加标记。停止标记筛选条件在停用词列表中找到字词时删除标记。

此请求还将文本字段 (`description`) 添加到映射中，并告知 OpenSearch 将新分析器用作其搜索分析器。您可以看到它仍然使用标准分析器作为其索引分析器。

最后，请注意在令牌过滤器中的行 `"updateable": true`。如果您以后想要更新搜索分析器自动执行，则此字段仅适用于搜索分析器，而不适用于索引分析器。

为了进行测试，请将一些文档添加到索引中：

```
POST _bulk
{ "index": { "_index": "my-index", "_id": "1" } }
{ "description": "ice cream" }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "description": "croissant" }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "description": "tennis shoe" }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "description": "hightop" }
```

然后使用同义词搜索它们：

```
GET my-index/_search
{
  "query": {
    "match": {
      "description": "gelato"
    }
  }
}
```

在这种情况下， OpenSearch 返回以下响应：

```
{
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 0.99463606,
    "hits": [{
      "_index": "my-index",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.99463606,
      "_source": {
        "description": "ice cream"
      }
    }]
  }
}
```

> **Tip**
>
> 字典文件使用与其大小成正比的 Java 堆空间。例如，2 GiB 字典文件可能会占用节点上 2 GiB 的堆空间。如果使用大型文件，请确保节点具有足够的堆空间来容纳它们。监控 `JVMMemoryPressure` 指标，并根据需要扩展集群。

使用预先打包的插件

OpenSearch Service 可让您关联预安装的可选 OpenSearch 插件，以与您的域名一起使用。预打包的插件包与某个特定 OpenSearch 版本兼容，并且只能与具有该版本的域相关联。适用于您域的可用程序包列表包括与您的域版本兼容的所有受支持插件。将插件与域关联后，该域上的安装流程就开始了。然后，您可以在向 S OpenSearch ervice 发出请求时引用和使用该插

关联和取消关联插件需要蓝绿部署。有关更多信息，请参阅 the section called “通常会导致 blue/green 部署的更改”。

可选插件包括语言分析器和自定义搜索结果。例如，Amazon Personalize 搜索排名插件使用机器学习，为您的客户进行搜索结果的个性化设置。有关此插件的更多信息，请参阅个性化搜索结果。OpenSearch有关受支持插件的完整列表，请参阅 the section called “插件（按引擎版本）”。

Sudachi 插件

对于 Sudachi 插件，当您重新关联字典文件时，它不会立即反映在域上。当作为配置更改或其他更新的一部分在域上运行下一个蓝绿部署时，字典会刷新。您也可以使用更新后的数据创建新的程序包，使用此新程序包创建新索引，利用现有索引重新创建新索引，然后删除旧索引。如果您更喜欢使用重新编制索引的方法，请使用索引别名，这样您的流量就不会受到干扰。

此外，Sudachi 插件仅支持二进制 Sudachi 字典，您可以使用 API 操作上传这些字典。CreatePackage有关预先构建的系统字典和编译用户字典流程的信息，请参阅 Sudachi 文档。

以下示例演示如何使用系统和用户字典以及 Sudachi 标记器。您必须将这些字典作为带有 tepe 的自定义软件包上传，`TXT-DICTIONARY`并在其他设置 IDs 中提供其软件包。

```
PUT sudachi_sample
{
  "settings": {
    "index": {
      "analysis": {
        "tokenizer": {
          "sudachi_tokenizer": {
            "type": "sudachi_tokenizer",
            "additional_settings": "{\"systemDict\": \"<system-dictionary-package-id>\",\"userDict\": [\"<user-dictionary-package-id>\"]}"
        }
        },
        "analyzer": {
          "sudachi_analyzer": {
            "filter": ["my_searchfilter"],
            "tokenizer": "sudachi_tokenizer",
            "type": "custom"
          }
        },
        "filter":{
          "my_searchfilter": {
            "type": "sudachi_split",
            "mode": "search"
          }
        }
      }
    }
  }
}
```

更新程序包

因为已为您更新了预打包的插件包，所以本节仅介绍如何更新自定义字典程序包。将字典的新版本上传到 Amazon S3 不会自动更新 Amazon S OpenSearch ervice 上的软件包。 OpenSearch Service 存储自己的文件副本，因此，如果您将新版本上传到 S3，则必须手动更新该版本。

您的每个关联域也存储其自己的文件副本。为了保持搜索行为的可预测性，域将继续使用其当前的软件包版本，直到您明确更新它们。要更新自定义软件包，请在中修改文件 Amazon S3 Control，在 S OpenSearch ervice 中更新包，然后应用此更新。

控制台

1. 在 OpenSearch 服务控制台中，选择软件包。
2. 选择一个软件包，然后选择 Update（更新）。
3. 提供文件的新的 S3 路径，然后选择更新软件包。
4. 返回到 Packages (程序包) 屏幕。
5. 当程序包状态为 Available (可用) 时，将其选中。然后选择一个或多个关联域，应用更新，并确认。等待关联状态更改为处于活动状态。
6. 下面的步骤因您配置索引的方式而异：
   - 如果您的域正在运行 OpenSearch Elasticsearch 7.8 或更高版本，并且仅使用带有可更新字段设置为 true 的搜索分析器，则无需执行任何进一步操作。 OpenSearch Service 使用 _plugins/_refesh_search_ analyzers API 自动更新您的索引。
   - 如果域运行的是 Elasticsearch 7.7 或更低版本，使用索引分析器，或者不使用 `updateable` 字段，请参阅the section called “使用新词典手动更新索引”。

虽然控制台是最简单的方法，但您也可以使用 Amazon CLI SDKs、或配置 API 更新S OpenSearch ervice 包。有关更多信息，请参阅Amazon CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

Amazon SDK

您可以使用自动执行更新过程，而不是在控制台中 SDKs 手动更新软件包。以下示例 Python 脚本将新的软件包文件上传到 Amazon S3，更新 S OpenSearch ervice 中的包，并将新包应用到指定的域。确认更新成功后，它会进行示例调用，以 OpenSearch证明已应用新的同义词。

必须提供 `host`、`region`、`file_name`、`bucket_name`、`s3_key`、`package_id`、`domain_name` 和 `query` 的值。

```
from requests_aws4auth import AWS4Auth
import boto3
import requests
import time
import json
import sys

host = ''  # The OpenSearch domain endpoint with https:// and a trailing slash. For
 example, https://my-test-domain.us-east-1.es.amazonaws.com/
region = ''  # For example, us-east-1
file_name = ''  # The path to the file to upload
bucket_name = ''  # The name of the S3 bucket to upload to
s3_key = ''  # The name of the S3 key (file name) to upload to
package_id = ''  # The unique identifier of the OpenSearch package to update
domain_name = ''  # The domain to associate the package with
query = ''  # A test query to confirm the package has been successfully updated

service = 'es'
credentials = boto3.Session().get_credentials()
client = boto3.client('opensearch')
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def upload_to_s3(file_name, bucket_name, s3_key):
    """Uploads file to S3"""
    s3 = boto3.client('s3')
    try:
        s3.upload_file(file_name, bucket_name, s3_key)
        print('Upload successful')
        return True
    except FileNotFoundError:
        sys.exit('File not found. Make sure you specified the correct file path.')


def update_package(package_id, bucket_name, s3_key):
    """Updates the package in OpenSearch Service"""
    print(package_id, bucket_name, s3_key)
    response = client.update_package(
        PackageID=package_id,
        PackageSource={
            'S3BucketName': bucket_name,
            'S3Key': s3_key
```

```
        }
    )
    print(response)


def associate_package(package_id, domain_name):
    """Associates the package to the domain"""
    response = client.associate_package(
        PackageID=package_id, DomainName=domain_name)
    print(response)
    print('Associating...')


def wait_for_update(domain_name, package_id):
    """Waits for the package to be updated"""
    response = client.list_packages_for_domain(DomainName=domain_name)
    package_details = response['DomainPackageDetailsList']
    for package in package_details:
        if package['PackageID'] == package_id:
            status = package['DomainPackageStatus']
            if status == 'ACTIVE':
                print('Association successful.')
                return
            elif status == 'ASSOCIATION_FAILED':
                sys.exit('Association failed. Please try again.')
            else:
                time.sleep(10)  # Wait 10 seconds before rechecking the status
                wait_for_update(domain_name, package_id)


def sample_search(query):
    """Makes a sample search call to OpenSearch"""
    path = '_search'
    params = {'q': query}
    url = host + path
    response = requests.get(url, params=params, auth=awsauth)
    print('Searching for ' + '"' + query + '"')
    print(response.text)
```

> **Note**
>
> 如果您在使用运行脚本时收到 “找不到软件包” 错误 Amazon CLI，则可能意味着 Boto3 正在使用 ~/.aws/config 中指定的任何区域，这不是您的 S3 存储桶所在的区域。或者运行 `aws configure` 并指定正确的区域，或者将区域显式添加到客户端：
>
> ```
> client = boto3.client('opensearch', region_name='us-east-1')
> ```

使用新词典手动更新索引

手动索引更新仅适用于自定义词典，不适用于预打包的插件。要使用更新的字典，如果满足以下任何条件，则必须手动更新索引：

- 您的域名运行弹性搜索 7.7 或更早版本。
- 您可以使用自定义软件包作为索引分析器。
- 您可以使用自定义软件包作为搜索分析器，但不包括可更新字段中返回的子位置类型。

要使用新的程序包文件更新分析器，您有两种选择：

- 关闭并打开要更新的任何索引：

```
POST my-index/_close
POST my-index/_open
```

- 重建索引。首先，创建使用更新的同义词文件（或全新文件）的索引。请注意，仅支持 UTF-8。

```
PUT my-new-index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "synonym_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["synonym_filter"]
          }
        },
```

```
        "filter": {
          "synonym_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F222222222"
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "synonym_analyzer"
      }
    }
  }
}
```

然后，将旧索引重新编制为该新索引：

```
POST _reindex
{
  "source": {
    "index": "my-index"
  },
  "dest": {
    "index": "my-new-index"
  }
}
```

如果您经常更新同义词文件，请使用索引别名来维护最新索引的一致路径：

```
POST _aliases
{
  "actions": [
    {
      "remove": {
        "index": "my-index",
        "alias": "latest-index"
      }
    },
```

```
    {
      "add": {
        "index": "my-new-index",
        "alias": "latest-index"
      }
    }
  ]
}
```

如果您不需要旧索引，请将其删除。

```
DELETE my-index
```

取消程序包关联并移除程序包

将程序包（无论是自定义字典还是预打包的插件）与域取消关联意味着，在创建新索引时，您不能再使用该程序包。程序包取消关联后，使用该程序包的现有索引将不再能使用该程序包。必须首先从所有索引中移除该程序包，然后再将其取消关联，否则取消关联操作将会失败。

控制台是将程序包与域取消关联并将其从 OpenSearch Service 中删除的最简单方法。从 S OpenSearch ervice 中移除程序包不会将程序包从 Amazon S3 上的原始位置移除。

取消程序包与域的关联

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在导航窗格中，选择域。
3. 选择域名，然后导航至 “软件包” 选项卡。
4. 选择程序包、Actions (操作)，然后选择 Dissociate (取消关联)。确认您的选择。
5. 等待程序包从列表中消失。您可能需要刷新浏览器。
6. 如果要将程序包与其他域一起使用，请在此处停止。要继续删除程序包（如果是自定义字典），请在导航窗格中选择程序包。
7. 选择程序包，然后选择 Delete (删除)。

或者，也可以使用 Amazon CLI SDKs、或配置 API 来解除关联和移除软件包。有关更多信息，请参阅Amazon CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

在 Amazon OpenSearch 服务中管理自定义插件

使用 OpenSearch 服务自定义插件，您可以扩展语言分析、自定义筛选、排名等领域的 OpenSearch 功能，从而可以打造个性化的搜索体验。的自定义插件 OpenSearch 可以通过扩展`org.opensearch.plugins.Plugin`类然后将其打包到`.zip`文件中来开发。

Amazon OpenSearch 服务目前支持以下插件扩展：

- AnalysisPlugin— 通过添加自定义分析器、字符分词器或用于文本处理的过滤器等来扩展分析功能。
- SearchPlugin— 通过自定义查询类型、相似度算法、建议选项和聚合增强搜索功能。
- MapperPlugin— 允许您在中创建自定义字段类型及其映射配置 OpenSearch，从而可以定义应如何存储和索引不同类型的数据。
- ScriptPlugin— 允许您在自定义脚本中添加自定义脚本功能，例如在搜索或索引期间进行评分、排序和字段值转换等操作。 OpenSearch

您可以使用 OpenSearch 服务控制台或自定义包的现有 API 命令上传插件并将插件与 Amazon S OpenSearch ervice 集群关联。您还可以使用该DescribePackages命令来描述您账户中的所有软件包，并查看 OpenSearch 版本和错误详细信息等详细信息。 OpenSearch 服务会验证插件包的版本兼容性、安全漏洞和允许的插件操作。有关自定义程序包的更多信息，请参阅the section called “程序包”。

OpenSearch 版本和 Amazon Web Services 区域 支持

运行以下 OpenSearch Amazon Web Services 区域版本 2.15 的 OpenSearch 服务域支持自定义插件：

- 美国东部（俄亥俄州）(us-east-2)
- 美国东部（弗吉尼亚北部）(us-east-1)
- 美国西部（俄勒冈州）(us-west-2)
- 亚太地区（孟买）(ap-south-1)
- 亚太地区（首尔）(ap-northeast-2)
- 亚太地区（新加坡）(ap-southeast-1)
- 亚太地区（悉尼）(ap-southeast-2)
- 亚太地区（东京）(ap-northeast-1)
- 加拿大（中部）(ca-central-1)

- 欧洲地区（法兰克福）(eu-central-1)
- 欧洲地区（爱尔兰）(eu-west-1)
- 欧洲（伦敦）(eu-west-2)
- 欧洲地区（巴黎）（eu-west-3）
- 南美洲（圣保罗）（sa-east-1）

> **Note**
>
> 自定义插件包含用户开发的代码。由用户开发的代码导致的任何问题（包括违反 SLA）都没有资格获得 SLA 积分。有关更多信息，请参阅 Amazon S OpenSearch ervice-服务等级协议。

主题

- 插件配额
- 先决条件
- 故障排除
- 使用控制台安装创建的
- 使用管理自定义插件 Amazon CLI
- 亚马逊 OpenSearch 服务定制包裹 Amazon KMS 集成

插件配额

- 您最多可以为每个区域的每个账户创建 25 个自定义插
- 一个插件的最大未压缩大小为 1 GB。
- 可以与单个域关联的最大插件数量为 20。此配额适用于所有插件类型的组合：可选、第三方和自定义。
- 运行 2.15 或更高 OpenSearch 版本的域名支持自定义插件。
- 您的插件`descriptor.properties`文件必须支持类似于 2.15.0 的引擎版本或任何 2.x.x 版本，其中补丁版本设置为零。
- 当您的网域使用自定义插件时，以下功能不可用：
  - 跨集群搜索
  - 跨集群复制

- 远程重新索引
- 自动调整
- the section called “带待机功能的多可用区”
- 集中式 OpenSearch 用户界面

先决条件

在您开始之前，确保您满足以下要求：

- `descriptor.properties`文件中插件支持的引擎版本应与`2.15.0`或类似`2.x.0`。也就是说，补丁版本必须为零。
- 您的域名必须启用以下功能：
  - Node-to-node 加密
  - 静态加密
  - `EnforceHTTPS`设置为 “真”

    另请参阅opensearch-https-required《Amazon Config 开发人员指南》。
  - 客户必须支持 Policy-M in- TLS-1-2-PFS-2023-10。您可以使用以下命令指定此支持。将*`placeholder value`*为您自己的信息：

    ```
    aws opensearch update-domain-config \
        --domain-name domain-name \
        --domain-endpoint-options '{"TLSSecurityPolicy":"Policy-Min-TLS-1-2-
    PFS-2023-10" }'
    ```

    有关更多信息，请参阅DomainEndpointOptions《亚马逊 OpenSearch 服务 API 参考》。

故障排除

如果系统返回错误`PluginValidationFailureReason : The provided plugin could not be loaded`，请参见the section called “由于版本兼容性，自定义插件安装失败”以获取故障排除信息。

使用控制台安装创建的

要将第三方插件与域相关联，请先将插件许可证和配置作为软件包导入。

安装自定义

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择 Packes (包)。
3. 选择 Import package（导入软件包）。
4. 对于名称，输入插件的唯一、易于识别的名称。
5. （可选）在描述中，提供有关软件包或其用途的所有有用详细信息。
6. 对于 Package 类型，选择 Plugin。
7. 对于 P ackage 来源，请输入路径或浏览到 Amazon S3 中的插件 ZIP 文件。
8. 对于OpenSearch 引擎版本，请选择插 OpenSearch 件支持的版本。
9. 对于 Package 加密，请选择是否为包自定义加密密钥。默认情况下，S OpenSearch ervice 使用加密插件包。 Amazon 拥有的密钥您可以改使用客户自主管理型密钥。
10. 选择 Import（导入）。

导入插件包后，将其与域关联。有关说明，请参阅the section called “导入软件包并将其与域关联”。

使用管理自定义插件 Amazon CLI

您可以使用 Amazon CLI 来管理许多自定义插件任务。

任务

- 使用创建 Amazon CLI
- 使用创建 Amazon CLI
- 创建或更新带有 Amazon KMS 密钥安全的自定义插件
- 将带有自定义插件的 OpenSearch 服务域升级到更高版本的 OpenSearch 使用 Amazon CLI
- 卸载和查看自定义插件的解除关联状态

使用创建 Amazon CLI

开始前的准备工作

在您将自定义插件与域关联之前，您必须将其上传到 Amazon Simple Service 存储桶。该存储桶必须位于要使用该插件的同一 Amazon Web Services 区域 位置。有关向 S3 存储桶添加对象的信息，请参阅 Amazon 简单存储服务用户指南中的上传对象。

如果插件包含敏感信息，请在上传时指定使用 S3 托管密钥的服务器端加密。上传文件后，记下其 S3 路径。路径格式为 `s3://amzn-s3-demo-bucket/file-path/file-name`。

> **Note**
>
> 在创建自定义插件时，您可以选择通过指定 Amazon Key Management Service (Amazon KMS) 密钥来保护自定义插件。有关信息，请参阅the section called “创建或更新带有 Amazon KMS 密钥安全的自定义插件”。

使用创建的 Amazon CLI

1. 运行以下 create-package 命令为您的自定义插件创建一个新软件包，确保满足以下要求：
   - 存储桶和密钥位置必须指向您运行命令的账户中 S3 存储桶中的插件`.zip`文件。
   - S3 存储桶必须位于创建软件包的同一区域。
   - `ZIP-PLUGIN`软件包仅支持`.zip`文件。
   - `.zip`文件内容必须符合插件预期的目录结构。
   - 的值`--engine-version`必须采用以下格式`OpenSearch_{MAJOR}.{MINOR}`。例如：**`OpenSearch_2.17`**。

   将*`placeholder values`*为您自己的信息：

```
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. （可选）使用 desc ribe- packages 命令查看`create-package`操作状态，包括发现的任何验证和安全漏洞。将*`placeholder values`*为您自己的信息：

```
aws opensearch describe-packages \
    --region region  \
    --filters '[{"Name": "PackageType","Value": ["ZIP-PLUGIN"]}, {"Name":
 "PackageName","Value": ["package-name"]}]'
```

命令返回类似于下文的信息：

```
{
    "PackageDetailsList": [{
        "PackageID": "pkg-identifier",
        "PackageName": "package-name",
        "PackageType": "ZIP-PLUGIN",
        "PackageStatus": "VALIDATION_FAILED",
        "CreatedAt": "2024-11-11T13:07:18.297000-08:00",
        "LastUpdatedAt": "2024-11-11T13:10:13.843000-08:00",
        "ErrorDetails": {
            "ErrorType": "",
            "ErrorMessage": "PluginValidationFailureReason : Dependency Scan
 reported 3 vulnerabilities for the plugin: CVE-2022-23307, CVE-2019-17571,
 CVE-2022-23305"
        },
        "EngineVersion": "OpenSearch_2.15",
        "AllowListedUserList": [],
        "PackageOwner": "OWNER-XXXX"
    }]
}
```

> **Note**
>
> 在`create-package`操作过程中，Amazon Serv OpenSearch ice 会检查该`ZIP-PLUGIN`值的版本兼容性、支持的插件扩展和安全漏洞。使用亚马逊 Inspector 服务对安全漏洞进行扫描。这些检查的结果显示在 API 响应的`ErrorDetails`字段中。

3. 使用 ass ociate-package 命令使用上一步中创建的软件包的软件包 ID 将插件与您选择的 OpenSearch 服务域关联起来。

> **Tip**
>
> 如果您有多个插件，则可以改为使用 ass ociate-packag es 命令在单个操作中将多个软件包关联到一个域。

将*`placeholder values`*为您自己的信息：

```
aws opensearch associate-package \
    --domain-name domain-name \
    --region region \
    --package-id package-id
```

> **Note**
>
> 使用蓝/绿部署过程安装和卸载插件。

4. （可选）使用list-packages-for-domain命令查看关联的状态。随着工作流程从到的进展，关联状态会发生变化ASSOCIATING。ACTIVE插件安装完成且插件准备就绪后，关联状态将更改为ACTIVE。

   将 *placeholder values* 替换为您自己的信息。

```
aws opensearch list-packages-for-domain \
    --region region \
    --domain-name domain-name
```

使用创建 Amazon CLI

使用 update-packag e 命令对插件进行更改。

> **Note**
>
> 在更新插件时，您可以选择通过指定 Amazon Key Management Service (Amazon KMS) 密钥来保护自定义插件。有关信息，请参阅the section called “创建或更新带有 Amazon KMS 密钥安全的自定义插件”。

使用创建的 Amazon CLI

- 运行以下命令。将 *placeholder values* 替换为您自己的信息。

```
aws opensearch update-package \
    --region region \
    --package-id package-id \
    --package-source S3BucketName=amzn-s3-demo-bucket,S3Key=s3-key \
```

```
    --package-description description
```

更新软件包后，您可以使用 associ ate-packages 或 assoc iate-p ackages 命令将软件包更新应用于域。

> **Note**
>
> 您可以使用审计、创建、更新、关联和取消关联插件上的操作。 Amazon CloudTrail有关更多信息，请参阅 使用监控亚马逊 OpenSearch 服务 API 调用 Amazon CloudTrail。

创建或更新带有 Amazon KMS 密钥安全的自定义插件

在创建或更新插件时，您可以通过指定 Amazon KMS 密钥来保护自定义插件。要实现此目的，请设置`PackageEncryptionOptions``true`并指定密钥的 Amazon 资源名称（ARN），如以下示例所示：

示例：创建具有 Amazon KMS 密钥安全性的自定义插件

```
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"
  }
```

示例：使用 Amazon KMS 密钥安全更新自定义插件

```
aws opensearch update-package \
    --region us-east-2  --package-name my-custom-package \
    --package-type ZIP-PLUGIN \
    --package-source S3BucketName=amzn-s3-demo-bucket,S3Key=my-s3-key
    --engine-version OpenSearch_2.15
"PackageConfigOptions": {
```

```
    ...
  }
  "PackageEncryptionOptions": {
    "Enabled": true,
    "KmsKeyId":"arn:aws:kms:us-east-2:111222333444:key/2ba228d5-1d09-456c-ash9-
daf42EXAMPLE"
  }
```

> **Important**
>
> 如果您指定的 Amazon KMS 密钥被禁用或删除，它可能会使关联的集群无法运行。

有关与自定义包 Amazon KMS 集成的更多信息，请参阅the section called “亚马逊 OpenSearch 服务定制包裹 Amazon KMS 集成”。

将带有自定义插件的 OpenSearch 服务域升级到更高版本的 OpenSearch 使用 Amazon CLI

当您需要将使用自定义插件的 OpenSearch 服务域升级到更高版本时 OpenSearch，请完成以下过程。

要将带有自定义插件的 OpenSearch 服务域升级到更高版本的 OpenSearch使用 Amazon CLI

1. 使用 create-package 命令为您的插件创建一个新软件包，指定新 OpenSearch 版本。

   确保所有引擎版本的插件的软件包名称相同。更改软件包名称会导致域升级过程在蓝/绿部署期间失败。
2. 按照中的步骤将您的域名升级到更高版本the section called “升级域”。

   在此过程中，Amazon Serv OpenSearch ice 会解除先前版本的插件包的关联，并使用蓝/绿部署安装新版本。

卸载和查看自定义插件的解除关联状态

要从任何域中卸载插件，你可以使用 dissociat e-package 命令。运行此命令还会删除所有相关的配置或许可证包。然后，您可以使用list-packages-for-domain命令查看解除关联的状态。

> **Tip**
>
> 您也可以使用 dissociate-pack ages 命令在单个操作中从一个域中卸载多个插件。

卸载并查看自定义插件的解除关联状态

1. 在每个索引中禁用该插件。这必须在取消插件程序包的关联之前完成。

   如果您尝试在从每个索引中禁用插件之前将其卸载，则蓝/绿部署过程仍处于该`Processing`状态。

2. 运行以下命令来卸该插 将 *`placeholder values`* 替换为您自己的信息。

```
aws opensearch dissociate-package \
    --region region \
    --package-id plugin-package-id \
    --domain-name domain name
```

3. （可选）运行list-packages-for-domain命令以查看解除关联的状态。

亚马逊 OpenSearch 服务定制包裹 Amazon KMS 集成

默认情况下，Amazon S OpenSearch ervice 定制包裹提供加密，以保护您的`ZIP-PLUGIN`包裹处于静态状态 Amazon 托管式密钥。

- Amazon 拥有的密钥— Amazon S OpenSearch ervice 定制包裹默认使用这些密钥来自动加密您的`ZIP-PLUGIN`包裹。您无法查看、管理或使用， Amazon 拥有的密钥 或者审计其使用情况。但是，无需执行任何操作或更改任何计划即可保护用于加密数据的密钥。有关更多信息，请参阅 Amazon Key Management Service 开发人员指南中的 Amazon 拥有的密钥。
- 客户托管密钥 — 您可以在创建`ZIP-PLUGIN`自定义软件包时选择客户托管密钥， Amazon 拥有的密钥 从而在现有密钥的基础上添加第二层加密。

  Amazon S OpenSearch ervice 自定义软件包支持使用您创建、拥有和管理的对称客户托管密钥，在现 Amazon 有自带加密的基础上添加第二层加密。由于您可以完全控制这一层加密，因此可以执行以下任务：

  - 建立和维护密钥政策
  - 制定和维护 Amazon Identity and Access Management (IAM) 策略和拨款
  - 启用和禁用密钥政策
  - 轮换密钥加密材料
  - 添加标签
  - 创建密钥别名
  - 计划密钥删除

有关更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的客户托管密钥。

> **Note**
>
> Amazon S OpenSearch ervice 自定义软件包会自动使用启用静态加密 Amazon 拥有的密钥 ，且不收取任何费用。但是，当使用客户托管密钥时，会 Amazon KMS 收取费用。有关定价的更多信息，请参阅 Amazon Key Management Service 定价。

Amazon S OpenSearch ervice 定制包裹服务如何使用补助金 Amazon KMS

OpenSearch 服务自定义程序包需要授权，才能使用您的客户托管密钥。

当您创建使用客户托管密钥加密的`ZIP-PLUGIN`程序包时，Amazon S OpenSearch ervice 自定义软件包服务将通过向发送CreateGrant请求来代表您创建授权 Amazon KMS。授 Amazon KMS 予 OpenSearch 服务访问您账户中 Amazon KMS 密钥的权限。S OpenSearch ervice 自定义软件包创建的授权有一个约束，即只有当请求包含带有您的自定义软件包 ID 的加密上下文时，才允许进行操作。

Amazon S OpenSearch ervice 定制软件包需要授权，才能将客户托管的密钥用于以下内部操作：

操作	描述
`DescribeKey`	将`DescribeKey` 请求发送到， Amazon KMS 以验证在创建插件包时输入的对称客户自主管理型密钥 ID 是否有效。
`GenerateDataKeyWithoutPlaintext`	将`GenerateDataKeyWithoutPlaintext` 请求发送到 Amazon KMS ，以生成由客户托管密钥加密的数据密钥。
`GenerateDataKey`	向发送`GenerateDataKey` 请求 Amazon KMS 以生成数据密钥，以便在内部复制软件包时对其进行加密。
`Decrypt`	将`Decrypt`请求发送 Amazon KMS 到，以解密加密的数据密钥，以使它们能够用于解密数据。

您可以随时撤消针对授权的访问权限，也可以删除服务访问客户托管密钥的权限。如果您这样做，S OpenSearch ervice 将无法访问由客户托管密钥加密的任何数据，这会影响依赖于该数据的所有操作。例如，如果您尝试关联一个 S OpenSearch ervice 无法访问的插件包，则该操作将返回`AccessDeniedException`错误。

创建客户托管密钥

您可以使用 Amazon Web Services Management Console 或创建对称的客户托管密钥。 Amazon KMS APIs

创建对称的客户托管密钥

- 按照Amazon Key Management Service 开发人员指南中创建 KMS 密钥中的步骤进行操作。

密钥策略

密钥策略控制对客户自主管理型密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管式密钥时，可以指定密钥策略。有关更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的在 Amazon KMS中使用密钥策略。

要将您的客户托管密钥与插件资源一起使用，您必须在密钥策略中允许以下 API 操作：

- `kms:CreateGrant` – 向客户托管密钥添加授权。授予对指定 Amazon KMS 密钥的控制权限，从而允许访问 S OpenSearch ervice 自定义软件包所需的授权操作。有关授权的更多信息，请参阅Amazon KMS 开发人员指南。

  这允许 S OpenSearch ervice 能够执行以下操作：

  - 调`GenerateDataKeyWithoutPlainText`用生成加密的数据密钥并将其存储以进行进一步验证。
  - 调`GenerateDataKey`用在内部复制插件包。
  - 调用`Decrypt`以在内部访问插件包。
  - 设置停用主体，以允许服务 `RetireGrant`。
- `kms:DescribeKey`— 提供客户托管式密钥详细信息以允许 Serv OpenSearch ice 验证密钥。
- `kms:GenerateDataKey`,`kms:GenerateDataKeyWithoutPlaintext`, `kms:Decrypt` — 授予 OpenSearch 服务自定义包访问权限，以便在授权中使用这些操作。

以下是您可以为 S OpenSearch ervice 创建程序包添加的策略声明示例：

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

```
    "Sid" : "Allow access to principals authorized to use OpenSearch Service custom
 packages",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [
      "kms:CreateGrant",
      "kms:GenerateDataKey",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:Decrypt"
    ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "custom-packages.region.amazonaws.com"
      },
      "StringEquals" : {
        "kms:EncryptionContext:packageId": "Id of the package"
      }
    }
  },
  {
    "Sid" : "Allow access to principals authorized to use Amazon OpenSearch Service
 custom packages",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [
      "kms:DescribeKey"
    ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "custom-packages.region.amazonaws.com"
      }
    }
  }
]
```

有关在策略中指定权限的更多信息，请参阅《Amazon Key Management Service 开发人员指南》Amazon KMS中的密钥策略。

有关密钥访问疑难解答的更多信息，请参阅《Amazon Key Management Service 开发人员指南》中的Amazon KMS 权限疑难解答。

指定 Amazon Service 创建程序包的 OpenSearch 客户自主管理型密钥

您可以指定客户自主管理型密钥用于软件`ZIP-PLUGIN`包的第二层加密。

创建插件包时，您可以通过输入密钥 ID 来指定数据密钥，S OpenSearch ervice 自定义软件包使用 Amazon KMS 密钥 ID 来加密插件包。

Amazon KMS 密钥 ID — Amazon KMS 客户托管密钥的密钥标识符。输入密钥 ID、密钥 ARN、别名名称或别名 ARN。

Amazon OpenSearch 服务自定义包裹加密上下文

加密上下文是一组可选的键值对，包含有关数据的其他上下文信息。

Amazon KMS 将加密上下文用作其他已经过验证的数据 (AD) 以支持经过身份验证的加密。在请求中包含加密上下文以加密数据时，将加密上下文 Amazon KMS 绑定到加密的数据。要解密数据，您必须在请求中包含相同的加密上下文。

Amazon OpenSearch 服务自定义包裹加密上下文

Amazon S OpenSearch ervice 自定义软件包在所有 Amazon KMS 加密操作中使用相同的加密上下文，其中密钥是`packageId`，值是插件包`package-id`的值。

使用加密上下文进行监控

使用对称的客户托管密钥来加密插件包时，您可以使用审计记录和日志中的加密上下文来识别客户托管密钥的使用情况。加密上下文还会显示在 Amazon CloudTrail 或 Amazon Logs 生成的 CloudWatch 日志中。

使用加密上下文控制对客户托管密钥的访问

您可以使用密钥政策和 IAM 策略中的加密上下文作为条件来控制对您的对称客户托管密钥的访问。您也可以在授予中使用加密上下文约束。

OpenSearch Service 自定义软件包在授权中使用加密上下文限制，以控制对您账户或区域中客户托管密钥的访问。授权约束要求授权允许的操作使用指定的加密上下文。

以下是密钥政策声明示例，用于授予对特定加密上下文的客户托管密钥的访问权限。此策略语句中的条件要求授权具有指定加密上下文的加密上下文约束。

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

监控 OpenSearch 创建程序包服务的加密密钥

当您在 OpenSearch 服务自定义包服务资源中使用 Amazon KMS 客户托管密钥时，您可以使用 CloudTrail 或 CloudWatch 日志来跟踪 OpenSearch 自定义包发送到的请求 Amazon KMS。

了解更多

以下资源提供有关静态数据加密的更多信息。

- 有关 Amazon KMS 基本概念的更多信息，请参阅Amazon KMS keys《Amazon Key Management Service 开发人员指南》。
- 有关安全最佳实践的更多信息 Amazon KMS，请参阅《Amazon 规范性指南》Amazon Key Management Service 最佳实践指南。

在 Amazon OpenSearch 服务中安装第三方插件

Amazon OpenSearch 服务支持选定合作伙伴提供的第三方插件。这些插件可以通过自定义分析器、分词器或加密功能等其他功能来增强您的 OpenSearch 设置。按照第三方开发人员提供的特定安装和配置说明进行操作，以确保与您的 OpenSearch 服务域正确集成。

> **Note**
>
> 您必须直接从第三方开发者那里获得并维护有效的许可证。某些提供商可能无法完全启用其插件 Amazon Web Services 区域，因此请向插件提供商查询可用性。

以下第三方插件可用于 S OpenSearch ervice：

- Portal26 加密插件（Titanium-Lockbox）— 使用 NIST FIPS 140-2 认证的加密技术在数据编制索引时对其进行加密。它包括自带密钥 (BYOK) 支持，允许您管理加密密钥以增强安全性。该插件由 Portal26 提供，要求 OpenSearch 版本 2.15 及更高版本。
- 名称匹配 (RNI)-匹配超过 24 种语言的姓名、组织、地址和日期，从而提高安全性和合规性。该插件由 Babel St et 提供，要求 OpenSearch 版本 2.15 及更高版本。

主题

- 先决条件
- 安装第三方插
- 后续步骤

先决条件

安装第三方插件，请执行以下步骤：

- 已获取插件配置和许可文件，并将这些文件上传到 Amazon S3 存储桶。该存储桶必须与域 Amazon Web Services 区域 相同。

- 第三方插件是一种自定义插件。确保该域满足自定义插件的先决条件。

安装第三方插

要将第三方插件与 OpenSearch 服务域关联，必须先上传三个单独的软件包：许可证包、配置包和插件包。

- 许可证包包含与插件相关的许可信息或元数据，格式为.json 或.xml。
- 配置包包含插件配置文件以及支持资产和设置。这些文件定义了插件的行为方式或与 OpenSearch之集成的方式。
- 插件包包含已编译的插件二进制文件，即 OpenSearch 运行的可执行代码。这是插件功能的核心。

上传两个软件包后，您可以将插件和许可证与兼容的域相关联。

控制台

要将第三方插件与域相关联，请先将插件许可证和配置作为软件包导入。

安装第三方插

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择 Packes (包)。
3. 首先，导入许可证包。选择 Import package（导入软件包）。
4. 对于 Package 类型，选择许可。
5. 对于 P ackage 来源，输入 Amazon S3 中许可证 JSON 或 XML 文件的路径。
6. 选择 Import（导入）。软件包显示在 “软件包” 页面的 “许可证” 选项卡上。
7. 现在，导入插件配置。再次选择 “导入包”。
8. 对于 Package 类型，选择配置。
9. 对于 Pack age 来源，输入 Amazon S3 中插件配置 ZIP 文件的路径。
10. 选择 Import（导入）。
11. 最后，导入插件本身。选择 Import package（导入软件包）。
12. 对于 Package 类型，选择 Plugin。
13. 对于 P ackage 来源，输入 Amazon S3 中插件 ZIP 文件的路径。
14. 选择插件支持的 OpenSearch 引擎版本。
15. 选择 Import（导入）。

将第三方插件与域名关联

1. 现在，将插件许可证和配置与域相关联。在左侧导航窗格中，选择 域。
2. 选择域的名称来打开其集群配置。
3. 导航到 “插件” 选项卡。
4. 选择 “关联软件包”，然后选择刚刚导入的插件、许可证和配置包。
5. 选定选择。
6. 选择下一步。查看要关联的软件包，然后选择 “关联”。

CLI

首先，使用 create- package 命令创建一个包含插件许可证的新软件包。S3Key必须指向 Amazon S3 中包含许可文本或元数据的.json 或.xml 文件。

```
aws opensearch create-package \
  --package-name plugin-license-package \
  --package-type PACKAGE-LICENSE \
  --package-source S3BucketName=my-bucket,S3Key=licenses/my-plugin-license.json
```

再次使用 create- package 命令创建包含插件配置的软件包。S3Key必须指向 Amazon S3 中符合插件预期目录结构的.zip 文件。

```
aws opensearch create-package \
  --package-name plugin-config-package \
  --package-type PACKAGE-CONFIG \
  --package-source S3BucketName=my-bucket,S3Key=path/to/package.zip
```

再次使用 create- package 命令创建包含插件本身的软件包。S3Key必须指向 Amazon S3 中的.zip 文件。

```
aws opensearch create-package \
  --package-name plugin-package \
  --package-type ZIP-PLUGIN \
  --package-source S3BucketName=my-bucket,S3Key=path/to/package.zip
```

最后，使用 ass ociate-package 命令通过为每个域指定软件包，将合作伙伴插件、许可证和配置链接到兼容的域。 IDs 将插件 ID 指定为其他软件包的先决条件，这意味着插件 ID 必须先与域关联，然后再关联其他软件包。

```
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"]}]'
```

后续步骤

关联完成后，您可以在特定索引上启用该插件，也可以根据需要根据需要对其进行配置。要将第三方插件功能应用于特定索引，请在创建索引期间修改索引设置或更新现有索引。例如，如果您的第三方插件包含自定义分析器，请在索引设置中引用它。

要在多个索引中一致地应用插件功能，请使用包含插件配置的索引模板。请务必查阅插件文档，了解如何为您的 OpenSearch 设置配置其功能。

使用 SQL 查询您的 Amazon Serv OpenSearch ice 数据

您可以使用 SQL 查询您的 Amazon Serv OpenSearch ice，而不是使用基于 JSON 的OpenSearch 查询 DSL。如果您已经熟悉该语言，或者希望将您的域与使用该语言的应用程序集成，那么使用 SQL 查询非常有用。运行 Elasticsearch 6.5 OpenSearch 或更高版本的域上提供 SQL 支持。

> **Note**
>
> 本文档介绍了 Serv OpenSearch ice 与各种版本的 SQL 插件以及 JDBC 和 ODBC 驱动程序之间的版本兼容性。有关基本和复杂查询、函数、元数据查询和聚合函数的语法信息，请参阅开源OpenSearch文档。

使用下表查找每个 OpenSearch 版本支持的 SQL 插件版本支持的 SQL 插件版本。

OpenSearch

OpenSearch 版	SQL 插件版本	显著功能
2.19.0	2.19.0.0	
2.18.0	2.18.0.0	
2.17.0	2.17.0.0	

OpenSearch 版	SQL 插件版本	显著功能
2.15.0	2.15.0.0	
2.13.0	2.13.0.0	
2.11.0	2.11.0.0	添加对 PPL 语言和查询的支持
2.9.0	2.9.0.0	添加 Spark 连接器，支持表和 PromQL 函数
2.7.0	2.7.0.0	添加 `datasource` API
2.5.0	2.5.0.0	
2.3.0	2.3.0.0	添加 `maketime` 和 `makedate` 日期时间函数
1.3.0	1.3.0.0	支持默认查询限制大小，以及可从值列表中选择的 IN 子句
1.2.0	1.2.0.0	增加新的可视化响应格式协议
1.1.0	1.1.0.0	支持将匹配函数作为 SQL 和 PPL 中的过滤器
1.0.0	1.0.0.0	支持查询数据流

Open Distro for Elasticsearch

Elasticsearch 版本	SQL 插件版本	显著功能
7.10	1.13.0	window 函数 NULL FIRST 和 LAST、CAST()函数，SHOW 和 DESCRIBE 命令
7.9	1.11.0	添加额外的日期/时间函数，按关键字排序
7.8	1.9.0	
7.7	1.8.0	
7.3	1.3.0	多个字符串和数字运算符
7.1	1.1.0	

调用示例

若要使用 SQL 查询数据，请使用以下格式将 HTTP 请求发送至 `_sql`：

```
POST domain-endpoint/_plugins/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

> **Note**
>
> 如果您的域运行的是 Elasticsearch OpenSearch，那么格式为。`_opendistro/_sql`

说明和差异

对 `_plugins/_sql` 的调用在请求正文中包含索引名称，因此与 bulk、mget 和 msearch 操作具有相同的访问策略注意事项。与往常一样，在向 API 操作授予权限时，请遵循最低特权原则。

有关将 SQL 与精细访问控制结合使用的安全注意事项，请参阅 the section called “精细访问控制”。

S OpenSearch QL 插件包含许多可调整的设置。在 S OpenSearch ervice 中，使用`_cluster/settings`路径，而不是插件设置路径 (`_plugins/_query/settings`)：

```
PUT _cluster/settings
{
  "transient" : {
    "plugins.sql.enabled" : true
  }
}
```

对于旧式 Elasticsearch 域，请将 `plugins` 替换为 `opendistro`：

```
PUT _cluster/settings
{
  "transient" : {
    "opendistro.sql.enabled" : true
  }
}
```

SQL Workbench

SQL Workbench 是一个 OpenSearch 控制面板用户界面，让您可运行按需 SQL 查询、将 SQL 转换为等效 REST，以及查看结果并将结果保存为文本、JSON、JDBC 或 CSV 格式。有关更多信息，请参阅查询 Workbench。

SQL CLI

SQL CLI 是一个独立的 Python 应用程序，您可以使用 `opensearchsql` 命令启动该应用程序。有关安装、配置和使用步骤，请参阅 SQL CLI。

JDBC 驱动程序

通过 Java 数据库连接 (JDBC) 驱动程序，您可以将Serv OpenSearch ice 域与您最喜欢的商业智能 (BI) 应用程序集成。要下载驱动程序，请单击此处。有关更多信息，请参阅GitHub存储库。

ODBC 驱动程序

开放式数据库连接 (ODBC) 驱动程序是针对 Windows 和 macOS 的只读 ODBC 驱动程序，让您可将 Microsoft Excel 等商业智能和数据可视化应用程序连接到 SQL 插件。

您可以在 OpenSearch 构件页面下载示例运行驱动程序文件。有关安装驱动程序的信息，请参阅上的 SQL 存储库 GitHub。

在 Amazon Service 中搜索 k 最近邻 (k-NN)。 OpenSearch

Amazon S OpenSearch ervice 的 k-nn 简称为其关联的 k 最近邻算法，允许您搜索向量空间中的点，并通过欧氏距离或余弦相似性查找这些点的 “最近邻域”。使用案例包括推荐（例如，音乐应用程序中的“您可能喜欢的其他歌曲”功能）、图像识别和欺诈检测。

> **Note**
>
> 本文档简要介绍了 k-nn 插件的简要概述，以及将该插件与托管 OpenSearch 式Service 一起使用时的限制。有关 k-nn 插件的全面文档，包括简单和复杂的示例、参数参考以及有关简单和复杂的示例、参数参考以及有关简单和复杂的示例、参数参考以及对应于 k-nn 插件的全面文档，请参阅开源OpenSearch 文档。该开源文档还介绍了有关性能微调和 k-NN-specific集群设置的信息。

k-NN 入门

要使用 k-NN，您必须使用 `index.knn` 设置创建索引，然后添加一个或多个数据类型为 `knn_vector` 的字段。

```
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` 数据类型支持最多 10,000 个浮点数的单个列表，浮点数的数量由所需的 `dimension` 参数定义。创建索引后，向其中添加一些数据。

```
POST _bulk

{ "index": { "_index": "my-index", "_id": "1" } }
{ "my_vector1": [1.5, 2.5], "price": 12.2 }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "my_vector1": [2.5, 3.5], "price": 7.1 }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "my_vector1": [3.5, 4.5], "price": 12.9 }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "my_vector1": [5.5, 6.5], "price": 1.2 }
{ "index": { "_index": "my-index", "_id": "5" } }
{ "my_vector1": [4.5, 5.5], "price": 3.7 }
{ "index": { "_index": "my-index", "_id": "6" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 10.3 }
```

```
{ "index": { "_index": "my-index", "_id": "7" } }
{ "my_vector2": [2.5, 3.5, 5.6, 6.7], "price": 5.5 }
{ "index": { "_index": "my-index", "_id": "8" } }
{ "my_vector2": [4.5, 5.5, 6.7, 3.7], "price": 4.4 }
{ "index": { "_index": "my-index", "_id": "9" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 8.9 }
```

然后，您可以使用 knn 查询类型搜索数据。

```
GET my-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  }
}
```

在这种情况下，k 是您希望查询返回的邻居数，但您还必须包含 `size` 选项。否则，您将获得针对每个分区（和每个分段）的 k 个结果，而不是针对整个查询的 k 个结果。k-NN 支持的最大 k 值为 10,000。

如果将 knn 查询与其他子句混合，则收到的结果数可能少于 k 个。在此示例中，`post_filter` 子句将结果数从 2 减少到 1。

```
GET my-index/_search

{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  },
  "post_filter": {
```

```
    "range": {
      "price": {
        "gte": 6,
        "lte": 10
      }
    }
  }
}
```

如果您需要在保持最佳性能的同时处理大量查询，则可以使用 `_msearch` API，借助 JSON 构建批量搜索，然后发送单个请求以执行多个搜索：

```
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 }} } }
```

以下视频演示了如何为 K-NN 查询设置批量向量搜索。

k-NN 差异、调整和限制

OpenSearch 允许您使用 API 修改所有 k-nn 设置。`_cluster/settings`在 OpenSearch 服务中，您可以更改除`knn.memory.circuit_breaker.enabled`和之外的所有设置`knn.circuit_breaker.triggered`。k-nn 统计数据包含在 A ma CloudWatch zon 指标中。

特别是，请检查每个数据节点上的`KNNGraphMemoryUsage`指标`knn.memory.circuit_breaker.limit`和实例类型的可用 RAM。 OpenSearch Service 将实例的一半内存用于 Java 堆（最大堆大小为 32 GiB）。默认情况下，k-NN 使用最多 50% 的剩余一半，因此具有 32 GiB RAM 的实例类型可以容纳 8 GiB 的图形 (32 * 0.5 * 0.5)。如果图形内存使用量超过此值，性能可能会受到影响。

您可以将在 2.x 或更高版本上创建的 k-nn 索引迁移到版本 2.17 UltraWarm或更高版本的域中或在域上进行冷存储。

对于热索引，k-nn 索引的清除缓存 api 和预热 api 会被阻止。当对索引启动第一个查询时，它会从 Amazon S3 下载图形文件并将图表加载到内存中。同样，当图表的 TTL 过期时，文件会自动从内存中删除。

在 Amazon OpenSearch 服务中进行跨集群搜索

Amazon S OpenSearch ervice 中的跨集群搜索允许您跨多个连接的域执行查询和聚合。使用多个较小的域而不是单个大型的域通常更有意义，尤其是在运行不同类型的工作负载时。

特定于工作负载的域允许您执行以下任务：

- 通过为特定工作负载选择实例类型来优化每个域。
- 跨工作负载建立故障隔离边界。这意味着，如果某个工作负载发生了故障，故障将包含在该特定域内，不会影响其他工作负载。
- 更轻松地跨域扩展。

跨集群搜索支持 OpenSearch 仪表板，因此您可以跨所有域创建可视化和仪表板。对于在域名之间传输的搜索结果，您需要支付标准 Amazon 的数据传输费用。

> **Note**
>
> 开源 OpenSearch 还有用于跨集群搜索的文档。与托管的 Amazon S OpenSearch ervice 域相比，开源集群的设置差异很大。最值得注意的是，在 “ OpenSearch 服务” 中，您可以使用 Amazon Web Services Management Console 而不是通过 curl 配置跨集群连接。此外，除了精细的访问控制外，托管服务还使用 Amazon Identity and Access Management (IAM) 进行跨集群身份验证。因此，我们建议使用本文档，而不是开源 OpenSearch 文档，为您的域配置跨集群搜索。

主题

- 限制
- 跨集群搜索先决条件
- 跨集群搜索定价
- 设置连接
- 移除连接
- 设置安全性和示例演练
- OpenSearch 仪表板

限制

跨集群搜索有几个重要的限制：

- 您无法将 Elasticsearch 域名连接到某个 OpenSearch 域名。
- 您无法连接到自行管理的 OpenSearch/Elasticsearch 集群。
- 要跨区域连接域名，两个域名都必须位于 Elasticsearch 7.10 或更高版本或。 OpenSearch
- 一个域最多可以有 20 个传出连接。同样，一个域最多可以有 20 个传入连接。换句话说，一个域最多可以连接到 20 个其他域。
- 源域的版本必须与目标域相同或更高。如果您在两个域之间设置了双向连接，并且想要升级其中的一个域或同时升级这两个域，则必须先删除其中的一个连接。
- 不能将自定义字典或 SQL 用于跨集群搜索。
- 您不能 Amazon CloudFormation 使用连接域名。
- 不能在 M3 和可突增（T2 和 T3）实例上使用跨集群搜索。

跨集群搜索先决条件

在设置跨集群搜索之前，请确保域满足以下要求：

- 两个 OpenSearch 域名，或者版本 6.7 或更高版本上的 Elasticsearch 域名
- 已启用精细访问控制
- Node-to-node 已启用加密

跨集群搜索定价

跨域搜索不收取额外的费用。

设置连接

“源”域指的是发出跨集群搜索请求的域。换句话说，源域是向其发送初始搜索请求的域。

“目标”域是源域查询的域。

跨集群连接是从源域到目标域的单向连接。这意味着目标域无法查询源域。但是，可以在相反的方向上设置另一个连接。

源域创建到目标域的“出站”连接。目标域接收来自源域的“入站”连接请求。

设置连接

1. 在域控制面板上，选择域，然后前往连接选项卡。
2. 在 Outbound connections（出站连接）部分中，选择 Request（请求）。
3. 对于 Connection alias（连接别名），输入您的连接的名称。
4. 在连接您 Amazon Web Services 账户 和地区的域名之间进行选择，也可以选择连接到其他账户或地区的域名。
   - 要连接到您 Amazon Web Services 账户 和地区的集群，请从下拉菜单中选择该域并选择请求。
   - 要连接到其他 Amazon Web Services 账户 或区域的集群，请选择远程域的 ARN，然后选择请求。要跨区域连接域名，两个域名都必须运行 Elasticsearch 7.10 或更高版本或。 OpenSearch
5. 要在执行集群查询时跳过不可用集群，请选择跳过不可用。此设置可确保即使一个或多个远程集群发生故障，跨集群查询仍能返回部分结果。
6. 跨集群搜索首先验证连接请求，以确保满足先决条件。如果发现域不兼容，则连接请求将进入 `Validation failed` 状态。
7. 连接请求验证成功后，将发送到目标域，在此需要进行审批。在此审批进行之前，连接将保持 `Pending acceptance` 的状态。当连接请求被目标域接受后，状态将更改为 `Active`，且目标域变为可供查询。
   - 域页面显示了目标域的整体域运行状况和实例运行状况的详细信息。只有域拥有者具备创建、查看、删除和监控域之间的连接的灵活性。

连接建立之后，在所连接域的节点之间流动的任何流量都将进行加密。如果将 VPC 域连接到非 VPC 域，且非 VPC 域是可接收 Internet 流量的公有终端节点，则域之间的跨集群流量仍然是加密的并且是安全的。

移除连接

移除连接会停止对其索引执行任何跨集群操作。

1. 在域控制面板上，转到 Connections（连接）选项卡。
2. 选择要移除的域连接，然后选择 Delete，然后确认删除。

您可以在源域或目标域上执行这些步骤以删除连接。删除连接之后，仍会显示 15 天，其状态为 `Deleted`。

不能删除具有活动跨集群连接的域。要删除域，请首先从该域中删除所有传入和传出连接。这是为了确保在删除域之前考虑到了跨集群域用户。

设置安全性和示例演练

1. 您将跨集群搜索请求发送到源域。
2. 源域根据其域访问策略评估该请求。由于跨集群搜索需要精细访问控制，因此我们建议对源域采用开放访问策略。

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:account:domain/src-domain/*"
    }
  ]
}
```

> **Note**
>
> 如果路径中包含远程索引，则必须在域 ARN 中对 URI 进行 URL 编码。例如，使用 `arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst%3Aremote_index` 而不是 `arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst:remote_index`。

如果除了精细访问控制，还选择使用限制性访问策略，那么策略必须至少允许访问 `es:ESHttpGet`。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:region:account:domain/src-domain/*"
    }
  ]
}
```

3. 源域中的精细访问控制将评估请求：

   - 该请求是否采用有效的 IAM 或 HTTP 基本凭证签名？
   - 如果是这样，用户是否有权执行搜索和访问数据？

   如果该请求仅搜索目标域中的数据（例如，`dest-alias:dest-index/_search`），则只需目标域的权限。

如果该请求在两个域中搜索数据（例如，source-index,dest-alias:dest-index/_search），则需要两个域的权限。

在精细访问控制中，除相关索引的标准 read 或 search 权限之外，用户还必须拥有 indices:admin/shards/search_shards 权限。

4. 源域将请求传递到目标域。目标域根据其域访问策略评估该请求。必须包含对目标域的 es:ESCrossClusterGet 权限：

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/dst-domain"
    }
  ]
}
```

   请确保对 /dst-domain，而不是 /dst-domain/* 应用了 es:ESCrossClusterGet 权限。

   但是，此最低策略仅允许跨集群搜索。要执行其他操作，例如，为文档编制索引和执行标准搜索，需要额外的权限。我们建议对目标域采取以下策略：

   JSON

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

```
            "*"
          ]
        },
        "Action": [
          "es:ESHttp*"
        ],
        "Resource": "arn:aws:es:region:account:domain/dst-domain/*"
      },
      {
        "Effect": "Allow",
        "Principal": {
          "AWS": "*"
        },
        "Action": "es:ESCrossClusterGet",
        "Resource": "arn:aws:es:region:account:domain/dst-domain"
      }
    ]
}
```

> **Note**
>
> 默认情况下，作为加密的一部分，域之间的所有跨集群搜索请求都会在传输过程中进行node-to-node加密。

5. 目标域执行搜索，并将结果返回到源域。
6. 源域将自己的结果（如果有）与目标域的结果相结合，并将它们返回给您。
7. 我们推荐采用 Postman 来测试请求：
   - 在目标域上，为文档编制索引：

     ```
     POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1

     {
       "Dracula": "Bram Stoker"
     }
     ```

   - 要从源域查询此索引，请在查询内包含目标域的连接别名。

     ```
     GET https://src-domain.us-east-1.es.amazonaws.com/<connection_alias>:books/
     _search
     ```

```
{
    ...
  "hits": [
    {
      "_index": "source-destination:books",
      "_type": "_doc",
      "_id": "1",
      "_score": 1,
      "_source": {
        "Dracula": "Bram Stoker"
      }
    }
  ]
}
```

您可以在域控制面板的连接选项卡上找到连接别名。

- 如果要在连接别名为 `cluster_b` 的 `domain-a -> domain-b` 与连接别名为 `cluster_c` 的 `domain-a -> domain-c` 之间设置连接，请按以下方式搜索 `domain-a`、`domain-b` 和 `domain-c`：

```
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"
    }
  }
}
```

响应

```
{
  "took": 150,
  "timed_out": false,
  "_shards": {
    "total": 3,
    "successful": 3,
    "failed": 0,
    "skipped": 0
```

```
  },
  "_clusters": {
    "total": 3,
    "successful": 3,
    "skipped": 0
  },
  "hits": {
    "total": 3,
    "max_score": 1,
    "hits": [
      {
        "_index": "local_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 1,
        "_source": {
          "user": "domino",
          "message": "This is message 1",
          "likes": 0
        }
      },
      {
        "_index": "cluster_b:b_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 2,
        "_source": {
          "user": "domino",
          "message": "This is message 2",
          "likes": 0
        }
      },
      {
        "_index": "cluster_c:c_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 3,
        "_source": {
          "user": "domino",
          "message": "This is message 3",
          "likes": 0
        }
      }
    ]
```

```
  }
}
```

如果您没有选择在连接设置中跳过不可用集群，搜索的所有目标集群必须可用才能成功运行搜索请求。否则，整个请求将失败——即使只有一个域不可用，也不会返回任何搜索结果。

OpenSearch 仪表板

您可以采用与单个域相同的方式可视化来自多个相连接域的数据，但必须使用 `connection-alias:index` 访问远程索引。因此，索引模式必须与 `connection-alias:index` 相匹配。

学习 Amazon S OpenSearch ervice 的排名

OpenSearch 使用名为 BM-25 的概率排名框架来计算相关性得分。如果独特的关键词在文档中出现较为频繁，BM-25 会为该文档分配更高的相关性分数。但是，此框架并没有考虑点击数据等用户行为，这可以进一步提高相关性。

学习排名是一个开源插件，可以让您使用机器学习和行为数据来调整文档的相关性。它使用 Ranklib 库中的 XGBoost 模型来重新评分搜索结果。Elasticsearch LTR 插件最初由 Conn OpenSource ec tions 开发，且 Wikimedia Foundation、Bonsai 和 Yelp Engineering 做出了重大贡献。该插件的 OpenSearch 版本源自 Elasticsearch LTR 插件。

学习排名需要 Elasticsearch 7.7 OpenSearch 或更高版本。要使用学习排名插件，您必须具有完全的管理员权限。要了解更多信息，请参阅the section called “修改主用户”。

> **Note**
>
> 本文档概述了 Learning to Rank 插件，此外还提供了有关如何开始使用此插件的帮助。有关完整文档，包括详细步骤和 API 说明，请参阅学习排名文档。

主题

- 学习排名入门
- 学习排名 API

学习排名入门

您需要提供判断列表，准备训练数据集，并在 Amazon S OpenSearch ervice 外训练模型。蓝色部分出现在 S OpenSearch ervice 之外：

步骤 1：初始化插件

要初始化学习排名插件，请将以下请求发送到您的S OpenSearch ervice 域：

```
PUT _ltr
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : ".ltrstore"
}
```

此命令创建一个隐藏的 `.ltrstore` 索引，用于存储元数据信息（如功能集和模型）。

步骤 2：创建判断列表

> **Note**
>
> 您必须在 S OpenSearch ervice 之外执行此步骤。

判断列表是机器学习模型从中学习的示例集合。您的判断列表中应包含对您很重要的关键词以及每个关键词的一组分级文档。

在此示例中，我们提供了电影数据集的判断列表。等级为 4 表示完全匹配。等级为 0 表示匹配不佳。

等级	Keyword	文档编号	电影名称
4	rambo	7555	Rambo
3	rambo	1370	Rambo III
3	rambo	1369	Rambo：First Blood 第 II 部分
3	rambo	1368	First Blood

准备以下格式的判断列表：

```
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"
```

有关判断列表的更完整示例，请参阅电影判断。

您可以在人类注释者的帮助下手动创建此判断列表，或者从分析数据以编程方式推断它。

步骤 3：构建功能集

功能是与文档相关性相对应的字段，例如 `title`、`overview`、`popularity score`（视图数），依此类推。

为每个功能构建具有 Mustache 模板的功能集。有关功能的更多信息，请参阅使用功能。

在此示例中，我们构建了包含 `title` 和 `overview` 字段的 `movie_features` 功能集：

```
POST _ltr/_featureset/movie_features
{
  "featureset" : {
      "name" : "movie_features",
      "features" : [
        {
          "name" : "1",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "title" : "{{keywords}}"
            }
          }
        },
        {
          "name" : "2",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "overview" : "{{keywords}}"
            }
          }
        }
      ]
    }
}
```

如果您查询原始 `.ltrstore` 索引，则返回您的功能集：

```
GET _ltr/_featureset
```

步骤 4：记录功能值

功能值是 BM-25 为每个功能计算的相关性分数。

将功能集和判断列表组合起来记录功能值。有关日志记录功能的更多信息，请参阅日志记录功能分数。

在此示例中，bool 查询使用筛选器检索分级文档，然后使用 sltr 查询选择功能集。ltr_log 查询将文档和功能组合在一起，以记录相应的功能值：

```
POST tmdb/_search
{
  "_source": {
    "includes": [
      "title",
      "overview"
    ]
  },
  "query": {
    "bool": {
      "filter": [
        {
          "terms": {
            "_id": [
              "7555",
              "1370",
              "1369",
              "1368"
            ]
          }
        },
        {
          "sltr": {
            "_name": "logged_featureset",
            "featureset": "movie_features",
            "params": {
              "keywords": "rambo"
            }
          }
        }
      ]
    }
  },
  "ext": {
    "ltr_log": {
```

```
      "log_specs": {
        "name": "log_entry1",
        "named_query": "logged_featureset"
      }
    }
  }
}
```

示例响应可能与以下内容下类似：

```
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4,
      "relation" : "eq"
    },
    "max_score" : 0.0,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1368",
        "_score" : 0.0,
        "_source" : {
          "overview" : "When former Green Beret John Rambo is harassed by local law
 enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
 rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
 sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
```

```
                "name" : "1"
              },
              {
                "name" : "2",
                "value" : 10.558305
              }
            ]
          }
        ]
      },
      "matched_queries" : [
        "logged_featureset"
      ]
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "7555",
      "_score" : 0.0,
      "_source" : {
        "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
        "title" : "Rambo"
      },
      "fields" : {
        "_ltrlog" : [
          {
            "log_entry1" : [
              {
                "name" : "1",
                "value" : 11.2569065
              },
              {
                "name" : "2",
                "value" : 9.936821
              }
            ]
          }
        ]
      },
      "matched_queries" : [
```

```
          "logged_featureset"
        ]
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 0.0,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1",
                  "value" : 6.334839
                },
                {
                  "name" : "2",
                  "value" : 10.558305
                }
              ]
            }
          ]
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 0.0,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
```

```
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
         "title" : "Rambo III"
       },
       "fields" : {
         "_ltrlog" : [
           {
             "log_entry1" : [
               {
                 "name" : "1",
                 "value" : 9.425955
               },
               {
                 "name" : "2",
                 "value" : 11.262714
               }
             ]
           }
         ]
       },
       "matched_queries" : [
         "logged_featureset"
       ]
     }
   ]
  }
}
```

在上一个示例中，第一个功能没有功能值，因为关键字 “rambo” 未出现在 ID 等于 1368 的文档的标题字段中。训练数据中缺少功能值。

步骤 5：创建训练数据集

> **Note**
>
> 您必须在 S OpenSearch ervice 之外执行此步骤。

下一步是将判断列表和功能值组合在一起以创建训练数据集。如果您的原始判断列表类似于以下内容：

```
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
```

将其转换为最终的训练数据集，如下所示：

```
4 qid:1 1:12.318474 2:10.573917 # 7555 rambo
3 qid:1 1:10.357875 2:11.950391 # 1370 rambo
3 qid:1 1:7.010513 2:11.220095 # 1369 rambo
3 qid:1 1:0.0 2:11.220095 # 1368 rambo
```

您可以手动执行此步骤，也可以编写程序来自动执行该步骤。

步骤 6：选择算法并构建模型

> **Note**
>
> 您必须在 S OpenSearch ervice 之外执行此步骤。

训练数据集到位后，下一步是使用 XGBoost 或 Ranklib 库构建模型。 XGBoost Ranklib 库允许您构建常见的模型，如 LambDamart、随机森林等。

有关使用 XGBoost 和 Ranklib 构建模型的步骤，请分别参见XGBoost和RankLib文档。要使用 Amazon SageMaker 构建 XGBoost 模型，请参阅XGBoost算法。

步骤 7：部署模型

构建模型后，将其部署到学习排名插件中。有关部署模型的更多信息，请参阅上传已训练模型。

在此示例中，我们使用 Ranklib 库构建了 `my_ranklib_model` 模型。

```
POST _ltr/_featureset/movie_features/_createmodel?pretty
{
  "model": {
    "name": "my_ranklib_model",
    "model": {
      "type": "model/ranklib",
      "definition": """## LambdaMART
## No. of trees = 10
## No. of leaves = 10
## No. of threshold candidates = 256
## Learning rate = 0.1
```

```
## Stop early = 100

<ensemble>
   <tree id="1" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-2.0</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-2.0</output>
               </split>
               <split pos="right">
                  <output>-2.0</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>2.0</output>
         </split>
      </split>
   </tree>
   <tree id="2" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.67031991481781</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.67031991481781</output>
```

```
                </split>
                <split pos="right">
                   <output>-1.6703200340270996</output>
                </split>
             </split>
          </split>
          <split pos="right">
             <output>1.6703201532363892</output>
          </split>
       </split>
    </tree>
    <tree id="3" weight="0.1">
       <split>
          <feature>2</feature>
          <threshold>10.573917</threshold>
          <split pos="left">
             <output>1.479954481124878</output>
          </split>
          <split pos="right">
             <feature>1</feature>
             <threshold>7.010513</threshold>
             <split pos="left">
                <feature>1</feature>
                <threshold>0.0</threshold>
                <split pos="left">
                   <output>-1.4799546003341675</output>
                </split>
                <split pos="right">
                   <output>-1.479954481124878</output>
                </split>
             </split>
             <split pos="right">
                <output>-1.479954481124878</output>
             </split>
          </split>
       </split>
    </tree>
    <tree id="4" weight="0.1">
       <split>
          <feature>1</feature>
          <threshold>10.357875</threshold>
          <split pos="left">
             <feature>1</feature>
             <threshold>0.0</threshold>
```

```
            <split pos="left">
               <output>-1.3569872379302979</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.3569872379302979</output>
               </split>
               <split pos="right">
                  <output>-1.3569872379302979</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>1.3569873571395874</output>
         </split>
      </split>
   </tree>
   <tree id="5" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.2721362113952637</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.2721363306045532</output>
               </split>
               <split pos="right">
                  <output>-1.2721363306045532</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>1.2721362113952637</output>
         </split>
      </split>
```

```
    </tree>
    <tree id="6" weight="0.1">
       <split>
          <feature>1</feature>
          <threshold>10.357875</threshold>
          <split pos="left">
             <feature>1</feature>
             <threshold>7.010513</threshold>
             <split pos="left">
                <feature>1</feature>
                <threshold>0.0</threshold>
                <split pos="left">
                   <output>-1.2110036611557007</output>
                </split>
                <split pos="right">
                   <output>-1.2110036611557007</output>
                </split>
             </split>
             <split pos="right">
                <output>-1.2110037803649902</output>
             </split>
          </split>
          <split pos="right">
             <output>1.2110037803649902</output>
          </split>
       </split>
    </tree>
    <tree id="7" weight="0.1">
       <split>
          <feature>1</feature>
          <threshold>10.357875</threshold>
          <split pos="left">
             <feature>1</feature>
             <threshold>7.010513</threshold>
             <split pos="left">
                <feature>1</feature>
                <threshold>0.0</threshold>
                <split pos="left">
                   <output>-1.165616512298584</output>
                </split>
                <split pos="right">
                   <output>-1.165616512298584</output>
                </split>
             </split>
```

```
            <split pos="right">
               <output>-1.165616512298584</output>
            </split>
         </split>
         <split pos="right">
            <output>1.165616512298584</output>
         </split>
      </split>
   </tree>
   <tree id="8" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.131177544593811</output>
               </split>
               <split pos="right">
                  <output>-1.131177544593811</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.131177544593811</output>
            </split>
         </split>
         <split pos="right">
            <output>1.131177544593811</output>
         </split>
      </split>
   </tree>
   <tree id="9" weight="0.1">
      <split>
         <feature>2</feature>
         <threshold>10.573917</threshold>
         <split pos="left">
            <output>1.1046180725097656</output>
         </split>
         <split pos="right">
            <feature>1</feature>
```

```
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.1046180725097656</output>
               </split>
               <split pos="right">
                  <output>-1.1046180725097656</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.1046180725097656</output>
            </split>
         </split>
      </split>
   </tree>
   <tree id="10" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.0838804244995117</output>
               </split>
               <split pos="right">
                  <output>-1.0838804244995117</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.0838804244995117</output>
            </split>
         </split>
         <split pos="right">
            <output>1.0838804244995117</output>
         </split>
      </split>
   </tree>
</ensemble>
```

```
"""
    }
  }
}
```

要查看模型，请发送以下请求：

```
GET _ltr/_model/my_ranklib_model
```

步骤 8：通过学习排名进行搜索

部署模型后，您已准备好进行搜索。

通过您正在使用的功能和您要执行的模型执行 `sltr` 查询：

```
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"
        }
      }
    }
  }
}
```

通过学习排名，您会看到 “Rambo” 作为第一个结果，因为我们已经为它分配了判断列表中的最高分数：

```
{
  "took" : 12,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 7,
      "relation" : "eq"
    },
    "max_score" : 13.096414,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "7555",
        "_score" : 13.096414,
        "_source" : {
          "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
          "title" : "Rambo"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 11.17245,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
          "title" : "Rambo III"
        }
      },
```

```
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1368",
        "_score" : 10.442155,
        "_source" : {
          "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 10.442155,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "31362",
        "_score" : 7.424202,
        "_source" : {
          "overview" : "It is 1985, and a small, tranquil Florida town is being rocked
by a wave of vicious serial murders and bank robberies. Particularly sickening to the
authorities is the gratuitous use of violence by two “Rambo” like killers who dress
themselves in military garb. Based on actual events taken from FBI files, the movie
depicts the Bureau’s efforts to track down these renegades.",
          "title" : "In the Line of Duty: The F.B.I. Murders"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "13258",
```

```
        "_score" : 6.43182,
        "_source" : {
          "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
 his family's stifling home life when he encounters Lee Carter (Will Poulter), the
 school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
 to make cinematic history by filming his own action-packed video epic. Together, these
 two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
 — and sometimes mishap-filled — cinematic adventure has begun to take on a life of its
 own!""",
          "title" : "Son of Rambow"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "61410",
        "_score" : 3.9719706,
        "_source" : {
          "overview" : "It's South Africa 1990. Two major events are about to happen:
 The release of Nelson Mandela and, more importantly, it's Spud Milton's first year
 at an elite boys only private boarding school. John Milton is a boy from an ordinary
 background who wins a scholarship to a private school in Kwazulu-Natal, South Africa.
 Surrounded by boys with nicknames like Gecko, Rambo, Rain Man and Mad Dog, Spud has
 his hands full trying to adapt to his new home. Along the way Spud takes his first
 tentative steps along the path to manhood. (The path it seems could be a rather long
 road). Spud is an only child. He is cursed with parents from well beyond the lunatic
 fringe and a senile granny. His dad is a fervent anti-communist who is paranoid that
 the family domestic worker is running a shebeen from her room at the back of the
 family home. His mom is a free spirit and a teenager's worst nightmare, whether it's
 shopping for Spud's underwear in the local supermarket",
          "title" : "Spud"
        }
      }
    ]
  }
}
```

如果您没有使用学习排名插件进行搜索，OpenSearch 会返回不同的结果：

```
POST tmdb/_search
{
  "_source": {
    "includes": ["title", "overview"]
```

```
  },
  "query": {
    "multi_match": {
      "query": "Rambo",
      "fields": ["title", "overview"]
    }
  }
}
```

```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 5,
      "relation" : "eq"
    },
    "max_score" : 11.262714,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 11.262714,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
 find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
 his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
 when Trautman is captured.",
          "title" : "Rambo III"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "7555",
        "_score" : 11.2569065,
```

```
        "_source" : {
          "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
          "title" : "Rambo"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1368",
        "_score" : 10.558305,
        "_source" : {
          "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 10.558305,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "13258",
        "_score" : 6.4600153,
        "_source" : {
          "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
his family's stifling home life when he encounters Lee Carter (Will Poulter), the
school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
```

```
 to make cinematic history by filming his own action-packed video epic. Together, these
 two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
 — and sometimes mishap-filled — cinematic adventure has begun to take on a life of its
 own!""",
          "title" : "Son of Rambow"
        }
      }
    ]
  }
}
```

根据您认为模型的执行情况，调整判断列表和功能。然后，重复步骤 2—8 以随着时间的推移改进排名结果。

学习排名 API

使用学习排名操作以编程方式使用功能集和模型。

创建存储

创建隐藏的 `.ltrstore` 索引，用于存储元数据信息（如功能集和模型）。

```
PUT _ltr
```

删除存储

删除隐藏的 `.ltrstore` 索引并重置插件。

```
DELETE _ltr
```

创建功能集

创建功能集。

```
POST _ltr/_featureset/<name_of_features>
```

删除功能集

删除功能集。

```
DELETE _ltr/_featureset/<name_of_feature_set>
```

获取功能集

检索功能集。

```
GET _ltr/_featureset/<name_of_feature_set>
```

创建模型

创建模型。

```
POST _ltr/_featureset/<name_of_feature_set>/_createmodel
```

删除模型

删除模型。

```
DELETE _ltr/_model/<name_of_model>
```

获取模型

检索模型。

```
GET _ltr/_model/<name_of_model>
```

获取统计信息

提供有关插件操作方式的信息。

```
GET _ltr/_stats
```

还可以使用筛选条件来检索单个统计数据：

```
GET _ltr/_stats/<stat>
```

此外，可以将信息限制为集群中的单个节点：

```
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
    }
  }
}
```

统计数据在两个级别（节点和集群）提供，如下表所示：

节点级统计

字段名称	描述
request_total_count	排名请求的总计数。
request_error_count	不成功请求的总计数。
cache	所有缓存（功能、功能集、模型）的统计数据。当用户查询插件并且模型已加载到内存中时，会发生缓存命中。
cache.eviction_count	缓存移出次数。
cache.hit_count	缓存命中次数。
cache.miss_count	缓存丢失次数。当用户查询插件并且模型尚未加载到内存中时，会发生缓存丢失。
cache.entry_count	缓存中的条目数。
cache.memory_usage_in_bytes	字节中使用的总内存。
cache.cache_capacity_reached	指示是否达到缓存限制。

集群级统计数据

字段名称	描述
存储	指示功能集和模型元数据的存储位置。（原定设置为 ".ltrstore"。否则，它的前缀为 ".ltrstore_"，并带有用户提供的名称）。
stores.status	索引状态。
stores.feature_sets	功能集数。
stores.features_count	功能数。

字段名称	描述
stores.model_count	型号数。
状态	基于功能存储索引状态（红色、黄色或绿色）和断路器状态（打开或关闭）的插件状态。
cache.cache_capacity_reached	指示是否达到缓存限制。

获取缓存统计信息

返回有关缓存和内存使用情况的统计信息。

```
GET _ltr/_cachestats

{
    "_nodes": {
        "total": 2,
        "successful": 2,
        "failed": 0
    },
    "cluster_name": "opensearch-cluster",
    "all": {
        "total": {
            "ram": 612,
            "count": 1
        },
        "features": {
            "ram": 0,
            "count": 0
        },
        "featuresets": {
            "ram": 612,
            "count": 1
        },
        "models": {
            "ram": 0,
            "count": 0
        }
    },
    "stores": {
        ".ltrstore": {
```

```
            "total": {
                "ram": 612,
                "count": 1
            },
            "features": {
                "ram": 0,
                "count": 0
            },
            "featuresets": {
                "ram": 612,
                "count": 1
            },
            "models": {
                "ram": 0,
                "count": 0
            }
        }
    },
    "nodes": {
        "ejF6uutERF20wOFNOXB61A": {
            "name": "opensearch1",
            "hostname": "172.18.0.4",
            "stats": {
                "total": {
                    "ram": 612,
                    "count": 1
                },
                "features": {
                    "ram": 0,
                    "count": 0
                },
                "featuresets": {
                    "ram": 612,
                    "count": 1
                },
                "models": {
                    "ram": 0,
                    "count": 0
                }
            }
        },
        "Z2RZNWRLSveVcz2c6lHf5A": {
            "name": "opensearch2",
            "hostname": "172.18.0.2",
```

```
            "stats": {
                ...
            }
        }
    }
}
```

清除缓存

清除插件缓存。使用此选项可刷新模型。

```
POST _ltr/_clearcache
```

在 Amazon S OpenSearch ervice 异步搜索

通过对 Amazon Ser OpenSearch vice 的异步搜索，您可以提交在后台执行的搜索查询，监控请求的进度，并在稍后阶段检索结果。您可以在搜索完成之前检索部分结果变得可用时检索这些结果。搜索完成后，保存结果以供日后检索和分析。

异步搜索需要 OpenSearch 1.0 或更高版本，或者 Elasticsearch 7.10 或更高版本。

本文档简要概述了异步搜索。此外还介绍了使用托管式 Amazon Service 域而不是开源 Amazon S OpenSearch ervice 域进行异步搜索 OpenSearch 的限制。有关异步搜索的完整文档，包括可用设置、权限和完整的 API 参考，请参阅 OpenSearch 文档中的 Asynchr onous search。

搜索调用示例

要执行异步搜索，请使用下列格式将 HTTP 请求发送到 `_plugins/_asynchronous_search`：

```
POST opensearch-domain/_plugins/_asynchronous_search
```

> **Note**
>
> 如果您使用的是 Elasticsearch 7.10 而不是某个 Elasticsearch OpenSearch 版本，请将所有异步搜索`_opendistro`请求`_plugins`替换为。

您可以指定以下异步搜索选项：

选项	描述	默认值	必需
wait_for_completion_timeout	指定您计划等待结果的时间量。您可以看到在这段时间内获得的任何结果，就像在正常搜索中一样。您可以根据 ID 轮询剩余的结果。最大值为 300 秒。	1 秒	否
keep_on_completion	指定是否要在搜索完成后将结果保存在集群中。您可以稍后检查存储的结果。	false	否
keep_alive	指定结果保存在集群中的时间量。例如，2d 意味着结果存储在集群中 48 小时。保存的搜索结果将在此时间段之后或取消搜索时被删除。请注意，这包括查询运行时。如果此时查询超出，则进程将自动取消此查询。	12 小时	否

示例请求

```
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**
>
> 适用于标准 _search 的所有请求参数查询。如果您使用的是 Elasticsearch 7.10 而不是某个 OpenSearch 版本，请替换为。_plugins _opendistro

异步搜索权限

支持异步搜索访问权限的精细控制。有关混合和匹配权限以适应您的使用案例的详细信息，请参阅异步搜索安全。

对于启用了细粒度访问控制的域，您需要角色的以下最低权限：

```
# 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'
```

对于已禁用细粒度访问控制的域，请使用您的 IAM 访问权限和私有密钥签署所有请求。您可以使用异步搜索 ID 访问结果。

异步搜索设置

OpenSearch 允许您使用 `_cluster/settings` API 更改所有可用的异步搜索设置。在 S OpenSearch ervice 中，您只能更改以下设置：

- `plugins.asynchronous_search.node_concurrent_running_searches`
- `plugins.asynchronous_search.persist_search_failures`

跨集群搜索

您可以跨集群执行异步搜索，但具有以下次要限制：

- 您只能在源域上运行异步搜索。

- 作为跨群集搜索查询的一部分，您不能最大限度地减少网络往返行程。

如果要在连接别名为 cluster_b 的 domain-a -> domain-b 与连接别名为 cluster_c 的 domain-a -> domain-c 之间设置连接，请按以下方式异步搜索 domain-a、domain-b 和 domain-c：

```
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": []
    }
  }
}
```

响应

```
{
  "id" :
 "Fm9pYzJyVG91U19xb0hIQUJnMHJfRFEAAAAAAAknghQ1OWVBczNZQjVEa2dMYTBXaTdEagAAAAAAAAAB",
  "state" : "RUNNING",
  "start_time_in_millis" : 1609329314796,
  "expiration_time_in_millis" : 1609761314796
}
```

有关更多信息，请参阅 the section called “跨集群搜索”。

UltraWarm

使用 UltraWarm 索引的异步搜索继续工作。有关更多信息，请参阅 the section called “UltraWarm 存储”。

> **Note**
>
> 您可以监控中的 Asynchronous 统计信息 CloudWatch。有关指标的完整列表，请参阅the section called “异步搜索指标”。

Amazon S OpenSearch ervice 中的时间点搜索

时间点（PIT）是一种搜索类型，可让您对固定时间的数据集运行不同的查询。通常，当您在不同的时间点对同一个索引运行相同的查询时，由于文档会不断被索引、更新和删除，所以会收到不同的结果。使用 PIT，您可以根据数据集的恒定状态进行查询。

PIT 搜索的主要用途是与 `search_after` 功能结合使用。这是中的首选分页方法 OpenSearch，特别是对于深度分页，因为它对冻结时间的数据集进行操作，未绑定到查询，并且支持向前和向后的一致分页。您可以在运行 2.5 OpenSearch 版本的域中使用 PIT。

> **Note**
>
> 本主题概述了 PIT，以及在托管式 Amazon Service 域（而不是在自主管理型 Amazon S OpenSearch ervice 域）上使用 PIT 时的一些注意事项 OpenSearch 。有关 PIT 的完整文档，包括全面的 API 参考，请参阅开源 OpenSearch 文档中的 Point in Tim e。

注意事项

在配置 PIT 搜索时，请考虑以下事项：

- 如果从运行 2.3 OpenSearch 版本的域升级，并且需要对 PIT 操作进行精细的访问控制，则需要手动添加这些操作和角色。
- PIT 没有弹性。节点重启、节点终止、蓝绿部署和 OpenSearch 进程重启会导致所有 PIT 数据丢失。
- 如果分片在蓝绿部署期间重新定位，则仅将实时数据段传输到新节点。PIT 持有的分片段（包括独占分片和与实时数据共享的分片）仍保留在旧节点上。
- PIT 搜索目前不适用于异步搜索。

创建 PIT

要运行 PIT 查询，请使用以下格式将 HTTP 请求发送至 `_search/point_in_time`：

```
POST opensearch-domain/my-index/_search/point_in_time?keep_alive=time
```

可以指定以下 PIT 选项：

选项	描述	默认值	必需
keep_alive	保留 PIT 的时间长度。每次使用搜索请求访问 PIT 时，PIT 的生命周期都会根据 keep_alive 参数的时间长度延长。创建 PIT 时此查询参数是必需的，但在搜索请求中则是可选的。		是
preference	指定用于执行搜索的节点或分片的字符串。	随机	否
routing	指定将搜索请求传输到某个具体分片的字符串。	该文件的 _id	否
expand_wildcards	指定可以匹配通配符模式的索引类型的字符串。支持逗号分隔值。有效值如下所示： • all：匹配任何索引或数据流，包括隐藏的索引或数据流。 • open：匹配开放的、非隐藏的索引或非隐藏的数据流。 • closed：匹配封闭的、非隐藏的索引或非隐藏的数据流。 • hidden：匹配隐藏的索引或数据流。必须与开放和/或封闭结合。 • none：不接受通配符模式。	open	否
allow_partial_pit_creation	用于指定是否创建部分失败的 PIT 的布尔值。	true	否

示例响应

```
{
    "pit_id":
 "o463QQEPbXktaW5kZXgtMDAwMDAxFnNOWU43ckt3U3IyaFVpbGE1UWEtMncAFjFyeXBsRGJmVFM2RTB6eVg1aVVqQncAA
    "_shards": {
        "total": 1,
        "successful": 1,
```

```
        "skipped": 0,
        "failed": 0
    },
    "creation_time": 1658146050064
}
```

当您创建 PIT 时，您会在响应中收到 PIT ID。这是您用来通过 PIT 进行搜索的 ID。

时间点权限

PIT 支持精细访问控制。如果您要升级到 2.5 OpenSearch 版域并需要精细访问控制，则需要手动创建具有以下权限的角色：

```
# Allows users to use all point in time search search functionality
point_in_time_full_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - '*'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/point_in_time/readall"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"


# Allows users to use point in time search search functionality for specific index
# All type operations like list all PITs, delete all PITs are not supported in this
 case

point_in_time_index_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - 'my-index-1'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"
```

对于 OpenSearch 版本为 2.5 及更高版本的域，您可以使用内置`point_in_time_full_access`角色。有关更多信息，请参阅 OpenSearch 文档中的安全模型。

PIT 设置

OpenSearch 允许您使用 `_cluster/settings` API 更改所有可用的 PI T 设置。在 S OpenSearch ervice 中，您当前无法修改设置。

跨集群搜索

您可以 PITs 跨集群创建 PITs、使用 PIT 搜索 IDs PITs、列出和删除集群，但有以下几项限制：

- 您 PITs 只能在源域上列出和删除所有。
- 作为跨群集搜索查询的一部分，您不能最大限度地减少网络往返行程。

有关更多信息，请参阅 the section called “跨集群搜索”。

UltraWarm

使用 PIT UltraWarm 索引的 PIT 搜索继续工作。有关更多信息，请参阅 the section called “UltraWarm 存储”。

> **Note**
>
> 您可以在中监控 PIT 搜索统计信息 CloudWatch。有关指标的完整列表，请参阅the section called “时间点指标”。

在 Amazon Service OpenSearch 中执行语义搜索

自 2.9 OpenSearch 版本 2.9 开始，您可以使用语义搜索帮助理解搜索查询并提高搜索相关性。您可以通过以下两种方式之一使用语义搜索：神经搜索和 k 最近邻（k-NN）搜索。

借助 OpenSearch 服务，您可以为 Amazon Web Services 服务和外部服务设置 AI 连接器。也可以通过控制台使用 Amazon CloudFormation 模板创建 ML 模型。有关更多信息，请参阅 the section called “CloudFormation 模板集成”。

有关语义搜索的完整文档，包括使用语义搜索的 step-by-step指南，请参阅开源文档中的 S emantic sear ch。 OpenSearch

Amazon S OpenSearch ervice 中的并发分段搜索

从 OpenSearch 版本 2.17 开始，并发区段搜索使用新设置来控制并发搜索行为。

- 在 2.17 版本中创建的新域名在 2xl 或更高版本的节点上，默认并发区段搜索设置为 a uto 模式。
- 升级到 2.17 的现有域将默认并发分段搜索设置为 a uto，具体取决于所有 2xl 或以上节点的实例类型，以及过去 1 周内集群的总 CPU 利用率是否低于 45%。
- 有关更多信息，请参阅并发区段搜索版本 2.17。

从 OpenSearch 版本 2.13 开始，您可以使用并发分段搜索功能，以帮助在查询阶段并行搜索分段。有关并发分段搜索的完整文档，请参阅开源 OpenSearch 文档中的 C oncur rent search。有关与并发区段搜索相关的 Amazon CloudWatch 指标的信息，请参阅实例指标和UltraWarm 指标。

使用 Amazon S OpenSearch ervice 的并发分段搜索功能时，将有一些额外的限制：

- 您无法在 S OpenSearch ervice 中启用索引级别的并发分段搜索。
- 默认情况下，S OpenSearch ervice 使用 2 个分片，并且有最大分片数量限制。

Amazon Ser OpenSearch vice 生成自然语言查询

Amazon Ser OpenSearch vice 中的自然语言查询生成功能让您能够通过自然语言来查询操作和安全日志数据。 OpenSearch 是一个高度可扩展和高性能的日志分析和搜索引擎，因此是探索日志数据的理想选择，现在您还可以使用自然语言来探索这些日志。此功能让您可以轻松识别问题，无需依赖 OpenSearch 管道处理语言（PPL），也不必在构建查询时查找数据定义。您可以在使用版本 2.13 及更高版本的 OpenSearch 服务域上使用自然语言查询生成功能。您必须已经启用精细访问控制。

此功能是使用OpenSearch 助手工具包构建的。如果要创建连接到大语言模型的类似功能，则可以使用该工具包来配置自己的代理和工具。

先决条件

在使用自然语言查询生成功能之前，您的域必须满足以下条件：

- 版本 2.13 或更高版本。
- 服务软件 R20240520-P4 或更高版本。
- 已启用精细访问控制。有关更多信息，请参阅 the section called “启用精细访问控制”。

入门

默认情况下，所有使用版本 2.13 及更高版本创建并且启用了精细访问控制的域都启用了自然语言查询生成功能。

对于其他域，请选择 “启用自然语言查询生成” 和 “Amazon Q 开发者功能” 将其启用。

启用后，导航到 OpenSearch 控制面板中的 “日志” 页面。选择事件资源管理器，然后使用查询助手提问。

配置 权限

如果在事先存在的 OpenSearch 服务域上启用自然语言查询生成功能，可能并未在该域上定义 query_assistant_acces s 角色。必须将非管理员用户映射到此角色，才能使用精细访问控制管理域上的温索引。要手动创建 query_assistant_access 角色，请执行以下步骤：

1. 在 “ OpenSearch 控制面板” 中，转至 “安全”，然后选择 “角色”。
2. 选择创建角色并配置以下集群权限：
   - `cluster:admin/opensearch/ml/config/get`
   - `cluster:admin/opensearch/ml/execute`
   - `cluster:admin/opensearch/ml/predict`
   - `cluster:admin/opensearch/ppl`
3. 将该角色命名为 query_assistant_acces。
4. 选择创建角色。query_assistant_access 角色现已可用。

> **Note**
>
> 对于要使用自然语言问题的索引，您还必须拥有 `indices:admin/mappings/get` 和 `read` 索引权限。

配置自动化

Flow Framework 是一个 OpenSearch 插件，为查询生成和对话聊天等用例提供了一种自动完成 OpenSearch 配置的方法。由于该插件会跟踪启用了自然语言查询生成功能的资源，因此流框架索引会为使用查询助手的每个域存储一个模板。

Flow Framework 让能够从一组预定义的模板中进行选择，也可以为机器学习连接器、工具、代理以及 OpenSearch 作为生成式模型的后端进行准备的其他组件创建自己的自动化。

在 Amazon OpenSearch 服务中使用 OpenSearch 控制面板

OpenSearch 仪表板是一款开源可视化工具，专为与之配合使用而设计 OpenSearch。Amazon OpenSearch 服务为每个 OpenSearch 服务域提供了控制面板的安装。仪表板在域中的热门数据节点上运行。

OpenSearch 仪表板是一种可视化工具，用于浏览和分析单个 OpenSearch 域中的数据。相比之下，集中式 OpenSearch 用户界面（也称为 OpenSearch 应用程序）是一个基于云的用户界面，可连接到多个 OpenSearch域、 OpenSearch 无服务器集合和 Amazon 数据源。它包括用于特定用例（例如可观察性和安全分析）的工作空间，并提供跨数据集的统一体验。虽然仪表板与各个域绑定，但集中式用户界面支持跨域数据集成和分析。有关更多信息，请参阅 OpenSearch 用户界面。

您可以在 OpenSearch 服务控制台的域名控制 OpenSearch 面板上找到指向仪表板的链接。对于正在运行的域名 OpenSearch，URL 为`domain-endpoint`/_dashboards/。对于运行旧版 Elasticsearch 的域，URL 为 `domain-endpoint`/_plugin/kibana。

使用此默认仪表板安装进行查询的超时时间为 300 秒。

> **Note**
>
> 本文档讨论了 Amazon OpenSearch 服务背景下的 OpenSearch 控制面板，包括连接该服务的不同方式。有关全面的文档，包括入门指南、仪表板创建说明、仪表板管理和仪表板查询语言 (DQL)，请参阅开源 OpenSearch 文档中的OpenSearch 仪表板。

控制对仪表板的访问权限

仪表板本身不支持 IAM 用户和角色，但 OpenSearch 服务提供了多种控制仪表板访问权限的解决方案：

- 启用面向控制面板的 SAML 身份验证。
- 将访问权限的精细控制和 HTTP 基本身份验证结合使用。
- 配置控制面板的 Cognito 身份验证。
- 对于公有访问阈，配置一个基于 IP 的访问策略（使用或不使用代理服务器）。
- 对于 VPC 访问阈，配置一个开放访问策略（使用或不使用代理服务器）并使用安全组来控制访问权限。要了解更多信息，请参阅the section called “关于 VPC 域的访问策略”。

使用代理从仪表板访问 OpenSearch 服务

> **Note**
>
> 仅当域使用公有访问权限并且您不想使用 Cognito 身份验证时，此过程才适用。请参阅the section called “控制对仪表板的访问权限 ”。

由于仪表板是一个 JavaScript 应用程序，因此请求来自用户的 IP 地址。基于 IP 的访问控制可能是不切实际的，这是因为，为了让每个用户能够访问 Kibana，需要加入白名单的 IP 地址绝对数量太巨大。一种解决方法是在仪表板和 OpenSearch 服务之间放置代理服务器。然后，您可以添加基于 IP 的访问策略，仅允许来自一个 IP 地址（即代理服务器）的请求。下图演示了此配置。

1. 这是您的 OpenSearch 服务域。IAM 提供对此域的授权访问权限。此外，基于 IP 的访问策略提供对代理服务器的访问权限。
2. 这是在 Amazon EC2 实例上运行的代理服务器。
3. 其他应用程序可以使用签名版本 4 签名过程向 OpenSearch 服务发送经过身份验证的请求。
4. 仪表板客户端通过代理连接到您的 OpenSearch 服务域。

要启用这种配置，需要在基于资源的策略中指定角色和 IP 地址。下面是示例策略：

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Resource":"arn:aws:es:us-west-2:111111111111:domain/my-domain/*",
         "Principal":{
            "AWS":"arn:aws:iam::111111111111:role/allowedrole1"
         },
         "Action":[
            "es:ESHttpGet"
         ],
         "Effect":"Allow"
      },
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":"*"
         },
         "Action":"es:*",
         "Condition":{
            "IpAddress":{
               "aws:SourceIp":[
                  "203.0.113.0/24",
                  "2001:DB8:1234:5678::/64"
               ]
            }
         },
         "Resource":"arn:aws:es:us-west-2:111111111111:domain/my-domain/*"
      }
   ]
}
```

我们建议您为运行代理服务器的 EC2 实例配置弹性 IP 地址。这样，如果有必要替换实例，仍然可以用相同的公有 IP 地址连接到实例。要了解更多信息，请参阅 Amazon EC2 用户指南中的弹性 IP 地址。

如果使用代理服务器和 Cognito 身份验证，则可能需要添加控制面板和 Amazon Cognito 的设置以避免 `redirect_mismatch` 错误。请参阅以下 `nginx.conf` 示例：

```
server {
```

```
    listen 443;
    server_name $host;
    rewrite ^/$ https://$host/_plugin/_dashboards redirect;

    ssl_certificate           /etc/nginx/cert.crt;
    ssl_certificate_key       /etc/nginx/cert.key;

    ssl on;
    ssl_session_cache  builtin:1000  shared:SSL:10m;
    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
    ssl_prefer_server_ciphers on;

    location /_plugin/_dashboards {
        # Forward requests to Dashboards
        proxy_pass https://$dashboards_host/_plugin/_dashboards;

        # Handle redirects to Cognito
        proxy_redirect https://$cognito_host https://$host;

        # Update cookie domain and path
        proxy_cookie_domain $dashboards_host $host;
        proxy_cookie_path / /_plugin/_dashboards/;

        # Response buffer settings
        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;
    }

    location ~ \/(log|sign|fav|forgot|change|saml|oauth2) {
        # Forward requests to Cognito
        proxy_pass https://$cognito_host;

        # Handle redirects to Dashboards
        proxy_redirect https://$dashboards_host https://$host;

        # Update cookie domain
        proxy_cookie_domain $cognito_host $host;
    }
}
```

> **Note**
>
> （可选）如果您选择配置专用的协调器节点，它将自动开始托管 OpenSearch 控制面板。因此可以提高 CPU 和内存等数据节点资源的可用性。由于数据节点资源的可用性提高，这将有助于提高域的整体韧性。

将仪表板配置为使用 WMS 地图服务器

默认安装的 OpenSearch 服务仪表板包括地图服务，但印度和中国地区的域名除外。地图服务最多支持 10 个缩放级别。

无论您的区域如何，您都可以将 Kibana 配置为使用不同的 Web 地图服务 (WMS) 服务器来提供坐标地图可视化。区域地图可视化只支持默认地图服务。

将控制面板配置为使用 WMS 地图服务器：

1. 打开控制面板。
2. 选择堆栈管理。
3. 选择 Advanced Settings (高级设置)。
4. 定位可视化:TileMap:. WMSdefaults
5. 将 `enabled` 更改为 `true` 并将 `url` 更改为有效 WMS 地图服务器的 URL：

```
{
  "enabled": true,
  "url": "wms-server-url",
  "options": {
    "format": "image/png",
    "transparent": true
  }
}
```

6. 选择保存更改。

要将新的默认值应用于可视化，您可能需要重新加载控制面板。如果已保存可视化，请在打开可视化后，选择 Options (选项)。验证是否已启用 WMS map server (WMS 地图服务器) 并且 WMS URL 包含首选地图服务器，然后选择 Apply changes (应用更改)。

> **Note**
>
> 地图服务通常具有许可费用或限制。您负责考虑有关指定的任何地图服务器的所有此类事项。您可能会发现来自美国地质调查局的地图服务对测试非常有用。

将本地仪表板服务器连接到 OpenSearch 服务

如果您已经投入了大量时间来配置自己的仪表板实例，则可以使用它来代替 S OpenSearch ervice 提供的默认 Dashboards 实例（或补充）。以下过程适用于将精细访问控制和开放访问策略结合使用的域。

将本地仪表板服务器连接到 OpenSearch 服务

1. 在您的 OpenSearch 服务域上，创建具有相应权限的用户：
   a. 在控制面板中，转到安全、内部用户，然后选择创建内部用户。
   b. 提供用户名和密码，然后选择创建。
   c. 转到角色，然后选择一个角色。
   d. 选择映射的用户，然后选择管理映射。
   e. 在用户中，添加您的用户名，然后选择映射。
2. 在自行管理的 Dashboards OSS 安装中下载并安装相应版本 OpenSearch 的安全插件。
3. 在您的本地 Dashboards 服务器上，打开`config/opensearch_dashboards.yml`文件并使用您之前创建的用户名和密码添加您的 OpenSearch 服务端点：

   ```
   opensearch.hosts: ['https://domain-endpoint']
   opensearch.username: 'username'
   opensearch.password: 'password'
   ```

   您可以使用以下示例 `opensearch_dashboards.yml` 文件：

   ```
   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]
```

要查看您的 OpenSearch 服务索引，请启动本地仪表板服务器，转到开发工具并运行以下命令：

```
GET _cat/indices
```

在仪表板中管理索引

OpenSearch 服务域上安装的仪表板提供了一个有用的用户界面，用于管理域中不同存储层中的索引。从 “控制面板” 主菜单中选择 “索引管理” UltraWarm，查看热存储和冷存储中的所有索引，以及由索引状态管理 (ISM) 策略管理的索引。使用索引管理可以在热存储和冷存储之间移动索引，并监视三个层之间的迁移。

请注意，除非启用了和 UltraWarm /或冷存储，否则您不会看到热索引、温索引和冷索引选项。

其他功能

在每个 OpenSearch 服务域上安装的默认仪表板还有一些其他功能：

- 各种OpenSearch插件的用户界面
- 租户
- 报告

  使用报告菜单可从“发现”页面生成按需 CSV 报告，以及仪表板或可视化的 PDF 或 PNG 报告。CSV 报告的行数限制为 10,000。
- 甘特图
- 笔记本

在 Amazon OpenSearch 服务中使用 OpenSearch 用户界面

OpenSearch UI（用户界面）是 Amazon S OpenSearch ervice 的现代化运营分析体验，它为您提供了一个统一的视图，便于您与多个来源的数据进行交互。与 OpenSearch 仪表板不同，仪表板仅适用于托管它的一个域或集合，而 OpenSearch 用户界面托管在 Amazon Web Services 云。这使 OpenSearch 用户界面能够在集群升级期间实现高可用性并保持正常运行，并且可以与多个数据源进行本机连接。有关 OpenSearch 仪表板的信息，请参阅 OpenSearch 仪表板。

以下是 OpenSearch UI 的主要功能：

- 支持多个数据源 — OpenSearch 用户界面可以连接多个数据源以创建全面视图。这包括 OpenSearch 域名和无服务器集合，以及集成 Amazon 数据源，例如亚马逊 CloudWatch、亚马逊安全湖和亚马逊简单存储服务 (Amazon S3) Simple Service。
- 升级期间零停机时间 — OpenSearch UI 托管在 Amazon Web Services 云。这意味着它 OpenSearch 仍然可以运行，并且可以在升级过程中从集群中检索数据。
- 工作空间 — 精心策划的团队协作空间，用于各种工作流程，例如可观察性、安全分析和搜索。您可以定义隐私设置并管理工作空间中协作者的权限。
- 单点登录 — OpenSearch UI 与 Amazon IAM Identity Center SAML 配合使用，通过 Amazon Identity and Access Management (IAM) 联合身份验证与您的身份提供商集成，为您的最终用户创建单点登录体验。
- Genai 支持的分析 — OpenSearch 用户界面支持自然语言查询生成，可帮助您生成正确的查询供您分析。 OpenSearch UI 还与 Amazon Q Developer 合作，提供 Amazon Q 聊天功能，帮助生成可视化效果、警报摘要、见解和推荐的异常检测器。
- 支持@@ 多种查询语言 — OpenSearch 用户界面支持管道处理语言 (PPL)、SQL、Lucene 和仪表板查询语言 (DQL)。
- 跨区域和跨账户支持 — OpenSearch 用户界面可以利用跨集群搜索功能连接不同账户和不同区域中的 OpenSearch 域名，以进行汇总分析和可视化。

要开始创建您的第一个 OpenSearch 用户界面，请按照中的说明进行操作the section called “入门”。

有关为 OpenSearch UI 发布的最新功能的信息，请参阅the section called “发行历史记录”。

主题

- 亚马逊 OpenSearch 服务用户界面发布历史记录
- 开始使用 Amazon OpenSearch 服务中的 OpenSearch 用户界面

- 使用启用 SAML 联合 Amazon Identity and Access Management
- 管理数据源关联和虚拟私有云访问权限
- 使用亚马逊 OpenSearch 服务工作空间
- 通过跨集群搜索实现跨区域和跨账户数据访问
- 管理从 VPC 终端节点对 OpenSearch UI 的访问

亚马逊 OpenSearch 服务用户界面发布历史记录

下表列出了所有版本的 Amazon S OpenSearch ervice OpenSearch 用户界面支持以及每个版本中包含的功能和增强功能。

更改	发行日期	描述
新特征	2025-04-16	OpenSearch 用户界面现在支持跨集群搜索。这使您可以将 OpenSearch 用户界面合二为一 Amazon Web Services 区域来访问不同区域的集群。这是通过将其配置为与同一区域中的集群连接的远程集群来完成的。有关更多信息，请参阅 the section called “通过跨集群搜索实现跨区域和跨账户数据访问”。
新特征	2025-03-31	Amazon Q Developer 现已在亚马逊 OpenSearch 服务中正式推出。有关信息，请参阅Amazon Q 支持。
新特征	2025-02-05	通过 (IAM) 联合身份验证的安全断言标记语言 2.0 (SAML) Amazon Identity and Access Management (SAML) 现在可以与 UI 配合使用。 OpenSearch 这样，您就可以为您的最终用户创建由身份提供商启动的单点登录 (SSO) 体验。有关更多信息，请参阅 the section called “通过 IAM 启用 SAML 联合”。
全新集成	2024-12-01	与 Amazon 的零 ETL 集成 CloudWatch 简化了日志数据分析和可视化，从而降低了技术开销和运营成本。有关更多信息，请参阅Amazon 新闻博客上的 Amazon CloudWatch 和 Amazon 新 OpenSearch 服务推出集成分析体验。
全新集成	2024-12-01	与 Amazon Security Lake 的零 ETL 集成使组织能够高效地搜索、分析其安全数据并从中获得切实可行的见解。有关更多信

更改	发行日期	描述
		息，请参阅Amazon 新闻博客上的 “介绍亚马逊 OpenSearch 服务和 Amazon Security Lake 集成以简化安全分析”。
初始版本	2024-11-07	OpenSearch UI 的首次公开发布。有关更多信息，请参阅 Amazon S OpenSearch ervice 在Amazon 大数据博客上发布下一代 OpenSearch 用户界面。

开始使用 Amazon OpenSearch 服务中的 OpenSearch 用户界面

在 Amazon OpenSearch 服务中，应用程序是 OpenSearch用户界面 (OpenSearch UI) 的实例。每个应用程序可以与多个数据源关联，一个源可以与多个应用程序关联。您可以使用不同的支持的身份验证选项为不同的管理员创建多个应用程序。

使用本主题中的信息来指导您完成使用 Amazon Web Services Management Console 或创建 OpenSearch UI 应用程序的过程 Amazon CLI。

主题

- 创建亚马逊 OpenSearch 服务应用程序所需的权限
- 创建 OpenSearch UI 应用程序
- 管理应用程序管理员

创建亚马逊 OpenSearch 服务应用程序所需的权限

在创建应用程序之前，请确认您已被授予执行该任务的必要权限。如有必要，请联系账户管理员寻求帮助。

一般权限

要在 S OpenSearch ervice 中使用应用程序，您需要具有以下策略中显示的权限。这些权限用于以下用途：

- 创建和管理应用程序需要这五个`es:*Application`权限。
- 在应用程序中添加、列出和删除标签需要这三种`es:*Tags`权限。
- 关联数据源需要`es:DescribeDomain`和`es:GetDirectQueryDataSource`权限。`aoss:BatchGetCollection`

- 访问数据源需要`aoss:APIAccessAlles:ESHttp*`、和 4 个`opensearch:*DirectQuery*`权限。
- 向 Amazon S OpenSearch ervice `iam:CreateServiceLinkedRole` 提供了在您的账户中创建服务相关角色 (SLR) 的权限。使用此角色可以让 OpenSearch UI 应用程序在您的账户中发布 Amazon CloudWatch 指标。有关更多信息，请参阅 the section called “权限”主题中的 使用服务相关角色创建 VPC 域和直接查询数据源。

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"
        }
    ]
}
```

创建使用 IAM 身份中心身份验证的应用程序的权限（可选）

默认情况下，控制面板应用程序使用 Amazon Identity and Access Management (IAM) 进行身份验证，以管理 Amazon 资源用户的权限。但是，您可以选择使用 IAM Identity Center 提供单点登录体验，该中心允许您使用现有的身份提供商登录 OpenSearch 界面应用程序。在这种情况下，您将在本主题后面的步骤中选择 “使用 IAM 身份中心进行身份验证” 选项，然后向 IAM Identity Center 用户授予访问 OpenSearch UI 应用程序所需的权限。）

要创建使用 IAM Identity Center 身份验证的应用程序，您需要以下权限。将 *placeholder values* 替换为您自己的信息。如有必要，请联系账户管理员寻求帮助。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IDC_Permissions",
            "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": "SLR_Permission",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService"
        },
        {
            "Sid": "PassRole_Permission",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::account-id}:role/iam-role-for-identity-
center"
        }
    ]
}
```

创建 OpenSearch UI 应用程序

使用以下过程之一创建指定应用程序名称、身份验证方法和管理员的应用程序。

主题

- 在控制台中创建使用 IAM 身份验证的 OpenSearch UI 应用程序
- 在控制台中创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch UI 应用程序
- 创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch UI 应用程序 Amazon CLI

在控制台中创建使用 IAM 身份验证的 OpenSearch UI 应用程序

在控制台中创建使用 IAM 身份验证的 OpenSearch UI 应用程序

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择 OpenSearch UI（控制面板）。
3. 选择创建应用程序。
4. 在应用程序名称中，输入应用程序的名称。
5. 请勿选中 “使用 IAM 身份中心进行身份验证” 复选框。有关通过身份验证创建应用程序的信息 Amazon IAM Identity Center，请参阅本主题the section called “在控制台中创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch UI 应用程序”后面的部分。
6. （可选）系统会自动将您添加为正在创建的应用程序的管理员。在OpenSearch 应用程序管理员管理区域中，您可以向其他用户授予管理员权限。

   > **Note**
   >
   > OpenSearch UI 应用程序管理员角色授予编辑和删除 OpenSearch UI 应用程序的权限。应用程序管理员还可以在 OpenSearch 用户界面应用程序中创建、编辑和删除工作区。

   要向其他用户授予管理员权限，请选择以下选项之一：

   - 向特定用户授予管理员权限 — 在OpenSearch应用程序管理员字段的属性弹出列表中，选择 IAM 用户或

     Amazon IAM Identity Center 用户，然后选择要向其授予管理员权限的单个用户。
   - 向所有用户授予管理员权限-您的组织或账户中的所有用户都被授予管理员权限。

7. （可选）在标签区域中，将一个或多个标签密钥 name/value 对应用于应用程序。

   标签是您分配给资源的可选元数据。标签可让您按不同的方式（如用途、拥有者或环境）对资源进行分类。

8. 选择创建。

在控制台中创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch UI 应用程序

要创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch UI 应用程序，您必须拥有本主题前面所述的 IAM 权限the section called “创建使用 IAM 身份中心身份验证的应用程序的权限（可选）”。

在控制台中创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch UI 应用程序

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择 OpenSearch UI（控制面板）。
3. 选择创建应用程序。
4. 在应用程序名称中，输入应用程序的名称。
5. （可选）要为您的组织或账户启用单点登录，请执行以下操作：

   a. 选中 “使用 IAM 身份中心进行身份验证” 复选框，如下图所示：

   b. 请执行以下操作之一：

      - 在 Ident ity Center 的 IAM 角色应用程序列表中，选择一个现有 IAM 角色，该角色为 IAM Identity Center 提供访问 OpenSearch 用户界面和相关数据源所需的权限。有关该角色必须拥有的权限，请参阅下一个项目符号中的策略。
      - 创建具有所需权限的新角色。使用 IAM 用户指南中的以下过程，其中包含创建新角色的指定选项以及必要的权限策略和信任策略。

- 过程：创建 IAM 策略（控制台）

  按照此过程中的步骤进行操作时，请将以下策略粘贴到策略编辑器 JSON 字段中：

  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": "*"
        }
    ]
}
```

- 步骤：使用自定义信任策略创建角色

按照此过程中的步骤进行操作时，请将 “自定义信任策略” 框中的占位符 JSON 替换为以下内容：

> **Tip**
>
> 如果您要向现有角色添加信任策略，请在该角色的 “信任关系” 选项卡上添加该策略。

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service":
 "application.opensearchservice.amazonaws.com"
            },
            "Action": [
                "sts:AssumeRole",
                "sts:SetContext"
            ],
            "Condition": {
                "ForAllValues:ArnEquals": {
                    "sts:RequestContextProviders":
 "arn:aws:iam::aws:contextProvider/IdentityCenter"
                }
            }
        }
    ]
}
```

c. 如果您的组织或账户中已经创建了 IAM 身份中心实例，则控制台会报告 Amazon OpenSearch Dashboards 已连接到 IAM Identity Center 的组织实例，如下图所示。

如果您的组织或账户中还没有 IAM Identity Center，则您或具有必要权限的管理员可以创建组织实例或账户实例。将 A mazon OpenSearch 控制面板连接到 IAM 身份中心区域为两者提供了选项，如下图所示：

在这种情况下，您可以在 IAM Identity Center 中创建账户实例进行测试，或者请求管理员在 IAM Identity Center 中创建组织实例。有关更多信息，请参阅《Amazon IAM Identity Center 用户指南》中的以下主题：

> **Note**
>
> 目前，OpenSearch 用户界面应用程序只能在与您的 IAM Identity Center 组织实例相同 Amazon Web Services 区域 的情况下创建。有关在创建应用程序后访问该区域的数据源的信息，请参阅the section called “通过跨集群搜索实现跨区域和跨账户数据访问”。

- IAM 身份中心的组织实例
- IAM Identity Center 的账户实例
- 启用 Amazon IAM Identity Center

6. （可选）系统会自动将您添加为正在创建的应用程序的管理员。在OpenSearch 应用程序管理员管理区域中，您可以向其他用户授予管理员权限，如下图所示：

> **Note**
>
> OpenSearch UI 应用程序管理员角色授予编辑和删除 OpenSearch UI 应用程序的权限。应用程序管理员还可以在 OpenSearch 用户界面应用程序中创建、编辑和删除工作区。

要向其他用户授予管理员权限，请选择以下选项之一：

- 向特定用户授予管理员权限 — 在OpenSearch应用程序管理员字段的属性弹出列表中，选择 IAM 用户或

  Amazon IAM Identity Center 用户，然后选择要向其授予管理员权限的单个用户。
- 向所有用户授予管理员权限-您的组织或账户中的所有用户都被授予管理员权限。

7. （可选）在标签区域中，将一个或多个标签密钥 name/value 对应用于应用程序。

   标签是您分配给资源的可选元数据。标签可让您按不同的方式（如用途、拥有者或环境）对资源进行分类。
8. 选择创建。

创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch UI 应用程序 Amazon CLI

要使用创建使用 Amazon IAM Identity Center 身份验证的 OpenSearch 用户界面应用程序，请使用带有以下 Amazon CLI选项的 create-ap plication 命令：

- `--name`— 应用程序的名称。
- `--iam-identity-center-options`—（可选）用于身份验证和访问控制的 IAM 身份中心实例和 IAM 角色。 OpenSearch

将 *placeholder values* 替换为您自己的信息。

```
aws opensearch create-application \
    --name application-name \
    --iam-identity-center-options "
          {
          \"enabled\":true,
          \"iamIdentityCenterInstanceArn\":\"arn:aws:sso:::instance/sso-instance\",
          \"iamRoleForIdentityCenterApplicationArn\":\"arn:aws:iam::account-
id:role/role-name\"
          }
    "
```

管理应用程序管理员

OpenSearch 用户界面应用程序管理员是具有编辑和删除 OpenSearch 用户界面应用程序权限的已定义角色。

默认情况下，作为 OpenSearch UI 应用程序的创建者，您是 OpenSearch UI 应用程序的第一位管理员。

使用控制台管理 OpenSearch UI 管理员

您可以在应用程序创建工作流程中为 OpenSearch 用户界面应用程序添加其他管理员 Amazon Web Services Management Console，也可以在应用程序创建后的 “编辑” 页面中添加其他管理员。

OpenSearch UI 应用程序管理员角色授予编辑和删除 OpenSearch UI 应用程序的权限。应用程序管理员还可以在 OpenSearch 用户界面应用程序中创建、编辑和删除工作区。

在应用程序详情页面上，您可以搜索 IAM 委托人的亚马逊资源名称 (ARN) 或搜索 IAM Identity Center 用户的姓名。

使用控制台管理 OpenSearch UI 管理员

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择 OpenSearch UI（控制面板）。
3. 在OpenSearch 应用程序区域中，选择现有应用程序的名称。
4. 选择 Edit (编辑)。
5. 要向其他用户授予管理员权限，请选择以下选项之一：
   - 向特定用户授予管理员权限 — 在OpenSearch应用程序管理员字段的属性弹出列表中，选择 IAM 用户或

     Amazon IAM Identity Center 用户，然后选择要向其授予管理员权限的单个用户。
   - 向所有用户授予管理员权限-您的组织或账户中的所有用户都被授予管理员权限。
6. 选择更新。

您可以移除其他管理员，但每个 OpenSearch UI 应用程序必须至少保留一名管理员。

使用管理 OpenSearch用户界面管理员 Amazon CLI

您可以使用创建和更新 OpenSearch UI 应用程序管理员 Amazon CLI。

使用创建 OpenSearch 用户界面管理员 Amazon CLI

以下是在创建 OpenSearch UI 应用程序时将 IAM 委托人和 IAM 身份中心用户添加为管理员的示例。

示例 1：创建将 IAM 用户添加为管理员的 OpenSearch UI 应用程序

运行以下命令创建将 IAM 用户添加为管理员的 OpenSearch UI 应用程序。将 *placeholder values* 替换为您自己的信息。

```
aws opensearch create-application \
    --name application-name \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"arn:aws:iam::account-id:user/user-id\"
        }
    "
```

示例 2：创建启用 IAM Identity Center 的 OpenSearch UI 应用程序，并将 IAM 身份中心用户 ID 添加为 OpenSearch UI 应用程序管理员

运行以下命令创建启用 IAM Identity Center 并以 OpenSearch UI 应用程序管理员身份添加 IAM Identity Center 用户 ID 的 OpenSearch UI 应用程序。将 *placeholder values* 替换为您自己的信息。

`key`指定要设置的配置项目，例如 OpenSearch UI 应用程序的管理员角色。有效值包括 `opensearchDashboards.dashboardAdmin.users` 和 `opensearchDashboards.dashboardAdmin.groups`。

*xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*表示分配给密钥的值，例如 IAM 用户的亚马逊资源名称 (ARN)。

```
aws opensearch create-application \
    --name myapplication \
    --iam-identity-center-options "
        {
        \"enabled\":true,
        \"iamIdentityCenterInstanceArn\":\"arn:aws:sso:::instance/ssoins-instance-id\",
        \"iamRoleForIdentityCenterApplicationArn\":\"arn:aws:iam::account-id:role/role-
name\"
        }
    " \
```

```
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"
        }
    "
```

使用更新 OpenSearch 用户界面管理员 Amazon CLI

以下是更新分配为现有 OpenSearch应用程序管理员的 IAM 委托人和 IAM Identity Center 用户的示例。

示例 1：将 IAM 用户添加为现有 OpenSearch应用程序的管理员

运行以下命令更新 OpenSearch UI 应用程序以将 IAM 用户添加为管理员。将 *placeholder values* 替换为您自己的信息。

```
aws opensearch update-application \
    --id myapplication \
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"arn:aws:iam::account-id:user/user-id\"
        }
    "
```

示例 2：更新 OpenSearch UI 应用程序以将 IAM Identity Center 用户 ID 添加为 OpenSearch UI 应用程序管理员

运行以下命令更新 OpenSearch UI 应用程序，以将 IAM Identity Center 用户 ID 添加为 OpenSearch UI 应用程序管理员。将 *placeholder values* 替换为您自己的信息。

`key`指定要设置的配置项目，例如 OpenSearch UI 应用程序的管理员角色。有效值包括 `opensearchDashboards.dashboardAdmin.users` 和 `opensearchDashboards.dashboardAdmin.groups`。

*xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*表示分配给密钥的值，例如 IAM 用户的亚马逊资源名称 (ARN)。

```
aws opensearch update-application \
    --id myapplication \
```

```
    --app-configs "
        {
        \"key\":\"opensearchDashboards.dashboardAdmin.users\",
        \"value\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"
        }
    "
```

使用启用 SAML 联合 Amazon Identity and Access Management

OpenSearch 用户界面支持安全断言标记语言 2.0 (SAML)，这是许多身份提供商使用的开放标准。这将启用与 Amazon Identity and Access Management (IAM) 的联合身份验证。有了这种支持，您的账户或组织中的用户可以通过担任 IAM 角色直接访问用户 OpenSearch 界面。您可以为最终用户创建由身份提供商启动 (IdP) 的单点登录体验，他们可以在外部身份提供商中进行身份验证，然后直接路由到您在 UI 中定义的页面。 OpenSearch 您还可以通过将最终用户或群组配置为扮演不同的 IAM 角色来实现精细的访问控制，这些角色具有不同的访问用户 OpenSearch 界面和相关数据源的权限。

本主题介绍 step-by-step如何通过 OpenSearch 用户界面配置 SAML 使用。在这些步骤中，我们以配置 Okta 身份和访问管理应用程序的步骤为例。其他身份提供商（例如 Azure Active Directory 和 Ping）的配置步骤类似。

主题

- 步骤 1：设置身份提供商应用程序 (Okta)
- 第 2 步：为 Okta 设置 Amazon 配置
- 步骤 3：在 IAM 中创建亚马逊 OpenSearch 服务访问策略
- 步骤 4：使用 SAML 验证身份提供商启动的单点登录体验

步骤 1：设置身份提供商应用程序 (Okta)

要使用带有 OpenSearch UI 的 SAML，第一步是设置您的身份提供商。

任务 1：创建 Okta 用户

1. 以具有管理权限的用户身份通过 https://login.okta.com/登录您的 Okta 组织。
2. 在管理员控制台的导航窗格的 “目录” 下，选择 “人员”。
3. 选择 Add person（添加人员）。
4. 在 “名字” 中，输入用户的名字。
5. 在姓氏中，输入用户的姓氏。

6. 在 “用户名” 中，以电子邮件格式输入用户的用户名。
7. 选择我将设置密码并输入密码
8. （可选）如果您不希望用户在首次登录时更改密码，请清除 “用户必须在首次登录时更改密码” 复选框。
9. 选择保存。

任务 2：创建和分配群组

1. 以具有管理权限的用户身份通过 https://login.okta.com/登录您的 Okta 组织。
2. 在管理员控制台的导航窗格目录下，选择群组。
3. 选择添加组。
4. 输入群组名称并选择 “保存”。
5. 选择新创建的群组，然后选择 “分配人员”。
6. 选择加号 (+)，然后选择完成。
7. （可选）重复步骤 1—6 以添加更多群组。

任务 3：创建 Okta 应用程序

1. 以具有管理权限的用户身份通过 https://login.okta.com/登录您的 Okta 组织。
2. 在管理员控制台的导航窗格的 “应用程序” 下，选择 “应用程序”。
3. 选择 Create App Integration（创建应用程序集成）。
4. 选择 SAML 2.0 作为登录方法，然后选择 “下一步”。
5. 输入应用程序集成的名称（例如**OpenSearch_UI**），然后选择 “下一步”。
6. 在应用程序中输入以下值；您无需更改其他值：
   a. 1. 对于单点登录 URL，请输入**https://signin.aws.amazon.com/saml**商业 Amazon 区域或您所在地区的特定网址。
   b. 2. 对于受众 URI（SP 实体 ID），请输入**urn:amazon:webservices**。
   c. 3. 对于 “姓名 ID 格式”，请输入**EmailAddress**。
7. 选择下一步。
8. 选择 “我是 Okta 客户，正在添加内部应用程序”，然后选择 “这是我们创建的内部应用程序”。
9. 选择完成。
10. 选择 “分配”，然后选择 “分配”。

11. 选择 “分配给群组”，然后选择要添加的群组旁边的 “分配”。
12. 选择完成。

任务 4：设置 Okta 高级配置

创建自定义 SAML 应用程序后，请完成以下步骤：

1. 以具有管理权限的用户身份通过 https://login.okta.com/登录您的 Okta 组织。

   在管理员控制台的 “常规” 区域中，选择 “SAML 设置” 下的 “编辑”。
2. 选择下一步。
3. 使用以下格式将 OpenSearch UI 端点的默认中继状态设置为：

   ```
   https://region.console.aws.amazon.com/aos/home?
   region=region#opensearch/applications/application-id/
   redirectToDashboardURL.
   ```

   以下是示例：

   ```
   https://us-east-2.console.aws.amazon.com/aos/home?region=us-
   east-2#opensearch/applications/abc123def4567EXAMPLE/
   redirectToDashboardURL
   ```
4. 在 “属性语句”（可选）下，添加以下属性：

   a. 使用 Role 属性以逗号分隔的格式提供 IAM 角色和身份提供商。在稍后的步骤中，您将在设置 Amazon 配置时使用相同的 IAM 角色和身份提供商。

   b. 将 user.login 设置为。RoleSessionName这用作担任该角色时颁发的临时证书的标识符。

   供参考：

名称	名称格式	格式	示例
https://aws.amazon.com/SAML/Attributes/Role	未指定	arn:aws:iam:: *aws-account-id* :role/role-name,arn:aws:iam	arn:aws:iam::111222333444:role/oktarole,arn:aws:iam::1

名称	名称格式	格式	示例
		:: *aws-account-id* :saml-provider/ *provider-name*	11222333444:saml-provider/oktaidp
https://aws.amazon.com/SAML/Attributes/RoleSessionName	未指定	user.login	user.login

5. 添加属性属性后，选择 “下一步”，然后选择 “完成”。

您的属性的格式应与下图所示的属性相似。默认中继状态值是指在您的账户或组织中的最终用户完成 Okta 的单点登录验证后为他们定义登录页面的 URL。您可以将其设置为 OpenSearch UI 中的任何页面，然后将该 URL 提供给其目标最终用户。

第 2 步：为 Okta 设置 Amazon 配置

完成以下任务来设置 Okta 的 Amazon 配置。

任务 1：收集 Okta 信息

在此步骤中，您需要收集 Okta 信息，以便以后可以在中 Amazon对其进行配置。

1. 以具有管理权限的用户身份通过 https://login.okta.com/登录您的 Okta 组织。
2. 在页面右下角的 “登录” 选项卡上，选择 “查看 SAML 设置说明”。
3. 记下身份提供商单点登录 URL 的值。在连接到任何第三方 SQL 客户端（例如 SQL Workbench/J）时，您可以使用此 URL。
4. 使用区块 4 中的身份提供商元数据，然后将元数据文件保存为.xml 格式（例如，`metadata.xml`）。

任务 2：创建 IAM 提供商

要创建您的 IAM 提供商，请完成以下步骤：

1. 登录 Amazon Web Services Management Console 并打开 IAM 控制台，网址为https://console.aws.amazon.com/iam/。
2. 在导航窗格的访问管理下，选择身份提供商。
3. 选择 Add provider（添加提供程序）。
4. 对于提供商类型，请选择 SAML。
5. 对于提供商名称，请输入一个名称。
6. 对于元数据文档，选择选择文件并上传您之前下载的元数据文件 (.xml)。
7. 选择 Add provider（添加提供程序）。

任务 3：创建 IAM 角色

要创建您的 Amazon Identity and Access Management 角色，请完成以下步骤：

1. 登录 Amazon Web Services Management Console 并打开 IAM 控制台，网址为https://console.aws.amazon.com/iam/。
2. 在导航窗格的访问管理下，选择角色。
3. 选择创建角色。
4. 对于可信实体类型，请选择 SAML 2.0 联合。
5. 对于基于 SAML 2.0 的提供商，请选择您之前创建的身份提供商。
6. 选择 “允许编程和 Amazon Web Services Management Console 访问”。
7. 选择下一步。
8. 在 “权限策略” 列表中，选中之前创建的策略和 “” 的复选框OpenSearchFullAccess。
9. 选择下一步。
10. 在 “查看” 区域的 “角色名称” 中，输入您的角色名称；例如，**`oktarole`**。
11. （可选）在描述中，输入角色用途的简短描述。
12. 选择创建角色。
13. 导航到您刚刚创建的角色，选择 “信任关系” 选项卡，然后选择 “编辑信任策略”。
14. 在 “编辑语句” 窗格的 “为 STS 添加操作” 下，选中相应的复选框TagSession。

15. 选择更新策略。

步骤 3：在 IAM 中创建亚马逊 OpenSearch 服务访问策略

本主题提供有关配置您的 IAM 角色以访问 OpenSearch 服务的信息。我们提供了两个群组`Alice`和的示例`Bob`，以演示如何通过 Okta 为您的用户组实现精细的访问控制。

Sample group: Alice

请求:

```
GET _plugins/_security/api/roles/alice-group
```

结果：

```
{
  "alice-group": {
    "reserved": false,
    "hidden": false,
    "cluster_permissions": [
      "unlimited"
    ],
    "index_permissions": [
      {
        "index_patterns": [
          "alice*"
        ],
        "dls": "",
        "fls": [],
        "masked_fields": [],
        "allowed_actions": [
          "indices_all"
        ]
      }
    ],
    "tenant_permissions": [
      {
        "tenant_patterns": [
          "global_tenant"
        ],
        "allowed_actions": [
          "kibana_all_write"
```

```
        ]
      }
    ],
    "static": false
  }
}
```

Sample group: Bob

请求:

```
GET _plugins/_security/api/roles/bob-group
```

结果：

```
{
  "bob-group": {
    "reserved": false,
    "hidden": false,
    "cluster_permissions": [
      "unlimited"
    ],
    "index_permissions": [
      {
        "index_patterns": [
          "bob*"
        ],
        "dls": "",
        "fls": [],
        "masked_fields": [],
        "allowed_actions": [
          "indices_all"
        ]
      }
    ],
    "tenant_permissions": [
      {
        "tenant_patterns": [
          "global_tenant"
        ],
        "allowed_actions": [
          "kibana_all_write"
        ]
```

```
      }
    ],
    "static": false
  }
}
```

您可以使用后端角色映射将 Amazon S OpenSearch ervice 域角色映射到 IAM 角色，如以下示例所示：

```
{
  "bob-group": {
    "hosts": [],
    "users": [],
    "reserved": false,
    "hidden": false,
    "backend_roles": [
      "arn:aws:iam::111222333444:role/bob-group"
    ],
    "and_backend_roles": []
  },
  "alice-group": {
    "hosts": [],
    "users": [],
    "reserved": false,
    "hidden": false,
    "backend_roles": [
      "arn:aws:iam::111222333444:role/alice-group"
    ],
    "and_backend_roles": []
  }
}
```

步骤 4：使用 SAML 验证身份提供商启动的单点登录体验

打开 “默认中继状态” 的 URL 以打开 Okta 身份验证页面。输入最终用户的凭据。系统会自动将您重定向到 OpenSearch UI。

您可以通过选择导航面板底部的用户图标来检查您当前的凭证，如下图所示：

您还可以通过访问导航面板底部的开发者工具并在控制台中运行查询来验证用户的精细访问控制权限。以下是示例查询。

Example 1: Displays information about the current user

请求:

```
GET _plugins/_security/api/account
```

结果：

```
{
  "user_name": "arn:aws:iam::XXXXXXXXXXXX:role/bob-group",
  "is_reserved": false,
  "is_hidden": false,
  "is_internal_user": false,
  "user_requested_tenant": null,
  "backend_roles": [
    "arn:aws:iam::XXXXXXXXXXXX:role/bob-group"
  ],
  "custom_attribute_names": [],
  "tenants": {
    "global_tenant": true,
    "arn:aws:iam::XXXXXXXXXXXX:role/bob-group": true
  },
  "roles": [
    "bob-group"
  ]
}
```

Example 2: Displays actions permitted for a user

请求:

```
GET bob-test/_search
```

结果：

```
{
  "took": 390,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
```

```
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 1,
    "hits": [
      {
        "_index": "bob-test",
        "_id": "ui01N5UBCIHpjO8Jlvfy",
        "_score": 1,
        "_source": {
          "title": "Your Name",
          "year": "2016"
        }
      }
    ]
  }
}
```

Example 3: Displays actions not permitted for a user

请求:

```
GET alice-test
```

结果：

```
{
  "error": {
    "root_cause": [
      {
        "type": "security_exception",
        "reason": "no permissions for [indices:admin/get]
 and User [name=arn:aws:iam::111222333444:role/bob-group,
 backend_roles=[arn:aws:iam::111222333444:role/bob-group], requestedTenant=null]"
      }
    ],
    "type": "security_exception",
    "reason": "no permissions for [indices:admin/get]
 and User [name=arn:aws:iam::111222333444:role/bob-group,
 backend_roles=[arn:aws:iam::111222333444:role/bob-group], requestedTenant=null]"
  },
  "status": 403
```

```
}
```

管理数据源关联和虚拟私有云访问权限

使用本节中的过程来管理数据源关联并为虚拟私有云 (VPC) 配置任何所需的访问权限。

主题

- 将数据源与 OpenSearch UI 应用程序关联
- 管理 VPC 中域的访问权限
- 在 VPC 中配置对 OpenSearch 无服务器集合的访问权限

将数据源与 OpenSearch UI 应用程序关联

创建 OpenSearch UI 应用程序后，您可以使用控制台或 Amazon CLI 将其与一个或多个数据源相关联。之后，最终用户可以从这些数据源中检索数据以进行搜索、使用仪表板等。

将数据源与 OpenSearch UI 应用程序关联（控制台）

使用控制台将数据源与 OpenSearch UI 应用程序关联

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 选择 OpenSearch UI（仪表板），然后选择 OpenSearch UI 应用程序的名称。
3. 在 “关联的数据源” 区域中，选择 “管理数据源”。
4. 从要与应用程序关联的 OpenSearch 域名和集合中进行选择。

> **Tip**
>
> 如果您找不到要查找的数据源，请联系您的管理员授予您必要的权限。有关更多信息，请参阅 the section called “创建使用 IAM 身份中心身份验证的应用程序的权限（可选）”。

5. 选择 “下一步”，然后选择 “保存”。

将数据源与应用程序关联后，将在应用程序详细信息页面上启用启动应用程序按钮。您可以选择 Launch Appl icati on 来打开 “欢迎使用 OpenSearch” 页面，您可以在其中创建和管理工作区。

有关使用工作空间的信息，请参阅the section called “使用亚马逊 OpenSearch 服务工作空间”。

管理 VPC 中域的访问权限

如果 VPC 中的某个 OpenSearch 域与应用程序关联，则 VPC 管理员必须使用控制台或授权 OpenSearch UI 和 VPC 之间的访问权限 Amazon CLI。

管理 VPC 中域的访问权限（控制台）

使用以下方法配置对 VPC 域的访问权限 Amazon Web Services Management Console：

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择域，然后选择 VPC 域的名称。

   –或者–

   选择创建域，然后配置该域的详细信息。
3. 选择 VPC 终端节点选项卡，然后选择授权委托人。
4. 在 “授权委托人” 对话框中，选择 “授权其他 Amazon 服务的委托人”，然后从列表中选择OpenSearch 应用程序（控制面板）。
5. 选择授权。

管理 VPC 中域的访问权限 (Amazon CLI)

使用 VPC 域授权 Amazon CLI

要使用授权 VPC 域 Amazon CLI，请运行以下命令。将 *placeholder values* 替换为您自己的信息。

```
aws opensearch authorize-vpc-endpoint-access \
 --domain-name domain-name \
 --service application.opensearchservice.amazonaws.com \
 --region region-id
```

使用控制台撤消 VPC 域关联

当不再需要关联时，VPC 域所有者可以使用以下步骤撤消访问权限。

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择域，然后选择 VPC 域的名称。

3. 选择 VPC 终端节点选项卡，然后选择OpenSearch 应用程序（控制面板）行的按钮。
4. 然后选择撤消访问权限。

要撤消 VPC 域关联，请使用 Amazon CLI

要撤消与 OpenSearch UI 应用程序的 VPC 域关联，请运行以下命令。将 *placeholder values* 替换为您自己的信息。

```
aws opensearch revoke-vpc-endpoint-access \
    --domain-name domain-name \
    --service application.opensearchservice.amazonaws.com \
    --region region-id
```

在 VPC 中配置对 OpenSearch 无服务器集合的访问权限

如果 VPC 中的某个 Amazon OpenSearch Serverless 集合与该应用程序关联，VPC 管理员可以通过创建新的网络策略并将其附加到集合中来授权访问。

在 VPC 中配置对 OpenSearch 无服务器集合的访问权限（控制台）

使用控制台配置对 VPC 中 OpenSearch 无服务器集合的访问权限

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航栏中，选择网络策略，选择网络策略的名称，然后选择编辑。

   –或者–

   选择 “创建网络策略”，然后配置策略的详细信息。
3. 在 “访问类型” 区域中，选择 “私有（推荐）”，然后选择 “Amazon 服务私有访问”。
4. 在搜索字段中，选择 “服务”，然后选择`application.opensearchservice.amazonaws.com`。
5. 在 “资源类型” 区域中，选中 “启用 OpenSearch 终端节点访问权限” 复选框。
6. 对于 “搜索馆藏”，或输入特定的前缀词，在搜索字段中选择 “集合名称”，然后输入或选择要与网络策略关联的集合名称。
7. 为新的网络策略选择 “创建”，或为现有网络策略选择 “更新”。

在 VPC 中配置对 OpenSearch 无服务器集合的访问权限 ()Amazon CLI

要配置对 VPC 中 OpenSearch 无服务器集合的访问权限，请使用 Amazon CLI

1. 创建一个类似于以下内容的.json 文件。将 *placeholder values* 替换为您自己的信息。

```
{
    "Description" : "policy-description",
    "Rules": [{
       "ResourceType" : "collection",
        "Resource" : ["collection/collection_name"]
     }],
    "SourceServices" : [
          "application.opensearchservice.amazonaws.com"
      ],
      "AllowFromPublic" : false
}
```

2. 为 VPC 中的集合创建或更新网络策略，使其与 OpenSearch UI 应用程序配合使用。

   Create a network policy

   运行以下命令。将 *placeholder values* 替换为您自己的信息。

```
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
```

   命令返回类似于下文的信息：

```
{
    "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

运行以下命令。将 *placeholder values* 替换为您自己的信息。

```
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
```

命令返回类似于下文的信息：

```
{
    "securityPolicyDetail": {
        "createdDate": ******,
        "lastModifiedDate": ******,
        "name": "network-policy-name",
        "policy": [
            {
                "SourceVPCEs": [],
                "AllowFromPublic": false,
```

```
                "Description": "",
                "Rules": [
                    {
                        "Resource": [
                            "collection/network-policy-name"
                        ],
                        "ResourceType": "collection"
                    }
                ],
                "SourceServices": [
                    "application.opensearchservice.amazonaws.com"
                ]
            }
        ],
        "policyVersion": "******",
        "type": "network"
    }
}
```

使用亚马逊 OpenSearch 服务工作空间

Amazon OpenSearch 服务支持创建多个特定用例的工作空间。每个工作区都为可观察性、安全分析和搜索等热门用例提供了精心策划的体验。Workspace 还支持协作者管理，因此您可以仅与目标合作者共享您的工作空间，并管理每个人的权限。

创建 OpenSearch UI 应用程序工作区

在创建 OpenSearch UI 应用程序并与数据源关联并且为该应用程序配置了用户权限之后，您可以启动该用户 OpenSearch 界面应用程序来创建工作空间。

要开始创建工作区，您可以选择应用程序详细信息页面上的启动应用程序按钮，或者使用 OpenSearch 用户界面应用程序 URL 在新的浏览器窗口中打开 OpenSearch UI 应用程序主页。

OpenSearch UI 应用程序提供了用于创建工作区的选项，并在主页中列出了所有现有的工作区，按用例分类。

有关支持的工作空间类型的更多信息，请参阅the section called “工作空间类型”。

工作空间隐私和协作者

您可以将工作空间的隐私设置定义为所有用户的默认权限级别。您可以在创建工作区时执行此操作，也可以修改现有工作区（在工作区 “协作者” 选项卡上）。有三种隐私选项可供选择：

- 合作者专用-只有您明确添加到工作区的协作者才能访问工作区。您可以为每个合作者定义权限级别。
- 任何人都可以查看 — 任何有权访问 OpenSearch UI 应用程序的人都可以访问工作区并查看其资产，但他们无法在工作区中进行任何更改。
- 任何人都可以编辑 — 任何有权访问 OpenSearch UI 应用程序的人都可以访问工作区、查看其中的资源以及对工作区中的资源进行更改。

在工作空间协作者选项卡上，您可以将 IAM 用户或角色以及 Amazon IAM Identity Center 用户添加为工作空间中的协作者。合作者有三个级别的权限可供选择：

- 只读-协作者只能查看工作区中的资产。如果将工作空间配置为使用 “任何人都可以编辑隐私” 设置，则此设置将被覆盖。
- 读写-协作者可以在工作区中查看和编辑资产。如果工作空间配置为使用 “任何人都可以查看隐私” 设置，则协作者仍然可以进行编辑。

- 管理员-合作者可以更新设置和删除工作空间。合作者还可以更改工作空间的隐私设置和管理合作者。创建工作区的用户将被自动分配为工作空间管理员。

工作空间类型

Amazon S OpenSearch ervice 提供五种工作空间类型，每种类型都有针对不同用例的不同功能：

- Obs ervab ility 工作区旨在通过监控日志、指标和跟踪来了解系统运行状况、性能和可靠性。
- Sec urity Analytics 工作区专为检测和调查系统和数据中的潜在安全威胁和漏洞而设计。
- 搜索工作区专为快速查找和浏览组织数据源中的相关信息而设计。
- Essentials 工作区专为 OpenSearch Serverless 作为数据源而设计，它允许分析数据以得出见解、识别模式和趋势，并快速做出数据驱动的决策。您可以在 E ssen tials 工作区中查找和浏览组织数据源的相关信息。
- Analytics（所有功能）工作区专为多用途用例而设计，支持 OpenSearch 服务用户界面（仪表板）中的所有可用功能。

通过跨集群搜索实现跨区域和跨账户数据访问

在 Amazon OpenSearch Serverless 中使用跨集群搜索，您可以跨多个连接的域执行查询和聚合。

Amazon OpenSearch Serverless 中的跨集群搜索使用源域和目标域的概念。跨集群搜索请求来自源域。目标域可以位于不同的 Amazon Web Services 区域（ Amazon Web Services 账户 或两者兼而有之）中，供源域进行查询。使用跨集群搜索，您可以将源域配置为在同一个账户中与您的 OpenSearch 用户界面关联，然后创建与目标域的连接。因此，即使目标网域位于不同的账户或区域中，您也可以使用 OpenSearch 用户界面处理来自目标域的数据。

您需要为传入和传出亚马逊 OpenSearch 服务的数据支付标准 Amazon 数据传输费用。您无需为 OpenSearch 服务域内的节点之间传输的数据付费。有关数据 “输入” 和 “传出” 费用的更多信息，请参阅 Amazon EC2 按需定价页面上的数据传输。

您可以使用跨集群搜索作为将 OpenSearch 用户界面与不同账户或不同区域中的集群关联的机制。默认情况下，作为加密的一部分，域之间的请求在传输过程中会进行 node-to-node加密。

> **Note**
>
> 开源 OpenSearch 工具还记录了跨集群搜索。请注意，与托管 Amazon OpenSearch Serverless 域相比，开源集群的开源工具设置有很大不同。

最值得注意的是，在 Amazon OpenSearch Serverless 中，您可以使用 Amazon Web Services Management Console 而不是请求来配置跨集群连接。cURL除了精细的访问控制外，托管服务还使用 Amazon Identity and Access Management (IAM) 进行跨集群身份验证。
因此，我们建议使用本主题中的内容来配置域名的跨集群搜索，而不是使用开源 OpenSearch 文档。

使用跨集群搜索时的功能差异

与普通域相比，使用跨集群搜索创建的目标域具有以下功能差异和要求：

- 您无法向远程集群写入或运行PUT命令。您对远程集群的访问权限是只读的。
- 源域和目标域都必须是 OpenSearch 域。您无法连接 Elasticsearch 域或用于用户界面的自行管理的 OpenSearch /Elasticsearch 集群。 OpenSearch
- 一个域最多可以有 20 个与其他域的连接。这包括传出和传入连接。
- 源域的版本必须与目标域相同或更高。 OpenSearch 如果要在两个域之间建立双向连接，则两个域的版本应相同。我们建议在建立连接之前将两个域名都升级到最新版本。如果在设置双向连接后需要更新域，则必须先删除该连接，然后再重新创建连接。
- 您不能对远程集群使用自定义词典或 SQL。
- 您不能 Amazon CloudFormation 使用连接域名。
- 不能在 M3 和可突增（T2 和 T3）实例上使用跨集群搜索。
- 跨集群搜索不适用于 Amazon OpenSearch Serverless 馆藏。

UI 的跨集群搜索先决条件 OpenSearch

在设置包含两个 OpenSearch 域的跨集群搜索之前，请确保您的域满足以下要求：

- 两个域都启用了精细访问控制
- Node-to-node 两个域都启用了加密

主题

- 通过跨集群搜索设置跨区域和跨账户数据访问的访问权限
- 在域之间创建连接
- 通过跨集群搜索测试跨区域和跨账户数据访问的安全设置
- 删除连接

通过跨集群搜索设置跨区域和跨账户数据访问的访问权限

当您向源域发送跨集群搜索请求时，该域会根据其域访问策略评估该请求。跨集群搜索需要精细的访问控制。以下是源域开放访问策略的示例。

JSON

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
          "AWS": [
            "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:111222333444:domain/src-domain/*"
    }
  ]
}
```

> **Note**
>
> 如果路径中包含远程索引，则必须在域 ARN 中对 URI 进行 URL 编码。
> 例如，使用以下 ARN 格式：
> `:arn:aws:es:us-east-1:111222333444:domain/my-domain/local_index,dst%3Aremote_index`
> 请勿使用以下 ARN 格式：
> `arn:aws:es:us-east-1:111222333444:domain/my-domain/local_index,dst:remote_index.`

如果您选择除了精细访问控制之外还使用限制性访问策略，则您的策略必须至少允许访问。`es:ESHttpGet`以下是示例：

JSON

```
{
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111222333444:user/john-doe"
        ]
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:us-east-1:account:domain/my-domain/*"
    }
  ]
}
```

源网域的@@ 精细访问控制会评估请求，以确定请求是使用有效的 IAM 还是 HTTP 基本证书签名的。如果是，则接下来会进行细粒度的访问控制，评估用户是否有权执行搜索和访问数据。

以下是搜索的权限要求：

- 如果请求仅搜索目标域上的数据（例如`dest-alias:dest-index/_search`），则只需要目标域的权限。
- 如果请求在两个域上搜索数据（例如`source-index,dest-alias:dest-index/_search`），则两个域都需要权限。
- 要使用精细的访问控制，除了相关索引的标准读取或搜索权限外，还需要该权限。`indices:admin/shards/search_shards`

源域将请求传递到目标域。目标域根据其域访问策略评估该请求。要支持 OpenSearch UI 中的所有功能，例如为文档编制索引和执行标准搜索，必须设置完全权限。以下是我们推荐的目标域名政策示例：

JSON

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

```
    {
       "Effect": "Allow",
       "Principal": {
         "AWS": [
           "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:us-east-2:111222333444:domain/my-destination-
domain/*"
    },
    {
    "Effect": "Allow",
        "Principal": {
    "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:us-east-2:111222333444:domain/"
    }
  ]
}
```

如果您只想执行基本搜索，则最低策略要求是在不支持通配符的情况下为目标域申请`es:ESCrossClusterGet`权限。例如，在前面的策略中，您可以将域名指定为*/my-destination-domain*，而不是指定*/my-destination-domain/**。

在这种情况下，目标域执行搜索并将结果返回到源域。源域将自己的结果（如果有）与目标域的结果相结合，并将它们返回给您。

在域之间创建连接

从源域到目标域的跨集群搜索连接是单向的。这意味着目标域名（在不同的账户或区域中）无法查询 OpenSearch 用户界面的本地源域名。源域创建与目标域的出站连接。目标域收到来自源域的入站连接请求。

在域之间创建连接

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航栏中，选择域名。
3. 选择要用作源域的域名，然后选择 “连接” 选项卡。
4. 在 “出站连接” 区域中，选择 “请求”。
5. 对于 Connection alias（连接别名），输入您的连接的名称。 OpenSearch 用户界面中使用连接别名来选择目标域。
6. 对于连接模式，请选择直接进行跨集群搜索或复制。
7. 要指定在搜索期间连接应跳过不可用的群集，请选中 “跳过不可用的群集” 复选框。选择此选项可确保无论一个或多个远程集群出现故障，您的跨集群查询都会返回部分结果。
8. 对于目标集群，请在 “连接到此集群” Amazon Web Services 账户和 “连接到另一个集群” 之间进行选择 Amazon Web Services 账户。
9. 对于远程域 ARN，请输入集群的 Amazon 资源名称 (ARN)。域名 ARN 可以位于域名详情页面的 “一般信息” 区域。

   该域必须满足以下要求：

   - ARN 必须采用以下格式。`arn:`*`partition`*`:es:`*`regionaccount-id`*`:`*`type`*`/`*`domain-id`*例如：

     ```
     arn:aws:es:us-east-2:111222333444:domain/my-domain
     ```

   - 必须将该域配置为使用 OpenSearch 版本 1.0（或更高版本）或 Elasticsearch 版本 6.7（或更高版本）。
   - 必须在域上启用细粒度访问控制。
   - 域名必须处于运行状态 OpenSearch。
10. 选择请求。

跨集群搜索首先验证连接请求，以确保满足先决条件。如果域不兼容，则连接请求将进入`Validation failed`状态。

如果成功验证了连接请求，则会将其发送到目标域，并且必须在目标域中获得批准。在获得批准之前，连接将保持`Pending acceptance`状态。当连接请求被目标域接受后，状态将更改为 `Active`，且目标域变为可供查询。

域页面显示了目标域的整体域运行状况和实例运行状况的详细信息。只有域拥有者具备创建、查看、删除和监控域之间的连接的灵活性。

连接建立之后，在所连接域的节点之间流动的任何流量都将进行加密。当您将 VPC 域连接到非 VPC 域并且非 VPC 域是可以接收来自 Internet 的流量的公共终端节点时，域之间的跨集群流量仍处于加密和安全状态。

通过跨集群搜索测试跨区域和跨账户数据访问的安全设置

通过跨集群搜索为跨区域和跨账户数据访问设置访问权限后，我们建议使用Postman用于协作 API 开发的第三方平台来测试设置。

要设置您的安全设置，请使用 Postman

1. 在目标域上，为文档编制索引。以下是请求示例：

```
POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1
{
"Dracula": "Bram Stoker"
}
```

2. 要从源域查询此索引，请在查询内包含目标域的连接别名。您可以在域控制面板的连接选项卡上找到连接别名。以下是示例请求和截断后的响应：

```
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. （可选）您可以创建在一次搜索中包含多个域名的配置。例如，假设您设置了以下内容：

   与之间的连接 `domain-adomain-b`，其连接别名为 `cluster_b`

   与之间的连接 `domain-adomain-c`，其连接别名为 `cluster_c`

   在这种情况下，您的搜索包括内容`domain-adomain-b`、和`domain-c`。以下是请求和响应示例：

   请求

```
GET https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_search
{
  "query": {
    "match": {
      "user": "domino"
    }
  }
}
```

   响应：

```
{
"took": 150,
  "timed_out": false,
  "_shards": {
"total": 3,
    "successful": 3,
    "failed": 0,
    "skipped": 0
  },
  "_clusters": {
"total": 3,
    "successful": 3,
    "skipped": 0
  },
  "hits": {
```

```
"total": 3,
    "max_score": 1,
    "hits": [
      {
"_index": "local_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 1,
        "_source": {
"user": "domino",
          "message": "This is message 1",
          "likes": 0
        }
      },
      {
"_index": "cluster_b:b_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 2,
        "_source": {
"user": "domino",
          "message": "This is message 2",
          "likes": 0
        }
      },
      {
"_index": "cluster_c:c_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 3,
        "_source": {
"user": "domino",
          "message": "This is message 3",
          "likes": 0
        }
      }
    ]
  }
}
```

如果您没有选择在连接设置中跳过不可用集群，搜索的所有目标集群必须可用才能成功运行搜索请求。否则，整个请求将失败——即使只有一个域不可用，也不会返回任何搜索结果。

删除连接

删除连接会停止在目标域上执行任何跨集群搜索操作。

您可以在源域或目标域上执行以下步骤来删除连接。移除连接后，该连接将在 15 天内保持可见状态。`Deleted`

不能删除具有活动跨集群连接的域。要删除域，请首先从该域中删除所有传入和传出连接。这是为了确保在删除域之前考虑到了跨集群域用户。

删除连接

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 在左侧导航栏中，选择域名。
3. 选择要删除的域名，然后选择 “连接” 选项卡。
4. 选择要删除的连接的名称。
5. 选择 “删除”，然后确认删除。

管理从 VPC 终端节点对 OpenSearch UI 的访问

您可以使用在 VPC 和 OpenSearch UI 之间创建私有连接 Amazon PrivateLink。使用此连接，您可以像访问同一 VPC 一样访问 OpenSearch UI 应用程序。这样，您就无需配置互联网网关、NAT 设备、VPN 连接或 Amazon Direct Connect 建立连接。您的 VPC 中的实例不需要公有 IP 地址即可访问 OpenSearch 界面。

要建立此私有连接，请先创建一个由提供支持的接口终端节点 Amazon PrivateLink。将在您为接口终端节点指定的每个子网中自动创建终端节点网络接口。这些是请求者管理的网络接口，用作发往 OpenSearch UI 应用程序的流量的入口点。

在 VPC 和 OpenSearch 用户界面之间创建私有连接

您可以使用 Amazon Web Services Management Console 或创建用于从 VPC 访问 OpenSearch 用户界面的私有连接 Amazon CLI。

在 VPC 和 OpenSearch UI 之间创建私有连接（控制台）

使用控制台在 VPC 和 OpenSearch UI 之间创建私有连接

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。

2. 在左侧导航栏的无服务器下，选择 VPC 终端节点。
3. 选择 Create VPC endpoint（创建 VPC 端点）。
4. 在名称中，输入端点的名称。
5. 对于 VPC，请选择您将从中访问 OpenSearch 用户界面应用程序的 VPC。
6. 对于子网，请选择一个您要从中访问 OpenSearch 用户界面应用程序的子网。

> **Note**
>
> 终端节点的 IP 地址和 DNS 类型取决于子网类型：
>
> - 双栈：如果所有子网都同时具有 IPv4 和 IPv6 地址范围。
> - IPv6: 如果所有子网都 IPv6 只是子网。
> - IPv4: 如果所有子网都有 IPv4 地址范围。

7. 对于安全组，选择一个或多个要与端点网络接口关联的安全组。

> **Note**
>
> 在此步骤中，您将限制您授权进入终端节点的入站流量的端口、协议和来源。确保安全组规则允许使用 VPC 终端节点与 OpenSearch UI 应用程序通信的资源也与终端节点网络接口通信。

8. 8. 选择创建端点。

在 VPC 和 OpenSearch 用户界面之间创建私有连接 (Amazon CLI)

要在 VPC 和 OpenSearch UI 之间创建私有连接，请使用 Amazon CLI

运行以下命令。将 *placeholder values* 替换为您自己的信息。

```
aws opensearchserverless create-vpc-endpoint \
    --region region \
    --endpoint endpoint \
    --name vpc_endpoint_name \
    --vpc-id vpc_id \
    --subnet-ids subnet_ids
```

更新 VPC 终端节点策略以允许访问 OpenSearch UI 应用程序

创建私有连接后，通过指定应用程序 ID 更新 VPC 终端节点策略以允许访问 VPC 终端节点策略中的 OpenSearch UI 应用程序。

有关更新 VPC 终端节点策略的信息，请参阅Amazon PrivateLink 指南中的更新 VPC 终端节点策略。

确保 VPC 终端节点策略包含以下声明。将 *placeholder value* 替换为您自己的信息。

```
{
    "Statement": [{
        "Action": ["opensearch:*"],
        "Effect": "Allow",
        "Principal": "*",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "opensearch:ApplicationId": ["opensearch-ui-application-id"]
            }
        }
    }]
}
```

在 VPC 终端节点策略中撤消对 OpenSearch UI 的访问权限

OpenSearch 用户界面需要 VPC 终端节点策略中的明确权限才能允许用户从 VPC 访问应用程序。如果您不再希望用户从 VPC 访问用户 OpenSearch 界面，则可以在终端节点策略中删除该权限。此后，用户在尝试访问用户 OpenSearch 界面时会遇到`403 forbidden`错误消息。

有关更新 VPC 终端节点策略的信息，请参阅Amazon PrivateLink 指南中的更新 VPC 终端节点策略。

以下是拒绝从 VPC 访问 UI 应用程序的 VPC 终端节点策略示例：

```
{
    "Statement": [{
        "Action": ["opensearch:*"],
        "Effect": "Allow",
        "Principal": "*",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "opensearch:ApplicationId": [""]
```

```
                }
            }
        }]
}
```

在 Amazon OpenSearch 服务中管理索引

将数据添加到 Amazon S OpenSearch ervice 后，您通常需要重新索引该数据、使用索引别名、将索引移到更具成本效益的存储空间或将其全部删除。本章介绍 UltraWarm 存储、冷存储和索引状态管理。有关 OpenSearch 索引的信息 APIs，请参阅OpenSearch 文档。

主题

- UltraWarm Amazon OpenSearch 服务的存储空间
- 用于 Amazon OpenSearch 服务的冷存储
- OpenSearch 针对 Amazon OpenSearch 服务进行了优化的存储
- Amazon S OpenSearch ervice 中的索引状态管理
- 使用索引汇总功能汇总 Amazon S OpenSearch ervice 中的索引
- 转换 Amazon OpenSearch 服务中的索引
- Amazon OpenSearch 服务的跨集群复制
- 使用远程重新索引迁移亚马逊 OpenSearch 服务索引
- 使用数据流管理 Amazon Ser OpenSearch vice 中的时间序列数据

UltraWarm Amazon OpenSearch 服务的存储空间

UltraWarm 通过经济高效的方式在 Amazon S OpenSearch ervice 上存储大量只读数据。标准数据节点使用“热”存储，其形式是连接到每个节点的实例存储或 Amazon EBS 卷。热存储为编制索引和搜索新数据提供尽可能快的性能。

UltraWarm 节点使用 Amazon S3 和高级缓存解决方案来提高性能，而不是使用附加的存储。对于您未主动写入、查询频率较低且不需要同样性能的索引， UltraWarm 可显著降低每 GiB 数据的成本。由于 warm 索引是只读的，除非您将其返回到热存储，所 UltraWarm 以 warm 最适合于不可变数据，如日志。

在中 OpenSearch，这些温索引的行为与任何其他索引一样。您可以使用相同的方法查询它们， APIs 也可以使用它们在 OpenSearch 控制面板中创建可视化。

主题

- 先决条件
- UltraWarm 存储要求和性能注意事项

- UltraWarm 定价
- 启用 UltraWarm
- 将索引迁移到 UltraWarm 存储
- 自动执行迁移
- 迁移调整
- 取消迁移
- 列出热索引和暖索引
- 将温索引返回到热存储
- 从快照恢复温索引
- 暖索引的手动快照
- 将温索引迁移到冷存储
- KNN 索引的最佳实践
- 禁用 UltraWarm

先决条件

UltraWarm 有几个重要的先决条件：

- UltraWarm 需要 OpenSearch 或 Elasticsearch 6.8 或更高版本。
- 要使用温存储，域必须具有专用的主节点。
- 使用带待机功能的多可用区域时，温节点的数量必须是所用可用区数量的整数倍。
- 如果您的域为数据节点使用 T2 实例类型，则无法使用温存储。
- 如果您的索引使用近似 k-NN (`"index.knn":true`)，则您可以将其从2.17及更高版本移到温存储。2.17 之前版本的域名可以升级到 2.17 以使用此功能，但是在 2.x 之前的版本上创建的 KNN 索引无法迁移到。 UltraWarm
- 如果您的域使用精细访问控制，则必须将用户映射到控制 OpenSearch 面板中的`ultrawarm_manager`角色进行 UltraWarm API 调用。

> **Note**
>
> `ultrawarm_manager`角色可能无法在某些预先存在的 OpenSearch 服务域上定义。如果没有在控制面板中看到角色，则需要手动创建它。

配置权限

如果在预先存在的 OpenSearch 服务域 UltraWarm 上启用，`ultrawarm_manager`角色可能无法在域上定义。必须将非管理员用户映射到此角色，才能使用精细访问控制管理域上的温索引。手动创建 `ultrawarm_manager` 角色，请执行下列步骤：

1. 在 “ OpenSearch 控制面板” 中，转至 “安全”，然后选择 “权限”。
2. 选择创建操作组并配置以下组：

组名	权限
`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. 选择角色和创建角色。
4. 将角色命名为 ultrawarm_manager。
5. 对于群集权限，选择 `ultrawarm_cluster` 和 `cluster_monitor`。
6. 对于索引，键入 `*`。
7. 对于索引权限，选择 `ultrawarm_index_read`、`ultrawarm_index_write` 和 `indices_monitor`。
8. 选择创建。
9. 创建角色之后，将其映射到将管理 UltraWarm 索引的任何用户或后端角色。

UltraWarm 存储要求和性能注意事项

如中所述the section called “计算存储要求”，热存储中的数据会产生大量开销：副本、Linux 预留空间和 OpenSearch 服务预留空间。例如，具有一个副本分片的 20 GiB 主分片需要大约 58 GiB 的热存储。

因为它使用 Amazon S3，因此不会 UltraWarm 产生这种开销。计算 UltraWarm 存储要求时，您只需考虑主碎片的大小。S3 中数据的持久性消除了对副本的需要，而 S3 会抽象掉任何操作系统或服务注意事项。同样的 20 GiB 分片需要 20 GiB 的温存储空间。如果您预配置一个 `ultrawarm1.large.search` 实例，则可以将其所有 20 TiB 的最大存储空间用于主分片。有关实例类型的摘要以及每个实例可以解决的最大存储量，请参阅the section called “UltraWarm 存储配额”。

使用 UltraWarm，我们仍然建议最大分片大小为 50 GiB。分配给每种 UltraWarm 实例类型的 CPU 核心数量和 RAM 量可以让您了解他们可以同时搜索的分片数量。请注意，虽然在 S3 中只有主碎片计入 S3 中的 UltraWarm存储，但 OpenSearch 控制面板和`_cat/indices`仍会将 UltraWarm 索引大小报告为所有主碎片和副本碎片的总和。

例如，每个 `ultrawarm1.medium.search` 实例具有两个 CPU 内核，并且可以在 S3 上寻址高达 1.5 TiB 的存储。其中两个实例具有 3 TiB 的存储组合，如果每个分片为 50 GiB，则可以使用大约 62 个分片。如果对集群的请求仅搜索其中的四个分片，则性能可能会非常优良。如果请求很宽泛并且搜索了所有 62 个分片，则四个 CPU 内核可能难以执行该操作。监控`WarmCPUUtilization`和`WarmJVMMemoryPressure`UltraWarm 指标以了解实例如何处理您的工作负载。

如果您的搜索范围广泛或频繁，请考虑将索引留在热存储中。与任何其他 OpenSearch 工作负载一样，要确定是否 UltraWarm 满足您的需求，最重要的步骤是使用真实的数据集执行有代表性的客户端测试。

UltraWarm 定价

使用热存储，您需要为预配置的内容支付费用。有些实例需要附加的 Amazon EBS 卷，而其他实例则包含实例存储。无论该存储空间是空还是满，您都要支付相同的价格。

使用 UltraWarm 存储，您只需按实际用量付费。一个 `ultrawarm1.large.search` 实例可以在 S3 上处理多达 20 TiB 的存储空间，但如果您仅存储 1 TiB 的数据，则只需为 1 TiB 的数据付费。与所有其他节点类型一样，您还需要按小时费率支付费用。 UltraWarm 有关更多信息，请参阅 the section called “定价”。

启用 UltraWarm

控制台是创建使用温存储的域的最简单方法。创建域时，选择启用温数据节点和所需的温节点数量。相同的基本过程适用于现有域，前提是它们满足先决条件。即使域状态从 Processing (正在处理) 变为 A ctiv e (活动) 后， UltraWarm 可能也会在几个小时内无法使用。

使用带待机功能的多可用区域时，温节点的数量必须是可用区数量的整数倍。有关更多信息，请参阅 the section called “带待机功能的多可用区”。

您也可以使用Amazon CLI或配置 API 来启用 UltraWarm，特别是中的`WarmEnabled``WarmCount`、和`WarmType`选项`ClusterConfig`。

> **Note**
>
> 域支持最大数量的温节点。有关更多信息，请参阅 the section called “限额”。

示例 CLI 命令

以下 Amazon CLI 命令创建一个具有三个数据节点、三个专用主节点、六个温节点和启用精细访问控制的域：

```
aws opensearch create-domain \
  --domain-name my-domain \
  --engine-version Opensearch_1.0 \
  --cluster-config
 InstanceCount=3,InstanceType=r6g.large.search,DedicatedMasterEnabled=true,DedicatedMasterType=
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"AWS":["123456789012"]},"Action":
["es:*"],"Resource":"arn:aws:es:us-west-1:123456789012:domain/my-domain/*"}]}' \
  --region us-east-1
```

有关更多信息，请参阅 Amazon CLI 命令参考。

示例配置 API 请求

对配置 API 的以下请求创建一个域，其中包含三个数据节点、三个专用主节点以及六个温节点（启用了精细访问控制并具有限制性访问策略）：

```
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/*\"}]}"
}
```

有关详细信息，请参阅《亚马逊 OpenSearch 服务 API 参考》。

将索引迁移到 UltraWarm 存储

如果您完成了对索引的写入，并且不再需要尽可能快的搜索性能，请将其从热索引转换为 UltraWarm：

```
POST _ultrawarm/migration/my-index/_warm
```

然后检查迁移的状态：

```
GET _ultrawarm/migration/my-index/_status

{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_SHARD_RELOCATION",
    "migration_type": "HOT_TO_WARM",
    "shard_level_status": {
      "running": 0,
      "total": 5,
      "pending": 3,
      "failed": 0,
      "succeeded": 2
    }
  }
}
```

索引运行状况必须为绿色才能执行迁移。如果您快速连续迁移多个索引，您可以获得所有迁移的明文摘要，类似于 _cat API：

```
GET _ultrawarm/migration/_status?v
```

```
index     migration_type state
my-index HOT_TO_WARM    RUNNING_SHARD_RELOCATION
```

OpenSearch Service 一次将一个索引迁移到。 UltraWarm队列中最多可包含 200 个迁移。任何超出限制的请求都将被拒绝。要检查队列中的当前迁移数，请监控 `HotToWarmMigrationQueueSize` 指标。在整个迁移过程中，索引仍然可用，无需停机。

迁移过程具有以下状态：

```
PENDING_INCREMENTAL_SNAPSHOT
RUNNING_INCREMENTAL_SNAPSHOT
FAILED_INCREMENTAL_SNAPSHOT
PENDING_FORCE_MERGE
RUNNING_FORCE_MERGE
FAILED_FORCE_MERGE
PENDING_FULL_SNAPSHOT
RUNNING_FULL_SNAPSHOT
FAILED_FULL_SNAPSHOT
PENDING_SHARD_RELOCATION
RUNNING_SHARD_RELOCATION
FINISHED_SHARD_RELOCATION
```

如这些状态所示，迁移可能会在快照、分片重新定位或强制合并期间失败。快照或分片重新定位期间的故障通常是由于节点故障或 S3 连接问题造成的。磁盘空间不足通常是强制合并失败的根本原因。

迁移完成后，同一 `_status` 请求返回错误。如果您在此时检查索引，您可以看到暖索引独有的一些设置：

```
GET my-index/_settings

{
  "my-index": {
    "settings": {
      "index": {
        "refresh_interval": "-1",
        "auto_expand_replicas": "false",
        "provided_name": "my-index",
        "creation_date": "1599241458998",
        "unassigned": {
          "node_left": {
            "delayed_timeout": "5m"
```

```
          }
        },
        "number_of_replicas": "1",
        "uuid": "GswyCdR0RSq0SJYmzsIpiw",
        "version": {
          "created": "7070099"
        },
        "routing": {
          "allocation": {
            "require": {
              "box_type": "warm"
            }
          }
        },
        "number_of_shards": "5",
        "merge": {
          "policy": {
            "max_merge_at_once_explicit": "50"
          }
        }
      }
    }
  }
}
```

- 在这种情况下，`number_of_replicas` 是不消耗磁盘空间的被动副本的数量。
- `routing.allocation.require.box_type` 指定索引应使用热节点而不是标准数据节点。
- `merge.policy.max_merge_at_once_explicit` 指定迁移期间要同时合并的段数。

暖存储中的索引是只读的，除非您将它们退回到热存储，这使得不可变的数据 UltraWarm 最适合于不可变的数据，例如日志。您可以查询索引并将其删除，但无法添加、更新或删除单个文档。如果尝试，您可能会遇到以下错误：。

```
{
  "error" : {
    "root_cause" : [
      {
        "type" : "cluster_block_exception",
        "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage
 exceeded flood-stage watermark, index has read-only-allow-delete block];"
      }
```

```
    ],
    "type" : "cluster_block_exception",
    "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage exceeded
 flood-stage watermark, index has read-only-allow-delete block];"
  },
  "status" : 429
}
```

自动执行迁移

我们建议在索引达到特定期限或满足其他条件后使用 the section called “索引状态管理” 自动执行迁移过程。请参阅演示该工作流程的示例策略。

迁移调整

向 UltraWarm 存储的索引迁移需要强制合并。每个 OpenSearch 索引由一定数量的分片组成，每个分片由一定数量的 Lucene 段组成。强制合并操作清除标记为删除的文档，并节省磁盘空间。默认情况下，会将索引 UltraWarm 合并到一个段中，kNN 索引除外，其中使用默认值 20。

您可以将此值更改为达 1,000 个分段，使用 `index.ultrawarm.migration.force_merge.max_num_segments` 设置。更高的值会加快迁移过程，但在迁移完成后会增加温索引的查询延迟。要更改设置，请执行下列请求：

```
PUT my-index/_settings
{
  "index": {
    "ultrawarm": {
      "migration": {
        "force_merge": {
          "max_num_segments": 1
        }
      }
    }
  }
}
```

要检查迁移过程的此阶段花费的时间，请监视 `HotToWarmMigrationForceMergeLatency` 指标。

取消迁移

UltraWarm 按顺序在队列中处理迁移。如果迁移在队列中，但尚未开始，您可以使用以下请求将迁移从队列中删除：

```
POST _ultrawarm/migration/_cancel/my-index
```

如果您的域使用精细访问控制，则必须使用 `indices:admin/ultrawarm/migration/cancel` 权限提出此请求。

列出热索引和暖索引

UltraWarm 增加了两个附加选项（类似于）以`_all`帮助管理热索引和暖索引。有关所有暖索引或热索引的列表，请提出以下请求：

```
GET _warm
GET _hot
```

您可以在指定索引的其他请求中使用这些选项，例如：

```
_cat/indices/_warm
_cluster/state/_all/_hot
```

将温索引返回到热存储

如果您需要再次写入索引，请将其迁移回热存储：

```
POST _ultrawarm/migration/my-index/_hot
```

您一次最多可以执行 10 项从温存储到热存储的排队迁移。 OpenSearch Service 按照迁移请求的排队顺序，一次处理一个请求。要查看当前数量，请监控`WarmToHotMigrationQueueSize`指标。

迁移完成后，检查索引设置以确保它们满足需求。索引使用一个副本返回热存储。

从快照恢复温索引

除用于自动快照的标准存储库之外，还 UltraWarm 增加了用于一个暖索引的第二存储库`cs-ultrawarm`。此存储库中的每个快照只包含一个索引。如果删除温索引，其快照将保留在 `cs-ultrawarm` 存储库中 14 天，就像任何其他自动执行的快照一样。

从 `cs-ultrawarm` 还原快照时，快照会恢复到温存储，而不是热存储。`cs-automated-enc` 和 `cs-automated` 存储库中的快照还原到热存储。

将 UltraWarm 快照恢复到温存储空间

1. 标识包含要还原的索引的最新快照：

```
GET _snapshot/cs-ultrawarm/_all?verbose=false

{
  "snapshots": [{
    "snapshot": "snapshot-name",
    "version": "1.0",
    "indices": [
      "my-index"
    ]
  }]
}
```

> **Note**
>
> 默认情况下，`GET _snapshot/<repo>` 操作显示存储库中每个快照的开始时间、结束时间和持续时间等详细数据信息。`GET _snapshot/<repo>` 操作从存储库包含的每个快照文件中检索信息。如果不需要开始时间、结束时间和持续时间，只需要快照名称和索引信息，我们建议您在列出快照时使用 `verbose=false` 参数，以最大限度地缩短处理时间并防止超时。

2. 如果索引已经存在，请将其删除：

```
DELETE my-index
```

如果不想删除索引，请将其返回到热存储和重新编制索引它。

3. 还原快照：

```
POST _snapshot/cs-ultrawarm/snapshot-name/_restore
```

UltraWarm 会忽略您在此还原请求中指定的任何索引设置，但您可以指定，如`rename_pattern`和`rename_replacement`。有关 OpenSearch 快照还原选项的摘要，请参阅OpenSearch 文档。

暖索引的手动快照

您可以手动拍摄暖索引的快照，但我们不建议这样做。自动执行 `cs-ultrawarm` 存储库已包含在迁移期间拍摄的每个热索引的快照，无需额外付费。

默认情况下，S OpenSearch ervice 不会在手动快照中包含温索引。例如，以下调用只包含热索引：

```
PUT _snapshot/my-repository/my-snapshot
```

如果您选择手动拍摄暖索引的快照，则需要考虑几个重要的因素。

- 您不能将热索引和暖索引混合使用。例如，下面的命令将失败：

  ```
  PUT _snapshot/my-repository/my-snapshot
  {
    "indices": "warm-index-1,hot-index-1",
    "include_global_state": false
  }
  ```

  如果将热索引和暖索引混合，则通配符（`*`）语句也会失败。

- 每个快照只能包含一个温索引。例如，下面的命令将失败：

  ```
  PUT _snapshot/my-repository/my-snapshot
  {
    "indices": "warm-index-1,warm-index-2,other-warm-indices-*",
    "include_global_state": false
  }
  ```

  此请求成功：

  ```
  PUT _snapshot/my-repository/my-snapshot
  {
    "indices": "warm-index-1",
    "include_global_state": false
  }
  ```

- 手动快照始终恢复到热存储，即使它们最初包含热索引也是如此。

将温索引迁移到冷存储

如果您的数据不经常查询 UltraWarm ，请考虑将其迁移到冷存储。冷存储适用于您偶尔访问的数据或不再处于活动状态的数据。您无法读取或写入冷索引，但可以在需要查询时免费将它们迁移回暖存储。有关说明，请参阅将索引迁移到冷存储。

KNN 索引的最佳实践

- Ultrawarm/Cold 等级适用于所有 KNN 索引引擎类型。我们建议使用 Lucene 引擎和磁盘优化的向量搜索的 KNN 索引使用它，这不需要将图形数据完全加载到堆外内存中。在将其与 FAISS 和 NMSLIB 等原生内存引擎一起使用时，必须考虑要主动搜索的分片图形大小，并相应地配置 UltraWarm实例，最好是实例类型的实例。`uw.large`例如，如果客户配置了 2 个`uw.large`实例，则每个实例将有大约 `knn.memory.circuit_breaker.limit * 61` GiB 的可用堆外内存。如果您的所有热查询都针对累积图形大小不超过可用堆外内存的分片，则可以获得最佳性能。如果由于驱逐和等待堆外内存可用而导致可用内存低于加载图表所需的内存，则延迟会受到影响。这就是为什么我们不建议将`uw.medium`实例用于使用内存引擎的用例或更高的搜索吞吐量案例，无论引擎如何。
- 迁移到的 KNN 索引 UltraWarm 不会被强制合并到单个片段。这样可以避免对出现 OOM 问题的热节点和温节点产生任何影响，因为图形大小对于内存引擎来说变得太大。由于每个分片的分段数量增加，这可能会导致消耗更多的本地缓存空间并允许更少的索引迁移到温层。您可以选择使用现有设置将索引强制合并到单个段，然后在将索引迁移到热层之前将其覆盖。有关更多信息，请参阅 the section called “迁移调整”。
- 如果您的用例很少搜索索引，并且不处理延迟敏感的工作负载，则可以选择将这些索引迁移到该 UltraWarm 层。这将帮助您缩小热层计算实例的规模，让 UltraWarm 分层计算处理对此类低优先级索引的查询。这还有助于隔离低优先级和高优先级索引的查询之间消耗的资源，这样它们就不会相互影响。

禁用 UltraWarm

控制台是禁用的最简单方法 UltraWarm。选择域、Actions（操作）和 Edit cluster configuration（编辑集群配置）。取消选择 “启用热数据节点”，然后选择 “保存更改”。还可以在 Amazon CLI 和配置 API 中使用 `WarmEnabled` 选项。

在禁用之前 UltraWarm，必须删除所有温索引或将其迁移回热存储。暖存储空间为空后，请等待五分钟，然后尝试禁用 UltraWarm。

用于 Amazon OpenSearch 服务的冷存储

冷存储允许您在 Amazon Ser OpenSearch vice 域中存储任何数量不经常访问的数据或历史数据，并以低于其他存储层的成本进行按需分析。如果您需要对旧数据进行定期研究或取证分析，则适合使用冷存储。适用于冷存储的数据的实际示例包括不经常访问的日志、为满足法规遵从性要求而必须保留的数据或具有历史价值的日志。

与UltraWarm存储类似，冷存储由 Amazon S3 提供支持。当您需要查询冷数据时，可以选择性地将其附加到现有 UltraWarm节点。您可以手动或使用索引状态管理策略管理冷数据的迁移和生命周期。

主题

- 先决条件
- 冷存储要求和性能注意事项
- 冷存储定价
- 启用冷存储
- 在控制 OpenSearch 面板中管理冷索引
- 将索引迁移到冷存储
- 自动迁移到冷存储
- 取消迁移到冷存储
- 列出冷索引
- 将冷索引迁移到温存储
- 从快照恢复冷索引
- 取消从冷存储迁移到热存储
- 更新冷索引元数据
- 删除冷索引
- 禁用冷存储

先决条件

冷存储具有以下先决条件：

- 冷存储需要 OpenSearch 或弹性搜索版本 7.9 或更高版本。
- 若要在 OpenSearch 服务域上启用冷存储，还必须在同一个域上启用热存储。
- 要使用冷存储，域必须具有专用的主节点。

- 如果您的域为数据节点使用 T2 或者 T3 实例类型，则无法使用冷存储。
- 如果您的索引使用近似 k-NN (`"index.knn":true`)，则可以将其移至 2.17 及更高版本的冷存储。2.17 之前版本的域名可以升级到 2.17 以使用此功能，但是在 2.x 之前的版本上创建的 KNN 索引无法迁移到 Cold。
- 如果您的域使用精细访问控制，则非管理员用户必须映射至控制 OpenSearch 面板中的`cold_manager`角色，以管理冷索引。

> **Note**
>
> 此`cold_manager`角色可能不存在于某些预先存在的 OpenSearch 服务域上。如果没有在控制面板中看到角色，则需要手动创建它。

配置权限

如果在预先存在的 OpenSearch 服务域上启用冷存储，则`cold_manager`角色可能无法在域上定义。如果您的域使用精细访问控制，则非管理员用户必须映射至此角色，以管理冷索引。手动创建 `cold_manager` 角色，请执行下列步骤：

1. 在 “ OpenSearch 控制面板” 中，转至 “安全”，然后选择 “权限”。
2. 选择创建操作组并配置以下组：

组名	权限
`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. 选择 Role (角色)，然后选择 Create role (创建角色)。
4. 将角色命名为 cold_manager。

5. 针对集群权限，选择您创建的 `cold_cluster` 组。
6. 针对索引，输入 `*`。
7. 针对索引权限，选择您创建的 `cold_index` 组。
8. 选择创建。
9. 创建角色之后，将其映射设置为管理冷索引的任何用户或后端角色。

冷存储要求和性能注意事项

由于冷存储使用 Amazon S3，因此它不会产生热存储的开销，例如副本、Linux 预留空间和 OpenSearch 服务预留空间。冷存储没有特定的实例类型，因为它没有附加任何计算容量。您可以将任意数量的数据存储在冷存储中。监控 Amazon CloudWatch 中的`ColdStorageSpaceUtilization`指标，查看您使用的冷存储空间量。

冷存储定价

与 UltraWarm 存储类似，使用冷存储，您只需为数据存储付费。冷数据没有计算成本，如果冷存储中没有数据，则不会收取计费。

在冷存储和热存储之间移动数据时，您不会产生任何传输费用。当索引在温存储和冷存储之间迁移时，您仍然只需为索引的一个副本付费。迁移完成后，索引将根据其迁移到的存储层计费。有关冷存储成本的更多信息，请参阅 Amazon OpenSearch 服务成本的更多信息。

启用冷存储

控制台是创建使用冷存储的域的最简单方法。创建域时，请先选择 “启用温数据节点”，因为您必须在同一个域上启用热存储。然后，选择 “启用冷存储”。

相同的基本过程适用于现有域，前提是它们满足先决条件。即使域状态从 Processing (正在处理) 变为 Active (活动) 后，冷存储可能也会在几个小时内无法使用。

您也可以使用 Amazon CLI 或者配置 API以启用冷存储。

示例 CLI 命令

以下 Amazon CLI 命令创建一个具有三个数据节点、三个专用主节点、启用冷存储和启用精细访问控制的域：

```
aws opensearch create-domain \
  --domain-name my-domain \
```

```
  --engine-version Opensearch_1.0 \
  --cluster-
config ColdStorageOptions={Enabled=true},WarmEnabled=true,WarmCount=4,WarmType=ultrawarm1.mediu
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --region us-east-2
```

有关更多信息，请参阅 Amazon CLI 命令参考。

示例配置 API 请求

以下对配置 API 的请求创建一个具有三个数据节点、三个专用主节点、启用冷存储和启用精细访问控制的域：

```
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"
}
```

有关详细信息，请参阅《亚马逊 OpenSearch 服务 API 参考》。

在控制 OpenSearch 面板中管理冷索引

您可以使用 OpenSearch 服务域中的现有控制面板界面管理热索引、温索引和冷索引。使用控制面板，您可以在温存储和冷存储之间迁移索引，并监控索引迁移状态，而无需使用 CLI 或配置 API。有关更多信息，请参阅管理 OpenSearch 控制面板中的索引。

将索引迁移到冷存储

将索引迁移到冷存储时，您可以为数据提供一个时间范围，以便更轻松地发现。您可以根据索引中的数据选择时间戳字段，手动提供开始和结束时间戳，或选择不指定时间戳。

参数	支持的值	描述
`timestamp_field`	索引映射中的日期/时间字段。	所提供字段的最小值和最大值将计算并存储为冷索引的

参数	支持的值	描述
		start_time 和 end_time 元数据。
start_time 和 end_time	使用以下格式之一： • strict_date_optional_time。例如：yyyy-MM-dd'T'HH:mm:ss.SSSZ 或 yyyy-MM-dd • 以毫秒为单位的纪元时间	所提供的值将存储为冷索引的 start_time 和 end_time 元数据。

如果您不想指定时间戳，请将 ?ignore=timestamp 添加到请求。

以下请求将热索引迁移到冷存储，并为该索引中的数据提供开始和结束时间：

```
POST _ultrawarm/migration/my-index/_cold
  {
    "start_time": "2020-03-09",
    "end_time": "2020-03-09T23:00:00Z"
  }
```

然后检查迁移的状态：

```
GET _ultrawarm/migration/my-index/_status

{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_METADATA_RELOCATION",
    "migration_type": "WARM_TO_COLD"
  }
}
```

OpenSearch Service 一次将一个索引迁移至冷存储。队列中最多可包含 100 个迁移。任何超出限制的请求都将被拒绝。要检查队列中的当前迁移数，请监控 WarmToColdMigrationQueueSize 指标。迁移过程具有以下状态：

```
ACCEPTED_COLD_MIGRATION - Migration request is accepted and queued.
```

```
RUNNING_METADATA_MIGRATION - The migration request was selected for execution and
 metadata is migrating to cold storage.
FAILED_METADATA_MIGRATION - The attempt to add index metadata has failed and all
 retries are exhausted.
PENDING_INDEX_DETACH - Index metadata migration to cold storage is completed. Preparing
 to detach the warm index state from the local cluster.
RUNNING_INDEX_DETACH - Local warm index state from the cluster is being removed. Upon
 success, the migration request will be completed.
FAILED_INDEX_DETACH - The index detach process failed and all retries are exhausted.
```

自动迁移到冷存储

我们建议在索引达到特定期限或满足其他条件后使用索引状态管理自动执行迁移过程。请参阅示例策略，其中演示了如何自动将索引从热存储迁移 UltraWarm 到冷存储。

> **Note**
>
> 需要显式 `timestamp_field` 才能使用索引状态管理策略将索引移动到冷存储。

取消迁移到冷存储

如果迁移到冷存储器已排队或处于失败状态，您可以使用以下请求取消迁移：

```
POST _ultrawarm/migration/_cancel/my-index

{
  "acknowledged" : true
}
```

如果您的域使用精细访问控制，则需要 `indices:admin/ultrawarm/migration/cancel` 权限提出此请求。

列出冷索引

在查询之前，您可以列出冷存储中的索引，以决定哪些索引要迁移到哪些索引，以 UltraWarm 进行进一步分析。以下请求列出所有冷索引（按索引名称排序）：

```
GET _cold/indices/_search
```

示例响应

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

过滤

您可以根据基于前缀的索引模式和时间范围偏移过滤冷索引。

以下请求列出了匹配 `event-*` 的前缀模式的索引：

```
GET _cold/indices/_search
 {
   "filters":{
      "index_pattern": "event-*"
```

```
    }
  }
```

示例响应

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
  "indices" : [
    {
      "index" : "events-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

以下请求将返回 2019-03-01 和 2020-03-01 之间的 start_time 和 end_time 元数据字段的索引：

```
GET _cold/indices/_search
{
  "filters": {
    "time_range": {
      "start_time": "2019-03-01",
      "end_time": "2020-03-01"
    }
  }
}
```

示例响应

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
  "indices" : [
    {
      "index" : "my-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
```

```
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2019-05-09T00:00Z",
      "end_time" : "2019-09-09T23:00Z"
    }
  ]
}
```

排序

您可以按索引名称或大小等元数据字段对冷索引进行排序。以下请求按降序列出了所有按大小排序的索引：

```
GET _cold/indices/_search
 {
 "sort_key": "size:desc"
 }
```

示例响应

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 5,
  "indices" : [
    {
      "index" : "my-index-6",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-9",
      "index_cold_uuid" : "mbD3ZRVDRI6ONqgEOsJyUA",
      "size" : 57922,
      "creation_date" : "2021-07-07T23:41:35.640Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-5",
```

```
      "index_cold_uuid" : "EaeXOBodTLiDYcivKsXVLQ",
      "size" : 32403,
      "creation_date" : "2021-07-08T00:12:01.523Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

其他有效的排序键是 `start_time:asc/desc`、`end_time:asc/desc` 和 `index_name:asc/desc`。

分页

您可以对冷索引列表进行分页。使用 `page_size` 参数（默认值为 10）配置每页要返回的索引数。冷索引中的每个 `_search` 请求将返回 `pagination_id`，您可以将其用于后续调用。

以下请求对冷索引的 `_search` 请求结果进行分页，并显示接下来的 100 个结果：

```
GET _cold/indices/_search?page_size=100
{
"pagination_id": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
}
```

将冷索引迁移到温存储

使用上一节中的筛选条件缩小冷索引列表的范围后，请将其迁移回您可以查询数据 UltraWarm 的位置，并使用它来创建可视化效果。

以下请求会将两个冷索引迁移回热存储：

```
POST _cold/migration/_warm
 {
 "indices": "my-index1,my-index2"
 }


{
  "acknowledged" : true
}
```

要检查迁移的状态并检索迁移 ID，请发送以下请求：

```
GET _cold/migration/_status
```

示例响应

```
{
  "cold_to_warm_migration_status" : [
    {
      "migration_id" : "tyLjXCA-S76zPQbPVHkOKA",
      "indices" : [
        "my-index1,my-index2"
      ],
      "state" : "RUNNING_INDEX_CREATION"
    }
  ]
}
```

要获取特定于索引的迁移信息，请包括索引名称：

```
GET _cold/migration/my-index/_status
```

您可以按索引的当前迁移状态列出索引，而不是指定索引。有效值包括 `_failed`、`_accepted` 和 `_all`。

以下命令可获取单个迁移请求中所有索引的状态：

```
GET _cold/migration/_status?migration_id=my-migration-id
```

使用状态请求检索迁移 ID。有关迁移的详细信息，请添加 `&verbose=true`。

您可以分批（10 个或更少）将索引从冷存储迁移到温存储，最多可同时迁移 100 个索引。任何超出限制的请求都将被拒绝。要检查目前执行的迁移数，请监控 `ColdToWarmMigrationQueueSize` 指标。迁移过程具有以下状态：

```
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.
```

从快照恢复冷索引

如果需要还原已删除的冷索引，您可以按照 the section called “从快照恢复温索引” 中的说明将其还原到暖层，然后再次将索引迁回冷层。无法将已删除的冷索引直接还原到冷层。 OpenSearch 删除后，服务将保留冷索引 14 天。

取消从冷存储迁移到热存储

如果从冷存储到热存储的索引迁移已排队或处于失败状态，您可以通过以下请求取消它：

```
POST _cold/migration/my-index/_cancel

{
  "acknowledged" : true
}
```

要取消一批索引的迁移（一次最多 10 个），请指定迁移 ID：

```
POST _cold/migration/_cancel?migration_id=my-migration-id

{
  "acknowledged" : true
}
```

使用状态请求检索迁移 ID。

更新冷索引元数据

您可以更新 `start_time` 和 `end_time` 字段，用于冷索引：

```
PATCH _cold/my-index
 {
 "start_time": "2020-01-01",
 "end_time": "2020-02-01"
 }
```

您无法更新在冷存储索引中的 `timestamp_field`。

> **Note**
>
> OpenSearch 控制面板不支持 PATCH 方法。使用 curl、Postman 或其他方法来更新冷元数据。

删除冷索引

如果不使用 ISM 策略，则可以手动删除冷索引。以下请求删除冷索引：

```
DELETE _cold/my-index

{
  "acknowledged" : true
}
```

禁用冷存储

OpenSearch 服务控制台是禁用冷存储的最简单方法。选择域，依次选择 Actions（操作）、Edit cluster configuration（编辑集群配置），然后取消选择 Enable cold storage（启用冷存储）。

要使用 Amazon CLI 或配置 API`ColdStorageOptions`，请在下方设置`"Enabled"="false"`。

在禁用冷存储之前，必须先删除所有冷索引或将其迁移回温存储，否则禁用操作将失败。

OpenSearch 针对 Amazon OpenSearch 服务进行了优化的存储

针对 Amazon S OpenSearch ervice 的 OpenSearch 优化实例系列是一种经济实惠的解决方案，用于存储大量数据。拥有 OR1 实例的域使用 Amazon Elastic Block Store（Amazon gp3 EBS）`io1`或卷作为主存储，并在数据到达时同步复制到 Amazon S3。这种存储结构提供更高的索引吞吐量和较高的耐久性。 OpenSearch 优化型实例系列还支持在发生故障时自动恢复数据。有关 OR1 实例类型选项的信息，请参阅the section called “当前一代实例类型”。

如果您要运行有大量索引的运营分析工作负载，例如日志分析、可观测性或安全分析，则可以受益于 OR1 实例更好的性能和计算效率。此外， OR1实例提供的自动数据恢复功能也可提高域的整体可靠性。

OpenSearch 服务向 Amazon 发送与存储相关的 OR1 指标。 CloudWatch有关可用指标的列表，请参阅 ???。

OR1 实例可按需使用，也可使用预留实例定价，对于在 Amazon EBS 和 Amazon S3 中预置的实例和存储按小时收费。

主题

- 限制
- 进行微调以提高摄取吞吐量
- OpenSearch 优化后的实例与其他实例有何不同
- 与 UltraWarm存储有何 OR1 不同
- 为域配置实 OR1 例

限制

将 OR1 实例用于域时，请注意以下限制。

- 新创建的域必须运行 2.11 及更高 OpenSearch 版本。
- 现有的域必须运行 2.15 及 OpenSearch 更高版本。
- 域必须启用静态加密。有关更多信息，请参阅 ???。
- 如果您的域使用专用主节点，则这些节点必须使用 Graviton 实例。有关专用主节点的更多信息，请参阅 ???。
- OR1 实例上的索引刷新间隔必须为 10 秒或更长。 OR1 实例的默认刷新间隔为 10 秒。

进行微调以提高摄取吞吐量

为了使 OR1 实例获得最佳的索引吞吐量，建议您执行以下操作：

- 使用大批量大小来提高缓冲区利用率。建议的大小为 10 MB。
- 使用多个客户端来提高并行处理性能。
- 活动主分片数量的设置要与数据节点的数量相匹配，从而尽可能提高资源利用率。

OpenSearch 优化后的实例与其他实例有何不同

OpenSearch 优化型实例与非优化型实例的区别如下：

- 对于 OpenSearch 优化型实例，仅在主分片上创建索引。

- 如果 OpenSearch 优化型实例配置了副本，则显示的索引创建速率可能低于实际水平。例如，假设有一个主分片和一个副本分片，则当实际索引创建速率为 2000 时，索引创建速率可能显示为 1000。
- OpenSearch 优化型实例在发送到远程源之前会执行缓冲操作。这会导致摄取延迟更高。

> **Note**
>
> 该`IndexingLatency`指标不受影响，因为该指标不包括同步 translog 的时间。

- 副本分片相比主分片可能有几秒的延迟。您可以使用 A `ReplicationLagMaxTime` mazon CloudWatch 指标监控延迟

与 UltraWarm存储有何 OR1 不同

OpenSearch Service 提供的 UltraWarm 实例以经济高效的方式存储大量只读数据。两者都在 OR1 本地将 UltraWarm 数据存储在 Amazon EBS 中，并远程将数据存储在 Amazon S3 中。但是，UltraWarm 实例 OR1 之间存在以下几个重要的区别：

- OR1 实例在本地和远程存储中都保存了一份数据。而 UltraWarm 对于，数据主要保存在远程存储中，从而降低存储成本。根据您的使用模式，可以将数据移动到本地存储。
- OR1 实例处于活动状态，可以接受读取和写入操作，而在您手动将数据移回热存储之前，UltraWarm 实例上的数据是只读的。
- UltraWarm 依靠索引快照来保证数据的持久性。 OR1 与此相对，实例在幕后执行复制和恢复。如果索引为红色， OR1 实例将自动从 Amazon S3 中的远程存储中恢复缺失的分片。恢复时间因要恢复的数据量而异。

有关 UltraWarm 存储的更多信息，请参阅???。

为域配置实 OR1 例

使用 Amazon Web Services Management Console 或 Amazon Command Line Interface (Amazon CLI) 创建新域时，您可以为数据节点选择 OR1 实例。您可以使用现有的工具来创建索引和查询数据。

控制台

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择 域。

3. 选择创建域。
4. 在数据节点数量部分中，展开实例系列菜单并选择OpenSearch 已优化。
5. 选择实例类型和其他存储设置。
6. 在 “加密” 部分中，确保选中 “启用静态数据加密”。
7. 配置域的其余部分，然后选择 “创建”。

Amazon CLI

要使用配置使用 OR1 存储空间的域 Amazon CLI，您必须在中提供特定 OR1 实例类型大小的值`InstanceType`。

以下示例创建一个 OR1 实例大小为的域`2xlarge`并启用静态加密。

```
aws opensearch create-domain \
  --domain-name test-domain \
  --engine-version OpenSearch_2.11 \
  --cluster-config
 "InstanceType=or1.2xlarge.search,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedMasterTy
 \
  --ebs-options "EBSEnabled=true,VolumeType=gp3,VolumeSize=200" \
  --encryption-at-rest-options Enabled=true \
  --advanced-security-options
 "Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions={MasterUserName=test-
user,MasterUserPassword=test-password}" \
  --node-to-node-encryption-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"AWS":"*"},"Action":"es:*","Resource":"arn:aws:es:us-
east-1:account-id:domain/test-domain/*"}]}'
```

Amazon S OpenSearch ervice 中的索引状态管理

通过 Amazon S OpenSearch ervice 中的索引状态管理 (ISM)，您可以定义自动执行日常任务的自定义管理策略，并将这些策略应用于索引和索引模式。您不再需要设置和管理外部进程来运行索引操作。

策略包含一个默认状态和一个状态列表，以供索引在这些状态之间转换。在每个状态中，您可以定义要执行的一系列操作和触发这些转换的条件。典型的使用案例是在一段时间后定期删除旧索引。例如，您可以定义一个策略以在 30 天后将索引移入 `read_only` 状态，然后在 90 天后最终将其删除。

将策略附加到索引后，ISM 会创建一个每 5 到 8 分钟（对于早于 1.3 版本的集群，则为 30 到 48 分钟）运行一次的作业，以执行策略操作、检查条件并将索引转换为不同的状态。该作业运行的基本时间是每 5 分钟一次，外加 0-60% 的随机抖动，以确保您不会同时看到所有索引的活动激增。如果集群状态为红色，则 ISM 不会运行作业。

ISM 需要 Elasticsearch 6.8 OpenSearch 或更高版本。

> **Note**
>
> 本文档简要概述了 ISM 和几个策略示例。此外还介绍了适用于 Amazon S OpenSearch ervice 域的 ISM 与 ISM 有何不同 OpenSearch。有关 ISM 的完整文档，包括全面的参数参考、对各项设置的说明以及 API 参考，请参阅 OpenSearch 文档中的 Index State Managem ent。

> **Important**
>
> 您不能再使用索引模板将 ISM 策略应用于新创建的索引。您可以继续使用 ISM 模板字段自动管理新创建的索引。此更新引入了使用此设置影响现有 CloudFormation 模板的重大更改。

创建一个 ISM 策略

开始使用索引状态管理

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 选择要为其创建 ISM 策略的域。
3. 从域的控制面板中，导航到控制 OpenSearch 面板 URL，然后使用您的主用户名和密码登录。URL 遵循以下格式：

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

4. 在控制面板中打开左侧导航面 OpenSearch 板，选择索引管理，然后选择创建策略。
5. 使用可视化编辑器或 JSON 编辑器创建策略。我们建议使用可视化编辑器，因为它提供了一种更结构化的策略定义方式。有关创建策略的帮助，请参阅下面的示例策略。
6. 创建策略后，请将它附加到一个或多个索引：

   ```
   POST _plugins/_ism/add/my-index
   {
   ```

```
  "policy_id": "my-policy-id"
}
```

> **Note**
>
> 如果您的域名运行的是旧版 Elasticsearch 版本，请使用 `_opendistro` 而非 `_plugins`。

或者，在 OpenSearch 控制面板中选择索引，然后选择应用策略。

示例策略

以下示例策略演示了如何自动执行常见 ISM 使用案例。

从热存储到冷存储

此示例策略将索引从热存储移动到热存储 UltraWarm，并最终移动到 冷库。然后，删除索引。

索引最初处于 `hot` 状态。10 天后，ISM 将其移动到 `warm` 状态。80 天后，即索引已存在 90 天后，ISM 将索引移动到 `cold` 状态。一年后，该服务向 Amazon Chime 房间发送通知以告知该索引正在被删除，然后永久删除该索引。

请注意，冷索引需要 `cold_delete` 操作而不是正常的 `delete` 操作。另请注意，数据中需要显式 `timestamp_field`，以便使用 ISM 管理冷索引。

```
{
  "policy": {
    "description": "Demonstrate a hot-warm-cold-delete workflow.",
    "default_state": "hot",
    "schema_version": 1,
    "states": [{
        "name": "hot",
        "actions": [],
        "transitions": [{
          "state_name": "warm",
          "conditions": {
            "min_index_age": "10d"
          }
        }]
      },
```

```
      {
        "name": "warm",
        "actions": [{
          "warm_migration": {},
          "retry": {
            "count": 5,
            "delay": "1h"
          }
        }],
        "transitions": [{
          "state_name": "cold",
          "conditions": {
            "min_index_age": "90d"
          }
        }]
      },
      {
        "name": "cold",
        "actions": [{
            "cold_migration": {
              "timestamp_field": "<your timestamp field>"
            }
          }
        ],
        "transitions": [{
          "state_name": "delete",
          "conditions": {
             "min_index_age": "365d"
          }
        }]
      },
      {
        "name": "delete",
        "actions": [{
          "notification": {
            "destination": {
              "chime": {
                "url": "<URL>"
              }
            },
            "message_template": {
              "source": "The index {{ctx.index}} is being deleted."
            }
          }
```

```
        },
        {
          "cold_delete": {}
        }]
      }
    ]
  }
}
```

减少副本数

此示例策略在七天后将副本计数减少到零以节省磁盘空间，然后在 21 天后删除索引。此策略假定索引是非关键索引，并不再接收写入请求。副本数量为零会带来一定的数据丢失风险。

```
{
  "policy": {
    "description": "Changes replica count and deletes.",
    "schema_version": 1,
    "default_state": "current",
    "states": [{
        "name": "current",
        "actions": [],
        "transitions": [{
          "state_name": "old",
          "conditions": {
            "min_index_age": "7d"
          }
        }]
      },
      {
        "name": "old",
        "actions": [{
          "replica_count": {
            "number_of_replicas": 0
          }
        }],
        "transitions": [{
          "state_name": "delete",
          "conditions": {
            "min_index_age": "21d"
          }
        }]
      },
```

```
      {
        "name": "delete",
        "actions": [{
          "delete": {}
        }],
        "transitions": []
      }
    ]
  }
}
```

拍摄索引快照

此示例策略使用 `snapshot` 操作，以便在索引包含至少一个文档时立即拍摄该索引的快照。`repository` 是您在 Amazon S3 中注册的手动快照存储库的名称。`snapshot` 是快照的名称。有关注册存储库的快照先决条件和步骤，请参阅 the section called “创建索引快照”。

```
{
  "policy": {
    "description": "Takes an index snapshot.",
    "schema_version": 1,
    "default_state": "empty",
    "states": [{
        "name": "empty",
        "actions": [],
        "transitions": [{
          "state_name": "occupied",
          "conditions": {
            "min_doc_count": 1
          }
        }]
      },
      {
        "name": "occupied",
        "actions": [{
          "snapshot": {
            "repository": "<my-repository>",
            "snapshot": "<my-snapshot>"
            }
          }],
          "transitions": []
      }
    ]
```

```
  }
}
```

ISM 模板

您可以在策略中设置 `ism_template` 字段，因此当您创建与模板模式匹配的索引时，策略会自动附加到该索引。在此示例中，以 "log" 开头的名称创建的任何索引都会自动匹配 ISM 策略 `my-policy-id`：

```
PUT _plugins/_ism/policies/my-policy-id
{
  "policy": {
    "description": "Example policy.",
    "default_state": "...",
    "states": [...],
    "ism_template": {
      "index_patterns": ["log*"],
      "priority": 100
    }
  }
}
```

有关更详细的示例，请参阅使用 ISM 模板进行自动回滚的策略示例。

差异

与 Elasticsearch 相比，适用于 Amazon S OpenSearch ervice 的 ISM 有几个区别。 OpenSearch

ISM 操作

- OpenSearch 服务支持三种独特的 ISM 操作：`warm_migration``cold_migration`、和`cold_delete`：
  - 如果您的域已UltraWarm启用，则`warm_migration`操作会将索引转换为温存储。
  - 如果您的域启用了冷存储，则 `cold_migration` 操作会将索引转换为冷存储，而 `cold_delete` 操作将从冷存储中删除此索引。

  即使这些操作中的其中一项操作未在设置的超时期限内完成，索引的迁移或删除仍会继续。设置上述的任何一个操作设置 error_notification 后，如果该操作未在超时期限内完成，则系统会通知您该操作失败，但此通知仅供您参考。实际操作没有固有的超时时间，并将继续运行直到最终成功或失败。

- 如果您的域运行 OpenSearch Elasticsearch 7.4 或更高版本，则 S OpenSearch ervice 支持 ISM open 和操作。close

- 如果您的域运行 OpenSearch Elasticsearch 7.7 或更高版本，则 Service 支持 OpenSearch ISM 操作。snapshot

冷存储 ISM 操作

对于冷索引，当您使用以下 ISM 时，必须指定?type=_cold参数 APIs：

- 添加策略
- 移除策略
- 更新策略
- 重试失败的索引
- 解释索引

这些冷索引 APIs 的 Index 有以下附加区别：

- 除非您在末尾使用，否则不支持通配符运算符。例如，支持 _plugins/_ism/<add, remove, change_policy, retry, explain>/logstash-*|，但不支持 _plugins/_ism/<add, remove, change_policy, retry, explain>/iad-*-prod。
- 不支持多个索引名称和模式。例如，支持 _plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs|，但不支持 _plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs,sample-data。

ISM 设置

OpenSearch 而且 Elasticsearch 让您可以使用 API 更改所有可用的 _cluster/settings ISM 设置。在 Amazon S OpenSearch ervice 上，您只能更改以下 ISM 设置：

- 集群级别设置：
  - plugins.index_state_management.enabled
  - plugins.index_state_management.history.enabled
- 索引级别设置：
  - plugins.index_state_management.rollover_alias

教程：实现索引状态管理过程的自动化

本教程演示如何实施自动执行日常索引管理任务的 ISM 策略，并将其应用于索引和索引模式。

通过 Amazon S OpenSearch ervice 中的@@ 索引状态管理 (ISM)，您可以自动执行定期索引管理活动，从而避免使用其他工具来管理索引生命周期。您可以创建一个策略，根据索引期限、大小和其他条件自动执行这些操作，所有这些操作都在您的 Amazon S OpenSearch ervice 域中完成。

OpenSearch Service 支持三个存储层：用于主动写入和低延迟分析的默认 “热” 状态， UltraWarm 用于存储高达 3 PB 只读数据的 “热” 状态，以及用于无限长期存档的冷存储。

本教程提供在每日索引中处理时间序列数据的使用案例示例。在本教程中，您将设置一个策略，该策略在 24 小时后自动拍摄每个附加索引的快照。然后，该策略会在两天后将索引从默认的热状态迁移至 UltraWarm 存储，并在 30 天后迁移至冷存储，最后在 60 天后删除索引。

先决条件

- 您的S OpenSearch ervicsearch 版本必须运行 Elasticsearch 版本 6.8 或更高版本。
- 您的域名必须已UltraWarm启用冷存储。
- 您必须为域注册一个手动快照存储库。
- 您的用户角色需要具有足够的权限才能访问S OpenSearch ervice 控制台。如有必要，验证并配置域的访问权限。

步骤 1：配置 ISM 策略

首先，在 ISM 策略配置 ISM OpenSearch 策略。

1. 在 S OpenSearch ervice 控制台中的域控制面板中，导航到 Das OpenSearch hot URL，然后使用您的主用户名和密码登录。URL 遵循此格式：*domain-endpoint*/_dashboards/。
2. 在 OpenSearch 控制面板中，选择 Add sample data（添加示例数据）并将一个或多个示例索引添加到您的域。
3. 打开左侧导航面板，然后依次选择 Index Management（索引管理）、Create policy（创建策略）。
4. 将该策略命名为 `ism-policy-example`。
5. 将默认策略替换为以下策略：

```
{
  "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` 字段会自动将策略附加到与指定 `index_patterns` 之一匹配的任何新创建的索引。在这种情况下，为所有以 `index-` 开头的索引。您可以修改此字段以匹配您的环境中的索引格式。有关更多信息，请参阅 ISM 模板。

6. 在策略的 `snapshot` 部分，将 *`snapshot-repo`* 替换为您为域注册的快照存储库的名称。您还可以选择替换 *`ism-snapshot`*，该名称是创建快照时的名称。
7. 选择创建。现在，State management policies（状态管理策略）页面上会显示该策略。

步骤 2：将该策略附加到一个或多个索引

现在，您已创建策略，请将其附加到集群中的一个或多个索引。

1. 转至 Hot indicies（热索引）选项卡并搜索 `opensearch_dashboards_sample`，其中列出了您在步骤 1 中添加的所有示例索引。
2. 选择所有索引并选择 A pply p olicy，然后选择您刚刚创建的ism-policy-example策略。

3. 选择应用。

在索引切换不同的状态时，您可以在 Policy managed indices（策略管理索引）页面上监控索引。

使用索引汇总功能汇总 Amazon S OpenSearch ervice 中的索引

Amazon S OpenSearch ervice 中的索引汇总使您能够定期将旧数据汇总到汇总索引中，从而降低存储成本。

您可以选择您感兴趣的字段，并使用索引汇总创建新的索引，只有这些字段聚合到较粗糙的时间存储桶中。您可以以相同的查询性能，以相当于成本的一小部分存储数月或数年的历史数据。

索引汇总需要 Elasticsearch 7.9 OpenSearch 或更高版本。

> **Note**
>
> 本文档有助您开始在 Amazon S OpenSearch ervice 中创建索引汇总作业。有关全面的文档，包括全部可用设置的列表和完整的 API 参考，请参阅 OpenSearch 文档中的 Index rolllups。

创建索引汇总作业

要开始使用，请在 OpenSearch控制面板中选择索引管理。选择汇总作业并选择创建汇总作业。

步骤 1：设置索引

设置源索引和目标索引。源索引是要汇总的索引。目标索引是保存索引汇总结果的位置。

创建索引汇总作业后，您无法更改索引选择。

步骤 2：定义聚合和指标

选择要汇总的聚合（术语和直方图）和指标（平均值、总计、最大值、最小值和值计数）的属性。确保不要添加大量高精细的属性，因为不会节省太多空间。

步骤 3：指定计划

指定计划，以便在摄入索引时汇总索引。默认情况下，将启用索引汇总作业。

步骤 4：审核并创建

检查您的配置，然后选择创建。

步骤 5：搜索目标索引

您可以使用标准的 `_search` API 来搜索目标索引。您无法访问目标索引中数据的内部结构，因为插件会在后台自动重写查询以适应目标索引。这是为了确保您可以对源索引和目标索引使用相同的查询。

要查询目标索引，请将 `size` 设置为 0：

```
GET target_index/_search
{
  "size": 0,
  "query": {
    "match_all": {}
  },
  "aggs": {
    "avg_cpu": {
      "avg": {
        "field": "cpu_usage"
      }
    }
  }
}
```

> **Note**
>
> OpenSearch 版本 2.2 及更高版本支持在一个请求中搜索多个汇总索引。 OpenSearch 2.2 之前的 Elasticsearch OSS 版本仅支持每次搜索一个汇总索引。

转换 Amazon OpenSearch 服务中的索引

索引汇总作业可让您通过将旧数据汇总到精简索引中来减少数据粒度，转换作业则可让您创建以特定字段为中心的数据的不同汇总视图，从而让您可以通过不同的方式对数据进行可视化或分析。

索引转换具有 OpenSearch 控制面板用户界面和 REST API。此功能需要 OpenSearch 1.0 或更高版本。

> **Note**
>
> 本文档简要概述了索引转换，有助您开始在 Amazon S OpenSearch ervice 域上使用索引转换。有关全面的文档和 REST API 参考，请参阅开源 OpenSearch 文档中的 Index transforms。

创建索引转换任务

如果集群中没有任何数据，请使用 OpenSearch 控制面板中的示例飞行数据尝试转换。添加数据后，启动 OpenSearch 控制面板。然后选择索引管理、转换任务和创建转换任务。

步骤 1：选择索引

在索引部分中，选择源索引和目标索引。您可以选择现有目标索引，也可以通过输入该索引的名称来创建新索引。

如果您只想转换源索引的一个子集，请选择添加数据筛选条件，然后使用 OpenSearch 查询 DSL 来指定源索引的子集。

步骤 2：选择字段

选择索引后，选择要在转换作业中使用的字段，以及是使用分组还是聚合。

- 您可以使用分组将数据放置在转换后的索引中的单独存储桶中。例如，如果要将示例航班数据中的所有机场目的地分组，请将字段添加到`DestAirportID`字段的`DestAirportID_terms`目标字段中，并且您可以在转换任务完成后 IDs 在转换后的索引中找到分组的机场。
- 另一方面，聚合让您可用执行简单的计算。例如，您可以在转换任务中包含一个聚合，以定义 `sum_of_total_ticket_price` 计算所有飞机票的总和。然后，您可以分析转换后索引中的新数据。

步骤 3：指定计划

预设情况下，转换任务处于启用状态，并按计划运行。对于 transform execution interval（转换执行间隔），请指定间隔（以分钟、小时或天为单位）。

步骤 4：审核并监控

检查您的配置，然后选择创建。然后监控转换任务状态列。

步骤 5：搜索目标索引

任务完成后，您可以使用标准的 `_search`API 来搜索目标索引。

例如，运行基于字段 `DestAirportID` 转换飞行数据的转换任务后，可以运行以下请求以返回值为 `SFO` 的所有字段：

```
GET target_index/_search
{
  "query": {
    "match": {
      "DestAirportID_terms" : "SFO"
    }
  }
}
```

Amazon OpenSearch 服务的跨集群复制

通过 Amazon S OpenSearch ervice 中的跨集群复制，您可以将用户索引、映射和元数据从一个 OpenSearch 服务域复制到另一个服务域。使用跨集群复制有助于确保在发生中断时进行灾难恢复，并允许您跨地理位置较远的数据中心复制数据以减少延迟。您需要为域间传输 Amazon 的数据支付标准数据传输费用。

跨集群复制遵循主动-被动复制模型，其中本地索引或关注者索引（复制数据的位置）从远程索引或领导者索引中提取数据。领导者索引是指数据源，或者要从中复制数据的索引。关注者索引是指数据目标，或者要将数据复制到 的索引。

跨集群复制适用于运行 Elasticsearch 7.10 或 1.1 或 OpenSearch 更高版本的域名。

> **Note**
>
> 本文档介绍如何从 Amazon OpenSearch 服务的角度设置跨集群复制。这包括使用 Amazon Web Services Management Console 来设置跨集群连接，这在自我管理 OpenSearch 的集群上是不可能的。有关完整文档，包括设置参考和全面的 API 参考，请参阅 OpenSearch 文档中的跨集群复制。

主题

- 限制
- 先决条件
- 权限要求
- 设置跨集群连接
- 开始复制
- 确认复制
- 暂停和恢复复制
- 停止复制
- 自动关注
- 升级已连接的域

限制

跨集群复制具有以下限制：

- 您无法在亚马逊 OpenSearch 服务域和自管集群 OpenSearch 或 Elasticsearch 集群之间复制数据。
- 您无法将索引从一个关注者域复制到另一个关注者域。如果要将索引复制到多个关注者域，则只能从单个领导者域中进行复制。
- 一个域可以通过入站和出站连接的组合连接到最多 20 个其他域。
- 最初设置跨集群连接时，领导者域的版本必须与关注者域相同或更高。
- 您不能使用 Amazon CloudFormation 来连接域名。
- 不能在 M3 和可突增（T2 和 T3）实例上使用跨集群复制。
- 您不能在 UltraWarm 或冷索引之间复制数据。这两个索引都必须位于热存储中。
- 删除领导者域的索引时，不会自动删除关注者域中对应的索引。

先决条件

在设置跨集群复制之前，请确保域满足以下要求：

- Elasticsearch 7.10 或 1.1 或 OpenSearch 更高版本
- 已启用精细访问控制
- Node-to-node 已启用@@ 加密

权限要求

为了开始复制，必须包括对远程（领导者）域的 `es:ESCrossClusterGet` 权限。我们建议对远程域采取以下 IAM policy。此策略还允许您执行其他操作，例如编制文档索引和执行标准搜索：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:account:domain/leader-domain/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/leader-domain"
    }
  ]
}
```

请确保对 `/leader-domain`，而不是 `/leader-domain/*` 应用了 `es:ESCrossClusterGet` 权限。

为了让非管理员用户执行复制活动，还需要将他们映射到适当的权限。大多数权限对应于特定的 REST API 操作。例如，`indices:admin/plugins/replication/index/_resume` 权限可让您恢复索引的复制。有关权限的完整列表，请参阅 OpenSearch 文档中的复制权限。

> **Note**
>
> 开始复制和创建复制规则的命令是特殊情况。由于它们在领导域和关注者域上调用后台进程，因此您必须在请求`follower_cluster_role`中传递`leader_cluster_role`和。OpenSearch 服务在所有后端复制任务中都使用这些角色。有关映射和使用这些角色的信息，请参阅 OpenSearch文档中的映射领导者和关注者集群角色。

设置跨集群连接

要将索引从一个域复制到另一个域，您需要在域之间建立跨集群连接。连接域最简单的方法是通过域控制面板的 Connections（连接）选项卡。还可使用配置 API 或 Amazon CLI。由于跨集群复制遵循“拉取”模型，因此您可从关注者域启动连接。

> **Note**
>
> 如果您之前连接两个域来执行跨集群搜索，则无法使用同一连接进行复制。连接在控制台中将标记为 `SEARCH_ONLY`。为了在以前连接的两个域之间执行复制，您必须先删除该连接，然后再重新创建它。完成此操作后，该连接可用于跨集群搜索和跨集群复制。

设置连接

1. 在 Amazon Ser OpenSearch vice 控制台中，选择关注者域，前往 “连接” 选项卡，然后选择 “请求”。
2. 对于 Connection alias（连接别名），输入您的连接的名称。
3. 在连接您 Amazon Web Services 账户 和地区的域名之间进行选择，也可以选择连接到其他账户或地区的域名。
   - 要连接到您 Amazon Web Services 账户 和地区的域名，请选择该域并选择请求。
   - 要连接到其他域 Amazon Web Services 账户 或区域中的域，请指定远程域的 ARN，然后选择请求。

OpenSearch 服务验证连接请求。如果域不兼容，则连接失败。如果验证成功，它将发送到目标域进行批准。目标域批准请求后，您可以开始复制。

跨集群复制支持双向复制。这意味着您可以创建从 A 域到 B 域的出站连接，以及从 B 域到 A 域的另一个出站连接。然后，您可以设置复制，使 A 域遵循 B 域中的索引，使 B 域遵循 A 域中的索引。

开始复制

建立跨集群连接后，您可以开始复制数据。首先，在领导者域中创建要复制的索引：

```
PUT leader-01
```

要复制该索引，请将以下命令发送到关注者域：

```
PUT _plugins/_replication/follower-01/_start
{
   "leader_alias": "connection-alias",
   "leader_index": "leader-01",
   "use_roles":{
      "leader_cluster_role": "all_access",
      "follower_cluster_role": "all_access"
   }
}
```

您可以在域控制面板的连接选项卡上找到连接别名。

为简单起见，本例假设管理员正在发出请求并对 `leader_cluster_role` 和 `follower_cluster_role` 使用 `all_access`。但是，在生产环境中，我们建议您在领导者和关注者索引上创建复制用户，并相应地进行映射。用户名必须完全相同。有关这些角色以及如何映射它们的信息，请参阅 OpenSearch文档中的映射领导者和关注者集群角色。

确认复制

要确认复制正在进行，请获取复制状态：

```
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
  }
```

```
}
```

领导者和关注者检查点值以负整数开始，反映您拥有的分片数量（-1 表示一个分片，-5 表示 5 个分片，依此类推）。随着每次进行更改，这些值会递增为正整数。如果值相同，则意味着索引已完全同步。您可以使用这些检查点值来度量域之间的复制延迟。

要进一步验证复制，请将文档添加到领导者索引：

```
PUT leader-01/_doc/1
{
   "Doctor Sleep":"Stephen King"
}
```

然后确认其在关注者索引上显示：

```
GET follower-01/_search

{
    ...
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "follower-01",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "Doctor Sleep" : "Stephen King"
        }
      }
    ]
  }
}
```

暂停和恢复复制

如果需要修复问题或减少领导域的负载，可以暂时暂停复制。将此请求发送到关注者域。确保包含空的请求体：

```
POST _plugins/_replication/follower-01/_pause
{}
```

然后获取状态以确保复制已暂停：

```
GET _plugins/_replication/follower-01/_status

{
  "status" : "PAUSED",
  "reason" : "User initiated",
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01"
}
```

完成更改后，恢复复制。将此请求发送到关注者域。确保包含空的请求体：

```
POST _plugins/_replication/follower-01/_resume
{}
```

无法在复制暂停超过 12 个小时后恢复复制。您必须停止复制，删除从索引，然后重新启动主项的复制。

停止复制

完全停止复制后，关注者索引会取消关注领导者并成为标准索引。停止复制后，您无法重新启动复制。

停止从关注者域进行复制。确保包含空的请求体：

```
POST _plugins/_replication/follower-01/_stop
{}
```

自动关注

您可以针对单个领导域定义一组复制规则，这些规则会自动复制匹配指定模式的索引。当领导域上的索引与其中一个模式相匹配时（例如，`books*`），则会在关注者域上创建匹配的关注者索引。OpenSearch Service 会复制与该模式匹配的所有现有索引以及您创建的新索引。它不会复制关注者域中已存在的索引。

要复制所有索引（系统创建的索引以及关注者域中已存在的索引除外），请使用通配符（`*`）模式。

创建复制规则

在关注者域上创建复制规则并指定跨集群连接的名称：

```
POST _plugins/_replication/_autofollow
{
   "leader_alias" : "connection-alias",
   "name": "rule-name",
   "pattern": "books*",
   "use_roles":{
      "leader_cluster_role": "all_access",
      "follower_cluster_role": "all_access"
   }
}
```

您可以在域控制面板的连接选项卡上找到连接别名。

为简单起见，本例假设管理员正在发出请求并使用 `all_access` 作为领导者和关注者域角色。但是，在生产环境中，我们建议您在领导者和关注者索引上创建复制用户，并相应地进行映射。用户名必须完全相同。有关这些角色以及如何映射它们的信息，请参阅 OpenSearch文档中的映射领导者和关注者集群角色。

要检索域中现有复制规则的列表，请使用自动关注统计数据 API 操作。

要测试规则，请创建一个与领导者域模式匹配的索引：

```
PUT books-are-fun
```

然后检查其副本是否出现在关注者域中：

```
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
```

删除复制规则

当您删除复制规则时，S OpenSearch ervice 会停止复制与该模式匹配的新索引，但会继续现有的复制活动，直到您停止复制这些索引。

从关注者域中删除复制规则：

```
DELETE _plugins/_replication/_autofollow
```

```
{
   "leader_alias" : "connection-alias",
   "name": "rule-name"
}
```

升级已连接的域

要升级具有跨集群连接的两个域的引擎版本，请先升级关注者域，然后再升级领导者域。不要删除两者之间的连接，否则复制会暂停，您将无法恢复。

使用远程重新索引迁移亚马逊 OpenSearch 服务索引

远程重新索引允许您将索引从一个 Amazon S OpenSearch ervice 域复制到另一个域中。您可以从任何 OpenSearch 服务域或自管理集群 OpenSearch 和 Elasticsearch 集群迁移索引。

远程域和索引是指数据源，或者是指要从中复制数据的域和索引。本地域和索引是指数据目标，或者是指要将数据复制到的域和索引。

远程重新索引需要在本地域上 OpenSearch 使用 1.0 或更高版本，或者 Elasticsearch 6.7 或更高版本。远程域的主要版本必须低于本地域或与其相同。Elasticsearch 版本被认为低于 OpenSearch版本，这意味着您可以将数据从 Elasticsearch 域名重新索引到域名。 OpenSearch 在同一主要版本中，远程域可以是任何次要版本。例如，支持从 Elasticsearch 7.10.x 到 7.9 的远程重新索引，但不支持 OpenSearch 1.0 到 Elasticsearch 7.10.x。

> **Note**
>
> 本文档介绍如何在亚马逊 OpenSearch 服务域之间重新索引数据。有关该`reindex`操作的完整文档，包括详细步骤和支持的选项，请参阅文档中的 Reind ex OpenSearch 文档。

主题

- 先决条件
- 在 OpenSearch 服务互联网域之间重新索引数据
- 当远程服务器位于 VPC 中时，在 OpenSearch 服务域之间重新编制数据索引
- 在非OpenSearch 服务域之间重新索引数据
- 重新索引大型数据集
- 远程重新索引设置

先决条件

远程重新索引具有以下要求：

- 远程域必须可从本地域访问。对于驻留在某个 VPC 中的远程域，本地域必须具有访问该 VPC 的权限。此流程因网络配置而异，但可能涉及连接到 VPN 或托管网络或使用本机 VPC 端点连接。要了解更多信息，请参阅the section called “VPC 支持”。
- 请求必须如任何其他 REST 请求一样由远程域授权。如果远程域启用了精细访问控制，则您必须具有对远程域执行重新索引并读取本地域上的索引的权限。有关更多安全注意事项，请参阅 the section called “精细访问控制”。
- 建议您在开始重新索引流程之前，在本地域中使用所需设置创建索引。
- 如果您的域为数据节点使用 T2 或 T3 实例类型，则无法使用远程重新索引。

在 OpenSearch 服务互联网域之间重新索引数据

最基本的场景是，远程索引与您的本地域 Amazon Web Services 区域 相同，具有可公开访问的终端节点，并且您已签署 IAM 证书。

在远程域中，指定要从中重新索引的远程索引和要重新索引的本地索引：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

您必须在远程域端点的末尾添加 443 以进行验证检查。

要验证索引是否已复制到本地域，请将此请求发送到本地域：

```
GET local_index/_search
```

如果远程索引位于与本地域不同的区域中，请传入其区域名称，例如在此示例请求中：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "region": "eu-west-1"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

对于像 Amazon GovCloud (US) 中国区域这样的隔离区域，则可能无法访问终端节点，因为这些区域无法识别您的 IAM 用户。

如果使用基本身份验证来保护远程域，请指定用户名和密码：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

当远程服务器位于 VPC 中时，在 OpenSearch 服务域之间重新编制数据索引

每个 OpenSearch 服务域都由其自己的内部虚拟私有云 (VPC) 基础设施组成。在现有 OpenSearch 服务 VPC 中创建新域时，将为 VPC 中的每个数据节点创建一个弹性网络接口。

由于远程重新索引操作是从远程 OpenSearch 服务域执行的，因此是在其自己的私有 VPC 内执行的，因此您需要一种访问本地域的 VPC 的方法。为此，您可以使用内置的 VPC 终端节点连接功能建立连接 Amazon PrivateLink，也可以配置代理。

如果您的本地域使用 OpenSearch 版本 1.0 或更高版本，则可以使用控制台或创建 Amazon PrivateLink 连接。 Amazon CLI Amazon PrivateLink 连接允许本地 VPC 中的资源私下连接到同一 VPC 中的远程 VPC 中的资源 Amazon Web Services 区域。

要创建 VPC 终端节点连接，要重新编制索引的源域必须位于本地 VPC 中，并且源域和目标域必须位于同一个 VPC 中。 Amazon Web Services 区域

使用重新索引数据 Amazon Web Services Management Console

您可以通过控制台使用远程重新索引，在共享 VPC 端点连接的两个域之间复制索引。

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.amazonaws.cn/aos/。
2. 在左侧导航窗格中，选择 域。
3. 选择本地域，或您希望将数据复制到的域。随即打开域详细信息页面。选择常规信息下方的连接选项卡，然后选择请求。
4. 在请求连接页面上，选择 VPC 端点连接为连接模式，然后输入其他相关详细信息。这些详细信息包括远程域，即您希望从中复制数据的域。然后选择请求。
5. 导航到远程域的详细信息页面，选择连接选项卡，找到入站连接表。选中刚刚从中创建连接的域（本地域）名称旁边的复选框。选择 Approve（批准）。
6. 导航回本地域，选择 Connections（连接）选项卡，找到 Outbound connections（出站连接）表。两个域之间的连接处于活动状态后，表中的 Endpoint（端点）列中将出现一个端点。复制端点。
7. 打开本地域的控制面板，在左侧导航栏中选择 Dev Tools（开发人员工具）。要确认本地域中尚不存在远程域索引，请运行以下 GET 请求。*remote-domain-index-name*用您自己的索引名称替换。

```
GET remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

在输出中，您将看到指示“未找到索引”的错误。

8. 在 GET 请求下方，创建一个 POST 请求并使用您的端点作为远程主机，如下所示。

```
POST _reindex
{
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

   运行此请求。

9. 再次运行 GET 请求。现在，输出中会显示“存在本地索引”。您可以查询此索引以验证是否 OpenSearch复制了远程索引中的所有数据。

使用 OpenSearch 服务 API 操作为数据重新编制索引

您可以通过 API 使用远程重新索引，在共享 VPC 端点连接的两个域之间复制索引。

1. 使用 CreateOutboundConnectionAPI 操作请求从本地域到远程域的新连接。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/cc/outboundConnection

{
   "ConnectionAlias": "remote-reindex-example",
   "ConnectionMode": "VPC_ENDPOINT",
   "LocalDomainInfo": {
      "AWSDomainInformation": {
         "DomainName": "local-domain-name",
         "OwnerId": "aws-account-id",
         "Region": "region"
      }
   },
   "RemoteDomainInfo": {
      "AWSDomainInformation": {
```

```
            "DomainName": "remote-domain-name",
            "OwnerId": "aws-account-id",
            "Region": "region"
        }
    }
}
```

您会在回复中收到一个 `ConnectionId`。保存此 ID，以便在下一步中使用。

2. 使用带有您的连接 ID 的 AcceptInboundConnectionAPI 操作来批准来自本地域的请求。

```
PUT https://es.region.amazonaws.com/2021-01-01/opensearch/cc/
inboundConnection/ConnectionId/accept
```

3. 使用 DescribeOutboundConnectionsAPI 操作检索远程域的终端节点。

```
{
    "Connections": [
        {
            "ConnectionAlias": "remote-reindex-example",
            "ConnectionId": "connection-id",
            "ConnectionMode": "VPC_ENDPOINT",
            "ConnectionProperties": {
                "Endpoint": "connection-endpoint"
            },
            ...
        }
    ]
}
```

保存*connection-endpoint*以在步骤 5 中使用。

4. 要确认本地域中尚不存在远程域索引，请运行以下 GET 请求。*remote-domain-index-name*用您自己的索引名称替换。

```
GET local-domain-endpoint/remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

在输出中，您将看到指示“未找到索引”的错误。

5. 创建一个 POST 请求并使用您的端点作为远程主机，如下所示。

```
POST local-domain-endpoint/_reindex
{
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

运行此请求。

6. 再次运行 GET 请求。现在，输出中会显示“存在本地索引”。您可以查询此索引以验证是否OpenSearch复制了远程索引中的所有数据。

如果远程域托管在 VPC 内并且您不想使用 VPN 端点连接功能，则必须使用可公开访问的端点配置代理。在这种情况下， OpenSearch 服务需要公共终端节点，因为它无法将流量发送到您的 VPC。

当您在 VPC 模式下运行域时，将在您的 VPC 中放置一个或多个端点。但是，这些端点仅适用于进入 VPC 内域的流量，而不允许进入 VPC 本身的流量。

远程重新索引命令是从本地域运行的，因此原始流量无法使用这些端点访问远程域。这就是此用例中需要代理的原因。代理域必须具有由公共证书颁发机构 (CA) 签名的证书。不支持自签名或私有 CA 签名证书。

在非OpenSearch 服务域之间重新索引数据

如果远程索引托管在 S OpenSearch ervice 之外，例如在自管理 EC2 实例中，请`true`将`external`参数设置为：

```
POST _reindex
{
```

```
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password",
      "external": true
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

在这种情况下，仅支持使用用户名和密码的基本身份验证。远程域必须具有可公开访问的终端节点（即使它与本地 OpenSearch 服务域位于同一 VPC 中）和由公有 CA 签名的证书。不支持自签名或私有 CA 签名证书。

重新索引大型数据集

远程重新索引将滚动请求发送到具有以下默认值的远程域：

- 5 分钟的搜索上下文
- 套接字超时 30 秒
- 1,000 的批处理大小

建议调整这些参数以适应您的数据。对于大型文档，可以考虑批量较小的超时 and/or 时间更长。有关更多信息，请参阅对结果进行分页。

```
POST _reindex?pretty=true&scroll=10h&wait_for_completion=false
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "socket_timeout": "60m"
    },
    "size": 100,
    "index": "remote_index"
  },
  "dest": {
```

```
    "index": "local_index"
  }
}
```

我们还建议将以下设置添加到本地索引以获得更好的性能：

```
PUT local_index
{
  "settings": {
    "refresh_interval": -1,
    "number_of_replicas": 0
  }
}
```

重新索引流程完成后，您可以设置所需的副本计数并删除刷新间隔设置。

要通过查询仅重新索引选择的文档子集，请将此请求发送到本地域：

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index",
    "query": {
      "match": {
        "field_name": "text"
      }
    }
  },
  "dest": {
    "index": "local_index"
  }
}
```

远程重新索引不支持切片，因此您不能并行对同一请求执行多个滚动操作。

远程重新索引设置

除了标准的重新索引选项外，S OpenSearch ervice 还支持以下选项：

选项	有效值	描述	必填
外部	布尔值	如果远程域不是 OpenSearch 服务域，或者您要在两个 VPC 域之间重新建立索引，请指定为`true`。	否
区域	字符串	如果远程域位于不同的区域中，请指定该区域名称。	否

使用数据流管理 Amazon Ser OpenSearch vice 中的时间序列数据

管理时间序列数据的典型工作流涉及多个步骤，例如创建翻转索引别名、定义写入索引以及为后备索引定义常见映射和设置。

Amazon S OpenSearch ervice 中的数据流有助于简化这一初始设置过程。对于基于时间的数据（例如通常仅追加的应用程序日志），数据流即可开箱即用。

数据流需要 1.0 或更高 OpenSearch 版本。

> **Note**
>
> 本文档介绍了一些有助您开始在 Amazon S OpenSearch ervice 域上使用数据流的基本步骤。有关全面的文档，请参阅 OpenSearch 文档中的 Data streams。

数据流入门

数据流在内部由多个支持索引组成。搜索请求被路由到所有后备索引，而索引请求路由到最新的写入索引。

步骤 1：创建索引模板

要创建数据流，首先需要创建一个索引模板，该模板将一组索引配置为数据流。这些区域有：`data_stream` 对象表示它是数据流而不是常规索引模板。索引模式与数据流的名称匹配：

```
PUT _index_template/logs-template
{
  "index_patterns": [
    "my-data-stream",
    "logs-*"
  ],
  "data_stream": {},
  "priority": 100
}
```

在这种情况下，每个引入的文档都必须具有 `@timestamp` 字段。您还可以将自己的自定义时间戳字段定义为 `data_stream` 对象中的属性：

```
PUT _index_template/logs-template
{
  "index_patterns": "my-data-stream",
  "data_stream": {
    "timestamp_field": {
      "name": "request_time"
    }
  }
}
```

步骤 2：创建数据流

创建索引模板后，您可以直接开始接收数据，而无需创建数据流。

因为我们有一个匹配的索引模板`data_stream`，因此 OpenSearch 会自动创建数据流：

```
POST logs-staging/_doc
{
  "message": "login attempt failed",
  "@timestamp": "2013-03-01T00:00:00"
}
```

步骤 3：将数据引入到数据流

要将数据引入到数据流中，您可以使用常规索引 APIs。确保您索引的每个文档都有一个时间戳字段。如果您尝试引入没有时间戳字段的文档，则会收到错误。

```
POST logs-redis/_doc
```

```
{
  "message": "login attempt",
  "@timestamp": "2013-03-01T00:00:00"
}
```

步骤 4：搜索数据流

您可以搜索数据流，就像搜索常规索引或索引别名一样。搜索操作适用于所有后备索引（流中存在的所有数据）。

```
GET logs-redis/_search
{
  "query": {
    "match": {
      "message": "login"
    }
  }
}
```

步骤 5：将鼠标移动到数据流

您可以设置索引状态管理 (ISM)策略来自动执行数据流的转换过程。在创建支持索引时，ISM 策略将应用于支持索引。将策略与数据流关联时，它仅影响该数据流的未来支持索引。您也不需要提供 `rollover_alias` 设置，因为 ISM 策略从后备索引推断出此信息。

> **Note**
>
> 如果将后备索引迁移至冷存储，则会从数据流中 OpenSearch 删除此索引。即使您将索引移回 UltraWarm，则索引保持独立，而不是原始数据流的一部分。从数据流中删除索引后，对数据流执行搜索不会返回索引中的任何数据。

> **Warning**
>
> 数据流写入索引无法迁移到冷存储。如果要将数据流数据迁移到冷存储，必须在迁移前滚动数据流。

步骤 6：管理控制 OpenSearch 面板中的数据流

要管理 OpenSearch 控制面板中的数据流，请打开OpenSearch控制面板，选择索引管理，选择索引或策略管理的索引。

步骤 7：删除数据流

删除操作首先删除数据流的支持索引，然后删除数据流本身。

要删除数据流及其所有隐藏的后备索引，请执行以下操作：

```
DELETE _data_stream/name_of_data_stream
```

监控 Amazon S OpenSearch ervice 中的数据

通过警报和异常检测功能，主动监控 Amazon Ser OpenSearch vice 中的数据。设置警报以在数据超过特定阈值时接收通知。异常检测使用机器学习自动检测流式数据中的任何异常值。您可以将异常检测与警报配对使用，以确保在检测到异常后立即通知您。

主题

- 在 Amazon OpenSearch 服务中配置提醒
- Service OpenSearch 中的异常检测

在 Amazon OpenSearch 服务中配置提醒

在 Amazon S OpenSearch ervice 中配置警报，以便在一个或多个索引中的数据满足特定条件时收到通知。例如，如果您的应用程序在一小时内记录了超过五个 HTTP 503 错误，您可能希望收到一封电子邮件；或者，如果在过去 20 分钟内未对任何新文档编制索引，您可能希望呼叫开发人员。

警报需要 OpenSearch 或 Elasticsearch 6.2 或更高版本。

> **Note**
>
> 本文档简要概述了提醒，并重点介绍了 Amazon Ser OpenSearch vice 域上的提醒与开源集群上的警报有何不同。 OpenSearch 有关完整的警报文档，包括全面的 API 参考、复合监视器的可用请求字段列表以及可用触发器和操作变量的描述，请参阅文档中的 OpenSearch 警报。

主题

- 提醒权限
- 开始使用警报
- 通知
- 差异

提醒权限

提醒支持访问权限的精细控制。有关混合和匹配权限以适应您的用例的详细信息，请参阅 OpenSearch 文档中的警报安全性。

要访问 OpenSearch 仪表板中的警报页面，您必须至少映射到`alerting_read_access`预定义的角色，或者被授予同等权限。该角色会授予查看提醒、目标和监控器的权限，但不会授予确认提醒或修改目标或监控器的权限。

开始使用警报

要创建警报，您需要配置监视器，这是一项按定义的计划运行并查询 OpenSearch 索引的作业。您还需要配置一个或多个触发器，这些触发器定义生成事件的条件。最后，您需要配置操作，即触发警报后进行的操作。

开始使用警报

1. 从 “ OpenSearch 控制面板” 主菜单中选择 “警报”，然后选择 “创建监视器”。
2. 创建每个查询、每个桶、每个集群的指标，或每个文档的监视器。有关说明，请参阅 Create a monitor（创建监视器）。
3. 对于 Triggers（触发器），创建一个或多个触发器。有关说明，请参阅 Create triggers（创建触发器）。
4. 对于 Actions（操作），为警报设置通知通道。在 Slack、Amazon Chime、自定义 webhook 或 Amazon SNS 之间进行选择。正如您所想象的那样，通知需要连接到通道。例如，要通知 Slack 频道或向第三方服务器发送自定义 webhook，您的 OpenSearch 服务域需要适当的网络连接。自定义 webhook 必须具有公有 IP 地址， OpenSearch 服务域才能向其发送警报。

> **Tip**
>
> 操作成功发送消息后，您负责确保对该消息的访问权限（例如，访问 Slack 频道）。如果您的域包含敏感数据，请考虑使用触发器而不采取任何操作，并定期检查仪表板中的警报。

通知

警报与通知集成，后者是一个统一的 OpenSearch通知系统。通知允许您配置要使用的通信服务，并查看相关的统计数据和故障排除信息。有关全面的文档，请参阅 OpenSearch 文档中的通知。

您的域名必须运行 OpenSearch 版本 2.3 或更高版本才能使用通知。

> **Note**
>
> OpenSearch 通知与 OpenSearch 服务通知是分开的，后者提供有关服务软件更新、Auto-Tune 增强功能的详细信息以及其他重要的域级信息。 OpenSearch 通知是特定于插件的。

从 2.0 OpenSearch 版开始，通知渠道取代了警报目的地。目标已被正式停用，今后所有警报通知都将通过通道进行管理。

当您将域名升级到版本 2.3 或更高版本时（由于 2.x 的 OpenSearch 服务支持从 2.3 开始），您的现有目的地将自动迁移到通知渠道。如果目标迁移失败，监视器将继续使用该目标，直到将监视器迁移到通知通道为止。有关更多信息，请参阅 OpenSearch 文档中有关目的地的问题。

要开始使用通知，请登录 OpenSearch 控制面板并选择 “通知”、“频道” 和 “创建频道”。

Amazon Simple Notification Service （Amazon SNS）是一种支持的通知渠道类型。要对用户进行身份验证，您要么需要为用户提供针对 Amazon SNS 的完全访问权限，要么让他们担任有权访问 Amazon SNS 的 IAM 角色。有关说明，请参阅 Amazon SNS as a channel type（Amazon SNS 作为通道类型）。

差异

与的开源版本相比 OpenSearch，Amazon Serv OpenSearch ice 中的警报有一些显著的区别。

警报设置

OpenSearch 服务允许您修改以下警报设置：

- `plugins.scheduled_jobs.enabled`
- `plugins.alerting.alert_history_enabled`
- `plugins.alerting.alert_history_max_age`
- `plugins.alerting.alert_history_max_docs`
- `plugins.alerting.alert_history_retention_period`
- `plugins.alerting.alert_history_rollover_period`
- `plugins.alerting.filter_by_backend_roles`

所有其他设置都使用您无法更改的默认值。

要禁用提醒，请发送以下请求：

```
PUT _cluster/settings
{
  "persistent" : {
    "plugins.scheduled_jobs.enabled" : false
  }
}
```

以下请求将提醒配置为在七天后自动删除历史记录索引，而不是默认的 30 天：

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.alerting.alert_history_retention_period": "7d"
  }
}
```

如果您之前创建了监控器，并且想停止创建每日提醒索引，请删除所有提醒历史记录索引：

```
DELETE .plugins-alerting-alert-history-*
```

要减少历史记录索引的分片计数，请创建索引模板。以下请求会将警报的历史索引设置为一个分片和一个副本：

```
PUT _index_template/template-name
{
  "index_patterns": [".opendistro-alerting-alert-history-*"],
  "template": {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 1
    }
  }
}
```

根据您对数据丢失的容忍度，您甚至可以考虑使用零副本。有关创建和管理索引模板的更多信息，请参阅 OpenSearch 文档中的索引模板。

Service OpenSearch 中的异常检测

Amazon OpenSearch S OpenSearch ervice 中的异常检测通过使用 Random Cut OpenSearch Forest (RCF) 算法，几乎实时地自动检测 OpenSearch 文档中的 Anomaly RCF 是一种无监督的机器学习算法，可对传入数据流的草图进行建模。该算法用于计算每个传入数据点的 `anomaly grade` 和 `confidence score` 值。异常检测使用这些值来区分数据中的异常与正常变化。

您可以将异常检测插件与提醒插件配合使用，以在检测到异常后立即通知您。

异常检测可用于运行任何 OpenSearch 7.4 或 OpenSearch 更高版本。除了 `t2.micro` 和 `t2.small` 以外的所有实例类型都支持异常检测。

> **Note**
>
> 本文档简要概述了在 Amazon OpenSearch S OpenSearch ervice 背景下的异常检测。如需全面的文档，包括详细步骤、API 参考，以及创建可视化效果和控制面板的步骤，请参阅开源 OpenSearch OpenSearch 文档中的 Anomaly det ection。

先决条件

异常检测具有以下先决条件：

- 异常检测需要 OpenSearch 7.4 OpenSearch 或更高版本。
- 异常检测仅支持在 Elasticsearch 版本 7.9 和更高版本以及所有版本的 OpenSearch 中的访问权限的精细控制。 OpenSearch在 Elasticsearch 7.9 之前，只有管理员用户可以创建、查看和管理检测器。
- 如果域使用精细访问控制，则非管理员用户必须映射到 OpenSearch 控制 OpenSearch 面板的 cold_manager `anomaly_read_access` 角色，`anomaly_full_access`以管理冷索引。

异常检测入门

要开始使用该功能，请选择 OpenSearch OpenSearch 控制面板中的异常检测

步骤 1：创建检测器

检测器即单个异常检测任务。您可以创建多个检测器，并且所有检测器可以同时运行，每个检测器分析来自不同来源的数据。

步骤 2：向检测器添加特征

特征就是检查是否存在异常的索引中的字段。检测器可以发现跨一个或多个特征的异常。必须为每个特征选择以下聚合之一：`average()`、`sum()`、`count()`、`min()` 或 `max()`。

> **Note**
>
> 异常检测仅`count()`支持在 Elasticsearch 7.7 或更高版本中 OpenSearch可用。对于 Elasticsearch 7.4，请使用如下所示的自定义表达式：
>
> ```
> {
>  "aggregation_name": {
>  "value_count": {
>  "field": "field_name"
>  }
>  }
> }
> ```

聚合方法决定了是什么构成异常。例如，如果选择 `min()`，则检测器将着重根据特征的最小值查找异常。如果选择 `average()`，则检测器将根据特征的平均值查找异常。每个检测器最多可以添加五个特征。

您可以配置以下可选设置（在 Elasticsearch 7.7 及更高版本中可用）：

- 类别字段-使用 IP 地址、产品 ID、国家/地区代码等维度对数据进行分类或切片。
- 窗口大小-设置要在检测窗口中考虑的数据流聚合间隔的数量。

设置要素后，预览样本异常并根据需要调整功能设置。

步骤 3：观察结果

cpu_ad
Running since 11/13/20 10:04 AM
Actions
Stop detector
Anomaly results
Detector configuration
Live anomalies
Live
View anomaly results during the last 60 intervals (60 minutes).
View full screen
Anomaly grade
Detector interval
1 minute
Latest anomaly grade
0.49
Latest confidence
0.97
now
Anomaly history
last 7 days
Show dates
Refresh
Set up alerts
Choose a filled rectangle in the heat map for a more detailed view of anomalies within that entity.
host
Top 10
By severity
Anomaly grade
0.0 (None)
(Critical) 1.0
i-wbjdaHzaCb
i-Q0OIR6VMfs
i-mQjnCBTgyc
i-9lsSvylCjm
i-HCfUHCWiiw
i-fiBmhHVIfW
i-1e8dUjWYxO
i-AHAboGswTH
i-WrSNK7zgys
i-2TjPzDvHtO
Time: 11-17 21:30:53 2020
Max anomaly grade: 0.97
Anomaly occurrences: 48
11-11 22:31:25 2020
11-13 08:07:25 2020
11-14 17:43:25 2020
11-16 03:19:25 2020
11-17 12:55:25 2020
Anomaly occurrence
Feature breakdown
i-mQjnCBTgyc
Anomaly occurrences
48
Anomaly grade
0.01-0.97
Confidence
0.97-0.97
Last anomaly occurrence
11/17/20 05:05 PM
Anomaly grade / confidence
Confidence
Anomaly grade
Anomaly occurrences (48)
Start time
End time
Entity
Data confidence
Anomaly grade
11/17/20 5:04 PM
11/17/20 5:05 PM
i-mQjnCBTgyc
0.97
0.15

- Live anomalies（实时异常）- 显示了过去 60 个间隔的实时异常结果。例如，如果间隔设置为 10，则显示过去 600 分钟的结果。此图表每 30 秒刷新一次。
- Anomaly history（异常历史记录）- 图表采用相应的置信度绘制异常等级。
- Feature breakdown（特征细分） - 根据聚合方法绘制特征。您可以更改检测器的日期和时间范围。
- Anomaly occurrence（异常发生）- 显示了检测到的每个异常的 `Start time`、`End time`、`Data confidence` 和 `Anomaly grade`

  如果您设置了类别字段，您会看到一个额外的热图图表，该图表将异常实体的结果相关联。选择填充的矩形可查看更详细的异常视图。

步骤 4：设置警报

要创建监视器以便在检测到任何异常情况时向您发送通知，请选择 Set up alerts (设置警报)。插件将您重定向到 Add monitor（添加监视器）页面，您可以在其中配置警报。

教程：使用异常检测功能检测高 CPU 使用率

本教程演示如何在 Amazon OpenSearch S OpenSearch ervice 中创建异常检测器，以检测高 CPU 使用率。您将使用 OpenSearch OpenSearch 控制面板配置检测器以监控 CPU 使用率，并在 CPU 使用率超过指定阈值时生成警报。

> **Note**
>
> 这些步骤适用于最新版本的 OpenSearc OpenSearch h，与适用于过去版本的步骤可能略有不同。

先决条件

- 您的 OpenSearch S OpenSearch ervice 域必须运行 Elasticsearch 7.4 或更高版本，或者任 OpenSearch
- 您必须将应用程序日志文件提取到包含 CPU 使用率数据的集群中。

步骤 1：创建检测器

首先，创建一个检测器，用于识别 CPU 使用率数据中的异常情况。

1. 在 OpenSearch OpenSearch 控制面板中，打开左侧面板菜单，然后依次选择 Anomaly detection（创建检测
2. 将检测器命名为 **high-cpu-usage**。
3. 对于数据来源，选择包含要识别其中异常情况的 CPU 使用率日志文件的索引。
4. 在您的数据中，选择 Timestamp field（时间戳字段）。您可以选择添加数据筛选条件。此数据筛选条件仅分析数据来源的子集，并减少不相关数据产生的噪音。
5. 设置 Detector interval（检测器间隔）为 2 分钟。此间隔定义检测器收集数据的时间（按分钟间隔）。
6. 在 Window delay（窗口延迟）中，添加 1-minute 延迟。此延迟会增加额外的处理时间，以确保窗口中的所有数据都存在。
7. 选择下一步。在异常检测控制面板的检测器名称下，选择 Configure model（配置模型）。
8. 对于 Feature name（功能名称），请输入 **max_cpu_usage**。对于 Feature state（功能状态），请选择 Enable feature（启用功能）。
9. 对于 Find anomalies based on（查找异常情况的依据），请选择 Field value（字段值）。
10. 对于 Aggregation method（聚合方法），请选择 **max()**。
11. 对于 Field（字段），请选择数据中的字段，以检查是否存在异常情况。例如，其可能称为 `cpu_usage_percentage`。
12. 所有其他设置保留为默认值，然后选择 Next（下一步）。
13. 忽略检测器任务设置并选择 Next（下一步）。
14. 在弹出窗口中，选择启动检测器（自动或手动）的时间，然后选择 Confirm（确认）。

现在，检测器已配置完毕，在其初始化之后，您可以在检测器面板的 Real-time results（实时结果）部分查看 CPU 使用率的实时结果。Live anomalies（实时异常情况）部分会显示实时提取数据时发生的任何异常情况。

步骤 2：配置警报

现在，您已创建检测器，请创建监视器，该监视器会在检测到满足检测器设置中指定条件的 CPU 使用率时，调用警报以向 Slack 发送消息。当来自一个或多个索引的数据满足调用警报的条件时，您将收到 Slack 通知。

1. 在 “ OpenSearch 控制面板” 中打开左侧面板菜单并选择 “警报”，然后选择 “创建监视器”。
2. 提供监视器名称。

3. 对于 Monitor type（监视器类型），请选择 Per-query monitor（每个查询监视器）。每个查询监视器运行指定的查询并定义触发器。
4. 对于 Monitor defining method（监视器定义方法），请选择 Anomaly detector（异常检测器），然后从 Detector（检测器）下拉菜单中选择您在之前部分中创建的检测器。
5. 对于 Schedule（计划），请选择监视器收集数据的频率以及您接收警报的频率。在本教程中，请将计划设置为每 7 分钟运行。
6. 在 Triggers（触发器）部分中，选择 Add trigger（添加触发器）。对于 Trigger name（触发器名称），请输入 **`High CPU usage`**。在本教程中，对于 Severity level（严重性级别），请选择 1，这是最高的严重性级别。
7. 对于 Anomaly grade threshold（异常等级阈值），请选择 IS ABOVE（超过）。在该菜单下的菜单中，选择要应用的等级阈值。在本教程中，请将 Anomaly grade（异常等级）设置为 0.7。
8. 对于 Anomaly confidence threshold（异常置信阈值），请选择 IS ABOVE（超过）。在该菜单下的菜单中，输入与您的异常等级相同的数字。在本教程中，请将 Anomaly confidence threshold（异常置信阈值）设置为 0.7。
9. 在 Actions（操作）部分中，选择 Destination（目标）。在 Name（名称）字段中，选择目标名称。在 Type（类型）菜单中，选择 Slack。在 Webhook URL 字段中，输入要接收警报的 Webhook URL。有关更多信息，请参阅 Sending messages using incoming webhooks（使用传入 Webhook 发送消息）。

10选择创建。

相关资源

- the section called “警报”

- the section called “异常检测”
- 异常检测 API

Amazon Q 亚马逊 OpenSearch 服务开发者

Amazon Q 开发者版是一款生成式人工智能。Amazon Q 可帮助 Amazon 客户对其中运行的应用程序进行编码、测试、部署、故障排除和优化 Amazon。

Amazon Q 与亚马逊 OpenSearch 服务的集成提供了以下生成功能：

- the section called “使用自然语言生成可视化效果”
- the section called “查看警报摘要和见解”
- the section called “在 “发现” 页面上查看 Amazon Q 生成的查询结果摘要”
- the section called “描述异常检测器”
- the section called “访问 Amazon Q 聊天室咨询 OpenSearch 服务问题”

要访问 OpenSearch 服务中与您的任务相关的 Amazon Q Developer 功能，请在 “提醒”、“发现” 和 “创建可视化” 页面上查找以下上下文图标。单击该图标可提示您寻求帮助，并使用本节中描述的功能。

选择控制台右上角的刷新图标。 OpenSearch Amazon Q 聊天支持回答有关 OpenSearch 服务特性和功能的问题。

> **Note**
>
> Amazon Q 聊天无法访问您的数据。因此，您无法让聊天机器人参与有关您的数据的对话。

有关更多信息，请参阅《Amazon Q 开发人员用户指南》中的为服务。 在 Amazon Q 用户指南中。

设置。 Amazon Web Services 区域

以下 Amazon Web Services 区域是 OpenSearch 错误响应示例。

- 美国东部（弗吉尼亚州北部）
- 美国西部（俄勒冈州）

- 亚太地区（孟买）
- 亚太地区（悉尼）
- 亚太地区（东京）
- 加拿大（中部）
- 欧洲地区（法兰克福）
- 欧洲地区（伦敦）
- 欧洲（巴黎）
- 南美洲（圣保罗）

设置 Amazon Q 数据集 OpenSearch 成

完成以下步骤以 OpenSearch 设置。

1. 使用以下步骤查看控制 OpenSearch 台的访问控制。有关更多信息，请参阅 Amazon 服务中的精细访问控制 OpenSearch 。
2. 确认您的数据源使用的是 2.17 或更高 OpenSearch 版本。
3. 在 Ama zon OpenS earch OpenSearce 中配置警报。

> **Note**
>
> Q 免费套餐提供 Amazon Q 开发者版 OpenSearch 功能。有关更多信息，请参阅 Amazon Q 用户指南中的了解服务等级。

使用自然语言生成可视化效果

为了帮助您进一步了解自己的运营数据，Amazon Q OpenSearch 服务开发人员支持使用自然语言提示来构建可视化效果。您可以在 “可视化” 页面或 “发现” 页面中使用自然语言生成如下例所示的可视化效果。

可视化可以根据不同的维度细分错误分析，从而加快故障排除。可视化还可以帮助您识别模式和趋势，加快决策，揭示关系并简化复杂数据。

使用自然语言生成可视化

1. 确认您已将 Amazon Q 设置为 OpenSearch 服务。
2. 在 “ OpenSearch 仪表板” 主菜单中，选择 “发现” 页面，然后选择一个数据源。
3. 从 Amazon Q 菜单中选择 “生成可视化”，如以下屏幕截图所示。

4. 如果您在 “发现” 页面上输入了自然语言查询，则在根据该查询创建可视化时，Amazon Q 会复制上下文。如果您尚未输入自然语言查询，请在 Amazon Q 文本框中输入提示，然后单击文本框旁边的按钮来运行查询。Amazon Q 开发者版操作设置可能需要几秒钟进行传播。
5. 要更改视觉对象右上角的刷新图标。键入新提示，例如 “将其更改为折线图”，然后选择 “应用”。

查看警报摘要和见解

在 Amazon OpenSearch Ser OpenSearch vice 中配置警报。为了帮助您快速了解警报并对其进行故障排除，您可以通过单击警报旁边的 Amazon Q Developer 图标来查看警报摘要。摘要提供有关触发警报的潜在问题的详细信息，并在可用时提供其他分析，以帮助您找到问题的根本原因。以下屏幕截图显示了错误响应示例。

如果您连接知识库以提供有关您的环境的其他背景信息（如本主题后面所述），Amazon Q 将创建有关警报的见解。Insights 提供详细信息和故障排除选项，以帮助您解决警报的根本原因。对于之前显示的提醒，Amazon Q 还提供了以下见解。

> **Note**
>
> 根据警报的性质和可用信息，Amazon Q 可以让您在 OpenSearch 控制面板的 “发现” 页面上查看警报数据。如果您在 Amazon Q 提醒摘要的底部看到 “在发现中查看” 按钮，请单击该按钮在 D is cover 中打开相应的数据集，其中包含警报数据的有效筛选器。

主题

- 开始前的准备工作
- 查看警报摘要和见解

开始前的准备工作

完成以下步骤来配置 Amazon Bedrock 知识库，这样 Amazon Q 就可以为 OpenSearch 服务提醒创建见解。

步骤 1：创建 LambdaInvokeOpenSearch MLCommons

在 Amazon Identity and Access Management (IAM) `LambdaInvokeOpenSearchMLCommonsRole` 中创建名为的新角色。 OpenSearch Service 使用此角色在中创建 AI 连接器 OpenSearch，该连接器有助于根据配置的知识库文章生成见解。您必须将此角色映射到 OpenSearch 服务`ml_full_access`角色，如步骤 2 中所述。

创建新角色时，在 “可信实体类型” 中选择 “Amazon 账户”。您不需要指定配置文件。在添加权限页面上，选择下一步。有关创建 IAM 角色的更多信息 Amazon ，请参阅。

步骤 2：将 LambdaInvokeOpenSearchMLCommons角色角色映射到 OpenSearch 服务 ml_full_access 角色

使用以下步骤将`LambdaInvokeOpenSearchMLCommonsRole`角色映射到 OpenSearch 服务`ml_full_access`角色。此映射还可以帮助 OpenSearch 服务创建 AI 连接器。

要将所需的 IAM 角色映射到 OpenSearch 服务 ml_full_access 角色

1. 打开 OpenSearch 服务控制面板数据管理页面。
2. 在数据访问和用户下，选择角色。
3. 使用搜索框找到功能`ml_full_access`包。
4. 在 ml_full_access 页面上，选择映射的用户选项卡。
5. 选择地图用户。
6. 在 “后端角色” 字段中，粘贴**`LambdaInvokeOpenSearchMLCommonsRole`**角色的 Amazon 资源名称 (ARN)，然后选择 Map。

步骤 3：使用配置 OpenSearch 服务知识库 Amazon CloudFormation

使用以下步骤配置 OpenSearch 服务知识库，以 Amazon CloudFormation 便 Amazon Q 可以生成见解。

配置见解知识库

1. 使用支持的模式登录亚马逊 OpenSearch 服务控制台https://console.aws.amazon.com/aos/主页 Amazon Web Services 区域。有关更多信息，请参阅 设置。 Amazon Web Services 区域。
2. 在导航窗格中，选择集成。
3. 在 “集成模板” 部分，选择 “通过 Amazon Bedrock 与知识库集成” 模板。如果您看不到此选项。
4. 在 “通过 Amazon Bedrock 与知识库集成” 磁贴中，选择 “配置域”，然后选择一个可用选项。OpenSearch Service 会打开预先填充必填字段的 Amazon CloudFormation 堆栈模板。该 Amazon CloudFormation 堆栈支持公有域和 VPC 域集成。
5. 选择创建堆栈。 Amazon CloudFormation 创建资源后，该服务会显示 Amazon Bedrock 代理AgentIdConnectorId、和。ModelId

在适用的情况下，Amazon Q 现在可以为 OpenSearch 服务提醒创建见解。

查看警报摘要和见解

使用以下过程查看分析中的筛选 OpenSearch 条件。

查看警报摘要和见解

1. 确认您已将 Amazon Q 设置为 OpenSearch 服务。
2. 确认您已为 OpenSearch 服务设置提醒。
3. 在 “ OpenSearch 控制面板” 主菜单中，选择 “警报”，然后选择 “警报”。
4. 选择控制台右侧的。设置 10 分钟来配置此设置。
5. 如果出现在警报摘要中，请选择查看见解，根据您配置的知识库查看有关警报的更多详细信息。
6. 如果警报摘要中存在，请选择 “在发现中查看”，在 OpenSearch 仪表板的 “发现” 页面上查看警报数据。

在 “发现” 页面上查看 Amazon Q 生成的查询结果摘要

OpenSearch 服务使您能够在 “发现” 页面上使用管道处理语言 (PPL) 查询语言使用自然语言提示查询数据。例如，您可以编写以下内容：

- 我的错误日志中有错误吗？
- 每周的平均请求量是多少？
- 上周有多少请求按响应码分组？

作为响应，Amazon Q 会根据前十条记录生成查询结果的自然语言摘要，如下所示：

在对警报进行故障排除时，自然语言查询生成和查询摘要的组合可以增加一个额外的查询级别，或者一种用户友好的方式来理解数据，而无需编写复杂的查询。

在 “发现” 页面上查看 Amazon Q 生成的查询结果摘要

1. 确认您已将 Amazon Q 设置为 OpenSearch 服务。
2. 在 OpenSearch 仪表板主菜单中，选择发现。
3. 在下拉列表中，选择 “设置”。
4. 在 Amazon Q 文本框中输入提示，然后单击文本框旁边的按钮运行查询。设置 10 分钟来配置此设置。初始查询后，必须选择 “为后续摘要生成摘要”。

> **Note**
>
> 您可以从 Dis cover 的 Amazon Q 下拉列表中关闭摘要生成。

描述异常检测器

Amazon OpenSearch S OpenSearch ervice 中的异常检测通过使用 Random Cut OpenSearch Forest Forest Forest Forest Forest Forest Forest Forest RCF 是一种无监督的机器学习算法，可对传入数据流的草图进行建模。该算法用于计算每个传入数据点的 `anomaly grade` 和 `confidence score` 值。异常检测使用这些值来区分数据中的异常与正常变化。

为了简化创建异常检测器的过程，Amazon Q 可以根据您在 “发现” 页面上选择的数据源生成建议的探测器。Amazon Q 支持针对任何语言推荐的异常检测器。

查看 Amazon Q 推荐的异常检测器

1. 确认您已将 Amazon Q 设置为 OpenSearch 服务。
2. 在 “ OpenSearch 仪表板” 主菜单中，选择 “发现” 页面，然后选择一个数据源。
3. 从 Amazon Q 菜单中选择 “建议异常检测器”，如以下屏幕截图所示。

   Amazon Q 开发者版操作设置可能需要几秒钟进行传播。

4. 选择创建探测器。

访问 Amazon Q 聊天室了解 OpenSearch相关问题

如果您对亚马逊 OpenSearch 服务有疑问，包括与服务特性或功能相关的概念或程序问题，可以向 Amazon Q De OpenSearch veloper 询问信息。聊天机器人支持对话聊天，可保留您的讨论背景。它还会保存您的对话历史记录以供日后参考。选择控制台右上角的刷新图标 OpenSearch 。在文本框中键入问题，然后单击 “开始”。完成该作业最长可能需要 10 分钟来配置此设置。

Amazon Q Developer in OpenSearch
Welcome testuser to the OpenSearch Assistant!
I can help answer questions about OpenSearch.
How can I help?
The OpenSearch Assistant may produce inaccurate information. Verify all information before using it in any environment or workload.
Ask me anything...
Go

以下是错误响应示例。

Can you suggest some node management best practices?

Based on your context of being in the OpenSearch Dashboard in the us-west-2 region, here are some node management best practices for Amazon OpenSearch Service: [1]

1. Implement Dedicated Manager Nodes:
   - Use dedicated manager nodes to improve cluster stability. [2]
   - Configure three dedicated manager nodes for optimal domain stability across three Availability Zones.
   - This offloads cluster management tasks from data nodes, enhancing overall performance.
2. Multi-AZ Deployment:
   - Deploy your domain across multiple Availability Zones for increased availability and resilience.
   - Consider using the Multi-AZ with Standby option for 99.99% availability and consistent performance.

3. Proper Sizing:

> **Note**
>
> Amazon Q 聊天无法访问您的数据。因此，您无法让聊天机器人参与有关您的数据的对话。

适用于 Amazon OpenSearch 服务的机器学习

ML Commons 是一个通过传输和 REST API 调用提供一组常见的机器学习 (ML) 算法的 OpenSearch 插件。这些调用为每个 ML 请求选择正确的节点和资源，并监控 ML 任务以确保正常运行。这使您可以利用现有的开源机器学习算法，减少开发新的机器学习功能所需的工作量。有关该插件的更多信息，请参阅 OpenSearch 文档中的机器学习。本章介绍如何在 Amazon OpenSearch 服务中使用该插件。

主题

- 亚马逊 OpenSearch 服务 ML 连接器适用于 Amazon Web Services 服务
- 适用于第三方平台的 Amazon OpenSearch 服务 ML 连接器
- 用于 Amazon CloudFormation 为语义搜索设置远程推理
- 不支持的 ML Commons 设置
- OpenSearch 服务流框架模板

亚马逊 OpenSearch 服务 ML 连接器适用于 Amazon Web Services 服务

当您将 Amazon S OpenSearch ervice 机器学习 (ML) 连接器与其他连接器一起使用时 Amazon Web Services 服务，您需要设置一个 IAM 角色才能将 OpenSearch 服务安全地连接到该服务。 Amazon Web Services 服务 你可以设置一个连接器来包含 Amazon A SageMaker I 和 Amazon Bedrock。在本教程中，我们将介绍如何创建从 OpenSearch 服务到 SageMaker 运行时的连接器。有关连接器的更多信息，请参阅受支持的连接器。

主题

- 先决条件
- 创建 OpenSearch 服务连接器

先决条件

要创建连接器，您必须拥有 Amazon A SageMaker I 域终端节点和授予 OpenSearch 服务访问权限的 IAM 角色。

设置亚马逊 A SageMaker I 域名

要部署您的机器学习模型，请参阅亚马逊 SageMaker A SageMaker I 开发人员指南中的在 Ama zon AI 中部署模型。记下模型的端点 URL，这是创建 AI 连接器所必需的。

创建 IAM 角色

设置 IAM 角色以向 OpenSearch 服务委派 SageMaker 运行时权限。要创建新角色，请参阅 IAM 用户指南中的创建 IAM 角色（控制台）。或者，您可以使用现有角色，前提是该角色具有相同的权限集。如果您确实创建了新角色而不是使用 Amazon 托管角色，请将本教程`opensearch-sagemaker-role`中的角色名称替换为您自己的角色名称。

1. 将以下托管 IAM 策略附加到您的新角色，以允许 OpenSearch 服务访问您的 SageMaker AI 终端节点。要将策略附加到角色，请参阅添加 IAM 身份权限。

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sagemaker:InvokeEndpointAsync",
                "sagemaker:InvokeEndpoint"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

2. 按照修改角色信任策略中的说明编辑角色的信任关系。您必须在`Principal`语句中指定 “OpenSearch 服务”：

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "opensearchservice.amazonaws.com"
                ]
            }
        }
    ]
}
```

建议您使用`aws:SourceAccount`和`aws:SourceArn`条件密钥来限制对某个具体域的访问。`SourceAccount`是属于域名所有者的 Amazon Web Services 账户 ID，`SourceArn`是域名的 ARN。例如，您可以将以下条件块添加到信任策略：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

配置 权限

要创建连接器，您需要权限才能将 IAM 角色传递给 S OpenSearch ervice。还需要对 `es:ESHttpPost` 操作的访问权限。要授予这两个权限，请将以下策略附加到 IAM 角色，该角色的凭据用于签署请求：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
    },
    {
```

```
        "Effect": "Allow",
        "Action": "es:ESHttpPost",
        "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
      }
    ]
  }
```

如果您的用户或角色没有 `iam:PassRole` 权限传递您的角色，则在下一步骤中尝试注册存储库时，您可能会遇到授权错误：

在 OpenSearch 仪表板中映射 ML 角色（如果使用精细的访问控制）

设置连接器时，精细访问控制会引入额外的步骤。即使将 HTTP 基本身份验证用于所有其他目的，也需要将 `ml_full_access` 角色映射到具有传递 `opensearch-sagemaker-role` 的 `iam:PassRole` 权限的 IAM 角色。

1. 导航到您的 OpenSearch 服务域的 OpenSearch 仪表板插件。您可以在 OpenSearch 服务控制台的域控制面板上找到控制面板终端节点。
2. 从主菜单中选择安全、角色，然后选择 ml_full_access 角色。
3. 选择映射的用户、管理映射。
4. 在后端角色下，添加具有传递 `opensearch-sagemaker-role` 权限的角色的 ARN。

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. 选择映射并确认在映射的用户下显示的用户或角色。

创建 OpenSearch 服务连接器

要创建连接器，请向 OpenSearch 服务域终端节点发送POST请求。您可以使用 curl、示例 Python 客户端、Postman 或其他方法来发送已签名的请求。请注意，您无法在 Kibana 控制台中使用 POST 请求。 请求的格式如下：

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
```

```
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
         "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
```

如果域位于某个虚拟私有云 (VPC) 中，则必须将您的电脑连接到该 VPC，请求才能成功创建 AI 连接器。访问 VPC 因网络配置而异，但通常包括连接到 VPN 或企业网络。要检查您是否可以访问您的 OpenSearch 服务域，请在 Web 浏览器`https://your-vpc-domain.region.es.amazonaws.com`中导航到并确认收到默认 JSON 响应。

示例 Python 客户端

Python 客户端比 HTTP 请求更容易自动化，并且具有更好的可重用性。要使用 Python 客户端创建 AI 连接器，请将以下示例代码保存到 Python 文件中。客户端需要 适用于 Python (Boto3) 的 Amazon SDK、`requests` 和 `requests-aws4auth` 程序包。

```
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)
```

适用于第三方平台的 Amazon OpenSearch 服务 ML 连接器

在本教程中，我们将介绍如何创建从 Service 到 Coh OpenSearch ere 的连接器。有关连接器的更多信息，请参阅受支持的连接器。

当您将 Amazon S OpenSearch ervice 机器学习 (ML) 连接器与外部远程模型一起使用时，您需要在中存储您的特定授权凭证 Amazon Secrets Manager。这可能是 API 密钥，也可以是用户名和密码组合。这意味着你还需要创建一个 IAM 角色，允许 OpenSearch 服务访问权限从 Secrets Manager 读取。

主题

- 先决条件
- 创建 OpenSearch 服务连接器

先决条件

要使用 S OpenSearch ervice 为 Cohere 或任何外部提供商创建连接器，您必须拥有一个 IAM 角色来授予 OpenSearch 服务访问权限 Amazon Secrets Manager，您可以在其中存储您的证书。您还必须将凭证存储在 Secrets Manager 中。

创建 IAM 角色

设置一个 IAM 角色以将 Secrets Manager 权限委派给 OpenSearch 服务。您也可以使用现有 `SecretManagerReadWrite` 角色。要创建新角色，请参阅 IAM 用户指南中的创建 IAM 角色（控制台）。如果您确实创建了新角色而不是使用 Amazon 托管角色，请将本教程`opensearch-secretmanager-role`中的角色名称替换为您自己的角色名称。

1. 将以下托管 IAM 策略附加到您的新角色，以允许 OpenSearch 服务访问您的 Secrets Manager 值。要将策略附加到角色，请参阅添加 IAM 身份权限。

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

2. 按照修改角色信任策略中的说明编辑角色的信任关系。您必须在`Principal`语句中指定“OpenSearch 服务”：

   JSON

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

   建议您使用`aws:SourceAccount`和`aws:SourceArn`条件密钥来限制对特定域的访问。`SourceAccount`是属于域名所有者的 Amazon Web Services 账户 ID，`SourceArn`是域名的 ARN。例如，您可以将以下条件块添加到信任策略：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

配置 权限

要创建连接器，您需要权限才能将 IAM 角色传递给 S OpenSearch ervice。还需要对`es:ESHttpPost` 操作的访问权限。要授予这两个权限，请将以下策略附加到 IAM 角色，该角色的凭据用于签署请求：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPost",
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

如果您的用户或角色没有 `iam:PassRole` 权限传递您的角色，则在下一步骤中尝试注册存储库时，您可能会遇到授权错误：

设置 Amazon Secrets Manager

要将您的授权凭证存储在 Secrets Manager 中，请参阅 Amazon Secrets Manager 用户指南中的创建 Amazon Secrets Manager 密钥。

在 Secrets Manager 接受您的键值对作为密钥后，您会收到一个格式为 `arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3` 的 ARN。请记录此 ARN 和密钥，您在下一步创建连接器时会用到。

在 OpenSearch 仪表板中映射 ML 角色（如果使用精细的访问控制）

设置连接器时，精细访问控制会引入额外的步骤。即使将 HTTP 基本身份验证用于所有其他目的，也需要将 `ml_full_access` 角色映射到具有传递 `opensearch-sagemaker-role` 的 `iam:PassRole` 权限的 IAM 角色。

1. 导航到您的 OpenSearch 服务域的 OpenSearch 仪表板插件。您可以在 OpenSearch 服务控制台的域控制面板上找到控制面板终端节点。
2. 从主菜单中选择安全、角色，然后选择 ml_full_access 角色。

3. 选择映射的用户、管理映射。
4. 在后端角色下，添加具有传递 `opensearch-sagemaker-role` 权限的角色的 ARN。

```
arn:aws:iam::account-id:role/role-name
```

5. 选择映射并确认在映射的用户下显示的用户或角色。

创建 OpenSearch 服务连接器

要创建连接器，请向 OpenSearch 服务域终端节点发送`POST`请求。您可以使用 curl、示例 Python 客户端、Postman 或其他方法来发送已签名的请求。请注意，您无法在 Kibana 控制台中使用 `POST` 请求。 请求的格式如下：

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
    "name": "Cohere Connector: embedding",
    "description": "The connector to cohere embedding model",
    "version": 1,
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}
```

此请求的请求正文在两个方面与开源连接器请求的请求正文不同。在`credential`字段内，您可以传递允许 OpenSearch 服务从 Secrets Manager 读取的 IAM 角色的 ARN，以及针对什么秘密的 ARN。在 `headers` 字段中，您使用密钥来引用密钥以及它来自 ARN 的事实。

如果域位于某个虚拟私有云 (VPC) 中，则必须将您的电脑连接到该 VPC，请求才能成功创建 AI 连接器。访问 VPC 因网络配置而异，但通常包括连接到 VPN 或企业网络。要检查您是否可以访问您的 OpenSearch 服务域，请在 Web 浏览器`https://your-vpc-domain.region.es.amazonaws.com`中导航到并确认收到默认 JSON 响应。

示例 Python 客户端

Python 客户端比 HTTP 请求更容易自动化，并且具有更好的可重用性。要使用 Python 客户端创建 AI 连接器，请将以下示例代码保存到 Python 文件中。客户端需要 适用于 Python (Boto3) 的 Amazon SDK、`requests` 和 `requests-aws4auth` 程序包。

```
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)
```

用于 Amazon CloudFormation 为语义搜索设置远程推理

从 2.9 OpenSearch 版开始，您可以使用带有语义搜索的远程推理来托管自己的机器学习 (ML) 模型。远程推理使用 ML Commons 插件允许您在机器学习服务（例如和 Amazon SageMaker AI Amazon）上远程托管模型推理 BedRock，并使用机器学习连接器将它们连接到亚马逊 OpenSearch 服务。

为了简化远程推理的设置，Amazon S OpenSearch ervice 在控制台中提供了一个Amazon CloudFormation模板。 CloudFormation 是一 Amazon Web Services 服务 款允许您通过将基础设施视为代码来建模、配置 Amazon 和管理第三方资源的工具。

该 OpenSearch CloudFormation 模板可自动执行模型配置过程，因此您可以轻松地在 OpenSearch 服务域中创建模型，然后使用模型 ID 来摄取数据并运行神经搜索查询。

在 Serv OpenSearch ice 2.12 及更高版本中使用神经稀疏编码器时，我们建议您在本地使用分词器模型，而不是远程部署。有关更多信息，请参阅 OpenSearch 文档中的稀疏编码模型。

主题

- 先决条件
- Amazon SageMaker AI 模板
- Amazon Bedrock 模板

先决条件

要在 S OpenSearch ervice 中使用 CloudFormation 模板，请完成以下先决条件。

设置 OpenSearch 服务域

在使用 CloudFormation 模板之前，您必须设置一个版本 2.9 或更高版本的 Amazon Serv OpenSearch ice 域并启用精细访问控制。创建 OpenSearch 服务后端角色以授予 ML Commons 插件为您创建连接器的权限。

该 CloudFormation 模板使用默认名称为您创建一个 Lambda IAM 角色`LambdaInvokeOpenSearchMLCommonsRole`，如果您想选择其他名称，则可以覆盖该名称。模板创建此 IAM 角色后，您需要授予 Lambda 函数调用您的 OpenSearch 服务域的权限。为此，请按照以下步骤`ml_full_access`将名为的角色映射到您的 OpenSearch 服务后端角色：

1. 导航到您的 OpenSearch 服务域的 OpenSearch 仪表板插件。您可以在 OpenSearch 服务控制台的域控制面板上找到控制面板终端节点。
2. 从主菜单中选择安全、角色，然后选择 ml_full_access 角色。
3. 选择映射的用户、管理映射。
4. 在后端角色下，添加需要权限才能调用您的域名的 Lambda 角色的 ARN。

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. 选择映射并确认在映射的用户下显示的用户或角色。

映射角色后，导航到您的域的安全配置，然后将 Lambda IAM 角色添加到您的 OpenSearch 服务访问策略中。

启用您的 Amazon Web Services 账户权限

您 Amazon Web Services 账户 必须拥有访问 CloudFormation 和 Lambda 的权限，以及 Amazon Web Services 服务 您为模板选择的任何内容（Runtime SageMaker 或 Amazon）的权限。 BedRock

如果您使用的是 Amazon Bedrock，则还必须注册您的模型。要注册您的模型，请参阅 Amazon BedRock 用户指南中的模型访问权限。

如果您使用自己的 Amazon S3 存储桶来提供模型项目，则必须将 CloudFormation IAM 角色添加到您的 S3 访问策略中。有关更多信息，请参阅《 IAM 用户指南》中的添加和删除 IAM 身份权限。

Amazon SageMaker AI 模板

Amazon SageMaker AI CloudFormation 模板定义了多种 Amazon 资源，以便为您设置神经插件和语义搜索。

首先，使用通过 Amazon SageMaker 模板与文本嵌入模型集成，在 SageMaker 运行时中将文本嵌入模型部署为服务器。如果您不提供模型终端节点，则 CloudFormation 创建一个 IAM 角色以允许 SageMaker Runtime 从 Amazon S3 下载模型工件并将其部署到服务器。如果您提供终端节点，则 CloudFormation 会创建一个允许 Lambda 函数访问 OpenSearch 服务域的 IAM 角色，或者，如果该角色已经存在，则更新和重复使用该角色。该端点通过 ML Commons 插件为用于 ML 连接器的远程模型提供服务。

然后使用通过 Amazon Sagemaker 与稀疏编码器集成模板，来创建一个让您的域设置远程推理连接器的 Lambda 函数。在 S OpenSearch ervice 中创建连接器后，远程推理可以在 Runtime 中 SageMaker 使用远程模型运行语义搜索。该模板会将域中的模型 ID 返回给您，以便您可以开始搜索。

使用 Amazon SageMaker AI CloudFormation 模板

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。
2. 在左侧导航窗格中，选择集成。
3. 在每个 Amazon SageMaker AI 模板下，选择配置域、配置公共域。
4. 按照 CloudFormation 控制台中的提示配置堆栈并设置模型。

> **Note**
>
> OpenSearch 服务还提供了一个单独的模板来配置 VPC 域。如果您使用此模板，则需要提供 Lambda 函数的 VPC ID。

Amazon Bedrock 模板

与亚马逊 A SageMaker I CloudFormation 模板类似，Amazon Bedrock CloudFormation 模板预配置了在 OpenSearch 服务和亚马逊 Bedrock 之间创建连接器所需的 Amazon 资源。

首先，该模板创建一个 IAM 角色，允许 future 的 Lambda 函数访问您的 OpenSearch 服务域。然后，该模板会创建一个让该域使用 ML Commons 插件创建连接器的 Lambda 函数。 OpenSearch 服务创建连接器后，远程推理设置就完成了，您可以使用 Amazon Bedrock API 操作运行语义搜索。

请注意，由于 Amazon Bedrock 托管自己的机器学习模型，因此您无需将模型部署到 SageMaker 运行时。相反，该模板使用了一个为 Amazon BedroCk 预先确定的端点，并跳过了端点预置步骤。

使用 Amazon Bedrock 模板 CloudFormation

1. 在https://console.aws.amazon.com/aos/家中打开亚马逊 OpenSearch 服务控制台。

2. 在左侧导航窗格中，选择集成。
3. 在通过 Amazon Bedrock 与 Amazon Titan 文本嵌入模型集成下，选择配置域、配置公有域。
4. 按照提示设置模型。

> **Note**
>
> OpenSearch 服务还提供了一个单独的模板来配置 VPC 域。如果您使用此模板，则需要提供 Lambda 函数的 VPC ID。

此外， OpenSearch 服务还提供以下 Amazon Bedrock 模板，用于连接 Cohere 模型和 Amazon Titan 多式联运嵌入模型：

- `Integration with Cohere Embed through Amazon Bedrock`
- `Integrate with Amazon Bedrock Titan Multi-modal`

不支持的 ML Commons 设置

Amazon OpenSearch 服务不支持使用以下 ML Commons 设置：

- `plugins.ml_commons.allow_registering_model_via_url`
- `plugins.ml_commons.allow_registering_model_via_local_file`

> **Important**
>
> 在生产集群上，请勿禁用群集设置`plugins.ml_commons.only_run_on_ml_node`（不要将其设置为`false`）。禁用此保护的选项是为了促进开发，但生产集群应使用连接器。有关更多信息，请参阅 the section called “连接器适用于 Amazon Web Services 服务”。

有关 ML Commons 设置的更多信息，请参阅 ML Commons cluster settings。

OpenSearch 服务流框架模板

Amazon S OpenSearch ervice 流程框架模板允许您通过为常见用例提供模板来自动执行复杂的 OpenSearch 服务设置和预处理任务。例如，您可以使用流框架模板来自动执行机器学习设置任务。亚

马逊 OpenSearch 服务流程框架模板在 JSON 或 YAML 文档中简要描述了设置过程。这些模板描述了用于对话聊天或查询生成的自动工作流程配置、AI 连接器、工具、代理和其他组件，这些组件可为生成模型的后端 OpenSearch 服务做好准备。

Amazon OpenSearch 服务流程框架模板可以根据您的特定需求进行自定义。要查看自定义流框架模板的示例，请参阅 flow-framework。有关 OpenSearch 服务提供的模板，请参阅工作流程模板。有关全面的文档，包括详细步骤、API 参考和所有可用设置的参考，请参阅开源 OpenSearch 文档中的自动配置。

> **Note**
>
> Flow-Framework 不支持 OpenSearch 服务 2.17 的后端角色筛选。

在 OpenSearch 服务中创建 ML 连接器

亚马逊 OpenSearch 服务流程框架模板允许您使用 ml-commons 中提供的创建连接器 API 来配置和安装机器学习连接器。您可以使用机器学习连接器将 OpenSearch 服务连接到其他 Amazon 服务或第三方平台。有关这方面的更多信息，请参阅 Creating connectors for third-party ML platforms。Amazon OpenSearch 服务流程框架 API 允许您自动执行 OpenSearch 服务设置和预处理任务，并可用于创建机器学习连接器。

在 S OpenSearch ervice 中创建连接器之前，必须执行以下操作：

- 创建亚马逊 A SageMaker I 域名。
- 创建一个 IAM 角色。
- 配置传递角色权限
- 在仪表板中映射流程框架和 ml-commons 角色。 OpenSearch

有关如何为服务设置机器学习连接器的更多信息，请参阅 Amazon 服务的 Amazon OpenSearch 服务 ML 连接器。 Amazon 要了解有关在第三方平台上使用 OpenSearch 服务 ML 连接器的更多信息，请参阅适用于第三方平台的 Amazon OpenSearch 服务 ML 连接器。

通过 flow-framework 服务创建连接器

要使用连接器创建流程框架模板，您需要向 OpenSearch 服务域终端节点发送POST请求。您可以使用 cURL、示例 Python 客户端、Postman 或其他方法来发送已签名的请求。POST 请求的格式如下：

```
POST /_plugins/_flow_framework/workflow
```

```
{
  "name": "Deploy Claude Model",
  "description": "Deploy a model using a connector to Claude",
  "use_case": "PROVISION",
  "version": {
    "template": "1.0.0",
    "compatibility": [
      "2.12.0",
      "3.0.0"
    ]
  },
  "workflows": {
    "provision": {
      "nodes": [
        {
          "id": "create_claude_connector",
          "type": "create_connector",
          "user_inputs": {
            "name": "Claude Instant Runtime Connector",
            "version": "1",
            "protocol": "aws_sigv4",
            "description": "The connector to BedRock service for Claude model",
            "actions": [
              {
                "headers": {
                  "x-amz-content-sha256": "required",
                  "content-type": "application/json"
                },
                "method": "POST",
                "request_body": "{ \"prompt\":\"${parameters.prompt}\",
 \"max_tokens_to_sample\":${parameters.max_tokens_to_sample},
 \"temperature\":${parameters.temperature},  \"anthropic_version\":
\"${parameters.anthropic_version}\" }",
                "action_type": "predict",
                "url": "https://bedrock-runtime.us-west-2.amazonaws.com/model/
anthropic.claude-instant-v1/invoke"
              }
            ],
            "credential": {
                "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-
role"
             },
            "parameters": {
              "endpoint": "bedrock-runtime.us-west-2.amazonaws.com",
```

```
                "content_type": "application/json",
                "auth": "Sig_V4",
                "max_tokens_to_sample": "8000",
                "service_name": "bedrock",
                "temperature": "0.0001",
                "response_filter": "$.completion",
                "region": "us-west-2",
                "anthropic_version": "bedrock-2023-05-31"
              }
            }
          }
        ]
      }
    }
}
```

如果域位于某个虚拟私有云（Amazon VPC）中，则必须连接到该 Amazon VPC，才能确保请求能够成功创建 AI 连接器。访问 Amazon VPC 的方法因网络配置而异，但通常会涉及连接到某个 VPN 或企业网络。要检查您是否可以访问您的 OpenSearch 服务域，请在 Web 浏览器`https://`*`your-vpc-domain`*`.`*`region`*`.es.amazonaws.com`中导航到并确认收到默认 JSON 响应。（*`placeholder text`*用您自己的值替换。

示例 Python 客户端

Python 客户端比 HTTP 请求更容易自动化，并且具有更好的可重用性。要使用 Python 客户端创建 AI 连接器，请将以下示例代码保存到 Python 文件中。客户端需要适用于 Python 的Amazon SDK（Boto3）、Requests:HTTP for Humans 和 requests-aws4auth 1.2.3 软件包。

```
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)
```

预定义的工作流模板

Amazon S OpenSearch ervice 为一些常见的机器学习 (ML) 用例提供了多个工作流程模板。使用模板可以简化复杂的设置，并为语义搜索或对话搜索等用例提供多种默认值。您可以在调用 Create Workflow API 时指定工作流模板。

- 要使用 OpenSearch 服务提供的工作流模板，请将模板用例指定为`use_case`查询参数。
- 要使用自定义工作流模板，请在请求正文中提供完整的模板。有关自定义模板的示例，请参阅示例 JSON 模板或示例 YAML 模板。

模板用例

下表概述了各种可用的模板、模板的描述以及所需的参数。

模板用例	描述	必需参数
`bedrock_titan_embe`	创建并部署 Amazon Bedrock 嵌入模型（默认为 `titan-embed-text-v1` ）	`create_connector.credential.roleArn`

模板用例	描述	必需参数
dding_model_deploy		
bedrock_titan_embedding_model_deploy	创建并部署 Amazon Bedrock 多模态嵌入模型（默认为 titan-embed-text-v1 ）	create_connector.credential.roleArn
cohere_embedding_model_deploy	创建并部署 Cohere 嵌入模型（默认为 embed-english-v 3.0 ）。	create_connector.credential.roleArn , create_connector.credential.secretArn
cohere_chat_model_deploy	创建并部署 Cohere 聊天模型（默认为 Cohere Command ）。	create_connector.credential.roleArn , create_connector.credential.secretArn
open_ai_embedding_model_deploy	创建并部署 OpenAI 嵌入模型（默认为 text-embedding-ada -002 ）。	create_connector.credential.roleArn , create_connector.credential.secretArn
openai_chat_model_deploy	创建并部署 OpenAI 聊天模型（默认为 gpt-3.5-turbo ）。	create_connector.credential.roleArn , create_connector.credential.secretArn
semantic_search_with_cohere_embedding	配置语义搜索并部署 Cohere 嵌入模型。您必须提供 Cohere 模型的 API 密钥。	create_connector.credential.roleArn , create_connector.credential.secretArn

模板用例	描述	必需参数
semantic_search_with_cohere_embedding_query_enricher	配置语义搜索并部署 Cohere 嵌入模型。添加一个 query_enricher 搜索处理器，用于设置神经查询的默认模型 ID。您必须提供 Cohere 模型的 API 密钥。	create_connector.credential.roleArn ，create_connector.credential.secretArn
multimodal_search_with_bedrock_titan	部署 Amazon Bedrock 多模态模型，并配置一个具有 text_image_embedding 处理器和用于多模态搜索的 k-nn 索引的摄取管道。您必须提供 Amazon 凭证	create_connector.credential.roleArn

> **Note**
>
> 对于所有需要密钥 ARN 的模板，默认设置是将密钥名称为 “密钥” 的密钥存储在 Secr Amazon ets Manager 中。

包含预训练模型的默认模板

Amazon Ser OpenSearch vice 另外提供了两个开 OpenSearch 源服务中没有的默认工作流程模板。

模板用例	描述
semantic_search_with_local_model	配置语义搜索并部署预训练模型 (msmarco-distilbert-base-tas-b)。添加neural_query_enricher 搜索处理器，该处理器为神经查询设置默认模型 ID，并创建名为 “” my-nlp-index 的链接 k-nn 索引。
hybrid_search_with_local_model	配置混合搜索并部署预训练模型 (msmarco-distilbert-base-tas-b)。添加neural_query_enricher 搜索处理器，

模板用例	描述
	该处理器为神经查询设置默认模型 ID，并创建名为 “” my-nlp-index 的链接 k-nn 索引。

配置 权限

如果您使用版本 2.13 或更高版本创建新域，则相关权限已经具备。如果您在先前存在的 2.11 或更早版本的 OpenSearch 服务域上启用流程框架，然后升级到 2.13 或更高版本，则必须定义角色。`flow_framework_manager`必须将非管理员用户映射到此角色，才能使用精细访问控制管理域上的温索引。手动创建 `flow_framework_manager` 角色，请执行下列步骤：

1. 在 “ OpenSearch 控制面板” 中，转至 “安全”，然后选择 “权限”。
2. 选择创建操作组并配置以下组：

组名	权限
`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. 选择角色和创建角色。
4. 将该角色命名为 flow_framework_manager。
5. 对于群集权限，选择 `flow_framework_full_access` 和 `flow_framework_read_access`。
6. 对于索引，键入 `*`。

7. 对于索引权限，选择 `indices:admin/aliases/get`、`indices:admin/mappings/get` 和 `indices_monitor`。
8. 选择创建。
9. 创建该角色之后，将其映射到负责管理流框架索引的任何用户或后端角色。

Amazon S OpenSearch ervice 安全分析

安全分析是一种 OpenSearch 解决方案，用于监控组织基础设施、监控异常活动、实时检测潜在安全威胁，并对预先配置的目标触发警报。您可以通过持续评估安全规则及查看自动生成的安全调查发现来监控安全事件日志中的恶意活动。此外，安全分析还可以生成自动警报，并将其发送到指定通知通道，例如 Slack 或电子邮件。

您可以使用安全分析插件检测常见威胁， out-of-the-box并从现有的安全事件日志（例如，防火墙日志、Windows 日志和身份验证审计日志）中生成重要安全见解。域必须运行 2.5 或更高 OpenSearch 版本，才能使用安全分析。

> **Note**
>
> 本文档简要概述了适用于 Amazon S OpenSearch ervice 的安全分析。包括关键概念以及配置权限的步骤。如需全面的文档，包括设置指南、API 参考和所有可用设置的参考，请参阅 OpenSearch 文档中的 S ecurity Analytics。

安全分析组件和概念

大量工具和功能为运行安全分析奠定了基础。构成插件的主要组件包括探测器、日志类型、规则、调查发现和警报。

日志类型

OpenSearch 支持多种类型的日志，并针对每种类型提供 out-of-the-box映射。您可以在创建探测器时指定日志类型并配置时间间隔，然后安全分析将自动激活以该间隔运行的一组相关规则。

探测器

探测器可识别数据索引中针对某一日志类型的一系列网络安全威胁。您可以将探测器配置为使用自定义规则和预包装 Sigma 规则来评估系统中发生的事件。然后，探测器将根据这些事件生成安全调查发现结果。有关探测器的更多信息，请参阅 OpenSearch 文档中的创建探测器。

规则

威胁检测规则定义探测器应用于摄取日志数据以识别安全事件的条件。安全分析支持导入、创建和自定义规则以满足您的要求，同时还提供预包装开源 Sigma 规则以检测日志中的常见威胁。安全将大量规则映射到由 MITRE ATT&CK 组织维护的规模不断壮大的攻击者策略和技术知识库。您可以使用 OpenSearch 控制面板或创建和使用规则。 APIs 有关规则的更多信息，请参阅 OpenSearch 文档中的使用规则。

调查发现

当探测器匹配规则与日志事件时，将生成调查发现。每项调查发现均包含唯一的选择规则、日志类型和规则严重性组合。调查发现不一定指向系统内部即将发生的威胁，但始终隔离感兴趣的事件。有关调查发现的更多信息，请参阅 OpenSearch 文档中的使用调查发现。

警报

创建探测器时，您可以指定一个或多个触发提醒的条件。提醒是指发送到首选通道的通知，例如 Slack 或电子邮件。您可以设置在探测器匹配一个或多个规则时触发提醒，而且可以自定义通知消息。有关提醒的更多信息，请参阅 OpenSearch 文档中的使用提醒。

探索安全分析

您可以使用 OpenSearch 控制面板可视化及深入了解安全分析插件。概述视图提供了多方面的信息，例如调查发现和提醒计数、近期调查发现和提醒、频繁检测规则以及探测器列表等。您可以查看由多个可视化效果图构成的摘要视图。例如，下图显示了给定时段内不同日志类型的调查发现和提醒趋势。

在页面下方，您可以查看最新调查发现和提醒。

此外，您还可以查看所有活动探测器中最常触发的规则的分布情况。这样可以帮助您检测和调查各种日志类型的不同类型恶意活动。

最后，还可以查看已配置探测器的状态。您也可以通过此面板导航到创建探测器工作流。

要配置您的安全分析设置，请使用规则页面创建规则，然后使用这些规则在探测器页面中编写探测器。要集中查看安全分析结果，您可以使用调查发现和提醒页面。

配置 权限

如果在预先存在的 OpenSearch 服务域上启用安全分析，courity Analytics 可能无法`security_analytics_manager`在域上定义。必须将非管理员用户映射到此角色，才能使用精细访问控制管理域上的温索引。手动创建 `security_analytics_manager` 角色，请执行下列步骤：

1. 在 “ OpenSearch 控制面板” 中，转至 “安全”，然后选择 “权限”。
2. 选择创建操作组并配置以下组：

组名	权限
security_analytics_full_access	• cluster:admin/opensearch/securityanalytics/alerts/* • cluster:admin/opensearch/securityanalytics/detector/* • cluster:admin/opensearch/securityanalytics/findings/* • cluster:admin/opensearch/securityanalytics/mapping/* • cluster:admin/opensearch/securityanalytics/rule/*
security_analytics_read_access	• cluster:admin/opensearch/securityanalytics/alerts/get • cluster:admin/opensearch/securityanalytics/detector/get • cluster:admin/opensearch/securityanalytics/detector/search • cluster:admin/opensearch/securityanalytics/findings/get • cluster:admin/opensearch/securityanalytics/mapping/get • cluster:admin/opensearch/securityanalytics/mapping/view/get • cluster:admin/opensearch/securityanalytics/rule/get • cluster:admin/opensearch/securityanalytics/rule/search

3. 选择角色和创建角色。
4. 将角色命名为 security_analytics_manager。

5. 对于群集权限，选择 security_analytics_full_access 和 security_analytics_read_access。
6. 对于索引，键入 *。
7. 对于索引权限，选择 indices:admin/mapping/put 和 indices:admin/mappings/get。
8. 选择创建。
9. 创建角色之后，将其映射到将管理安全分析索引的任何用户或后端角色。

故障排除

无此类索引错误

如果您没有探测器，打开安全分析控制面板，可能会在右下角看到一条通知，通知内容为 [index_not_found_exception] no such index [.opensearch-sap-detectors-config]。您可以忽略此通知，通知会在几秒钟内消失，且创建探测器后不会再次显示。

Amazon Service OpenSearch 中的可观察性

适用于 Amazon Serv OpenSearch ice 的 Amazon OpenSearch 控制面板的原定设置安装包含了可观察性插件。借助该插件，您可以使用管道式处理语言（PPL）将数据驱动的事件可视化，以便探索、发现和查询存储的数据。 OpenSearch此插件需要 OpenSearch 1.2 或更高版本。

可观察性插件为采集和监控来自常见数据源的指标、日志和跟踪信息提供了统一的体验。在一个位置集中采集和监控数据，可实现整个基础设施的全栈 end-to-end可观察性。

> **Note**
>
> 本文档简要概述了可观测性 OpenSearch 。有关可观测性插件的全面文档（包括权限），请参阅 Observability。

每个人探索数据的过程都是不同的。如果您刚接触数据探索和可视化的创建，则建议您尝试如下所示的工作流。

使用事件分析来探索数据

例如，假设您正在 Ser OpenSearch vice 域中收集航班数据，并且想了解哪家航空公司上个月抵达匹兹堡国际机场的航班最多。您会编写下面的 PPL 查询：

```
source=opensearch_dashboards_sample_data_flights |
    stats count() by Dest, Carrier |
    where Dest = "Pittsburgh International Airport"
```

此查询将从名为 `opensearch_dashboards_sample_data_flights` 的索引中提取数据。然后使用 `stats` 命令获取总航班数据，并根据目的地机场和航空公司进行分组。最后，它使用 `where` 子句从结果中筛选出到达匹兹堡国际机场的航班。

下面是显示的上个月数据的样子：

Observability / Event analytics / Explorer

Pittsburgh Flights × + Add new

source=opensearch_dashboards_sample_data_flights | stats count() by Dest, Carrier | where Dest = "Pittsburgh International Airport"

PPL

Month to date Show dates Refresh Save

Events Visualizations

Search field na

Query fields

Carrier

count()

Dest

Selected Fields

Available Fields

Carrier	count()	Dest
BeatsWest	5	Pittsburgh International Airport
Logstash Airways	6	Pittsburgh International Airport
OpenSearch Dashboards Airlines	6	Pittsburgh International Airport
OpenSearch-Air	11	Pittsburgh International Airport

您可以在查询编辑器中选择 PPL 按钮来获取每个 PPL 命令的用法信息和示例：

by Dest, Carrier

	count()
	1
	2
	1
	1
	2
	1
	1
	4
irlines	1
	1
	1
	1

OpenSearch PPL Reference Manual

stats × Learn More

stats

Description

Using `stats` command to calculate the aggregation from search result.

The following table catalogs the aggregation functions and also indicates how the NULL/MISSING values is handled:

Function	NULL	MISSING
COUNT	Not counted	Not counted
SUM	Ignore	Ignore
AVG	Ignore	Ignore
MAX	Ignore	Ignore
MIN	Ignore	Ignore

Syntax

stats <aggregation>... [by-clause]...

下面来看一个更复杂的例子，即查询有关航班延误的信息：

```
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
```

查询中的每个命令都会影响最终输出：

- `source=opensearch_dashboards_sample_data_flights` – 从与上例相同的索引中提取数据
- `where FlightDelayMin > 0` – 从数据中筛选出发生延误的航班
- `stats sum(FlightDelayMin) as minimum_delay, count() as total_delayed by Carrier` – 获取每家航空公司的总最短延误时间和延误航班总数
- `eval avg_delay=minimum_delay / total_delayed` – 用最短延误时间除以延误航班总数，得出每家航空公司的平均延误时间
- `sort - avg_delay` – 按平均延误时间降序对结果进行排序

通过此查询，您可以确定 Dashboar OpenSearch ds Airlines 的平均延误更少。

avg_delay	Carrier	minimum_delay	total_delayed
212	Logstash Airways	4470	21
184	OpenSearch-Air	4245	23
155	BeatsWest	2025	13
153	OpenSearch Dashboards Airlines	4305	28

有关更多示例 PPL 查询，请参阅 Event analytics（事件分析）页面的 Queries and Visualizations（查询和可视化）。

创建可视化

正确查询到您感兴趣的数据后，可以将这些查询另存为可视化：

然后将这些可视化添加到操作面板来比较不同的数据片段。利用笔记本来组合可以与团队成员共享的各种可视化和代码段。

使用跟踪分析功能更深入探索

您可以利用跟踪分析功能将 OpenSearch 数据中的事件流可视化，从而识别和修复分布式应用程序中的性能问题。

Amazon S OpenSearch ervice 的跟踪分析

您可以使用可 OpenSearch 观察性插件中包含的跟踪分析，从而分析来自分布式应用程序的跟踪数据。Trace Analytics 需要 Elastics 7.9 OpenSearch 或更高版本。

在分布式应用程序中，单个操作（如用户单击按钮）可触发一系列扩展事件。例如，应用程序前端可能会调用后端服务，后端服务调用另一个服务，该服务可查询数据库，该服务处理查询并返回结果。然后，第一个后端服务向前端发送确认，这将更新 UI。

您可以使用跟踪分析来帮助您可视化此事件流并识别性能问题。

> **Note**
>
> 本文档简要概述了跟踪分析。有关全面的文档，请参阅开源 OpenSearch 文档中的 Trace Analytics。

先决条件

Trace Analytics 要求您在应用程序中添加工具，并使用 OpenTelemetry支持的库（例如 Jaeger 或 Zipkin）生成跟踪数据。此步骤完全发生在 S OpenSearch ervice 之外。 OpenTelemetry 文档的 Amazon Distro 包含许多可用帮助您入门的编程语言示例应用程序，包括 Java、Python、Go 和 JavaScript。

将工具插入添加到应用程序后，C OpenTelemetryollector 从应用程序接收数据并将其格式化为 OpenTelemetry 数据。请参阅上的GitHub接收器列表。 Amazon 的发行版 OpenTelemetry 包括一个接收器. Amazon X-Ray

最后，您可以使用Amazon OpenSearch Ingestion对 OpenTelemetry 数据进行格式化以供使用 OpenSearch。

OpenTelemetry 收集器示例配置

要将 OpenTelemetry Collector 与一起使用Amazon OpenSearch Ingestion，请尝试以下示例配置：

```
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 示例配置

若要将跟踪数据发送到S OpenSearch ervice 域，请尝试以下示例 OpenSearch Ingestion 管道配置。有关创建管道的说明，请参阅the section called “创建管道”。

```
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"
```

您在 `sts_role_arn` 选项中指定的管道角色必须具有接收器的写权限。有关为管道角色配置权限的说明，请参阅the section called “设置角色和用户”。

探索跟踪数据

控制面板视图按 HTTP 方法和路径将跟踪组合在一起，以便您可以查看与特定操作相关的平均延迟、错误率和趋势。对于更集中的视图，请尝试按跟踪组名称进行筛选。

要向下钻取组成跟踪组的迹线，请选择右侧列中的迹线数。然后选择一个单独的跟踪获取详细的摘要。

服务视图列出了应用程序中的所有服务，以及显示各种服务之间如何相互连接的交互式地图。与控制面板（有助于按操作识别问题）不同，服务图可帮助您按服务识别问题。尝试按错误率或延迟进行排序，了解应用程序的潜在问题区域。

使用管道处理语言查询 Amazon S OpenSearch ervice 数据

管道处理语言 (PPL) 是一种查询语言，允许您使用管道 (|) 语法查询存储在 Ama OpenSearch zon Service 中的数据。PPL 需要 Elasticsearch 7.9 OpenSearch 或更高版本。

> **Note**
>
> 本文档简要概述了适用于 Amazon S OpenSearch ervice 的 PPL。有关详细步骤和完整的命令参考，请参阅开源 OpenSearch 文档中的 PPL。

PPL 语法由管道字符 (|) 分隔的命令组成，其中数据从左到右流经每个管道。例如，PPL 语法用于查找具有 HTTP 403 或 503 错误的主机数、将它们聚合到每个主机并按影响顺序对它们进行排序，如下所示：

```
source = dashboards_sample_data_logs | where response='403' or response='503' | stats
 count(request) as request_count by host, response | sort -request_count
```

要开始使用，请在 OpenSearch 控制面板中选择 Query Workbench，然后选择 PP L。使用 `bulk` 操作索引一些示例数据：

```
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"}
```

以下示例在 `age` 大于 18 的账户索引中返回文档的 `firstname` 和 `lastname` 字段：

```
search source=accounts | where age > 18 | fields firstname, lastname
```

示例响应

id	firstname	lastname
0	琥珀色	杜克
1	哈蒂	债券
2	纳内特	贝茨
3	戴尔	亚当斯

您可以使用一组完整的只读命令，如 `search`、`where`、`fields`、`rename`、`dedup`、`stats`、`sort`、`eval`、`head`、`top` 和 `rare`。PPL 插件支持所有 SQL 函数，包括数学、三角函数、日期时间、字符串、聚合和高级运算符和表达式。要了解更多信息，请参阅 OpenSearch PPL 参考手册》）。

Amazon OpenSearch 服务的最佳运营实践

本章提供操作亚马逊 OpenSearch 服务域名的最佳实践，并包括适用于许多用例的一般指南。每个工作负载都是独一无二的，具有独特的特征，因此不存在完全适合所有使用案例的万能建议。最重要的最佳实践是通过连续的周期部署、测试和优化域，以找到工作负载的最佳配置、稳定性和成本。

主题

- 监控和提醒
- 分片策略
- 稳定性
- 性能
- 安全性
- 成本优化
- 调整亚马逊 OpenSearch 服务域名的大小
- Amazon Service 中的 OpenSearch
- Amazon OpenSearch 服务中的专用协调器节点
- Amazon OpenSearch 服务中的专用主节点

监控和提醒

以下最佳实践适用于监控您的 OpenSearch 服务域。

配置 CloudWatch 警报

OpenSearch 服务向 Amazon CloudWatch 发送绩效指标。定期查看您的集群和实例指标，并根据您的工作负载性能配置推荐的 CloudWatch 警报。

启用日志发布

OpenSearch 该服务在 Amazon Logs 中公开 OpenSearch 错误日志、搜索慢日志、索引慢日志和审核 CloudWatch 日志。搜索慢速日志、索引慢速日志和错误日志有助于排查性能和稳定性问题。审计日志仅在您启用精细访问权限控制后可用，可跟踪用户活动。有关更多信息，请参阅 OpenSearch 文档中的日志。

搜索慢速日志和索引慢速日志是了解搜索和索引操作性能以及进行问题排查的重要工具。为所有生产域启用搜索和索引慢速日志传输。您还必须配置日志阈值， CloudWatch 否则将无法捕获日志。

分片策略

分片将您的工作负载分布在 OpenSearch 服务域中的数据节点上。正确配置索引有助于提高域的整体性能。

当您将数据发送到 OpenSearch 服务时，将该数据发送到索引。索引类似于数据库表，以文档为行，以字段为列。创建索引时，您可以告诉您要创建 OpenSearch 多少个主分片。主分片是完整数据集的独立分区。 OpenSearch Service 会自动将您的数据分发到索引中的主分片上。您还可以配置索引的副本。每个副本分片都会完整复制该索引的主分片。

OpenSearch Service 会将每个索引的分片映射到集群中的数据节点。它会确保索引的主分片和副本分片位于不同的数据节点上。第一个副本确保索引中的数据有两份。您应始终至少使用一个副本。更多的副本可提供额外的冗余和读取容量。

OpenSearch 向包含属于索引的分片的所有数据节点发送索引请求。它首先会将索引请求发送到包含主分片的数据节点，然后再发送到包含副本分片的数据节点。协调器节点将搜索请求路由到属于该索引的所有分片的主分片或副本分片。

例如，对于具有五个主分片和一个副本的索引，每个索引请求将接触 10 个分片。相比之下，搜索请求会发送到 n 个分片，其中 n 是主分片的数量。对于具有五个主分片和一个副本的索引，每个搜索查询会接触该索引中的五个分片（主分片或副本分片）。

确定分片和数据节点数

使用以下最佳实践来确定域的分片和数据节点数量。

分片大小 – 磁盘上的数据大小直接取决于源数据的大小，并且会随着您为更多数据创建索引而变化。该 source-to-index比率可能差异很大，从 1:10 到 10:1 或更高，但通常在 1:1.10 左右。您可以使用该比率来预测磁盘上的索引大小。您还可以索引一些数据并检索实际的索引大小，以确定工作负载的比率。获得预测索引大小后，请设置分片数量，以确保每个分片的容量介于 10-30GiB 之间（对于搜索工作负载）或 30-50GiB 之间（对于日志工作负载）。50GiB 应为最大值，请务必为增长做好规划。

分片数量 – 在数据节点中分布分片对域的性能有很大影响。当您的索引包含多个分片时，请尝试使分片计数为数据节点数的倍数。这有助于确保分片在数据节点之间均匀分布，防止出现热节点。例如，假设您有 12 个主分片，则数据节点计数应为 2、3、4、6 或 12。但是，分片数量不如分片大小重要，如果您只有 5GiB 的数据，则仍应使用单个分片。

每个数据节点的分片数 – 一个节点可以容纳的分片总数应与节点的 Java 虚拟机（JVM）堆内存成正比。尽量确保每 GiB 堆内存的分片数量为 25 个或以下。例如，具有 32GiB 堆内存的节点应容纳不超过 800 个分片。尽管分片分布可能因您的工作负载模式而异，但 Elasticsearch 每个节点的分片限制为 1,000 个，2.17 及更高版本的分片限制为 OpenSearch 1.1 到 2.15，4,000 个。 OpenSearch借助 cat/allocation API 可以快速查看跨数据节点的分片数量和分片存储总量。

分片 CPU 比率 – 当索引或搜索请求中涉及分片时，它会使用 vCPU 来处理请求。建议以每个分片 1.5 个 vCPU 为初始扩缩点。如果您的实例类型为 8 vCPUs，请设置您的数据节点数量，使每个节点的分片不超过六个。请注意，这是一个近似值。请务必测试您的工作负载并相应地扩展集群。

有关存储卷、分片大小和实例类型的建议，请参阅以下资源：

- the section called “调整域大小”
- the section called “PB 规模”

避免存储偏斜

当集群中有一个或多个节点拥有一个或多个索引的存储数量比例高于其他节点时，会发生存储偏斜。存储偏斜的迹象包括 CPU 利用率不均衡、间歇性和不均匀的延迟以及跨数据节点的队列不均衡。要确定是否存在偏斜问题，请参阅以下问题排查部分：

- the section called “节点分片和存储偏斜”
- the section called “索引分片和存储偏斜”

稳定性

以下最佳实践适用于维护稳定和健康的 OpenSearch 服务域。

随时了解最新动态 OpenSearch

服务软件更新

OpenSearch Servic e 会定期发布软件更新，以增加功能或以其他方式改进您的域名。更新不会更改 OpenSearch 或 Elasticsearch 引擎的版本。我们建议您安排定期运行 DescribeDomainAPI 操作，如果`UpdateStatus`是，则启动服务软件更新`ELIGIBLE`。如果您未在一定时间内（通常为两周）更新域名，S OpenSearch ervice 会自动执行更新。

OpenSearch 版本升级

OpenSearch 该服务会定期增加对社区维护版本的 OpenSearch支持。当最新 OpenSearch 版本可用时，请务必升级到最新版本。

OpenSearch 服务同时升级两个 OpenSearch 仪表板 OpenSearch 和控制面板（如果您的域名运行的是传统引擎，则可以同时升级 Elasticsearch 和 Kibana）。如果集群具有专用主节点，则无需停机即可完成升级。否则，群集在选择主节点时可能会在升级后的几秒钟内没有响应。 OpenSearch 在部分或全部升级期间，仪表板可能不可用。

升级域的方式有两种：

- 就地升级 – 这个选项比较简单，因为您保留的是同一个集群。
- 快照/还原升级 – 这个选项适用于在新集群上测试新版本或在集群之间迁移。

无论使用哪种升级方式，我们都建议您建立一个仅用于开发和测试的域，并将其升级到新版本，然后再升级生产域。在创建测试域时，对于部署类型，选择 Development and testing（开发和测试）。务必在域升级后立即将所有客户端升级到兼容版本。

提高快照性能

为防止快照在处理过程中卡住，专用主节点的实例类型应与分片计数相符。有关更多信息，请参阅 the section called “为专用主节点选择实例类型”。此外，每个节点每 GiB Java 堆内存不应超过 25 个分片（推荐）。有关更多信息，请参阅 the section called “选择分片数量”。

启用专用主节点

专用主节点可提高集群稳定性。专用主节点会执行集群管理任务，但不保留索引数据也不响应客户端请求。通过下载集群管理任务，可提高域的稳定性，并能够在不停机的情况下实施某些配置更改。

启用并使用三个专用主节点，以跨三个可用区优化域稳定性。使用带待机功能的多可用区部署将为您配置三个专用主节点。有关实例类型建议，请参阅the section called “为专用主节点选择实例类型”。

跨多个可用区进行部署

为在服务中断时防止数据丢失并尽量减少集群停机时间，您可以在同一 Amazon Web Services 区域中的两个或三个可用区之间分布节点。最佳做法是使用带待机功能的多可用区进行部署，配置三个可用区，其中两个可用区处于可用状态，一个可用区充当备用可用区，每个索引具有两个副本分片。此配置允许 Ser OpenSearch vice 将副本分片分发到与其对应的主分片 AZs 不同的分片。可用区之间的集群通信不会产生跨可用区数据传输费用。

可用区是每个 区域内的隔离位置。使用双可用区配置时，失去一个可用区意味着您将损失一半的域容量。使用三个可用区可进一步减少失去单个可用区的影响。

控制摄取流量和缓冲

我们建议使用 _bulk API 操作限制请求总数。发送一个包含 5000 个文档的 `_bulk` 请求要比发送 5000 个包含单个文档的请求更高效。

为了确保最佳操作稳定性，有时需要限制甚至暂停索引请求的上游流。限制索引请求的速率是处理意外或偶尔出现的请求峰值的重要机制，否则这些峰值可能会导致集群不堪重负。考虑在上游架构中构建流量控制机制。

下图显示了日志提取架构的多个组件选项。配置聚合层，以留出足够的空间来缓冲传入的数据，应对突发的流量高峰以及满足短暂的域维护需求。

为搜索工作负载创建映射

对于搜索工作负载，创建映射来定义如何 OpenSearch 存储和索引文档及其字段。将 `dynamic` 设置为 `strict`，可防止意外添加新字段。

```
PUT my-index
{
  "mappings": {
    "dynamic": "strict",
    "properties": {
      "title": { "type" : "text" },
      "author": { "type" : "integer" },
```

```
      "year": { "type" : "text" }
    }
  }
}
```

使用索引模板

您可以使用索引模板来告知 OpenSearch 如何在创建索引时对其进行配置。在创建索引之前配置索引模板。然后在创建索引时，它将继承模板的设置和映射。您可以将多个模板应用于单个索引，因此可以在一个模板中指定设置，而在另一个模板中指定映射。此策略允许用一个模板来设置多个索引的通用设置，同时使用单独的模板进行更具体的设置和映射。

以下设置对于在模板中配置非常有用：

- 主分片和副本分片的数量
- 刷新间隔时间（刷新频率以及使最近对索引的更改可供搜索的频率）
- 动态映射控制
- 显式字段映射

以下示例模板包含了所有这些设置：

```
{
   "index_patterns":[
      "index-*"
   ],
   "order": 0,
   "settings": {
      "index": {
         "number_of_shards": 3,
         "number_of_replicas": 1,
         "refresh_interval": "60s"
      }
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "field_name1": {
            "type": "keyword"
         }
      }
   }
```

```
}
```

即使设置和映射很少更改，在其中集中定义设置和映射 OpenSearch 也比更新多个上游客户端更易于管理。

使用索引状态管理来管理索引

如果您正在管理日志或时间序列数据，我们建议使用索引状态管理（ISM）。ISM 允许您自动执行定期性的索引生命周期管理任务。借助 ISM，您可以创建触发索引别名滚动、拍摄索引快照、在存储层之间移动索引以及删除旧索引的策略。您甚至可以将 ISM 滚动操作作为替代数据生命周期管理策略，以避免分片偏斜。

首先，您需要设置一个 ISM 策略。有关示例，请查看 the section called “示例策略”。然后，将该策略附加到一个或多个索引。如果您在策略中包含 ISM 模板字段，S OpenSearch ervice 会自动将策略应用于与指定模式匹配的任何索引。

删除未使用的索引

定期检查集群中的索引并确定任何未使用的索引。拍摄这些索引的快照，以便将它们存储在 S3 中，然后将其删除。移除未使用的索引可减少分片数量，从而能在节点之间实现更均衡的存储分布和资源利用率。即使索引处于空闲状态，它们在内部索引维护活动期间也会消耗一些资源。

您可以使用 ISM 自动拍摄快照并在一段时间后删除索引，而不是手动删除未使用的索引。

使用多个域以实现高可用性

要跨多个区域实现超过 99.9% 正常率的高可用率，请考虑使用两个域。对于较小或变化缓慢的数据集，您可以设置跨集群复制以保持主备模式。在此模式中，只写入领导者域，但可以从任何一个域中读取。对于较大的数据集和快速变化的数据，建议在摄取管道中配置双重传输，以便在双主模式下将所有数据独立写入两个域。

在设计上游和下游应用程序架构注意失效转移的需要。务必要测试失效转移流程以及其他灾难恢复流程。

性能

以下最佳实践有利于调整域以优化性能。

优化批量请求大小和压缩

批量大小取决于您的数据、分析和集群配置，但建议从每个批量请求 3-5MiB 开始。

使用 gzip 压缩来减少请求和响应的有效负载大小，从而发送请求和接收来自您的 OpenSearch 域的响应。您可以在 OpenSearch Python 客户端中使用 gzip 压缩，也可以通过在客户端添加以下标头来使用：

- `'Accept-Encoding': 'gzip'`
- `'Content-Encoding': 'gzip'`

要优化批量请求大小，请先将批量请求大小设为 3MiB。然后，慢慢增加请求大小，直到索引性能停止改善。

> **Note**
>
> 要在运行 Elasticsearch 版本 6.x 的域上启用 gzip 压缩，您必须在集群级别设置 `http_compression.enabled`。在 Elasticsearch 7.x 版本和的所有版本中，默认情况下，此设置均为真。 OpenSearch

降低批量请求响应的大小

要减小 OpenSearch 响应的大小，请使用`filter_path`参数排除不必要的字段。确保不要筛选掉识别或重试失败请求所必需的任何字段。有关更多信息以及示例，请参阅 the section called “减小响应大小”。

优化刷新间隔时间

OpenSearch 索引最终具有读取一致性。刷新操作会使对索引执行的所有更新都可供搜索。默认刷新间隔为一秒，这意味着 OpenSearch 在向索引写入时每秒执行一次刷新。

刷新索引的频率越低（刷新间隔时间越长），索引的整体性能越好。增加刷新间隔时间的缺点是，从索引更新到新数据可供搜索之间的延迟会更长。请将刷新间隔时间设置为可以容忍的最高值，从而提高整体性能。

我们建议将所有索引的 `refresh_interval` 参数设置为 30 秒或更长时间。

启用自动调整

Auto-Tune 使用 OpenSearch 集群中的性能和使用率指标来建议节点上队列大小、缓存大小和 Java 虚拟机 (JVM) 设置的更改。这些可选更改可提高集群速度和稳定性。您可以随时恢复到默认的 OpenSearch 服务设置。除非您显式禁用，否则新域会默认启用自动调整。

我们建议在所有域上启用自动调整功能，并设置定期维护时段或定期查看其建议。

安全性

以下最佳实践有利于域的安全保护。

启用精细访问控制

精细的访问控制允许您控制谁可以访问 OpenSearch 服务域中的某些数据。与一般访问控制相比，精细访问控制可为每个集群、索引、文档和字段提供自己指定的访问策略。访问条件可基于多个因素，包括请求访问权限的人的角色以及他们计划对数据执行的操作。例如，您可以授予一个用户写入索引的权限，而授予另一个用户只读取索引数据而不进行任何更改的权限。

精细访问控制允许具有不同访问要求的数据存在于同一个存储空间中，而不会遇到安全或合规性问题。

我们建议为您的域启用精细访问控制。

在 VPC 中部署域

将 OpenSearch 服务域置于虚拟私有云 (VPC) 中有助于实现 OpenSearch 服务与 VPC 内其他服务之间的安全通信，无需互联网网关、NAT 设备或 VPN 连接。所有流量都安全地保存在 Amazon 云中。由于进行了逻辑隔离，与使用公共端点的域相比，驻留在 VPC 中的域有一层额外的安全性。

我们建议您在 VPC 中创建您的域。

应用限制性访问策略

即使您的域部署在 VPC 中，实施分层安全保护也是最佳做法。务必要对当前的访问策略进行配置检查。

对您的域应用基于资源的限制性访问策略，并在授予对配置 API 和 API 操作的访问权限时遵循最低权限原则。 OpenSearch 原则上，访问策略中应避免使用匿名用户主体 `"Principal": {"AWS": "*" }`。

但在某些情况下，使用开放访问策略是可以接受的，例如启用精细访问控制时。开放访问策略允许您在难以或无法为请求签名时访问域，例如从某些客户端和工具访问。

启用静态加密

OpenSearch 服务域提供静态数据加密，有助于防止未经授权访问您的数据。静态加密使用 Amazon Key Management Service (Amazon KMS) 来存储和管理您的加密密钥，使用 256 位密钥的高级加密标准算法 (AES-256) 来执行加密。

如果您的域存储了敏感数据，则应启用静态数据加密。

启用 node-to-node加密

Node-to-node 除了 OpenSearch 服务中的默认安全功能外，加密还提供了额外的安全层。它为其中配置的节点之间的所有通信实现了传输层安全 (TLS)。 OpenSearch Node-to-node加密，通过 HTTPS 发送到您的 OpenSearch 服务域的任何数据在传输过程中均保持加密状态，同时在节点之间进行分发和复制。

如果您的域名存储敏感数据，请启用 node-to-node加密。

使用监视器 Amazon Security Hub

使用监控您对 OpenSearch 服务的使用情况，因为它与安全最佳实践有关Amazon Security Hub。Security Hub 使用安全控件来评估资源配置和安全标准，以帮助您遵守各种合规框架。有关使用 Security Hub 评估 OpenSearch 服务资源的更多信息，请参阅Amazon Security Hub 用户指南中的Amazon OpenSearch Service 控件。

成本优化

以下最佳实践适用于优化和节省 OpenSearch 服务成本。

使用最新一代实例类型

OpenSearch 服务始终采用新的 Amazon EC2 实例类型，以更低的成本提供更好的性能。我们建议始终使用最新一代的实例。

不要将 T2 或者 `t3.small` 实例用于生产域，因为在持续接受重载时，这些实例会变得不稳定。`r6g.large` 实例适用于小型生产工作负载（既作为数据节点又作为专用主节点）。

使用最新的 Amazon EBS gp3 卷

OpenSearch 数据节点需要低延迟和高吞吐量存储才能提供快速的索引和查询。通过使用 Amazon EBS gp3 卷，您可以获得更高的基准性能（IOPS 和吞吐量），而成本却比之前提供的 Amazon EBS

gp2 卷类型低 9.6%。您可以使用 gp3 预置额外的 IOPS 和吞吐量，不受卷大小的影响。这些卷也比上一代卷更稳定，因为它们不使用突增额度。gp3 卷类型还将 gp2 per-data-node 卷类型的卷大小限制提高了一倍。有了更大的卷，您可以通过增加每个数据节点的存储量来降低被动数据的费用。

时间序列日志数据的使用 UltraWarm 和冷存储

如果您使用 OpenSearch 日志分析，请将数据移至 UltraWarm 或冷存储以降低成本。使用索引状态管理（ISM）在存储层之间迁移数据并管理数据留存。

UltraWarm为在 S OpenSearch ervice 中存储大量只读数据提供了一种经济实惠的方法。 UltraWarm 使用 Amazon S3 进行存储，这意味着数据是不可变的，只需要一个副本。您只需支付相当于索引中主分片大小的存储费用。 UltraWarm 查询的延迟会随着为查询提供服务所需的 S3 数据量而增加。在节点上缓存数据后，对索引的查询与对热 UltraWarm 索引的查询类似。

冷存储也由 S3 提供支持。当你需要查询冷数据时，你可以有选择地将其附加到现有 UltraWarm 节点。冷数据产生的托管存储成本与相同 UltraWarm，但冷存储中的对象不会消耗 UltraWarm 节点资源。因此，冷存储提供了大量的存储容量，而不会影响 UltraWarm 节点大小或数量。

UltraWarm 当您有大约 2.5 TiB 的数据需要从热存储中迁移时，就会变得具有成本效益。监控您的填充率，并计划在达到该数据量 UltraWarm之前将索引移至。

检查有关预留实例的建议

在获得良好的性能和计算消耗基准后，可以考虑购买预留实例 (RIs)。无预付的 1 年期预留实例的折扣从大约 30% 起，所有需要预付的 3 年期预留实例折扣最大可达 50%。

观察稳定运行至少 14 天后，请查看Amazon Cost Management 用户指南中的访问预订建议。Amazon S OpenSearch ervic e 标题显示了具体的 RI 购买建议和预计节省的费用。

调整亚马逊 OpenSearch 服务域名的大小

没有完美的方法可以调整亚马逊 OpenSearch 服务域名的大小。但是，从了解您的存储需求、服务及其 OpenSearch 本身开始，就可以对硬件需求做出有根据的初步估计。此估计可以作为调整域大小的大多数关键方面的有用起始点：用代表性工作负载测试它们并监控其性能。

主题

- 计算存储要求
- 选择分片数量
- 选择实例类型和测试

计算存储要求

大多数 OpenSearch 工作负载分为两大类之一：

- 长效索引：您编写的代码将数据处理为一个或多个 OpenSearch 索引，然后随着源数据的变化定期更新这些索引。一些常见示例为网站、文档和电子商务搜索。
- 滚动索引：数据持续流入一组具有索引周期和保留期限的临时索引，例如一组将保留 2 周的每日索引。一些常见示例是日志分析、时间序列处理和点击流分析。

对于长期有效的索引工作负载，您可以检查磁盘上的源数据并轻松确定它使用的存储空间。如果数据来自多个来源，只需一起添加这些来源。

对于滚动索引，您可以将代表性时间周期内生成的数据量乘以保留周期。例如，如果您每小时生成 200MiB 日志数据，即每天 4.7GiB，如果您的保留周期为 2 周，则任何给定时间的数据均为 66GiB。

但是，您的源数据的大小只是您的存储要求的一个方面。您还必须考虑以下方面：

- 副本数量：每个副本都是主分片的一个完整副本，索引的存储大小显示主分片和副本分片占用的大小。默认情况下，每个 OpenSearch 索引都有一个副本。我们建议至少具有一个，以防数据丢失。副本还可以提高搜索性能，因此如果您有需要进行大量读取操作的工作负载，则可能需要更多副本。使用 `PUT /my-index/_settings` 更新索引的 `number_of_replicas` 设置。
- OpenSearch 索引开销：索引的磁盘大小各不相同。源数据加上索引的总大小通常为源数据的 110%，索引最多为源数据的 10%。为您的数据编制索引后，您可以使用 `_cat/indices?v` API 和 `pri.store.size` 值计算确切的开销。`_cat/allocation?v` 还提供了一个有用的摘要。
- 操作系统预留空间：默认情况下，Linux 将保留 5% 的文件系统供 `root` 用户处理关键流程、进行系统恢复和防止磁盘碎片问题。
- OpenSearch 服务开销： OpenSearch 服务预留每个实例存储空间的 20%（最多 20 GiB），用于分段合并、日志和其他内部操作。

  由于这个 20GiB 的最大值，预留空间的总量可能会相差悬殊，这具体取决于您的域中的实例数量。例如，某个域可能有三个 `m6g.xlarge.search` 实例，每个实例的存储空间为 500GiB，总存储空间为 1.46TiB。在这种情况下，只有 60GiB 的总预留空间。另一个域可能有 10 个 `m3.medium.search` 实例，每个实例的存储空间为 100GiB，总存储空间为 0.98TiB。在这里，总预留空间为 200GiB，即使第一个域大 50%。

  在以下公式中，我们对开销采用“最坏情况”估算值。该估算值包括额外的可用空间，可帮助最大限度地减少节点故障和可用区中断的影响。

总之，如果您在任何给定的时间有 66GiB 的数据并且需要一个副本，则最低 存储要求更接近 66 * 2 * 1.1 / 0.95 / 0.8 = 191GiB。您可以将此计算一般化，如下所示：

源数据 *（1 + 副本数量）*（1 + 索引开销）/（1-Linux 预留空间）/（1- OpenSearch 服务开销）= 最低存储要求

或者，您可以使用此简化版本：

源数据 *（1 + 副本数量）* 1.45 = 最小存储要求

存储空间不足是集群不稳定的最常见原因之一。所以，在选择实例类型、实例数量和存储量时，您应该交叉核对数字。

存在其他存储注意事项：

- 如果您的最小存储要求超过 1 PB，请参阅 the section called “PB 规模”。
- 如果您有滚动索引并希望使用热-暖架构，请参阅 the section called “UltraWarm 存储”。

选择分片数量

在您了解存储要求后，便可以调查您的索引策略。默认情况下，在 S OpenSearch ervice 中，每个索引分为五个主分片和一个副本（总共 10 个分片）。这种行为不同于开源 OpenSearch，后者默认为一个主分片和一个副本分片。由于无法轻松更改现有索引的主分片的数量，在为第一个文档编制索引之前，应决定分片数量。

选择分片数量的总体目标是跨集群中的所有数据节点均匀分配索引。但是，这些分片不应该太大或太多。一般准则是，对于搜索延迟属于关键性能目标的工作负载，建议尽量将分片大小保持在 10-30GiB 之间；而对于写入密集型工作负载（如日志分析），则建议尽量将分片大小保持在 30-50GiB 之间。

较大的分片可能使故障恢复变得困难，但是由于每个分片会占用一定数量的 CPU 和内存，因此拥有过多的小分片可能会导致性能问题和内存不足错误。 OpenSearch 换句话说，分片应该足够小，以便底层 S OpenSearch ervice 实例可以处理它们，但不能太小以至于给硬件带来不必要的压力。

例如，假设您有 66GiB 的数据。您不希望该数字随着时间的推移而增加，您希望将每个分片保持在 30GiB 左右。因此，您的分片数量应大约为 66 * 1.1 / 30 = 3。您可以将此计算一般化，如下所示：

（源数据 + 增长空间）*（1 + 索引开销）/所需的分片大小 = 主分片的大约数量

此等式可帮助补偿今后的数据增长。如果您预计这相同的 66GiB 数据将在下一年增长到原来的四倍，则分片的数量大约为 (66 + 198) * 1.1 / 30 = 10。但是，请记住，您还没有 这额外的 198GiB 数据。

检查以确保为未来所做的这一准备不会创建目前消耗大量 CPU 和内存的多余微小分片。在这种情况下，66 * 1.1 / 10 分片 = 7.26GiB/分片，将消耗额外的资源且小于建议的大小范围。你可以考虑六个分片的更多 middle-of-the-road方法，这样你今天就有 12-GiB 的分片，将来有 48-GiB 的分片。而且，您可能更愿意从 3 个分片开始且在分片超过 50GiB 时为您的数据重新编制索引。

一个不太常见的问题涉及限制每个节点的分片数量。如果您适当地调整分片大小，您通常会在遇到此限制之前，很长时间才会用完磁盘空间。例如，`m6g.large.search` 实例的最大磁盘大小为 512 GiB。如果您的磁盘利用率低于 80%，并且分片大小为 20 GiB，则可容纳大约 20 个分片。弹性搜索 7. x 及更 OpenSearch高版本，以及所有不超过 2.15 的版本，每个节点的分片上限为 1,000 个。要调整每个节点的最大分区数，请配置 `cluster.max_shards_per_node` 设置。对于 OpenSearch 2.17 及更高版本， OpenSearch 服务支持每 16GB 的 JVM 堆内存使用 1000 个分片，每个节点最多支持 4000 个分片。有关示例，请参阅集群设置。有关分片计数的更多信息，请参阅the section called “分片计数配额”。

适当调整分片大小几乎总是使您低于此限制，但您也可以考虑针对每 GiB 的 Java 堆的分片数。在给定节点上，每 GiB 的 Java 堆不超过 25 个分片。例如，一个 `m5.large.search` 实例有一个 4 GiB 堆，因此每个节点应该有不超过 100 个分片。对于该分片计数，每个分片的大小大约为 5 GiB，远低于我们建议的大小。

选择实例类型和测试

当您计算存储要求并选择您需要的分片数量后，您可以开始进行硬件决策。硬件要求可能因工作负载而差异悬殊，但我们仍然可以提供一些基本建议。

一般而言，每个实例类型的存储限制映射到轻型工作负载可能需要的 CPU 和内存量。例如，某个 `m6g.large.search` 实例的最大 EBS 卷大小为 512GiB，该实例具有 2 个 vCPU 核心和 8GiB 内存。如果您的群集有许多分片，需要执行税收聚合、频繁更新文档或处理大量查询，则这些资源可能不足以满足您的需求。如果您的集群属于其中一个类别，请尝试从每 100GiB 存储要求更接近 2 个 vCPU 核心和 8GiB 内存的配置开始。

> **Tip**
>
> 有关分配给每种实例类型的硬件资源的摘要，请参阅 Amazon OpenSearch 服务定价。

但是，即使这些资源也可能不足。一些 OpenSearch 用户报告说，他们需要很多次这些资源来满足他们的需求。要为您的工作负载寻找合适的硬件，您必需进行明智的初步估计、使用代表性工作负载进行测试、调整并再次测试。

步骤 1：进行初步估计

首先，我们建议至少有三个节点，以避免潜在 OpenSearch的问题，例如大脑分裂状态（通信中断导致集群有两个主节点）。如果您有三个专用主节点，我们仍建议至少将两个数据节点用于复制。

步骤 2：计算每个节点的存储需求

如果您的存储要求为 184GiB 而建议的最小节点数量为三个，则可以使用等式 184/3 = 61GiB 来找到每个节点需要的存储量。在此示例中，您可以选择三个 `m6g.large.search` 实例，每个实例使用一个 90GiB 的 EBS 存储卷，以便您有一个安全网和一些随时间增长的空间。此配置提供了 6 个 vCPU 核心和 24GiB 内存，因此适合更轻型的工作负载。

有关更具体的示例，请考虑 14TiB (14336 GiB) 存储要求和重型工作负载。在这种情况下，您可能会选择从 2 * 144 = 288 个 vCPU 核心和 8 * 144 = 1152GiB 内存开始测试。这些数量计为约 18 个 `i3.4xlarge.search` 实例。如果您不需要快速的本地存储，您还可以测试 18 个 `r6g.4xlarge.search` 实例，每个实例使用 1TiB EBS 存储卷。

如果您的群集包含数百 TB 的数据，请参阅the section called “PB 规模”。

步骤 3：执行代表性测试

配置集群后，您可以使用先前计算的分片数量添加索引，使用真实的数据集执行一些具有代表性的客户端测试，并监控 CloudWatch 指标以了解集群如何处理工作负载。

步骤 4：成功或迭代

如果性能满足您的需求，测试成功且 CloudWatch 指标正常，则集群已准备就绪。记得设置 CloudWatch 警报以检测不健康的资源使用情况。

如果性能不可接受、测试失败，或者 `CPUUtilization` 或 `JVMMemoryPressure` 很高，则您可能需要选择其他实例类型 (或添加实例) 并继续测试。添加实例时， OpenSearch 会自动重新平衡整个集群中分片的分配。

因为在动力过剩的集群中测量超额容量比在动力不足的集群中测量容量不足更简单，所以我们建议从比您认为您所需的集群更大的集群开始。接下来，测试并缩减为具有额外资源的高效集群，以确保活动增加期间稳定运行。

生产集群或具有复杂状态的集群可从专用主节点中受益，从而提高性能和集群可靠性。

Amazon Service 中的 OpenSearch

Amazon S OpenSearch ervice 域提供多达 10 PB 的附加存储。您可以将域配置为 1000 个`OR1.16xlarge.search`实例类型，每种实例类型都有 36 TB 的存储空间。由于规模上的显著差异，针对此大小的域的建议不同于我们的一般建议。本节讨论创建域、成本、存储空间和分片大小的注意事项。

虽然本节经常引用`i3.16xlarge.search`实例类型，但您可以使用其他几种实例类型来达到总域存储空间的 10 PB。

创建域

此大小的域超出了每个域 80 个实例的默认限制。要请求将服务限制提升到每个域最多 1000 个实例，请在 Su Amazon pport Center 上打开一个案例。

定价

在创建此大小的域之前，请查看 Amazon S OpenSearch ervice 定价页面以确保相关成本符合您的预期。检查 the section called “UltraWarm 存储” 以查看热-温架构是否适合您的使用案例。

存储

这些`i3`实例类型旨在提供快速的本地非易失性内存 express (NVMe) 存储。由于此本地存储与 Amazon Elastic Block Store 相比往往具有性能优势，当您在 Ser OpenSearch vice 中选择这些实例类型时，EBS 卷不是合适的选项。如果您更喜欢 EBS 存储，请使用其他实例类型，如 `r6.12xlarge.search`。

分片大小和计数

常用 OpenSearch 准则是不超过每个分片 50 GB。考虑到容纳大型域和可用于 `i3.16xlarge.search` 实例的资源所需的分片数量，建议使用 100 GB 的分片大小。

例如，如果您有 450 TB 的源数据并且需要一个副本，您的最低 存储要求更接近 450 TB * 2 * 1.1 / 0.95 = 1.04 PB。有关此计算的说明，请参阅the section called “计算存储要求”。尽管 1.04 PB/15 TB = 70 个实例，但您可以选择 90 个或更多 `i3.16xlarge.search` 实例来为自己提供存储安全网，处理节点失败和账户，从而适应数据量随着时间的推移发生的变化。每个实例都会将存储需求下限增加 20GiB，但对于这个规模的磁盘而言，20GiB 几乎可以忽略不计。

控制分片数量很棘手。 OpenSearch 用户通常每天轮换索引，并保留一周或两周的数据。在这种情况下，您可能会发现，区分“活动”和“非活动”分片很有用。活动分片，就是经常发生读取或写入的分片。非活动分片可能为一些读取请求提供服务，但基本上是闲置的。一般而言，您应该把活动分片的数量保持在数千以下。当活动分片的数量接近 10,000 时，会出现相当大的性能和稳定性风险。

要计算主分片的数量，请使用以下公式：450,000 GB * 1.1/每个分片 100 GB = 4,950 个分片。将这个数字翻倍以考虑副本是 9900 个分片，这在所有分片都处于活动状态时表示主要考虑因素。但是，如果您轮换索引，而且 1/7 或 1/14 的分片在给定日期处于活动状态（分别是 1414 或 707 个分片），则集群可能工作正常。与往常一样，对您的域进行大小调整和配置的最重要的步骤是使用真实的数据集执行有代表性的客户端测试。

Amazon OpenSearch 服务中的专用协调器节点

Amazon S OpenSearch ervice 中的专用协调器节点是从数据节点卸载协调任务的专用节点。这些任务包括管理搜索请求和托管 OpenSearch 仪表板。通过分离这些功能，专用协调器节点可以减少数据节点的负载，从而使它们能够专注于数据存储、索引和搜索操作。这提高了群集的整体性能和资源利用率。

此外，专用协调器节点有助于减少 VPC 配置所需的私有 IP 地址数量，从而提高网络管理效率。此设置可使索引吞吐量提高多达 15%，查询性能提高 20%，具体取决于工作负载特征。

何时使用专用协调器节点

在以下情况下，专用协调器节点最有用。

- 大型集群-在数据量大或查询复杂的环境中，将协调任务卸载到专用节点可以提高集群性能。
- 频繁查询 — 涉及频繁搜索查询或聚合的工作负载，尤其是那些具有复杂日期直方图或多个聚合的工作负载，可以受益于更快的查询处理。
- 大量使用仪表板- OpenSearch 仪表板可能占用大量资源。将此责任移交给专用的协调器节点可以减轻数据节点的压力。

架构和行为

在集 OpenSearch 群中，专用协调器节点负责两个关键职责。

- 请求处理-这些节点接收传入的搜索请求并将其转发到存储相关数据的相应数据节点。然后，他们将来自多个数据节点的结果整合到一个全局结果集中，然后返回给客户端。
- 仪表板托管 — 协调器节点管理 OpenSearch 仪表板，从而减轻数据节点托管 OpenSearch 仪表板和处理相关流量的额外负担。

在 VPC 域中，为专用的协调器节点分配的是弹性网络接口 (ENIs)，而不是数据节点。这种安排有助于减少所需的私有 IP 地址数量 VPCs，从而提高网络效率。通常，专用协调器节点约占数据节点总数的 10%。

要求和限制

专用协调器节点具有以下要求和限制。

- 所有 OpenSearch 版本和 Elasticsearch 版本 6.8 至 7.10 都支持专用协调器节点。
- 要启用专用协调器节点，您的域必须启用专用主节点。有关更多信息，请参阅 the section called “专用主节点”。
- 配置专用协调器节点可能会产生额外费用。但是，资源效率的提高和性能的增强证明了投资的合理性，对于大型或复杂的集群来说尤其如此。

配置专用协调器节点

执行以下步骤，在现有域中配置专用协调器节点。在配置协调器节点之前，请确保您的域已启用专用主节点。

控制台

要在中配置专用的协调器节点 Amazon Web Services Management Console

1. 在https://console.aws.amazon.com/aos/家中登录亚马逊 OpenSearch 服务控制台。
2. 选择 “域名”，然后选择要修改的域名。
3. 在集群配置部分，选择编辑。
4. 选择启用专用协调器节点。
5. 选择要配置的实例类型和协调器节点数量。
6. 选择保存更改。域名更新可能需要几分钟。

Amazon CLI

要使用配置专用协调器节点 Amazon CLI，请使用update-domain-config命令。以下示例在一个域中配置了三个`r6g.large.search`协调器节点。

```
aws opensearch update-domain-config \
  --domain-name my-opensearch-domain \
```

```
 --cluster-config
InstanceCount=3,InstanceType=r6g.large.search,DedicatedCoordinatorCount=3,ZoneAwarenessEnabled
```

此命令启用专用协调器节点，设置协调器节点的实例类型和数量，并启用区域感知以提高可用性。

最佳实践

使用专用协调器节点时，请考虑以下最佳实践。

- 对于大多数用例，请使用通用实例。它们在成本和性能之间提供了一种平衡的方法。内存优化型实例非常适合需要大量内存资源的工作负载，例如涉及复杂聚合或大规模搜索的工作负载。
- 一个好的起点是将 5% 到 10% 的数据节点配置为专用协调器节点。例如，如果您的域有 90 个特定实例类型的数据节点，请考虑预配置 5 到 9 个相同实例类型的协调器节点。

> **Note**
>
> 实例类型的可用性因地区而异。为协调器节点选择实例类型时，请确认您选择的实例类型在目标区域可用。创建或修改域名时，您可以在 OpenSearch 服务控制台中查看实例类型的可用性。

- 为了最大限度地降低单点故障的风险，请配置至少两个专用的协调器节点。这样可以确保即使一个节点出现故障，您的群集也能保持运行状态。
- 如果您使用跨区域搜索，请在目标域中配置专用的协调器节点。源域通常不处理协调任务。
- 对于索引密集型环境，可以考虑使用与数据节点的实例大小相匹配的 CPU 优化型实例，以获得最佳性能。
- 对于内存密集型工作负载，请使用稍大的实例类型作为专用协调器节点，以帮助管理增加的内存需求。
- 追踪 A `CoordinatorCPUUtilization` mazon 的 CloudWatch 指标。如果持续超过 80%，则可能表明您需要更大或更多的协调器节点来处理负载。

按群集大小划分的节点推荐

使用以下准则作为起点，根据您的集群大小配置专用协调器节点。根据工作负载特征和性能指标调整节点的数量和类型。

集群大小	推荐的协调器节点	实例类型
小型（最多 50 个节点）	3-5 个节点	通用型
中型（50-100 个节点）	5-9 个节点	内存优化型
大型（100 多个节点）	10-15 个节点	内存优化型

Amazon OpenSearch 服务中的专用主节点

Amazon Ser OpenSearch vice 使用专用的主节点来提高集群稳定性。专用主节点执行群集管理任务，但不保留数据也不响应数据上传请求。此群集管理任务的卸载可增加您的域的稳定性。与所有其他节点类型一样，您为每个专用主节点支付小时费率。

专用主节点执行以下群集管理任务：

- 跟踪集群中的所有节点。
- 跟踪集群中的索引数量。
- 跟踪属于每个索引的分片数量。
- 维护集群中节点的路由信息。
- 在状态更新后更新集群状态，如在集群中创建索引和添加或移除节点。
- 在集群中的所有节点之间复制集群状态的更改。
- 通过发送心跳信号（用于监控集群中数据节点可用性的周期信号）来监控所有集群节点的运行状况。

下图显示了一个包含 10 个实例的 OpenSearch 服务域。七个实例是数据节点，三个是专用主节点。只有一个专用主节点处于活跃状态。两个灰色专用主节点作为备份，以防活跃的专用主节点发生故障。所有数据上传请求由七个数据节点提供，并且所有群集管理任务均卸载到活动的专用主节点。

专用主节点的数量

我们建议您使用带备用空间的多可用区，这会向每个生产 OpenSearch 服务域添加三个专用的主节点。如果使用不带待机功能的多可用区或单可用区进行部署，我们仍然建议使用三个专用主节点。请勿选择偶数专用主节点。选择专用主节点的数量时，请考虑以下事项：

- Ser OpenSearch vice 明确禁止使用一个专用主节点，因为在出现故障时您没有备份。如果您尝试创建只有一个专用主节点的域，您会收到验证异常。
- 如果您有两个专用主节点，您的集群就没有必需的仲裁节点数，无法在发生故障时选择新的主节点。

  仲裁节点数为专用主节点数/2 + 1（向下取整到最近的整数）。在这种情况下，2/2 + 1 = 2。由于一个专用主节点发生了故障且仅存在一个备份，因此集群没有达到仲裁节点数且无法选择新的主节点。
- 三个专用主节点（建议的数量）可在主节点发生故障时提供两个备份节点和必要的仲裁节点数 (2) 来选择新主节点。
- 四个专用主节点并不比三个好，如果您使用多个可用区，可能会导致问题。

- 如果一个主节点发生故障，您有选择新主节点所需的仲裁节点数 (3)。如果两个节点发生故障，则您会失去该仲裁节点数，正如您具有三个专用主节点时一样。
- 在三个可用区配置中，两个可用区 AZs 有一个专用主节点，一个可用区有两个专用主节点。如果该亚利桑那州出现中断，则其余两个亚利桑那州 AZs 没有必要的法定人数 (3) 来选出新的主管。

- 拥有五个专用主节点与拥有三个节点效果一样，并且允许您在维持仲裁节点数的同时丢失两个节点。但是，由于在任何给定时间只有一个专用主节点处于活跃状态，因此此配置意味着您需要为四个空闲节点付费。许多用户发现，这一级别的故障转移保护有些过剩。

如果一个集群拥有符合主服务器条件的节点数量为偶数， OpenSearch 则为 Elasticsearch 版本 7。 x 及之后会忽略一个节点，因此投票配置始终为奇数。在此情况下，4 个专用主节点实质上相当于 3 个（2 比 1）。

> **Note**
>
> 如果您的集群没有必要的仲裁节点数来选择新的主节点，则集群的写入和 读取请求均会失败。此行为与 OpenSearch 默认行为不同。

为专用主节点选择实例类型

OpenSearch 服务域和实例配额

虽然专用主节点不处理搜索和查询请求，但它们的大小与实例大小及其管理的实例、索引和分片数量高度相关。对于生产群集，我们建议专用主节点至少采用以下实例类型。

这些建议基于典型工作负载，可能根据您的需求而有所不同。具有许多分片或字段映射的集群可受益于更大的实例类型。有关更多信息，请参阅 Amazon S OpenSearch ervice 的推荐 CloudWatch 警报，以确定您是否需要使用更大的实例类型。

RAM	Elasticsearch 和 Ser OpenSearch vice 的最大节点支持 1.x 到 2.15	Elasticsearch 和 Ser OpenSearc h vice 的最大 Shard Support 1.x 到 2.15	OpenSearch 服务 2.17 及更高版本的 Max Node Support	适用于 OpenSearch 服务 2.17 及更高版本的 Max Shard Support
2 GB	不适用	不适用	10	1K

RAM	Elasticsearch 和 Ser OpenSearch vice 的最大节点支持 1.x 到 2.15	Elasticsearch 和 Ser OpenSearc h vice 的最大 Shard Support 1.x 到 2.15	OpenSearch 服务 2.17 及更高版本的 Max Node Support	适用于 OpenSearch 服务 2.17 及更高版本的 Max Shard Support
4 GB	不适用	不适用	10	5K
8 GB	10	10K	30	15K
16 GB	30	30K	60	30K
32 GB	75	40K	120	60K
64 GB	125	75K	240	120K
128 GB	200	75K	480	240K
256GB	不适用	不适用	1002	500K

Amazon OpenSearch 服务的推荐 CloudWatch 警报

CloudWatch 当 CloudWatch 指标在一段时间内超过指定值时，警报会执行操作。例如，如果您的集群运行状况超过一分钟，您可能需要 Amazon `red`给您发送电子邮件。本节包括一些推荐的 Amazon S OpenSearch ervice 警报以及如何响应警报。

您可以使用自动部署这些警报 Amazon CloudFormation。有关示例堆栈，请参阅相关GitHub存储库。

> **Note**
>
> 如果您部署 CloudFormation 堆栈，则`KMSKeyError`和`KMSKeyInaccessible`警报将处于某种`Insufficient Data`状态，因为这些指标仅在域名遇到加密密钥问题时才会出现。

有关配置警报的更多信息，请参阅亚马逊 CloudWatch 用户指南中的创建亚马逊 CloudWatch警报。

警报	事务
ClusterStatus.red 最大值 >= 1 达到 1 分钟，1 次连续时间	至少有一个主分片其及副本未分配给节点。请参阅the section called “红色集群状态”。
ClusterStatus.yellow 最大值 >= 1 持续 1 分钟，连续 5 次	至少有一个副本分片未分配给节点。请参阅the section called “黄色集群状态”。
FreeStorageSpace 最小值 <= 20480 达到 1 分钟，1 次连续时间	您的集群中的节点已降至 20GiB 的可用存储空间。请参阅the section called “缺少可用存储空间”。此值以 MiB 为单位，因此我们建议将其设置为每个节点的存储空间的 25%，而不是 20480。
ClusterIndexWrites Blocked 大于等于 1 达到 5 分钟，1 次连续时间	您的群集正在阻止写入请求。请参阅the section called “ClusterBlockException”。
Nodes 最小值 < x 达到 1 天，1 次连续时间	x 是您的集群中的节点数。此警报表示您的群集中至少有一个节点无法访问的时间已达到一天。请参阅the section called “集群节点失败”。

AutomatedSnapshotFailure 最大值 >= 1 达到 1 分钟，1 次连续时间	自动快照失败。此故障通常由红色群集运行状况导致。请参阅the section called “红色集群状态”。
	
	有关所有自动快照的摘要和一些有关故障的信息，您也可以尝试以下操作：
	
	`GET domain_endpoint /_snapshot/cs-automated/_all`
	`GET domain_endpoint /_snapshot/cs-automated-enc/_all`

CPUUtilization 或 WarmCPUUtilization 最大	有时可能会出现 100% 的 CPU 利用率，但是持续的高利用率是有问题的。考虑使用更大的实例类型或添加实例。

警报	事务
值 >= 80% 达到 15 分钟，连续 3 次	
`JVMMemoryPressure` 最大值 >= 95% 达到 1 分钟，连续 3 次	如果使用量增加，群集可能会遇到内存不足错误。考虑垂直缩放。OpenSearch 服务将实例内存的一半用于 Java 堆，堆大小不超过 32 GiB。您最多可以将实例的 RAM 垂直扩展至 64GiB，此时可以通过添加实例水平扩展。
`OldGenJVMMemoryPressure` 最大值 >= 80% 达到 1 分钟，连续 3 次	
`MasterCPUUtilization` 最大值 >= 50% 达到 15 分钟，连续 3 次	考虑将更大的实例类型用于您的专用主节点。由于其在集群稳定性和蓝/绿部署中的作用，专用主节点的 CPU 使用率应比数据节点低。
`MasterJVMMemoryPressure` 最大值 >= 95% 达到 1 分钟，连续 3 次	
`MasterOldGenJVMMemoryPressure` 最大值 >= 80% 达到 1 分钟，连续 3 次	
`KMSKeyError` >= 1 达到 1 分钟，1 次连续时间	用于 Amazon KMS 加密域中静态数据的加密密钥已禁用。重新启用它可恢复正常操作。有关更多信息，请参阅 the section called “静态加密”。

警报	事务
KMSKeyInaccessible >= 1 达到 1 分钟，1 次连续时间	用于 Amazon KMS 加密您域中静态数据的加密密钥已被删除或已撤销其对 Serv OpenSearch ice 的授权。您无法恢复处于此状态的域。但如果您具有手动快照，则可以使用它迁移到新域。要了解更多信息，请参阅the section called “静态加密”。
shards.active >= 30000 达到 1 分钟，1 次连续时间	活动主分区和副本分区的总数大于 30000。轮换索引的频率可能过于频繁。请考虑使用 ISM 在索引达到特定使用期限之后将其移除。
5*xx* 警报 >= OpenSearchRequests 的 10%	一个或多个数据节点可能会重载，或者请求无法在空闲超时期限内完成。请考虑切换为更大的实例类型，或向集群添加更多节点。请确认您遵循以下分区和集群架构最佳实践。
MasterReachableFromNode 最大值 < 1 达到 5 分钟，1 次连续时间	此警报指示主节点已停止或无法访问。这些故障通常是网络连接问题或 Amazon 依赖问题造成的。
ThreadpoolWriteQueue 平均值 >= 100 达到 1 分钟，1 次连续时间	集群正在经历高索引并发。请检查和控制索引请求，或增加集群资源。
ThreadpoolSearchQueue 平均值 >= 500 达到 1 分钟，1 次连续时间	集群正在经历高搜索并发。请考虑扩展集群。您也可以增加搜索队列大小，但过度增加搜索队列大小可能会导致出现内存不足错误。
ThreadpoolSearchQueue 最大值 >= 5000 达到 1 分钟，1 次连续时间	

警报	事务
`ThreadpoolSearchRejected` 增加 SUM >=1{ 数学表达式 DIFF ( )} 达到 1 分钟，1 次连续时间	这些警报会通知您可能会影响性能和稳定性的域问题。
`ThreadpoolWriteRejected` 增加 SUM >=1{ 数学表达式 DIFF ( )} 达到 1 分钟，1 次连续时间	

> **Note**
>
> 如果您只是想查看指标，请参阅 the section called “监控集群指标”。

您可能会考虑的其他警报

考虑根据您经常使用的 OpenSearch 服务功能配置以下警报。

警报	事务
`WarmFreeStorageSpace` >= 10%	您已达到免费预热存储空间总量的 10%。 `WarmFreeStorageSpace` 以 MiB 为单位测量可用暖存储空间的总和。 UltraWarm 使用 Amazon S3 而不是连接的磁盘。
`HotToWarmMigrationQueueSize` >= 20 达到 1 分钟，3 次连续时间	大量索引同时从热索引移动到 UltraWarm 存储索引。请考虑扩展集群。
`HotToWarmMigration`	配置此警报，以便在尝试轮询每日索引时，在 `HotToWarmMigrationSuccessCount` 延迟大于 24 小时的时候通知您。

警报	事务
SuccessLatency >= 1 天，1 次连续时间	
WarmJVMMemoryPressure 最大值 >= 95% 达到 1 分钟，连续 3 次	如果使用量增加，群集可能会遇到内存不足错误。请考虑垂直扩展。OpenSearch 服务将实例内存的一半用于 Java 堆，堆大小不超过 32 GiB。您最多可以将实例的 RAM 垂直扩展至 64GiB，此时可以通过添加实例水平扩展。
WarmOldGenJVMMemoryPressure 最大值 >= 80% 达到 1 分钟，连续 3 次	
WarmToColdMigrationQueueSize >= 20 达到 1 分钟，3 次连续时间	大量索引同时从冷存储迁移 UltraWarm 到冷存储。请考虑扩展集群。
HotToWarmMigrationFailureCount >= 1 达到 1 分钟，1 次连续时间	迁移可能会在快照、分区重新定位或强制合并期间失败。快照或分片重新定位期间的故障通常是由于节点故障或 S3 连接问题造成的。磁盘空间不足通常是强制合并失败的根本原因。
WarmToColdMigrationFailureCount >= 1 达到 1 分钟，1 次连续时间	如果在尝试将索引元数据迁移到冷存储时失败，迁移通常也会失败。在删除暖索引群集状态时，也可能会发生故障。
WarmToColdMigrationLatency >= 1 天，1 次连续时间	配置此警报，以便在尝试轮询每日索引时，在 WarmToColdMigrationSuccessCount 延迟大于 24 小时的时候通知您。

警报	事务
`AlertingDegraded` >= 1 达到 1 分钟，1 次连续时间	警报索引为红色，或者一个或多个节点未按计划运行。
`ADPluginUnhealthy` >= 1 达到 1 分钟，1 次连续时间	异常检测插件未正常工作，原因是故障率过高，或者其中一个正在使用的索引为红色。
`AsynchronousSearchFailureRate` >= 1 达到 1 分钟，1 次连续时间	过去一分钟至少有一次异步搜索失败，这可能意味着协调器节点出现故障。异步搜索请求的生命周期仅在协调器节点上管理，因此，如果协调器关闭，则请求将会失败。
`AsynchronousSearchStoreHealth` >= 1 达到 1 分钟，1 次连续时间	持久索引中的异步搜索响应存储的运行状况为红色。您可能会存储大量异步响应，这可能会破坏集群的稳定性。请尝试将异步搜索响应限制为 10MB 或更少。
`SQLUnhealthy` >= 1 达到 1 分钟，3 次连续时间	SQL 插件正在返回 5 个 xx 响应代码或将无效的查询 DSL 传递给。OpenSearch排查客户端对插件发出的请求是否存在问题。
`LTRStatus.red` >= 1 达到 1 分钟，1 次连续时间	运行 Learning to Rank 插件所需的索引中至少有一个缺少主分片，且不起作用。

Amazon OpenSearch 服务的一般参考

Amazon OpenSearch 服务支持各种实例、操作、插件和其他资源。

主题

- Amazon OpenSearch 服务支持的实例类型
- Amazon OpenSearch 服务中按引擎版本划分的功能
- Amazon OpenSearch 服务中按引擎版本划分的插件
- Amazon OpenSearch 服务中支持的操作
- 亚马逊 OpenSearch 服务配额
- Amazon S OpenSearch ervice 预留实例
- Amazon OpenSearch 服务中其他支持的资源

Amazon OpenSearch 服务支持的实例类型

Amazon OpenSearch 服务支持以下实例类型。并非所有区域都支持所有实例类型。有关可用性的详细信息，请参阅 Amazon OpenSearch 服务定价。

有关哪种实例类型适合您的使用案例的信息，请参阅the section called “调整域大小”、the section called “EBS 卷大小配额”和the section called “网络配额”。

当前一代实例类型

为了获得最佳性能，我们建议您在创建新的 OpenSearch 服务域时使用以下实例类型。

实例类型	实例	限制
i4i	i4i.large.search i4i.xlarge.search i4i.2xlarge.search	i4i 实例类型需要 Elasticsearch 5.1 或更高版本或任何版本 OpenSearch，并且不支持 EBS 卷存储。

实例类型	实例	限制
	i4i.4xlarge.search i4i.8xlarge.search i4i.12xlarge.search i4i.16xlarge.search i4i.24xlarge.search i4i.32xlarge.search	
i4g	i4g.large.search i4g.xlarge.search i4g.2xlarge.search i4g.4xlarge.search i4g.8xlarge.search i4g.16xlarge.search	i4g 实例类型需要 Elasticsearch 7.9 或更高版本或任何版本 OpenSearch，并且不支持 EBS 存储卷。

实例类型	实例	限制
Graviton3	c7g.large.Search c7g.xlarge.Search c7g.2xLarge.Search c7g.4xLarge.Search c7g.8xlarge.Search c7g.12xLarge.Search c7g.16xLarge.Search m7g.large.Search m7g.xlarge.Search m7g.2xLarge.Search m7g.4xLarge.Search m7g.8xlarge.Search m7g.12xLarge.Search	Graviton3 仅支持。 GP3

实例类型	实例	限制
	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 r7gd.4xLarge.search	

实例类型	实例	限制
	r7gd.8xlarge.Search r7gd.12xLarge.search r7gd.16xLarge.Search	
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 实例类型需要 OpenSearch 2.11 或更高版本。 • OR1 实例仅与其他 Graviton 实例类型的主节点（c6g、m6g、r6g）兼容。

实例类型	实例	限制
OR2	or2.medium.search or2.large.search or2.xlarge.search or2.2xlarge.search or2.4xlarge.search or2.8xlarge.search or2.12xlarge.search or2.16xlarge.search	

实例类型	实例	限制

OM2	`om2.large.search`
	`om2.xlarge.search`
	`om2.2xlarge.search`
	`om2.4xlarge.search`
	`om2.8xlarge.search`
	`om2.12xlarge.search`
	`om2.16xlarge.search`	

实例类型	实例	限制

lm4gn	`im4gn.large.search`
	
	`im4gn.xlarge.search`
	
	`im4gn.2xlarge.search`
	
	`im4gn.4xlarge.search`
	
	`im4gn.8xlarge.search`
	
	`im4gn.16xlarge.search`
		• im4GN 实例类型需要 Elasticsearch 7.9 或更高版本或任何版本 OpenSearch，并且不支持 EBS 存储卷。
		• lm4gn 实例仅与其他 Graviton 实例类型（C6g、M6g、R6g、R6gd）兼容。您无法在同一集群中组合 Graviton 和非 Graviton 实例。

实例类型	实例	限制

C5	`c5.large.search`
	`c5.xlarge.search`
	`c5.2xlarge.search`
	`c5.4xlarge.search`
	`c5.9xlarge.search`
	`c5.18xlarge.search`
		C5 实例类型需要 Elasticsearch 5.1 或更高版本或任何版本。OpenSearch

实例类型	实例	限制

C6g	`c6g.large.search`
	`c6g.xlarge.search`
	`c6g.2xlarge.search`
	`c6g.4xlarge.search`
	`c6g.8xlarge.search`
	`c6g.12xlarge.search`
		• C6g 实例类型需要 Elasticsearch 7.9 或更高版本或任何版本。OpenSearch
		• C6g 实例仅与其他 Graviton 实例类型（Im4gn、M6g、R6g、R6gd）兼容。您无法在同一集群中组合 Graviton 和非 Graviton 实例。

实例类型	实例	限制

I3	`i3.large.search`
	`i3.xlarge.search`
	`i3.2xlarge.search`
	`i3.4xlarge.search`
	`i3.8xlarge.search`
	`i3.16xlarge.search`	

M5	`m5.large.search`
	`m5.xlarge.search`
	`m5.2xlarge.search`
	`m5.4xlarge.search`
	`m5.12xlarge.search`
		M5 实例类型需要 Elasticsearch 5.1 或更高版本或任何版本。OpenSearch

实例类型	实例	限制

M6g	`m6g.large.search`
	`m6g.xlarge.search`
	`m6g.2xlarge.search`
	`m6g.4xlarge.search`
	`m6g.8xlarge.search`
	`m6g.12xlarge.search`
		• M6g 实例类型需要 Elasticsearch 7.9 或更高版本或任何版本。OpenSearch
		• M6g 实例仅与其他 Graviton 实例类型（Im4gn、C6g、R6g、R6gd）兼容。您无法在同一集群中组合 Graviton 和非 Graviton 实例。

R5	`r5.large.search`
	`r5.xlarge.search`
	`r5.2xlarge.search`
	`r5.4xlarge.search`
	`r5.12xlarge.search`
		R5 实例类型需要 Elasticsearch 5.1 或更高版本或任何版本。OpenSearch

实例类型	实例	限制

R6g	`r6g.large.search`
	`r6g.xlarge.search`
	`r6g.2xlarge.search`
	`r6g.4xlarge.search`
	`r6g.8xlarge.search`
	`r6g.12xlarge.search`
		• R6g 实例类型需要 Elasticsearch 7.9 或更高版本或任何版本。OpenSearch
		• R6g 实例仅与其他 Graviton 实例类型（Im4gn、C6g、M6g、R6gd）兼容。您无法在同一集群中组合 Graviton 和非 Graviton 实例。

实例类型	实例	限制

R6gd	`r6gd.large.search`
	`r6gd.xlarge.search`
	`r6gd.2xlarge.search`
	`r6gd.4xlarge.search`
	`r6gd.8xlarge.search`
	`r6gd.12xlarge.search`
	`r6gd.16xlarge.search`
		• R6gD 实例类型需要 Elasticsearch 7.9 或更高版本或任何版本 OpenSearch，并且不支持 EBS 存储卷。
		• R6gd 实例仅与其他 Graviton 实例类型（Im4gn、C6g、M6g、R6g）兼容。您无法在同一集群中组合 Graviton 和非 Graviton 实例。

T3	`t3.small.search` `t3.medium.search`	• T3 实例类型需要 Elasticsearch 5.6 或更高版本或任何版本。OpenSearch • 仅在域没有预置备用实例时，才能使用 T3 实例类型。有关更多信息，请参阅 the section called “不带待机功能的多可用区”。 • 仅当域的实例计数小于或等于 10 时，才能使用 T3 实例类型。 • T3 实例类型不支持 UltraWarm存储、冷存储或自动调整。

实例类型	实例	限制

c7i	`c7i.large.search`
	`c7i.xlarge.search`
	`c7i.2xlarge.search`
	`c7i.4xlarge.search`
	`c7i.8xlarge.search`
	`c7i.12xlarge.search`
	`c7i.16xlarge.search`
		• c7i 实例需要 Elasticsearch 5.1 或更高版本或任何版本 OpenSearch，并且仅支持存储卷。 GP3

实例类型	实例	限制

m7i	`m7i.large.search`
	`m7i.xlarge.search`
	`m7i.2xlarge.search`
	`m7i.4xlarge.search`
	`m7i.8xlarge.search`
	`m7i.12xlarge.search`
	`m7i.16xlarge.search`
		• m7i 实例需要 Elasticsearch 5.1 或更高版本或任何版本 OpenSearch，并且仅支持存储卷。 GP3

实例类型	实例	限制

r7i	`r7i.large.search`
	`r7i.xlarge.search`
	`r7i.2xlarge.search`
	`r7i.4xlarge.search`
	`r7i.8xlarge.search`
	`r7i.12xlarge.search`
	`r7i.16xlarge.search`
		• r7i 实例需要 Elasticsearch 5.1 或更高版本或任何版本 OpenSearch，并且仅支持存储卷。 GP3

上一代实例类型

OpenSearch Service 为已经围绕这些实例类型优化了应用程序但尚未升级的用户提供上一代实例类型。建议您使用当前一代实例类型以获得最佳性能，但我们会继续支持以下上一代实例类型。

实例类型	实例	限制
C4	`c4.large.search`	

实例类型	实例	限制

	`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 实例类型不支持静态数据加密、精细访问控制或集群搜索。
		• M3 实例类型因 OpenSearch 版本而异，还有其他限制。要了解更多信息，请参阅the section called “M3 实例类型无效”。

实例类型	实例	限制
M4	m4.large.search 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 实例类型不支持静态数据加密或精细访问控制。

实例类型	实例	限制

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`
		• 仅当您的域的实例计数小于或等于 10 时，才能使用 T2 实例类型。
		• `t2.micro.search` 实例类型仅支持 Elasticsearch 1.5 和 2.3。
		• T2 实例类型不支持静态数据加密、精细访问控制、存储、冷 UltraWarm存储、跨集群搜索或自动调整。

> **Tip**
>
> 我们经常建议，对于专用主节点和数据节点，可以使用不同的实例类型。

Amazon OpenSearch 服务中按引擎版本划分的功能

许多 OpenSearch 服务功能都有最低 OpenSearch 版本要求或传统的 Elasticsearch OSS 版本要求。如果您满足某个功能的最低版本，但该功能在您的域中不可用，请更新您的域的服务软件。

功能	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
自定义插件	2.15	不包括
专用协调器节点	1.0	6.8
VPC 支持	1.0	1.0
要求对到达域的所有流量使用 HTTPS		
多可用区支持		
专用主节点		
自定义程序包		
自定义端点		
慢速日志发布		
错误日志发布	1.0	5.1
静态数据加密		
仪表板的 Cognito		

功能	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
身份验证 OpenSearch		
就地升级		
Curator 支持	不包括	5.1
每小时自动快照	1.0	5.3
Node-to-node 加密	1.0	6.0
Java 高级别 REST 客户端支持		
HTTP 请求和响应压缩		
警报	1.0	6.2
SQL	1.0	6.5
跨集群搜索	1.0	6.7
精细访问控制		
仪表板的 SAML 身份验证 OpenSearch		
自动优化		

功能	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
远程重建索引		
UltraWarm	1.0	6.8
索引状态管理		
按欧几里得距离的 k-NN	1.0	7.1
异常检测	1.0	7.4
按余弦相似性的 k-NN	1.0	7.7
学习排名		
管道处理语言	1.0	7.9
OpenSearch 仪表板报告		
OpenSearch 仪表板追踪分析		
基于 ARM 的 Graviton 实例		
冷存储		

功能	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
k-NN 的 Hamming 距离、L1 Norm 距离和 Painless 脚本	1.0	7.10
异步搜索		
索引转换	1.0	不包括
跨集群复制	1.1	7.10
ML Commons	1.3	不包括
通知	2.3	不包括
时间点搜索	2.5	不包括
机器学习连接器	2.9	不包括
语义搜索	2.9	不包括
多模态语义搜索	2.1.1	不包括
直接查询数据源	2.1.1	不包括
并发分段搜索	2.13	不包括
自然语言查询生成	2.13	不包括

功能	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
Amazon Q 支持	2.17	不包括

有关支持其中的部分功能和其他功能的插件的信息，请参阅the section called “插件（按引擎版本）”。有关每个版本的 OpenSearch API 的更多信息，请参阅 the section called “支持的操作”。

Amazon OpenSearch 服务中按引擎版本划分的插件

Amazon OpenSearch 服务域名预先打包了来自 OpenSearch 社区的插件。该服务会自动为您部署和管理插件，但它会根据您为域选择的旧版 Elasticsearch OSS 版本部署不同的插件。 OpenSearch

下表按 OpenSearch 版本列出了插件以及旧版 Elasticsearch OSS 的兼容版本。它只包括你可能与之交互的插件，并不全面。 OpenSearch 服务使用其他插件来启用核心服务功能，例如用于快照的 S3 存储库插件和用于优化和监控的 Perfor OpenSearchmance Analyzer 插件。有关域上运行的所有插件的完整列表，请提出以下请求：

```
GET _cat/plugins?v
```

插件	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
HanLP	2.1.1	不支持
希伯来语分析	2.1.1	不支持
Amazon Personalize 搜索排名	2.9	不支持
神经搜索	2.9	不支持
安全分析	2.5	不支持
OpenSearch notifications	2.3	不支持

插件	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
ML Commons	1.3	不支持
Sudachi 分析（推荐用于日语）	1.3	不支持
STConvert	1.3	不支持
拼音分析	1.3	不支持
Nori 分析	1.3	不支持
OpenSearch 可观察性	1.2	不支持
OpenSearch 跨集群复制	1.1	7.10
OpenSearch 异步搜索	1.0	7.10
IK（中文）分析	1.0	7.7
越南语分析		
泰语分析		
学习排名		
OpenSearch 异常检测	1.0	7.4
OpenSearch k-NN	1.0	7.1

插件	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
OpenSearch 索引状态管理	1.0	6.8
OpenSearch 安全	1.0	6.7
OpenSearch SQL	1.0	6.5
OpenSearch 警报	1.0	6.2
乌克兰语分析	1.0	5.3
映射器大小	1.0	5.3
映射器 Murmur3	1.0	5.1
Ingest 用户代理处理器	1.0	5.1
Ingest Attachment 处理器	1.0	5.1
Stempel Polish 分析	1.0	5.1
智能中文分析	1.0	5.1
Seunjeon 韩语分析	1.0	5.1
拼音分析	1.0	2.3

插件	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
日语 (kuromoji) 分析	1.0	包含在所有域中
ICU 分析	1.0	包含在所有域中

可选插件

除了预先安装的默认插件外，Amazon S OpenSearch ervice 还支持多个可选的语言分析器插件。您可以使用 Amazon Web Services Management Console 和将插件 Amazon CLI 与域关联、取消插件与域的关联以及列出所有插件。可选的插件包与特定 OpenSearch 版本兼容，并且只能与具有该版本的域名相关联。

请注意，对于 Sudachi 插件，当您重新关联字典文件时，它不会立即反映到域中。当下一次 blue/green 部署作为配置更改或其他更新的一部分在域上运行时，字典会刷新。您也可以使用更新后的数据创建新的程序包，使用此新程序包创建新索引，利用现有索引重新创建新索引，然后删除旧索引。如果您更喜欢使用重新编制索引的方法，请使用索引别名，这样您的流量就不会受到干扰。

可选插件使用 `ZIP-PLUGIN` 程序包类型。有关可选插件的更多信息，请参阅 the section called “程序包”。

Amazon OpenSearch 服务中支持的操作

OpenSearch 该服务支持许多版本 OpenSearch 和传统的 Elasticsearch OSS。以下各节显示了 OpenSearch 服务为每个版本支持的操作。

主题

- 值得注意的 API 差异

值得注意的 API 差异

新名单 APIs

为了支持具有大量索引和分片的大型集群，我们引入了支持分页 APIs 的新 List，例如 _。list/indices and _list/shardsList API 以分页格式检索有关索引和分片的统计信息。这简化了处理包含许多索引的响应的任务。

- `_list/indices`: _list/indices
- `_list/shards`: _list/shards

对现有内容的更改 APIs

为了支持大型集群，我们在 `_cluster/stats` API 中增加了对添加额外指标筛选器的支持，以支持仅检索相关的统计数据响应，例如`_cluster/stats/<metric>/nodes/<node-filters>`和`_cluster/stats/<metric>/<index_metric>/nodes/<node-filters>`。有关详细信息，请参阅 _clu ster/stats。

我们在 `_cat/shards` API 中添加了通过指定`cancel_after_time_interval`请求参数来取消任务的支持。有关详细信息，请参阅 _cat /shards。

限制 _cat API 的响应大小

为了支持跨数据和温节点的实例总数超过 200 的大型集群，我们对返回的索引数量设定了 10K 的限制。`_cat/segments` API如果响应中的索引数量超过此限制，API 将返回 429 错误。为避免这种情况，您可以在查询中指定索引模式筛选器，例如`_cat/segments/<index-pattern>`。

设置和统计数据

OpenSearch 服务仅接受使用 “flat” 设置表单的 `_cluster/settings` API 的 PUT 请求。它拒绝使用扩展设置表单的请求。

```
// Accepted
PUT _cluster/settings
{
  "persistent" : {
    "action.auto_create_index" : false
  }
}

// Rejected
PUT _cluster/settings
{
  "persistent": {
    "action": {
      "auto_create_index": false
    }
  }
```

```
}
```

高级 Java REST 客户端使用扩展表单，因此，如果您需要发送设置请求，请使用低级客户端。

在 Elasticsearch 5.3 之前， OpenSearch 服务域上`_cluster/settings`的 API 仅支持 HTTP PUT 方法，不支持该`GET`方法。 OpenSearch 以及更高版本的 Elasticsearch 都支持该`GET`方法，如以下示例所示：

```
GET https://domain-name.region.es.amazonaws.com/_cluster/settings?pretty
```

这是一个返回示例：

```
{
  "persistent": {
    "cluster": {
      "routing": {
        "allocation": {
          "cluster_concurrent_rebalance": "2",
          "node_concurrent_recoveries": "2",
          "disk": {
            "watermark": {
              "low": "1.35gb",
              "flood_stage": "0.45gb",
              "high": "0.9gb"
            }
          },
          "node_initial_primarirecoveries": "4"
        }
      }
    },
    "indices": {
      "recovery": {
        "max_bytper_sec": "40mb"
      }
    }
  }
}
```

如果您比较来自开源 OpenSearch 集群和 OpenSearch 服务的响应以获取某些设置和统计信息 APIs，您可能会注意到缺少字段。 OpenSearch 服务会编辑暴露服务内部信息的某些信息，例如来自的文件系统数据路径`_nodes/stats`或来自的操作系统名称和版本。`_nodes`

收缩

_shrink API 可能导致升级、配置更改和域删除操作失败。建议不要在运行 Elasticsearch 版本 5.3 或 5.1 的域上使用它。这些版本上存在的错误可能会导致收缩索引的快照还原操作失败。

如果您在其他 Elasticsearch 或 OpenSearch版本上使用_shrink该 API，请在开始缩小操作之前提出以下请求：

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": "name-of-the-node-to-shrink-to",
    "index.blocks.read_only": true
  }
}
```

然后，在收缩操作完成后发出以下请求：

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}

PUT https://domain-name.region.es.amazonaws.com/shrunken-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}
```

新名单 APIs

为了支持具有大量索引和分片的大型集群，我们引入了支持分页的新列表 APIs ，即_list/indices和。_list/shardsList API 以分页格式检索有关索引和分片的统计信息。这简化了处理包含许多索引的响应的任务。有关的更多信息_list/indices，请参阅列出索引。有关的更多信息_list/shards，请参阅列出分片。

对现有内容的更改 APIs

为了支持大型集群，我们在`_cluster/stats/<metric>/nodes/<node-filters>`和中添加了支持`_cluster/stats/<metric>/<index_metric>/nodes/<node-filters>`。有关的更多信息`_cluster/stats`，请参阅集群统计信息。

限制 _cat 的响应大小 APIs

为了支持跨数据和温节点的实例总数超过 200 的大型集群，我们对 _cat/segments API 返回的索引数量设定了 10,000 个限制。如果响应中的索引数量超过此限制，API 将返回429错误。为避免这种情况，您可以在查询中指定索引模式筛选器（例如，`_cat/segments/<index-pattern>`）。

此外，现在还支持通过指定`cancel_after_time_interval`请求参数取消任务的 `_cat/shards` API。有关这方面的更多信息，请参阅 CAT 分片。

为专用主节点选择实例类型

下表提供了为专用主节点选择适当实例类型的建议：

RAM	支持的最大节点数	支持的最大分片
2 GB	10	1000
4 GB	10	5000
8 GB	30	15000
16 GB	60	30000
32 GB	120	60000
64 GB	240	120,000
128 GB	480	240,000
256GB	1002	500,000

OpenSearch 版本 2.18 和 2.19

有关 OpenSearch 2.18 和 2.19 操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。有关这些版本中变更的更多详细信息，请参阅 2.18 发行说明和 2.19 发行说明。

OpenSearch 版本 2.17

对于 OpenSearch 2.17， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

> **Note**
>
> 从 OpenSearch 2.17 开始，该`cluster.max_shards_per_node`设置无法修改。对于 OpenSearch 2.17 及更高版本， OpenSearch 服务支持每 16GB 的 JVM 堆内存使用 1000 个分片，每个节点最多支持 4000 个分片。

- 索引路径中的所有操作（例如 `/index-name /_forcemerge`、`/index-name /update/id` 和 `/index-name /_close`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- 适用于多个属性的 `/_cluster/settings` [4]：
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_list`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugins/_asynchronous_search`
- `/_plugins/_alerting`
- `/_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`

- 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
- search.phase_took_enabled
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。

2. 对具有消息正文的 `/_search/scroll` 的 DELETE 请求必须在 HTTP 标头中指定 `"Content-Length"`。默认情况下，大多数客户端会添加此标头。为避免`scroll_id`值中的=字符出现问题，请使用请求正文（而不是查询字符串）将`scroll_id`值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考PUT方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括针对异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

> **Note**
>
> 目前，对于具有备用多可用区（可用区）的客户，不允许更改`cluster.max_shards_per_node`设置功能。

OpenSearch 版本 2.15

对于 OpenSearch 2.15， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

- 索引路径中的所有操作（例如 `/index-name /_forcemerge`、`/index-name /update/id` 和 `/index-name /_close`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugins/_asynchronous_search`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search/pipeline`
- `/_search/point_in_time`
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`

- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
  - cluster.search.request.slowlog.threshold.info
  - cluster.search.request.slowlog.threshold.debug
  - cluster.search.request.slowlog.threshold.trace
  - search.phase_took_enabled
- /_cluster/state
- /_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
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

• /_cluster/stats • /_count • /_dashboards		

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 2.13

对于 OpenSearch 2.13， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

• 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close） • /_alias • /_aliases • /_all • /_analyze • /_bulk • /_cat（/_cat/nodeattrs 除外）	• /_delete_by_query [1] • /_explain • /_field_caps • /_field_stats • /_flush • /_ingest/pipeline • /_ltr • /_mapping • /_mget • /_msearch • /_mtermvectors • /_nodes	• /_refresh • /_reindex [1] • /_render • /_resolve/index • /_rollover • /_scripts [3] • /_search[2] • /_search/pipeline • /_search/point_in_time • /_search profile • /_shard_stores

- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
  - cluster.search.request.slowlog.threshold.info
  - cluster.search.request.slowlog.threshold.debug
  - cluster.search.request.slowlog.threshold.trace
- /_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
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

• search.phase_took_enabled • /_cluster/state • /_cluster/stats • /_count • /_dashboards		

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 2.11

对于 OpenSearch 2.11， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

• 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close） • /_alias • /_aliases • /_all • /_analyze	• /_delete_by_query [1] • /_explain • /_field_caps • /_field_stats • /_flush • /_ingest/pipeline • /_ltr • /_mapping • /_mget	• /_refresh • /_reindex [1] • /_render • /_resolve/index • /_rollover • /_scripts [3] • /_search[2] • /_search/pipeline

- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_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. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。

4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 2.9

对于 OpenSearch 2.9， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

- 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_plugins/_notifications
- /_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

- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 2.7

对于 OpenSearch 2.7， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

- 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

update/*id* 和 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 `/_tasks` 操作，以验证请求是否成功完成。
2. 对具有消息正文的 `/_search/scroll` 的 DELETE 请求必须在 HTTP 标头中指定 `"Content-Length"`。默认情况下，大多数客户端会添加此标头。为避免`scroll_id`值中的=字符出现问题，请使用请求正文（而不是查询字符串）将`scroll_id`值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 2.5

对于 OpenSearch 2.5， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

- 索引路径中的所有操作（例如 `/index-name /_forcemerge` 、`/index-name /update/id` 和 `/index-name /_close`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugins/_asynchronous_search`
- `/_plugins/_alerting`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search/point_in_time`
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`

- 适用于多个属性的 /_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/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 2.3

对于 OpenSearch 2.3， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

- 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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/_sql
- /_percolate
- /_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	• /_rank_eval	

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 1.3

对于 OpenSearch 1.3， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

• 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close） • /_alias • /_aliases • /_all • /_analyze	• /_delete_by_query [1] • /_explain • /_field_caps • /_field_stats • /_flush • /_ingest/pipeline • /_ltr • /_mapping • /_mget	• /_refresh • /_reindex [1] • /_render • /_resolve/index • /_rollover • /_scripts [3] • /_search[2] • /_search profile • /_shard_stores

- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。

4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 1.2

对于 OpenSearch 1.2， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

- 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_plugins/_security
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

• indices.breaker.fielddata.limit • indices.breaker.request.limit • indices.breaker.total.limit • cluster.max_shards_per_node • /_cluster/state • /_cluster/stats • /_count • /_dashboards	• /_plugins/_sql • /_percolate • /_rank_eval	

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 1.1

对于 OpenSearch 1.1， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

• 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /	• /_delete_by_query [1] • /_explain • /_field_caps	• /_refresh • /_reindex [1] • /_render

update/*id* 和 /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs 除外)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 `/_tasks` 操作，以验证请求是否成功完成。
2. 对具有消息正文的 `/_search/scroll` 的 DELETE 请求必须在 HTTP 标头中指定 `"Content-Length"`。默认情况下，大多数客户端会添加此标头。为避免`scroll_id`值中的=字符出现问题，请使用请求正文（而不是查询字符串）将`scroll_id`值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

OpenSearch 版本 1.0

对于 OpenSearch 1.0， OpenSearch 服务支持以下操作。有关大多数操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。

- 索引路径中的所有操作（例如 `/index-name /_forcemerge` 、`/index-name /update/id` 和 `/index-name /_close`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugins/_asynchronous_search`
- `/_plugins/_alerting`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`

- 适用于多个属性的 /_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/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_plugins/_transforms
- /_percolate
- /_rank_eval
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 7.10 版

对于 Elasticsearch 7.10， OpenSearch 服务支持以下操作。

- 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_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
- /_opendistro/_asynchronous_search
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_plugins/_replication
- /_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

• /_count		

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。
6. 旧索引模板 (_template) 被替换为以 Elasticsearch 7.8 为开头的可组合模板 (_index_template)。可组合模板优先于旧模板。如果没有可组合模板匹配给定索引，则旧模板仍然可以匹配并应用。该_template操作仍然适用于 OpenSearch 及更高版本的 Elasticsearch OSS，但是对这两种模板类型的 GET 调用会返回不同的结果。

Elasticsearch 7.9 版

对于 Elasticsearch 7.9， OpenSearch 服务支持以下操作。

• 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close）	• /_delete_by_query [1]	• /_refresh
• /_alias	• /_explain	• /_reindex [1]
• /_aliases	• /_field_caps	• /_render
• /_all	• /_field_stats	• /_resolve/index
• /_analyze	• /_flush	• /_rollover
• /_bulk	• /_index_template [6]	• /_scripts [3]
• /_cat（/_cat/nodeattrs 除外）	• /_ingest/pipeline	• /_search[2]
	• /_ltr	• /_search profile
	• /_mapping	• /_shard_stores
	• /_mget	• /_shrink[5]
	• /_msearch	• /_snapshot

- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的一般 OpenSearch 操作，不包括异常检测、ISM 等特定插件支持的操作。
5. 请参阅the section called “收缩”。

6. 旧索引模板 (`_template`) 被替换为以 Elasticsearch 7.8 为开头的可组合模板 (`_index_template`)。可组合模板优先于旧模板。如果没有可组合模板匹配给定索引，则旧模板仍然可以匹配并应用。该`_template`操作仍然适用于 OpenSearch 及更高版本的 Elasticsearch OSS，但是对这两种模板类型的 GET 调用会返回不同的结果。

Elasticsearch 7.8 版

对于 Elasticsearch 7.8， OpenSearch 服务支持以下操作。

- 索引路径中的所有操作（例如 `/index-name /_forcemerge`、`/index-name /update/id` 和 `/index-name /_close`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- 适用于多个属性的 `/_cluster/settings` [4]：
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_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/_security`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_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.request.limit • indices.breaker.total.limit • cluster.max_shards_per_node	• /_opendistro/_sql • /_percolate • /_plugin/kibana • /_rank_eval	

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。
6. 旧索引模板 (_template) 被替换为以 Elasticsearch 7.8 为开头的可组合模板 (_index_template)。可组合模板优先于旧模板。如果没有可组合模板匹配给定索引，则旧模板仍然可以匹配并应用。该_template操作仍然适用于 OpenSearch 及更高版本的 Elasticsearch OSS，但是对这两种模板类型的 GET 调用会返回不同的结果。

Elasticsearch 7.7 版

对于 Elasticsearch 7.7， OpenSearch 服务支持以下操作。

• 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close） • /_alias • /_aliases	• /_cluster/state • /_cluster/stats • /_count • /_delete_by_query [1] • /_explain • /_field_caps	• /_refresh • /_reindex [1] • /_render • /_rollover • /_scripts [3] • /_search[2]

- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。

4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 7.4 版

对于 Elasticsearch 7.4， OpenSearch 服务支持以下操作。

- 索引路径中的所有操作（例如 /*index-name* /_forcemerge 、/*index-name* /update/*id* 和 /*index-name* /_close）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_cluster/settings [4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

• indices.breaker.request.limit • indices.breaker.total.limit • cluster.max_shards_per_node	• /_plugin/kibana • /_rank_eval	

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 7.1 版

对于 Elasticsearch 7.1， OpenSearch 服务支持以下操作。

• 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*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 • /_mapping	• /_refresh • /_reindex [1] • /_render • /_rollover • /_scripts [3] • /_search[2] • /_search profile • /_shard_stores • /_shrink[5] • /_snapshot

- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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

- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval

- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 6.8 版

对于 Elasticsearch 6.8， OpenSearch 服务支持以下操作。

- 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

• cluster.max_shards_per_node • cluster.blocks.read_only		

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 6.7 版

对于 Elasticsearch 6.7， OpenSearch 服务支持以下操作。

• 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*） • /_alias • /_aliases • /_all • /_analyze • /_bulk • /_cat（/_cat/nodeattrs 除外）	• /_cluster/state • /_cluster/stats • /_count • /_delete_by_query [1] • /_explain • /_field_caps • /_field_stats • /_flush • /_ingest/pipeline • /_mapping • /_mget • /_msearch	• /_refresh • /_reindex [1] • /_render • /_rollover • /_scripts [3] • /_search[2] • /_search profile • /_shard_stores • /_shrink[5] • /_snapshot • /_split • /_stats

- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 6.5 版

对于 Elasticsearch 6.5， OpenSearch 服务支持以下操作。

- 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。

2. 对具有消息正文的 `/_search/scroll` 的 DELETE 请求必须在 HTTP 标头中指定 `"Content-Length"`。默认情况下，大多数客户端会添加此标头。为避免`scroll_id`值中的`=`字符出现问题，请使用请求正文（而不是查询字符串）将`scroll_id`值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 6.4 版

对于 Elasticsearch 6.4， OpenSearch 服务支持以下操作。

- 索引路径中除了 `/index-name /_close` 以外的所有操作（例如 `/index-name /_forcemerge` 和 `/index-name /update/id`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- 适用于多个属性的 `/_cluster/settings` [4]：
  - `action.auto_create_index`
- `/_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`

• action.search.shard_count.limit • indices.breaker.fielddata.limit • indices.breaker.request.limit • indices.breaker.total.limit		

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 6.3 版

对于 Elasticsearch 6.3， OpenSearch 服务支持以下操作。

• 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*） • /_alias • /_aliases • /_all • /_analyze	• /_cluster/state • /_cluster/stats • /_count • /_delete_by_query [1] • /_explain • /_field_caps • /_field_stats • /_flush	• /_refresh • /_reindex [1] • /_render • /_rollover • /_scripts [3] • /_search[2] • /_search profile • /_shard_stores

- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_cluster/settings [4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 6.2 版

对于 Elasticsearch 6.2， OpenSearch 服务支持以下操作。

- 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 `/_tasks` 操作，以验证请求是否成功完成。
2. 对具有消息正文的 `/_search/scroll` 的 DELETE 请求必须在 HTTP 标头中指定 `"Content-Length"`。默认情况下，大多数客户端会添加此标头。为避免`scroll_id`值中的=字符出现问题，请使用请求正文（而不是查询字符串）将`scroll_id`值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 6.0 版

对于 Elasticsearch 6.0， OpenSearch 服务支持以下操作。

- 索引路径中除了 `/index-name /_close` 以外的所有操作（例如 `/index-name /_forcemerge` 和 `/index-name /update/id`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- 适用于多个属性的 `/_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`
- `/_percolate`
- `/_plugin/kibana`
- `/_refresh`
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search` [2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink` [5]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

• action.auto_create_index • action.search.shard_count.limit • indices.breaker.fielddata.limit • indices.breaker.request.limit • indices.breaker.total.limit	• /_reindex [1]	

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 5.6 版

对于 Elasticsearch 5.6， OpenSearch 服务支持以下操作。

• 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*） • /_alias • /_aliases	• /_cluster/state • /_cluster/stats • /_count • /_delete_by_query [1] • /_explain • /_field_caps	• /_render • /_rollover • /_scripts [3] • /_search[2] • /_search profile • /_shard_stores

- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_cluster/settings [4]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 5.5 版

对于 Elasticsearch 5.5， OpenSearch 服务支持以下操作。

- 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_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
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 `/_tasks` 操作，以验证请求是否成功完成。
2. 对具有消息正文的 `/_search/scroll` 的 DELETE 请求必须在 HTTP 标头中指定 `"Content-Length"`。默认情况下，大多数客户端会添加此标头。为避免`scroll_id`值中的=字符出现问题，请使用请求正文（而不是查询字符串）将`scroll_id`值传递给 Ser OpenSearch vice。
3. 有关使用脚本的注意事项，请参阅the section called “其他支持的资源”。
4. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
5. 请参阅the section called “收缩”。

Elasticsearch 5.3 版

对于 Elasticsearch 5.3， OpenSearch 服务支持以下操作。

- 索引路径中除了 `/index-name /_close` 以外的所有操作（例如 `/index-name /_forcemerge` 和 `/index-name /update/id`）
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`（`/_cat/nodeattrs` 除外）
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- 适用于多个属性的 `/_cluster/settings`[3]：
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query`[1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_percolate`
- `/_plugin/kibana`
- `/_refresh`
- `/_render`
- `/_rollover`
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[4]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query`[1]
- `/_validate`

• action.auto_create_index • action.search.shard_count.limit • indices.breaker.fielddata.limit • indices.breaker.request.limit • indices.breaker.total.limit	• /_reindex [1]	

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 请参考 PUT 方法。有关 GET 方法的信息，请参阅the section called “值得注意的 API 差异”。此列表仅指 OpenSearch 服务支持的通用 Elasticsearch 操作，不包括针对异常检测、ISM 等的插件特定支持的操作。
4. 请参阅the section called “收缩”。

Elasticsearch 5.1 版

对于 Elasticsearch 5.1， OpenSearch 服务支持以下操作。

• 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /update/*id*） • /_alias • /_aliases • /_all	• /_cluster/state • /_cluster/stats • /_count • /_delete_by_query [1] • /_explain • /_field_caps • /_field_stats	• /_render • /_rollover • /_search[2] • /_search profile • /_shard_stores • /_shrink[3] • /_snapshot

- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于若干属性的 /_cluster/settings （仅 PUT）：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 集群配置更改可能会在这些操作完成前将其中断。建议您与这些操作一起使用 /_tasks 操作，以验证请求是否成功完成。
2. 对具有消息正文的 /_search/scroll 的 DELETE 请求必须在 HTTP 标头中指定 "Content-Length"。默认情况下，大多数客户端会添加此标头。为避免scroll_id值中的=字符出现问题，请使用请求正文（而不是查询字符串）将scroll_id值传递给 Ser OpenSearch vice。
3. 请参阅the section called “收缩”。

Elasticsearch 2.3 版

对于 Elasticsearch 2.3， OpenSearch 服务支持以下操作。

- 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-name* /_forcemerge 和 /*index-name* /_recovery ）
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cache/clear （仅索引）
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/health
- 适用于若干属性的 /_cluster/settings (仅 PUT)：
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - threadpool.get.queue_size
  - threadpool.bulk.queue_size
  - threadpool.index.queue_size
  - threadpool.percolate.queue_size
  - threadpool.search.queue_size
  - threadpool.suggest.queue_size
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_render
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

Elasticsearch 1.5 版

对于 Elasticsearch 1.5， OpenSearch 服务支持以下操作。

- 索引路径中除了 /*index-name* /_close 以外的所有操作（例如 /*index-nam*
- /_cluster/stats

e /_optimize 和 /index-name /_warmer)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat
- /_cluster/health
- 适用于若干属性的 /_cluster/settings (仅 PUT)：
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - threadpool.get.queue_size
  - threadpool.bulk.queue_size
  - threadpool.index.queue_size
  - threadpool.percolate.queue_size
  - threadpool.search.queue_size
  - threadpool.suggest.queue_size
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_plugin/kibana3
- /_plugin/migration
- /_refresh
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

亚马逊 OpenSearch 服务配额

您的 Amazon 账户对每项 Amazon 服务都有默认配额（以前称为限制）。除非另有说明，否则，每个配额是区域特定的。

要查看 OpenSearch 服务域和实例、Amazon OpenSearch Serverl OpenSearch ess 和 Amazon Ingestion 的配额，请参阅中的亚马逊 OpenSearch 服务配额。Amazon Web Services 一般参考

要在中查看 OpenSearch 服务配额 Amazon Web Services Management Console，请打开 Serv ice Quotas 控制台。在导航窗格中，选择Amazon 服务，然后选择亚马逊 OpenSearch服务。要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。

UltraWarm 存储配额

下表列出了 UltraWarm 实例类型和每种类型可以使用的最大存储量。有关的更多信息 UltraWarm，请参阅the section called “UltraWarm 存储”。

实例类型	最大存储空间
ultrawarm 1.medium.se	1.5 TiB
ultrawarm 1.large.sear	20 TiB

每个 AZ 的数据节点数

下表列出了可用区部署的数据节点总数，总限制表示每个限制的数据节点数量，包括热节点和温节点数量。每种类型可以使用的存储。

可用区配置	热点节点数量限制	热节点数量限制	总极限（热 + 热）
1-AZ	334	250	334
2-AZ	668	500	668
3-AZ	1002	750	1002

按实例系列划分的节点总数限制

下表列出了按实例系列划分的节点总数限制。

实例系列	ElasticSearch OpenSearch 最高 2.15	OpenSearch 2.17 及更高版本	默认限制
T2	10	10	10
T3	10	10	10
M3、C4、M4、R4、C5、M5、R5、I2、I3	10	200	80
Graviton 2、Graviton 3	200	400	80
C7、r7i、m7i、i4i	200	400	80
OR1.medium.searc OR1.large.search OR2.medium.searc OR2.large.search OM2.large.search	200	400	80
OR1.xlarge.search 及更高版本 OR2.xlarge.search 及更高版本 OM2.xlarge.search 及更高版本	200	1002	80
Ultrawarm	150	750	150

EBS 卷大小配额

下表显示了 OpenSearch 服务支持的每种实例类型的 EBS 卷的最小和最大大小。有关哪些实例类型包括实例存储和其他硬件详情的信息，请参阅 Amazon S OpenSearch ervice 定价。

- 如果您在创建域时在 EBS 卷类型下选择磁存储，则除了 t2.small 和 t2.medium 以及所有不支持磁存储的 Graviton 实例（m6g、c6g、r6g 和 r6gD）之外，所有实例类型的最大卷大小均为 100 GiB。对于下表中列出的最大大小，选择 SSD 选项之一。
- 一些较旧版本的实例类型包含实例存储，但也支持 EBS 存储。如果您为这些实例类型之一选择了 EBS 存储，则不会 添加存储卷。您可以使用 EBS 卷或实例存储，但不能同时使用二者。

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
t2.micro.search	10 GiB	35 GiB	不适用
t2.small.searc	10 GiB	35 GiB	不适用
t2.medium.search	10 GiB	35 GiB	不适用
t3.small.searc	10 GiB	100GiB	100GiB
t3.medium.search	10 GiB	200 GiB	200 GiB
m3.medium.search	10 GiB	100GiB	不适用
m3.large.search	10 GiB	512 GiB	不适用
m3.xlarge.search	10 GiB	512 GiB	不适用
m3.2xlarge.search	10 GiB	512 GiB	不适用
m4.large.search	10 GiB	512 GiB	不适用
m4.xlarge.search	10 GiB	1 TiB	不适用
m4.2xlarge.search	10 GiB	1.5 TiB	不适用
m4.4xlarge.search	10 GiB	1.5 TiB	不适用
m4.10xlarge.search	10 GiB	1.5 TiB	不适用
m5.large.search	10 GiB	512 GiB	1 TiB
m5.xlarge.search	10 GiB	1 TiB	2 TiB

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
m5.2xlarge.search	10 GiB	1.5 TiB	3 TiB
m5.4xlarge.search	10 GiB	3 TiB	6 TiB
m5.12xlarge.search	10 GiB	9 TiB	18TiB
m6g.large.search	10 GiB	512 GiB	1 TiB
m6g.xlarge.search	10 GiB	1 TiB	2 TiB
m6g.2xlarge.search	10 GiB	1.5 TiB	3 TiB
m6g.4xlarge.search	10 GiB	3 TiB	6 TiB
m6g.8xlarge.search	10 GiB	6 TiB	12 TiB
m6g.12xlarge.search	10 GiB	9 TiB	18TiB
c4.large.search	10 GiB	100GiB	不适用
c4.xlarge.search	10 GiB	512 GiB	不适用
c4.2xlarge.search	10 GiB	1 TiB	不适用
c4.4xlarge.search	10 GiB	1.5 TiB	不适用
c4.8xlarge.search	10 GiB	1.5 TiB	不适用
c5.large.search	10 GiB	256 GiB	256 GiB
c5.xlarge.search	10 GiB	512 GiB	512 GiB
c5.2xlarge.search	10 GiB	1 TiB	1 TiB
c5.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c5.9xlarge.search	10 GiB	3.5 TiB	3.5 TiB
c5.18xlarge.search	10 GiB	7 TiB	7 TiB

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
c6g.large.search	10 GiB	256 GiB	256 GiB
c6g.xlarge.search	10 GiB	512 GiB	512 GiB
c6g.2xlarge.search	10 GiB	1 TiB	1 TiB
c6g.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c6g.8xlarge.search	10 GiB	3 TiB	3 TiB
c6g.12xlarge.search	10 GiB	4.5 TiB	4.5 TiB
r3.large.search	10 GiB	512 GiB	不适用
r3.xlarge.search	10 GiB	512 GiB	不适用
r3.2xlarge.search	10 GiB	512 GiB	不适用
r3.4xlarge.search	10 GiB	512 GiB	不适用
r3.8xlarge.search	10 GiB	512 GiB	不适用
r4.large.search	10 GiB	1 TiB	不适用
r4.xlarge.search	10 GiB	1.5 TiB	不适用
r4.2xlarge.search	10 GiB	1.5 TiB	不适用
r4.4xlarge.search	10 GiB	1.5 TiB	不适用
r4.8xlarge.search	10 GiB	1.5 TiB	不适用
r4.16xlarge.search	10 GiB	1.5 TiB	不适用
r5.large.search	10 GiB	1 TiB	2 TiB
r5.xlarge.search	10 GiB	1.5 TiB	3 TiB
r5.2xlarge.search	10 GiB	3 TiB	6 TiB

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
r5.4xlarge.search	10 GiB	6 TiB	12 TiB
r5.12xlarge.search	10 GiB	12 TiB	24TiB
r6g.large.search	10 GiB	1 TiB	2 TiB
r6g.xlarge.search	10 GiB	1.5 TiB	3 TiB
r6g.2xlarge.search	10 GiB	3 TiB	6 TiB
r6g.4xlarge.search	10 GiB	6 TiB	12 TiB
r6g.8xlarge.search	10 GiB	8 TiB	16 TiB
r6g.12xlarge.search	10 GiB	12 TiB	24TiB
r6gd.large.search	不适用	不适用	不适用
r6gd.xlarge.search	不适用	不适用	不适用
r6gd.2xlarge.search	不适用	不适用	不适用
r6gd.4xlarge.search	不适用	不适用	不适用
r6gd.8xlarge.search	不适用	不适用	不适用
r6gd.12xlarge.search	不适用	不适用	不适用
r6gd.16xlarge.search	不适用	不适用	不适用
i2.xlarge.search	10 GiB	512 GiB	不适用
i2.2xlarge.search	10 GiB	512 GiB	不适用
i3.large.search	不适用	不适用	不适用
i3.xlarge.search	不适用	不适用	不适用
i3.2xlarge.search	不适用	不适用	不适用

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
i3.4xlarge.search	不适用	不适用	不适用
i3.8xlarge.search	不适用	不适用	不适用
i3.16xlarge.search	不适用	不适用	不适用
or1.medium.search	20 GiB	不适用	768 GiB
或者 1.large.search	20 GiB	不适用	1532 GiB
或者1.xlarge.search	20 GiB	不适用	3 TiB
或者 1.2xlarge.search	20 GiB	不适用	6 TiB
或者 1.4xlarge.search	20 GiB	不适用	12 TiB
或者 1.8xlarge.search	20 GiB	不适用	16 TiB
或者 1.12xlarge.search	20 GiB	不适用	24TiB
或者 1.16xlarge.search	20 GiB	不适用	36 TiB
or2.medium.search	20 GiB	不适用	768 GiB
or2. large.search	20 GiB	不适用	1532 GiB
or2.xlarge.search	20 GiB	不适用	3 TiB
或者2.2xlarge.search	20 GiB	不适用	6 TiB
或者 2.4xlarge.search	20 GiB	不适用	12 TiB
或者 2.8xlarge.search	20 GiB	不适用	16 TiB
或者2.12xlarge.search	20 GiB	不适用	24TiB
或者 2.16xlarge.search	20 GiB	不适用	36 TiB
om2.large.search	20 GiB	不适用	768 GiB

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
om2.xlarge.search	20 GiB	不适用	2 Tib
om2.2xlarge.search	20 GiB	不适用	3 Tib
om2.4xlarge.search	20 GiB	不适用	6 Tib
om2.8xlarge.search	20 GiB	不适用	12 Tib
om2.12xlarge.search	20 GiB	不适用	18 Tib
om2.16xlarge.search	20 GiB	不适用	24 Tib
im4gn.large.search	不适用	不适用	不适用
im4gn.xlarge.search	不适用	不适用	不适用
im4gn.2xlarge.search	不适用	不适用	不适用
im4gn.4xlarge.search	不适用	不适用	不适用
im4gn.8xlarge.search	不适用	不适用	不适用
im4gn.16xlarge.search	不适用	不适用	不适用
c7g.large.Search	10 GiB	不适用	256 GiB
c7g.xlarge.Search	10 GiB	不适用	512 GiB
c7g.2xLarge.Search	10 GiB	不适用	1 TiB
c7g.4xLarge.Search	10 GiB	不适用	1.5 TiB
c7g.8xlarge.Search	10 GiB	不适用	3 TiB
c7g.12xLarge.Search	10 GiB	不适用	4.5 TiB
c7g.16xLarge.Search	10 GiB	不适用	6 TiB
m7g.medium.Search	10 GiB	不适用	512 GiB

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
m7g.large.Search	10 GiB	不适用	768 GiB
m7g.xlarge.Search	10 GiB	不适用	2 TiB
m7g.2xLarge.Search	10 GiB	不适用	3 TiB
m7g.4xLarge.Search	10 GiB	不适用	6 TiB
m7g.8xlarge.Search	10 GiB	不适用	12 TiB
m7g.12xLarge.Search	10 GiB	不适用	18TiB
m7g.16xLarge.Search	10 GiB	不适用	24TiB
r7g.medium.Search	10 GiB	不适用	768 GiB
r7g.large.Search	10 GiB	不适用	1.5 TiB
r7g.xlarge.Search	10 GiB	不适用	3 TiB
r7g.2xLarge.Search	10 GiB	不适用	6 TiB
r7g.4xLarge.Search	10 GiB	不适用	12 TiB
r7g.8xlarge.Search	10 GiB	不适用	16 TiB
r7g.12xLarge.Search	10 GiB	不适用	24TiB
r7g.16xLarge.Search	10 GiB	不适用	36 TiB
r7gd.large.search	不适用	不适用	不适用
r7gd.xlarge.search	不适用	不适用	不适用
r7gd.2xLarge.Search	不适用	不适用	不适用
r7gd.4xLarge.search	不适用	不适用	不适用
r7gd.8xlarge.Search	不适用	不适用	不适用

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
r7gd.12xLarge.Search	不适用	不适用	不适用
r7gd.16xLarge.Search	不适用	不适用	不适用
i4i.large.search	10 GiB	不适用	不适用
i4i.xlarge.search	10 GiB	不适用	不适用
i4i.2xlarge.search	10 GiB	不适用	不适用
i4i.4xlarge.search	10 GiB	不适用	不适用
i4i.8xlarge.search	10 GiB	不适用	不适用
i4i.12xlarge.search	10 GiB	不适用	不适用
i4i.16xlarge.search	10 GiB	不适用	不适用
i4i.24xlarge.search	10 GiB	不适用	不适用
i4i.32xlarge.search	10 GiB	不适用	不适用
i4g.large.search	10 GiB	不适用	不适用
i4g.xlarge.search	10 GiB	不适用	不适用
i4g.2xlarge.search	10 GiB	不适用	不适用
i4g.4xlarge.search	10 GiB	不适用	不适用
i4g.8xlarge.search	10 GiB	不适用	不适用
i4g.16xlarge.search	10 GiB	不适用	不适用
c7i.large.search	10 GiB	不适用	256 GiB
c7i.xlarge.search	10 GiB	不适用	512 GiB
c7i.2xlarge.search	10 GiB	不适用	1 TiB

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
c7i.4xlarge.search	10 GiB	不适用	1.5 TiB
c7i.8xlarge.search	10 GiB	不适用	3 TiB
c7i.12xlarge.search	10 GiB	不适用	4.5 TiB
c7i.16xlarge.search	10 GiB	不适用	6 TiB
m7i.large.search	10 GiB	不适用	768 GiB
m7i.xlarge.search	10 GiB	不适用	2 TiB
m7i.2xlarge.search	10 GiB	不适用	3 TiB
m7i.4xlarge.search	10 GiB	不适用	6 TiB
m7i.8xlarge.search	10 GiB	不适用	12 TiB
m7i.12xlarge.search	10 GiB	不适用	18TiB
m7i.16xlarge.search	10 GiB	不适用	24TiB
r7i.large.search	10 GiB	不适用	1.5 TiB
r7i.xlarge.search	10 GiB	不适用	3 TiB
r7i.2xlarge.search	10 GiB	不适用	6 TiB
r7i.4xlarge.search	10 GiB	不适用	12 TiB
r7i.8xlarge.search	10 GiB	不适用	16 TiB
r7i.12xlarge.search	10 GiB	不适用	24TiB
r7i.12xlarge.search	10 GiB	不适用	36 TiB

网络配额

下表显示了 HTTP 请求负载的最大值。

实例类型	HTTP 请求负载的最大值
t2.micro.search	10 MiB
t2.small.searc	10 MiB
t2.medium.search	10 MiB
t3.small.searc	10 MiB
t3.medium.search	10 MiB
m3.medium.search	10 MiB
m3.large.search	10 MiB
m3.xlarge.search	100 MiB
m3.2xlarge.search	100 MiB
m4.large.search	10 MiB
m4.xlarge.search	100 MiB
m4.2xlarge.search	100 MiB
m4.4xlarge.search	100 MiB
m4.10xlarge.search	100 MiB
m5.large.search	10 MiB
m5.xlarge.search	100 MiB
m5.2xlarge.search	100 MiB
m5.4xlarge.search	100 MiB

实例类型	HTTP 请求负载的最大值
m5.12xlarge.search	100 MiB
m6g.large.search	10 MiB
m6g.xlarge.search	100 MiB
m6g.2xlarge.search	100 MiB
m6g.4xlarge.search	100 MiB
m6g.8xlarge.search	100 MiB
m6g.12xlarge.search	100 MiB
c4.large.search	10 MiB
c4.xlarge.search	100 MiB
c4.2xlarge.search	100 MiB
c4.4xlarge.search	100 MiB
c4.8xlarge.search	100 MiB
c5.large.search	10 MiB
c5.xlarge.search	100 MiB
c5.2xlarge.search	100 MiB
c5.4xlarge.search	100 MiB
c5.9xlarge.search	100 MiB
c5.18xlarge.search	100 MiB
c6g.large.search	10 MiB
c6g.xlarge.search	100 MiB

实例类型	HTTP 请求负载的最大值
c6g.2xlarge.search	100 MiB
c6g.4xlarge.search	100 MiB
c6g.8xlarge.search	100 MiB
c6g.12xlarge.search	100 MiB
r3.large.search	10 MiB
r3.xlarge.search	100 MiB
r3.2xlarge.search	100 MiB
r3.4xlarge.search	100 MiB
r3.8xlarge.search	100 MiB
r4.large.search	100 MiB
r4.xlarge.search	100 MiB
r4.2xlarge.search	100 MiB
r4.4xlarge.search	100 MiB
r4.8xlarge.search	100 MiB
r4.16xlarge.search	100 MiB
r5.large.search	100 MiB
r5.xlarge.search	100 MiB
r5.2xlarge.search	100 MiB
r5.4xlarge.search	100 MiB
r5.12xlarge.search	100 MiB

实例类型	HTTP 请求负载的最大值
r6g.large.search	100 MiB
r6g.xlarge.search	100 MiB
r6g.2xlarge.search	100 MiB
r6g.4xlarge.search	100 MiB
r6g.8xlarge.search	100 MiB
r6g.12xlarge.search	100 MiB
r6gd.large.search	100 MiB
r6gd.xlarge.search	100 MiB
r6gd.2xlarge.search	100 MiB
r6gd.4xlarge.search	100 MiB
r6gd.8xlarge.search	100 MiB
r6gd.12xlarge.search	100 MiB
r6gd.16xlarge.search	100 MiB
i2.xlarge.search	100 MiB
i2.2xlarge.search	100 MiB
i3.large.search	100 MiB
i3.xlarge.search	100 MiB
i3.2xlarge.search	100 MiB
i3.4xlarge.search	100 MiB
i3.8xlarge.search	100 MiB

实例类型	HTTP 请求负载的最大值
i3.16xlarge.search	100 MiB
or1.medium.search	10 MiB
或者 1.large.search	100 MiB
或者1.xlarge.search	100 MiB
或者 1.2xlarge.search	100 MiB
或者 1.4xlarge.search	100 MiB
或者 1.8xlarge.search	100 MiB
或者 1.12xlarge.search	100 MiB
或者 1.16xlarge.search	100 MiB
or2.medium.search	100 MiB
or2. large.search	100 MiB
or2.xlarge.search	100 MiB
或者2.2xlarge.search	100 MiB
或者 2.4xlarge.search	100 MiB
或者 2.8xlarge.search	100 MiB
或者2.12xlarge.search	100 MiB
或者 2.16xlarge.search	100 MiB
om2.large.search	100 MiB
om2.xlarge.search	100 MiB
om2.2xlarge.search	100 MiB

实例类型	HTTP 请求负载的最大值
om2.4xlarge.search	100 MiB
om2.8xlarge.search	100 MiB
om2.12xlarge.search	100 MiB
om2.16xlarge.search	100 MiB
im4gn.large.search	100 MiB
im4gn.xlarge.search	100 MiB
im4gn.2xlarge.search	100 MiB
im4gn.4xlarge.search	100 MiB
im4gn.8xlarge.search	100 MiB
im4gn.16xlarge.search	100 MiB
i4i.large.search	100 MiB
i4i.xlarge.search	100 MiB
i4i.2xlarge.search	100 MiB
i4i.4xlarge.search	100 MiB
i4i.8xlarge.search	100 MiB
i4i.12xlarge.search	100 MiB
i4i.16xlarge.search	100 MiB
i4i.24xlarge.search	100 MiB
i4i.32xlarge.search	100 MiB
i4g.large.search	100 MiB

实例类型	HTTP 请求负载的最大值
i4g.xlarge.search	100 MiB
i4g.2xlarge.search	100 MiB
i4g.4xlarge.search	100 MiB
i4g.8xlarge.search	100 MiB
i4g.16xlarge.search	100 MiB
c7i.large.search	100 MiB
c7i.xlarge.search	100 MiB
c7i.2xlarge.search	100 MiB
c7i.4xlarge.search	100 MiB
c7i.8xlarge.search	100 MiB
c7i.12xlarge.search	100 MiB
c7i.16xlarge.search	100 MiB
m7i.large.search	100 MiB
m7i.xlarge.search	100 MiB
m7i.2xlarge.search	100 MiB
m7i.4xlarge.search	100 MiB
m7i.8xlarge.search	100 MiB
m7i.12xlarge.search	100 MiB
m7i.16xlarge.search	100 MiB
r7i.large.search	100 MiB

实例类型	HTTP 请求负载的最大值
r7i.xlarge.search	100 MiB
r7i.2xlarge.search	100 MiB
r7i.4xlarge.search	100 MiB
r7i.8xlarge.search	100 MiB
r7i.12xlarge.search	100 MiB
r7i.16xlarge.search	100 MiB

分片大小配额

以下各节列出了各种实例系列的最大分片大小。

实例类型	不带待机功能的多可用区	带待机功能的多可用区
R5、C5、M5	不适用	65 GiB
I3	不适用	65 GiB
R6g、C6g、M6g、R6gd	不适用	65 GiB
OR1, OR2, OM2	100GiB	65 GiB
Im4gn	不适用	65 GiB

要请求提高配额，请联系 Amazon Support。

分片计数配额

以下部分列出了 OpenSearch 版本的最大分片数。

引擎版本	限制	备注
Elasticsearch 1.5 到 6.x	没有默认限制	

引擎版本	限制	备注
Elasticsearch 7.x	1000	可以通过集群更改默认限制。max_shards_per_node 设置。
OpenSearch 1.x 到 2.15	1000	可以通过集群更改默认限制。max_shards_per_node 设置。
OpenSearch 2.17 及更高版本	每 16 GB 堆中有 1000 个，最大值为 4000	无法更改默认限制。

Java 进程配额

OpenSearch 服务将 Java 进程的堆大小限制为 32 GiB。高级用户可指定用于字段数据的堆的百分比。有关更多信息，请参阅the section called “高级集群设置”和the section called “JVM OutOfMemoryError”。

域策略配额

OpenSearch 服务将域的访问策略限制在 100 KiB 以内。

Amazon S OpenSearch ervice 预留实例

与标准按需实例相比，Amazon S OpenSearch ervice 预留实例 (RIs) 提供大幅折扣。实例本身相同；RIs 只是对账户中使用的按需型实例所应用的账单折扣。对于使用量可预测的长寿命应用程序，随着时间的推移， RIs 可以节省大量资金。

OpenSearch 此项服务 RIs 需要一年期或三年期并且具有影响折扣率的三个付款选项：

- 无费用预付 – 无需支付任何预付费用。期限内，每小时按已折扣的每小时费率收费。
- 预付部分费用 – 提前支付部分费用，期限内，每小时按已折扣的每小时费率收费。
- 预付全费 – 提前支付全部费用。期限内，不再按每小时费率收费。

一般而言，预付款越多意味着折扣越大。无法取消预留实例—预留它们时，您承诺支付整个期限的费用并且预付款不可退款。

RIs 不灵活；它们只适用于您预留的确切实例类型。例如，8 个 `c5.2xlarge.search` 实例预留不适用于 16 个 `c5.xlarge.search` 实例或 4 个 `c5.4xlarge.search` 实例。但是，只要实例类型、区域、系列和大小相匹配，属于某个组织的关联账户 Amazon Organizations 就可以从预留实例拥有的账户中使用任何未使用的折扣申请。有关更多信息，请参阅 Amazon OpenSearch 服务定价和常见问题解答。

购买预留实例（控制台）

利用控制台，可查看现有预留实例和购买新的预留实例。

购买预留实例

1. 转到 https://aws.amazon.com，然后选择 Sign In to the Console (登录控制台)。
2. 在 “分析” 下，选择 “亚马逊 OpenSearch 服务”。
3. 从导航窗格中选择 Reserved Instance Leases（预留实例租赁）。

   在此页面上，可以查看现有预留。如果具有许多预留，可以筛选它们以更轻松地识别和查看特定预留。

   > **Tip**
   > 如果您未看到 Reserved Instance Leases（预留实例租赁）链接，请在 Amazon Web Services 区域中创建域。

4. 选择 Order Reserved Instance（对预留实例排序）。
5. 提供唯一的描述性名称。
6. 选择实例类型和实例数量。有关指南，请参阅the section called “调整域大小”。
7. 选择期限长度和付款选项。仔细阅读付款详细信息。
8. 选择下一步。
9. 仔细阅读购买摘要。购买的预留实例不可退款。
10. 选择排序。

购买预留实例 (Amazon CLI)

Amazon CLI 具有用于查看优惠、购买预订和查看预订的命令。以下命令和示例响应显示指定的产品 Amazon Web Services 区域：

```
aws opensearch describe-reserved-instance-offerings --region us-east-1
{
  "ReservedInstanceOfferings": [
    {
      "FixedPrice": x,
      "ReservedInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "UsagePrice": 0.0,
      "PaymentOption": "PARTIAL_UPFRONT",
      "Duration": 31536000,
      "InstanceType": "m4.2xlarge.search",
      "CurrencyCode": "USD"
    }
  ]
}
```

有关每个返回值的说明，请参阅下表。

字段	描述
FixedPrice	预留的前期成本。
ReservedInstanceOfferingId	产品 ID。如果要预留产品，则记下此值。
RecurringCharges	预留的小时费率。
UsagePrice	旧字段。对于 S OpenSearch ervice，此值始终为 0。
PaymentOption	无预付费用、预付部分费用或预付全费。
Duration	期限长度 (秒)： • 31536000 秒为一年。 • 94608000 秒为三年。

字段	描述
InstanceType	预留的实例类型。有关分配给每个实例类型的硬件资源的信息，请参阅 Amazon S OpenSearch ervice 定价。
CurrencyCode	FixedPrice 和 RecurringChargeAmount 的货币。

下一个示例购买预留实例：

```
aws opensearch purchase-reserved-instance-offering --reserved-instance-offering-
id 1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a --reservation-name my-reservation --instance-
count 3 --region us-east-1
{
  "ReservationName": "my-reservation",
  "ReservedInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a"
}
```

最后，您可以使用以下示例列出指定区域中的预留：

```
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` 为 Unix 纪元时间，其是自 1970 年 1 月 1 日午夜 (UTC) 以来经历的秒数。例如，1522872571 纪元时间为 2018 年 4 月 4 日 20:09:31 (UTC)。可以使用在线转换器。

要了解有关上述示例中所用命令的更多信息，请参阅 Amazon CLI 命令引用。

购买预留实例 (Amazon SDKs)

Amazon SDKs （除 Android 和 iOS 之外 SDKs）支持 Amazon S OpenSearch ervice API 引用中定义的所有操作，包括：

- `DescribeReservedInstanceOfferings`
- `PurchaseReservedInstanceOffering`
- `DescribeReservedInstances`

此示例脚本使用中的OpenSearchService低级别 Python 客户端购买预留实例。 适用于 Python (Boto3) 的 Amazon SDK 您必须为 `instance_type` 提供一个值。

```
import boto3
from botocore.config import Config

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-east-1'
)

client = boto3.client('opensearch', config=my_config)
```

```
instance_type = '' # e.g. m4.2xlarge.search


def describe_RI_offerings(client):
    """Gets the Reserved Instance offerings for this account"""

    response = client.describe_reserved_instance_offerings()
    offerings = (response['ReservedInstanceOfferings'])
    return offerings


def check_instance(offering):
    """Returns True if instance type is the one you specified above"""

    if offering['InstanceType'] == instance_type:
        return True

    return False


def get_instance_id():
    """Iterates through the available offerings to find the ID of the one you
 specified"""

    instance_type_iterator = filter(
        check_instance, describe_RI_offerings(client))
    offering = list(instance_type_iterator)
    id = offering[0]['ReservedInstanceOfferingId']
    return id


def purchase_RI_offering(client):
    """Purchase Reserved Instances"""

    response = client.purchase_reserved_instance_offering(
        ReservedInstanceOfferingId = get_instance_id(),
        ReservationName = 'my-reservation',
        InstanceCount = 1
    )
    print('Purchased reserved instance offering of type ' + instance_type)
    print(response)


def main():
```

```
    """Purchase Reserved Instances"""
    purchase_RI_offering(client)
```

有关安装和使用的更多信息 Amazon SDKs，请参阅Amazon 软件开发工具包。

调查费用

Cost Explorer 是一款免费工具，可用于查看过去 13 个月的支出数据。分析这些数据可以帮助您识别趋势并了解是否 RIs 适合您的用例。如果您已有 RIs，则可以按购买选项进行分组并显示摊销成本，以便将该支出与按需实例的支出进行比较。还可设置使用预算以确保您可以充分利用预留。有关更多信息，请参阅 Amazon Billing 用户指南中的使用 Cost Explorer 分析费用。

Amazon OpenSearch 服务中其他支持的资源

本主题介绍了 Amazon OpenSearch 服务支持的其他资源。

bootstrap.memory_lock

OpenSearch 服务`bootstrap.memory_lock`在中启用`opensearch.yml`，这会锁定 JVM 内存并防止操作系统将其交换到磁盘。这适用于所有受支持的实例类型，以下内容除外：

- `t2.micro.search`
- `t2.small.search`
- `t2.medium.search`
- `t3.small.search`
- `t3.medium.search`

脚本编写模块

OpenSearch 服务支持 Elasticsearch 5 的脚本编写。 x 及更高版本的域。该服务不支持为 1.5 或 2.3 编写脚本。

支持的脚本选项包括：

- Painless
- Lucene 表达式
- Mustache

对于 Elasticsearch 5.5 及更高版本的域以及所有 OpenSearch 域， OpenSearch 服务支持使用终端节点存储的`_scripts`脚本。Elasticsearch 5.3 和 5.1 域仅支持内联脚本。

TLS 传输

OpenSearch 服务支持端口 80 上的 HTTP 和端口 443 上的 HTTPS，但不支持 TLS 传输。

亚马逊 OpenSearch 服务教程

本章包括几个使用 Amazon S OpenSearch ervice 的 start-to-finish教程，包括如何迁移到该服务、构建简单的搜索应用程序以及在 OpenSearch 控制面板中创建可视化。

主题

- 教程：在 Amazon S OpenSearch ervice 中创建和搜索文档
- 教程：迁移到亚马逊 OpenSearch 服务
- 教程：使用 Amazon OpenSearch 服务创建搜索应用程序
- 教程：使用 S OpenSearch ervice 和 OpenSearch 控制面板可视化显示客户支持呼叫

教程：在 Amazon S OpenSearch ervice 中创建和搜索文档

在本教程中，您将了解如何在 Amazon S OpenSearch ervice 中创建和搜索文档。您将以 JSON 文档的形式将数据添加到索引。 OpenSearch Service 会围绕您添加的第一个文档创建索引。

本教程介绍如何发出 HTTP 请求以创建文档、自动生成文档 ID 以及如何对文档执行基本搜索和高级搜索。

> **Note**
>
> 本教程使用具有开放访问权限的域。为了获得最高级别的安全性，我们建议您将域置于虚拟私有云（VPC）内。

先决条件

本教程包含以下先决条件：

- 你必须有一个 Amazon Web Services 账户.
- 您必须具有一个活动的 S OpenSearch ervice 域。

将文档添加到索引

要将文档添加到索引，您可以使用任何 HTTP 工具，例如 P ostman、cURL 或 OpenSearch 控制面板控制台。这些示例假定您使用控制 OpenSearch 面板中的开发人员控制台。如果您使用其他工具，请提供完整的 URL 和凭证（如有必要）进行相应调整。

将文档添加到索引

1. 导航到域的 OpenSearch 控制面板 URL。您可以在 Ser OpenSearch vice 控制台中找到域的控制面板 URL。URL 遵循以下格式：

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

2. 使用您的主用户名和密码登录。
3. 打开左侧导航面板，然后选择 Dev Tools（开发人员工具）。
4. 用于创建新资源的 HTTP 动词是 PUT，您将使用它创建新文档和索引。在控制台中，输入以下命令：

   ```
   PUT fruit/_doc/1
   {
     "name":"strawberry",
     "color":"red"
   }
   ```

   PUT 请求创建一个名为 fruit（水果）的索引，并使用 ID 1 将单个文档添加到索引。它将生成以下响应：

   ```
   {
     "_index" : "fruit",
     "_type" : "_doc",
     "_id" : "1",
     "_version" : 1,
     "result" : "created",
     "_shards" : {
       "total" : 2,
       "successful" : 2,
       "failed" : 0
     },
     "_seq_no" : 0,
     "_primary_term" : 1
   ```

```
}
```

创建自动生成的 ID IDs

OpenSearch Service 可以自动为您的文档生成一个 ID。要 IDs 创建的 ID（与之前的请求相比）。

在开发人员控制台中，输入以下请求：

```
POST veggies/_doc
{
  "name":"beet",
  "color":"red",
  "classification":"root"
}
```

此请求将创建一个名为 veggies（蔬菜）的索引，并将文档添加到索引。它将生成以下响应：

```
{
  "_index" : "veggies",
  "_type" : "_doc",
  "_id" : "3WgyS4IB5DLqbRIvLxtF",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

请注意，`_id`响应中的其他 ID 表示自动创建的 ID。

> **Note**
>
> 您无需在 URL 中 _doc 之后提供任何内容，ID 通常位于该位置。因为您使用生成的 ID 创建文档，所以您无需提供 ID。这是为更新预留的。

使用 POST 命令更新文档

要更新文档，您可以通过 ID 号码使用 HTTP POST 命令。

首先，创建 ID 为 42 的文档：

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow"
}
```

然后，使用该 ID 更新文档：

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow",
  "classification":"berries"
}
```

此命令使用新字段 `classification` 更新文档。它将生成以下响应：

```
{
  "_index" : "fruits",
  "_type" : "_doc",
  "_id" : "42",
  "_version" : 2,
  "result" : "updated",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 1,
  "_primary_term" : 1
}
```

> **Note**
>
> 如果您尝试更新不存在的文档，S OpenSearch ervice 会创建文档。

执行批量操作

您可以在一个请求中使用 POST _bulk API 操作对一个或多个索引执行多个操作。批量操作命令采用以下形式：

```
POST /_bulk
<action_meta>\n
<action_data>\n
<action_meta>\n
<action_data>\n
```

每个操作都需要两行 JSON。首先，您需要提供操作描述或元数据。在下一行中，您需要提供数据。每个部分使用换行符（\n）分隔。插入的操作描述可能如下所示：

```
{ "create" : { "_index" : "veggies", "_type" : "_doc", "_id" : "7" } }
```

包含数据的下一行可能如下所示：

```
{ "name":"kale", "color":"green", "classification":"leafy-green" }
```

总的来说，元数据和数据表示批量操作中的单个操作。您可以在一个请求中执行许多操作，如下所示：

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "35" } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "36" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "37" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "38" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "39" } }
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
{ "delete" : { "_index" : "vegetables", "_id" : "1" } }
```

请注意，最后一个操作是 delete。delete 操作之后将没有数据。

搜索文档

现在，您的集群中已存在数据，您可以搜索这些数据。例如，您可能想要搜索所有根茎类蔬菜，或获取所有绿叶蔬菜的数量，亦或是查找每小时记录的错误数量。

基本搜索

基本搜索如下所示：

```
GET veggies/_search?q=name:l*
```

此请求生成包含生菜文档的 JSON 响应。

高级搜索

您可以在请求正文中以 JSON 形式提供查询选项，从而执行更高级的搜索：

```
GET veggies/_search
{
  "query": {
    "term": {
      "name": "lettuce"
    }
  }
}
```

此示例还使用生菜文档生成 JSON 响应。

排序

您可以使用排序执行更多此类查询。首先，您需要重新创建索引，因为自动字段映射默认选择无法排序的类型。发送以下请求以删除和重新创建索引：

```
DELETE /veggies

PUT /veggies
{
   "mappings":{
      "properties":{
         "name":{
            "type":"keyword"
         },
         "color":{
            "type":"keyword"
         },
         "classification":{
```

```
            "type":"keyword"
         }
      }
   }
}
```

然后，使用数据重新填充索引：

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "7"  } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "8" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "9" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "10" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "11" } }
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
```

现在，您可以使用排序进行搜索。此请求按分类添加升序排序：

```
GET veggies/_search
{
  "query" : {
    "term": { "color": "green" }
  },
  "sort" : [
      "classification"
  ]
}
```

相关资源

有关更多信息，请参阅以下资源：

- 入门
- 为数据建立索引
- 搜索数据

教程：迁移到亚马逊 OpenSearch 服务

索引快照是从自行管理 OpenSearch 或旧版 Elasticsearch 集群迁移到亚马逊服务的常用方式。OpenSearch 总体而言，此过程包括以下步骤：

1. 拍摄现有集群的快照，然后将快照上载到 Amazon S3 存储桶。
2. 创建 OpenSearch 服务域。
3. 向 S OpenSearch ervice 授予访问存储桶的权限，并确保您有权使用快照。
4. 在 OpenSearch 服务域上恢复快照。

此演练提供了更详细的步骤和替代选项（如果适用）。

拍摄并上载快照

尽管您可以使用 rep ository-s3 插件将快照直接拍摄到 S3，但您必须在每个节点上安装该插件，进行调整`opensearch.yml`（或者`elasticsearch.yml`如果使用 Elasticsearch 集群），重启每个节点，添加您的 Amazon 凭据，最后拍摄快照。此插件是持续使用或迁移大型集群的绝佳选择。

对于较小的集群，一次性方法是拍摄共享文件系统快照，然后使用将其上传 Amazon CLI 到 S3。如果您已有快照，请跳至步骤 4。

拍摄快照并将其上载到 Amazon S3

1. 将 `path.repo` 设置添加到所有节点上的 `opensearch.yml`（或 `Elasticsearch.yml`），然后重新启动每个节点。

   ```
   path.repo: ["/my/shared/directory/snapshots"]
   ```

2. 注册快照存储库，这是您拍摄快照之前所必需的。存储库只是一个存储位置：共享文件系统、Amazon S3、Hadoop Distributed File System (HDFS) 等。在这种情况下，我们将使用共享文件系统（“fs”）：

   ```
   PUT _snapshot/my-snapshot-repo-name
   {
     "type": "fs",
     "settings": {
       "location": "/my/shared/directory/snapshots"
     }
   ```

```
}
```

3. 拍摄快照：

```
PUT _snapshot/my-snapshot-repo-name/my-snapshot-name
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

4. 安装 Amazon CLI，然后运行 `aws configure` 以添加凭证。
5. 导航到快照目录。然后运行以下命令以创建新的 S3 存储桶，并将快照目录的内容上传到该存储桶：

```
aws s3 mb s3://amzn-s3-demo-bucket --region us-west-2
aws s3 sync . s3://amzn-s3-demo-bucket --sse AES256
```

此操作可能需要一些时间，具体取决于快照大小和互联网连接速度。

创建域

尽管控制台是创建域的最简单方法，但在这种情况下，您已经打开并 Amazon CLI 安装了终端。修改以下命令以创建符合您需要的域：

```
aws opensearch create-domain \
  --domain-name migration-domain \
  --engine-version OpenSearch_1.0 \
  --cluster-config InstanceType=c5.large.search,InstanceCount=2 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-user-password}' \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"AWS":["*"]},"Action":
["es:ESHttp*"],"Resource":"arn:aws:es:us-west-2:123456789012:domain/migration-domain/
*"}]}' \
```

```
  --region us-west-2
```

如果不修改，此命令会创建一个具有两个数据节点的可访问互联网的域，每个节点都有 100 GiB 的存储。它还可以通过 HTTP 基本身份验证和所有加密设置实现精细访问控制。如果您需要更高级的安全配置（例如 VPC），请使用 OpenSearch 服务控制台。

在发布命令之前，请更改域名、主用户凭证和账号。指定与 S3 存储桶相同的 Amazon Web Services 区域 OpenSearch/Elasticsearch 版本以及与您的快照兼容的版本。

> **Important**
>
> 快照只能向前兼容，并且只能与一个主要版本兼容。例如，您无法从 OpenSearch 1 恢复快照。 Elasticsearch 上的 x 集群 7. x 个集群，只有一个 OpenSearch 1。 x 或 2。 x 集群。次要版本也很重要。您无法从 5.3.2 OpenSearch 服务域上自行管理的 5.3.3 集群恢复快照。我们建议选择您的快照支持的最新版本 OpenSearch 或 Elasticsearch。有关兼容版本的表格，请参阅the section called “使用快照迁移数据”。

提供权限以访问 S3 存储桶。

在 Amazon Identity and Access Management (IAM) 控制台中，创建具有以下权限和信任关系的角色。创建角色时，选择 S3作为 Amazon 服务。将该角色命名为 `OpenSearchSnapshotRole`，以便于查找。

权限

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket"
      ]
    },
    {
```

```
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

信任关系

JSON

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

然后向您的个人 IAM 角色授予权限以代入 `OpenSearchSnapshotRole`。创建以下策略并将其附加到您的身份。

权限

JSON

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

```
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
    }
  ]
}
```

在 OpenSearch仪表板中映射快照角色（如果使用精细的访问控制）

如果启用了细粒度访问权限，即使您将 HTTP 基本身份验证用于所有其他目的，也需要将 `manage_snapshots` 角色映射到您的 IAM 角色，以便使用快照。

授予您的身份使用快照的权限

1. 使用您在创建 OpenSearch 服务域时指定的主用户凭据登录控制面板。您可以在 OpenSearch 服务控制台中找到控制面板 URL。其格式为 `https://domain-endpoint/_dashboards/`。
2. 从主菜单中选择安全、角色，然后选择 manage_snapshots 角色。
3. 选择映射的用户、管理映射。
4. 在相应字段中添加个人 IAM 角色的域 ARN。ARN 必须采用以下格式之一：

```
arn:aws:iam::123456789123:user/user-name
```

```
arn:aws:iam::123456789123:role/role-name
```

5. 选择 Map（映射）并确认在 Mapped users（映射的用户）下显示的角色。

还原快照。

此时，您可以通过两种方式访问您的 OpenSearch 服务域：使用主用户证书进行 HTTP 基本身份验证或使用 IAM 凭证进行 Amazon 身份验证。由于快照使用没有主用户概念的 Amazon S3，因此您必须使用您的 IAM 证书在 OpenSearch 服务域中注册快照存储库。

大多数编程语言都有库来协助对请求进行签名，但更简单的方法是使用像 Postman 这样的工具，并将您的 IAM 凭证放入授权部分中。

还原快照。

1. 无论您选择如何对请求进行签名，第一步都是注册存储库：

```
PUT _snapshot/my-snapshot-repo-name
{
  "type": "s3",
  "settings": {
    "bucket": "amzn-s3-demo-bucket",
    "region": "us-west-2",
    "role_arn": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
  }
}
```

2. 然后在存储库中列出快照，并找到要还原的快照。此时，您可以继续使用 Postman，也可以切换到像 curl 这样的工具。

   速记

```
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. 还原快照。

   速记

```
POST _snapshot/my-snapshot-repo-name/my-snapshot-name/_restore
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

   curl

```
curl -XPOST -u 'master-user:master-user-password' https://domain-endpoint/
_snapshot/my-snapshot-repo-name/my-snapshot-name/_restore \
  -H 'Content-Type: application/json' \
  -d '{"indices":"migration-index1,migration-index2,other-indices-
*","include_global_state":false}'
```

4. 最后，验证索引是否已按预期还原：

   速记

```
GET _cat/indices?v
```

   curl

```
curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/_cat/
indices?v
```

此时，迁移已完成。您可以将客户端配置为使用新的 OpenSearch 服务终端节点，调整域大小以适应您的工作负载，检查索引的分片数，切换到 IAM 主用户，或者开始在控制面板中 OpenSearch 构建可视化效果。

教程：使用 Amazon OpenSearch 服务创建搜索应用程序

使用 Amazon Serv OpenSearch ice 创建搜索应用程序的常用方法是使用网络表单将用户查询发送到服务器。然后，您可以授权服务器 OpenSearch APIs 直接调用，并让服务器向 Serv OpenSearch ice 发送请求。但是，如果您想编写不依赖服务器的客户端代码，则应针对安全和性能风险作出补偿。不建议允许未签名的公开访问权限。 OpenSearch APIs 用户可能会访问不安全的终端节点，或者通过过于广泛的查询（或过多的查询）影响集群性能。

本章介绍一个解决方案：使用 Amazon API Gateway 将用户限制为其中的一个子集， OpenSearch APIs 并 Amazon Lambda 对从 API Gateway 到 OpenSearch 服务的请求进行签名。

> **Note**
>
> 标准 API Gateway 和 Lambda 定价适用，但不能超出本教程的限制使用量，成本应忽略不计。

先决条件

本教程的先决条件是 OpenSearch 服务域。如果您还没有，请按照创建 OpenSearch 服务域中的步骤创建一个。

步骤 1：为示例数据建立索引

下载 sample-movies.zip，解压它，然后使用 _bulk API 操作将 5000 个文档添加到 movies 索引：

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/_bulk
{ "index": { "_index": "movies", "_id": "tt1979320" } }
{"directors":["Ron
 Howard"],"release_date":"2013-09-02T00:00:00Z","rating":8.3,"genres":
["Action","Biography","Drama","Sport"],"image_url":"http://ia.media-imdb.com/images/
M/MV5BMTQyMDE0MTY0OV5BMl5BanBnXkFtZTcwMjI2OTI0OQ@@._V1_SX400_.jpg","plot":"A re-
creation of the merciless 1970s rivalry between Formula One rivals James Hunt and
 Niki Lauda.","title":"Rush","rank":2,"running_time_secs":7380,"actors":["Daniel
 Brühl","Chris Hemsworth","Olivia Wilde"],"year":2013,"id":"tt1979320","type":"add"}
{ "index": { "_index": "movies", "_id": "tt1951264" } }
{"directors":["Francis Lawrence"],"release_date":"2013-11-11T00:00:00Z","genres":
["Action","Adventure","Sci-Fi","Thriller"],"image_url":"http://ia.media-imdb.com/
images/M/
MV5BMTAyMjQ3OTAxMzNeQTJeQWpwZ15BbWU4MDU0NzA1MzAx._V1_SX400_.jpg","plot":"Katniss
 Everdeen and Peeta Mellark become targets of the Capitol after
 their victory in the 74th Hunger Games sparks a rebellion in
 the Districts of Panem.","title":"The Hunger Games: Catching
 Fire","rank":4,"running_time_secs":8760,"actors":["Jennifer Lawrence","Josh
 Hutcherson","Liam Hemsworth"],"year":2013,"id":"tt1951264","type":"add"}
...
```

请注意，上方是一个示例命令，其中包含一小部分可用数据。要执行 _bulk 操作，您需要复制和粘贴 sample-movies 文件的全部内容。有关更多说明，请参阅the section called “选项 2：上传多个文档”。

您也可以使用以下 curl 命令实现相同的结果：

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

步骤 2：创建并部署 Lambda 函数

在 API Gateway 中创建 API 前，创建将请求传递到的 Lambda 函数。

创建 Lambda 函数

在此解决方案中，API Gateway 将请求传递给 Lambda 函数，该函数查询 OpenSearch 服务并返回结果。由于此示例函数使用的是外部库，您需要创建一个部署程序包并将其上传到 Lambda。

创建部署包

1. 打开命令提示符并创建 `my-opensearch-function` 项目目录。例如，在 macOS 上，请执行以下操作：

```
mkdir my-opensearch-function
```

2. 导航到 `my-sourcecode-function` 项目目录。

```
cd my-opensearch-function
```

3. 复制以下 Python 示例代码的内容，并且使用名为 `opensearch-lambda.py` 的新文件将其保存。将您的区域和主机端点添加到文件中。

```
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. 在新的 package 目录中安装外部库。

```
pip3 install --target ./package boto3
pip3 install --target ./package requests
pip3 install --target ./package requests_aws4auth
```

5. 使用已安装库在根目录下创建部署程序包。以下命令可在项目目录中生成 my-deployment-package.zip 文件。

```
cd package
zip -r ../my-deployment-package.zip .
```

6. 将 opensearch-lambda.py 文件添加到 zip 文件的根目录。

```
cd ..
```

```
zip my-deployment-package.zip opensearch-lambda.py
```

有关创建 Lambda 函数和部署程序包的更多信息，请参阅 Amazon Lambda 开发人员指南中的使用 .zip 文件归档部署 Python Lambda 函数和本指南中的 the section called “创建 Lambda 部署程序包”。

使用 Lambda 控制台创建函数

1. 在家中导航到 Lambda 控制台。https://console.aws.amazon.com/lambda/在左侧导航窗格中，选择函数。
2. 选择创建函数。
3. 配置以下字段：
   - 函数名称：opensearch-function
   - 运行时：Python 3.9
   - 架构：x86_64

   保留所有其他默认选项，然后选择创建函数。
4. 在函数摘要页面的代码源部分，选择从下拉列表中上传，然后选择 .zip 文件。找到您创建的 `my-deployment-package.zip` 文件，然后选择保存。
5. 处理程序是函数代码中处理事件的方法。在运行时设置下，选择编辑，根据 Lambda 函数所在的部署包中的文件名更改处理程序名称。鉴于您的文件名为 `opensearch-lambda.py`，请将处理程序重命名为 *`opensearch-lambda`*`.lambda_handler`。有关更多信息，请参阅 Python 中的 Lambda 函数处理程序中的。

步骤 3：在 API Gateway 中创建 API

使用 API Gateway 可以创建更有限的 API，并简化与 OpenSearch `_search` API 交互的过程。API Gateway 还可让您启用安全功能，如 Amazon Cognito 身份验证和请求限制。执行以下步骤来创建和部署 API：

创建和配置 API

使用 API Gateway 控制台创建 API

1. 在https://console.aws.amazon.com/apigateway/家中导航到 API Gateway 控制台。在左侧导航窗格中，选择APIs。
2. 定位REST API（非私有），然后选择构建。
3. 在下一页中，找到新建 API 部分，确保选中新建 API。
4. 配置以下字段：
   - API 名称：opensearch-api
   - 描述：用于搜索亚马逊 OpenSearch 服务域名的公共 API
   - 端点类型：区域
5. 选择创建 API。
6. 选择操作和创建方法。
7. 在下拉菜单中选择GET，然后单击复选标记进行确认。
8. 配置以下设置，然后选择保存：

设置	值
集成类型	Lambda 函数
使用 Lambda 代理集成	是
Lambda 区域	*us-west-1*
Lambda 函数	`opensearch-lambda`
使用原定设置超时	是

配置该方法请求

选择方法请求并配置以下设置：

设置	值
授权	NONE
请求验证器	验证查询字符串参数和标头

设置	值
必需的 API 密钥	false

在 URL 查询字符串参数下，选择添加查询字符串并配置以下参数：

设置	值
名称	q
必需	是

部署 API 并配置阶段

借助 API Gateway 控制台，您可以创建部署并将其与新的或现有阶段相关联，从而部署 API。

1. 选择操作和部署 API。
2. 对于部署阶段选择新阶段并将阶段命名为 `opensearch-api-test`。
3. 选择部署。
4. 在阶段编辑器中配置以下设置，然后选择保存更改：

设置	值
启用限制	是
费率	1000
突增	500

这些设置将配置一个 API，该 API 只有一个方法：一个针对终端节点根的 `GET` 请求 (`https://some-id.execute-api.us-west-1.amazonaws.com/search-es-api-test`)。该请求需要单个参数 (q) - 查询字符串要搜索的。调用后，该方法会将请求传递到将运行 `opensearch-lambda` 函数的 Lambda。有关更多信息，请参阅在 Amazon API Gateway 中创建 API 和在 Amazon API Gateway 中部署 REST API。

步骤 4：（可选）修改域访问策略

您的 OpenSearch 服务域必须允许 Lambda 函数向`GET`索引发出请求。`movies`如果您的域具有已启用精细访问控制的开放访问策略，则可以将其保持原样：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

您也可以选择建立更加精细的域访问策略。例如，以下最低策略提供了对整个 `movies` 索引的 `opensearch-lambda-role`（通过 Lambda 创建）访问：要获取 Lambda 自动创建的角色的确切名称，请转到 Amazon Identity and Access Management (IAM) 控制台，选择角色，然后搜索 “lambda”。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/service-role/opensearch-lambda-
role-1abcdefg"
      },
      "Action": "es:ESHttpGet",
```

```
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/movies/
_search"
    }
  ]
}
```

> **Important**
>
> 如果您为域启用了精细的访问控制，则还需要在 OpenSearch 仪表板中将角色映射到用户，否则您将看到权限错误。

配置 Lambda 执行角色权限

除了配置域访问策略外，您还必须确保 Lambda 执行角色具有访问您的 OpenSearch 服务域所必需的 IAM 权限。Lambda 函数需要特定的权限，具体取决于您使用的是托管域还是无 OpenSearch 服务集合。

对于托管 OpenSearch 服务域：

将以下 IAM 策略附加到您的 Lambda 执行角色以允许其向您的 OpenSearch 服务域发出请求：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPost"
      ],
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

对于无服务器 OpenSearch 服务集合：

如果您使用的是无服务器 OpenSearch 服务，请将以下 IAM 策略附加到您的 Lambda 执行角色：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aoss:*",
      "Resource": "arn:aws:aoss:us-west-1:123456789012:collection/collection-id"
    }
  ]
}
```

要将这些策略附加到您的 Lambda 执行角色，请执行以下操作：

1. 导航到 https://console.aws.amazon.com/iam/ 的 IAM 控制台。
2. 选择角色并搜索您的 Lambda 执行角色（通常命名`opensearch-lambda-role-xxxxxxxx`）。
3. 选择添加权限，然后选择创建内联策略。
4. 选择 JSON 选项卡并粘贴上面的相应策略，将占位符值替换为实际资源 ARNs。
5. 选择查看策略，提供类似的名称`OpenSearchAccess`，然后选择创建策略。

> **Note**
>
> 如果没有这些 IAM 权限，即使域访问策略允许请求，您的 Lambda 函数在尝试查询您的 OpenSearch 服务域时也会收到 “访问被拒绝” 错误。

有关访问策略的更多信息，请参阅 the section called “配置访问策略”。

映射 Lambda 角色（如果使用精细访问控制）

精细访问控制将在您能测试应用程序之前引入一个额外步骤。即使您将 HTTP 基本身份验证用于所有其他目的，也需要将 Lambda 角色映射到用户，否则您将看到权限错误。

1. 导航到该域的 OpenSearch 控制面板 URL。

2. 从主菜单中，选择安全、角色，然后选择 `all_access` 链接和需要将 Lambda 角色映射到的角色。
3. 选择映射的用户、管理映射。
4. 在 Backend roles（后端角色）下，添加 Lambda 角色的 Amazon 资源名称（ARN）。ARN 应采用 `arn:aws:iam::`*`123456789123`*`:role/service-role/`*`opensearch-lambda-role-1abcdefg`* 形式。
5. 选择映射并确认在映射的用户下显示的用户或角色。

步骤 5：测试 Web 应用程序

测试 Web 应用程序

1. 下载 sample-site.zip，解压后在常用文本编辑器中打开 `scripts/search.js`。
2. 更新 `apigatewayendpoint` 变量以指向您的 API Gateway 端点，并在给定路径末尾添加反斜线。您可以选择阶段，然后选择 API 的名称，即可在 API Gateway 中快速找到端点。`apigatewayendpoint` 变量应采用 `https://`*`some-id`*`.execute-api.`*`us-west-1`*`.amazonaws.com/opensearch-api-test/` 形式。
3. 打开 `index.html` 并尝试运行对 thor、house 和其他几个术语的搜索。

排除 CORS 错误

尽管 Lambda 函数在响应中包含支持 CORS 的内容，但您仍可能会看到以下错误：

```
Access to XMLHttpRequest at '<api-gateway-endpoint>' from origin 'null' has been
 blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present in the
 requested resource.
```

如果，请尝试以下操作：

1. 在 GET 资源上启用 CORS。在 Advanced (高级) 西方，设置 Access-Control-Allow-Credentials 为 `'true'`。
2. 在 API Gateway 中重新部署 API (Action (操作)、Deploy API (部署 API))。
3. 删除并重新添加 Lambda 函数触发器。重新添加，选择添加触发器并创建调用函数的 HTTP 端点。该触发器必须具有以下配置：

触发器	API	部署阶段	安全性
API Gateway	opensearch-api	opensearch-api-test	打开

后续步骤

本章只是一个展示概念的起始点。您可以考虑以下修改：

- 将您自己的数据添加到 OpenSearch 服务域。
- 将方法添加到您的 API。
- 在 Lambda 函数中，修改搜索查询或提高不同的字段。
- 以不同的方式呈现结果或修改 `search.js` 以向用户显示不同的字段。

教程：使用 S OpenSearch ervice 和 OpenSearch 控制面板可视化显示客户支持呼叫

本章是对以下情况的完整演练：企业收到一些客户支持呼叫，想要对其进行分析。每个呼叫的主题是什么？ 多少是正面的？ 多少是负面的？ 经理如何搜索或查看这些呼叫的脚本？

手动工作流可能包括：员工倾听录音、记录每个呼叫的主题并确定每个客户的互动是否积极。

此类流程需要大量人力。假设每次呼叫的平均时间为 10 分钟，则每位员工每天只能接听 48 次呼叫。排除人为偏见，他们生成的数据将高度准确，但数据量 将达到最少：只有调用的主题和一个表示客户是否满意的布尔值。涉及任何内容 (如完整脚本) 都会占用大量时间。

使用 Amazon S3、Amazon T ranscribe、Amazon Comprehend 和 A OpenSearch mazon Service，你可以用很少的代码自动执行类似的流程，最终获得更多的数据。例如，您可以获取呼叫的完整脚本、该脚本中的关键字以及呼叫的总体“情绪”(正面、负面、中性或混合)。然后，您可以使用 OpenSearch OpenSearch 控制面板搜索数据并实现数据可视化。

虽然您可以按原样使用本演练，但我们的目的是在 Service 中为文档创建索引之前，激发您有关如何丰富 JSON 文档的 OpenSearch 创意。

估计成本

通常，执行本演练中步骤的成本低于 2 美元。本演练使用以下资源：

- 传输和存储数据少于 100 MB 的 S3 存储桶

  要了解更多信息，请参阅 Amazon S3 定价。
- OpenSearch 具有一个`t2.medium`实例和 10GiB EBS 存储的Service 域

  要了解更多信息，请参阅 Amazon S OpenSearch ervice 定价。
- Amazon Transcribe 的几次呼叫

  要了解更多信息，请参阅 Amazon Transcribe 定价。
- 到 Amazon Comprehend 的几种自然语言处理呼叫

  要了解更多信息，请参阅 Amazon Comprehend 定价。

主题

- 步骤 1：配置先决条件
- 步骤 2：复制示例代码
- (可选) 步骤 3：索引示例数据
- 步骤 4：分析和可视化您的数据
- 步骤 5：清除资源和后续步骤

步骤 1：配置先决条件

继续操作之前，必须具有以下资源。

先决条件	描述
Amazon S3 存储桶	有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的创建存储桶。
OpenSearch 服务域	数据的目的地。有关更多信息，请参阅创建 OpenSearch 服务域。

如果还没有这些资源，可以使用以下 Amazon CLI 命令创建这些资源：

```
aws s3 mb s3://my-transcribe-test --region us-west-2
```

```
aws opensearch create-domain --domain-name my-transcribe-test --engine-version
 OpenSearch_1.0 --cluster-config  InstanceType=t2.medium.search,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 --access-
policies '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::123456789012:root"},"Action":"es:*","Resource":"arn:aws:es:us-
west-2:123456789012:domain/my-transcribe-test/*"}]}' --region us-west-2
```

> **Note**
>
> 这些命令使用 `us-west-2` 区域，但您可以使用 Amazon Comprehend 支持的任何区域。要了解更多信息，请参阅 Amazon Web Services 一般参考。

步骤 2：复制示例代码

1. 将以下 Python 3 示例代码复制并粘贴到名为 `call-center.py` 的新文件中：

```
import boto3
import datetime
import json
import requests
from requests_aws4auth import AWS4Auth
import time
import urllib.request

# Variables to update
audio_file_name = '' # For example, 000001.mp3
bucket_name = '' # For example, my-transcribe-test
```

```
domain = '' # For example, https://search-my-transcribe-test-12345.us-
west-2.es.amazonaws.com
index = 'support-calls'
type = '_doc'
region = 'us-west-2'

# Upload audio file to S3.
s3_client = boto3.client('s3')

audio_file = open(audio_file_name, 'rb')

print('Uploading ' + audio_file_name + '...')
response = s3_client.put_object(
    Body=audio_file,
    Bucket=bucket_name,
    Key=audio_file_name
)

# # Build the URL to the audio file on S3.
# # Only for the us-east-1 region.
# mp3_uri = 'https://' + bucket_name + '.s3.amazonaws.com/' + audio_file_name

# Get the necessary details and build the URL to the audio file on S3.
# For all other regions.
response = s3_client.get_bucket_location(
    Bucket=bucket_name
)
bucket_region = response['LocationConstraint']
mp3_uri = 'https://' + bucket_name + '.s3-' + bucket_region + '.amazonaws.com/' +
 audio_file_name

# Start transcription job.
transcribe_client = boto3.client('transcribe')

print('Starting transcription job...')
response = transcribe_client.start_transcription_job(
    TranscriptionJobName=audio_file_name,
    LanguageCode='en-US',
    MediaFormat='mp3',
    Media={
        'MediaFileUri': mp3_uri
    },
    Settings={
        'ShowSpeakerLabels': True,
```

```
        'MaxSpeakerLabels': 2 # assumes two people on a phone call
    }
)

# Wait for the transcription job to finish.
print('Waiting for job to complete...')
while True:
    response =
 transcribe_client.get_transcription_job(TranscriptionJobName=audio_file_name)
    if response['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    else:
        print('Still waiting...')
    time.sleep(10)

transcript_uri = response['TranscriptionJob']['Transcript']['TranscriptFileUri']

# Open the JSON file, read it, and get the transcript.
response = urllib.request.urlopen(transcript_uri)
raw_json = response.read()
loaded_json = json.loads(raw_json)
transcript = loaded_json['results']['transcripts'][0]['transcript']

# Send transcript to Comprehend for key phrases and sentiment.
comprehend_client = boto3.client('comprehend')

# If necessary, trim the transcript.
# If the transcript is more than 5 KB, the Comprehend calls fail.
if len(transcript) > 5000:
    trimmed_transcript = transcript[:5000]
else:
    trimmed_transcript = transcript

print('Detecting key phrases...')
response = comprehend_client.detect_key_phrases(
    Text=trimmed_transcript,
    LanguageCode='en'
)

keywords = []
for keyword in response['KeyPhrases']:
    keywords.append(keyword['Text'])
```

```
print('Detecting sentiment...')
response = comprehend_client.detect_sentiment(
    Text=trimmed_transcript,
    LanguageCode='en'
)

sentiment = response['Sentiment']

# Build the Amazon OpenSearch Service URL.
id = audio_file_name.strip('.mp3')
url = domain + '/' + index + '/' + type + '/' + id

# Create the JSON document.
json_document = {'transcript': transcript, 'keywords': keywords, 'sentiment':
 sentiment, 'timestamp': datetime.datetime.now().isoformat()}

# Provide all details necessary to sign the indexing request.
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region,
 'opensearchservice', session_token=credentials.token)

# Index the document.
print('Indexing document...')
response = requests.put(url, auth=awsauth, json=json_document, headers=headers)

print(response)
print(response.json())
```

2. 更新最初的六个变量。
3. 使用以下命令安装所需的程序包：

```
pip install boto3
pip install requests
pip install requests_aws4auth
```

4. 将您的 MP3 放在与相同的目录中`call-center.py`并运行脚本。示例输出如下：

```
$ python call-center.py
Uploading 000001.mp3...
Starting transcription job...
Waiting for job to complete...
Still waiting...
Still waiting...
```

```
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Detecting key phrases...
Detecting sentiment...
Indexing document...
<Response [201]>
{u'_type': u'call', u'_seq_no': 0, u'_shards': {u'successful': 1, u'failed': 0,
 u'total': 2}, u'_index': u'support-calls4', u'_version': 1, u'_primary_term': 1,
 u'result': u'created', u'_id': u'000001'}
```

`call-center.py` 可执行许多操作：

1. 该脚本会将音频文件（在本例中为，但是 Amazon Tr MP3 anscribe 支持多种格式）上传到您的 S3 存储桶。
2. 它将音频文件的 URL 发送给 Amazon Transcribe 并等待转录任务完成。

   完成转录任务所需的时间取决于音频文件的长度。假定需要数分钟而非数秒。

   > **Tip**
   >
   > 要提高转录质量，可为 Amazon Transcribe. 配置自定义词汇表。

3. 转录任务完成后，该脚本将提取脚本、将其剪裁为 5,000 个字符，并将其发送到 Amazon Comprehend 进行关键字和情绪分析。
4. 最后，该脚本会将完整脚本、关键字、情绪和当前时间戳添加至 JSON 文档，并在 S OpenSearch ervice 中为该文档创建索引。

> **Tip**
>
> LibriVox有可用于测试的公共领域有声读物。

(可选) 步骤 3：索引示例数据

如果您手头没有大量呼叫记录，可以为 sample-calls.zip 中的示例文档创建索引，其效果与 `call-center.py` 相当。

1. 创建一个名为 `bulk-helper.py` 的文件：

```
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. 为 `host` 和 `region` 更新最初的两个变量。
3. 使用以下命令安装所需的程序包：

```
pip install opensearch-py
```

4. 下载并解压缩 sample-calls.zip。
5. 将 `sample-calls.bulk` 与 `bulk-helper.py` 放在同一目录中并运行帮助程序。示例输出如下：

```
$ python bulk-helper.py
{
  "errors": false,
  "items": [
    {
      "index": {
        "_id": "1",
        "_index": "support-calls",
        "_primary_term": 1,
        "_seq_no": 42,
        "_shards": {
          "failed": 0,
          "successful": 1,
          "total": 2
        },
        "_type": "_doc",
        "_version": 9,
        "result": "updated",
        "status": 200
      }
    },
    ...
  ],
  "took": 27
}
```

步骤 4：分析和可视化您的数据

现在，您在 S OpenSearch ervice 中有一些数据，可以使用 OpenSearch 控制面板来可视化这些数据。

1. 导航到 `https://search-domain.region.es.amazonaws.com/_dashboards`。
2. 在使用 OpenSearch 控制面板之前，您需要索引模式。控制面板使用索引模式来将分析范围缩小到一个或多个索引。要匹配 `call-center.py` 创建的 `support-calls` 索引，转到堆栈管理、索引模式，并定义索引模式`support*`，然后选择下一步。
3. 对于 Time Filter field name (时间筛选字段名称)，选择 timestamp (时间戳)。
4. 现在，您可以开始创建可视化了。选择 Visualize (可视化)，然后添加新的可视化。
5. 选择饼图和 `support*` 索引模式。

6. 默认可视化是基本的，因此请选择 Split Slices (拆分切片) 来创建更有趣的可视化。

   对于 Aggregation，选择 Terms。对于 Field (字段)，请选择 sentiment.keyword。然后选择 Apply changes (应用更改) 和 Save (保存)。

7. 返回到 Visualize (可视化) 页面，然后添加其他可视化。这次请选择水平条形图。
8. 选择 Split Series (拆分序列)。

   对于 Aggregation，选择 Terms。对于 Field (字段)，请选择 keywords.keyword，并将 Size (大小) 更改为 20。然后选择 Apply Changes (应用更改) 和 Save (保存)。

9. 返回到 Visualize (可视化) 页面并添加一个最终的可视化、一个垂直条形图。
10. 选择 Split Series (拆分序列)。对于 Aggregation (聚合)，请选择 Date Histogram (日期直方图)。对于 Field (字段)，请选择 timestamp (时间戳) 并将 Interval (间隔) 更改为 Daily (每日)。
11. 选择 Metrics & Axes (指标和轴)，并将 Mode (模式) 更改为 normal (正常)。
12. 选择 Apply Changes (应用更改) 和 Save (保存)。

13. 现在，您有三个可视化内容，可以将它们添加到控制面板。选择 Dashboard (控制面板)，创建一个控制面板，并添加您的可视化内容。

步骤 5：清除资源和后续步骤

为避免产生不必要的费用，请删除 S3 存储桶和S OpenSearch ervice 域。要了解更多信息，请参阅亚马逊简单存储服务用户指南中的删除存储桶和本指南中的删除 OpenSearch 服务域。

与 MP3 文件相比，转录需要的磁盘空间要少得多。您可能缩短 MP3 保留窗口 (例如，将呼叫记录保留时限从三个月缩短为一个月)，保留多年的脚本，并仍可节省存储成本。

您还可以使用 Amazon Step Functions 和 Lambda, 自动完成转录流程，在建立索引之前添加其他元数据，或创建更复杂的可视化内容来配合您的具体使用案例。

Amazon S OpenSearch ervice 重命名：更改摘要

2021 年 9 月 8 日，我们的搜索与分析套件更名为 Amazon S OpenSearch ervice。 OpenSearch 服务支持 OpenSearch 以及传统 Elasticsearch OSS。以下各节介绍了随重命名而更改的不同服务部分，以及您需要执行哪些操作来确保域继续正常运行。

其中一些更改仅在您将域从 Elasticsearch 升级到。 OpenSearch在其他情况下（例如，在“账单和成本管理”控制台中），体验将立即更改。

请注意，此列表并不详尽。虽然产品的其他部分也发生了更改，但这些更新是最相关的。

新 API 版本

新版本的 OpenSearch 服务配置 API (2021-01-01) 可用于传统 Elasticsearc OpenSearch h OSS。21 个 API 操作被替换为更简洁和与引擎无关的名称（例如，`CreateElasticsearchDomain`更改为`CreateDomain`），但是 S OpenSearch ervice 继续支持这两个 API 版本。

我们建议您使用新的 API 操作来创建和管理后续域。请注意，当您使用新 API 操作创建域时，您需要指定采用格式 `Elasticsearch_X.Y` 或者 `OpenSearch_X.Y` 的 `EngineVersion` 参数，而不仅仅是版本号。如果不指定版本，则默认为的最新版本 OpenSearch。

升级 Amazon CLI 到 1.20.40 或更高版本`aws opensearch ...`以便用于创建和管理域。有关新的 CLI 格式，请参阅 OpenSearch CLI 引用。

重命名的实例类型

Amazon S OpenSearch ervice 中的实例类型现在采用了格式，`<type>.<size>.search`例如，`m6g.large.`*`search`*而不`m6g.large.`*`elasticsearch`*是。您不需要采取任何措施。现有域将自动启动引用 API 以及“账单和成本管理”控制台中的新实例类型。

如果您拥有预留实例 (RIs)，则您的合同不会受到更改的影响。旧的配置 API 版本仍与旧的命名格式兼容，但如果要使用新的 API 版本，则需要使用新格式。

访问策略更改

以下各节介绍了更新访问策略需要执行的操作。

IAM 策略

我们建议您更新 IAM 策略以使用重命名的 API 操作。但是，S OpenSearch ervice 将继续通过内部复制旧的 API 权限来保持现有策略。例如，如果您当前有权执行 `CreateElasticsearchDomain` 操作，则您现在可以调用 `CreateElasticsearchDomain`（旧的 API 操作）和 `CreateDomain`（新的 API 操作）。这同样适用于显式拒绝。有关更新的 API 操作的列表，请参阅策略元素引用。

SCP 策略

服务控制策略 (SCPs) 与标准 IAM 相比，增加了一层复杂性。为了防止 SCP 策略中断，您需要将旧的和新的 API 操作添加到您的每个 SCP 策略中。例如，如果用户当前具有 `CreateElasticsearchDomain` 的允许权限，您还需要授予它们 `CreateDomain` 的允许权限，以便他们能够保留创建域的能力。这同样适用于显式拒绝。

例如：

```
"Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:CreateElasticsearchDomain",
        "es:CreateDomain"
         ...
      ],
    },
      "Effect": "Deny",
      "Action:" [
        "es:DeleteElasticsearchDomain",
        "es:DeleteDomain"
         ...
```

新资源类型

OpenSearch Service 引入了以下新的资源类型：

资源	描述
`AWS::OpenSearchService::Domain`	表示 Amazon S OpenSearch ervice 域。此资源存在于服务级别，并不特定于域上运行的软件。它适用于 Amazon CloudFormation 和 Amazon

资源	描述
	Resource Groups等服务，您可以在其中创建和管理整个服务的资源。 有关将 Elasticsearch CloudFormation 中定义的域升级到 Elasticsearch 的说明 OpenSearch，请参阅 CloudFormation 用户指南中的备注。
`AWS::OpenSearch::Domain`	表示在域上运行的 OpenSearch /Elasticsearch 软件。此资源适用于Amazon CloudTrail和之类的服务 Amazon Config，这些服务引用的是域上运行的软件，而不是整个 OpenSearch 服务。这些服务现在包含单独的资源类型，用于运行 Elasticsearch (`AWS::Elasticsearch::Domain`) 和运行 OpenSearch (`AWS::OpenSearch::Domain`) 的域。

> **Note**
>
> 在中 Amazon Config，您将继续在现有`AWS::Elasticsearch::Domain`资源类型下看到您的数据长达数周，即使您将一个或多个域升级到也是如此 OpenSearch。

Kibana 重命名为控制面板 OpenSearch

OpenSearch 控制面板，Kibana 的 Amazon 替代品，是一种开源虚拟化工具，专为与之结合使用而设计。 OpenSearch将域从 Elasticsearch 升级到后 OpenSearch，`/_plugin/kibana`终端节点更改为。`/_dashboards` OpenSearch Service 会将所有请求重定向到新终端节点，但如果您在任何 IAM 策略中使用 Kibana 终端节点，请更新这些策略以包含新的`/_dashboards`终端节点。

如果您正在使用the section called “仪表板的 SAML 身份验证 OpenSearch ”，则在将域升级到之前 OpenSearch，您需要将身份提供商 (IdP) 中 URLs 配置的所有 Kibana 从更改为。`/_plugin/kibana` `/_dashboards`最常见 URLs 的是断言使用者服务 (ACS) URLs 和接收方 URLs。

OpenSearch 仪表板的默认`kibana_read_only`角色已重命名为`opensearch_dashboards_read_only`，该`kibana_user`角色已重命名为`opensearch_dashboards_user`。此更改适用于所有新创建的 1 OpenSearch . x 域。运行服务

软件 R20211203 或更高版本的域。如果将现有域升级到服务软件 R20211203，则角色名称将保持不变。

重命名的 CloudWatch 指标

域运行的多项 CloudWatch 指标发生了变化 OpenSearch。当您将域升级到时 OpenSearch，指标会自动更改，当前 CloudWatch 警报将中断。在将集群从 Elasticsearch OpenSearch 版本升级到版本之前，请确保更新 CloudWatch 警报以使用新指标。

更改了以下指标：

原始指标名称	新名称
KibanaHealthyNodes	OpenSearchDashboardsHealthyNodes
KibanaConcurrentConnections	OpenSearchDashboardsConcurrentConnections
KibanaHeapTotal	OpenSearchDashboardsHeapTotal
KibanaHeapUsed	OpenSearchDashboardsHeapUsed
KibanaHeapUtilization	OpenSearchDashboardsHeapUtilization
KibanaOS1MinuteLoad	OpenSearchDashboardsOS1MinuteLoad
KibanaRequestTotal	OpenSearchDashboardsRequestTotal
KibanaResponseTimesMaxInMillis	OpenSearchDashboardsResponseTimesMaxInMillis
ESReportingFailedRequestSysErrCount	KibanaReportingFailedRequestSysErrCount
ESReportingRequestCount	KibanaReportingRequestCount
ESReportingFailedRequestUserErrCount	KibanaReportingFailedRequestUserErrCount

原始指标名称	新名称
ESReportingSuccessCount	KibanaReportingSuccessCount
ElasticsearchRequests	OpenSearchRequests

有关 OpenSearch 服务发送给 Amazon 的指标的完整列表 CloudWatch，请参阅the section called “监控集群指标”。

账单和成本管理控制台更改

账单和成本管理控制台以及成本和使用情况报告中的历史数据将继续使用旧服务名称，因此在搜索数据时，您需要开始使用同时适用于 Amazon S OpenSearch er vice 和旧式 Elasticsearch 名称的筛选条件。如果现有已保存的报告，请更新筛选器以确保它们也包含 S OpenSearch ervice。当您的 Elasticsearch 使用率减少并增加 Elasticsearch 使用率时，您最初可能会收到警报 OpenSearch，但它会在几天内消失。

除了服务名称以外，所有报告、账单和价格列表 API 操作的以下字段也将更改：

字段	旧格式	新格式
实例类型	m5.large.elasticsearch	m5.large.search
产品系列	Elasticsearch 实例 Elasticsearch 卷	亚马逊 OpenSearch 服务实例 Amazon OpenSearch 服务卷
定价说明	c5.18xlarge.elasticsearch 实例小时（或部分小时）5.098 美元 - 欧盟	c5.18xlarge.search 实例小时（或部分小时）5.098 美元 - 欧盟
实例系列	ultrawarm.elasticsearch	ultrawarm.search

新事件格式

OpenSearch 服务发送给亚马逊 EventBridge 和亚马逊的事件格式 CloudWatch 已更改，特别是该`detail-type`字段。源字段 (`aws.es`) 保持不变。有关每种事件类型的完整格式，请参阅 the section called “监控事件”。如果您具有取决于旧格式的现有事件规则，请确保更新它们以符合新格式。

什么保持不变？

以下特性和功能以及未列出的其他特性和功能将保持不变：

- 服务委托人 (`es.amazonaws.com`)
- 供应商代码
- 域 ARNs
- 域终端节点

开始吧：将您的域名升级到 OpenSearch 1.x

OpenSearch 1.x x 支持从 Elasticsearch 版本 6.8 和 7 进行升级。 x。有关升级域的说明，请参阅 the section called “升级域（控制台）”。如果您使用的是 Amazon CLI 或配置 API 来升级域，则需要指定 a `TargetVersion` s `OpenSearch_1.x`。

OpenSearch 1.x x 引入了一个名为 “启用兼容模式” 的附加域设置。由于某些 Elasticsearch OSS 客户端和插件在连接之前检查集群版本，因此兼容模式设置 OpenSearch 为报告其版本为 7.10，以便这些客户端继续工作。

首次创建 OpenSearch 域时，或者 OpenSearch 从 Elasticsearch 版本升级到时，您可以启用兼容模式。如果未设置，则创建域时参数默认为 `false`，并且在升级域时默认为 `true`。

要使用配置 API 启用兼容模式，请将 `override_main_response_version` 设置为 `true`：

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/upgradeDomain
{
  "DomainName": "domain-name",
  "TargetVersion": "OpenSearch_1.0",
  "AdvancedOptions": {
    "override_main_response_version": "true"
   }
}
```

要在现有 OpenSearch域上启用或禁用兼容模式，您需要使用 OpenSearch _cluster/ settings API 操作：

```
PUT /_cluster/settings
{
  "persistent" : {
    "compatibility.override_main_response_version" : true
  }
}
```

对亚马逊 OpenSearch 服务进行故障排除

本主题介绍如何识别和解决常见的 Amazon OpenSearch 服务问题。遇到问题时请先参阅本节中的信息，然后再联系 Amazon 支持。

无法访问 OpenSearch 仪表板

OpenSearch 控制面板端点不支持已签名的请求。如果域的访问控制策略仅向特定 IAM 角色授予访问权限，并且您尚未配置 Amazon Cognito 身份验证，在尝试访问 Dashboards 时您可能会收到以下错误：

```
"User: anonymous is not authorized to perform: es:ESHttpGet"
```

如果您的 OpenSearch 服务域使用 VPC 访问权限，则可能不会收到此错误，但请求可能会超时。要了解有关更正此问题和您可以使用的各种配置选项的更多信息，请参阅 the section called “控制对仪表板的访问权限 ”、the section called “关于 VPC 域的访问策略” 和 the section called “身份和访问管理”。

无法访问 VPC 域

请参阅 the section called “关于 VPC 域的访问策略”和the section called “测试 VPC 域”。

集群处于只读状态

与早期的 Elasticsearch 版本 OpenSearch 和 Elasticsearch 7 相比。 x 使用不同的系统进行群集协调。在此新系统中，如果集群发生法定数量损失，则在您采取操作之前集群将不可用。法定数量损失可能会有两种形式：

- 如果您的集群使用专用主节点，在一半或更多的节点不可用时将出现法定数量损失。
- 如果您的集群不使用专用主节点，在一半或更多的数据节点不可用时将出现法定数量损失。

如果出现法定人数损失并且您的集群有多个节点，S OpenSearch ervice 会恢复法定人数并将集群置于只读状态。您有两种选择：

- 删除只读状态，并按原样使用集群。
- 从快照还原集群或单独的索引。

如果您更希望原样使用集群，请使用以下请求验证集群的运行状况为绿色：

```
GET _cat/health?v
```

如果集群运行状况为红色，我们建议从快照还原集群。您也可以参阅the section called “红色集群状态”，了解故障排除步骤。如果集群运行状况为绿色，使用以下请求确保所有预期索引存在：

```
GET _cat/indices?v
```

然后，运行一些搜索以验证预期数据存在。如果满足这些情况，您可以使用以下请求移除只读状态：

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.blocks.read_only": false
  }
}
```

如果出现法定人数损失并且您的群集只有一个节点，S OpenSearch ervice 会替换该节点，并且不会将群集置于只读状态。否则，您具有相同的选项：原样使用集群或者从快照还原。

在这两种情况下， OpenSearch 服务都会向您发送两个事件Amazon Health Dashboard。第一个事件告知您发生了法定数量损失。第二次发生在 S OpenSearch ervice 成功恢复法定人数之后。有关使用的更多信息 Amazon Health Dashboard，请参阅《Amazon Health 用户指南》。

红色集群状态

红色集群状态表示至少有一个主分片及其副本未分配给节点。 OpenSearch 无论索引的状态如何，服务都会尝试自动拍摄所有索引的快照，但是当红色群集状态仍然存在时，快照会失败。

红色集群状态的最常见原因是集群节点发生故障，以及连续处理繁重负载造成 OpenSearch 进程崩溃。

> **Note**
>
> OpenSearch 无论集群状态如何，服务都会将自动快照存储 14 天。因此，如果集群的红色状态持续超过两周，系统会删除最后一个运行正常的自动快照，您可能会永久丢失集群数据。如果您的 OpenSearch 服务域进入红色群集状态， Amazon Web Services 支持 可能会与您联

系，询问您是想自己解决问题还是希望支持团队提供帮助。您可以设置 CloudWatch 警报，以便在出现红色集群状态时通知您。

最终，红色分片会导致红色集群，而红色索引会导致红色分片。要识别导致红色群集状态的索引，OpenSearch 有一些帮助 APIs。

- `GET /_cluster/allocation/explain` 选择它找到的第一个未分配的分片，并说明为何无法将它分配给节点：

```
{
    "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` 显示每个索引的运行状况、文档数量以及磁盘使用情况：

```
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
```

删除红色索引是修复红色集群状态的最快方式。根据红色群集状态的原因，您可以扩展 OpenSearch 服务域以使用更大的实例类型、更多实例或更多基于 EBS 的存储，并尝试重新创建有问题的索引。

如果删除存在问题的索引不可行，您可以还原快照、从索引中删除文档、更改索引设置、减少副本数量或删除其他索引以释放磁盘空间。重要步骤是在重新配置 OpenSearch 服务域之前，先解决红色群集

状态。重新配置具有红色集群状态的域会将问题复杂化，并导致该域卡在正在处理配置状态，直到问题解决。

自动修复红色集群

如果您的集群状态持续显示红色超过一个小时，S OpenSearch ervice 会尝试通过重新路由未分配的分片或从过去的快照中恢复来自动修复该问题。

如果无法修复一个或多个红色索引，并且集群状态总共保持为红色 14 天，则只有在群集至少满足以下条件之一时， OpenSearch 服务才会采取进一步的措施：

- 只有一个可用区
- 没有专用主节点
- 包含具爆发能力的实例类型（T2 或 T3）

此时，如果您的集群符合这些条件之一，S OpenSearch ervice 将在接下来的 7 天内向您发送每日通知，说明如果您不修复这些索引，则所有未分配的分片都将被删除。如果您的集群状态在 21 天后仍为红色，S OpenSearch ervice 会删除所有红色索引上未分配的分片（存储和计算）。您将在 OpenSearch 服务控制台的 “通知” 面板中收到每个事件的通知。有关更多信息，请参阅 the section called “集群运行状况事件”。

从连续处理繁重负载恢复

要确定红色集群状态是否源于在数据节点上连续处理繁重负载，请监控以下集群指标。

相关指标	描述	恢复
JVMMemory压力	指定用于集群中所有数据节点的 Java 堆的百分比。查看此指标的统计数据最大值，并试图尽可能地减少内存压力下降，因为 Java 垃圾收集器无法回收充足的内存。这种模式可能由于复杂的查询或大型数据字段而产生。 x86 实例类型使用 Concurrent Mark Sweep (CMS) 垃圾回收器，后者随应用程序线程一起运行以保持较短的暂	为 JVM 设置内存断路器。有关更多信息，请参阅 the section called “JVM OutOfMemoryError”。 如果问题仍然存在，请删除不必要索引、减少对域的请求的数量或降低其复杂性、添加实例或使用更大的实例类型。

相关指标	描述	恢复
	停。如果 CMS 在正常回收期间无法回收足够的内存，会触发完整垃圾回收，这可能会导致应用程序暂停时间较长并影响集群稳定性。 基于 ARM 的 Graviton 实例类型使用 Garbage-First (G1) 垃圾回收器，它与 CMS 类似，但使用额外的短暂停和堆碎片整理来进一步减少对完整垃圾回收的需求。 无论哪种情况，如果内存使用量继续增长到垃圾收集器在完全垃圾收集期间可以回收的容量之外，则会因内存不足错误而 OpenSearch 崩溃。对于所有实例类型，一个很好的经验法则是将使用率保持为低于 80%。 `_nodes/stats/jvm` API 提供了一个有用的 JVM 统计数据、内存池使用情况和垃圾收集信息摘要： `GET domain-endpoint /_nodes/stats/jvm?pretty`	
CPUUtilization	指定用于集群中各数据节点的 CPU 资源的百分比。查看此指标的 Maximum 统计数据，并查找连续模式的高使用率。	添加数据节点或增加现有数据节点的实例类型的大小。
节点	指定集群中的节点数。查看此指标的最小值统计数据。这个值会在该服务为集群部署新的实例队列时波动。	添加数据节点。

黄色集群状态

黄色集群状态意味着所有索引的主分片分配给集群中的节点，但是至少有一个索引的副本分片不是如此。由于 Ser OpenSearch vice 无法为其他节点分配副本，因此单节点集群始终以黄色群集状态进行初始化。要获得绿色集群状态，请增加节点数。有关更多信息，请参阅 the section called “调整域大小”。

创建新索引后或节点出现故障后，多节点群集可能会短暂显示黄色群集状态。当在集群中 OpenSearch 复制数据时，此状态会自行解决。磁盘空间不足也可能导致黄色群集状态；只有节点具有容纳副本分片的磁盘空间时，群集才能分发副本分片。

ClusterBlockException

您可能会由于以下原因收到 `ClusterBlockException` 错误。

缺少可用存储空间

如果集群中一个或多个节点的存储空间小于 1) 20% 的可用存储空间或 2) 20 GiB 存储空间的最小值，则添加文档和创建索引等基本写入操作可能会开始失败。 the section called “计算存储要求”提供了 OpenSearch 服务如何使用磁盘空间的摘要。

为避免出现问题，请在 OpenSearch 服务控制台中监控`FreeStorageSpace`指标，并创建 CloudWatch 警报，以便在低于特定阈值时`FreeStorageSpace`触发。 `GET /_cat/allocation?v`还提供了分片分配和磁盘使用情况的有用摘要。要解决与存储空间不足相关的问题，请扩展您的 OpenSearch 服务域以使用更大的实例类型、更多实例或更多基于 EBS 的存储。

JVM 内存压力过高

当 Pr JVMMemoryes sure 指标在 30 分钟内超过 92% 时，S OpenSearch ervice 会触发保护机制并阻止所有写入操作以防止集群进入红色状态。当打开保护时，写入操作将失败并返回 `ClusterBlockException` 错误，无法创建新索引，并且会引发 `IndexCreateBlockException` 错误。

当 Pr JVMMemoryes sure 指标在五分钟内恢复到 88% 或更低时，保护将被禁用，对集群的写入操作也将解除阻止。

JVM 内存压力过高可能是由于对集群的请求数量激增、节点间的分片分配不平衡、集群中的分片过多、字段数据或索引映射爆炸或无法处理传入负载的实例类型造成的。也可能是由于在查询中使用聚合、通配符或较宽的时间范围造成的。

要减少集群的流量并解决 JVM 内存压力过高的问题，请尝试以下一项或多项操作：

- 扩展域，使每个节点的最大堆大小为 32GB。
- 通过删除旧的或未使用的索引来减少分片的数量。
- 使用 `POST index-name/_cache/clear?fielddata=true` API 操作清除数据缓存。请注意，清除缓存可能会中断正在进行的查询。

一般而言，为了避免将来出现 JVM 内存压力过高的问题，遵循以下最佳实践：

- 避免在文本字段上进行聚合，或者将您的索引的映射类型更改为 `keyword`。
- 通过选择正确的分片数量优化搜索和索引请求。
- 设置索引状态管理（ISM）策略以定期移除未使用的索引。

迁移到带待机功能的多可用区时出错

当您将现有域迁移到带待机功能的多可用区时，可能会出现以下问题。

从无备用域的域迁移到有备用域的域期间，创建索引、索引模板或 ISM 策略

如果您在将域从不带待机功能的多可用区迁移到带待机功能的多可用区时创建索引，且索引模板或 ISM 策略未遵循建议数据复制指南，则可能导致数据不一致，迁移可能会失败。为避免这种情况，请使用三倍数据副本数（包括主节点和副本）创建新索引。您可以使用 `DescribeDomainChangeProgress` API 检查迁移进度。如果遇到副本计数错误，请修复错误，然后联系 Amazon 支持重新尝试迁移。

数据副本数量错误

如果域中的数据副本数量出错，则迁移到带待机功能的多可用区将发生失败。

JVM OutOfMemoryError

JVM `OutOfMemoryError` 通常意味着触发了以下一种 JVM 断路器。

断路器	描述	集群设置属性
父级断路器	允许所有断路器占用的 JVM 堆内存百分比总和。原定设置值是 95%。	`indices.breaker.total.limit`
现场数据断路器	将单个数据字段加载到内存时所允许的 JVM 堆内存百分比。默认值为 40%。如果您通过大型字段上载数据，您可能需要提高此限制。	`indices.breaker.fielddata.limit`
请求断路器	允许用于响应服务请求的数据结构所占的 JVM 堆内存百分比。默认值为 60%。如果您的服务请求涉及计算聚合，您需要提高此限制。	`indices.breaker.request.limit`

集群节点失败

Amazon EC2 实例可能会出现意外终止和重启的情况。通常， OpenSearch 服务会为您重新启动节点。但是， OpenSearch 集群中的一个或多个节点可能会停留在发生故障状态。

要检查此情况，请在 OpenSearch 服务控制台上打开您的域名控制面板。转到 Cluster health（集群运行状况）选项卡，然后找到 Total nodes（总节点）指标。查看所报告的节点数是否少于您为集群配置的数量。如果指标显示一个或多个节点出现故障已超过 1 天，请联系 Amazon 支持。

您也可以设置 CloudWatch 警报，以便在出现此问题时通知您。

> **Note**
>
> 总节点指标在更改集群配置和服务例行维护期间不准确。此行为是预期的。该指标会报告正确数量的集群节点。要了解更多信息，请参阅the section called “配置更改”。

为了保护您的集群免受节点意外终止和重启的影响，请为 OpenSearch 服务域中的每个索引至少创建一个副本。

超过最大分片限制

OpenSearch 以及 7. x 个版本的 Elasticsearch 的默认设置为每个节点不超过 1,000 个分片。OpenSearch如果请求（例如创建新索引）会导致您超过此限制，/Elasticsearch 会抛出错误。如果您遇到此错误，则有多个选项：

- 将更多数据节点添加到集群。
- 增加 `_cluster/settings/cluster.max_shards_per_node` 设置。
- 使用 _shrink API 来减少节点上的分片数。

域卡在 Processing（正在处理）状态

当您的 OpenSearch 服务域处于配置更改过程中时，它会进入 “处理中” 状态。当您启动配置更改时，在 OpenSearch 服务创建新环境时，域状态将更改为 “处理中”。在新环境中，S OpenSearch ervice 会启动一组新的适用节点（例如数据、主节点或 UltraWarm）。迁移完成后，较旧节点将被终止。

如果出现以下任一情况，集群可能会卡在“Processing（正在处理）”状态：

- 一组新的数据节点启动失败。
- 将分片迁移到新的数据节点集不成功。
- 验证检查因错误而失败。

有关每种情况下的详细解决步骤，请参阅为什么我的 Amazon S OpenSearch ervice 域名处于 “处理中” 状态？ 。

EBS 可爆发容量余额低

OpenSearch 当您的一个通用型 (SSD) 卷上的 EBS 突发余额低于 70% 时，服务会向您发送控制台通知，如果余额低于 20%，则会向您发送后续通知。要解决此问题，您可以纵向扩展集群，也可以减少读取和写入 IOPS，从而增加可爆发容量余额。对于具有 gp3 卷类型的域以及具有卷大小超过 1000 GiB 的 gp2 卷的域，突增余额保持在 0。有关更多信息，请参阅通用型 SSD 卷（gp2）。您可以使用该`BurstBalance` CloudWatch 指标监控 EBS 突发平衡。

在调整卷大小期间，EBS 指标会增加

在修改 Amazon Elastic Block Store 卷大小时，您可能会发现各种 EBS 指标暂时增加`Write Throughput`，例如`Write Throughput Micro bursting`、`Disk Queue Depth`、和`IOPS`。这是调整大小操作期间的预期行为，通常持续几秒钟。但是，持续时间可能会因修改的卷大小而异。

为避免延迟问题和请求被拒绝，请仅在集群运行状况良好且网络流量较低时才执行 EBS 卷大小调整。

无法启用审核日志

尝试使用 OpenSearch 服务控制台启用审核日志发布时，可能会遇到以下错误：

为 CloudWatch 日志组指定的资源访问策略未向 Amazon S OpenSearch ervice 授予足够的权限来创建日志流。请检查资源访问策略。

如果遇到此错误，请验证您的策略 `resource` 元素包含正确的日志组 ARN。如果是这样，请执行以下步骤：

1. 等待几分钟。
2. 在 Web 浏览器中刷新页面。
3. 选择 Select existing group（选择现有组）。
4. 对于现有日志组，选择在收到错误消息之前创建的日志组。
5. 在访问策略部分，选择 Select existing policy（选择现有策略）。
6. 对于现有策略中，选择您在收到错误消息之前创建的策略。
7. 请选择启用。

如果在重复该进程多次后仍然存在错误，请联系 Amazon 支持。

无法关闭索引

OpenSearch 服务仅支持 Elasticsearch 7.4 OpenSearch 及更高版本的 `_close`API。如果您使用较老的版本，要从快照还原某个索引，则可以删除现有索引（在为快照重新建立索引之前或之后）。

客户端许可检查

Logstash 和 Beats 的默认发行版包括专有许可检查，并且无法连接到的开源版本。 OpenSearch确保将这些客户端的 Apache 2.0 (OSS) 发行版与 OpenSearch 服务一起使用。

请求限制

如果您收到持久 `403 Request throttled due to too many requests` 或 `429 Too Many Requests` 错误，请考虑垂直扩展：如果有效负载会导致内存使用量超过 Java 堆的最大大小，Amazon S OpenSearch ervice 会限制请求。

无法通过 SSH 登录节点

您不能使用 SSH 访问 OpenSearch 集群中的任何节点，也不能直接修改`opensearch.yml`。而是使用控制台 Amazon CLI、或 SDKs 来配置您的域。您也可以使用 OpenSearch REST APIs 指定一些集群级别的设置。要了解更多信息，请参阅亚马逊 OpenSearch 服务 API 参考和the section called “支持的操作”。

如果您需要更深入地了解集群的性能，则可以将错误日志和慢速日志发布到 CloudWatch。

“对象的存储类无效”快照错误

OpenSearch 服务快照不支持 S3 Glacier 存储类别。如果 S3 存储桶包含将对象转换为 S3 Glacier 存储类的生命周期规则，则在尝试列出快照时可能会遇到此错误。

如果需要从存储桶中还原快照，请从 S3 Glacier 还原对象，将对象复制到新存储桶，然后将新存储桶注册为快照存储库。

主机标头无效

OpenSearch 服务要求客户端在请求标头`Host`中指定。有效的 `Host` 值是没有 `https://` 的域终端节点，例如：

```
Host: search-my-sample-domain-ih2lhn2ew2scurji.us-west-2.es.amazonaws.com
```

如果您在发出请求时收到`Invalid Host Header`错误，请检查您的客户端或代理是否在`Host`标头中包含 OpenSearch 服务域终端节点（而不是其 IP 地址）。

M3 实例类型无效

OpenSearch 服务不支持向正在运行或 Elasticsearch 版本 6.7 及更高版本的现有域中添加 OpenSearch 或修改 M3 实例。您可以继续在 Elasticsearch 6.5 及更低版本中使用 M3 实例。

我们建议您选择更新的实例类型。对于运行 OpenSearch 或 Elasticsearch 6.7 或更高版本的域名，以下限制适用：

- 如果现有域不使用 M3 实例，则不能再对其进行更改。
- 如果将现有域从 M3 实例类型更改为其他实例类型，则无法切换回去。

启用后，热门查询停止工作 UltraWarm

在域 UltraWarm 上启用时，如果该设置没有预先存在的替代项，Serv OpenSearch ice 会自动将该值`search.max_buckets`设置为，`10000`以防止内存密集型查询使热节点饱和。如果您的热门查询使用的存储桶超过 10,000 个，则在您启用 UltraWarm后，它们可能会停止工作。

由于亚马逊 OpenSearch 服务的托管性质，您无法修改此设置，因此您需要提交支持案例以提高限制。限制增加不需要高级支持订阅。

升级后无法降级

就地升级是不可撤消的，但如果您联系 Amazon Support，他们可以帮助您在新域上还原自动的升级前快照。例如，如果您将域从 Elasticsearch 5.6 升级到 6.4，Support 可以 Amazon 帮助您在新 Elasticsearch 5.6 域上恢复升级前的快照。如果您制作了原始域的手动快照，您可以自行执行该步骤。

需要所有 Amazon Web Services 区域的域摘要

以下脚本使用 Amazon d EC2 escribe- regions Amazon CLI 命令创建可 OpenSearch 提供服务的所有区域的列表。然后它会调list-domain-names用每个区域：

```
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
```

您将对于每个区域接收以下输出：

```
Listing domains in region:'us-west-2'...
[
  {
    "DomainName": "sample-domain"
```

```
  }
]
```

OpenSearch 服务不可用的区域会返回 “无法连接到终端节点 URL”。

使用 OpenSearch 仪表板时出现浏览器错误

当您使用仪表板查看服务域中的数据时，您的浏览器会将服务错误消息封装在 HTTP 响应对象中。OpenSearch 您可以使用 Web 浏览器中常用的开发人员工具 (如 Chrome 中的开发者工具) 来查看基础服务错误和帮助您进行调试工作。

在 Chrome 中查看服务错误

1. 从 Chrome 顶部菜单栏中，依次选择 View、Developer、Developer Tools。
2. 选择网络选项卡。
3. 在状态列中，选择状态为 500 的任何 HTTP 会话。

在 Firefox 中查看服务错误

1. 从菜单中，依次选择工具、Web 开发人员和网络。
2. 选择任何状态为 500 的 HTTP 会话。
3. 选择响应选项卡以查看服务响应。

节点分片和存储偏斜

节点分片偏斜是指集群中有一个或多个节点的分片数量明显多于其他节点的现象。节点存储偏斜是指集群中有一个或多个节点的存储数量（`disk.indices`）明显多于其他节点的现象。虽然这两种现象都可能是暂时的，例如在某个域替换了一个节点并且仍在为其分配分片时，但如果这两种现象持续存在，则应予以解决。

要识别这两种类型的偏斜，请运行 _cat/allocation API 操作并比较响应中的 `shards` 和 `disk.indices` 条目：

```
shards    | disk.indices  | disk.used  | disk.avail   | disk.total   | disk.percent |
host     | ip        | node
   264    |      465.3mb  |   229.9mb  |      1.4tb   |      1.5tb   |            0 |
 x.x.x.x  | x.x.x.x   | node1
```

```
   115    |        7.9mb  |    83.7mb  |     49.1gb   |     49.2gb   |           0 |
x.x.x.x  | x.x.x.x  | node2
   264    |      465.3mb  |   235.3mb  |      1.4tb   |      1.5tb   |           0 |
 x.x.x.x  | x.x.x.x  | node3
   116    |        7.9mb  |    82.8mb  |     49.1gb   |     49.2gb   |           0 |
x.x.x.x  | x.x.x.x  | node4
   115    |        8.4mb  |      85mb  |     49.1gb   |     49.2gb   |           0 |
x.x.x.x  | x.x.x.x  | node5
```

虽然某些存储偏斜属于正常现象，但任何超过平均值 10% 的存储偏斜都属于显著。当分片分布出现偏斜时，CPU、网络和磁盘带宽使用情况也会出现偏斜。由于数据越多通常意味着索引和搜索操作越多，因此负载最大的节点往往也是资源最紧张的节点，而负载较轻的节点意味着容量未得到充分利用。

修复方法：将分片数量设置为数据节点数量的一定倍数，从而确保每个索引都跨数据节点均匀分布。

索引分片和存储偏斜

索引分片偏斜是指一个或多个节点拥有某个索引的分片数量比其他节点多的现象。索引存储偏斜是指一个或多个节点拥有某个索引的总存储容量的比例过大的现象。

索引偏斜比节点偏斜更难识别，因为它需要对 _cat/shards API 输出执行一些操作。如果集群或节点指标中存在偏斜的迹象，则应调查索引偏斜。以下是表明存在索引偏斜的常见迹象：

- 数据节点的某个子集上出现 HTTP 429 错误
- 跨数据节点的索引或搜索操作排队不均匀
- 数据节点之间的 JVM 堆 and/or CPU 利用率不均衡

修复方法：将分片数量设置为数据节点数量的一定倍数，从而确保每个索引都跨数据节点均匀分布。如果您仍然看到索引存储或分片偏差，则可能需要强制分片重新分配，服务域的每次蓝/绿部署都会发生这种情况。 OpenSearch

在选择 VPC 访问后出现未授权的操作

使用 OpenSearch 服务控制台创建新域时，您可以选择选择 VPC 或公共访问。如果您选择 VPC 访问，则 OpenSearch 服务会查询 VPC 信息，如果您没有适当的权限，则服务会失败：

```
You are not authorized to perform this operation. (Service: AmazonEC2; Status Code:
 403; Error Code: UnauthorizedOperation
```

要启用此查询，您必须有权访问 `ec2:DescribeVpcs`、`ec2:DescribeSubnets` 和 `ec2:DescribeSecurityGroups` 操作。此要求仅适用于控制台。如果您使用 Amazon CLI 创建和配置带有 VPC 终端节点的域，则无需访问这些操作。

在创建 VPC 域后卡在加载状态

在创建使用 VPC 访问权限的新域后，该域的配置状态可能会卡在正在加载。如果出现此问题，则说明您所在地区可能已禁用 Amazon Security Token Service（Amazon STS）。

要将 VPC 终端节点添加到您的 VPC， OpenSearch 服务需要代入`AWSServiceRoleForAmazonOpenSearchService`角色。因此， Amazon STS 必须启用才能创建在给定区域中使用 VPC 访问的新域。要了解有关启用和禁用的更多信息 Amazon STS，请参阅 IAM 用户指南。

对 OpenSearch API 的请求被拒绝

随着对 OpenSearch API 引入基于标签的访问控制，您可能会开始看到以前从未出现的访问被拒绝的错误。这可能是因为您的一个或多个访问策略包含使用 `ResourceTag` 条件的 `Deny`，而且这些条件现在正在得到遵守。

例如，如果域具有标签 `environment=production`，则以下策略仅用于拒绝从配置 API 访问 `CreateDomain` 操作。尽管操作列表还包括 `ESHttpPut`，但拒绝声明不适用于该操作或任何其他 `ESHttp*` 操作。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:CreateDomain",
      "es:ESHttpPut"
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
```

```
        }
      }
    }]
  }
```

由于增加了对 OpenSearch HTTP 方法的标签的支持，像上面这样的基于 IAM 身份的策略将导致附加的用户被拒绝访问该`ESHttpPut`操作。以前，在没有标签验证的情况下，附加的用户仍然能够发送 PUT 请求。

如果在将域更新为服务软件 R20220323 或更高版本后开始看到访问被拒绝错误，请检查基于身份的访问策略，看看是否是这种情况，并在必要时进行更新以允许访问。

无法从 Alpine Linux 连接

Alpine Linux 将 DNS 响应大小限制为 512 字节。如果您尝试从 Alpine Linux 3.18.0 或更低版本连接到您的 OpenSearch 服务域，则如果该域位于 VPC 中且节点超过 20 个，则 DNS 解析可能会失败。如果使用 Alpine Linux 3.18.0 以上版本，则应该能够解析超过 20 台主机。有关更多信息，请参阅 Alpine Linux 3.18.0 发布说明。

如果您的域名位于 VPC 中，我们建议使用其他 Linux 分配（例如 Debian、Ubuntu、CentOS、Red Hat Enterprise Linux 或 Amazon Linux 2）来连接到该域。

Search Backpressure 请求过多

基于 CPU 的准入控制是一种守卫机制，根据节点的当前容量主动限制节点请求数量，包括流量自然增长和峰值。如果请求过多，拒绝时将返回 HTTP 429“请求过多”状态代码。此错误表示集群资源不足、资源密集型搜索请求或工作负载意外激增。

Search Backpressure 将提供拒绝原因，有助于微调资源密集型搜索请求。对于流量高峰，建议使用指数回退和抖动进行客户端重试。

在使用开发工具包时出现证书错误

由于 Amazon SDKs 使用计算机上的 CA 证书，因此当您尝试使用 SDK 时，更改 Amazon 服务器上的证书可能会导致连接失败。虽然错误消息是不同的，但通常包含以下文本：

```
Failed to query OpenSearch
...
```

```
SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

您可以通过保留计算机的 CA 证书和操作系统来防止这些故障 up-to-date。如果您在公司环境中遇到这个问题而且未管理您自己的计算机，则可能需要请求管理员来协助处理更新过程。

以下列表显示了最低的操作系统和 Java 版本：

- 安装了 2005 年 1 月或之后更新的微软 Windows 版本的信任列表 CAs 中至少包含一个必需的版本。
- 带有 Java 的 Mac OS X 10.4 适用于 Mac OS X 10.4 版本 5（2007 年 2 月）、Mac OS X 10.5（2007 年 10 月）及更高版本的信任列表 CAs中至少包含一个必需的版本。
- Red Hat Enterprise Linux 5（2007 年 3 月）、6 和 7 以及 CentOS 5、6 和 7 的默认可信 CA 列表 CAs 中都至少包含一个必需的。
- Java 1.4.2_12（2006 年 5 月）、5 Update 2（2005 年 3 月）以及所有更高版本，包括 Java 6（2006 年 12 月）、7 和 8，其默认可信 CA 列表 CAs 中至少包含一个必需的。

这三个证书颁发机构为：

- Amazon Root CA 1
- Starfield Services Root Certificate Authority – G2
- Starfield Class 2 Certification Authority

来自前两个机构的根证书可从 Amazon Trust Services 获得，但保留计算机 up-to-date是更直接的解决方案。要了解有关 ACM 提供的证书的更多信息，请参阅。Amazon Certificate Manager FAQs

> **Note**
>
> 目前，us-east-1 区域的 OpenSearch 服务域使用来自不同机构的证书。我们计划近期将该区域更新为使用这些新的证书颁发机构。

由于版本兼容性，自定义插件安装失败

问题：由于插件和正在运行的 OpenSearch 实例之间的版本不匹配，插件安装失败。系统返回以下错误：

```
PluginValidationFailureReason : The provided plugin could not be loaded.
```

原因：该插件是为 OpenSearch $ 编译的*{MAJOR}*。 $*{MINOR}.{PATCH}*，但您的环境正在运行 OpenSearch $ *{MAJOR}*。 $ *{MINOR}* .0。 OpenSearch 出于稳定性和安全性的考虑，需要在插件和核心 OpenSearch安装之间进行精确的版本匹配。

可能的修复方法：使用 OpenSearch 版本 $ 构建插件*{MAJOR}*。 $ *{MINOR}* .0 以匹配您的集群版本。

验证和更新版本 OpenSearch

1. 使用集群的 API 或控制面板运行以下命令。将 *default placeholder values* 替换为您自己的信息。

   API 请求：

   ```
   curl -X GET your-opensearch-endpoint/
   ```

   仪表板中的开发工具控制台：

   ```
   GET /
   ```

   该命令以以下格式返回信息。

   ```
   {
     "name": "node-id",
     "cluster_name": "account-id:domain-name",
     "cluster_uuid": "cluster-uuid",
     "version": {
       "distribution": "opensearch",
       "number": "2.17.0",
       "build_type": "tar",
       "build_hash": "unknown",
       "build_date": "2024-12-17T11:00:09.799828091Z",
       "build_snapshot": false,
       "lucene_version": "9.11.1",
       "minimum_wire_compatibility_version": "7.10.0",
       "minimum_index_compatibility_version": "7.0.0"
     },
     "tagline": "The OpenSearch Project: https://opensearch.org/"
   }
   ```

2. 如果版本号不是 $ *{MAJOR}*。 $ *{MINOR}* .0，通过执行以下操作来重建插件：

a. 更新插件`descriptor.properties`以指定版本 $ *{MAJOR}*。 $ *{MINOR}* .0。
b. 使用适用于您的项目类型的命令重新生成插件。
c. 使用新建的文件运行 update- package 命令。`.zip`
d. 运行 ass ociate-packag e 命令以关联您在上一步中运行该`update-package`命令时创建的最新插件版本。

Amazon OpenSearch 服务的文档历史记录

本主题介绍了 Amazon OpenSearch 服务的重要更改。服务软件更新增加了对新功能、安全补丁、错误修复和其他改进的支持。要使用新功能，您可能需要更新域上的服务软件。有关更多信息，请参阅 the section called “服务软件更新”。

服务功能将逐步推广到服务可用 Amazon Web Services 区域 的地方。我们仅在第一次发布时更新了此文档。我们不提供有关区域可用性的信息，也不会宣布后续区域支持情况。有关服务功能的区域可用性以及订阅更新通知的信息，请参阅新增内容 Amazon？

如需有关更新的通知，您可以订阅 RSS 源。

> **Note**
>
> 补丁版本：以“-P”和数字结尾的服务软件版本（例如 R20211203-P4）是补丁版本。补丁可能包括性能改进、次要错误修复以及安全修复或状态改进。由于补丁不包括新功能或重大更改，因此通常不会对用户或文档产生直接影响，这就是本文档历史记录中没有包含每个补丁的具体细节的原因。

变更	说明	日期
新主题：Atlassian 与 Ingestion 的集成 OpenSearch	现在，你可以使用 Amazon OpenSearch Ingestion 管道来处理来自 Atlassian Jira 和 Confluence 服务的数据。通过这种集成，您可以将完整的 Jira 项目和 Confluence 空间同步到，从而创建统一的可搜索知识库。 OpenSearch该集成通过持续监控和自动同步更新来保持实时相关性，同时为特定项目、问题类型和内容提供灵活的筛选选项。有关更多信息，请参阅在 Atlass ian 服务中使用 OpenSearch 摄取管道。	2025 年 6 月 5 日

使用自定义插件管理 Amazon CLI	现在，您可以使用管理亚马逊 OpenSearch 服务的自定义插件 Amazon CLI。此功能允许您通过简化的命令行工作流程安装、更新、保护和卸载自定义插件。您还可以使用 Amazon KMS 密钥保护插件，并在更新到较新版本时管理插件升级 OpenSearch。有关更多信息，请参阅使用管理自定义插件 Amazon CLI和 Amazon S OpenSearch ervice 定制包 Amazon KMS 集成	2025 年 5 月 27 日
支持 OpenSearch 2.18 和 2.19	亚马逊 OpenSearch 服务支持 2.18 和 2.19 OpenSearch 版本。有关更多信息，请参阅 OpenSearch 2.19 和 2. 18 版本说明。有关 OpenSearch 2.18 和 2.19 操作的信息，请参阅 OpenSearch REST API 参考或特定插件的 API 参考。	2025 年 4 月 30 日

新增支持在 OpenSearch Ingestion 中指定管道角色	当您在 Amazon OpenSearch Ingestion 中创建或更新管道时，您不再是 YAML 格式管道配置中的指定管道角色。相反，在控制台中，您可以在新的管道角色字段中指定角色。如果您使用的是 Amazon CLI，则使用新`--pipeline-role-arn` 参数。有关更多信息，请参阅以下主题： • 在 Amazon OpenSearch Ingestion 中设置角色和用户 • 创建 Amazon OpenSearch Ingestion 管道 • 教程：使用 Amazon Ingestion 将数据提取到域中 OpenSearch • 教程：使用 Amazon Ingestion 将数据提取到集合中 OpenSearch • CreatePipeline • UpdatePipeline	2025年4月22日
新主题	疑难解答主题自定义插件安装由于版本兼容性而失败，说明了遇到错误时应采取的步骤`PluginValidationFailureReason : The provided plugin could not be loaded` 。此错误通常是由插件和正在运行的 OpenSearch 实例之间的版本不匹配造成的。	2025年4月21日

新功能： OpenSearch UI 兼容跨集群搜索	OpenSearch 用户界面现在与跨集群搜索兼容。这使您可以将 OpenSearch 用户界面合二为一 Amazon Web Services 区域 来访问不同区域的集群。这是通过将其配置为与同一区域中的集群连接的远程集群来完成的。有关更多信息，请参阅通过跨集群搜索进行跨区域和跨账户数据访问。	2025 年 4 月 16 日
修改内容：“在 Amazon OpenSearch 服务中使用 OpenSearch 用户界面”	我们全面修改并扩展了 “在亚马逊 OpenSearch 服务中使用 OpenSearch 用户界面” 一章，包括扩展的程序、其他设置细节以及亚马逊 OpenSearch 服务对 UI 支持的新版本历史记录。 OpenSearch	2025 年 4 月 15 日
Amazon Q 亚马逊 OpenSearch 服务开发者	Amazon Q 与亚马逊 OpenSearch 服务的集成提供了以下生成功能： • 使用自然语言生成可视化效果 • 查看警报摘要和见解 • 在 “发现” 页面上查看 Amazon Q 生成的查询结果摘要 • 查看推荐的异常探测器 • 访问 Amazon Q 聊天室了解 OpenSearch相关问题	2025 年 3 月 31 日

AmazonOpenSearchServiceRolePolicy 托管策略已更新	该策略 OpenSearch 允许更新仅由管理的任何 Amazon IAM Identity Center 应用程序的访问范围 OpenSearch。	2025 年 3 月 28 日
OR2 和 Amazon OpenSearch 服务的 OM2 实例	Amazon OpenSearch 服务现在支持 OR2 和 OM2 实例。在内部基准测试中， OR2索引吞吐量比 R7 高出 26%，比 R OR1 7 高 70%。 OM2 比 m7g好15%，比 OR1 m7g高出 66%。	2025 年 3 月 25 日
直接查询 CloudWatch 日志和安全湖	Amazon Ser OpenSearch vice 现在支持直接查询 CloudWatch 日志和安全湖中的数据。	2024 年 12 月 1 日
k-nn 指数	从 OpenSearch 版本 2.17 开始，您可以将 k-nn 索引移到 UltraWarm 和冷存储层。	2024 年 11 月 13 日
OpenSearch 2.17 支持	亚马逊 OpenSearch 服务现在支持 2.17 OpenSearch 版。此版本包含 2.16 和 2.17 版本中包含的所有功能。有关更多信息，请参阅 2.16 和 2. 17 版本说明。	2024 年 11 月 13 日
OpenSearch 2.15 支持	亚马逊 OpenSearch 服务现在支持 2.15 OpenSearch 版。此版本包含 2.14 和 2.15 版本中包含的所有功能。有关更多信息，请参阅 2.14 和 2. 15 发行说明。	2024 年 10 月 11 日

新的服务相关角色	Amazon S OpenSearch ervice 添加了一个名为的服务相关角色`AWSServiceRoleForOpensearchIngestionSelfManagedVpce`，该角色允许 Amazon OpenSearch Ingestion 向具有自我管理 VPC 终端 Amazon CloudWatch 节点的管道发送指标数据。	2024 年 6 月 12 日
直接查询	Amazon S OpenSearch ervice 现在支持直接对存储在 Amazon S3 中的数据运行查询，无需将数据提取到 OpenSearch 索引中。	2024 年 5 月 22 日
OpenSearch 2.13 支持	亚马逊 OpenSearch 服务现在支持 2.13 OpenSearch 版。此版本包含版本 2.12 和 2.13 的所有功能。有关更多信息，请参阅 2.12 和 2.13 发布说明。	2024 年 5 月 21 日
Amazon OpenSearch Ingestion 支持 Data Prepper 版本 2.7	Amazon OpenSearch Ingestion 增加了对 Data Prepper 版本 2.7 的支持。有关更多信息，请参阅 2.7 发布说明。	2024 年 4 月 4 日
Amazon Web Services 服务 OpenSearch 无服务器集合的私有访问权限	现在 Amazon Web Services 服务，您可以根据网络访问策略授予特定的（例如 Amazon Bedrock）访问您的 OpenSearch 无服务器集合的权限。	2024 年 3 月 28 日

就地 EBS 更新	现在，无需在 Amazon Service 中部署蓝/绿，即可对域名进行一些 EBS 更改。 OpenSearch	2024 年 2 月 14 日
配置更改可见性	现在，您可以在亚马逊 OpenSearch 服务控制台中使用配置 API 来跟踪域配置更改。	2024 年 2 月 6 日
向量搜索集合正式发布	Amazon OpenSearch Serverless 矢量搜索集合现已正式上市。在预览阶段取得了以下显著改进： • 向量搜索集合现在支持包含数十亿个向量的工作负载，每个向量最多有 128 个维度。 • OpenSearch 仪表板现在支持矢量搜索集合。	2023 年 11 月 29 日
OR1 实例	Amazon OpenSearch 服务现在支持 OR1 实例类型。	2023 年 11 月 29 日
使用 Amazon S3 直接查询（预览版）	直接查询提供了一种完全托管的解决方案，可在交易数据写入 Amazon S3 存储桶后的几秒钟内即可在 Amazon S OpenSearch ervice 中提供这些数据。	2023 年 11 月 29 日
用于时间序列集合的 10TiB 容量	Amazon OpenSearch Serverless 为时间序列集合增加了对高达 10 TiB 的索引数据的支持。此版本还支持所有类型的集合的最大允许容量 OCUs 为 200，并且能够在创建集合时禁用备用副本。	2023 年 11 月 29 日

OpenSearch 2.11 支持	亚马逊 OpenSearch 服务现在支持 2.11 OpenSearch 版。此版本包含 2.10 和 2.11 版本的所有功能。有关更多信息，请参阅 2.10 和 2.11 版本注释。	2023 年 11 月 17 日
Amazon OpenSearch Ingestion 支持 Data Prepper 版本 2.6	Amazon OpenSearch Ingestion 增加了对 Data Prepper 版本 2.6 的支持。有关更多信息，请参阅 2.6 发布说明。此外，您还可以将 Amazon Dynamodb 指定为管道源。有关更多信息，请参阅在 Amaz OpenSearch on DynamoDB 中使用采集管道。	2023 年 11 月 17 日
Amazon OpenSearch Ingestion 支持 Data Prepper 版本 2.5	Amazon OpenSearch Ingestion 增加了对 Data Prepper 版本 2.5 的支持。有关更多信息，请参阅 2.5 版本注释。此外，您现在可以将 OpenSearch 服务域或 OpenSearch 无服务器集合指定为管道源。有关更多信息，请参阅 Data Prepper 文档中的OpenSearch 源插件。	2023 年 11 月 17 日
CloudFormation 用于远程推理的模板	为了简化语义搜索的远程推理设置，Amazon S OpenSearc h ervice 在控制台中提供了一个 Amazon CloudFormation 模板，可以自动执行模型配置过程。	2023 年 11 月 7 日

服务相关角色策略更新	添加服务相关角色策略分配和取消`AmazonOpenSearchServiceRolePolicy` 分配 IPv6 地址所需的权限。已弃用的 Elasticsearch 策略 `AmazonElasticsearchServiceRolePolicy` 也已更新，以确保向后兼容。	2023 年 10 月 26 日
Amazon OpenSearch 无服务器生命周期政策	Amazon OpenSearch Serverless 引入了索引生命周期策略，以简化对数据保留和删除的管理。现在，您可以在控制台中使用 APIs 或配置界面来设置时间序列集合的数据保留策略，从而无需创建每日索引或脚本来删除旧数据。	2023 年 10 月 25 日
Im4gn 实例支持	亚马逊 OpenSearch 服务现在支持 im4gn 实例类型。Im4gn 实例针对管理大型数据集和每个 vCPU 都需要高存储密度的工作负载进行了优化。	2023 年 10 月 20 日
管理选项	Amazon Ser OpenSearch vice 现在提供了多种管理选项，如果您需要对域名问题进行故障排除，这些选项可以提供精细控制。这些选项包括在数据节点上重新启动 OpenSearch 进程的能力和重新启动数据节点的能力。	2023 年 10 月 17 日

可选插件	亚马逊 OpenSearch 服务增加了对四个新的语言分析器插件的支持：Nori（韩语）、Sudachi（日语）、拼音（中文）和 STConvert 分析（中文），以及 Amazon Personalize 搜索排名插件。	2023 年 10 月 16 日
OpenSearch 2.9 支持	亚马逊 OpenSearch 服务现在支持 2.9 OpenSearch 版。此版本包含 2.8 和 2.9 版本的所有功能。有关更多信息，请参阅 2.8 和 2.9 版本注释。	2023 年 10 月 2 日
ML 连接器	亚马逊 OpenSearch 服务增加了对机器学习 (ML) 连接器的支持。连接器便于访问托管在其他 Amazon Web Services 服务平台或第三方机器学习 (ML) 平台上的机器学习模型。	2023 年 9 月 6 日
Amazon OpenSearch Ingestion 增加了对 Data Prepper 版本 2.4 的支持	Amazon OpenSearch Ingestion 增加了对 Data Prepper 版本 2.4 的支持。有关更多信息，请参阅 2.4 版本注释。此外，您现在可以将 Amazon Managed Streaming for Apache Kafka（Amazon MSK）指定为管道源。	2023 年 8 月 31 日
用于时间序列采集的 6 TiB 容量	Amazon OpenSearch Serverless 为时间序列集合增加了对高达 6 TiB 的索引数据的支持。此版本还支持搜索和时间序列集合的最大允许容量为 100 OCUs 。	2023 年 8 月 15 日

向量搜索集合	Amazon OpenSearch Serverless 增加了创建矢量搜索集合的选项，您可以使用该集合来存储矢量嵌入，从而实现相似性和语义搜索。	2023 年 7 月 26 日
OpenSearch 2.7 支持	亚马逊 OpenSearch 服务现在支持 2.7 OpenSearch 版。此版本包含 2.6 和 2.7 版本的所有功能。有关更多信息，请参阅 2.6 和 2.7 版本注释。	2023 年 7 月 10 日
Data Prepper 2.3 支持	Amazon OpenSearch Ingestion 增加了对 Data Prepper 版本 2.3 的支持。有关更多信息，请参阅 2.3 版本注释。此外，您现在可以将 Amazon Security Lake 指定为管道源。	2023 年 6 月 26 日
带待机功能的多可用区	Amazon S OpenSearch ervice 增加了跨三个可用区 (AZs) 部署域的选项，每个可用区都包含完整的数据副本，其中一个可用区中的节点 AZs 充当备用区域。带待机功能的多可用区部署选项可在基础设施出现故障时提供 99.99% 的可用性和稳定的性能。	2023 年 5 月 3 日

新的服务相关角色	亚马逊 OpenSearch 服务添加了一个名为的服务相关角色`AWSServiceRoleForAmazonOpenSearchIngestionService`，该角色允许 Amazon OpenSearch Ingestion 向发送指标数据。Amazon CloudWatch	2023 年 4 月 26 日
Amazon OpenSearch Ingestion	Amazon OpenSearch Ingestion 是一个完全托管的数据收集器，可向 OpenSearch 服务域和 OpenSearch 无服务器集合提供实时日志和跟踪数据。 OpenSearch 通过摄取，您无需使用 Logstash 或 Jaeger 等第三方解决方案将数据提取到您的域名和集合中。	2023 年 4 月 26 日
OpenSearch 2.5 支持	亚马逊 OpenSearch 服务现在支持 2.5 OpenSearch 版。此版本包含 2.4 和 2.5 版本的所有功能。有关更多信息，请参阅 2.4 和 2.5 版本注释。	2023 年 3 月 13 日

非高峰期维护窗口	Amazon S OpenSearch ervice 增加了非高峰时段，即每天 10 小时、低流量的时间段，在此期间，它可以安排需要部署的服务软件更新和自动调整优化。 blue/green 非高峰时段更新有助于最大限度地减少高流量时段对集群专用主节点造成的压力。 对于在 2 月 16 日之后创建的新域，非高峰窗口会自动配置为当地时间晚上 10:00 至上午 8:00 之间。对于现有域，您需要指明窗口。	2023 年 2 月 16 日
在域创建期间配置 SAML 身份验证	Amazon OpenSearch 服务现在支持在创建域名时配置 SAML 身份验证。以前，您必须在创建域后才能配置 SAML 选项。	2023 年 2 月 1 日
VPC 域的远程重新索引	Amazon OpenSearch 服务增加了两个域之间的 VPC 终端节点连接选项。现在，无需反向代理即可使用远程重新索引将索引从一个 VPC 域复制到另一个 VPC 域。您的 VPC 域必须运行服务软件 R20221114 或更高版本才能使用此功能。	2023 年 1 月 31 日

Amazon OpenSearch 无服务器正式上市	Amazon OpenSearch Serverless 现已正式上市。在预览阶段取得了以下显著改进： • 现在，当收集端点的流量减少 OCUs 时，容量可以缩小到配置的最小值。 • 允许索引和 OCUs 搜索的最大值从 20 增加到 50。每个 OCU 都包含足够的热临时存储，可存储 120 GiB 的索引数据。 • 现在，您可以在创建集合时配置数据访问设置，而不必在单独的工作流程中进行配置。	2023 年 1 月 25 日
异步试运行	Amazon S OpenSearch ervice 现在支持异步试运行，允许您在更改配置之前执行验证检查，并通知您更改是否会导致部署。 blue/green	2023 年 1 月 19 日
新的服务相关角色	Amazon Serv OpenSearch ice 添加了一个名为的服务相关角色`AWSServiceRoleForAmazonOpenSearchServerless` ，该角色允许 OpenSearch Serverless 向发送指标数据。 Amazon CloudWatch	2022 年 11 月 29 日

Amazon OpenSearch 无服务器预览	Amazon OpenSearch Serverless 是一种针对亚马逊 OpenSearch 服务的按需、自动扩展、无服务器配置。Serverless 消除了配置、配置和调整集群的操作复杂性。OpenSearch	2022 年 11 月 29 日
OpenSearch 2.3 支持	亚马逊 OpenSearch 服务现在支持 2.3 OpenSearch 版。此版本包含 2.0、2.1 和 2.2 部分版本的所有功能。有关更多信息，请参阅 2.0、2.1、2.2 和 2.3 发行说明。版本 2.3 包含重大更改。有关更多信息，请参阅支持的升级途径。	2022 年 11 月 15 日
通知插件支持	Amazon S OpenSearch ervice 现在支持通知插件，该插件为来自 OpenSearch 插件的所有通知提供了一个中心位置。从版本 2.0 开始，警报目标已经停用，取而代之的是通知通道。	2022 年 11 月 15 日
Kibana 7.1.1 支持	运行 Elasticsearch 7.1 的亚马逊 OpenSearch 服务域现在支持 Kibana 7.1.1 的最新补丁版本，该补丁增加了错误修复并提高了安全性。当您将 7.1 域更新为服务软件 R20221114 时， OpenSearch 服务会自动将其升级到此补丁版本。	2022 年 11 月 15 日

Kibana 6.8.13 支持	运行 Elasticsearch 6.8 的亚马逊 OpenSearch 服务域现在支持 Kibana 6.8.13 的最新补丁版本，该补丁增加了错误修复并提高了安全性。当你将 6.8 域更新为服务软件 R20221114 时， OpenSearch 服务会自动将其升级到此补丁版本。	2022 年 11 月 15 日
Kibana 6.3.2 支持	运行 Elasticsearch 6.3 的亚马逊 OpenSearch 服务域现在支持 Kibana 6.3.2 的最新补丁版本，该补丁增加了错误修复并提高了安全性。当你将 6.3 域更新为服务软件 R20221114 时， OpenSearch 服务会自动将其升级到此补丁版本。	2022 年 11 月 15 日
Amazon PrivateLink	借助亚马逊 OpenSearch 服务托管的 VPC 终端节点，您可以使用接口 VPC 终端节点直接连接到 OpenSearch 服务 VPC 域，而不必通过互联网进行连接。 OpenSearch 服务管理的 VPC 终端节点只能在配置终端节点的 VPC 内访问，或者在路由表和安全组允许的情况下，从与配置终端节点的 VPC 的任何 VPCs 对等节点进行访问。您的 VPC 域必须运行服务软件 R20220928 或更高版本，才能连接到接口 VPC 端点。	2022 年 11 月 7 日

错误修复和性能改进	服务软件 R20220928 包括错误修复和性能增强，其中包括改进的 SAML 日志记录。此更新还将默认租户更改为 `Global` 而不是 `Private`。	2022 年 10 月 3 日
改进的 API 参考	Amazon OpenSearch 服务提供了经过改进的、包罗万象的配置 API 参考。新的参考包含所有可用的操作和数据类型、示例请求和响应语法，以及指向所有支持语言的相应 SDK 参考的链接。	2022 年 9 月 13 日
蓝/绿验证	现在，Amazon S OpenSearch ervice 会在 blue/green 部署之前执行验证检查，如果您的域名不符合更新条件，则会显示验证错误。	2022 年 8 月 16 日
OpenSearch 1.3 支持	亚马逊 OpenSearch 服务现在支持 1.3 OpenSearch 版。有关更多信息，请参阅 1.3 版本注释。	2022 年 7 月 27 日
ML Commons 插件支持	Amazon S OpenSearch ervice 增加了对 ML Commons 插件的支持，该插件通过传输和 REST API 调用提供了一组常见的机器学习算法。您还可以通过 PPL 命令与 ML Commons 插件进行交互。	2022 年 7 月 27 日

gp3 卷支持	Amazon OpenSearch 服务增加了对 gp3 EBS 通用型 SSD 卷类型的支持。您可以在创建或修改卷时指定额外的预调配 IOPS 和吞吐量。	2022 年 7 月 26 日
增强了最佳实践文档	Amazon S OpenSearch ervice 文档提供了有关创建和操作 OpenSearch 服务域的改进操作最佳实践和一般建议。	2022 年 7 月 6 日
与服务限额的集成	现在，您可以从 Service Quotas 控制台查看亚马逊 OpenSearch 服务的配额并请求增加配额。	2022 年 6 月 29 日
对 API 进行基于标签的 OpenSearch 访问控制	现在，您可以使用标签来控制对的访问 OpenSearch APIs。以前，您只能使用标签来控制对配置 API 的访问。	2022 年 6 月 16 日
跨区域的跨集群搜索	现在， Amazon Web Services 区域 只要两个域都运行 Elasticsearch 7.10 或更高版本或任何版本，就支持跨集群搜索。OpenSearch	2022 年 6 月 14 日
单个 Kibana 5.6 支持	亚马逊 OpenSearch 服务增加了对单个 Kibana 5.6.16 的支持。借助单个 Kibana 5.6.16，您可以在连接到 Elasticsearch 版本 5.1、5.3、5.5 和 5.6 时将 Kibana 5.6 作为前端。您必须在服务软件 R20220323 或更高版本上才能使用单个 Kibana 5.6。	2022 年 4 月 22 日

R20220323-P1	亚马逊 OpenSearch 服务最近发布了服务软件更新 R20220323，但由于出现问题，该更新随后被撤回。我们建议您将域更新至补丁发行版 R20220323-P1 或更高版本，这将修复该问题。	2022 年 4 月 22 日
OpenSearch 1.2 支持	亚马逊 OpenSearch 服务现在支持 1.2 OpenSearch 版。有关更多信息，请参阅 1.2 版本注释。	2022 年 4 月 22 日
可观察性	Amazon Serv OpenSearch ice OpenSearch 控制面板的默认安装包括可观察性插件，您可以使用该插件使用管道处理语言 (PPL) 对数据驱动的事件进行可视化，从而浏览和查询您的数据。该插件需要 OpenSearch 1.2 或更高版本以及服务软件 R20220323 或更高版本。	2022 年 4 月 22 日
Kibana 7.7.1 支持	运行 Elasticsearch 7.7 的亚马逊 OpenSearch 服务域现在支持 Kibana 7.7 的最新补丁版本，该补丁增加了错误修复并提高了安全性。当您将 7.7 域更新为服务软件 R20220323 或更高版本时， OpenSearch 服务会自动将其升级到此补丁版本。	2022 年 4 月 22 日

JVM 内存压力指标更改	Amazon S OpenSearch ervice 更改了 `JVMMemoryPressure` CloudWatch 指标的逻辑，以更准确地反映内存利用率。以前，这些指标只考虑 JVM 堆的老一代内存池。在这次更改后，该指标还将考虑新一代内存池。将域更新到服务软件 R20220323 后，您可能会看到 `JVMMemoryPressure` 、`MasterJVMMemoryPressure` 和/或 `WarmJVMMemoryPressure` 指标增加。	2022 年 4 月 22 日
将自定义字典与 IK（中文）分析插件结合使用	Amazon S OpenSearch ervice 现在支持使用带有 IK（中文）分析插件的自定义词典。	2022 年 4 月 22 日
现有域上的跨集群复制	Amazon S OpenSearch ervice 取消了只能在 2020 年 6 月 3 日当天或之后创建的域上实施跨集群搜索和跨集群复制的限制。现在，您可以在所有域上启用这些功能，而无论它们是在何时创建的。两个域都必须使用服务软件 R20220323 或更高版本。	2022 年 4 月 22 日
蓝/绿部署可见性	Amazon Ser OpenSearch vice 现在可以更清楚地了解蓝/绿部署的进度。您可以在控制台中或使用配置 API 监控这些详细信息。	2022 年 1 月 27 日

对现有域的精细访问控制	您现在可以对现有域启用精细访问控制。您可以为开放/基于 IP 的访问策略启用临时迁移期，以确保用户可以在您创建和映射角色的时候继续访问您的域。对现有域启用精细访问控制需要服务软件 R20211203 或更高版本。	2022 年 1 月 6 日
重命名的 OpenSearch 仪表板角色	对于服务软件 R20211203 ，`kibana_user` 角色已重命名为 `opensearch_dashboards_user` ，`kibana_read_only` 已重命名为 `opensearch_dashboards_read_only` 。此更改适用于所有新创建的 1 OpenSearch . x 个域名。对于升级到服务软件 R20211203 的现有 OpenSearch 域，角色保持不变。	2022 年 1 月 4 日
OpenSearch 1.1 支持	亚马逊 OpenSearch 服务现在支持 1.1 OpenSearch 版。有关更多信息，请参阅 1.1 发行说明。	2022 年 1 月 4 日
ISM 可视化编辑器	默认安装的 Amazon OpenSearch 服务 OpenSearch 控制面板现在支持 ISM 策略的可视化编辑器。此功能需要 OpenSearch 1.1 或更高版本。	2022 年 1 月 4 日

防止跨服务混淆代理更新	Amazon S OpenSearch ervice 支持在 IAM 资源策略中使用`aws:SourceArn` 和`aws:SourceAccount` 全局条件上下文密钥来防止出现混淆的代理问题。您必须在服务软件 R20211203 或更高版本上才能使用这些条件键。	2022 年 1 月 4 日
Log4j 补丁	服务软件 R20211203-P2 按照 CVE-2021-44228 和 CVE-2021-45046 中公告的建议更新了 OpenSearch 服务中使用的 Log4j 版本。该补丁适用于运行所有版本 OpenSearch 和 Elasticsearch 的域名。OpenSearch 服务将继续在内部更新各种 Log4j 版本，它们不一定仅限于最新版本的 Log4j。您的域上的 Log4j 版本取决于该域正在运行的软件版本。但无论使用哪个 Log4j 版本，只要运行的是 R20211203-P2 或更高版本，您的域都将包含解决 CVE-2021-44228 和 CVE-2021-45046 问题所需的 Log4j 更新。	2021 年 12 月 15 日
跨集群复制	跨集群复制允许您将索引、映射和元数据从一个 OpenSearch 服务域复制到另一个服务域。跨集群复制需要运行 Elasticsearch 7.10 或 1.1 或 OpenSearch 更高版本的域。	2021 年 10 月 5 日

新的 Amazon托管策略	Amazon S OpenSearch ervice 的推出包括新的 Amazon托管策略和旧政策的弃用。	2021 年 9 月 8 日
Kibana 6.4.3 支持	运行旧版 Elasticsearch 6.4 版本的亚马逊 OpenSearch 服务域现在支持 Kibana 6.4 的最新补丁版本，该补丁增加了错误修复并提高了安全性。OpenSearch 服务会自动将域升级到此补丁版本。	2021 年 9 月 8 日
数据流	Amazon S OpenSearch ervice 增加了对数据流的支持，从而简化了管理时间序列数据的过程。您的域必须运行 OpenSearch 1.0 或更高版本才能使用数据流。	2021 年 9 月 8 日
亚马逊 OpenSearch 服务	Amazon 重命名亚马逊 OpenSearch 服务以移除传统的 “Elasticsearch” 品牌。亚马逊 OpenSearch 服务支持 OpenSearch 传统的 Elasticsearch 操作系统。创建集群时，您可以选择使用哪个搜索引擎。 OpenSearch 该服务与 Elasticsearch OSS 7.10（该软件的最终开源版本）具有广泛的兼容性。	2021 年 9 月 8 日

冷存储	冷存储是针对不常访问或历史数据的新存储层。冷索引只占用 S3 存储空间，没有附加到这些索引的计算。冷存储需要运行 Elasticsearch 7.9 或更高版本以及服务软件 R20210426 或更高版本的域。	2021 年 5 月 13 日
基于 ARM 的 Graviton 实例	亚马逊 OpenSearch 服务现在支持基于 ARM 的 Graviton 实例类型（M6G、C6G、R6G 和 R6GD）。引力实例类型可用于运行 Elasticsearch 7.9 或更高版本以及服务软件 R20210331 或更高版本的新域和现有域。	2021 年 5 月 4 日
ISM 模板	Amazon S OpenSearch ervice 增加了对 ISM 模板的支持，如果索引与策略中定义的模式相匹配，则允许您自动将 ISM 策略附加到索引。ISM 模板需要服务软件 R20210426 或更高版本。此更新还会启用 `policy_id` 设置，这意味着您不能再使用索引模板将 ISM 策略应用于新创建的索引。此更新为使用此设置的现有 CloudFormation 模板引入了一项重大更改。	2021 年 4 月 27 日
Elasticsearch 7.10 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 7.10。有关更多信息，请参阅 7.10 发行说明。	2021 年 4 月 21 日

异步搜索	Amazon S OpenSearch ervice 现在支持异步搜索，允许您在后台运行搜索请求。异步搜索需要运行 Elasticsearch 7.10 或更高版本以及服务软件 R20210331 或更高版本的域。	2021 年 4 月 21 日
适用于配置 API 的基于标签的访问控制	现在，您可以使用 Amazon 标签来控制对 Amazon ES 配置 API 的访问权限。	2021 年 3 月 2 日
自动调整	Amazon S OpenSearch ervice 添加了 Auto-Tune，它使用集群中的性能和使用率指标来建议对节点上 JVM 设置的更改。自动调整需要运行 Elasticsearch 6.7 或更高版本以及服务软件 R20201117 或更高版本的域。	2021 年 2 月 24 日
跟踪分析	Kibana for Amazon S OpenSearch ervice 的默认安装现在包括跟踪分析插件，可让您监控来自分布式应用程序的跟踪数据。该插件需要运行 Elasticsearch 7.9 或更高版本以及服务软件 R20210201 或更高版本的域。	2021 年 2 月 17 日

分片指标	Amazon S OpenSearch ervice 添加了以下用于跟踪分片状态的 CloudWatch 指标：`Shards.active` 、`Shards.unassigned` 、`Shards.delayedUnassigned` `Shards.activePrimary` 、`Shards.initializing` 、`Shards.relocating` 。这些指标可用于运行服务软件 R20210201 或更高版本的域。	2021 年 2 月 17 日
Kibana 报告	默认安装的 Kibana for Amazon OpenSearch 服务现在支持 “发现”、“可视化” 和 “控制面板” 页面的按需报告。此功能需要 Elasticsearch 7.9 或更高版本以及服务软件 R20210201 或更高版本。	2021 年 2 月 17 日
Kibana 5.6.16 支持	运行 Elasticsearch 5.6 的亚马逊 OpenSearch 服务域现在支持 Kibana 5.6 的最新补丁版本，该补丁增加了错误修复并提高了安全性。Amazon ES 将域自动升级到此补丁版本。	2021 年 2 月 17 日
现有域的加密	亚马逊 OpenSearch 服务现在支持在运行 Elasticsearch 6.7 或更高版本的现有域上启用静态数据 node-to-node加密和加密。启用这些设置后，您无法禁用它们。	2021 年 1 月 27 日

远程重建索引	Amazon S OpenSearch ervice 现在支持远程重新索引，允许您从远程域迁移索引。此功能需要服务软件 R20201117 或更高版本。	2020 年 11 月 24 日
管道式处理语言	亚马逊 OpenSearch 服务现在支持管道处理语言 (PPL)，这是一种查询语言，允许您使用管道 (|) 语法来查询存储在 Elasticsearch 中的数据。此功能需要服务软件 R20201117 或更高版本。要了解更多信息，请参阅。	2020 年 11 月 24 日
Kibana 笔记本	Amazon S OpenSearch ervice 增加了对 Kibana 笔记本的支持，允许您将实时可视化和叙事文本组合到一个界面中。此功能需要服务软件 R20201117 或更高版本。	2020 年 11 月 24 日
甘特图	默认安装的 Kibana for Amazon OpenSearch 服务现在支持一种新的可视化类型，即甘特图。此功能需要服务软件 R20201117 或更高版本。	2020 年 11 月 24 日
Elasticsearch 7.9 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 7.9。有关更多信息，请参阅 7.9 发行说明。	2020 年 11 月 24 日

异常检测更新	Amazon S OpenSearch ervice 的异常检测增加了对高基数的支持，允许您使用 IP 地址、产品 ID、国家/地区代码等维度对异常进行分类。此功能需要服务软件 R20201117 或更高版本。	2020 年 11 月 24 日
动态字典更新	Amazon S OpenSearch ervice 现在允许您更新搜索分析器，而无需重新编制索引。您可以更新部分或全部域上的字典文件，Amazon ES 随着时间的推移跟踪软件包版本，以便您了解更改内容和时间的历史记录。此功能需要服务软件 R20201019 或更高版本。	2020 年 11 月 17 日
自定义端点	亚马逊 OpenSearch 服务现在支持自定义终端节点，允许您为亚马逊 ES 域名提供一个新的 URL。如果您曾经交换过域，则可以保留相同的 URL。此功能需要服务软件 R20201019 或更高版本。	2020 年 11 月 5 日
新增语言插件	亚马逊 OpenSearch 服务现在在运行 Elasticsearch 7.7 或更高版本且服务软件 R20201019 或更高版本的域名上支持 IK（中文）分析、越南语分析和泰语分析插件。	2020 年 10 月 28 日
Elasticsearch 7.8 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 7.8。有关更多信息，请参阅 7.8 发行说明。	2020 年 10 月 28 日

适用于 Kibana 的 SAML 身份验证	Amazon S OpenSearch ervice 现在支持 Kibana 的 SAML 身份验证，允许您使用第三方身份提供商登录 Kibana、管理精细访问控制、搜索数据和构建可视化效果。此功能需要服务软件 R20201019 或更高版本。	2020 年 10 月 27 日
T3 实例	Amazon OpenSearch 服务现在支持`t3.small`和`t3.medium` 实例类型。	2020 年 9 月 23 日
审核日志	Amazon S OpenSearch ervice 现在支持对您的数据进行审核日志，它允许您跟踪失败的登录尝试、用户对索引、文档和字段的访问等。此功能需要服务软件 R20200910 或更高版本。	2020 年 9 月 16 日
UltraWarm 更新	UltraWarm for Amazon S OpenSearch ervice 添加了新指标、新设置、更大的迁移队列和取消 API。这些更新需要服务软件 R20200910 或更高版本。有关更多信息，请参阅 。	2020 年 9 月 14 日
学习排名	Amazon S OpenSearch ervice 现在支持开源 Learning to Rank 插件，该插件允许您使用机器学习技术来提高搜索相关性。此功能需要服务软件 R20200721 或更高版本。	2020 年 7 月 27 日

k-NN 余弦相似性	k 最近邻 (k-NN) 现在允许您通过余弦相似性以及欧氏距离搜索“最近邻域”。此功能需要服务软件 R20200721 或更高版本。	2020 年 7 月 23 日
gzip 压缩	Amazon S OpenSearch ervice 现在支持对大多数 HTTP 请求和响应进行 gzip 压缩，这样可以减少延迟并节省带宽。此功能需要服务软件 R20200721 或更高版本。	2020 年 7 月 23 日
Elasticsearch 7.7 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 7.7。有关更多信息，请参阅 7.7 发行说明。	2020 年 7 月 23 日
Kibana 地图服务	Kibana for Amazon OpenSearch 服务的默认安装现在包括 WMS 地图服务器，但印度和中国地区的域名除外。	2020 年 6 月 18 日
SQL 改进	Amazon Serv OpenSearch ice 的 SQL 支持现在支持许多新操作、用于数据探索的专用 Kibana 用户界面和交互式 CLI。	2020 年 6 月 3 日
跨集群搜索	Amazon Ser OpenSearch vice 允许您跨多个连接的域执行跨集群查询和聚合。	2020 年 6 月 3 日
异常检测	Amazon OpenSearch 服务允许您近乎实时地自动检测异常。	2020 年 6 月 3 日

UltraWarm	UltraWarm Amazon S OpenSearch ervice 的存储空间已退出公开预览版，现已正式发布。该功能现在支持更广泛的版本和 Amazon Web Services 区域. 有关更多信息，请参阅 。	2020 年 5 月 5 日
自定义字典	Amazon Ser OpenSearch vice 允许您上传用于集群的自定义词典文件。这些文件通过告诉 Elasticsearch 忽略某些高频词语或将多个词语视为等效，从而改善搜索结果。	2020 年 4 月 21 日
Elasticsearch 7.4 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 7.4。有关更多信息，请参阅受支持的版本。	2020 年 3 月 12 日
k-NN	亚马逊 OpenSearch 服务增加了对 k-nearest Neighbor (k-nn) 搜索的支持。k-nn 需要服务软件 R20200302 或更高版本。	2020 年 3 月 3 日
索引状态管理	Amazon S OpenSearch ervice 增加了索引状态管理 (ISM)，它允许您自动执行日常任务，例如在索引达到一定年龄时将其删除。此功能需要使用服务软件 R20200302 或更高版本。	2020 年 3 月 3 日

Elasticsearch 5.6.16 支持	Amazon Serv OpenSearch ice 现在支持 5.6 版的最新补丁版本，该版本增加了错误修复并提高了安全性。Amazon ES 将现有的 5.6 个域自动升级到此版本。请注意，此 Elasticsearch 版本错误地将其版本报告为 5.6.17。	2020 年 3 月 2 日
精细访问控制	Amazon S OpenSearch ervice 现在支持精细的访问控制，该控制可在索引、文档和字段级别提供安全保护，还支持 Kibana 多租户以及为您的集群提供可选的 HTTP 基本身份验证。	2020 年 2 月 11 日
UltraWarm 存储（预览）	Amazon Ser OpenSearch vice 新增 UltraWarm了一个使用 Amazon S3 的温存储层和先进的缓存解决方案来提高性能。对于不主动写入且查询频率较低的索引， UltraWarm 存储可显著降低每 GiB 的成本。	2019 年 12 月 3 日
中国区域的加密功能	中国（北京）区域和`cn-north-1` 中国（宁夏）区域现已提供静态数据 node-to-node加密`cn-northwest-1` 和加密功能。	2019 年 11 月 20 日
需要 HTTPS	现在，您可以要求 Amazon ES 域的所有流量都通过 HTTPS 到达。在配置域时，请选中 Require HTTPS (要求 HTTPS) 框。此功能需要服务软件 R20190808 或更高版本。	2019 年 10 月 3 日

Elasticsearch 7.1 和 6.8 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 7.1 和 6.8。有关更多信息，请参阅受支持的版本。	2019 年 8 月 13 日
每小时快照	Amazon S OpenSearch ervice 现在不是每日快照，而是按小时为运行 Elasticsearch 5.3 及更高版本的域拍摄快照，这样您就可以更频繁地进行备份来恢复数据。	2019 年 7 月 8 日
Elasticsearch 6.7 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 6.7。有关更多信息，请参阅受支持的版本。	2019 年 5 月 29 日
SQL 支持	亚马逊 OpenSearch 服务现在允许您使用 SQL 查询数据。SQL 支持需要服务软件 R20190418 或更高版本。	2019 年 5 月 15 日
5 系列实例类型	亚马逊 OpenSearch 服务现在支持 M5、C5 和 R5 实例类型。相较于上一代实例类型，这些新类型能够提供更高的性价比。有关更多信息，请参阅限制。	2019 年 4 月 24 日
Elasticsearch 6.5 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 6.5。	2019 年 4 月 8 日
提示	当来自一个或多个 Amazon ES 指数的数据满足特定条件时，亚马逊 OpenSearch 服务警报会通知您。警报需要服务软件 R20190221 或更高版本。	2019 年 3 月 25 日

三可用区支持	Amazon OpenSearch 服务现在支持许多地区的三个可用区。此版本还包含简化的控制台体验。此多 AZ 需要服务软件 R20181023 或更高版本。	2019 年 2 月 7 日
Elasticsearch 6.4 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 6.4。	2019 年 1 月 23 日
200 节点集群	Amazon ES 现在能让您创建最多包含 200 个数据节点的集群，总共 3 PB 的存储空间。	2019 年 1 月 22 日
服务软件更新	Amazon ES 现在允许为您的域手动更新服务软件以更快地受益于新功能或在低流量时间更新。要了解更多信息，请参阅。	2018 年 11 月 20 日
新 CloudWatch 指标	Amazon ES 在 Amazon ES 控制台中现在提供节点级指标和新的集群运行状况和实例运行状况选项卡。	2018 年 11 月 20 日
中国（北京）区域支持	亚马逊 OpenSearch 服务现已在 cn-north-1 区域推出，它支持 M4、C4 和 R4 实例类型。	2018 年 10 月 17 日
Node-to-node 加密	Amazon S OpenSearch ervice 现在支持 node-to-node加密，当 Amazon ES 在整个集群中分发数据时，它可以对您的数据进行加密。	2018 年 9 月 18 日
就地版本升级	Amazon OpenSearch 服务现在支持就地版本升级。	2018 年 8 月 14 日

Elasticsearch 6.3 和 5.6 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 6.3 和 5.6。	2018 年 8 月 14 日
错误日志	亚马逊 ES 现在允许你向亚马逊发布 Elasticsearch 错误日志。 CloudWatch	2018 年 7 月 31 日
中国（宁夏）区域预留实例	Amazon ES 现在在中国（宁夏）区域提供预留实例。	2018 年 5 月 29 日
预留实例	Amazon ES 现在提供预留实例支持。	2018 年 5 月 7 日

早期更新

下表介绍了 2018 年 5 月之前对 Amazon ES 的一些重要更改。

更改	描述	日期
适用于 Kibana 的 Amazon Cognito 身份验证	Amazon ES 现在提供了针对 Kibana 的登录页保护。要了解更多信息，请参阅the section called “控制面板的 Amazon Cognito 身份验证 OpenSearch ”。	2018 年 4 月 2 日
Elasticsearch 6.2 支持	亚马逊 OpenSearch 服务现在支持 Elasticsearch 版本 6.2。	2018 年 3 月 14 日
韩语分析插件	Amazon ES 现已支持 Seunjeon 韩语分析插件的内存优化版本。	2018 年 3 月 13 日
实例访问控制更新	对 Amazon ES 域中访问控制策略的更改现在可立即生效。	2018 年 3 月 7 日
PB 规模	Amazon ES 现支持 I3 实例类型和多达 1.5 PB 的总域存储空间。要了解更多信息，请参阅the section called “PB 规模”。	2017 年 12 月 19 日

更改	描述	日期
静态数据加密	Amazon ES 现在支持静态数据加密。要了解更多信息，请参阅the section called “静态加密”。	2017 年 12 月 7 日
Elasticsearch 6.0 支持	Amazon ES 现在支持 Elasticsearch 版本 6.0。有关迁移方面的注意事项和说明，请参阅 the section called “升级域”。	2017 年 12 月 6 日
VPC 支持	Amazon ES 现在允许您在 Amazon Virtual Private Cloud 中启动域。VPC 支持提供了额外的安全层并简化了 Amazon ES 和 VPC 中的其他服务之间的通信。要了解更多信息，请参阅the section called “VPC 支持”。	2017 年 10 月 17 日
慢速日志发布	Amazon ES 现在支持将慢速日志发布到 CloudWatch 日志。要了解更多信息，请参阅the section called “监控日志”。	2017 年 16 月 10 日
Elasticsearch 5.5 支持	Amazon ES 现在支持 Elasticsearch 版本 5.5。 您现在可以在不联系 Amazon Web Services 支持 的情况下还原自动快照，并使用 `_scripts` API 存储脚本。	2017 年 9 月 7 日
Elasticsearch 5.3 支持	Amazon ES 增加了对 Elasticsearch 版本 5.3 的支持。	2017 年 6 月 1 日
每个群集可拥有更多实例和 EBS 容量	Amazon ES 现在支持每个群集最多 100 个节点和 150 TB EBS 容量。	2017 年 4 月 5 日
加拿大 (中部) 和欧洲 (伦敦) 支持	Amazon ES 增加了对以下区域的支持：加拿大（中部）ca-central-1 和欧洲（伦敦）eu-west-2。	2017 年 3 月 20 日
更多实例和更大的 EBS 卷	Amazon ES 增加了对更多实例和更大 EBS 卷的支持。	2017 年 2 月 21 日
Elasticsearch 5.1 支持	Amazon ES 增加了对 Elasticsearch 版本 5.1 的支持。	2017 年 1 月 30 日
支持拼音分析插件	Amazon ES 现在内置集成拼音分析插件，可以对数据运行“声音”查询。	2016 年 12 月 22 日

更改	描述	日期
美国东部 (俄亥俄) 支持	Amazon ES 增加了对以下区域的支持：美国东部（俄亥俄）us-east-2。	2016 年 10 月 17 日
新的性能指标	Amazon ES 增加了性能指标 `ClusterUsedSpace` 。	2016 年 7 月 29 日
Elasticsearch 2.3 支持	Amazon ES 增加了对 Elasticsearch 版本 2.3 的支持。	2016 年 7 月 27 日
亚太地区 (孟买) 支持	Amazon ES 增加了对以下区域的支持：亚太地区（孟买）ap-south-1。	2016 年 6 月 27 日
每个群集更多实例	Amazon ES 将每个群集的最大实例数（实例数量）从 10 增加到 20。	2016 年 5 月 18 日
亚太地区 (首尔) 支持	Amazon ES 增加了对以下区域的支持：亚太地区（首尔）ap-northeast-2。	2016 年 1 月 28 日
Amazon ES	首次发布。	2015 年 10 月 1 日

Amazon 词汇表

有关最新 Amazon 术语，请参阅《Amazon Web Services 词汇表 参考资料》中的Amazon 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。