aws

开发人员指南

亚马逊 OpenSearch 服务

亚马逊 OpenSearch 服务: 开发人员指南

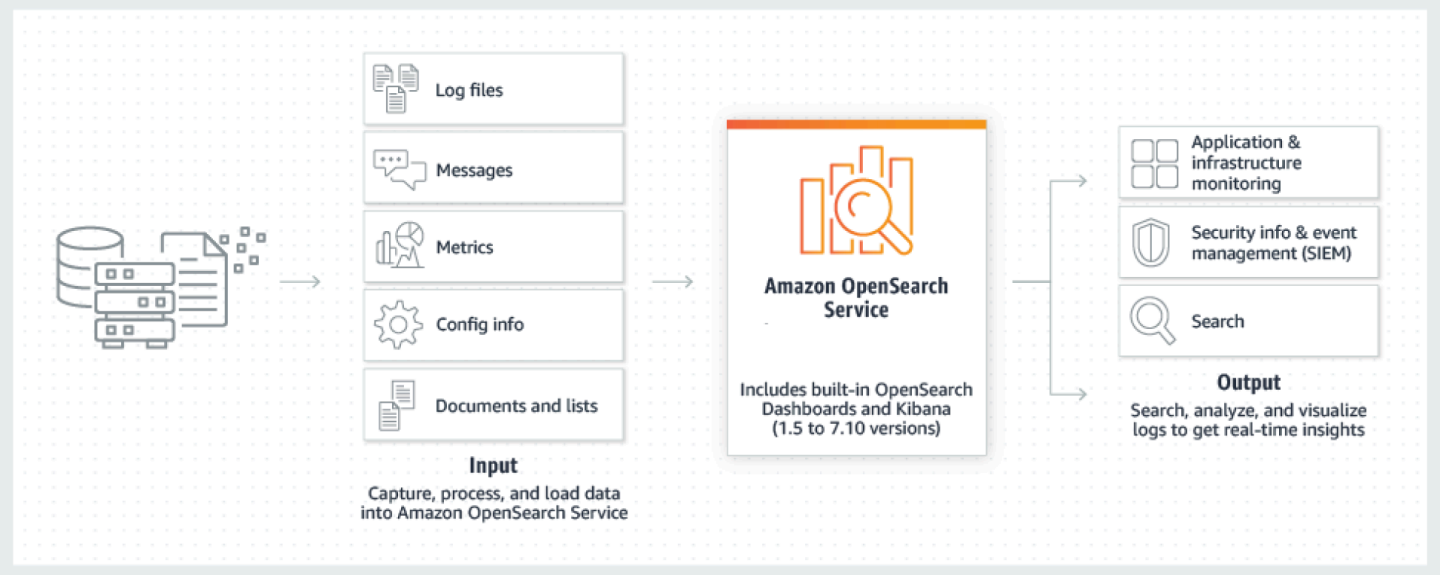
Table of Contents

什么是亚马逊 OpenSearch 服务？

Amazon OpenSearch Service 是一项托管服务，可以轻松地在 AWS 云中部署、操作和扩展 OpenSearch 集群。亚马逊 OpenSearch 服务支持 OpenSearch 传统的 Elasticsearch OSS（最高 7.10，即该软件的最终开源版本）。创建集群时，您可以选择使用哪种搜索引擎。

OpenSearch是一个完全开源的搜索和分析引擎，用于日志分析、实时应用程序监控和点击流分析等用例。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。它还可以自动检测和替换出现故障的 OpenSearch 服务节点，从而减少与自我管理基础架构相关的开销。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 OpenSearch 服务节点。

您可以使用 OpenSearch 服务控制台在几分钟内设置和配置域。如果您更喜欢编程访问，则可以使用、AWS 软件开发工具包或 Terraform。AWS CLI

Amazon OpenSearch 服务的特点

OpenSearch 服务包括以下功能：

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。
- 高达 3 PB 的附加存储空间
- 为只读数据提供经济实惠UltraWarm的冷存储

安全性

- AWS 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
- 在同一区域的两个或三个可用区（称为多可用 AWS 区）之间分配节点
- 利用专用主节点来卸载集群管理任务
- 用于备份和恢复 OpenSearch 服务域的自动快照

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 仪表板实现数据可视化
- 与 Amazon 集成， CloudWatch 用于监控 OpenSearch 服务域指标和设置警报
- 与集成 AWS CloudTrail ，用于审计配置 API 对 OpenSearch 服务域的调用
- 与亚马逊 S3、亚马逊 Kinesis 和亚马逊 DynamoDB 集成，用于将流数据加载到服务中 OpenSearch
- 数据超过特定阈值时从 Amazon SNS 发出的警报

何时使用 OpenSearch 与亚马逊 OpenSearch 服务相比

使用下表来帮助您确定预配置的 Amazon OpenSearch 服务还是自行管理 OpenSearch 是您的正确选择。

OpenSearch	亚马逊 OpenSearch 服务
• 您的组织愿意手动监控和维护自行配置的集群，并且有具备相应技能的人员。 • 你想对你的代码进行全面的编译级控制。 • 您的组织更喜欢或独一无二地使用开源软件。 • 你有一个多云战略，需要的不是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。 • 您希望能够灵活地随心所欲地使用、修改和扩展产品。 • 您希望在新功能发布后立即访问它们。	• 您不想手动管理、监控和维护您的基础架构。 • 您想要利用Amazon S3的耐用性和低成本优势，通过跨存储层对数据进行分层来管理不断增长的分析成本的简单方法。 • 你想利用与其他数据库的集成， AWS 服务 例如 DynamoDB、Amazon DocumentDB（兼容 MongoDB）、IAM 和。 CloudWatch CloudFormation • 在预防性维护和生产期间出现问题时，您需要轻松获得帮助。 AWS Support • 您想利用自我修复、主动维护、弹性和备份等功能。

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 有关更多信息，请参阅 Amazon OpenSearch 无服务器。

Amazon OpenSearch Ingestion

Amazon OpenSearch Ingestion 是一款完全托管的数据收集器，由 Data Prepper 提供支持，可向亚马逊 OpenSearch 服务域和无服务器集合提供实时日志和 OpenSearch 跟踪数据。可让您筛选、扩充、转换、标准化和聚合数据，以便进行下游分析和可视化。有关更多信息，请参阅 Amazon OpenSearch Ingesti on。

OpenSearch 和 Elasticsearch 支持的版本

OpenSearch 服务目前支持以下 OpenSearch 版本：

- 2.13、2.11、2.9、2.7、2.5、2.3、1.3、1.2、1.1、1.0

OpenSearch 服务还支持以下旧版 Elasticsearch OSS 版本：

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

有关更多信息，请参阅 the section called “支持的操作”、the section called “功能（按引擎版本）” 和 the section called “插件（按引擎版本）”。

如果您启动新的 OpenSearch 服务项目，我们强烈建议您选择支持的最新 OpenSearch 版本。如果您的现有域使用的是较旧的 Elasticsearch 版本，您可以选择保留该域或迁移您的数据。有关更多信息，请参阅 the section called “升级域”。

亚马逊 OpenSearch 服务的定价

对于 OpenSearch 服务，您需要为 EC2 实例的每小时使用量以及连接到您的实例的任何 EBS 存储卷的累积大小付费。标准 AWS 数据传输费用也适用。

但是，存在一些明显的数据传输异常。如果一个域使用多个可用区，则 OpenSearch 服务不会为可用区之间的流量计费。在分片分配和重新平衡期间，域内会发生大量数据传输。 OpenSearch 服务既不计量也不为此流量计费。同样， OpenSearch 服务不对 UltraWarm/冷节点和 Amazon S3 之间的数据传输收费。

有关全部定价详情，请参阅 Amazon OpenSearch 服务定价。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”。

亚马逊 OpenSearch 服务入门

开始之前，如果您还没有账户，请先先注册一个 AWS 账户。设置账户后，完成亚马逊 OpenSearch 服务入门教程。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的大小
- 使用域访问策略或精细访问控制控制对域的访问权限。
- 手动索引数据或来自其他 AWS 服务的索引
- 使用OpenSearch 仪表板搜索您的数据并创建可视化效果

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 “正在迁移到 OpenSearch Service”。

相关服务

OpenSearch 服务通常与以下服务一起使用：

Amazon CloudWatch

OpenSearch 服务域会自动向发送指标， CloudWatch 以便您可以监控域的运行状况和性能。有关更多信息，请参阅 使用 Amazon 监控 OpenSearch 集群指标 CloudWatch。

CloudWatch 日志也可以反其道而行之。您可以将 CloudWatch 日志配置为将数据流式传输到 OpenSearch 服务进行分析。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”。

AWS CloudTrail

AWS CloudTrail 用于获取您账户的 OpenSearch 服务配置 API 调用和相关事件的历史记录。有关更多信息，请参阅 使用 AWS CloudTrail 监控 Amazon OpenSearch Service API 调用。

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”。

AWS IAM

AWS Identity and Access Management (IAM) 是一项 Web 服务，可用于管理对 OpenSearch 服务域的访问权限。有关更多信息，请参阅 the section called “Identity and Access Management”。

AWS Lambda

AWS Lambda 是一项计算服务，允许您在不预置或管理服务器的情况下运行代码。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 OpenSearch 服务的更多信息，请参阅the section called “从 Amazon DynamoDB 表中加载流数据”。

Amazon QuickSight

您可以使用 Amazon QuickSight 控制面板可视化来自 OpenSearch 服务的数据。有关更多信息，请参阅《亚马逊 QuickSight 用户指南》 QuickSight中的在亚马逊上使用亚马逊 OpenSearch 服务。

> **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 OpenSearch 服务

主题

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

报名参加 AWS 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

   当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，应为用户分配管理访问权限，并仅使用 root 用户来执行需要 root 用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 https://aws.amazon.com/ 并选择我的账户来查看当前的账户活动并管理您的账户。

创建具有管理权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。AWS Management Console在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的以根用户身份登录。

2. 为您的根用户启用多重身份验证 (MFA)。

   有关说明，请参阅 I A M 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备（控制台）。

创建具有管理权限的用户

1. 启用 IAM Identity Center

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的启用 AWS IAM Identity Center。

2. 在 IAM 身份中心中，向用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《用户指南》 IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限。

以具有管理权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

  有关使用 IAM Identity Center 用户登录的帮助，请参阅AWS 登录 用户指南中的登录 AWS 访问门户。

为其他用户分配访问权限

1. 在 IAM Identity Center 中，创建一个遵循应用最低权限权限的最佳实践的权限集。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的创建权限集。

2. 将用户分配到群组，然后为该群组分配单点登录访问权限。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 “添加群组”。

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AWS 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：

- 中的用户和群组 AWS IAM Identity Center：

  创建权限集合。按照《AWS IAM Identity Center 用户指南》中创建权限集的说明进行操作。
- 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》中为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

如果用户想在 AWS 外部进行交互，则需要编程访问权限 AWS Management Console。授予编程访问权限的方式取决于正在访问的用户类型 AWS。

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
人力身份 （在 IAM Identity Center 中管理的用户）	使用临时证书签署向 AWS CLI、 AWS 软件开发工具包或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 • 有关的 AWS CLI，请参阅《AWS Command Line Interface 用户指南》 AWS IAM Identity Center中的 “配置 AWS CLI 要使用”。 • 有关 AWS 软件开发工具包、工具和 AWS API，请参阅《软件开发工具包和AWS 工具参考指南》中的 IAM 身份中心身份验证。

哪个用户需要编程式访问权限？	目的	方式
IAM	使用临时证书签署向 AWS CLI、 AWS 软件开发工具包或 AWS API 发出的编程请求。	按照 IAM 用户指南中的将临时证书与 AWS 资源配合使用中的说明进行操作。
IAM	（不推荐使用） 使用长期凭证签署向 AWS CLI、 AWS 软件开发工具包或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 • 有关信息 AWS CLI，请参阅用户指南中的使用 IAM 用户证书进行身份验证。AWS Command Line Interface • 有关 AWS SDK 和工具，请参阅 S AWS DK 和工具参考指南中的使用长期凭证进行身份验证。 • 有关 AWS API，请参阅 IAM 用户指南中的管理 IAM 用户的访问密钥。

安装和配置 AWS CLI

如果要使用 OpenSearch 服务 API，则必须安装最新版本的 AWS Command Line Interface (AWS CLI)。您不需要通过控制台使用 OpenSearch 服务，也可以按照中的步骤在没有 CLI 的情况下开始使用Amazon Opensearch Service 入门。 AWS CLI

要设置 AWS CLI

1. 要安装 AWS 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。 AWS CLI
2. 要配置 AWS CLI 和安全设置您的访问 AWS 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 命令提示符下输入以下命令。

```
aws opensearch help
```

AWS CLI 命令使用配置 AWS 区域 中的默认值，除非您使用参数或配置文件进行设置。要使用参数 AWS 区域 进行设置，可以将该`--region`参数添加到每个命令中。

要使用配置文件 AWS 区域 进行设置，请先在`~/.aws/config`文件或文件中添加命名的配置`%UserProfile%/.aws/config`文件（适用于 Microsoft Windows）。按 AWS 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来设置您的 AWS 区域 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

本节中大多数面向控制台的主题都从OpenSearch 服务控制台开始。如果您尚未登录您的 AWS 账户，请登录，然后打开OpenSearch 服务控制台并继续下一部分继续开始使用 OpenSearch 服务。

Amazon Opensearch Service 入门

本教程介绍如何使用 Amazon OpenSearch Service 创建和配置测试域。OpenSearch Service 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。

本教程将指导您完成快速启动并运行 OpenSearch Service 域的基本步骤。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。有关从自行托管的 OpenSearch 集群迁移到 OpenSearch Service 的信息，请参阅 the section called “正在迁移到 OpenSearch Service”。

在本教程中，您可以通过使用 OpenSearch Service 控制台、AWS CLI 或 AWS 开发工具包来完成步骤：有关安装和设置 AWS CLI 的信息，请参阅AWS Command Line Interface用户指南。

第 1 步：创建 Amazon OpenSearch Service 域。

> **Important**
>
> 这是一个简明教程，用于配置测试 Amazon Opensearch Service 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。

OpenSearch Service 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、AWS CLI 或 AWS 开发工具包创建 OpenSearch Service 域。

要使用控制台创建 OpenSearch Service 域

1. 转至 http://aws.amazon.com，然后选择 Sign In to the Console。
2. 在 Analytics 下，选择 Amazon OpenSearch Service。
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 网络下，选择 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 Service 域以便编制索引

第 2 步：将数据上传到 Amazon OpenSearch Service 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 OpenSearch Service。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 OpenSearch Service 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 OpenSearch Service 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 OpenSearch Service 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/_bulk' --data-
binary @bulk_movies.json -H 'Content-Type: application/json'
```

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

第 3 步：在 Amazon OpenSearch Service 中搜索文档

要在 Amazon Opensearch Service 域中搜索文档，请使用 Opensearch 搜索 API。也可以使用 OpenSearch 控制面板在域中搜索文档。

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 OpenSearch 结合使用而设计。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 OpenSearch Services 域中搜索文档

1. 导航到域的 OpenSearch 控制面板 URL。您可以在 OpenSearch Service 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

第 4 步：删除 Amazon OpenSearch Service 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 OpenSearch Service 域

1. 登录到 Amazon Opensearch Service 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。

后续步骤

现在您已知道如何创建域和索引数据，您可能想尝试以下的一些练习：

- 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 OpenSearch Service 的教程之一。有关更多信息，请参阅教程。

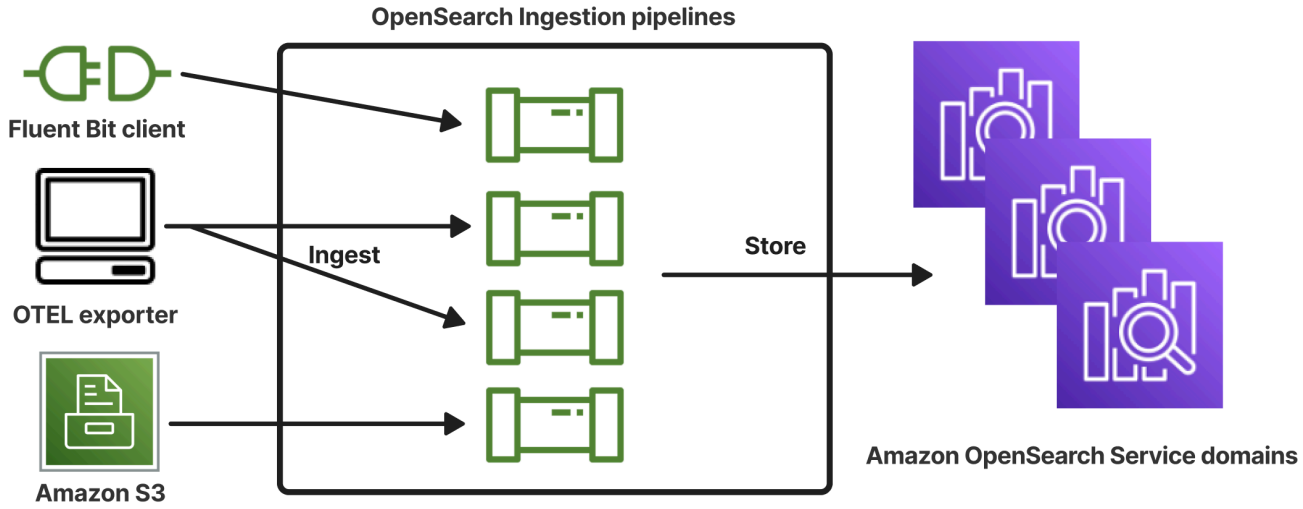
Amazon OpenSearch Ingestion

Amazon OpenSearch Ingestion 是一个完全托管的无服务器数据收集器，可向亚马逊 OpenSearch 服务域和无服务器集合提供实时日志、指标和 OpenSearch 跟踪数据。

借助 OpenSearch Ingestion，您不再需要使用 Logstash 或 Jaeger 等第三方解决方案将数据提取到您的服务域和无服务器集合中。 OpenSearch OpenSearch 您可以将数据生成器配置为向 OpenSearch Ingestion 发送数据。然后，它会自动将数据传输到您指定的域或集合。您还可以将 OpenSearch Ingestion 配置为在交付数据之前对其进行转换。

此外，借 OpenSearch 助 Ingestion，您无需担心配置服务器、管理和修补软件或扩展服务器集群。您可以直接在中配置摄取管道 AWS Management Console， OpenSearch Ingestion 负责管理和扩展它们。

OpenSearch 摄取是 Ama OpenSearch zon 服务的一部分。由开源数据收集器 Data Prepper 提供支持，该收集器可以筛选、扩充、转换、标准化和聚合数据，用于下游分析和可视化。



主题

- 重要概念
- OpenSearch 摄入的好处
- 限制
- 支持的 Data Prepper 版本
- 扩缩管道
- OpenSearch 摄取定价
- 支持的 AWS 区域

- OpenSearch 摄取配额
- 在 Amazon OpenSearch Ingestion 中设置角色和用户
- Amazon OpenSearch Ingestion 入门
- Amazon OpenSearch Ingestion 中的管道功能概述
- 创建 Amazon OpenSearch Ingestion 管道
- 查看 Amazon OpenSearch Ingestion 管道
- 更新 Amazon OpenSearch Ingestion 管道
- 停止和启动 Amazon OpenSearch Ingestion 管道
- 删除 Amazon OpenSearch Ingestion 管道
- Amazon OpenSearch Ingestion 管道支持的插件和选项
- 使用 Amazon OpenSearch Ingestion 管道集成
- 使用 Amazon OpenSearch Ingestion 在域名和集合之间迁移数据
- 使用 AWS SDK 与 Amazon OpenSearch Ingestion 进行交互
- Amazon OpenSearch Ingestion 中的安全性
- 标记 Amazon OpenSearch Ingestion 管道
- 使用 Amazon CloudWatch 记录和监控 Amazon OpenSearch Ingestion
- Amazon OpenSearch Ingestion 的最佳实践

重要概念

在开始使用 OpenSearch Ingestion 时，您可以从了解以下概念中受益：

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。基于推送的源（例如 HTTP 和 OTel 日志）将记录流式传输到提取端点。基于拉取的源（例如 OTel 跟踪和 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。不支持将自管理 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。

OpenSearch 摄入的好处

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 AWS SDK 和 OpenSearch Ingestion 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。

限制

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 AWS 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

创建管道时，使用必需的 `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。有关更多信息，请参阅 the section called “指定管道版本”。

目前， OpenSearch 采集管道配置了 2.7 版 Data Prepper。有关信息，请参阅 2.7 版本说明。有关 Data Prepper 每个版本中的功能和错误修复的信息，请参阅说明页面。 OpenSearch Ingestion 并不支持特定主要版本的每个次要版本。

更新管道的 YAML 配置文件时，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”。

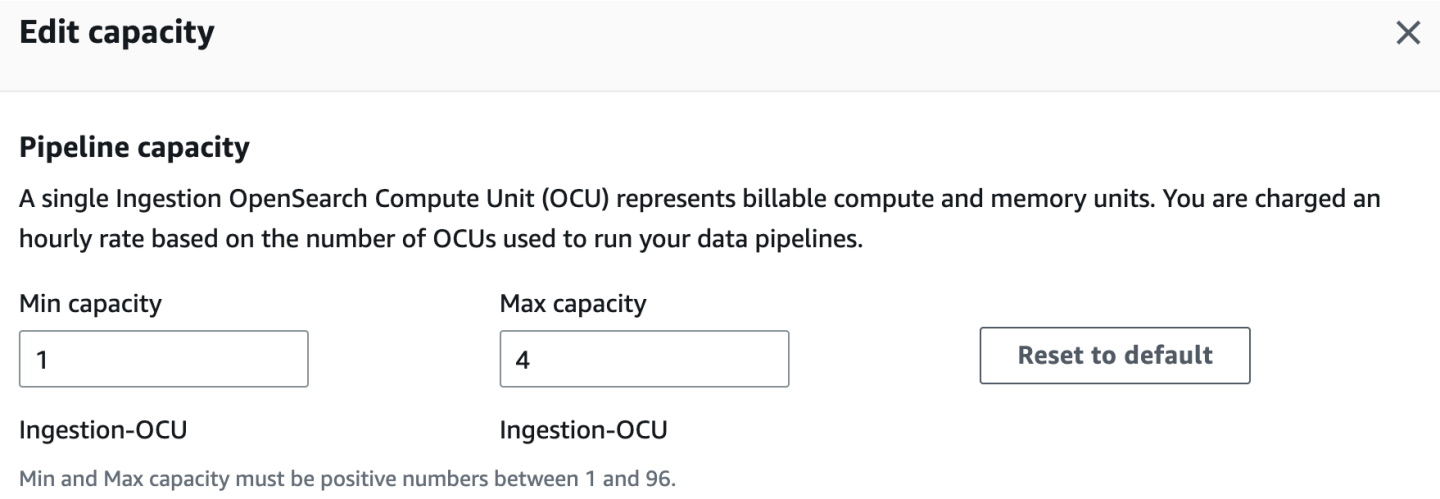
扩缩管道

您无需自己配置和管理管道容量。 OpenSearch Ingestion 会根据您指定的最小和最大摄取 OpenSearch 计算单位（摄取 OCU），根据您的估计工作负载自动扩展您的管道容量。

每个 Ingestion OCU 由大约 8 GiB 的内存和 2 个 vCPU 组成。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。

可以指定以下值：

- 最小容量 — 管道可以将容量缩减到此数量的 Ingestion OCU。指定的最小容量也是管道的起始容量。
- 最大容量 — 管道可以将容量扩展到此数量的 Ingestion OCU。



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。 OpenSearch Ingestion 会根据您的设置自动扩展您的管道的摄取 OCU 数量，以处理采集工作负载。在任何具体时间，您都只需为您管道中正在被使用的 Ingestion OCU 付费。

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 Ingestion OCU 的最低配置。

您可以指定至少 1 个 Ingestion OCU，无状态管道最多 96 个 Ingestion OCU，有状态管道最多 48 个 Ingestion OCU。对于基于推送的源，建议至少有 2 个 Ingestion OCU。启用持久缓冲功能后，您可以指定最少 2 个、最多 384 个摄入 OCU。

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 OCU 分布在可用区 (AZ) 上。AZ 数量取决于指定的最小容量。

例如，如果您指定至少 2 个计算单位，则在任何给定时间使用的 Ingestion OCU 均匀分布在 2 个可用区上。如果您指定至少 3 个或更多计算单位，则 Ingestion OCU 将平均分布在 3 个可用区上。建议您预调配至少两个 Ingestion OCU，以确保摄取管道的可用性达到 99.9%。

当管道处于 `Create failed`、`Creating`、`Deleting` 和 `Stopped` 状态时，您无需为 Ingestion OCU 付费。

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

在任何具体时间，您只需为分配给管道的 Ingestion OCU 数量付费，无论是否有数据流经管道。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 AWS 区域

OpenSearch 该 OpenSearch 服务的子集提供摄取 AWS 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AWS 一般参考。

OpenSearch 摄取配额

有关 OpenSearch 摄取资源的默认配额列表，请参阅 A mazon OpenSearch 服务配额。

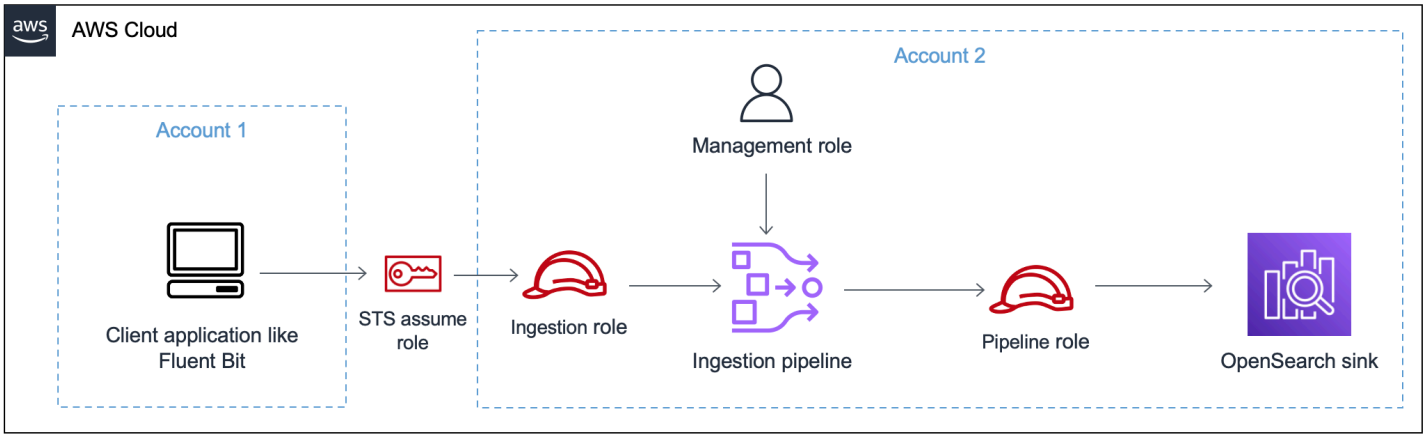
在 Amazon OpenSearch Ingestion 中设置角色和用户

Amazon OpenSearch Ingestion 使用各种权限模型和 IAM 角色，以允许源应用程序写入管道以及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。

姓名	描述
管理角色	任何管理管道的主体（通常是“管道管理员”）都需要管理访问权限，其中包括 `osis:CreatePipeline` 和 `osis:UpdatePipeline` 之类的权限。这些权限允许用户管理管道，但并不需要向管道写入数据。
管道角色	管道角色是在管道的 YAML 配置中指定的，它为管道提供写入域或集合接收器以及从基于拉取的源中读取所需的权限。有关更多信息，请参阅以下主题： • the section called “授予管道对域的访问权限” • the section called “授予管道访问集合的权限”
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

管理角色

除了创建和修改管道所需的基本 `osis:*` 权限外，您还需要管道角色资源的 `iam:PassRole` 权限。任何 AWS 服务 接受角色的人都必须使用此权限。OpenSearch Ingestion 每次需要将数据写入接收器时都会担任该角色。这有助于管理员确保仅批准的用户可配置具有能够授予权限的角色的 OpenSearch Ingestion。有关更多信息，请参阅向用户授予将角色传递给 AWS 服务 的权限。

如果您使用的是（AWS Management Console使用蓝图并稍后检查您的管道），则需要以下权限才能创建和更新管道：

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

```
            "iam:PassRole"
         ]
      }
   ]
}
```

如果您使用的是（AWS CLI不预先验证您的管道或使用蓝图），则需要以下权限才能创建和更新管道：

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

管道角色

管道需要一定的权限才能向其接收器写入。这些权限取决于接收器是 OpenSearch 服务域还是 OpenSearch 无服务器集合。

此外，管道可能需要权限才能从源应用程序（如果源应用程序是基于拉取的插件）拉取，以及写入 S3 死信队列（如果已配置）。

主题

- 写入域接收器

- 写入集合接收器
- 写入死信队列

写入域接收器

OpenSearch Ingestion 管道需要权限才能写入配置为其接收器的 OpenSearch Service 域。这些权限包括能够描述域以及向其发送 HTTP 请求。

为了向您的管道提供写入接收器所需的权限，请先创建具有所需权限的 AWS Identity and Access Management (IAM) 角色。公有和 VPC 管道需要相同的权限。然后，在域访问策略中指定管道角色，以便该域可以接受来自管道的写入请求。

最后，将角色 ARN 指定为管道配置中 sts_role_arn 选项的值：

```
version: "2"
source:
  http:
    ...
processor:
  ...
sink:
  - opensearch:
      ...
      aws:
        sts_role_arn: arn:aws:iam::{your-account-id}:role/pipeline-role
```

有关完成其中每个步骤的说明，请参阅允许管道访问域。

写入集合接收器

OpenSearch Ingestion 管道需要权限才能写入配置为其接收器的 OpenSearch 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，创建 IAM 角色，该角色具有访问所有资源的 `aoss:BatchGetCollection`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。最后，将角色 ARN 指定为管道配置中 sts_role_arn 选项的值。

有关完成其中每个步骤的说明，请参阅允许管道访问集合。

写入死信队列

如果将管道配置为写入死信队列 (DLQ)，则必须在 DLQ 配置中包含 `sts_role_arn` 选项。此角色中包含的权限允许管道访问您指定为 DLQ 事件目的地的 S3 存储桶。

您必须在所有管道组件中使用相同的 `sts_role_arn`。因此，您必须为提供 DLQ 访问权限的管道角色附加单独的权限策略。至少必须允许该角色对存储桶资源执行 `S3:PutObject` 操作：

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

然后，您可以在管道的 DLQ 配置中指定角色：

```
  ...
  sink:
    opensearch:
      dlq:
        s3:
          bucket: "my-dlq-bucket"
          key_path_prefix: "dlq-files"
          region: "us-west-2"
          sts_role_arn: "arn:aws:iam::123456789012:role/pipeline-role"
```

提取角色

OpenSearch Ingestion 目前支持的所有源插件（S3 除外）都使用基于推送的架构。这意味着源应用程序将数据推送到管道，而不是管道从源中拉取数据。

因此，您必须向源应用程序授予将数据摄取到 OpenSearch Ingestion 管道所需的权限。至少，必须向签署请求的角色授予 `osis:Ingest` 操作的权限，从而允许其向管道发送数据。公有和 VPC 管道端点需要相同的权限。

以下示例策略允许关联主体将数据摄取到名为 my-pipeline 的单个管道：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PermitsWriteAccessToPipeline",
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:us-west-2:{your-account-id}:pipeline/my-pipeline"
    }
  ]
}
```

有关更多信息，请参阅the section called “使用管道集成”。

跨账户提取

您可能需要从其他（AWS 账户例如应用程序账户）将数据摄取到管道中。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：

```
{
  "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 AssumeRole 授予管道账户中提取角色的权限。

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 AWS 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。

在管道配置中指定角色之前，必须使用相应的信任关系对其进行配置，然后在域访问策略中授予域访问权限。

主题

步骤 1：创建管道角色

您在管道配置的 sts_role_arn 参数中指定的角色必须具有允许其向域接收器发送数据的附加权限策略。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配置的 sts_role_arn 角色中为其提供写入单个域的最低权限：

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

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：

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

此外，建议您在策略中使用 aws:SourceAccount 和 aws:SourceArn 条件密钥来防止出现混淆代理人问题。源账户是管道所有者。

例如，您可以将以下条件块添加到策略：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "{your-account-id}"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:osis:{region}:{your-account-id}:pipeline/*"
    }
}
```

步骤 2：在域访问策略中添加管道角色

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 ingestion-domain 的域写入数据：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
```

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

步骤 3：映射管道角色（仅适用于使用精细访问控制的域）

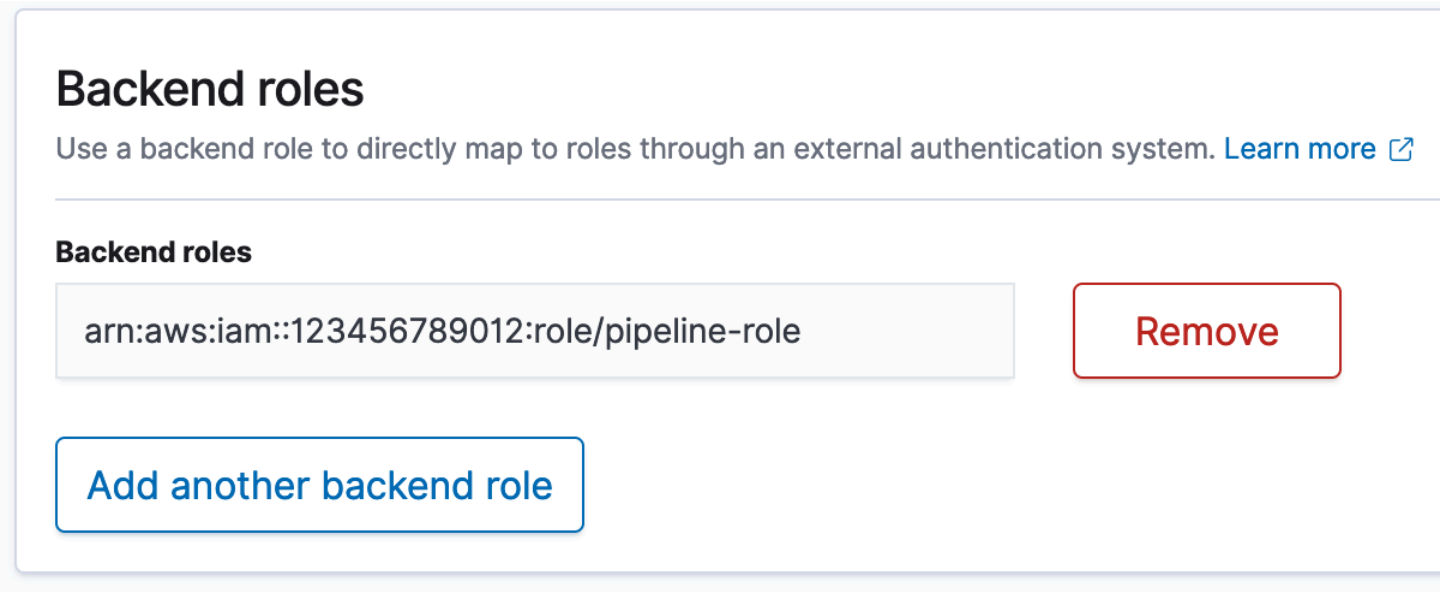
如果您的域使用精细访问控制进行身份验证，则需要采取额外步骤，为管道提供域访问权限。步骤因域配置而异：

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，并且它与管道角色 (`sts_role_arn`) 不同，则需要将管道角色映射到 OpenSearch`all_access`后端角色。实际上是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。相反，你需要将管道角色 (`sts_role_arn`) 映射到 OpenSearch`all_access`后端角色。实际上是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：主角色与管道角色相同（不常见）– 如果您使用 IAM ARN 作为主用户，且主用户与作为管道角色的 ARN 相同 (`sts_role_arn`)，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。这种场景并不常见，因为绝大多数环境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：



步骤 4：在管道配置中指定角色

为成功创建管道，必须在管道配置中将您在步骤 1 中创建的管道角色指定为 sts_role_arn 参数。管道扮演此角色是为了签署对 OpenSearch 服务域接收器的请求。

在 `sts_role_arn` 字段中，指定 IAM 管道角色 ARN：

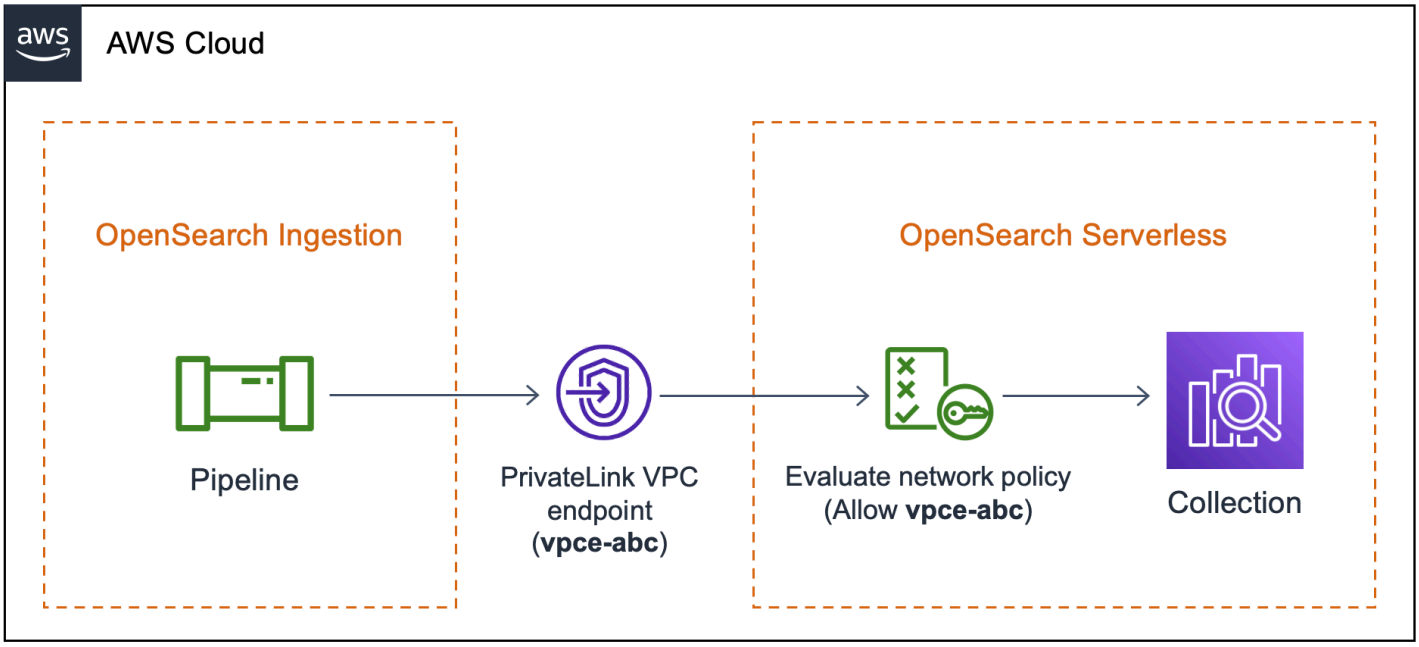
```
version: "2"
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log: [ "%{COMMONAPACHELOG}" ]
  sink:
    - opensearch:
        hosts: [ "https://search-{domain-name}.us-east-1.es.amazonaws.com" ]
        index: "my-index"
        aws:
          region: "{region}"
          sts_role_arn: "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
```

有关必要参数和不支持的参数的完整参考，请参阅 the section called “支持的插件和选项”。

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 AWS 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。在管道配置中指定角色之前，必须为其配置适当的信任关系，然后通过数据访问策略向其授予数据访问权限。

在创建管道期间， OpenSearch Ingestion 会在管道和 OpenSearch Serverless 集合之间创建 AWS PrivateLink 连接。来自管道的所有流量都经过此 VPC 终端节点并路由到集合。为了访问集合，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

限制

以下限制适用于写入 OpenSearch 无服务器集合的管道：

- oTel 跟踪组处理器目前不适用于 OpenSearch 无服务器集合接收器。
- 目前，OpenSearch Ingestion 仅支持旧版`_template`操作，而 OpenSearch Serverless 支持可组合操作。`_index_template`因此，如果管道配置包含 `index_type` 选项，则必须将其设置为 `management_disabled`。

提供对管道的网络访问

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馆藏，或者是否必须以私密方式访问该馆藏。有关网络策略的更多信息，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，在创建具有 OpenSearch Serverless 集合接收器的管道时，必须使用network_policy_name选项提供关联网络策略的名称。

例如：

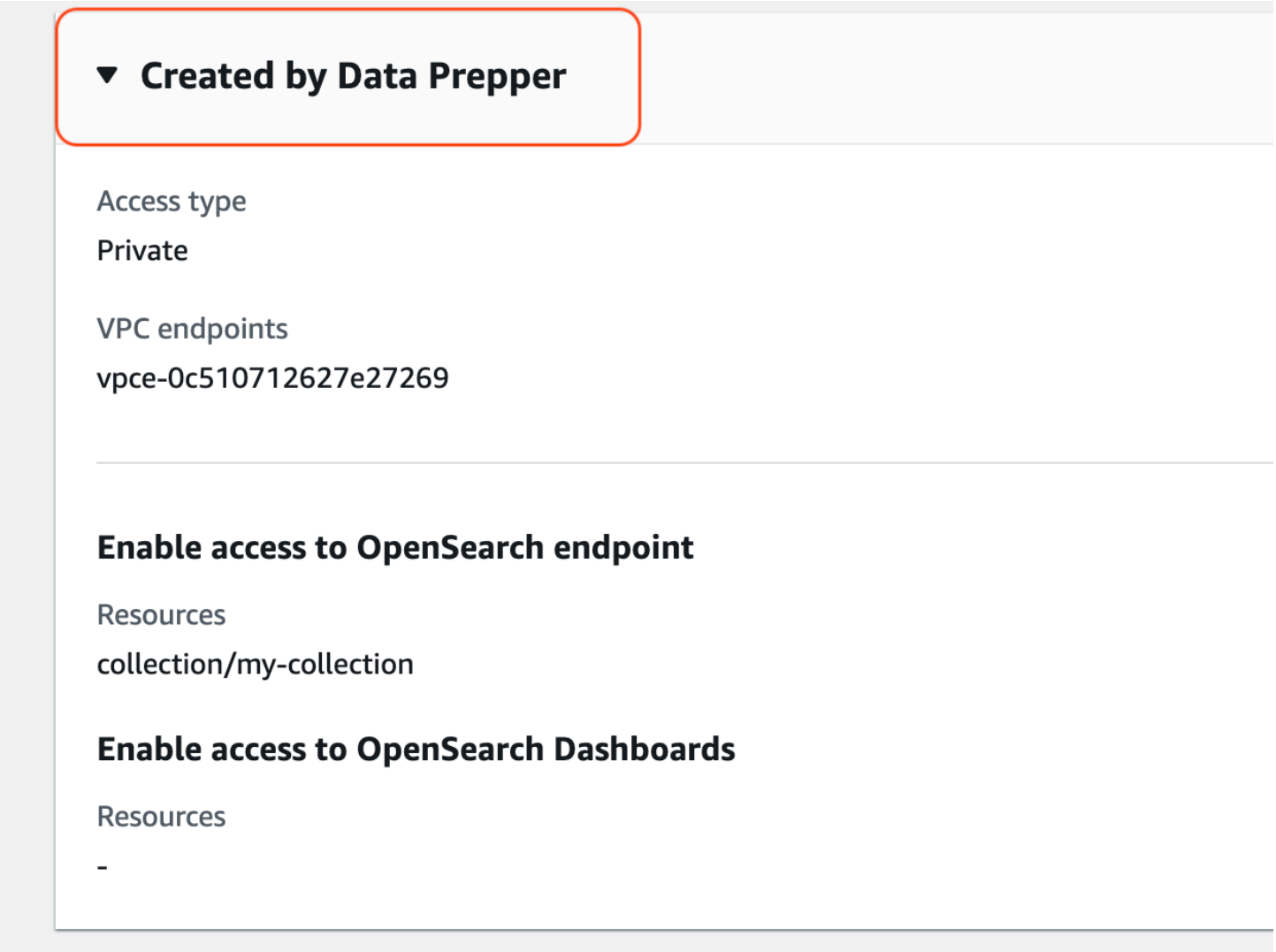
```
...
sink:
   - opensearch:
       hosts: [ "https://{collection-id}.{region}.aoss.amazonaws.com" ]
       index: "my-index"
        aws:
          serverless: true
          serverless_options:
            network_policy_name: "{network-policy-name}"
```

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。该规则授予对连接管道和集合的 VPC 终端节点的访问权限。

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，为集合指定公共访问权限的规则会优先于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。它还会修改网络策略，将 VPC 终端节点从允许的终端节点列表中删除。如果您重启管道，它会重新创建 VPC 终端节点，并使用终端节点 ID 重新更新网络策略。

步骤 1：创建管道角色

您在管道配置的 sts_role_arn 参数中指定的角色必须具有允许其向集合接收器发送数据的附加权限策略。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配置的 sts_role_arn 角色中为其提供写入集合的最低权限：

```
{
    "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：

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

步骤 2：创建集合

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

数据访问政策

为集合创建数据访问策略，向管道角色授予所需权限。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，指定您在上一步中创建的管道角色的 Amazon 资源名称 (ARN)。

网络访问策略

为馆藏创建网络访问策略。您可以将数据提取到公共集合或 VPC 集合中。例如，以下策略提供对单个 OpenSearch 无服务器托管的 VPC 终端节点的访问权限：

```
[
   {
      "Description":"Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/{collection-name}"
            ]
         }
      ],
      "AllowFromPublic": false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ]
   }
]
```

> **Important**
>
> 您必须在管道配置的network_policy_name选项中指定网络策略的名称。在创建管道时，OpenSearch Ingestion 会更新此网络策略以允许访问它在管道和集合之间自动创建的 VPC 终端节点。有关工作流配置示例，请参阅步骤 3。有关更多信息，请参阅 the section called “提供对管道的网络访问”。

步骤 3：创建管道

最后，创建一个管道，在其中指定管道角色和集合详细信息。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。

务必执行以下操作：

- 对于 `hosts` 选项，指定您在步骤 2 中创建的集合的端点。
- 对于 `sts_role_arn` 选项，指定您在步骤 1 中创建的管道角色的 Amazon 资源名称 (ARN)。
- 将 `serverless` 选项设置为 `true`。
- 将该network_policy_name选项设置为附加到集合的网络策略的名称。 OpenSearch Ingestion 会自动更新此网络策略，以允许从其在管道和集合之间创建的 VPC 进行访问。有关更多信息，请参阅 the section called “提供对管道的网络访问”。

```
version: "2"
log-pipeline:
  source:
    http:
        path: "/log/ingest"
  processor:
    - date:
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://{collection-id}.{region}.aoss.amazonaws.com" ]
        index: "my-index"
        aws:
          serverless: true
          serverless_options:
            network_policy_name: "{network-policy-name}" # If the policy doesn't exist,
 a new policy is created.
          region: "us-east-1"
          sts_role_arn: "arn:aws:iam::{account-id}:role/{pipeline-role}"
```

有关必要参数和不支持的参数的完整参考，请参阅 the section called “支持的插件和选项”。

Amazon OpenSearch Ingestion 入门

Amazon OpenSearch Ingestion 支持将数据摄取到托管 OpenSearch Service 域和 OpenSearch 无服务器集合中。以下教程将指导您完成每个用例启动和运行管道（每个用例）的基本步骤。

> **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。有关更多详细信息，请参阅 the section called “创建管道”。

在本教程中，您将完成以下步骤：

1. 创建管道角色。
2. 创建域。
3. 创建管道。
4. 摄取一些样本数据。

在本教程中，您将创建以下资源：

- 名为 `ingestion-pipeline` 的管道
- 管道将写入的名为 `ingestion-domain` 的域
- 管道将担任一个名为 `PipelineRole` 的 IAM 角色，以便写入域

所需的权限

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。这些权限允许您创建管道角色 (`iam:Create`)、创建或修改域 (`es:*`) 以及使用管道 (`osis:*`)。

此外，还需要对管道角色资源拥有 `iam:PassRole` 权限。此权限允许您将管道角色传递给 OpenSearch Ingestion，以便它可以向域中写入数据。

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

```
            "osis:*",
            "iam:Create*",
            "es:*"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/PipelineRole"
         ],
         "Effect":"Allow",
         "Action":[
            "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 AWS 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。

```
{
    "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**
>
> 为简单起见，本教程使用较为宽泛的访问策略。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 “授予管道对域的访问权限”。

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：

```
{
   "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

接下来，创建一个名为 ingestion-domain 的域，以向其中摄取数据。

导航到亚马逊 OpenSearch 服务控制台 https://console.aws.amazon.com/aos/home 并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，您可以配置具有 VPC 访问权限的域和/或使用精细访问控制。要使用精细的访问控制，请参阅映射管道角色。

域必须拥有授予 PipelineRole 访问权限的访问策略，这是在上一步中创建的。管道将扮演此角色（在管道配置中名为 sts_role_arn），以便将数据发送到服务域接收器。 OpenSearch

确保该域具有以下域-级别访问策略，该策略授予对该域的 PipelineRole 访问权限。替换成您自己的区域和账户 ID：

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

有关创建域-级别访问策略的更多信息，请参阅基于资源的访问策略。

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

> **Note**
>
> 记住域端点（例如 `https://search-ingestion-domain.us-east-1.es.amazonaws.com`）。在下一步中，您将使用它来配置管道。

步骤 3：创建管道

现在，您已经拥有了具有相应访问权限的域和角色，可以创建管道了。

要创建管道

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 创建管道。
3. 将管道命名为 ingestion-pipeline，并将容量设置保留为默认值。
4. 在本教程中，您将创建一个使用 Http 源插件的名为 `log-pipeline` 的简单子管道。此插件接受 JSON 数组格式的日志数据。您将指定单个 OpenSearch 服务域作为接收器，并将所有数据提取到`application_logs`索引中。

   在管道配置下，将以下 YAML 配置粘贴到编辑器中：

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/test_ingestion_path"
  processor:
    - date:
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://search-ingestion-domain.us-east-1.es.amazonaws.com" ]
        index: "application_logs"
        aws:
          sts_role_arn: "arn:aws:iam::{your-account-id}:role/PipelineRole"
          region: "us-east-1"
```

> **Note**
>
> 该 `path` 选项指定用于提取的 URI 路径。对于基于拉取的源而言，此为必需选项。有关更多信息，请参阅 the section called “指定提取路径”。

5. 将 `hosts` URL 替换为您在上一节中创建（或修改）的域的端点。将 `sts_role_arn` 参数替换为 `PipelineRole` 的 ARN。
6. 选择验证管道并确保验证成功。
7. 为简单起见，本教程为管道配置公有访问权限。在 网络下，选择 公有访问权限。

   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 访问权限”。
8. 保持日志发布为启用状态，以防您在完成本教程时遇到任何问题。有关更多信息，请参阅 the section called “监控管道日志”。

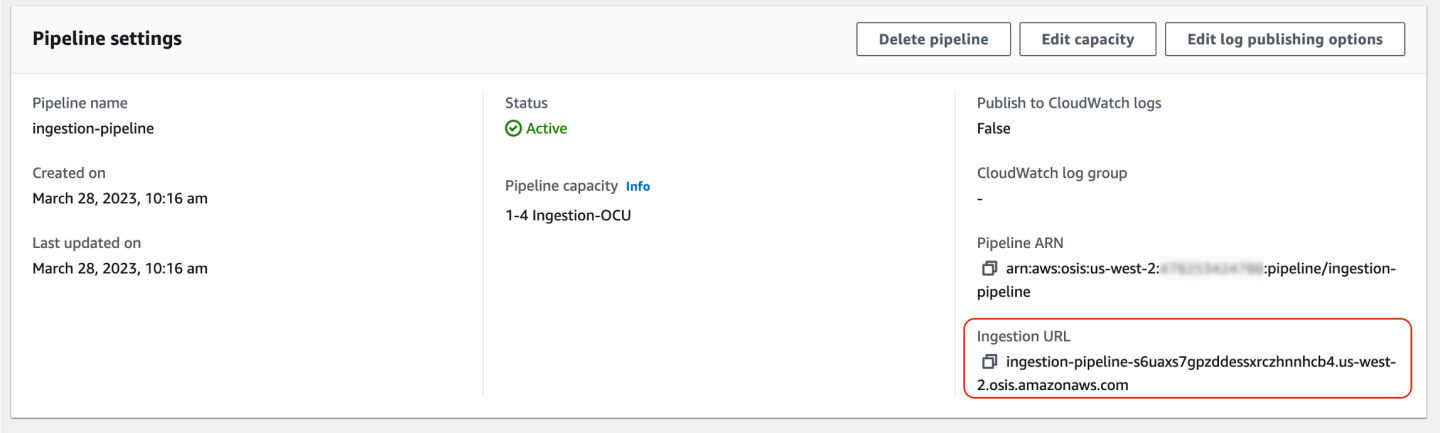
   指定以下日志组名称：`/aws/vendedlogs/OpenSearchIngestion/ingestion-pipeline/audit-logs`
9. 选择下一步。检查您的管道配置，然后选择创建管道。管道需要 5-10 分钟才能变为活动状态。

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 application_logs 索引发送单个日志文件：

```
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/log-pipeline/
test_ingestion_path
```

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 AWS 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”

在本教程中，您将完成以下步骤：

1. 创建管道角色。
2. 创建集合。
3. 创建管道。
4. 摄取一些样本数据。

在本教程中，您将创建以下资源：

- 名为 `ingestion-pipeline-serverless` 的管道
- 管道将写入的名为 `ingestion-collection` 的集合
- 名为 `PipelineRole` 的 IAM 角色，管道将担任此角色，对集合执行写入操作

所需的权限

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。这些权限允许您创建管道角色 (`iam:Create*`)、创建或修改集合 (`aoss:*`) 以及使用管道 (`osis:*`)。

此外，还需要对管道角色资源拥有 `iam:PassRole` 权限。此权限允许您将管道角色传递给 OpenSearch Ingestion，以便它可以将数据写入集合。

```
{
```

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

步骤 1：创建管道角色

首先，创建一个管道将扮演的角色，以便访问 OpenSearch 无服务器集合接收器。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AWS 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. 选择 JSON，然后将下面的策略粘贴到编辑器中。

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

```
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:aoss:{region}:{your-account-id}:collection/{collection-
id}"
    },
    {
      "Action": [
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:UpdateSecurityPolicy"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aoss:collection": "{collection-name}"
        }
      }
    }
  ]
}
```

4. 选择 “下一步”，选择 “下一步”，然后命名您的策略collection-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：

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

8. 请选择 Next（下一步）。然后搜索并选择 collection-pipeline-policy（您刚刚创建的）。
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：创建集合

接下来，创建集合，将数据摄取到其中。将集合命名为 `ingestion-collection`。

1. 导航到亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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. 在网络访问设置下，将访问类型更改为公有。
5. 所有其他设置保留为默认值，然后选择 Next（下一步）。
6. 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/PipelineRole",
      "arn:aws:iam::{your-account-id}:role/Admin"
```

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

7. 替换 Principal 元素。第一个主体应指定您创建的管道角色。第二个主体应指定用户或角色，供您稍后用于查询集合。
8. 请选择 Next（下一步）。命名访问策略，pipeline-domain-access然后再次选择 Next。
9. 查看集合配置并选择 Submit（提交）。

当集合处于活动状态时，请记下 End point 下的 OpenSearch 端点（例如https://*{collection-id}*.us-east-1.aoss.amazonaws.com）。您创建管道时需要用到。

步骤 3：创建管道

现在，您已经拥有集合和具有相应访问权限的角色，可以创建管道了。

要创建管道

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. 选择创建管道。
3. 将管道命名为 serverless-ingestion，并将容量设置保留为默认值。
4. 在本教程中，我们将创建名为 log-pipeline 且使用 HTTP 源插件的简单子管道。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

   在管道配置下，将以下 YAML 配置粘贴到编辑器中：

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/test_ingestion_path"
  processor:
    - date:
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://{collection-id}.us-east-1.aoss.amazonaws.com" ]
```

```
        index: "my_logs"
        aws:
          sts_role_arn: "arn:aws:iam::{your-account-id}:role/PipelineRole"
          region: "us-east-1"
          serverless: true
```

5. 将 hosts URL 替换为您在上一节中创建的集合端点。将 sts_role_arn 参数替换为 PipelineRole 的 ARN。或者，修改 region。
6. 选择验证管道并确保验证成功。
7. 在本教程中，为简单起见，我们将配置管道的公共访问权限。在 网络下，选择 公有访问权限。

   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 访问权限”。
8. 保持日志发布为启用状态，以防您在完成本教程时遇到任何问题。有关更多信息，请参阅 the section called “监控管道日志”。

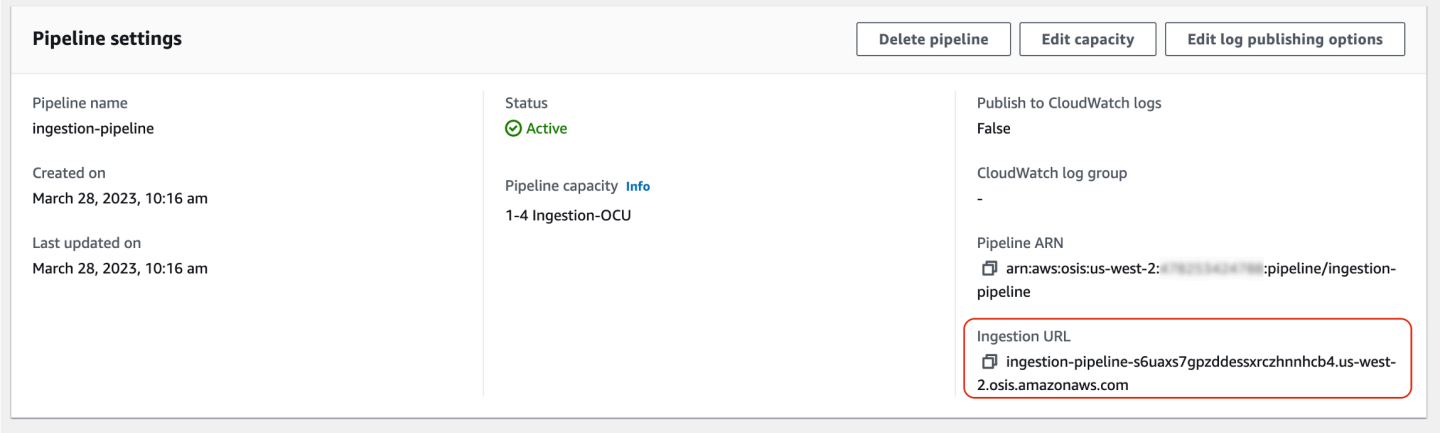
   指定以下日志组名称：/aws/vendedlogs/OpenSearchIngestion/serverless-ingestion/audit-logs
9. 请选择 Next（下一步）。检查您的管道配置，然后选择创建管道。管道需要 5-10 分钟才能变为活动状态。

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，摄取一些示例数据。以下示例请求使用 awscurl 向 my_logs 索引发送单个日志文件：

```
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/log-pipeline/
test_ingestion_path
```

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

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，以增加数据的持久性。您可以使用持久缓冲来为所有支持的基于推送的源提取数据，而无需设置独立的缓冲区。这些包括 HTTP 以及日志、跟踪和指标的 OpenTelemetry 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 会根据您为管道指定的摄取 OpenSearch 计算单位（摄取 OCU）自动确定所需的缓冲容量。

默认情况下，管道使用加密 AWS 拥有的密钥 缓冲区数据。这些管道不需要管道角色的任何额外权限。或者，您可以指定客户托管密钥并将以下 IAM 权限添加到管道角色：

```
{
    "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《AWS Key Management Service 开发人员指南》中的客户托管密钥。

> **Note**
>
> 如果您禁用持久缓冲功能，则您的管道将更新为完全在内存缓冲上运行。

调整最大请求负载大小

如果您为管道启用永久缓冲，则最大请求负载大小默认为 1 MB。默认值可提供最佳性能。但是，如果您的客户端发送的请求超过 1 MB，则可以增加此值。要调整最大有效载荷大小，请在源配置中设置该max_request_length选项。就像永久缓冲一样，此选项仅支持 HTTP 以及日志、跟踪和指标的 OpenTelemetry 来源。

该max_request_length选项的唯一有效值是 1mb、1.5mb、2mb、2.5mb、3mb、3.5mb 和 4mb。如果指定不同的值，则会收到错误消息。

以下示例演示了如何在管道配置中配置最大负载大小：

```
...
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
      max_request_length: 4mb
  processor:
...
```

如果您没有为管道启用永久缓冲，则所有源的max_request_length选项值默认为 10 MB，并且无法修改。

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 (DLQ) 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：

- `max_retries`用于 OpenSearch 水槽的东西已经用完了。 OpenSearch 对于此选项，摄取至少需要 16。
- 由于出现错误条件，事件被接收器拒绝。

配置

要为子管道配置死信队列，请在 `opensearch` 接收器配置中指定 `dlq` 选项：

```
apache-log-pipeline:
  ...
  sink:
    opensearch:
      dlq:
        s3:
          bucket: "my-dlq-bucket"
          key_path_prefix: "dlq-files"
          region: "us-west-2"
          sts_role_arn: "arn:aws:iam::123456789012:role/dlq-role"
```

写入此 S3 DLQ 的文件将采用以下命名模式：

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

有关更多信息，请参阅死信队列 (DLQ)。

有关配置 `sts_role_arn` 角色的说明，请参阅the section called “写入死信队列”。

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`。

生成文档 ID

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

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_field: "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。

`create` 操作可确保不会覆盖具有相同 ID 的文档。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_field: "my_doc_id_field"
```

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_field: 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"
   }
}
```

生成路由 ID

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_field: 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。

E nd-to-end 致谢

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。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，请在源配置中包含以下`acknowledgments`选项：

```
s3-pipeline:
  source:
    s3:
      acknowledgments: true
...
```

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 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

先决条件和所需角色

要创建 OpenSearch 摄取管道，您必须拥有以下资源：

- 一个 IAM 角色， OpenSearch Ingestion 为了写入接收器而将担任该角色。您将在您的管道配置中包含此角色 ARN。

- 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 “授予管道对域的访问权限”
- the section called “授予管道访问集合的权限”

> **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 “步骤 3：映射管道角色（仅适用于使用精细访问控制的域）”。

所需权限

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: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（您在管道配置中为 `sts_role_arn` 选项指定的 ARN）上，或者如果您计划在每个管道中使用不同的角色，则仅 `*`。

例如，以下策略授予创建管道的权限：

```
{
   "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: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

配置管道时，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:
    ...
```

当您选择 “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”。

每当你对启动蓝/绿部署的管道进行更改时， OpenSearch Ingestion 都可以将其升级到当前在管道 YAML 文件中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 o Tel 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，假设您为名为 `logs` 的提取管道指定了以下条目子管道：

```
entry-pipeline:
  source:
    http:
      path: "/my/test_path"
```

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://logs-abcdefgh.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 `path: "/${pipelineName}/test_path"`），则关联子管道的名称将替代变量。在本例中，它为 `https://logs.us-west-2.osis.amazonaws.com/entry-pipeline/test_path`。

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。 AWS CLI

控制台

要创建管道

1. 登录亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
2. 在左侧导航窗格中选择管道，然后选择创建管道。
3. 输入管道的名称。
4. （可选）选择启用持久缓冲功能。持久缓冲功能将您的数据存储在跨多个可用区的、基于磁盘的缓冲区中。更多信息，请参阅持久缓冲功能。如果启用永久缓冲区，请选择加密缓冲区数据的 AWS Key Management Service 密钥。
5. 在摄取 OpenSearch 计算单位 (OCU) 中配置最小和最大管道容量。有关更多信息，请参阅 the section called “扩缩管道”。
6. 在管道配置下，以 YAML 格式提供您的管道配置。单个管道配置文件可以包含 1-10 个子管道。每个子管道都由一个来源、零个或多个处理器以及一个接收器组成。对于 OpenSearch 摄取，接收器必须始终是 OpenSearch 服务域。有关支持的选项列表，请参阅 the section called “支持的插件和选项”。

   > **Note**
   >
   > 您必须在每个子管道中包含 `sts_role_arn` 和 `sigv4` 选项。管道扮演中定义的角色`sts_role_arn`来签署对域的请求。有关更多信息，请参阅 the section called “授予管道对域的访问权限”。

   以下示例配置文件使用 HTTP 源和 Grok 插件来处理非结构化日志数据并将其发送到 OpenSearch 服务域。子管道命名为 `log-pipeline`。

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/log/ingest"
  processor:
    - grok:
        match:
          log: [ '%{COMMONAPACHELOG}' ]
    - date:
        from_time_received: true
```

```
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://search-my-domain.us-east-1.es.amazonaws.com" ]
        index: "apache_logs"
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/{pipeline-role}"
          region: "us-east-1"
```

> **Note**
>
> 如果您在 YAML 管道定义中指定多个接收器，则它们必须全部为同一个 OpenSearch 服务域。一个 OpenSearch 摄取管道无法写入多个不同的域。

您可以构建自己的管道配置，也可以选择上传文件并导入自行管理的 Data Prepper 管道的现有配置。此外，您也可以使用配置蓝图。

7. 配置管道后，选择验证管道以确认您的配置正确无误。如果验证失败，请修复错误并重新运行验证。
8. 在网络配置下，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 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 管理的终端节点。
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 AWS 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 HTTP）到达接 OpenSearch 入管道。
VPC 连接选项	如果您的源是自行管理的终端节点，请将您的管道连接到 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 访问权限”。

9. （可选）在标签下，将一个或多个标签（键值对）添加到您的管道。有关更多信息，请参阅 the section called “标记管道”。
10. （可选）在“日志发布选项”下，开启向 Amazon Logs 发布管道 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”。
11. 选择 下一步。
12. 检查您的管道配置，然后选择创建。

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。

AWS CLI

create-pipeline 命令接受以字符串形式或在 .yaml 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 个 Ingestion OCU，最多 10 个 Ingestion OCU
- 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"
```

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 “使用管道集成”。

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 AWS 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

使用蓝图创建管道

不要从头开始创建管道定义，而是使用配置蓝图，这些蓝图是针对常见提取场景（例如 Trace Analytics 或 Apache 日志）的预调配 YAML 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. 登录亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
2. 在左侧导航窗格中选择管道，然后选择创建管道。
3. 选择一个蓝图。管道配置会填充您所选用例的子管道。
4. 查看注释的文本，该文本将指导您完成蓝图的配置。

> **Important**
>
> 初始的管道蓝图是无效的。您需要进行一些修改，例如提供 AWS 区域 和角色 ARN 以用于身份验证，否则管道验证将失败。

CLI

要使用获取所有可用蓝图的列表 AWS 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 a Role ARN with access to the domain. This role should have
 a trust relationship with osis-pipelines.amazonaws.com\n          # sts_role_arn:
 \"arn:aws:iam::123456789012:role/Example-Role\"\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            #
 sts_role_arn: \"arn:aws:iam::123456789012:role/Example-Role\"\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 AWS Management Console、AWS CLI 或 OpenSearch Ingestion API，查看有关 Amazon OpenSearch Ingestion 管道的详细信息。

控制台

查看管道

1. 登录到位于 https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 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：

   - `Creating`— 正在创建管道。
   - `Active`— 管道处于活动状态，并且可以摄取数据。
   - `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。
   - `Starting`— 管道正在启动。
   - `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。

当管道处于 `Create failed`、`Creating`、`Deleting` 和 `Stopped` 状态时，您无需为 Ingestion OCU 付费。

CLI

要使用 AWS CLI 查看管道，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 Ingestion 管道，请调用 ListPipelines 和 GetPipeline 操作。

更新 Amazon OpenSearch Ingestion 管道

您可以使用 AWS Management Console、或 OpenSearch Ingestion API 更新 Amazon OpenSearch Ingestion 管道。 AWS CLI OpenSearch 当您更新管道的 YAML 配置时，Ingestion 会启动蓝/绿部署。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：

- 您可以编辑管道的容量限制、日志发布选项和 YAML 配置。无法编辑其名称或网络设置。
- 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。
- 更新管道的 YAML 配置时， OpenSearch Ingestion 会启动蓝/绿部署。有关更多信息，请参阅 the section called “使用蓝绿部署进行管道更新”。
- 更新管道的 YAML 配置时， OpenSearch Ingestion 会自动将您的管道升级到工作流配置中指定的 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。只有在更新管道 YAML 配置时才需要此权限，而修改日志发布或容量限制等其他设置则不需要此权限。

例如，以下策略授予更新管道的权限：

```
{
   "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 AWS Management Console、或 OpenSearch Ingestion API 更新 Amazon OpenSearch Ingestion 管道。 AWS CLI

控制台

更新管道

1. 登录亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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。您可以编辑管道的容量限制、日志发布选项和 YAML 配置。无法编辑其名称或网络设置。
4. 完成更改后，选择 Save (保存)。

CLI

要使用更新管道 AWS 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 当您更新管道的 YAML 配置时，Ingestion 会启动蓝/绿部署流程。

蓝绿部署是指创建用于管道更新的新环境并在这些更新完成后将流量路由至新环境的实践。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。蓝绿部署本身不会对性能产生任何影响；但是，如果管道配置以某种方式改变了性能，则性能可能会发生变化。

OpenSearch 在蓝/绿部署期间，Ingestion 会阻止自动缩放。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 YAML 配置文件时， OpenSearch Ingestion 可以自动将您的管道升级到工作流配置中指定的 Data Prepper 主版本支持的最新次要版本。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。

停止和启动 Amazon OpenSearch Ingestion 管道

停止和启动 Amazon OpenSearch Ingestion 管道可以帮助您控制开发和测试环境的成本。您可以暂时停止管道，而不是每次使用管道时设置和停用管道。

主题

- 停止和启动 OpenSearch Ingestion 管道概述
- 停止 OpenSearch Ingestion 管道
- 启动 OpenSearch Ingestion 管道

停止和启动 OpenSearch Ingestion 管道概述

如果无需向管道中提取数据，您可以在相应时段停止管道。您可以在需要使用时再次启动管道。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。

如果您需要将管道保持运行状态，但具有的容量超过所需的容量，请不要使用启动和停止。如果您的管道成本太高或不太繁忙，请考虑降低其最大容量限制。有关更多信息，请参阅the section called “扩缩管道”。

停止 OpenSearch Ingestion 管道

要使用 OpenSearch Ingestion 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. 登录到位于 https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 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。

AWS CLI

要使用 AWS CLI 停止管道，请调用带有以下参数的 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 OpenSearch Ingestion 管道

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. 登录到位于 https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 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。

AWS CLI

要使用 AWS CLI 启动管道，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 AWS Management Console、AWS CLI 或 OpenSearch Ingestion API 删除 Amazon OpenSearch Ingestion 管道。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. 登录到位于 https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 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，然后选择删除。
4. 确认删除并选择删除。

CLI

要使用 AWS CLI 删除管道，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删除管道操作：

- `PipelineName` – 管道的名称。

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

与开源 Data Prepper 相比，Amazon OpenSearch Ingestion 支持源代码、处理器和接收器的子集。此外，OpenSearch Ingestion 对每个支持的插件的可用选项施加了一些限制。以下各节介绍了 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：

- Amazon DocumentDB
- DynamoDB
- OpenSearch

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

处理器：

- 聚合
- 异常探测器
- CSV
- 日期
- 解压缩
- 剖析
- 删除事件
- 地理知识产权
- Grok
- 键值
- 地图到清单

- 变异事件（处理器系列）
- 变异字符串（处理器系列）
- 混淆处理
- OTel 指标
- OTel 跟踪组
- OTel 跟踪
- 解析图标
- 解析 JSON
- 解析 XML
- 选择条目
- 服务映射
- 跟踪对等转发服务器
- 截断
- 用户代理

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

无状态处理器执行诸如转换和筛选之类的操作，而有状态处理器则执行诸如聚合之类的操作，这些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 Ingestion OCU。如果管道包含任何有状态的处理器，则最大容量限制为 48 个 Ingestion OCU。但是，如果管道启用了永久缓冲，则它最多可以有 384 个仅包含无状态处理器的摄入 OCU，如果它包含任何有状态的处理器，则最多可以有 192 个摄入 OCU。有关更多信息，请参阅 the section called “扩缩管道”。

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

配置要求和限制

除非下面另有说明，否则上面列出的受支持插件的 Data Prepper 配置参考中描述的所有选项都允许在 OpenSearch Ingestion 管道中使用。以下各节解释了 OpenSearch Ingestion 对某些插件选项施加的限制。

> **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，有些选项无法传递给 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 水槽
- 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 水槽

s OpenSearchink 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。YAML 定义文件中的所有主机必须指向同一端点。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
- bulk_size

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 插件中的serverless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。

使用 Amazon OpenSearch Ingestion 管道集成

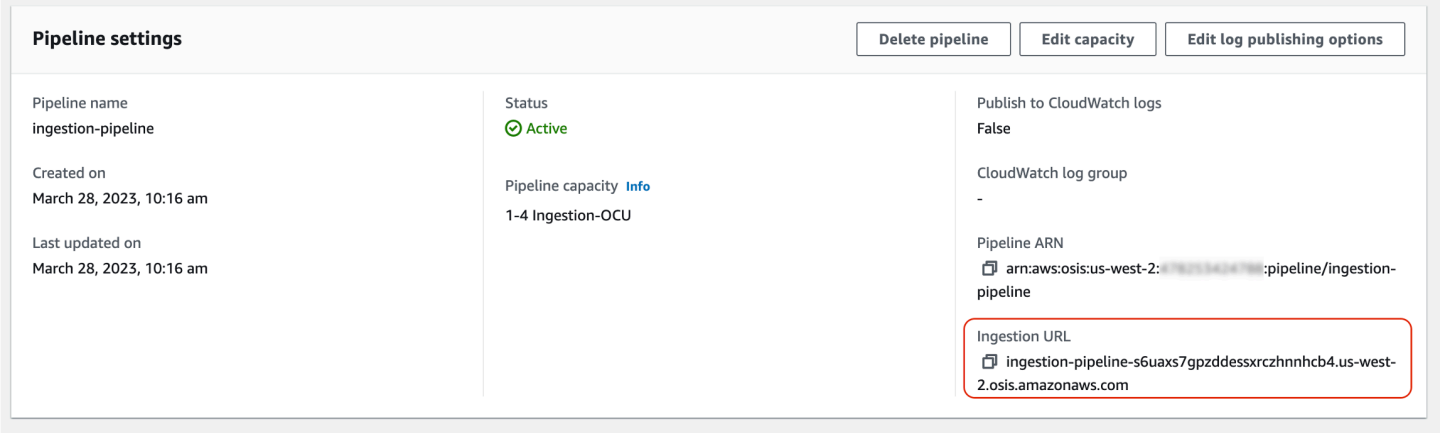
为了成功将数据提取到 Amazon OpenSearch Ingestion 管道，您必须将您的客户端应用程序（源）配置为将数据发送到管道终端节点。您的来源可能是客户端，例如 Fluent Bit 日志、 OpenTelemetry收集器或简单的 S3 存储桶。每个客户端的确切配置各不相同。

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 AWS 服务名称 (`osis`) 和主机端点，它们必须是管道端点。

主题

构建摄取端点

为将数据摄取到管道，请将其发送到摄取端点。要查找摄取 URL，请导航到管道设置页面并复制摄取 URL：



要为基于拉取的源（例如，OTel 跟踪和 OTel 指标）构建完整摄取端点，请将管道配置中的摄取路径添加到摄取 URL。

例如，假设管道配置的摄取路径如下所示：

```
entry-pipeline:
  source:
    http:
      path: "/my/test_path"
```

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-west-2.osis.amazonaws.com/my/test_path`。

有关更多信息，请参阅 the section called “指定提取路径”。

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:us-east-1:{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 AWS 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（ AWS 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：

   ```
   {
     "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`：

```
{
  "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 Amaz OpenSearch on DynamoDB 上使用采集管道

您可以将 OpenSearch 采集管道与 DynamoDB 配合使用，将 DynamoDB 表事件（例如创建、更新和删除）流式传输到亚马逊服务域和集合。 OpenSearch OpenSearch Ingestion 管道整合了变更数据捕获 (CDC) 基础架构，以提供一种高规模、低延迟的方式来持续流式传输 DynamoDB 表中的数据。

您可以通过两种方式使用 DynamoDB 作为处理数据的来源：有或没有完整初始快照。

完整的初始快照是 DynamoDB 使用恢复 (PITR) 功能拍摄的表point-in-time 的备份。DynamoDB 将此快照上传到 Amazon S3。从那里，In OpenSearch gestion 管道将其发送到域中的一个索引，或者将其分区到域中的多个索引。为了保持 DynamoDB 中的数据一致 OpenSearch 性，管道将 DynamoDB 表中的所有创建、更新和删除事件与保存在一个或多个索引中的文档同步。 OpenSearch

当您使用完整的初始快照时，您的 OpenSearch 摄取管道会首先提取快照，然后开始从 DynamoDB Streams 读取数据。它最终会赶上并保持 DynamoDB 和之间近乎实时的数据一致性。 OpenSearch选择此选项时，必须在表中同时启用 PITR 和 DynamoDB 流。

您也可以使用 OpenSearch Ingestion 与 DynamoDB 的集成在没有快照的情况下流式传输事件。如果您已经拥有来自其他机制的完整快照，或者您只想通过 DynamoDB Streams 从 DynamoDB 表中流式传输当前事件，请选择此选项。选择此选项时，只需要在表中启用 DynamoDB 流。

有关此集成的更多信息，请参阅开发人员指南中的 DynamoDB 零 ETL 与 OpenSearch 亚马逊服务的集成。Amazon DynamoDB

主题

先决条件

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：

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

```
                "dynamodb:DescribeExport"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{account-id}:table/my-table/export/*"
            ]
        },
        {
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeStream",
                "dynamodb:GetRecords",
                "dynamodb:GetShardIterator"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{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/{exportPath}/*"
            ]
        }
    ]
}
```

您也可以使用 AWS 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:*us-west-2*:*{account-id}*:key/*my-key-id*。以下策略包括使用客户托管密钥所需的权限：

```
{
    "Sid": "allowUseOfCustomManagedKey",
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
```

```
        "kms:Decrypt"
    ],
    "Resource": arn:aws:kms:us-west-2:{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:us-west-2:{account-id}:table/table-a"
        export:
          s3_bucket: "my-bucket"
          s3_prefix: "export/"
        stream:
          start_position: "LATEST"
      aws:
        region: "us-west-2"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.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 “使用蓝图创建管道”。

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 中的每个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": {`	DynamoDB 支持数字。

数据类型	OpenSearch	DynamoDB

	<pre>"MixedNumberAttribute": {
	"type": "float"
	}
	}
	}
	}
	}</pre>
	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。

String	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
	
	<pre>{
	"template": {
	"mappings": {
	"properties": {
	"PartType": {
	"type": "keyword"
	}
	}
	}
	}
	}</pre>
		DynamoDB 支持字符串。

数据类型	OpenSearch	DynamoDB
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。
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。

数据类型	OpenSearch	DynamoDB
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。
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。

数据类型	OpenSearch	DynamoDB
列出	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。
设置	OpenSearch 根据集合中的内容为 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 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 会将所有由于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。 AWS 区域
- 您的 DynamoDB 表 OpenSearch 和摄取管道必须相同。 AWS 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。

在 Amaz OpenSearch on DocumentDB 中使用采集管道

您可以将 OpenSearch 采集管道与 Amazon DocumentDB 配合使用，将文档更改（例如创建、更新和删除）流式传输到 OpenSearch 亚马逊服务域和馆藏。 OpenSearch 摄取管道可以利用变更数据捕获 (CDC) 机制（如果您的亚马逊文档数据库集群上可用）或者 API 轮询来提供一种高规模、低延迟的方式来持续流式传输来自亚马逊文档数据库集群的数据。

您可以通过两种方式使用 Amazon DocumentDB 作为数据源来处理数据：有或没有完整的初始快照。

完整的初始快照是对整个 Amazon DocumentDB 集合的批量查询。亚马逊 DocumentDB 将此快照上传到亚马逊 S3。从那里，In OpenSearch gestion 管道将其发送到域中的一个索引，或者将其分区到域中的多个索引。为了保持 Amazon DocumentDB 中的数据并保持 OpenSearch 一致，该管道会将 Amazon DocumentDB 集合中的所有创建、更新和删除事件与保存在一个或多个索引中的文档同步。 OpenSearch

当您使用完整的初始快照时，您的 OpenSearch 摄取管道会首先提取快照，然后开始从 Amazon DocumentDB 变更流中读取数据。它最终会赶上，并保持了 Amazon Doc OpenSearch umentDB 和. 之间近乎实时的数据一致性。

您也可以使用 OpenSearch Ingestion 与 Amazon DocumentDB 的集成在没有快照的情况下直播事件。如果您已经拥有来自其他机制的完整快照，或者您只想使用更改流流流流式传输来自 Amazon DocumentDB 集合的时事，请选择此选项。

使用这两个选项，如果您在工作流配置中启用数据流，则必须在 Amazon DocumentDB 馆藏上启用更改流。如果您只使用满载或导出，则无需启用更改流。

先决条件

在创建 OpenSearch 摄取管道之前，请执行以下步骤：

1. 按照亚马逊 DocumentDB 开发者指南中创建亚马逊 DocumentD B 集群中的步骤，创建具有读取数据权限的 Amazon Documen tD B 集群。如果您使用 CDC 基础设施，请确保将您的 Amazon DocumentDB 集群配置为发布变更流。
2. 使用在您的亚马逊文档数据库集群上设置身份验证。 AWS 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。
3. 如果您使用更改流订阅 Amazon DocumentDB 馆藏中的数据更改，请使用参数将保留期延长至最多 7 天，从而避免数据丢失。`change_stream_log_retention_duration`默认情况下，Change streams 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。
4. 创建 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。
5. 将基于资源的策略附加到您的网域，或者将数据访问策略附加到您的馆藏。这些访问策略允许 OpenSearch Ingestion 将数据从您的 Amazon DocumentDB 集群写入您的域名或集合。

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

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

```
}
```

要创建具有访问集合或域名写入数据的正确权限的 IAM 角色，请参阅域必需权限和集合必需权限。

步骤 1：配置管道角色

设置 Amazon DocumentDB 管道先决条件后，配置要在工作流配置中使用的管道角色，并在该角色中添加以下 Amazon DocumentDB 权限：

```
{
    "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。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，要填充索引名称，该`getMetadata`函数将*`documentdb_collection`*用作元数据键。如果要在不使用`getMetadata`方法的情况下使用不同的索引名称，则可以使用该配置`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: ${aws_secrets:secret:password}
      aws:
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      s3_bucket: "bucket-name"
      s3_region: "bucket-region"
      s3_prefix: "path" #optional path for storing the temporary data
      collections:
        - collection: "dbname.collection"
          export: true
          stream: true
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      index: "${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"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
        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 “使用蓝图创建管道”。

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。AWS Management Console 为此，请找到 “网络配置” 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，或者选择自己的 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 与用于摄取的 CIDR 不同。 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。如果您在 Amazon DocumentDB 中为同一属性混合使用多种数据类型，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。在这些情况下，您应该提供一个显式的映射模板，该模板可以选择最灵活的数字类型，例如： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	亚马逊 DocumentDB 支持整数。
长整型	OpenSearch 自动将 Amazon DocumentDB 长整数值映射到 OpenSearch多头。	亚马逊 DocumentDB 支持多头。

数据类型	OpenSearch	Amazon DocumentDB
	OpenSearch 根据第一个发送的文档动态映射字段。如果您在 Amazon DocumentDB 中为同一属性混合使用多种数据类型，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。在这些情况下，您应该提供一个显式的映射模板，该模板可以选择最灵活的数字类型，例如： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	

数据类型	OpenSearch	Amazon DocumentDB
String	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" } } } } }`	亚马逊 DocumentDB 支持字符串。

数据类型	OpenSearch	Amazon DocumentDB
Double	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。如果您在 Amazon DocumentDB 中为同一属性混合使用多种数据类型，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。在这些情况下，您应该提供一个显式的映射模板，该模板可以选择最灵活的数字类型，例如： `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	亚马逊 DocumentDB 支持双打。

数据类型	OpenSearch	Amazon DocumentDB
Date	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" } } } } }`	亚马逊 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" } } } } }`	亚马逊 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。	亚马逊 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，您可以以双级精度查询和聚合字段。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。	亚马逊 DocumentDB 支持小数。
正则表达式	正则表达式类型创建嵌套字段。这些包括*<myFieldName>* .pattern和*<myFieldName>* .options。	亚马逊 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" } } } } }`	亚马逊 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。	亚马逊 DocumentDB 支持 ObjectID。
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。后续值仍然可以是亚马逊 DocumentDB 的空值。	亚马逊 DocumentDB 支持空类型字段。

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 未定义的值。	亚马逊 DocumentDB 支持未定义的类型字段。
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。后续值仍然可以是亚马逊 DocumentDB minKey 值。	亚马逊 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。后续值仍然可以是亚马逊 DocumentDB maxKey 值。	亚马逊 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。 AWS 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。 AWS 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。它不支持亚马逊 DocumentDB 弹性集群。
- OpenSearch Ingestion 集成仅支持 AWS Secrets Manager 作为您的 Amazon DocumentDB 集群的身份验证机制。
- 您无法更新现有工作流配置以从其他数据库或集合中提取数据。相反，您必须创建一个新管道。

在 Confluent OpenSearch Kafka 云中使用采集管道

你可以使用 Confluent Kafka 作为 OpenSearch Ingestion 中的来源，将数据从 Confluent Kafka 集群流式传输到亚马逊 OpenSearch 服务域或亚马逊无服务器集合。 OpenSearch OpenSearch Ingestion 支持在公共和私有网络空间中处理来自自管 Kafka 的流数据。

连接到 Confluent 公共 Kafka 云

您可以使用 OpenSearch Ingestion 管道从具有公共配置的 Confluent Kafka 集群中流式传输数据（必须公开解析引导服务器 DNS 名称）。为此，你需要一个 OpenSearch 采集管道、一个融合的 Kafka 集群作为源，以及一个亚马逊服务 OpenSearch 域或一个亚马逊 OpenSearch 无服务器集合作为目标。

要迁移数据，必须具备以下条件：

- 充当源的 Confluent Kafka 集群。集群应包含您要迁移的数据。
- 作为目的地的亚马逊 OpenSearch 服务域名或亚马逊 OpenSearch 无服务器集合。

- Kafka 集群应使用来自 AWS Secrets Manager的凭据启用身份验证。

要求

要在自行管理的 OpenSearch 或 Elasticsearch 源集群上启用 AWS Secrets Manager 基于身份验证的功能，您必须

- 按照轮换密钥中的步骤在你的 Confluent Kafka 集群上设置身份验证。 AWS Secrets ManagerAWS Secrets Manager
- 在 IAM 中创建具有写入亚马逊 OpenSearch 服务域或亚马逊 OpenSearch 无服务器集合权限的管道角色。您还必须指定读取证书的权限 AWS Secrets Manager。要实现此目的，应按照以下步骤进行：
  - 将基于资源的策略附加到您的 Amazon S OpenSearch ervice 域或将数据访问策略附加到您的馆藏。这些访问策略允许 OpenSearch Ingestion 将数据从您的自行管理 OpenSearch 或 Elasticsearch 源集群写入您的亚马逊服务 OpenSearch 域或您的亚马逊无服务器集合。OpenSearch
- 参考蓝 OpenSearch 图创建摄取管道。

  完成这些步骤后，您的管道将自动开始处理源集群中的数据，并将其提取到您的亚马逊 OpenSearch 服务域或亚马逊 OpenSearch 无服务器收集目标中。您可以在 Ingesti OpenSearch on 管道中使用各种处理器对摄取的数据执行任何转换。

IAM 角色和权限

以下示例域访问策略允许您在下一步中创建的管道角色向 Amazon S OpenSearch ervice 域写入数据。确保使用自己的 ARN 更新资源。

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

管理网络接口需要以下权限：

```
{
    "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 AWS Secrets Manager 服务中读取机密所需的权限：

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

创建管道

将策略附加到管道角色后，使用 Confluent Kafka 数据迁移管道蓝图创建管道。该蓝图包括在 Kafka 和目标之间迁移数据的默认配置。

- 您可以指定多个 Amazon OpenSearch 服务域作为数据的目的地。此功能允许有条件地路由或将传入的数据复制到多个 Amazon OpenSearch Servicedomains。
- 您可以将数据从源 Confluent Kafka 集群迁移到亚马逊无服务器 V OpenSearch PC 集合。确保在管道配置中提供网络访问策略。
- 您可以使用融合架构注册表来定义和融合架构。

以下示例管道将数据从 Confluent Kafka 集群提取到亚马逊服务域： OpenSearch

```
version: "2"
kafka-pipeline:
  source:
    kafka:
      # Encryption is always required
      encryption:
        type: "ssl"
      topics:
        - name: "topic_4"
          group_id: "demoGroup"
      bootstrap_servers:
        # TODO: for public confluent kafka use public booststrap server dns
        - "<<bootstrap-server>>.us-west-2.aws.private.confluent.cloud:9092"
      authentication:
        sasl:
          plain:
            username: "${{aws_secrets:confluent-kafka-secret:username}}"
            password: "${{aws_secrets:confluent-kafka-secret:password}}"
      # Schema is optional
      schema:
        type: confluent
        registry_url: https://<<registry-url>>.us-west-2.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://<<opensearchdomain>>.us-west-2.es.amazonaws.com" ]
```

```
        index: "enterprise-confluent-demo"
        aws:
          sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
          region: "<<aws-region>>"
extension:
 aws:
   secrets:
     confluent-kafka-secret:
       secret_id: "enterprise-kafka-credentials"
       region: "<<aws-region>>"
       sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
     schema-secret:
       secret_id: "self-managed-kafka-schema"
       region: "<<aws-region>>"
       sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
```

在 VPC 中连接到 Confluent Kafka 云

您可以使用 OpenSearch 摄取管道从具有公共配置的 Confluent Kafka 集群中流式传输数据。为此，请设置一个以 Confluent Kafka 为来源、亚马逊服务 OpenSearch 域或亚马逊 OpenSearch 无服务器集合作为目标的 OpenSearch 摄取管道。管道处理来自你的 kafka 集群的所有流数据，并将数据提取到目标集群。

Confluent Kafka 网络配置

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

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

你可以使用 Confluent 托管 Kafka 作为从 Confluent 云中提取数据的来源。为实现这一目标，您需要设置一个管道，将 Kafka 配置为来源，将 Amazon Serv OpenSearch ice 域或 Amazon OpenSearch Serverless 集合配置为接收器。这便于将数据从 Kafka 迁移到指定目的地。迁移还支持使用 confluent 注册表或根本不使用注册表。

要执行数据迁移，您需要以下资源：

- 充当源的 Confluent Kafka 集群，包含你打算迁移的数据。
- 目标目的地，例如亚马逊 OpenSearch 服务域名或亚马逊 OpenSearch 无服务器集合作为接收器。
- 有权访问 Confluent VPC 的亚马逊 VPC 的 VPC ID。
- Kafka 集群应使用来自 AWS Secrets Manager的凭据启用身份验证。

要求

要在 Kafka 集群上设置摄取，需要满足以下条件：

- 您必须在 Kafka 集群上启用 AWS Secrets Manager 基于身份验证的功能。
  - 使用在 Kafka 集群上设置身份验证。 AWS Secrets Manager按照轮换密钥中的步骤启用AWS Secrets Manager 密钥轮换。
- 您需要提供 VPC CIDR 以供 OpenSearch 摄取服务使用。
  - 如果您使用 AWS 管理控制台创建管道，则还必须将 Amazon OpenSearch Ingestion 管道附加到您的 VPC，才能使用 Confluent Kafka 作为来源。为此，请找到 “网络配置” 部分，选中 “连接到 VPC” 复选框，然后选择您的 CIDR 或手动输入要由摄取使用的任何 /24 CIDR。 OpenSearch OpenSearch Ingestion 选择使用的 CIDR 应不同于运行 Confluent 托管 Kafka 的 VPC CIDR。有关需要避免使用的 Confluent Kafka CIDR 的更多信息，请点击此处。以下是 OpenSearch 接入服务可用于创建网络连接的默认 CIDR 选项。
    - 10.99.20.0/24
    - 192.168.36.0/24
    - 172.21.56.0/24
- 您需要在 IAM 中创建一个管道角色，该角色具有访问亚马逊 OpenSearch 服务域或 Amazon OpenSearch Serverless 集合的权限以及从中 AWS Secrets Manager读取密钥的权限。
  - 将基于资源的策略附加到您的 Amazon Ser OpenSearch vicedomain，或者将亚马逊 OpenSearch 无服务器数据访问策略附加到您的馆藏。这些访问策略允许 OpenSearch Ingestion 将数据从你的 Kafka 写入你的亚马逊 OpenSearch 服务域或亚马逊 OpenSearch 无服务器集合。

- 对于具有连接功能的 Confluent Kafka，请配置 AWS PrivateLink

  VPC DHCP 选项。应启用 DNS 主机名和 DNS 解析。

  - 域名：aws.private.confluent.cloud

    domain-name-servers: 亚马逊提供了 DNS

IAM 角色和权限

以下示例域访问策略允许管道角色向 Amazon S OpenSearch ervice 域写入数据。

> **Note**
>
> 您需要使用自己的 AR `resource` N 进行更新。

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

以下示例提供了管理网络接口所需的权限：

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

以下示例提供了从中读取密钥所需的权限 AWS Secrets Manager：

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

```
}
```

以下示例提供了写入亚马逊 OpenSearch 服务域所需的权限：

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

创建管道

将策略附加到管道角色后，您可以使用 Confluent Kafka 数据迁移管道蓝图来创建您的管道。该蓝图包括在 Kafka 和目标之间迁移数据的默认配置。

- 您可以指定多个 Amazon OpenSearch 服务域作为数据的目的地。此功能允许有条件地将传入的数据路由或复制到多个 Amazon OpenSearch 服务中。
- 您可以将数据从源 Confluent Kafka 集群迁移到亚马逊无服务器 V OpenSearch PC 集合。确保在管道配置中提供网络访问策略。
- 您可以使用 Confluent 架构注册表来定义和 Confluent 架构。

管道配置示例

```
version: "2"
kafka-pipeline:
  source:
    kafka:
      # Encryption is always required
      encryption:
        type: "ssl"
      topics:
        - name: "topic_4"
          group_id: "demoGroup"
      bootstrap_servers:
```

```
        # TODO: for public confluent kafka use public booststrap server dns
        - "<<bootstrap-server>>.us-west-2.aws.private.confluent.cloud:9092"
      authentication:
        sasl:
          plain:
            username: "${{aws_secrets:confluent-kafka-secret:username}}"
            password: "${{aws_secrets:confluent-kafka-secret:password}}"
      # Schema is optional
      schema:
        type: confluent
        registry_url: https://<<registry-url>>.us-west-2.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://<<opensearchdomain>>.us-west-2.es.amazonaws.com" ]
        index: "enterprise-confluent-demo"
        aws:
          sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
          region: "<<aws-region>>"
extension:
  aws:
    secrets:
      confluent-kafka-secret:
        secret_id: "enterprise-kafka-credentials"
        region: "<<aws-region>>"
        sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
      schema-secret:
        secret_id: "self-managed-kafka-schema"
        region: "<<aws-region>>"
        sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
```

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 MSK）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 AWS PrivateLink 用于连接亚马逊 MSK。您可以从 Amazon MSK 和 Amazon MSK 无服务器集群中提取数据。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步骤。

主题

亚马逊 MSK 先决条件

在创建 OpenSearch 摄取管道之前，请执行以下步骤：

1. 按照《适用于 Apache Managed Streaming 的亚马逊管理流媒体 Kafk a 开发者指南》中创建集群中的步骤创建 Amazon MSK 预配置集群。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 AWS 账户。此策略允许 OpenSearch Ingestion 创建与您的 Amazon MSK 集群的 AWS 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 AWS 账户时，适用以下策略：

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

如果您的 Amazon MSK 集群与您的管道 AWS 账户 不同，请改为附加以下策略。请注意，只有预配置的 Amazon MSK 集群才能进行跨账户访问，Amazon MSK 无服务器集群无法进行跨账户访问。的 ARN AWS `principal` 应该是您为管道 YAML 配置提供的相同管道角色的 ARN：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{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:us-east-1:{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* 是私有端点 (single-VPC) 引导 URL 之一。的值`--replication-factor`应为`2`或`3`，具体取决于您的 Amazon MSK 集群拥有的区域数量。`--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。同样，确保 *BootstrapServerString* 是私有端点 (single-VPC) 引导 URL 之一。

Amazon MSK 无服务器先决条件

在创建 OpenSearch 摄取管道之前，请执行以下步骤：

1. 按照 Apache Managed Streaming for Apache Kafka 开发者指南中创建 MSK 无服务器集群中的步骤创建亚马逊 MSK 无服务器集群。
2. 集群处于 “活动” 状态后，按照将集群策略附加到 MSK 集群中的步骤来附加以下策略。确保使用自身 ARN 更新 `resource`。

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

```
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{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:us-east-1:{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 AWS 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 AWS 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 引导网址之一。的值`--replication-factor`应为`2`或`3`，具体取决于您的 Amazon MSK Serverless 集群拥有的区域数量。`--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 引导网址之一。

步骤 1：配置管道角色

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster",
                "kafka:DescribeClusterV2",
                "kafka:GetBootstrapBrokers"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:topic/cluster-name/cluster-
id/topic-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:group/cluster-name/*"
            ]
        }
    ]
}
```

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

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

步骤 3：（可选）使用 AWS 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。 AWS Glue 在 AWS 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 AWS Glue 读取访问权限。您可以使用名为的 AWS 托管策略AWSGlueSchemaRegistryReadonlyAccess。此外，您的注册表必须 AWS 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 (OCU)

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。如果主题包含大量分区（例如，超过 96 个，即每个管道的最大 OCU），我们建议您将管道配置为 1–96 个 OCU。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。通过向同一个主题和使用者组添加另一个包含一组新 OCU 的管道，几乎可以线性扩展吞吐量。

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
- 亚马逊 S3 跨账户作为来源

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 AWS 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`地图。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：

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action":[
          "s3:ListBucket",
          "s3:GetBucketLocation",
          "s3:GetObject"
       ],
      "Resource": "arn:aws:s3:::my-bucket/*"
    },
    {
       "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:
      ...
      sts_role_arn: arn:aws:iam::{account-id}:role/pipeline-role
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-1.amazonaws.com/{account-id}/ingestion-queue"
      compression: "none"
      aws:
        region: "us-east-1"
        # IAM role that the pipeline assumes to read data from the queue. This role
 must be the same as the pipeline role.
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
  processor:
  - grok:
      match:
        message:
        - "%{COMMONAPACHELOG}"
  - date:
      destination: "@timestamp"
      from_time_received: true
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
      index: "index-name"
      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"
```

如果您在 Amazon S3 上处理小文件时发现 CPU 使用率较低，请考虑通过修改该`workers`选项的值来增加吞吐量。有关更多信息，请参阅 S 3 插件配置选项。

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。

要将您的管道配置为从 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 “使用蓝图创建管道”。

一次性扫描

一次性计划扫描运行一次。在 YAML 配置中，您可以使用 `start_time` 和 `end_time` 指定希望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: "us-east-1"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      acknowledgments: true
      scan:
        buckets:
          - bucket:
              name: my-bucket-1
              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-1.es.amazonaws.com"]
        index: "index-name"
        aws:
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
        dlq:
          s3:
            bucket: "my-bucket-1"
            region: "us-east-1"
            sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
```

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
```

停止管道会移除任何先前存在的关于管道在停止之前扫描过的对象的引用。如果单个扫描管道停止，它将在启动后重新扫描所有对象，即使它们已经被扫描。如果您需要停止单个扫描管道，建议您在重新启动管道之前更改时间窗口。

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。按名字筛选不需要你停止并启动管道。为此，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 YAML 配置中，`interval` 将指定定期扫描频率，范围介于 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 “使用蓝图创建管道”。

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 ND-JSON。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。

要使用此类过滤器创建管道，请使用预配置的汇过滤器蓝图。有关更多信息，请参阅 the section called “使用蓝图创建管道”。

亚马逊 S3 跨账户作为来源

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 用户指南中的存储桶拥有者授予跨账户存储桶权限。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 YAML 配置，`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 AWS 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。

要将您的管道配置为从 Security Lake 读取，请使用预配置的 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 “使用蓝图创建管道”。

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

在创建 OpenSearch 摄取管道之前，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。
- 对于订阅用户凭证，请添加计划在其中创建管道的 AWS 账户 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。

步骤 1：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：

```
{
   "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.{region}.amazonaws.com/{account-id}/
AmazonSecurityLake-abcde-Main-Queue"
    aws:
      ...
      sts_role_arn: arn:aws:iam::{account-id}:role/pipeline-role
processor:
  ...
sink:
  - opensearch:
      ...
```

步骤 2：创建管道

向管道角色添加权限后，使用预配置的 S3 蓝图创建管道。有关更多信息，请参阅 the section called “使用蓝图创建管道”。

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.{region}.amazonaws.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。

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 AWS 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-1.osis.amazonaws.com
  Port 443
  URI /log/ingest
  Format json
  aws_auth true
  aws_region us-east-1
  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-1.es.amazonaws.com"]
        index: "index_name"
        index_type: custom
        bulk_size: 20
        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"
```

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，它为不同的语言和子项目（如 Fluent Bit）提供软件开发工具包。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 AWS 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-1.osis.amazonaws.com/apache-log-pipeline/logs
  json_array true

  <auth>
    method aws_sigv4
    aws_service osis
    aws_region us-east-1
    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-1.es.amazonaws.com"]
        index: "index_name"
        aws_sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
        aws_region: "us-east-1"
        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-1.osis.amazonaws.com。
- service 值必须为 osis。
- OTLP/HTTP 导出器的compression选项必须与管道源上的compression选项相匹配。OpenTelemetry

```
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/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:
          # 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://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        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"
```

有关另一个管道示例，请参阅预配置的跟踪分析蓝图。有关更多信息，请参阅 the section called “使用蓝图创建管道”。

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”

使用 Amazon OpenSearch Ingestion 在域名和集合之间迁移数据

您可以使用 OpenSearch 摄取管道在亚马逊 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，在该管道中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。这可以有效地将您的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。此域或集合包含您要迁移的数据。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。
- 您要将数据迁移到的单独域或 VPC 集合。此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。您可以在工作流配置中包含此角色的 Amazon 资源名称 (ARN)。有关更多信息，请参阅以下资源：
  - the section called “授予管道对域的访问权限”
  - the section called “授予管道访问集合的权限”

主题

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。不支持公共馆藏。
- 您不能在单个管道配置中指定 VPC 和公共域的组合。
- 在单个管道配置中，您最多可以有 20 个非管道接收器。
- 在单个管道配置 AWS 区域 中，最多可以指定三个不同的接收器。
- 如果具有多个接收器的管道中断时间过长，或者没有配置足够的容量来接收传入的数据，则随着时间的推移，具有多个接收器的管道的处理速度可能会降低。

OpenSearch 服务即来源

您指定为源的域或集合是数据迁移的来源。

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 中创建新的权限策略以附加到管道角色。确保您允许从源代码读取和写入接收器的权限。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 “授予管道对域的访问权限”和the section called “授予管道访问集合的权限”。
2. 在管道角色中指定以下权限以从源代码读取：

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"es:ESHttpGet",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_cat/indices",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/scroll",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpPost",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search/
point_in_time",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search/scroll"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpDelete",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/
point_in_time",
            "arn:aws:es:us-east-1:{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 “使用蓝图创建管道”。

> **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.us-east-1.es.amazonaws.com" ]
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          region: "us-east-1"
        index: "response-2xx"
        routes:
          - 2xx_status
    - opensearch:
        hosts: [ "https://search-response-5xx.us-east-1.es.amazonaws.com" ]
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          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。您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 “使用蓝图创建管道”。

使用 AWS SDK 与 Amazon OpenSearch Ingestion 进行交互

本节包含一个关于如何使用 AWS SDK 与 Amazon OpenSearch Ingestion 进行交互的示例。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 AWS SDK for Python (Boto3) 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，将访问策略中的账户 ID 替换为您的 AWS 账户 ID。您也可以选择修改 `region`。

```
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.

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)

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-1:123456789012:domain\/{domainName}\"}},{{\"Effect\":\"Allow\",\"Action\":
\"es:ESHttp*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012: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::123456789012:role\/PipelineRole
\"}},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012: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
     sts_role_arn: \"arn:aws:iam::123456789012:role/PipelineRole\"\n          region:
 \"us-east-1\"'
        response = osis.create_pipeline(
```

```
            PipelineName=pipelineName,
            MinUnits=4,
            MaxUnits=9,
            PipelineConfigurationBody=definition
        )

        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 中的安全性

AWS 十分重视云安全性。作为 AWS 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是 AWS 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云的安全性 – AWS负责保护在AWS云中运行AWS服务的基础设施。AWS还向您提供可安全使用的服务。作为 AWS 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。
- 云中的安全性 - 您的责任由您使用的 AWS 服务决定。您还需要对其它因素负责，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。

此文档将帮助您了解如何在使用 OpenSearch Ingestion 时应用责任共担模型。以下主题说明如何配置 OpenSearch Ingestion 以实现您的安全性和合规性目标。您还会了解如何使用其他 AWS 服务以帮助您监控和保护 OpenSearch Ingestion 资源。

主题

- 为 Amazon OpenSearch Ingestion 管道配置 VPC 访问权限
- 适用于 Amazon OpenSearch Ingestion 的 Identity and Access 管理
- 使用 AWS CloudTrail 记录 Amazon OpenSearch Ingestion API 调用

为 Amazon OpenSearch Ingestion 管道配置 VPC 访问权限

您可以使用接口 VPC 终端节点访问您的 Amazon OpenSearch Ingestion 管道。VPC 是专为您服务的虚拟网络 AWS 账户。它在逻辑上与 AWS 云中的其他虚拟网络隔离。通过 VPC 终端节点访问管道可实现 OpenSearch Ingestion 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 AWS 云中。

OpenSearch Ingestion 通过创建由提供支持的接口端点来建立此私有连接。 AWS PrivateLink我们在创建管道时指定的每个子网中创建一个终端节点网络接口。这些是请求者管理的网络接口，是发往 OpenSearch 摄取管道的流量的入口点。您也可以选择自己创建和管理接口端点。

使用 VPC 可以强制数据流通过 VPC 边界内的 OpenSearch 摄取管道，而不是通过公共互联网。非 VPC 内部管道通过面向公众的端点和互联网收发数据。

具有 VPC 访问权限的管道可以写入公共或 VPC OpenSearch 服务域，也可以写入公共或 VPC OpenSearch 无服务器集合。

主题

注意事项

为管道配置 VPC 时，请考虑以下事项。

- 管道不必与其接收器位于同一 VPC 中。您也不需要在两个 VPC 之间建立连接。 OpenSearch Ingestion 负责为你连接它们。
- 您只能为管道指定一个 VPC。
- 与公共管道不同，VPC 管道必须与其写入的域或集合接收器 AWS 区域 相同。
- 您可以选择将管道部署到 VPC 的一个、两个或三个子网中。子网分布在部署您的摄取 OpenSearch 计算单元 (OCU) 的相同可用区中。
- 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。如果您不提供安全组， OpenSearch Ingestion 将使用在 VPC 中指定的默认安全组。

限制

具有 VPC 访问权限的管道有以下限制。

- 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 终端节点或公共终端节点启动管道，但不能两者兼而有之。在创建管道时只能选择其一。
- 在配置了具有 VPC 访问权限的管道后，您无法将其移至其他 VPC，也无法更改其子网或安全组设置。
- 如果您的管道写入使用 VPC 访问权限的域或集合接收器，则在创建管道后，您将无法返回并更改接收器（VPC 或公共）。必须删除，然后使用新的接收器重新创建管道。您仍然可以从公共接收器切换到具有 VPC 访问权限的接收器。
- 您无法提供对 VPC 管道的跨账户摄取访问权限。

先决条件

在配置具有 VPC 访问权限的管道之前，您必须执行以下操作：

- 创建 VPC

  要创建您的 VPC，您可以使用 Amazon VPC 控制台、 AWS CLI 或其中一个 AWS 软件开发工具包。有关更多信息，请参阅 Amazon VPC 用户指南 中的使用 VPC。如果您已有 VPC，请跳过此步骤。
- 预留 IP 地址

  OpenSearch Inge stion 会在您在创建管道时指定的每个子网中放置一个 elastic network 接口。每个网络接口都与一个 IP 地址关联。每个子网必须为网络接口保留一个 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 访问权限 AWS CLI。

控制台

您可以在管道创建期间配置 VPC 访问权限。在网络下，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 终端节点，还是让 OpenSearch Ingestion 为您创建终端节点。

设置	描述
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 AWS 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 HTTP）到达接 OpenSearch 入管道。
VPC 连接选项	如果您的源是自行管理的终端节点，请将您的管道连接到 VPC。选择提供的默认 CIDR 选项之一，或使用自定义 CIDR。

CLI

要使用配置 VPC 访问权限 AWS 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"
```

自管 VPC 终端节点

创建管道时，您可以使用端点管理来创建具有自我管理终端节点或服务管理终端节点的管道。端点管理是可选的，默认为由 OpenSearch Ingestion 管理的端点。

要在中创建带有自行管理 VPC 终端节点的管道 AWS Management Console，请参阅使用 OpenSearch 服务控制台创建管道。要在中创建带有自行管理 VPC 终端节点的管道 AWS CLI，您可以在 create- pipelin `--vpc-options` e 命令中使用参数：

```
--vpc-options SubnetIds=subnet-abcdef01234567890,VpcEndpointManagement=CUSTOMER
```

在指定终端节点服务时，您可以自己为管道创建终端节点。要查找您的终端节点服务，请使用 get-pipelin e 命令，该命令会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 AWS Management Console 或的 VPC 终端节点 AWS CLI。

如果您使用自行管理的 VPC 终端节点，则必须在您的 VPC enableDnsHostnames 中启用 DNS 属enableDnsSupport性和和。请注意，如果您的管道中包含可以停止并重启的自管理终端节点，则必须在您的账户中重新创建 VPC 终端节点。

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的 VPC 终端节点， OpenSearch Ingestion 需要一个名为的服务相关角色来AWSServiceRoleForAmazonOpenSearchIngestionService访问您的 VPC、创建管道终端节点并将网络接口放置在您的 VPC 的子网中。

如果您选择自我管理的 VPC 终端节点， OpenSearch Ingestion 需要一个名为的服务相关角色。AWSServiceRoleForOpensearchIngestionSelfManagedVpce有关这些角色、其权限以及如何删除它们的更多信息，请参阅the section called “管道创建角色”。

OpenSearch 当您创建摄取管道时，Ingestion 会自动创建角色。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。角色创建后，您可以在 AWS Identity and Access Management (IAM) 控制台中查看该角色。

适用于 Amazon OpenSearch Ingestion 的 Identity and Access 管理

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 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 AWS 服务 ，无需支付额外费用。

主题

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》中的创建 IAM policy。

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 AWS 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。 适用于 OpenSearch 无服务器的基于身份的策略示例

用于 OpenSearch 摄取的策略资源

支持策略资源	是

管理员可以使用 AWS 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密钥	否

管理员可以使用 AWS 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 AWS 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 AWS 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 policy 元素：变量和标签。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 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 AWS，这些属性称为标签。您可以向 IAM 实体（用户或角色）和许多 AWS 资源附加标签。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》中的什么是 ABAC？。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 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的AWS 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 AWS Management Console 使用的是临时证书。例如，当您 AWS 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 切换到角色（控制台）。

您可以使用 AWS CLI 或 AWS 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 AWS。 AWS 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 AWS 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 AWS 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`名为的服务相关角色也`AWSServiceRoleForOpensearchIngestionSelfManagedVpce`适用于具有自我管理 VPC 终端节点的管道。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 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS 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 policy。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。

有关 Amazon OpenSearch Ingestion 定义的操作和资源类型（包括每种资源类型的 ARN 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。

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 AWS 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 AWS 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 AWS 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AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 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 AWS 托管策略 或 工作职能的AWS 托管策略。
- 应用最低权限 – 在使用 IAM policy 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 policy 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 AWS 服务的（例如）使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 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 policy，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 policy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 IAM Acess Analyzer 策略验证。

- 需要多重身份验证 (MFA)-如果 AWS 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 配置受 MFA 保护的 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 中的安全最佳实操。

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 AWS 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 AWS CLI 或 AWS 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：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ]
        }
    ]
}
```

或者，您可以使用the section called “AmazonOpenSearchIngestionReadOnlyAccess” AWS 托管策略，该策略授予对所有 OpenSearch Ingestion 资源的只读访问权限。 AWS 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。

```
{
    "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。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:region:123456789012:pipeline/*",
            "Action": [
                "osis:Ingest"
            ],
            "Effect": "Allow"
```

```
        }
    ]
}
```

使用 AWS CloudTrail 记录 Amazon OpenSearch Ingestion API 调用

Amazon OpenSearch Ingestion 与 AWS CloudTrail 集成，后者是在 OpenSearch Ingestion 中提供用户、角色或 AWS 服务所采取操作的记录的服务。

CloudTrail 以事件的形式捕获针对 OpenSearch Ingestion 的所有 API 调用。捕获的调用包括来自 OpenSearch Service 控制台 OpenSearch Ingestion 部分的调用，以及针对 OpenSearch Ingestion 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 控制台中的 Event history（事件历史记录）中查看最新事件。

使用 CloudTrail 收集的信息，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。

要了解有关 CloudTrail 的更多信息，请参阅《AWS CloudTrail用户指南》。

CloudTrail 中的 OpenSearch Ingestion 信息

在您创建 AWS 账户 时，将在该账户上启用 CloudTrail。当 OpenSearch Ingestion 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在事件历史记录中。您可以在 AWS 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 CloudTrail 事件历史记录查看事件。

要持续记录 AWS 账户 中的事件（包括 OpenSearch Ingestion 的事件），请创建跟踪。通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service（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AWS 区域。

此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Simple Storage Service（Amazon S3）桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知

- 从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

所有 OpenSearch Ingestion 操作都由 CloudTrail 记录，并记录在 OpenSearch Ingestion API 参考中。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（AWS Identity and Access ManagementIAM）用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其它 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 OpenSearch Ingestion 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 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具体顺序显示。

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 AWS 为 AWS 资源分配的元数据标记。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 AWS 资源。许多 AWS 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 OpenSearch Ingestion 管道。
- 跟踪您的 AWS 成本。您可以在 AWS Billing and Cost Management 控制面板上激活这些标签。AWS 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 AWS Billing 用户指南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 Service 控制台、AWS CLI、OpenSearch Ingestion API 或 AWS SDK 在管道中添加、管理和移除标签。

主题

所需权限

OpenSearch Ingestion 使用以下 AWS Identity and Access Management Access Analyzer (IAM) 权限来标记管道：

- `osis:TagResource`
- `osis:ListTagsForResource`
- `osis:UntagResource`

有关每个权限的更多信息，请参阅《服务授权参考》中的 OpenSearch Ingestion 操作、资源和条件键。

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. 登录到位于 https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 控制台。
2. 在左侧导航窗格中，选择摄取。
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 Save（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《AWS 管理控制台入门指南》中的标签编辑器。

使用标签 (AWS CLI)

要使用 AWS CLI 标记管道，请发送 `TagResource` 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 CloudWatch 记录和监控 Amazon OpenSearch Ingestion

Amazon OpenSearch Ingestion 将指标和日志发布到 Amazon CloudWatch。

主题

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 Amazon CloudWatch Logs。CloudWatch Logs 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 `FATAL` 日志级别。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 CloudWatch Logs，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：

```
{
    "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. 登录到位于 https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 控制台。
2. 在左侧导航窗格中选择摄取，然后选择要为其启用日志的管道。
3. 选择编辑日志发布选项。
4. 选择发布到 CloudWatch Logs。

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，`/aws/vendedlogs/OpenSearchService/OpenSearchIngestion`）下的所有日志组授予权限。

   > **Important**
   >
   > 必须在日志组名称中包含前缀 `vendedlogs`，否则创建失败。

6. 选择 Save（保存）。

CLI

要使用 AWS CLI 启用日志发布，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 CloudWatch 监控 Amazon OpenSearch Ingestion。Amazon CloudWatch 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 CloudWatch 原始数据的图表。

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
- 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`

指标后缀	描述
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）。 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

指标后缀	描述
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。 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

指标后缀	描述
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
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

指标后缀	描述
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

指标后缀	描述
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 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

指标后缀	描述
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）。

指标后缀	描述
	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 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
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。

指标后缀	描述
	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 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。 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

指标后缀	描述
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
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

指标后缀	描述
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
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

指标后缀	描述
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
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。 相关统计数据：总计 维度：PipelineName
actionConcludeGrou pEventsDropped.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
actionConcludeGrou pEventsProcessingE rrors.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
currentAggregateGr oups.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。

指标后缀	描述
	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.datePro cessingMatchSuccess.count。

指标后缀	描述
datePro cessingMatchSuccess.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
datePro cessingMatchFailure.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

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

指标后缀	描述
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
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 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。

指标后缀	描述
	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

指标后缀	描述
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 接收器。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 OpenSearch Service 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 OpenSearch Service 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

指标后缀	描述
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。 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

指标后缀	描述
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
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

指标后缀	描述
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
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

指标后缀	描述
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
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

指标后缀	描述
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
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）。

指标	描述
	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 OCU) 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

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。

推荐 CloudWatch 警报

当一段时间内 CloudWatch 指标超出指定的值时，CloudWatch 警报将会执行某个操作。例如，您可能希望 AWS 在集群运行状况为 `red` 的时间超过 1 分钟时向您发送电子邮件。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，请参阅的《Amazon CloudWatch 用户指南》中的创建 Amazon CloudWatch 警报。

警报	问题
`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	该管道无法写入 OpenSearch 接收器。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）。
`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`	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。

警报	问题
平均 >= 80% 达到 15 分钟，连续 3 次	
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	问题
dynamodb.exportJobFailure.count 总计为 1	尝试触发导出到 Amazon S3 失败。
opensearch.EndtoEndLatency.avg 平均 > X 达到 15 分钟，连续 4 次	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。导出后 EndtoEndLatency 会更高，但随着时间的推移，它会赶上最新的 DynamoDB 流，从而降低。
dyanmodb.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 >= opensearc	发送到 DLQ 的记录数量多于 OpenSearch 接收器。查看 OpenSearch 接收器插件指标，以调查和确定根本原因。

警报	问题
h.documentSuccess.count 总计达到 1 分钟，连续 1 次	
grok.grokProcessingTimeouts.count 总计 = recordsIn.count sum 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。
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 s3ObjectsNotFound.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。

警报	问题
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。
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。

警报	问题
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。
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。S OpenSearch erverless OpenSearch 集合是根据应用程序需求扩展计算容量的集群。这与 OpenSearch 服务配置 OpenSearch 域形成鲜明对比，后者需要手动管理其容量。

OpenSearch Serverless 为不频繁、间歇性或不可预测的工作负载提供了一种简单、经济实惠的选择。它非常经济高效，因为它会自动扩展计算容量，以匹配应用程序的使用情况。

OpenSearch 无服务器集合具有与预配置 OpenSearch 服务域相同的高容量、分布式和高可用性存储量。

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”。

主题

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 是亚马逊 OpenSearch 服务的按需无服务器配置。Serverless 消除了配置、配置和调整集群的操作复杂性。 OpenSearch对于不想自行管理集群的组织或没有专门资源或专业知识来运营大型 OpenSearch 集群的组织来说，这是一个不错的选择。借 OpenSearch 助 Serverless，您可以轻松搜索和分析大量数据，而不必担心底层基础设施和数据管理。

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。集合比需要手动配置的自我管理 OpenSearch 集群更易于使用。

集合具有与预配置 OpenSearch 服务域相同的高容量、分布式和高可用性存储卷，但由于不需要手动配置和调整，因此它们消除了更多的复杂性。数据在集合中传输时会被加密。 OpenSearch Serverless 还支持 OpenSearch 仪表板，它为分析数据提供了直观的界面。

无服务器集合目前运行 OpenSearch 版本 2.0.x。随着新版本的发布， OpenSearch Serverless 将自动升级您的集合，以使用新功能、错误修复和性能改进。

主题

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”。

开始使用

要开始使用 OpenSearch Serverless，请使用 OpenSearch 服务控制台、或其中一个 AWS 软件开发工具包创建一个或多个集合。 AWS CLI有关帮助您快速启动和运行集合的教程，请参阅the section called “ OpenSearch 无服务器入门”。

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”。

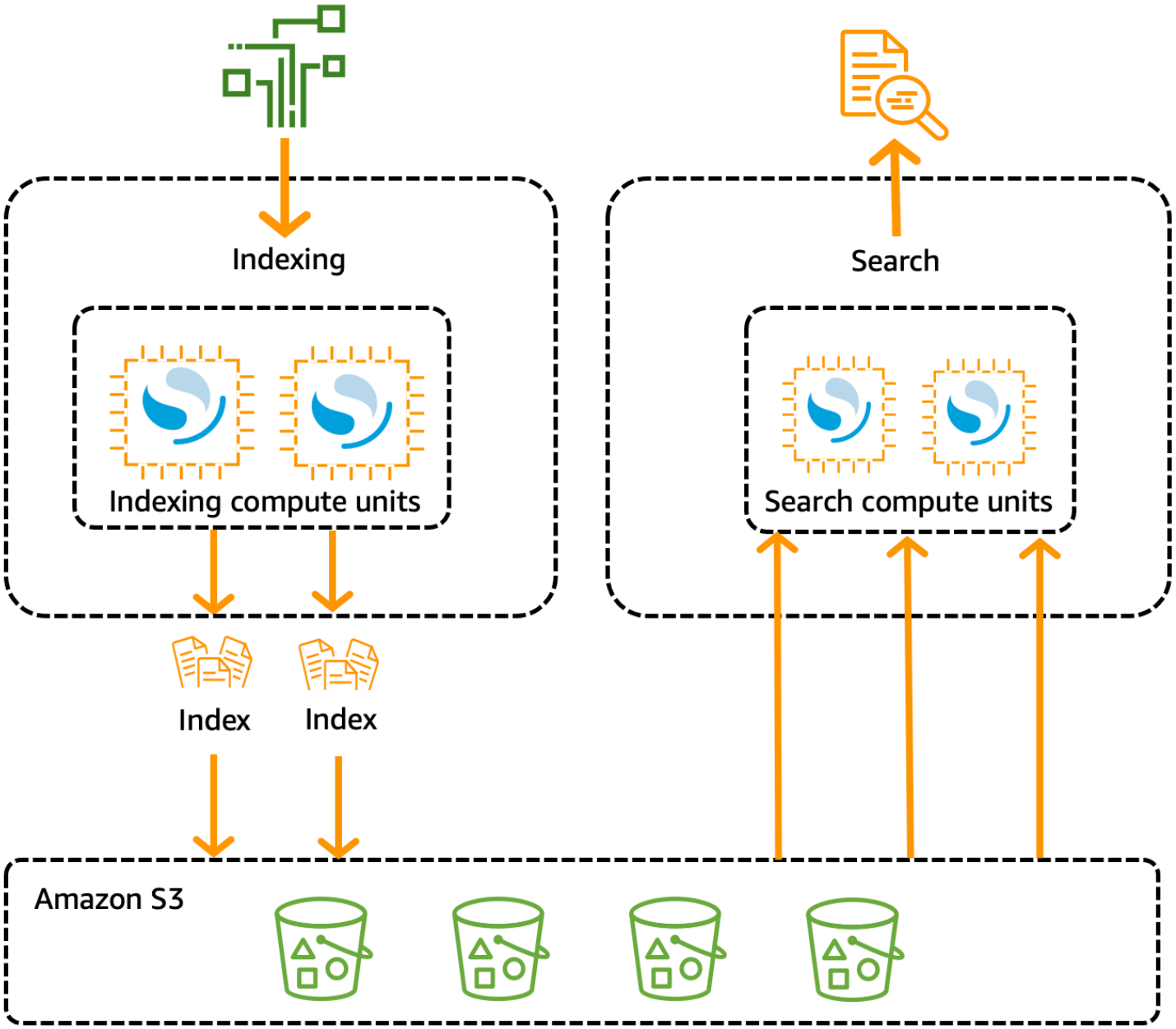
工作方式

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 (OCU) 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。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 OCU，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，并且仅实例化两个 OCU。默认情况下，冗余活动副本已启用，这意味着总共为账户中的第一个集合实例化了四个 OCU。

即使在任何集合端点上都没有活动，这些 OCU 也仍然存在。所有后续集合都将共享这些 OCU。当您在同一个账户中创建其他集合时， OpenSearch Serverless 仅根据您指定的容量限制，根据需要添加额外的 OCU 以支持这些集合。随着计算使用量的减少，容量会向下扩展。

有关如何为这些 OCU 计费的信息，请参阅the section called “ OpenSearch 无服务器定价”。

选择集合类型

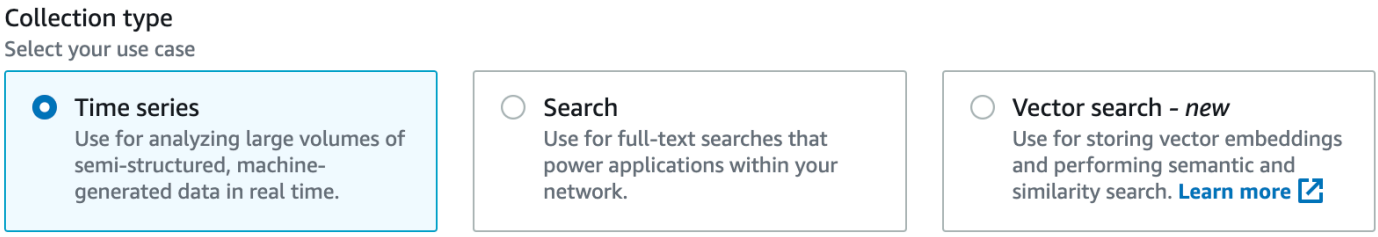
OpenSearch Serverless 支持三种主要的集合类型：

Time series（时间序列）：专注于实时分析机器生成的大量半结构化数据的日志分析部分，用于了解操作、安全性、用户行为和业务方面的情况。

Search（搜索）：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（如电子商务网站搜索和内容搜索）提供支持的全文搜索。

向量搜索 – 对向量嵌入执行语义搜索，用于简化向量数据管理，并为机器学习 (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）提供支持。

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 “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。

OpenSearch 无服务器定价

在 OpenSearch Serverless 中，您需要为以下组件付费：

- 数据摄取计算
- 搜索和查询计算
- 保留在 Amazon S3 中的存储

OCU 按小时计费，精确到秒。在您的账户对账单中，您会看到按 OCU 小时数计算的条目，其中包含用于数据摄取的标签和用于搜索的标签。您还需要按月为存储在 Amazon S3 中的数据付费。您无需为使用 OpenSearch 仪表板付费。

创建集合并启用冗余活动副本时，您需要支付至少 2 个 OCU [0.5 OCU x 2] 用于摄取和 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。所有后续集合均可共享这些 OCU。

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 为增量添加额外的 OCU。您可以为自己的账户配置 OCU 的最大数量，以控制成本。

> **Note**
>
> 具有唯一性的收藏 AWS KMS keys 无法与其他收藏共享 OCU。

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。在任何给定时间配置的 OCU 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 AWS 区域

OpenSearch 无服务器 AWS 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AWS 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 “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。
- 矢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 “创建、列出和删除集合”。
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	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。

功能	OpenSearch 服务	OpenSearch 无服务器
	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”。	有关更多信息，请参阅 the section called “管理容量限制”。
Billing	您需要按小时支付使用 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。 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 “ OpenSearch 无服务器定价”。
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”。
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 AWS 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”。
API	使用 OpenSearch 服务 API 操作以编程方式与OpenSearch 服务进行交互。	使用 OpenSearch 无服务器 API 操作以编程方式与OpenSearch 无服务器交互。

功能	OpenSearch 服务	OpenSearch 无服务器
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”。
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。
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 提供者，然后将关联的用户和组 ID 包括在数据访问策略中。 有关更多信息，请参阅 the section called “SAML 身份验证”。

功能	OpenSearch 服务	OpenSearch 无服务器
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。

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 “创建、列出和删除集合”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

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：

```
{
  "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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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” AWS 拥有的密钥。这是 OpenSearch Serv AWS 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，请同时选择 “启用对 OpenSearch 端点的访问” 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`。

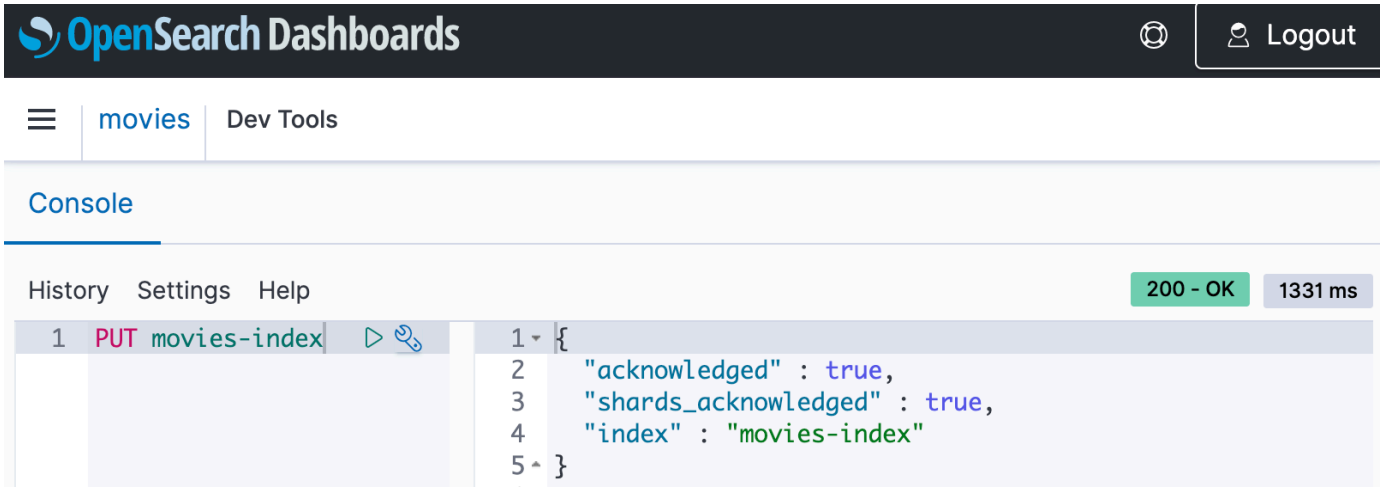
步骤 3：上传并搜索数据

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。
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 “创建、列出和删除集合”。
- 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 无服务器集合

您可以使用控制台、AWS CLI 和 API、AWS SDK 和 AWS CloudFormation 创建 Amazon OpenSearch 无服务器集合。

主题

- 创建、列出和删除 Amazon OpenSearch 无服务器产品系列
- 使用向量搜索集合
- 使用 Amazon OpenSearch 无服务器数据生命周期策略
- 使用 AWS SDK 与 Amazon OpenSearch 无服务器进行交互
- 使用 AWS CloudFormation 创建 Amazon OpenSearch 无服务器集合

创建、列出和删除 Amazon OpenSearch 无服务器产品系列

Amazon OpenSearch Serverless 中的集合是由一个或多个代表分析工作负载的索引组成的逻辑分组。OpenSearch Service 会自动管理和调整馆藏，只需要最少的手动输入。

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 Serverless 使用以下 AWS 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。 AWS 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. 导航到亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home/。
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：
   - 是您的账户所独有的 AWS 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：
   - 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。
   - 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。
   - 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. 在 “部署类型” 下，为您的集合选择冗余设置。默认情况下，每个集合都是以冗余方式创建的，这意味着索引和搜索 OpenSearch 计算单元 (OCU) 在不同的可用区中都有自己的备用副本。出于开发和测试目的，您可以选择禁用冗余，这样可以将集合中的 OCU 数量减少到两个。有关更多信息，请参阅 the section called “工作方式”。
7. 在 “加密” 下，选择用于加密数据的密 AWS KMS 钥。 OpenSearch 如果您输入的集合名称与加密策略中定义的模式匹配，Serverless 会通知您。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”。
8. 在 Network access settings（网络访问设置）下，配置集合的网络访问权限。

   - 对于访问类型，选择公共或私有。然后，指定哪些 VPC 终端节点 AWS 服务 可以访问该集合。
     - 用于访问的 VPC 终端节点-指定允许访问的一个或多个 VPC 终端节点。要创建 VPC 端点，请参阅the section called “VPC 端点”。
     - AWS 服务 私人访问-选择一个或多个支持的服务以允许访问。
   - 对于资源类型，选择是通过其OpenSearch端点（通过 curl、Postman 等进行 API 调用）、通过OpenSearch 仪表板端点（使用可视化效果并通过控制台进行 API 调用）还是同时通过两者访问集合。

   > **Note**
   >
   > AWS 服务 私有访问权限仅适用于 OpenSearch终端节点，不适用于 OpenSearch 仪表板端点。

   OpenSearch 如果您输入的集合名称与网络策略中定义的模式匹配，Serverless 会通知您。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”。
9. （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”。
10. 选择 下一步。
11. 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 “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匹配，Serverless 会通知您。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”。

12. 选择 下一步。
13. 在 Data access policy settings（数据访问策略设置）下，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。
14. 查看集合配置并选择 Submit（提交）。

在 OpenSearch Serverless 创建集合时，集合状态将更改`Creating`为。

创建集合（CLI）

在使用创建集合之前 AWS CLI，必须有一个加密策略，其资源模式必须与该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 AWS 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

使用创建收藏夹后 AWS Management Console，您可以导航到该集合的 OpenSearch 仪表板网址。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。

如果您已有 OpenSearch 仪表板 URL 可用但不在 AWS Management Console，则从浏览器调用仪表板 URL 将重定向到控制台。输入 AWS 凭据后，您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 仪表板。

OpenSearch 仪表板控制台超时为一小时，不可配置。

> **Note**
>
> 2023 年 5 月 10 日， OpenSearch 推出了 OpenSearch 仪表板的通用全局端点。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`。为了确保向后兼容，我们将继续使用以下格式支持现有集合特定的 OpenSearch 仪表板端点`https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/_dashboards`。

查看集合

您可以在亚马逊 OpenSearch 服务控制台的 “收藏夹” 选项卡 AWS 账户 上查看您的现有馆藏。

要列出收藏品及其ID，ListCollections请发送请求。

```
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（包括 OpenSearch 终端节点和 OpenSearch 仪表板端点）的更多详细信息，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 “收藏夹” 面板中，选择要删除的收藏夹。
2. 选择 Delete（删除），然后确认删除。

要使用删除收藏夹 AWS 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 OpenSearch API 操作。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。

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 AWS 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：

```
{
  "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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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 (OCU) 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。本教程使用开发工具。

索引和搜索“movies”（电影）集合中的数据

1. 要为您的新集合创建单一索引，请在开发工具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，请在开发工具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 OCU，请联系 Su AWS pp ort 以提高您的账户的最大 OpenSearch 计算单位 (OCU)。

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 AWS 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。

使用 Amazon OpenSearch 无服务器数据生命周期策略

Amazon OpenSearch 无服务器时间序列集合的数据生命周期策略决定了该集合中数据的保留时间。OpenSearch 无服务器会在您配置的时间段内保留数据。

您可以为 AWS 账户 中每个时间序列集合的每个索引配置单独的数据生命周期策略。OpenSearch 无服务器在索引中保留文档的时间至少与您在策略中配置的保留期限相同。然后，它会自动尽最大程度将其删除，通常在 48 小时内或保留期的 10% 时间范围内（以较长者为准）。

只有时间序列集合支持数据生命周期策略。搜索集合或向量搜索集合不支持。

主题

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 无服务器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 无服务器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
}
```

所需权限

OpenSearch 无服务器的生命周期策略使用以下 AWS 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：

```
{
    "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 无服务器未设置索引保留期。

```
{
      "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	资源名称和/或模式的列表。模式由前缀和通配符 (*) 组成，允许将关联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，OpenSearch 无服务器会无限期保留文档。

下面是一些示例：

```
{
  "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 (AWS CLI)

要使用 OpenSearch 无服务器 API 操作创建数据生命周期策略，请使用 CreateLifeCyclePolicy 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生命周期策略：

```
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}]}"
```

要在 JSON 文件中提供策略，请使用 `--policy file://my-policy.json` 格式

查看数据生命周期策略

在创建集合之前，您可能想预览账户中的现有数据生命周期策略，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 ListLifecyclePolicies 请求将列出您账户中的所有数据生命周期策略：

```
aws opensearchserverless list-lifecycle-policies --type retention
```

请求将返回有关所有已配置的数据生命周期策略的信息。要查看某一具体策略中定义的模式规则，请在响应的 `lifecyclePolicySummaries` 元素内容中查找策略信息。请注意此策略的 `name` 和 `type`，并使用 BatchGetLifecyclePolicy 请求中的这些属性来接收具有以下策略详细信息的响应：

```
{
    "lifecyclePolicySummaries": [
        {
            "type": "retention",
            "name": "my-policy",
            "policyVersion": "MTY2MzY5MTY1MDA3Ml8x",
            "createdDate": 1663691650072,
            "lastModifiedDate": 1663691650072
        }
```

```
    ]
}
```

要将结果限制为包含某些具体集合或索引的策略，您可以包含资源筛选器：

```
aws opensearchserverless list-lifecycle-policies --type retention --resources
 "index/autoparts-inventory/*"
```

要查看有关某个具体策略的详细信息，请使用 BatchGetLifecyclePolicy 命令。

更新数据生命周期策略

在修改数据生命周期策略时，所有关联集合都将受到影响。要在 OpenSearch 无服务器控制台中更新数据生命周期策略，请展开数据生命周期策略，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 更新数据生命周期策略，请使用 UpdateLifecyclePolicy 命令。您必须在请求中包括策略版本。您可以使用 `ListLifecyclePolicies` 或 `BatchGetLifecyclePolicy` 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 文档更新数据生命周期策略：

```
aws opensearchserverless update-lifecycle-policy \
    --name my-policy \
    --type retention \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

在更新策略与强制执行新保留期限之间，可能会有几分钟的延迟。

删除数据生命周期策略

删除数据生命周期策略后，该策略将不再适用于任何匹配的索引。要在 OpenSearch 无服务器控制台中删除策略，请选择该策略，然后选择 删除。

您也可以使用 DeleteLifecyclePolicy 命令：

```
aws opensearchserverless delete-lifecycle-policy --name my-policy --type retention
```

使用 AWS SDK 与 Amazon OpenSearch 无服务器进行交互

本节包含关于如何使用 AWS SDK 与 Amazon OpenSearch 无服务器进行交互的示例。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 AWS SDK for Python (Boto3) 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 中的 JavaScript 的 SDK 以及适用于 JavaScript 的 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。

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()
```

使用 AWS CloudFormation 创建 Amazon OpenSearch 无服务器集合

您可以使用 AWS CloudFormation 创建 Amazon OpenSearch 无服务器资源，如集合、安全策略和 VPC 端点。有关 OpenSearch 无服务器 CloudFormation 的全面参考资料，请参阅《AWS CloudFormation 用户指南》中的 Amazon OpenSearch 无服务器。

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 访问权限来保护您的集合。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

使用 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 (OCU) 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”

创建第一个集合时， OpenSearch Serverless 会实例化总共四个 OCU（两个用于索引，两个用于搜索）。即使在没有索引或搜索活动时，这些 OCU 也始终存在。所有后续集合都可以共享这些 OCU（具有唯一 AWS KMS 密钥的集合除外，这些集合会实例化自己的四个 OCU 集）。如果需要，随着索引和搜索使用量的增长， OpenSearch Serverless 会自动扩展并添加其他 OCU。当集合端点上的流量减少时，容量将缩减到数据大小所需的最少 OCU 数。它最多可以缩小到 1 个 OCU [0.5 OCU x 2] 用于索引，缩小到 1 个 OCU [0.5 OCU x 2] 用于搜索。

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**
>
> 矢量搜索集合无法与搜索集合和时间序列集合共享 OCU，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。将为您的第一个矢量集合创建一组新的 OCU。矢量集合的 OCU 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该数量的 OCU。
- 最大搜索容量 — OpenSearch Serverless 可以将搜索容量增加到这个数量的 OCU。

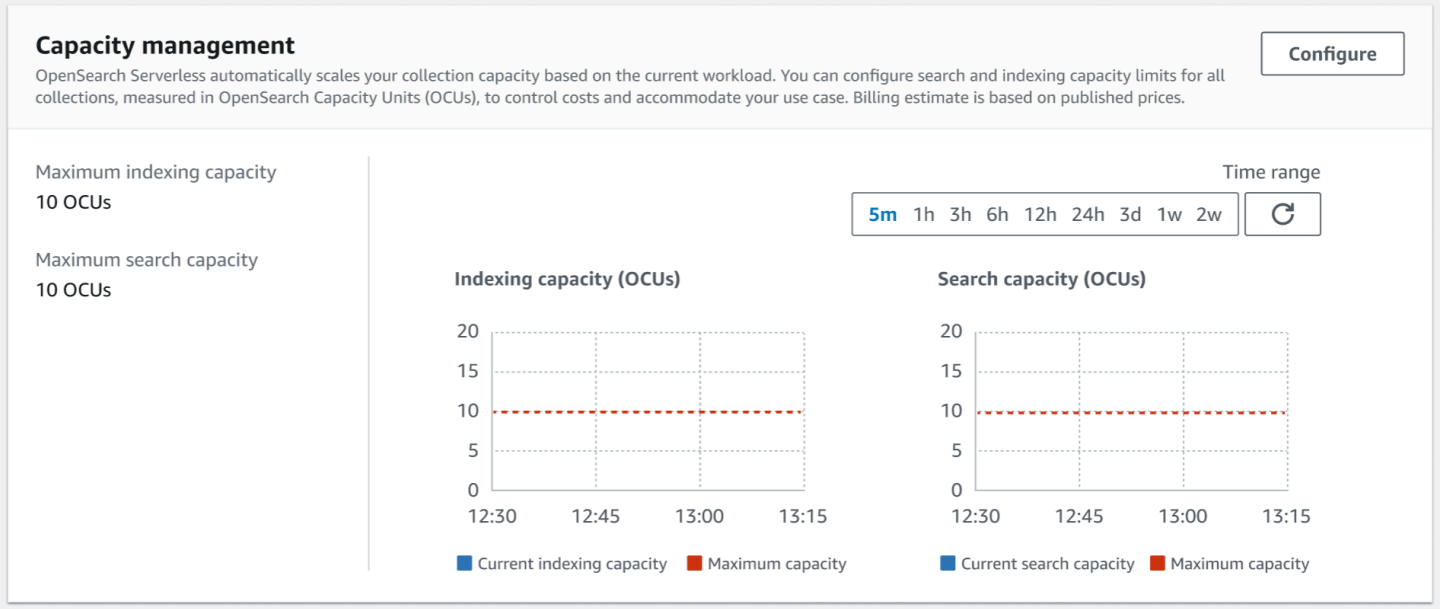
> **Note**
>
> 目前，容量设置仅应用于账户级。您无法配置每个集合的容量限制。

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 会自动扩展馆藏的 OCU 数量，以处理索引和搜索工作负载。

主题

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 AWS CLI，UpdateAccountSettings请发送请求：

```
aws opensearchserverless update-account-settings \
    --capacity-limits '{ "maxIndexingCapacityInOCU": 8,"maxSearchCapacityInOCU": 9 }'
```

最大容量限制

对于所有三种类型的集合，默认的最大容量为 10 个 OCU 用于索引，10 个 OCU 用于搜索。一个账户允许的最小容量为 1 个 OCU [0.5 OCU x 2] 用于索引，1 个 OCU [0.5 OCU x 2] 用于搜索。对于所有集合，最大允许容量为 200 个 OCU 用于编制索引，200 个 OCU 用于搜索。您可以将 OCU 计数配置为从 1 到最大允许容量之间的任意数字，以 2 的倍数表示。

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 10 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`Search0CU`和`Indexing0CU`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”。

将数据提取到 Ama OpenSearch zon 无服务器集合中

这些部分详细介绍了支持将数据提取到 Ama OpenSearch zon Serverless 集合中的采集管道。它们还涵盖了您可以用来与 OpenSearch API 操作进行交互的一些客户端。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 摄入
- Fluent Bit
- Amazon Data Firehose
- Fluentd
- Go

- Java
- JavaScript
- Logstash
- Python
- Ruby
- 与其他客户端签署 HTTP 请求

所需的最低权限

要将数据采集到 OpenSearch Serverless 集合中，写入数据的委托人必须具有在数据访问策略中分配的以下最低权限：

```
[
   {
      "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/*target-collection*/*) 或索引子集 (index/*target-collection*/*logs**) 的权限，而不是指定单个目标索引。

有关所有可用的 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 摄入

您可以使用 Amazon OpenSearch Ingestion，而不是使用第三方客户端将数据直接发送到 OpenSearch 无服务器集合。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。

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。有关使用 OpenSearch Ingestion 向集合添加数据的说明，请参阅。the section called “授予管道访问集合的权限”

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 AWS 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**
>
> 你必须拥有 for Fluent Bit 镜像的 2.30.0 或更高版本才能与 Serverless 集成。 AWS OpenSearch

示例配置：

配置文件的输出示例部分显示了如何使用 OpenSearch Serverless 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 无服务器作为送货目的地。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 以供交付的 IAM 角色必须在数据访问策略中指定，并且必须具有目标集合`aoss:WriteDocument`的最低权限，并且您必须有一个预先存在的索引才能向其发送数据。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。

Fluentd

您可以使用 Fluentd OpenSearch 插件从基础架构、容器和网络设备收集数据，然后将其发送到 OpenSearch 无服务器集合。Calyptia 维护 Fluentd 的一个发行版，其中包含 Ruby 和 SSL 的所有下游依赖项。

使用 Fluentd 向无服务器发送数据 OpenSearch

1. 从 https://www.fluentd.org/download 下载 Calyptia Fluentd 的版本 1.4.2 或更高版本。此版本默认包含 OpenSearch 插件，该插件支持 OpenSearch 无服务器。
2. 安装 软件包。根据您的操作系统，请按照 Fluentd 文档中的说明进行操作：

   - Red Hat Enterprise Linux/CentOS/Amazon Linux
   - Debian/Ubuntu
   - Windows
   - MacOSX
3. 添加将数据发送到 OpenSearch 无服务器的配置。此示例配置会将消息“test”（测试）发送到单个集合。务必执行以下操作：

   - 对于`host`，请指定您的 OpenSearch 无服务器集合的终端节点。
   - 对于 `aws_service_name`，请指定 `aoss`。

```
<source>
@type sample
tag test
test {"hello":"world"}
</source>

<match test>
@type opensearch
host https://collection-endpoint.us-east-1.aoss.amazonaws.com
port 443
index_name fluentd
aws_service_name aoss
</match>
```

4. 运行 Calyptia Fluentd，以开始将数据发送到该集合。例如，在 Mac 上，您可以运行以下命令：

```
sudo launchctl load /Library/LaunchDaemons/calyptia-fluentd.plist
```

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 Serverless 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-j ava 客户端与指定的 OpenSearch Serverless 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();
```

以下示例代码再次建立安全连接，然后搜索索引。

```
import org.opensearch.client.opensearch.OpenSearchClient;

SdkHttpClient httpClient = ApacheHttpClient.builder().build();

OpenSearchClient client = new OpenSearchClient(
    new AwsSdk2Transport(
        httpClient,
        "...us-west-2.aoss.amazonaws.com", // serverless collection endpoint
        "aoss" // signing service name
        Region.US_WEST_2, // signing service region
        AwsSdk2TransportOptions.builder().build()
    )
```

```
);

Response response = client.generic()
    .execute(
        Requests.builder()
            .endpoint("/" + "users" + "/_search?typed_keys=true")
            .method("GET")
            .json("{"
                + "    \"query\": {"
                + "        \"match_all\": {}"
                + "    }"
                + "}")
            .build());

httpClient.close();
```

JavaScript

以下示例代码使用的 opensearch-js 客户端与指定的 S OpenSearch erverless 集合建立安全连接、创建单个索引、添加文档和删除索引。 JavaScript 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

此示例使用了 Node.js JavaScript 中开发工具包的版本 3。

```
const { defaultProvider } = require('@aws-sdk/credential-provider-node');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () => {
                const credentialsProvider = defaultProvider();
                return credentialsProvider();
            },
        }),
        node: '' # // serverless collection endpoint
```

```
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index }, )).body) {
        console.log((await client.indices.create({ index })).body);
    }

    // add a document to the index
    const document = { foo: 'bar' };
    const response = await client.index({
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}

main();
```

Version 2

此示例使用了 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。

使用 Logstash 向无服务器发送数据 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 托管 Logstash OSS 软件，预装了 OpenSearch 输出插件：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
```

从插件的 2.0.0 版本开始， AWS 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 AWS 插件并安装该插件：`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. 为了使 OpenSearch 输出插件与 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`。这些参数是插件与 OpenSearch Serverless 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 存储桶中的文件中获取输入并将其发送到 OpenSearch 无服务器集合：

```
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 客户端与指定的 OpenSearch Serverless 集合建立安全连接，创建单个索引并搜索该索引。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-sigv4Gem 开箱即用地提供对 OpenSearch 无服务器和 OpenSearch 服务的访问权限。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

当您与其他客户端构建 HTTP 请求时，对 OpenSearch 无服务器集合的请求进行签名时，以下要求适用。

- 必须将服务名称指定为 aoss。
- 所有 AWS 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 进行索引
- 使用 Postman 编制索引

使用 cURL 进行索引

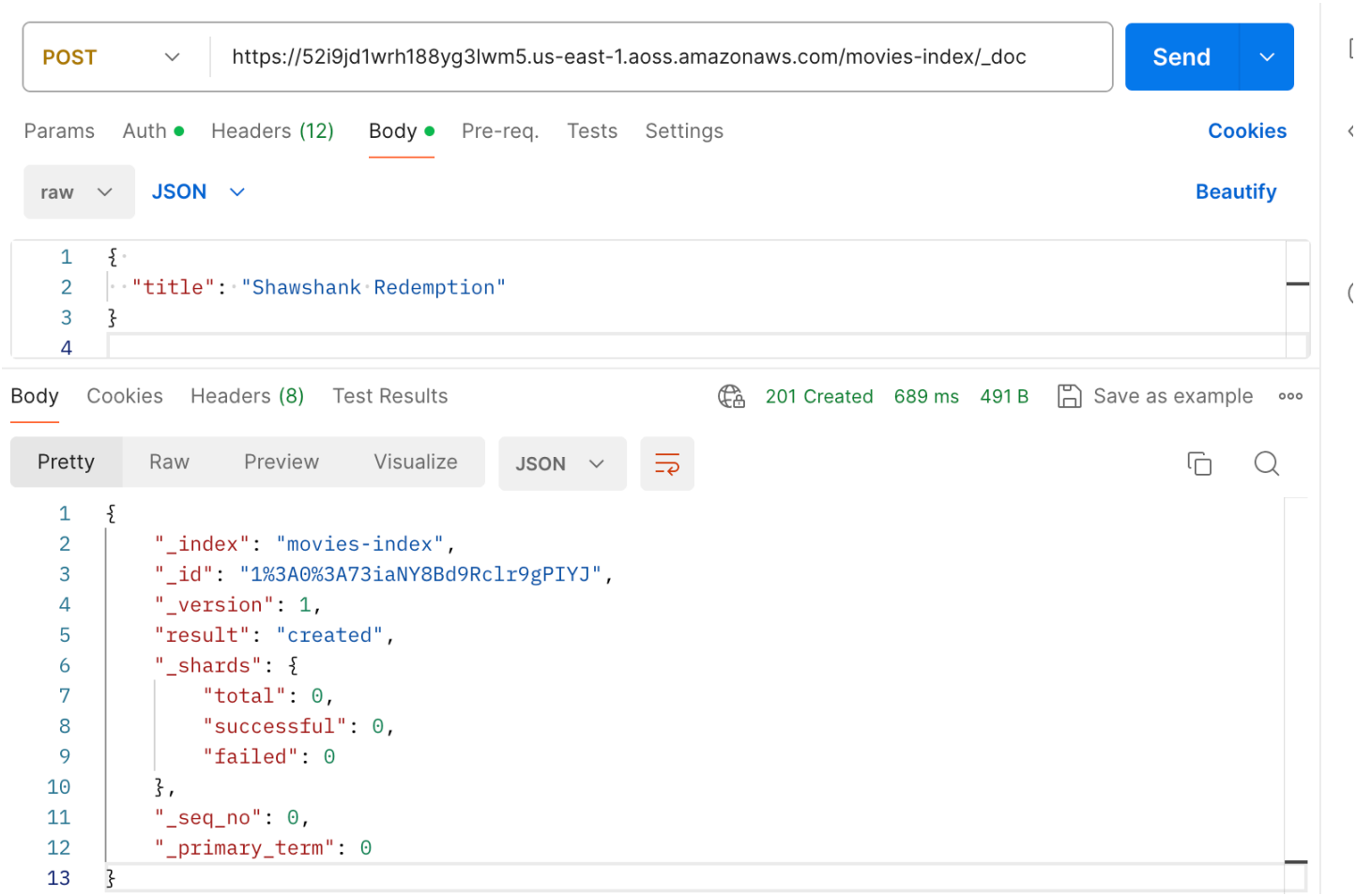
以下示例请求使用客户端 URL 请求库 (cURL) 将单个文档发送到集合movies-index中名为的索引：

```
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 编制索引

下图显示了如何使用 Postman 向集合发送请求。有关身份验证的说明，请参阅 P ostman 中的 “使用 AWS 签名进行身份验证” 身份验证工作流程。

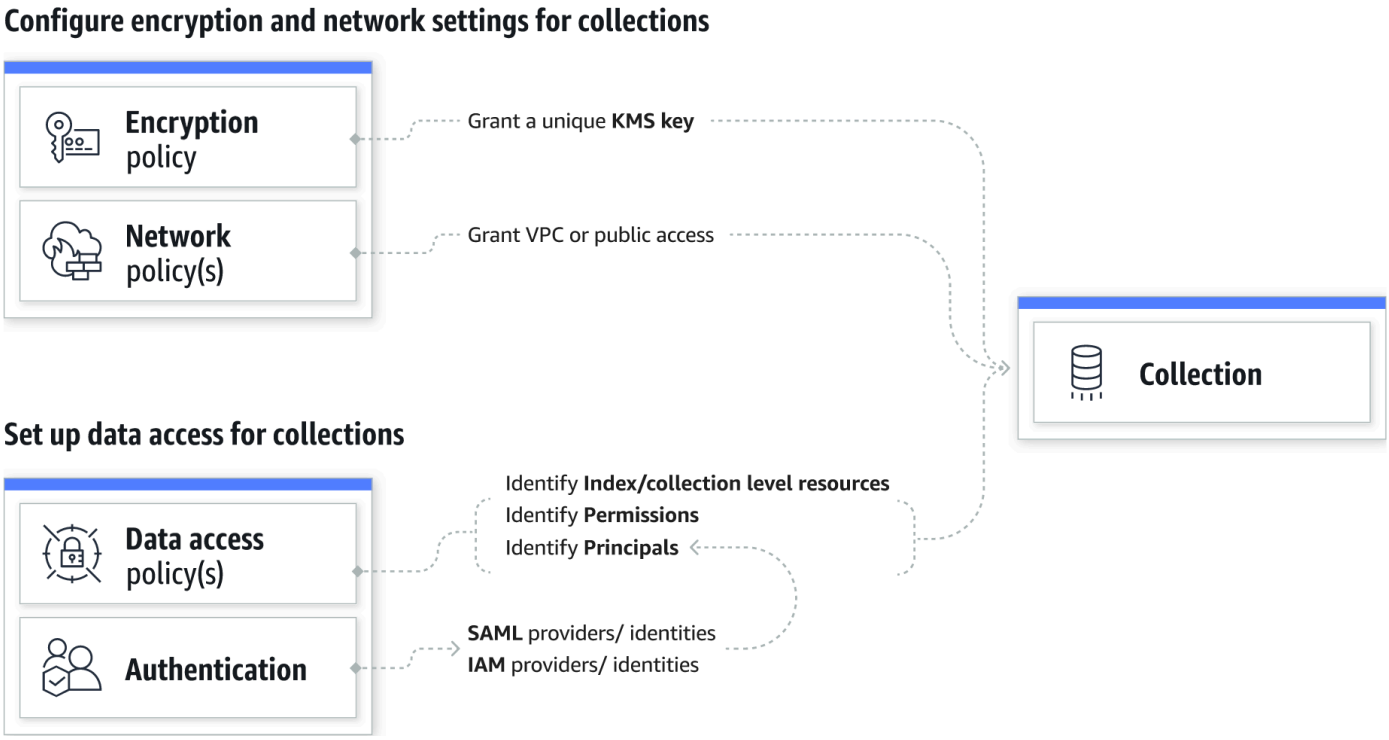


Amazon OpenSearch Serverless 中的安全概述

Amazon OpenSearch Serverless 中的安全与亚马逊 OpenSearch 服务中的安全有根本的区别，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

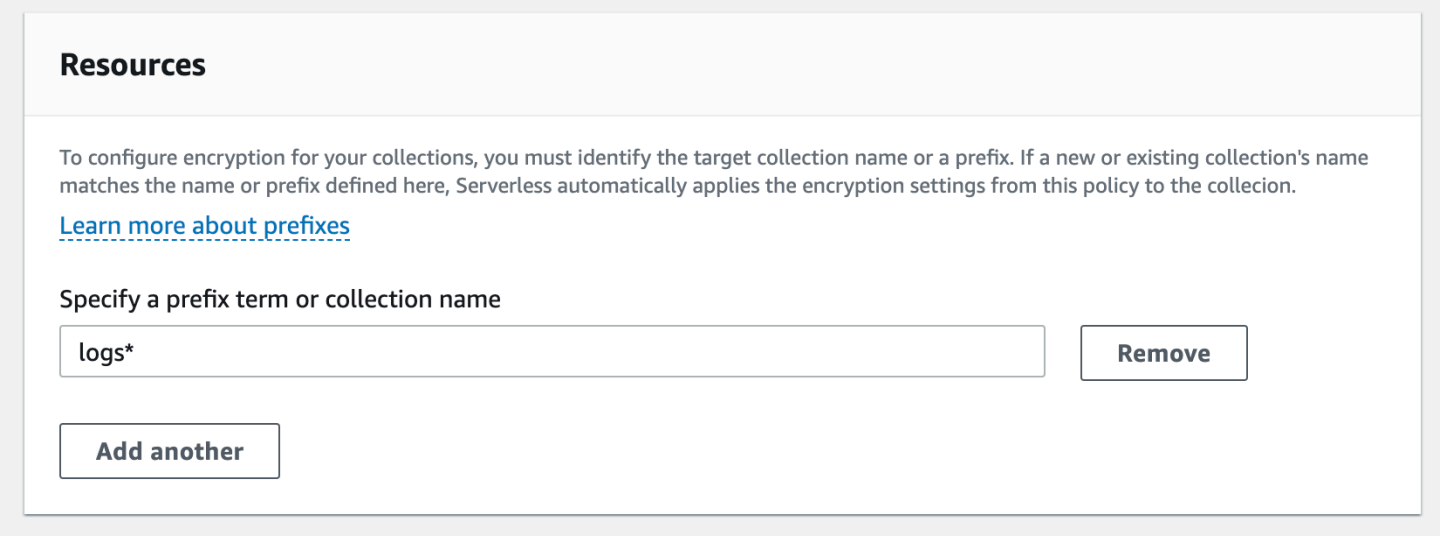
- 加密策略
- 网络策略
- 数据访问策略
- IAM 和 SAML 身份验证
- 基础设施安全性

- Amazon OpenSearch Serverless 中的安全入门
- 适用于 Amazon OpenSearch 无服务器的 Identity and Access Management
- 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 ()AWS PrivateLink
- 适用于 Amazon 无服务器的 SAM OpenSearch L 身份验证
- Amazon OpenSearch Serverless 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。 AWS 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。



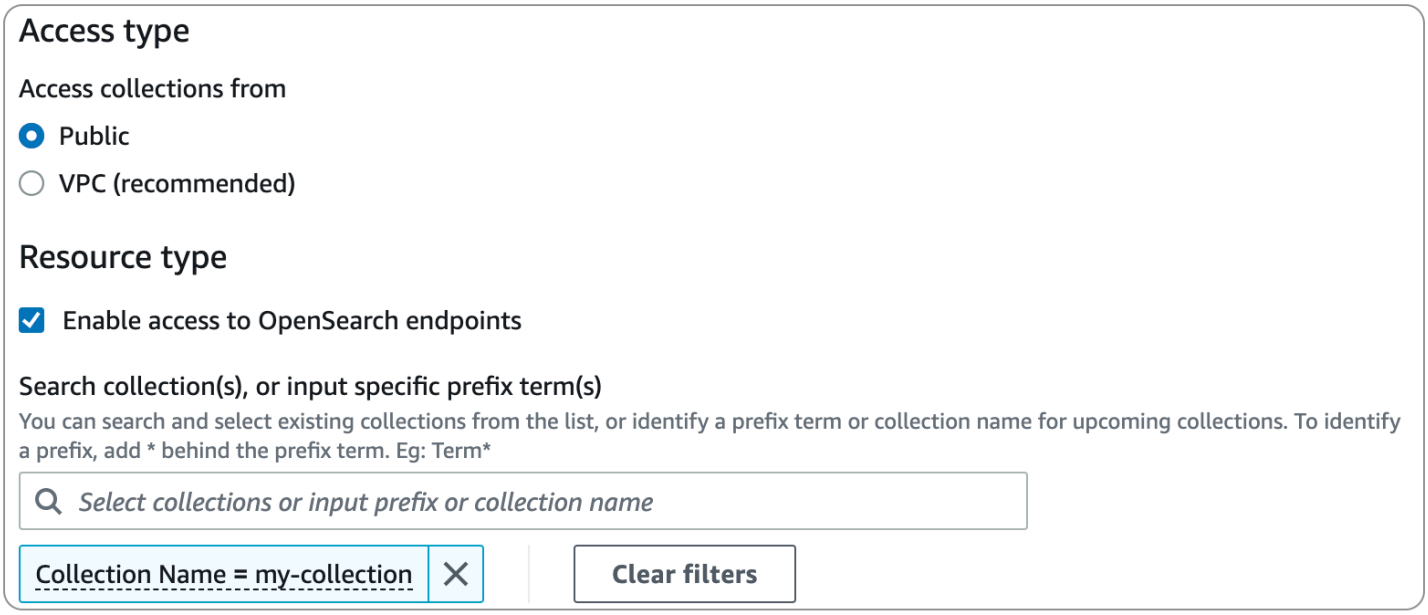
加密策略可以简化创建和管理集合的过程，尤其是当您以编程方式创建和管理集合时。您只需指定名称即可创建集合，并在创建时自动为其分配加密密钥。

网络策略

网络策略定义了您的馆藏是可以私下访问，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 AWS 服务

之类的 AWS 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。访问类型可以是公共的，也可以是私有的。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。



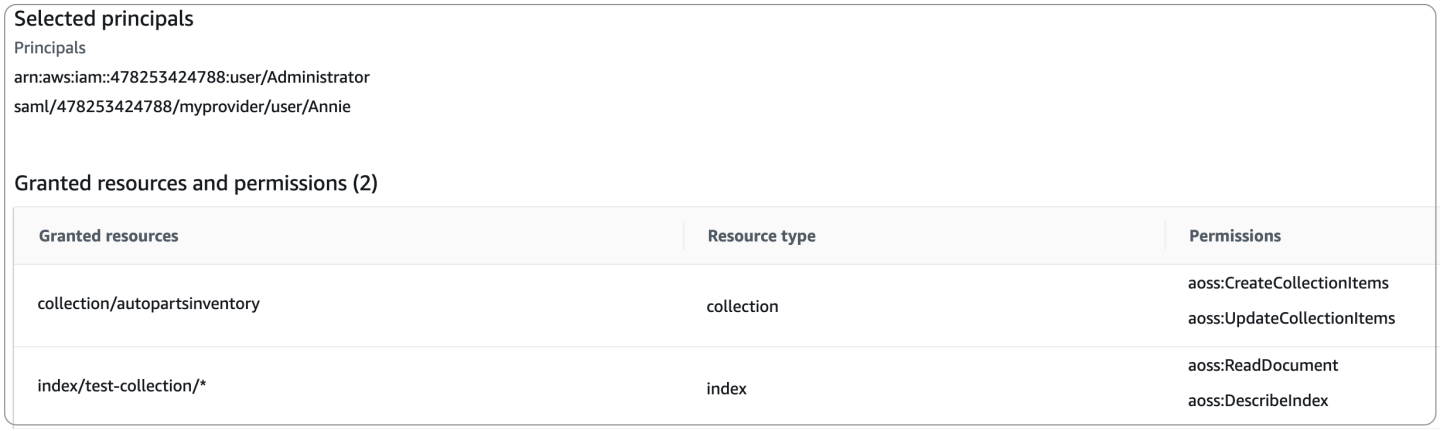
如果您计划在网络策略中配置 VPC 访问权限，则必须先创建一个或多个OpenSearch 无服务器托管的 VPC 终端节点。这些终端节点允许您像访问您的 VPC 一样访问 OpenSearch 无服务器，无需使用互联网网关、NAT 设备、VPN 连接或 AWS Direct Connect 连接。

对的私有访问权限 AWS 服务 只能应用于集合的 OpenSearch 端点，不能应用于 OpenSearch 仪表板端点。 AWS 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。授予的资源可以是集合/索引名称或带有通配符 (*) 的模式。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。



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。

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”。

基础设施安全性

Amazon OpenSearch Serverless 受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 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 Elastic Load Balan cing 文档中的 TLS 协议和密码。

此外，您必须使用访问密钥 ID 和与 IAM 委托人关联的私有访问密钥签署请求。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

Amazon OpenSearch Serverless 中的安全入门

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 AWS 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

本教程将指导您完成使用AWS Management Console设置集合的步骤。有关使用 AWS CLI 的相同步骤，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：

```
{
  "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 AWS KMS OpenSearch Serverless 用于加密集合的密钥。您可以使用 AWS 托管式密钥或其他密钥对集合进行加密。在本教程中，为简单起见，我们将使用 AWS 托管式密钥对集合进行加密。

创建加密策略

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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，保持选中使用 AWS 拥有的密钥。
7. 选择创建。

步骤 3：创建网络策略

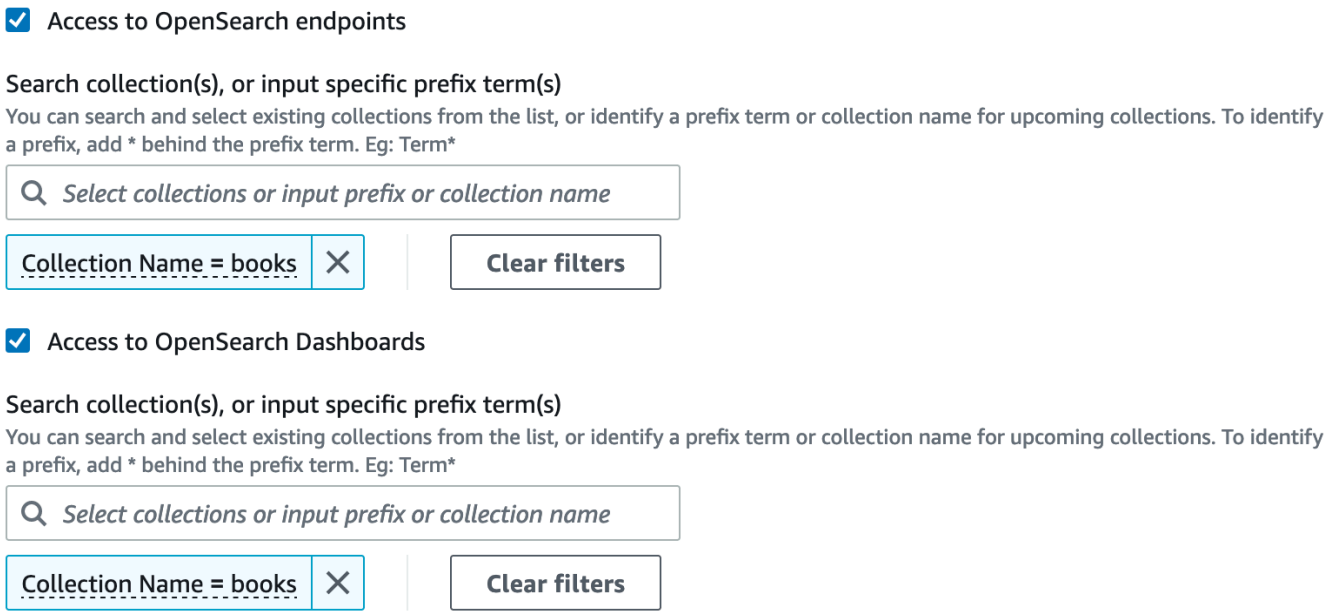
网络策略决定您的馆藏是否可通过互联网从公共网络访问，或者是否必须通过 OpenSearch 无服务器托管的 VPC 终端节点进行访问。在本教程中，我们将配置公共访问权限。

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 OpenSearch端点访问权限和 OpenSearch控制面板访问权限。

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. 请选择 Next（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. 请选择 Next（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。

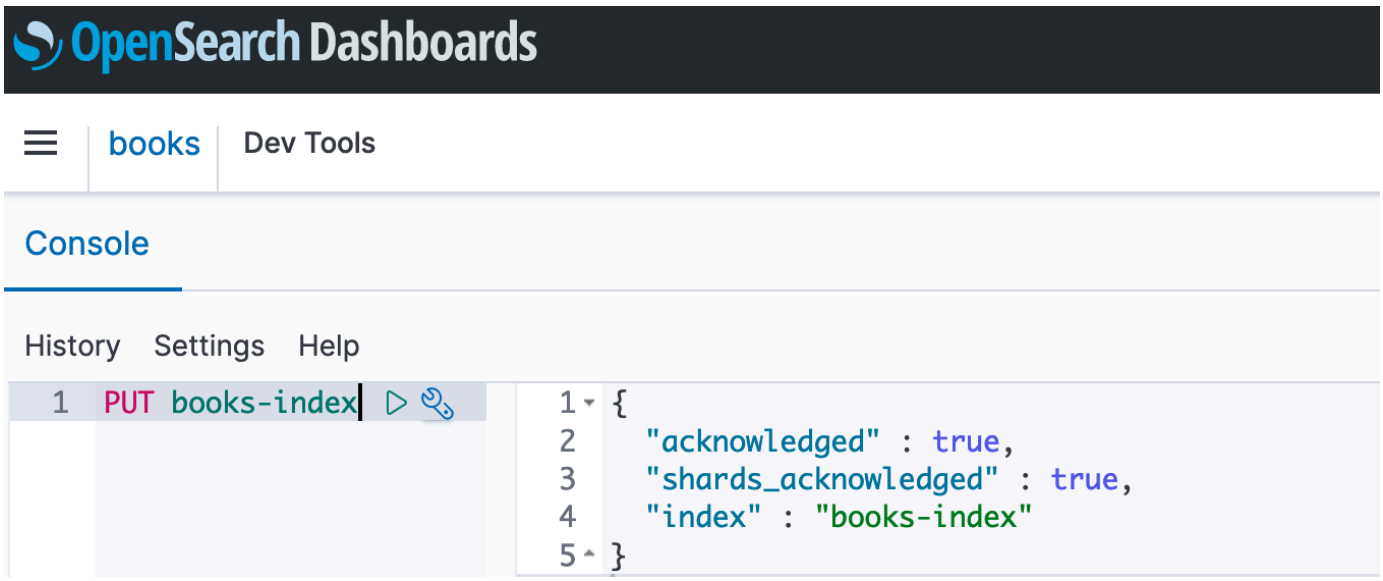
步骤 6：上传和搜索数据

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AWS访问策略中指定的委托人的访问权限和密钥登录控制面 OpenSearch 板。
4. 在 OpenSearch 仪表板中，打开左侧导航菜单并选择开发工具。
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AWS 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、用于加密的 AWS 托管式密钥，以及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 AWS Identity and Access Management 策略，该策略拥有执行本教程中的步骤所需的最低权限。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 AWS 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 无服务器的 Identity and Access Management

AWS Identity and Access Management (IAM) 是一项 AWS 服务，可以帮助管理员安全地控制对 AWS 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）以及获得授权（拥有权限）使用 OpenSearch 无服务器资源。IAM 是一项无需额外费用即可使用的 AWS 服务。

主题

- 适用于 OpenSearch 无服务器的基于身份的策略
- 适用于 OpenSearch 无服务器的策略操作

- 适用于 OpenSearch 无服务器的策略资源
- 适用于 Amazon OpenSearch 无服务器的策略条件键
- 包含 OpenSearch 无服务器的 ABAC
- 配合使用临时凭证与 OpenSearch 无服务器
- 适用于 OpenSearch 无服务器的服务相关角色
- 适用于 OpenSearch 无服务器的基于身份的策略示例

适用于 OpenSearch 无服务器的基于身份的策略

支持基于身份的策略	可以

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》中的创建 IAM 策略。

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。

适用于 OpenSearch 无服务器的基于身份的策略示例

要查看 OpenSearch 无服务器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”。

适用于 OpenSearch 无服务器的策略操作

支持策略操作	可以

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。策略操作通常与相关联的 AWS 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 无服务器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 OpenSearch 无服务器基于身份的策略的示例，请参阅 适用于 OpenSearch 无服务器的基于身份的策略示例。

适用于 OpenSearch 无服务器的策略资源

支持策略资源	可以

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。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": "*"
```

适用于 Amazon OpenSearch 无服务器的策略条件键

支持特定于服务的策略条件键	可以

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。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，您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 元素中指定多个密钥，则 AWS 使用逻辑 AND 运算评估它们。如果您要为单个条件密钥指定多个值，则 AWS 使用逻辑 OR 运算来评估条件。在授予语句的权限之前必须满足所有的条件。

您也可以在指定条件时使用占位符变量。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 policy 元素：变量和标签。

AWS 支持全局条件键和特定于服务的条件键。要查看所有 AWS 全局条件键，请参阅《IAM 用户指南》中的 AWS 全局条件上下文键。

除了基于属性的访问权限控制（ABAC）外，OpenSearch 无服务器还支持以下条件键：

- 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 无服务器条件键的列表，请参阅《服务授权参考》中的适用于 Amazon OpenSearch 无服务器的条件键。要了解您可以对哪些操作和资源使用条件键，请参阅 Amazon OpenSearch 无服务器定义的操作。

包含 OpenSearch 无服务器的 ABAC

支持 ABAC（策略中的标签）	可以

基于属性的访问权限控制 (ABAC) 是一种授权策略，该策略基于属性来定义权限。在 AWS 中，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）以及 AWS 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，需要使用 `aws:ResourceTag/`*`key-name`*、`aws:RequestTag/`*`key-name`* 或 `aws:TagKeys` 条件密钥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》中的什么是 ABAC？。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”。

配合使用临时凭证与 OpenSearch 无服务器

支持临时凭证	可以

某些 AWS 服务 在使用临时凭证登录时无法正常工作。有关更多信息，包括 AWS 服务 与临时凭证配合使用，请参阅《IAM 用户指南》中的使用 IAM 的 AWS 服务。

如果您不使用用户名和密码而用其他方法登录到 AWS Management Console，则使用临时凭证。例如，当您使用贵公司的单点登录（SSO）链接访问 AWS 时，该过程将自动创建临时凭证。当您以用户身份登录控制台，然后切换角色时，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》中的切换到角色（控制台）。

您可以使用 AWS CLI 或者 AWS API 创建临时凭证。之后，您可以使用这些临时凭证访问 AWS。AWS 建议您动态生成临时凭证，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。

适用于 OpenSearch 无服务器的服务相关角色

支持服务相关角色	可以

服务相关角色是一种与 AWS 服务 相关的服务角色。服务可以担任代表您执行操作的角色。服务相关角色显示在您的 AWS 账户中，并由该服务拥有。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”。

适用于 OpenSearch 无服务器的基于身份的策略示例

默认情况下，用户和角色没有创建或修改 OpenSearch 无服务器资源的权限。他们也无法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS 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 policy。然后，管理员可以向角色添加 IAM policy，并且用户可以代入角色。

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 policy。

有关 Amazon OpenSearch 无服务器定义的操作和资源类型的详细信息，包括每种资源类型的 ARN 格式，请参阅《服务授权参考》中的适用于 Amazon OpenSearch 无服务器的操作、资源和条件键。

主题

- 策略最佳实操
- 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

策略最佳实操

基于身份的策略非常强大。它们确定某人是否可以创建、访问或删除您账户中的 OpenSearch 无服务器资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 OpenSearch 无服务器资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- AWS 托管式策略及转向最低权限许可入门 – 要开始向用户和工作负载授予权限，请使用 AWS 托管式策略来为许多常见使用场景授予权限。您可以在 AWS 账户 中找到这些策略。建议通过定义特定于您的使用场景的 AWS 客户管理型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AWS 托管式策略或工作职能的 AWS 托管式策略。
- 应用最低权限 – 在使用 IAM policy 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 policy 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。如果通过特定 AWS 服务（例如 AWS CloudFormation）使用服务操作，您还可以使用条件来授予对服务操作的访问权限。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 policy，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 policy 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，有助于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 IAM Access Analyzer 策略验证。

- 需要多重身份验证（MFA）– 如果您所处的场景要求您的 AWS 账户 中有 IAM 用户或根用户，请启用 MFA 来提高安全性。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》中的配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实践的更多信息，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 Service 控制台中访问 OpenSearch 无服务器，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的 AWS 账户中的 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。

对于只需要调用 AWS CLI 或 AWS API 的用户，无需为其提供最低控制台权限。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 Service 控制台中访问 OpenSearch 无服务器：

```
{
    "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

此策略是“集合管理员”策略的示例，该策略允许用户管理和支配 Amazon OpenSearch 无服务器集合。用户可以创建、查看和删除集合。

```
{
    "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 无服务器集合的详细信息。用户无法修改集合或任何相关联的安全策略。

```
{
    "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

数据面板 API 操作由您在 OpenSearch 无服务器中使用的函数组成，这些函数用于从服务中获取实时值。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 无服务器数据面板 API 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 无服务器需要这两个新 IAM 权限才能使用集合资源。`aoss:APIAccessAll` 权限允许通过浏览器访问数据面板，`aoss:DashboardsAccessAll` 权限允许通过浏览器访问 OpenSearch 控制面板。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面板 API，以及访问其账户中所有集合的 OpenSearch 控制面板。

```
{
    "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"
```

```
        }
    ]
}
```

aoss:APIAccessAll 和 aoss:DashboardsAccessAll 授予对集合资源的完整 IAM 权限，而控制面板权限还提供了 OpenSearch 控制面板访问权限。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。拒绝 aoss:DashboardsAccessAll 也同样如此。

OpenSearch 无服务器仅支持主体的 IAM policy 中条件设置中的源 IP 地址用于数据面板调用：

```
"Condition": {
    "IpAddress": {
         "aws:SourceIp": "52.95.4.14"
    }
}
```

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 AWS Key Management Service (AWS 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。

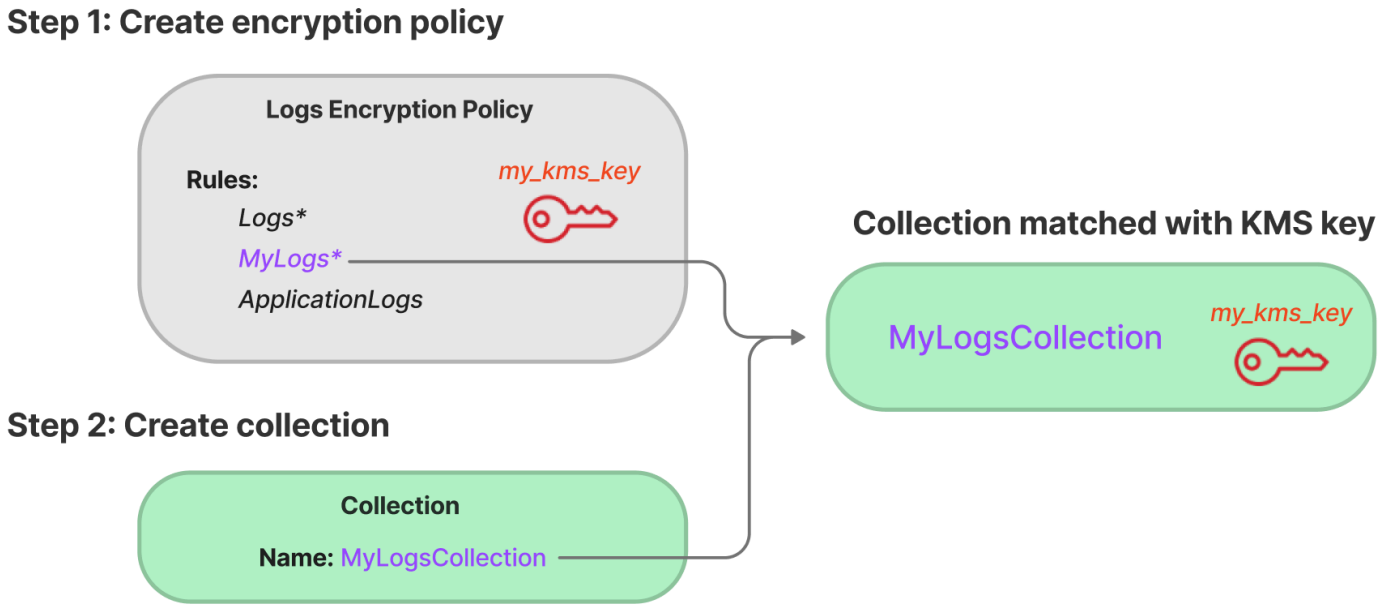
主题

- 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 AWS KMS
- 创建加密策略（控制台）
- 创建加密策略（AWS 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 AWS 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 **-< ####** > 的加密策略并将其附加到集合。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 AWS 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 AWS 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 (OCU)。每个具有唯一密钥的集合都需要其自己的 4 个 OCU。
- 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 AWS 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。

```
{
   "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：

```
{
  "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 AWS Key Management Service 开发人员指南中的在 AWS KMS中使用密钥策略。

OpenSearch 无服务器如何使用授权 AWS 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 AWS KMS代表您创建授权。中的授权 AWS 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 AWS 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 AWS 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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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 AWS 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（AWS 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 Serverless 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 API 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。

私人访问权限可以应用于以下一项或两项：

- OpenSearch 无服务器托管的 VPC 终端节点
- 支持， AWS 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。如果他们尝试直接从公共网络调用 OpenSearch API，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

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，第一条规则为以该术语开头的所有集合指定了对集合终端节点和仪表板终端节点的 VPC 终端节点的访问权限marketing*。它还指定了 Amazon Bedrock 访问权限。

> **Note**
>
> AWS 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 AWS 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 AWS 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 AWS 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。 AWS 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 AWS 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：

```
{
    "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，指定公共访问权限的规则将优先于为两个规则共用的任何集合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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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）。

元素	描述
Access type（访问类型）	选择公共访问或私有访问权限。然后，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。 • AWS 服务 私人访问权限-选择一个或多个支持的访问权限 AWS 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（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** AWS 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 AWS 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，您可以选择现有集合以将策略设置应用于它，和/或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（AWS 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。

要指定私有访问权限，请包括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 AWS 服务。当前，支持以下服务标识符：
  - `bedrock.amazonaws.com`— Amazon Bedrock

以下示例网络策略仅为以前缀`log*`开头的集合提供对 VPC 终端节点和 Amazon Bedrock 以及对收集端点的私有访问权限。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（AWS 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 AWS 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：

```
{
    "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需的 IAM 权限

OpenSearch Serverless 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。

```
{
    "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`	资源名称和/或模式的列表。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 “支持 OpenSearch 的 API 操作和权限”。

元素	描述
Principal	要向其授予访问权限的一个或多个主体的列表。主体可以是 IAM 角色 ARN，也可以是 SAML 身份。这些主体必须位于当前 AWS 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，由其他 AWS 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 “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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
2. 在左侧导航窗格中，展开 Serverless（无服务器），然后选择 Data access control（数据访问控制）。
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 “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 AWS Support。

> **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（AWS 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 MTY2NDA1NDE4MDg1OF8x \
    --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 Serverless 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

虽然您无法使用跨账户身份或跨账户集合创建数据访问策略，但您仍然可以使用代入角色选项设置跨账户访问权限。例如，如果*account-a*拥有一个*account-b*需要访问的集合，则来自的用户*account-b*可以在中扮演角色*account-a*。该角色必须具有 IAM 权限aoss:APIAccessAll和aoss:DashboardsAccessAll，并包含在上的数据访问策略中*account-a*。

使用接口终端节点访问 Amazon OpenSearch Serverless ()AWS PrivateLink

您可以使用 AWS 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 AWS 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。

您可以通过创建由 AWS 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《AWS PrivateLink 指南》中的通过 AWS PrivateLink访问 AWS 服务。

主题

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 AWS 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。没有 OpenSearch 无服务器 VPC 终端节点的 VPC 中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）是根据您在为无服务器创建接口终端节点时提供的子网确定的。 OpenSearch

> **Note**
>
> 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。 AWS 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。

VPC 和网络访问策略

要向您的集合 OpenSearch 的 API 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中指定多个 VPC 端点 ID，因此我们建议您为每个需要访问集合的 VPC 创建一个 VPC 端点。这些 VPC 可以属于与拥有 Serv OpenSearch erless 集合和网络策略的 AWS 账户不同的账户。我们建议您不要在两个账户之间创建 VPC 到 VPC 对等或其他代理解决方案，这样一个账户的 VPC 可以使用另一个账户的 VPC 端点。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。

VPC 和端点策略

Amazon OpenSearch Serverless 支持 VPC 的终端节点策略。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 AWS 委托人可以使用该终端节点访问您的 AWS 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（ AWS 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：

```
{
    "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：

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

```
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CollectionName": [
                        "coll-abc"
                    ]
                }
            }
        }
    ]
}
```

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：

```
{
    "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：

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SamlPrincipal": [
                        "saml/123456789012/idp123/user/user1234"]
                }
            }
        }
    ]
}
```

有关在 Amazon Serverless 中使用 SAML 身份验证的更多信息，请参阅亚马逊 OpenSearch 无服务器的 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：

```
{
    "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"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": [
                    "123456789012"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

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 AWS 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 AWS 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 AWS 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 端点。

- 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 OpenSearch 无服务器 API 为无服务器创建接口端点。 OpenSearch

为 OpenSearch 无服务器集合创建接口端点

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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。
   - 终端节点的 IP 地址和 DNS 类型取决于子网类型
     - Dualstack：如果所有子网都有 IPv4 和 IPv6 地址范围
     - IPv6：如果所有子网都是仅限 IPv6 的子网
     - 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-050f79086ee71ac05`）。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。

下一步：为端点授予针对集合的访问权限

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”。

适用于 Amazon 无服务器的 SAM OpenSearch L 身份验证

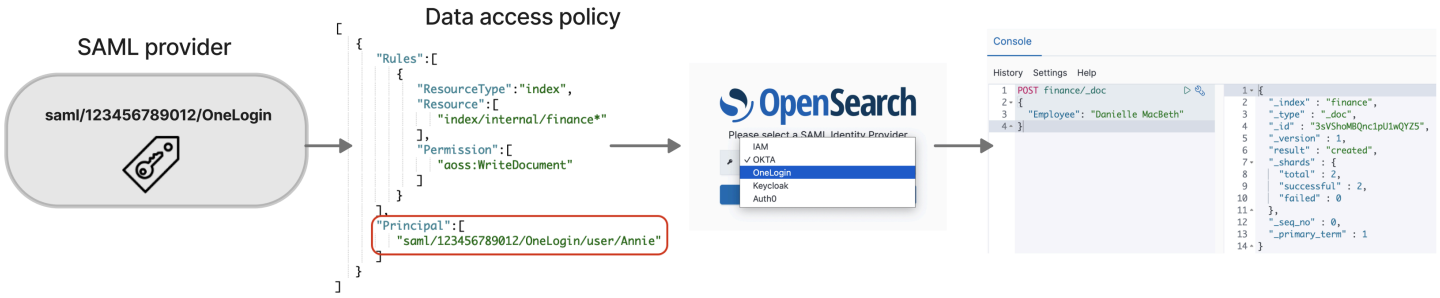
借助适用于 Amazon OpenSearch Serverless 的 SAML 身份验证，您可以使用现有的身份提供商为无服务器集合的 OpenSearch 控制面板终端节点提供单点登录 (SSO)。

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 支持的身份源列表及其配置步骤，请参阅 I AM Identity Center 用户指南中的入门教程。

> **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。此策略将授予其针对集合和/或索引的某些权限。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（AWS 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。

所需权限

OpenSearch 无服务器的 SAML 身份验证使用以下 AWS 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：

```
{
    "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. 登录亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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 Allowed Callback URLs（允许的回调 URL）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 AWS 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 AWS 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，然后在 `Principal` 语句中指定您的 SAML 用户和/或组 ID：

```
[
   {
      "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（AWS 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 Serverless 合规性验证

作为多项合规计划的一部分，第三方审计机构对 Amazon OpenSearch Serverless 的安全 AWS 性和合规性进行评估。这些计划包括 SOC、PCI 和 HIPAA。

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 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。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了在这些基础上 AWS 部署以安全性和合规性为重点的基准环境的步骤。
- 在 A@@ mazon Web Services 上构建 HIPAA 安全与合规架构 — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

> **Note**
>
> 并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅符合 HIPAA 要求的服务参考。

- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS 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。
- AWS Security Hub— 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控件评估您的 AWS 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件的列表，请参阅 Security Hub 控件参考。
- Amazon GuardDuty — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 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。
- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

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 无服务器集合，以便您可以对该信息进行分类和筛选。标签是您或 AWS 为 AWS 资源分配的元数据标记。

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`。

借助标签，您可以执行以下操作：

- 标识和整理您的 AWS 资源。许多 AWS 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 OpenSearch 无服务器集合。
- 跟踪您的 AWS 成本。您可以在 AWS Billing and Cost Management 控制面板上激活这些标签。AWS 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 AWS Billing 用户指南中的使用成本分配标签。

在 OpenSearch 无服务器中，主要资源是一个集合。您可以使用 OpenSearch Service 控制台、AWS CLI、OpenSearch 无服务器 API 操作或 AWS SDK 在集合中添加、管理和移除标签。

所需权限

OpenSearch 无服务器使用以下 AWS Identity and Access Management Access Analyzer（IAM）权限来标记集合：

- `aoss:TagResource`
- `aoss:ListTagsForResource`
- `aoss:UntagResource`

使用标签 (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. 登录到位于 https://console.aws.amazon.com/aos/home 的 Amazon OpenSearch Service 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 Save（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《AWS 管理控制台入门指南》中的标签编辑器。

使用标签 (AWS CLI)

要使用 AWS CLI 标记集合，请发送 TagResource 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
```

可以借助 ListTagsForResource 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
```

可以使用 UntagResource 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 Serverless 中支持的操作和插件

Amazon OpenSearch Serverless 支持各种 OpenSearch 插件以及中 OpenSearch提供的部分索引、搜索和元数据 API 操作。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

- 支持 OpenSearch 的 API 操作和权限
- 支持的 OpenSearch 插件

支持 OpenSearch 的 API 操作和权限

下表列出了 OpenSearch Serverless 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** 此权限也适用于使用 OpenSearch 仪

数据访问策略权限	OpenSearch API 操作	描述和注意事项
		表板上的示例数据创建索引。
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> • 头 <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 • 索引存在 • CAT 索引（响应中不包括health或status字段。）
aoss:WriteDocument	• 删除 <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`	• GET <index>/_analyze • 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 • HEAD <index>/_doc/<id> • HEAD <index>/_source/<id> • POST <index>/_analyze • POST <index>/_explain/<id> • POST <index>/_count • POST <index>/_field_caps • POST <index>/_rank_eval • POST <index>/_search • POST _analyze • POST _field_caps • POST _search	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

数据访问策略权限	OpenSearch API 操作	描述和注意事项
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 无服务器集合预先打包了来自社区的以下插件。 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 Serverless 还包括所有作为模块发布的插件。

监控 Amazon OpenSearch 无服务器

监控是维护 Amazon OpenSearch Serverless 和其他 AWS 解决方案的可靠性、可用性和性能的重要组成部分。 AWS 提供以下监控工具，用于监视 OpenSearch Serverless、在出现问题时报告以及在适当时自动采取措施：

- Amazon 会实时 CloudWatch监控您的 AWS 资源和您运行 AWS 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 CloudWatch 跟踪您的 Amazon EC2 实例的 CPU 使用率或其他指标，并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。
- AWS CloudTrail 捕获由某个 AWS 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。您可以识别哪些用户和帐户拨打了电话 AWS、发出呼叫的源 IP 地址以及呼叫发生的时间。有关更多信息，请参阅 AWS CloudTrail 用户指南。
- 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，并在其他事件发生 AWS 服务 时触发自动操作。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

您可以使用监控 Amazon OpenSearch Serverless CloudWatch，它会收集原始数据并将其处理为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。

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 会在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 计算单元 (OCU) 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（例如身份验证或可用性问题）失败时，Serverless 会发出此指标。 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 计算单元 (OCU) 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 无服务器将索引数据存储在 Amazon S3 中。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 AWS CloudTrail

Amazon OpenSearch Serverless 与 AWS CloudTrail一项服务集成，可记录用户、角色或 AWS 服务在 Serverless 中执行的操作。

CloudTrail 将 OpenSearch 无服务器的所有 API 调用捕获为事件。捕获的调用包括来自 OpenSearch 服务控制台的无服务器部分的调用和对 OpenSearch 无服务器 API 操作的代码调用。

如果您创建了跟踪，则可以将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 OpenSearch 无服务器的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “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 Serverless 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《AWS CloudTrail 用户指南》。

OpenSearch 中的无服务器信息 CloudTrail

CloudTrail 在您创建账户 AWS 账户 时已在您的账户上启用。在 OpenSearch Serverless 中发生活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在中查看、搜索和下载最近发生的事件 AWS 账户。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录您的事件 AWS 账户，包括 OpenSearch 无服务器的事件，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 AWS 区域。

跟踪记录 AWS 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 AWS 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：

- 创建跟踪记录概述
- CloudTrail 支持的服务和集成
- 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个地区的 CloudTrail 日志文件和接收来自多个账户的 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。

每个事件或日记账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：

- 请求是使用根证书还是 AWS 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 AWS 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，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。

事件表示来自任何源的单个请求。它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的顺序出现。

以下示例显示了演示该`CreateCollection`操作的 CloudTrail 日志条目。

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"AssumedRole",
      "principalId":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/test-user",
      "accountId":"123456789012",
      "accessKeyId":"access-key",
      "sessionContext":{
         "sessionIssuer":{
            "type":"Role",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:role/Admin",
            "accountId":"123456789012",
            "userName":"Admin"
         },
         "webIdFederationData":{

         },
         "attributes":{
            "creationDate":"2022-04-08T14:11:34Z",
            "mfaAuthenticated":"false"
         }
      }
   },
   "eventTime":"2022-04-08T14:11:49Z",
   "eventSource":"aoss.amazonaws.com",
   "eventName":"CreateCollection",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"AWS Internal",
   "userAgent":"aws-cli/2.1.30 Python/3.8.8 Linux/5.4.176-103.347.amzn2int.x86_64 exe/
x86_64.amzn.2 prompt/off command/aoss.create-collection",
   "errorCode":"HttpFailureException",
```

```
   "errorMessage":"An unknown error occurred",
   "requestParameters":{
      "accountId":"123456789012",
      "name":"test-collection",
      "description":"A sample collection",
      "clientToken":"d3a227d2-a2a7-49a6-8fb2-e5c8303c0718"
   },
   "responseElements": null,
   "requestID":"12345678-1234-1234-1234-987654321098",
   "eventID":"12345678-1234-1234-1234-987654321098",
   "readOnly":false,
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "recipientAccountId":"123456789012",
   "eventCategory":"Management",
   "tlsDetails":{
      "clientProvidedHostHeader":"user.aoss-sample.us-east-1.amazonaws.com"
   }
}
```

使用 Amazon 监控 OpenSearch 无服务器事件 EventBridge

亚马逊 OpenSearch 服务与亚马逊集成 EventBridge ，可通知您某些影响您域名的事件。来自 AWS 服务的事件几乎实时 EventBridge 地传送到。同样的事件也会发送到亚马逊 CloudWatch 的前身 Amazon Events 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 AWS Lambda
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams
- 激活 AWS 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

当 OpenSearch 无服务器事件发生时，您可以使用AWS 用户通知来接收通知。事件是 OpenSearch 无服务器环境变化的指标，例如当你达到 OCU 使用量的最大限制时。 Amazon EventBridge 接收事件并

将通知发送到 AWS 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 当发生以下与 OCU 相关的事件之一 EventBridge 时，Serverless 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，Serverless 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，Serverless 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 AWS于开源 OpenSearch 集群。创建域时，您需要指定其设置、实例类型、实例数量和存储分配。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。您可以使用 OpenSearch 服务控制台、 AWS Command Line Interface (AWS CLI) 或 AWS 软件开发工具包的说明完成每个过程。

创建 OpenSearch 服务域

本节介绍如何使用 OpenSearch 服务控制台或使用 AWS 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. 转至 http://aws.amazon.com，然后选择登录到控制台。
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 AWS 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 “支持的版本”。

   （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”。
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。

设置	描述
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	如果选择了某个预调配 IOPS SSD 卷类型，则可以输入该卷可支持的每秒 I/O 操作数（IOPS）。

13. （可选）如果您选择了gp3卷类型，请展开 “高级设置”，指定存储价格中包含的额外 IOPS（每个数据节点每预配置 3 TiB 卷大小最多可达 16,000 个）和吞吐量（每个数据节点每预配置 3 TiB 卷大小最多可达 1,000 MiB/s），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 AWS 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的 VPC。
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	选择双堆栈或 IPv4 作为 IP 地址类型。双堆栈允许您跨 IPv4 和 IPv6 地址类型共享域资源，也是推荐选项。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 API。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 “OpenSearch 控制面板的 Amazon Cognito 认证”。
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 “Identity and Access Management”。

> **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 “N ode-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 AWS 自有密钥”，让 S OpenSearch ervice 代表您创建 AWS 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，选择开始时间以计划需要蓝绿部署的服务软件更新和自动调整优化。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 (AWS CLI)

您可以使用，而不是使用控制台创建 OpenSearch 服务域 AWS CLI。有关语法，请参阅 AWS 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 AWS 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 (AWS SDK)

AWS 软件开发工具包（安卓和 iOS 软件开发工具包除外）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 AWS 软件开发工具包”。有关安装和使用软件开发 AWS 工具包的更多信息，请参阅AWS 软件开发套件。

创建 OpenSearch 服务域 (AWS CloudFormation)

OpenSearch 服务与一项服务集成 AWS CloudFormation，该服务可帮助您对 AWS 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AWS 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 “Identity and Access Management” 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. 转至 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。蓝绿部署创建用于域更新的闲置环境，复制生产环境，在完成更新后将用户路由到新环境。在蓝绿部署中，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

- 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

通常会导发蓝/绿部署的更改

以下操作会引发蓝/绿部署：

- 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
- 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
- 减小 EBS 卷大小
- 更改 EBS 卷大小、IOPS 或吞吐量（前提是您所做的最后一次更改正在进行或发生在 6 小时以前）
- 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。如果更改已在进行中，则新请求将被拒绝。您可以使用 `DescribeDomainChangeProgress` API 查看当前更改状态。

通常不会导发蓝/绿部署的更改

在大多情况下，以下操作不会引发蓝/绿部署：

- 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
- 如果您的域有专用主节点，请更改专用主实例类型或数量（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
- 将每个数据节点的卷大小增加到 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蓝/绿部署，请在更新域名之前进行试运行（如果此选项可用）。有些更改不提供试运行选项。我们通常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蓝/绿部署，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.aws.amazon.com/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. 在 Edit cluster configuration（编辑集群配置）页面上，您可以更改实例类型、节点数及任何其他配置。在摘要面板中确认更改后，选择 Run（运行）。
5. 试运行完成后，结果将自动显示在页面底部，同时显示试运行 ID。结果会通知您所做的更改属于哪个类别：

   - 启动蓝绿部署
   - 不需要蓝绿部署
   - 包含在保存更改之前需要解决的验证错误

   请注意，每次试运行都会覆盖之前的试运行。如需稍后查看每次试运行的详细信息，请务必保存试运行 ID。每次试运行的有效期为 90 天，或者直到您进行配置更新。
6. 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)
```

启动和跟踪配置更改

> **Note**
>
> 您一次可以申请一项配置更改。您还可以在单个请求中对多个配置更改进行分组。请等待您的域名状态变成，Active然后再请求任何其他配置更改。

您可以在 Amazon S OpenSearch ervice 控制台中查看 “域处理状态” 和 “配置更改状态” 字段，以跟踪域名和配置更改。您还可以通过 API 响应中的和ConfigChangeStatus参数来跟踪域

名DomainProcessingStatus和配置的更改。有关更多信息，请参阅 OpenSearch 服务 API 参考中的DomainStatus数据类型。

域处理状态可见性：通过查看控制台中的 “域处理状态” 字段，您可以轻松确定域的配置状态。同样，DomainProcessingStatusAPI 参数可用于识别状态。以下值是域的处理状态：

- 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。

> **Note**
>
> Modifying在域需要移动分片才能完成配置更改的情况下，您可能会看到该状态。为了向后兼容，Processing参数的行为在 API 响应中保持不变，并且在核心配置更改完成后立即设置为 false，无需等待分片移动完成。

- 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。

配置状态可见性：配置更改可以由操作员启动（例如添加新的数据节点、更改实例类型），也可以由服务启动（例如自动调整和非高峰时段更新）。您可以在 Amazon S OpenSearch ervice 控制台的配置更改状态字段和 ConfigChangeStatus API 响应中找到最新配置更改详细信息的状态。以下值表示域的配置状态：

- Pending: 配置更改请求已提交。
- Initializing: 服务正在初始化配置更改请求。
- Validating：服务部门正在验证请求的更改和所需的资源。
- Awaiting user inputs：当操作员预计某些配置更改（例如实例类型更改）会继续进行时适用。您可以编辑配置更改。
- Applying changes: 服务正在应用请求的配置更改。
- Cancelled: 配置更改已取消。如果您收到验证失败状态，则可以在控制台中单击 “取消” 或调用 CancelDomainConfigChange API 操作。如果执行此操作，则所有已应用的更改都将被回滚。

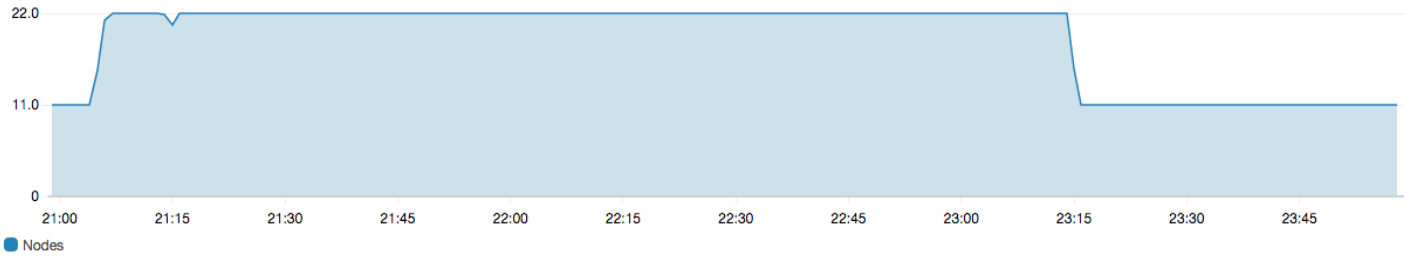
- `Completed`: 请求的配置更改已成功完成。
- `Validation Failed`: 请求的更改验证失败。不应用任何配置更改。

> **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”。在验证失败事件期间，您可以取消、重试或编辑配置更改。

API 摘要：您可以使用`DescribeDomain`、`DescribeDomainChangeProgress`和 `DescribeDomainConfig` API 操作来获取详细的配置更新状态。此外，您还可以`CancelDomainConfigChange`使用在验证失败时取消更新。有关更多信息，请参阅OpenSearch 服务 API 文档

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。当 S OpenSearch ervice 将数据从旧集群复制到新集群时，它还可以增加搜索和索引延迟。在集群上保持充足的容量很重要，这有助于处理与这些蓝/绿部署相关的开销。

> **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。您还可以使用 DescribeDomainChangeProgressAPI 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所需的资源以启动蓝/绿部署。
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

阶段名称	描述
	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）。
动态更新	当更新不需要蓝/绿部署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

在蓝/绿部署期间，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：

- 随着集群有更多需要管理的节点，领导节点的使用量会暂时增加。
- 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。
- 蓝/绿部署期间，随着集群负载的增加，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蓝/绿部署。

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。

问题	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。

问题	错误代码	故障排除步骤
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 AWS 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 AWS CLI 命令： `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 AWS 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 AWS CLI 命令进行配置： `aws cognito-idp create-user-pool-domain --domain my-domain --user-pool-id id`

问题	错误代码	故障排除步骤
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 AWS SDK 手动配置角色。 AWS 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。
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 AWS Key Management Service (AWS 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 AWS 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。

问题	错误代码	故障排除步骤
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。

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。

问题	错误代码	故障排除步骤
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。如果您的域使用仅 IPv4 地址类型，请使用仅 IPv4 的子网。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。如果您的域使用仅 IPv4 地址类型，请使用仅 IPv4 的子网。如果您的域使用双堆栈模式，请使用双堆栈子网。

Amazon 服务中的 OpenSearch 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 S OpenSearch ervice 会定期发布服务软件更新，以增加功能或以其他方式改进您的域名。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信息 OpenSearch，请参见the section called “升级域”。

主题

可选更新与必需更新

OpenSearch 服务有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。

无论您是否计划更新，如果您对域进行更改以导致蓝/绿部署，S OpenSearch ervice 都会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 服务将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 AWS CLI，请在创建或更新域名`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 (CVE) 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 会自动安排这些更新，并在预定更新前 72 小时（三天）通过电子邮件和控制台通知通知您。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。如果您未对必需的更新采取任何操作，也没有进行任何导致蓝/绿部署的域更改，则 OpenSearch 服务可以在域的非高峰期限内，在指定的截止日期（通常为上线后 14 天）之后的任何时间启动更新。

无论何时安排更新，如果您对域名进行更改以导致蓝/绿部署，S OpenSearch ervice 都会自动为您更新您的域。

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。当您选择 “更新” 时， OpenSearch 服务会将请求置于队列中，并在有时间时开始更新。
- 当您启动服务软件更新时， OpenSearch 服务会在更新开始和完成更新时发送通知。
- 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 OpenSearch 服务控制台 AWS CLI、或其中一个软件开发工具包请求服务软件更新。

控制台

请求服务软件更新

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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。

AWS CLI

发送启动服务软件更新的start-service-software-update AWS 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。这种PENDING_UPDATE状态的持续时间可能会有很大差异，这取决于您 AWS 区域 和 OpenSearch 服务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 AWS 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 服务会在响应中提供其他可用时段建议。

AWS 软件开发工具包

此示例 Python 脚本使用中的 d escribe_domain 和 start_service_software_update 方法 AWS SDK for Python (Boto3) 来检查域是否有资格进行服务软件更新，如果有资格进行服务软件更新，则开始更新。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 服务域在当地时间晚上 10:00 至上午 8:00 之间每天有 10 小时的时段，我们将其视为非高峰时段。 OpenSearch 服务使用此窗口来安排域的服务软件更新。非高峰期更新有助于最大限度地减少在流量较高时段对集群专用主节点的压力。 OpenSearch 未经您的同意，服务无法在 10 小时之外启动更新。

- 对于可选更新，S OpenSearch ervice 会通知您更新的可用性，并提示您在即将到来的非高峰时段安排更新。
- 对于所需的更新，S OpenSearch ervice 会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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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

要使用在即将到来的非高峰时段安排更新 AWS 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、需要更新、已启动、已完成或失败时，服务会发送通知。您可以在 OpenSearch 服务控制台的通知面板上查看这些通知。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。您可以使用配置规则，EventBridge 以便在收到事件时发送电子邮件或执行特定操作。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”。

要查看发送给 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。

省/自治区/直辖市	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”。

省/自治区/直辖市	描述
高错误率	OpenSearch 集群在尝试处理请求时返回了大量 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	Split brain 意味着你的 OpenSearch 集群有多个主节点，并且已经分成两个集群，这些集群永远不会自行重新加入。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 AWS Support。
Amazon Cognito 集成问题	您的域名对 OpenSearch 控制面板使用身份验证， 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，请重新创建缺失的资源并配置 OpenSearch 服务域以使用该资源。
其他 服务问题	OpenSearch 服务本身的问题可能会导致您的域名显示为不符合更新资格。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 AWS Support。

为 Amazon OpenSearch 服务定义非高峰时段

创建 Amazon S OpenSearch ervice 域时，您可以定义每天 10 小时的时段，该窗口被视为非高峰时段。 OpenSearch 只要有可能，服务就会使用此窗口来安排服务软件更新和自动调整优化，这些更新和自动调整需要在相对较低的流量时间内部署蓝/绿。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。

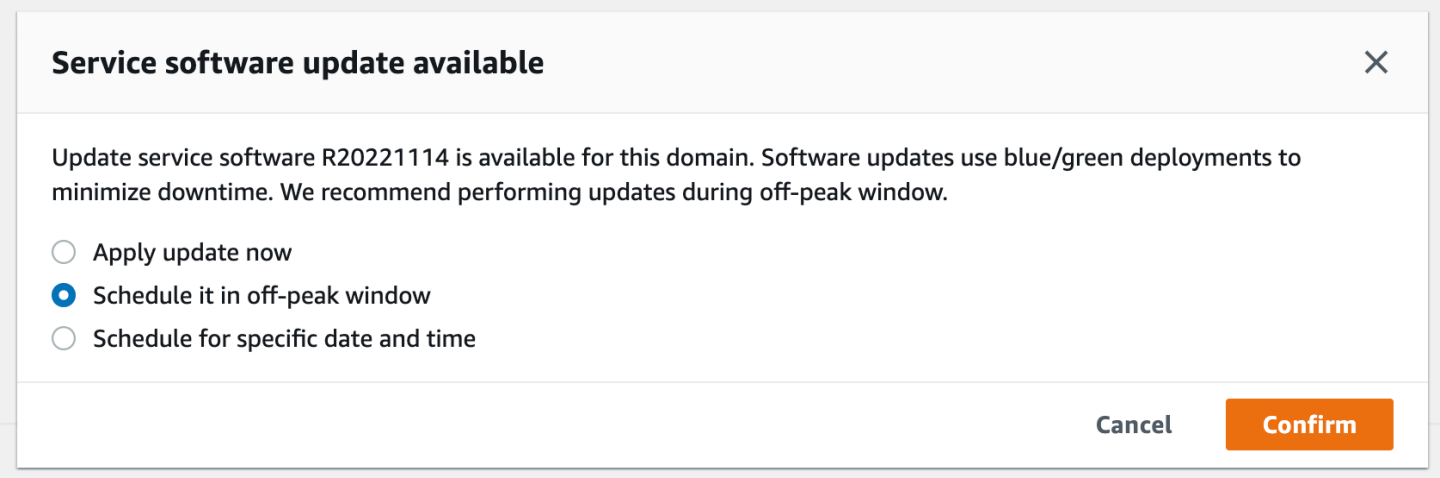
> **Note**
>
> 非高峰窗口于 2023 年 2 月 16 日推出。默认情况下，在此日期之前创建的所有域都禁用非高峰窗口。您必须手动启用和配置这些域的非高峰窗口。默认情况下，在此日期之后创建的所有域都启用非高峰窗口。启用域的非高峰窗口后，您将无法再将其禁用。

主题

非高峰窗口服务软件更新

OpenSearch 服务有两大类服务软件更新：可选的和必需的。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。



对于所需的更新，S OpenSearch ervice 会自动安排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）来计划蓝绿部署。您无法禁用域的非高峰窗口。有关需要蓝绿部署的自动调整优化的列表，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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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

要使用修改非高峰时段AWS 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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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 Save changes（保存更改）。

CLI

要使用配置自定义非高峰时段AWS CLI，UpdateDomainConfig请发送请求并以 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 AWS 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性和当前状态。

要查看域的计划操作

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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AWS 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 服务会通知您定期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. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
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 Save changes（保存更改）。

CLI

要使用重新安排操作AWS 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 服务会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，选择域名以打开其配置。
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 Auto-Tune 维护窗口迁移到非高峰时段AWS CLI，请发送UpdateDomainConfig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”。

主题

- 开始使用通知
- 通知严重性
- 示例 EventBridge 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:ListNotifications` 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：

```
{
  "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

严重性	描述	示例
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 AWS 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。

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 AWS 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
- 数据节点的 GP3 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。
- 带待机功能的多可用区仅适用于 `m5`、`c5`、`r5`、`r6g`、`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。
- 您只能使用预调配 IOPS 固态硬盘、通用型 SSD (GP3) 或带待机功能的实例备份存储。
- 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

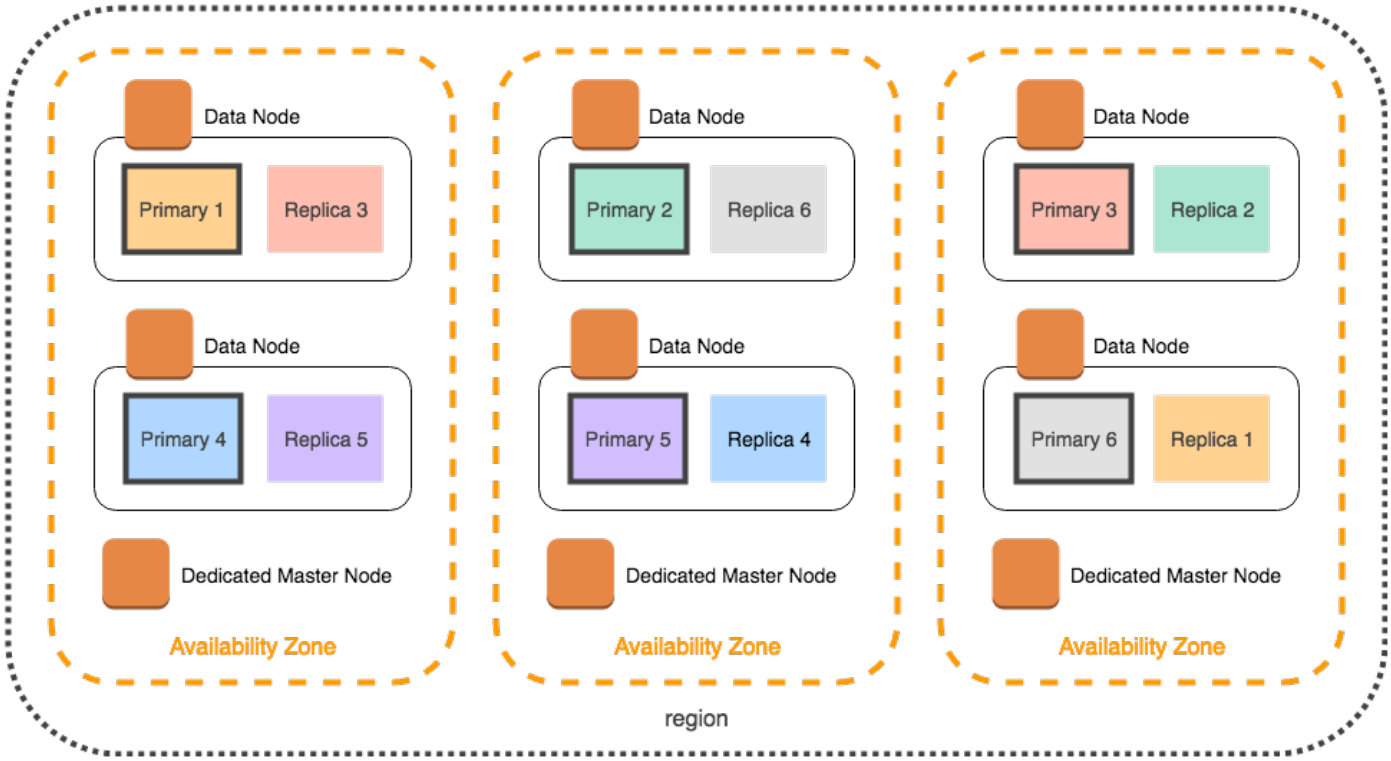
OpenSearch 服务仍支持不带备用状态的多可用区，可用性高达 99.9%。节点分布在各个可用区中，可用性取决于可用区的数量和数据副本。而在带有待机功能的情况下，您必须按照最佳实践配置您的域，而在没有待机功能的情况下，您可以自己选择可用区、节点和副本的数量。除非您的现有工作流程会因创建带有待机功能的域而中断，否则我们不建议使用此选项。

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 AWS 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

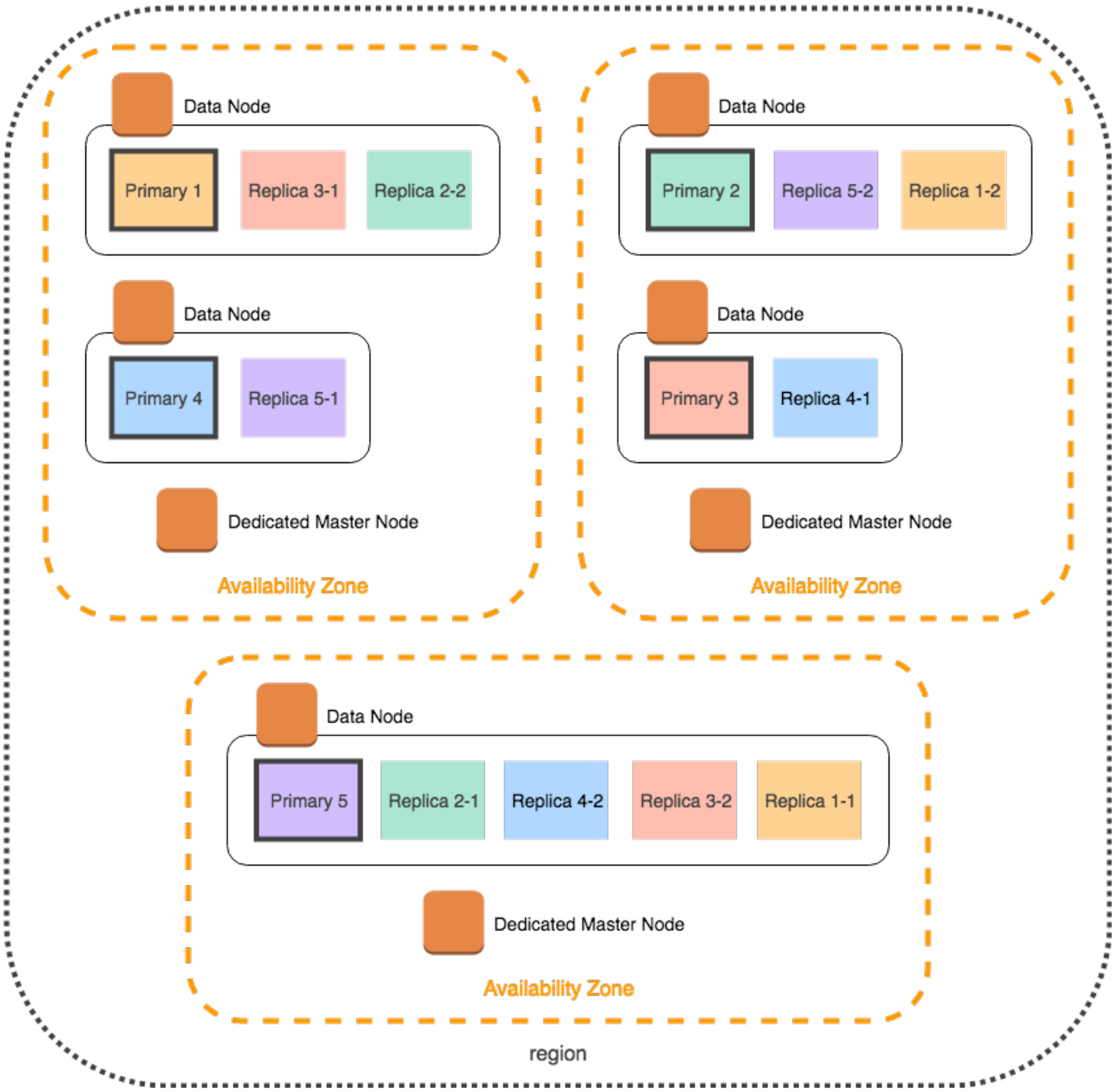
如果您启用不带待机功能的多可用区，则应该为集群中的每个索引创建至少一个副本。如果没有副本，S OpenSearch ervice 就无法将您的数据副本分发到其他可用区。幸运的是，所有索引的默认配置均为副本数量等于 1。如下图所示，S OpenSearch ervice 会尽最大努力将主分片及其对应的副本分片分配到不同的区域。



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。

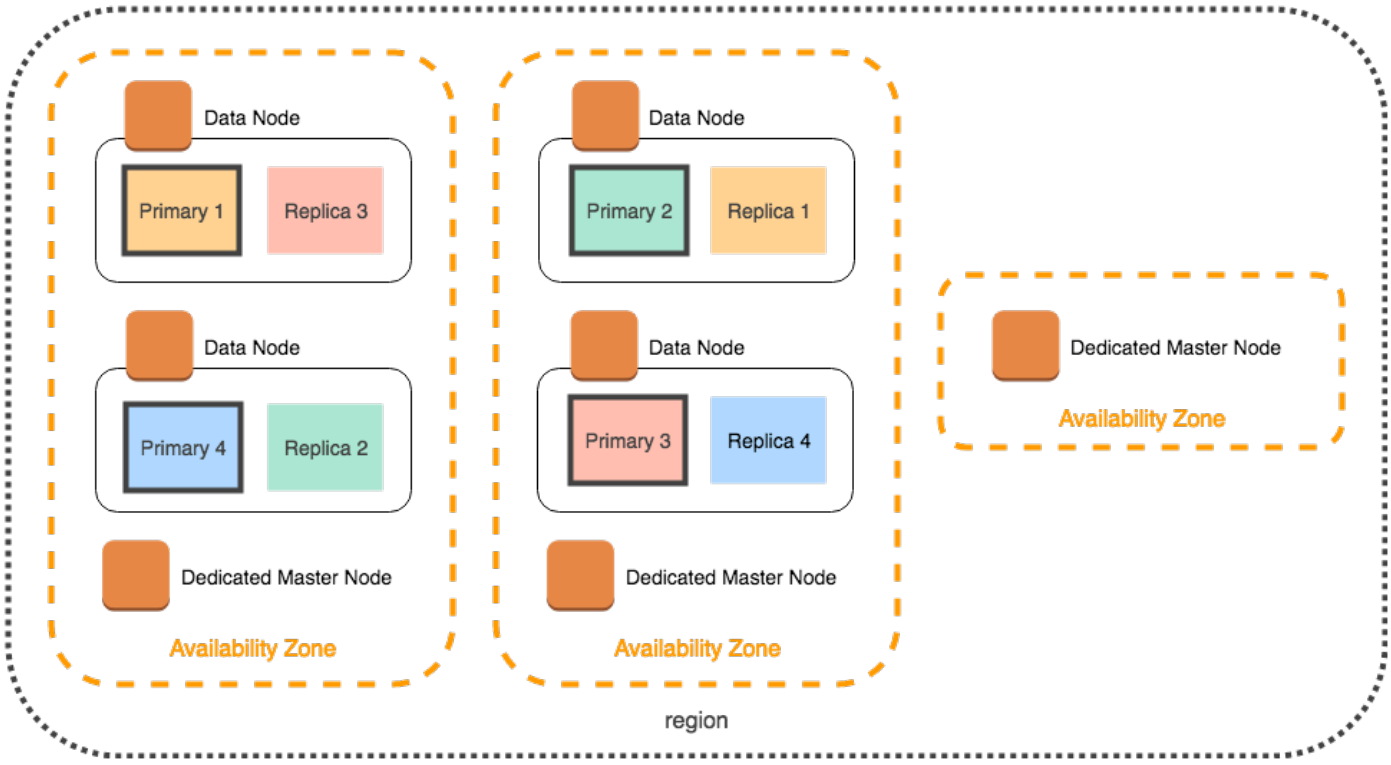


为避免这些可能导致单个节点紧张并损害性能的情况，如果您计划每个索引有两个或更多副本，建议您选择带待机功能的多可用区或一个为三的倍数的实例计数。

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 AWS 资源启动到虚拟私有云 (VPC) 中。VPC 是专为您服务的虚拟网络 AWS 账户。它在逻辑上与 AWS 云中的其他虚拟网络隔绝。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 AWS 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。

主题

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

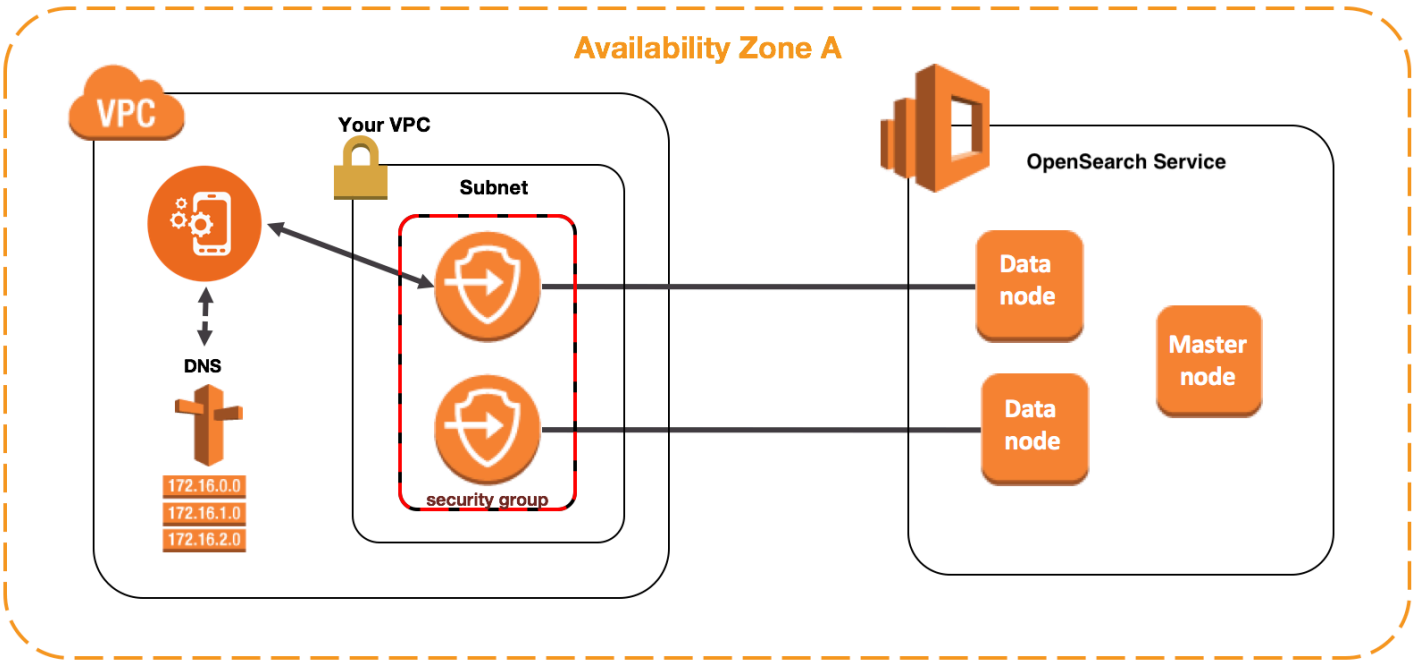
在 VPC 内运营 OpenSearch 服务域有以下限制：

- 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 OpenSearch 仪表板的访问权限”。

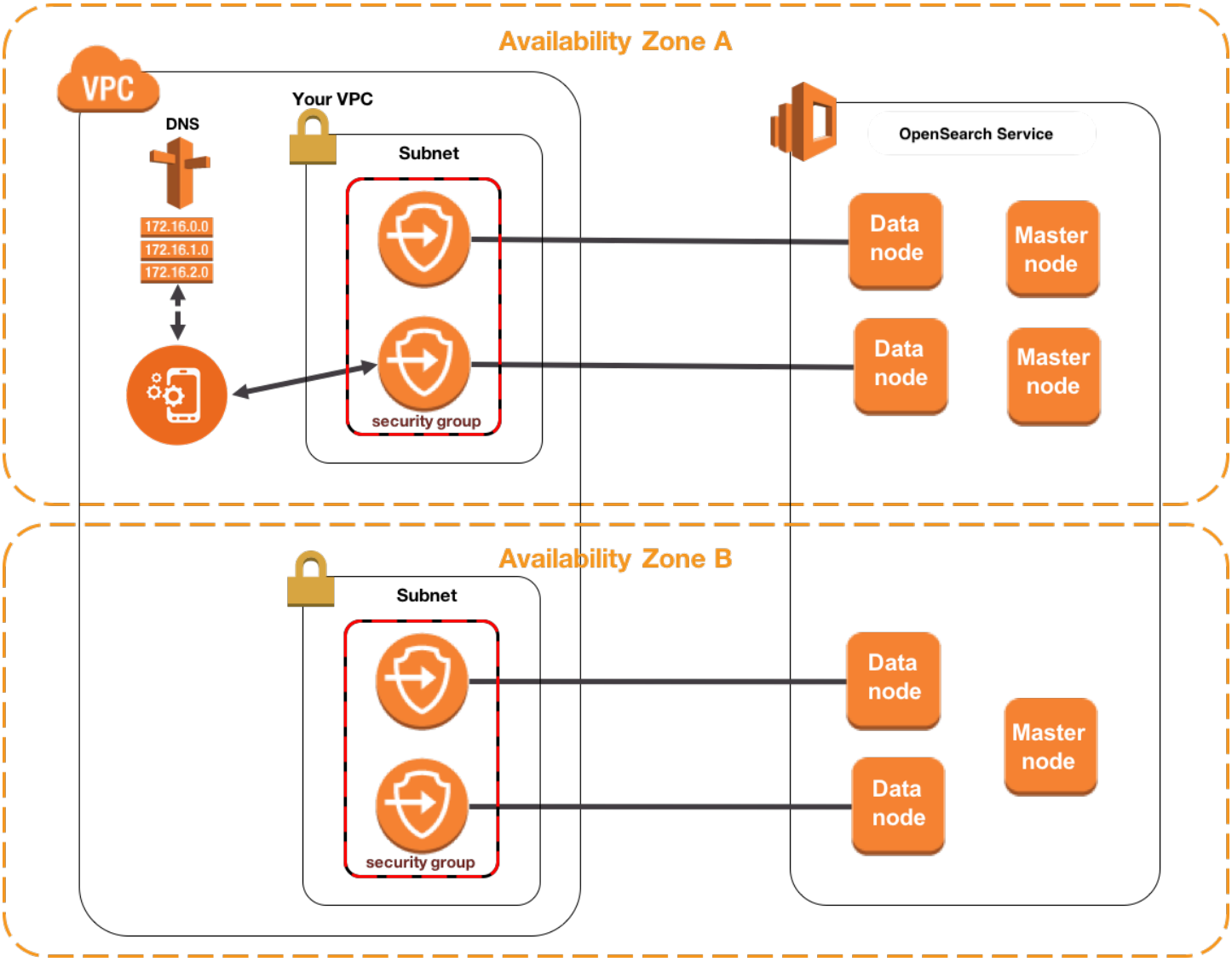
架构

为了支持 VPC， OpenSearch 服务会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 服务会从您的子网的 IPv4 地址范围内为每个 ENI 分配一个私有 IP 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，VPC 还可让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、 AWS CLI 或其中一个 AWS 软件开发工具包。有关更多信息，请参阅 Amazon VPC 用户指南 中的使用 VPC。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：

Type	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*

   第一条规则可让您的 SSH 连接到您的 EC2 实例。第二个允许 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.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 Network Interfaces 部分看到网络接口及其相关的 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。

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，它向服务委派权限，使其可以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。

在 S OpenSearch ervice 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”。

在亚马逊 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”。

主题

- 使用索引状态管理自动执行快照
- 将 Curator 用于快照

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 Simple Storage Service 用户指南中的创建存储桶。 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：
	
	`{`
	`  "Version": "2012-10-17",`
	`  "Statement": [{`
	`  "Action": [`
	`  "s3:ListBucket"`
	`  ],`
	`  "Effect": "Allow",`
	`  "Resource": [`

先决条件	描述

```
        "arn:aws:s3::: s3-bucket-name "
      ]
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3::: s3-bucket-name /*"
      ]
    }
  ]
}
```

有关附加托管策略的说明，请参阅 IAM 用户指南中的添加 IAM 身份权限。

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：

```
{
  "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 用户指南中的修改角色信任策略。

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：
	
	{
	"Version": "2012-10-17",
	"Statement": [
	{
	"Effect": "Allow",
	"Action": "iam:PassRole",
	"Resource": "arn:aws:iam:: *123456789012* :role/*TheSnapshotRole* "
	},
	{
	"Effect": "Allow",
	"Action": "es:ESHttpPut",
	"Resource": "arn:aws:es: *region*:*123456789012* :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* "}

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 AWS 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 ARN 置于 Backend roles（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。

此请求采用以下形式：

```
PUT domain-endpoint/_snapshot/my-snapshot-repo-name
{
  "type": "s3",
  "settings": {
    "bucket": "s3-bucket-name",
    "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 AWS 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 AWS 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 开发人员指南中的使用采用 Amazon S3 托管加密密钥的服务器端加密保护数据。

或者，您可以使用 AWS KMS 密钥对用作快照存储库的 S3 存储桶进行服务器端加密。如果您使用这种方法，请确保为用于加密 S3 存储桶的 AWS KMS 密钥提供TheSnapshotRole权限。有关更多信息，请参阅AWS 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 AWS 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 AWS SDK for Python (Boto3)、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": "s3-bucket-name",
    "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 AWS Support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）`snapshot` 操作，以根据索引的年龄、大小或文档数量的变化自动触发索引快照。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
```

升级亚马逊 OpenSearch 服务域名

> **Note**
>
> OpenSearch 而且 Elasticsearch 版本升级不同于服务软件更新。有关更新服务域的服务软件的信息 OpenSearch ，请参阅the section called “服务软件更新”。

亚马逊 OpenSearch 服务为运行 OpenSearch 1.0 或更高版本或 Elasticsearch 5.1 或更高版本的域名提供就地升级。如果您使用诸如 Amazon Data Firehose 或 Amazon CloudWatch Logs 之类的 OpenSearch 服务将数据流式传输到服务，请在迁移 OpenSearch 之前检查这些服务是否支持较新版本的。

主题

支持的升级途径

目前， OpenSearch 服务支持以下升级路径：

之前版本	目标版本
OpenSearch 1.3 或 2。 x	OpenSearch 2。 x 版本 2.3 具有以下重大更改： • 在 2.0 版本中，该`type`参数已从所有 OpenSearch API 端点中删除。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（热索引或冷索引），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。对于不兼容的索引 UltraWarm 或冷索引，请将其迁移到热存储，重新索引数据，然后将其迁移回温存储或冷存储。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 Service 重命名。
Elasticsearch 6.8	Elasticsearch 7 x 或 OpenSearch 1。 x **Important** Elasticsearch 7.0 和 OpenSearch 1.0 包含许多重大更改。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 中的重大更改，请参阅Amazon OpenSearch Service 重命名。

之前版本	目标版本
	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，特定 API 在请求正文中不再需要映射类型（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 服务会检查是否存在可能阻碍升级的问题，除非这些检查成功，否则不会继续执行下一步操作。
2. 快照 — OpenSearch 服务会拍摄 OpenSearch 或 Elasticsearch 集群的快照，除非快照成功完成，否则不会继续执行下一步操作。如果升级失败， OpenSearch 服务将使用此快照将集群恢复到其原始状态。有关更多信息，请参阅 the section called “升级后无法降级”。
3. 升级- OpenSearch 服务会启动升级，升级可能需要 15 分钟到几个小时才能完成。 OpenSearch 在部分或全部升级期间，仪表板可能不可用。

开始升级（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。

如果集群有专用的主节点，则 OpenSearch 升级无需停机即可完成。否则，集群在选择主节点时可能会在升级后几秒钟无响应。

将域名升级到更高版本 OpenSearch 或 Elasticsearch

1. 创建您的域的手动快照。此快照用作备份，如果您想恢复使用先前 OpenSearch 版本，则可以在新域上恢复该快照。
2. 转至 http://aws.amazon.com，然后选择登录到控制台。
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。如果您要升级到某个 OpenSearch 版本，则会出现 “启用兼容模式” 选项。如果您启用此设置，则将其版本 OpenSearch 报告为 7.10，以允许 Elasticsearch OSS 客户端和 Logstash 等插件继续使用亚马逊服务。 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。

开始升级（CLI）

您可以使用以下操作为您的域识别 OpenSearch 或 Elasticsearch 的正确版本、开始就地升级、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 AWS CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

开始升级（SDK）

此示例使用中的OpenSearchService低级 Python 客户端 AWS SDK for Python (Boto) 来检查域是否有资格升级到特定版本，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

当您启动 OpenSearch 或 Elasticsearch 版本升级时，OpenSearch 服务会首先执行一系列验证检查，以确保您的域名符合升级条件。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。

问题	描述
不支持可选插件	当您使用可选插件升级域名时，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 个版本的 Elasticsearch，其默认设置为每个节点不超过 1,000 个分片。如果您当前集群中的某个节点超过此设置，则 OpenSearch 服务将不允许您升级。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 AWS Support。
找不到主节点	OpenSearch 服务找不到集群的主节点。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 AWS Support。
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。
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 AWS Support。
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。

问题	描述
	如果 OpenSearch 服务无法检查快照是否正在进行中，则升级可能会失败。
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 AWS Support。
索引不兼容	一个或多个索引与目标版本不兼容。如果您从旧版本 OpenSearch 或 Elasticsearch 迁移索引，则可能会出现此问题。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 仪表板别名问题	`.dashboards` 已配置为别名并映射到不兼容的索引，可能来自早期版本的 Dashboard OpenSearch s。重新编制索引并重试。
红色控制面板状态	OpenSearch 仪表板状态为红色。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 服务服务问题	OpenSearch 服务本身的问题可能会导致您的域名显示为不符合升级资格。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 AWS Support。

使用快照迁移数据

就地升级是将域名升级到更高版本 OpenSearch 或 Elasticsearch 版本的更简单、更快速、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版本 OpenSearch 或 Elasticsearch 版本的域中。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 1.3 或 2。 x	OpenSearch 2。 x	1. 查看 OpenSearch 2.3 版的重大更改，看看是否需要调整索引或应用程序。
		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。

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": "*",`

之前版本	目标版本	迁移过程
		"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，_restoreREST 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. 查看 OpenSearch 1.0 的重大更改，看看是否需要调整索引或应用程序。
		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 域恢复到该 OpenSearch 域。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** 1.5 快照与 2.3 兼容，但 OpenSearch 服务 2.3 域不支持该`_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。

为亚马逊 OpenSearch 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以让您更轻松地引用自己的 OpenSearch 和 OpenSearch 控制面板网址。你可以加入贵公司的品牌，也可以只使用比标准 easier-to-remember 端点更短的端点。

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。

您可以通过在 AWS Certificate Manager (ACM) 中生成证书或导入自己的证书来保护自定义终端节点。

新域的自定义终端节点

您可以使用服务控制台或配置 API 为新的 OpenSearch OpenSearch 服务域启用自定义终端节点。AWS CLI

要自定义终端节点（控制台）

1. 在 OpenSearch 服务控制台中，选择创建域并提供该域的名称。
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 AWS 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 AWS 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，并且必须与您的 OpenSearch 服务域位于同一个账户中。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AWS 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

要向现有 OpenSearch 服务域添加自定义终端节点，请选择编辑并执行上述步骤 2-4。

后续步骤

为 OpenSearch 服务域启用自定义终端节点后，您可以在 Amazon Route 53（或您的首选 DNS 服务提供商）中创建别名记录映射。创建 CNAME 映射将使您能够将流量路由到您的自定义终端节点及其子域。如果没有此映射，您将无法将流量路由到您的自定义终端节点。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域。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。您的自定义终端节点的双堆栈功能取决于您将 CNAME 记录指向的服务生成的终端节点。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。

如果您对 OpenSearch控制面板使用 SAML 身份验证，则必须使用新的 SSO 网址更新您的 IdP。

您可以使用 Amazon Route 53 创建别名记录类型，将您的域的自定义终端节点指向双堆栈搜索终端节点。要创建别名记录类型，必须将您的域配置为使用双堆栈 IP 地址类型。你可以使用 Route 53 API 来做到这一点。

要使用 Route 53 API 创建别名记录类型，请指定域名的别名目标。您可以在 OpenSearch 服务控制台的自定义终端节点部分的托管区域（双堆栈）字段中找到您的域的别名目标，也可以使用 `DescribeDomain` API 并复制的值`DomainEndpointV2HostedZoneId`。

自动调整 Amazon OpenSearch Service

Amazon OpenSearch Service 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 OpenSearch 服务设置。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 OpenSearch Service 控制台中查看其建议。

自动调整可在运行任何 OpenSearch 版本或 Elasticsearch 6.7 或更高版本的域上的商业 AWS 区域 中在受支持的实例类型上使用。

主题

- 更改类型
- 启用或禁用自动调整

- 计划自动调整增强功能
- 监控自动调整更改

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，OpenSearch Services 将实例的 RAM 的 50% 用于 JVM 堆，最大堆大小为 32 GiB。 增加此百分比可以给 OpenSearch 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（对于 `t2` 实例类型为 2%，对于 `t3.small` 为 1%），OpenSearch 将会限制所有新的 `_search` 和 `_bulk` 请求，直到现有请求完成。

更改类型	类别	描述
		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

默认情况下，OpenSearch 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 AWS 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/home 打开 Amazon OpenSearch Service 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 AWS CLI 启用自动调整，请发送 UpdateDomainConfig 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/home 打开 Amazon OpenSearch Service 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 UpdateDomainConfig 请求中纳入 `UseOffPeakWindow`：

```
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 EventBridge 配置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 EventBridge 的每个自动调整事件的格式，请参阅 the section called “自动调整事件”。

为 Amazon OpenSearch 服务域名添加标签

标签允许您将任意信息分配给 Amazon S OpenSearch ervice 域，这样您就可以对这些信息进行分类和筛选。标签是您定义并与 OpenSearch 服务域关联的键值对。您可以使用这些标签通过对标签相似的

资源的费用进行分组来跟踪成本。 AWS 不会对你的标签应用任何语义含义。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
标签密钥	标签键是标签的名称。密钥必须是与其关联的 OpenSearch 服务域所独有的。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 project/Trinity 和 cost-center/Trinity 的标签集中，可以存在键值对。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。 AWS 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，您可以将标签键定义为`project`，标签值为`Salix`，表示 OpenSearch 服务域已分配给 Salix 项目。您也可以使用诸如或之类的密钥使用标签将 OpenSearch 服务域指定为用于测试`environment=test`或生产`environment=production`。尝试使用一组一致的标签密钥，以便更轻松地跟踪与 OpenSearch 服务域关联的元数据。

您还可以使用标签来整理 AWS 账单，以反映您自己的成本结构。为此，请注册以获取包含标签键值的 AWS 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多个 OpenSearch 服务域，然后整理账单信息以查看多个服务中每个域的总费用。有关更多信息，请参阅 https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html 账单和成本管理文档中的AWS 使用成本分配标签。

> **Note**
>
> 对标签进行缓存以用于授权。因此，在 OpenSearch 服务域上添加和更新标签可能需要几分钟才能发布。

使用标签 (控制台)

控制台是标记域的最简单方法。

创建标签（控制台）

1. 转至 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《AWS 管理控制台入门指南》中的标签编辑器。

使用标签 (AWS CLI)

您可以使用 AWS 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 资源名称。
`--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
```

您可以使用--remove-tags命令从 OpenSearch 服务域中移除标签。

语法

```
remove-tags --arn=<domain_arn> --tag-keys Key=<key>,Value=<value>
```

参数	描述
--arn	附加标签的 OpenSearch 服务域的亚马逊资源名称 (ARN)。
--tag-keys	要从服务域中移除的一组以空格分隔的键值对。 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 服务域的亚马逊资源名称 (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
```

使用标签 (AWS SDK)

AWS 软件开发工具包（Android 和 iOS 软件开发工具包除外）支持《亚马逊 OpenSearch 服务 API 参考》中定义的所有操作，包括AddTagsListTags、和RemoveTags操作。有关安装和使用软件开发 AWS 工具包的更多信息，请参阅AWS 软件开发套件。

Python

此示例使用适用于 Python 的 AWS 开发工具包 (Boto) 中的OpenSearchService低级 Python 客户端向域添加标签、列出附加到该域的标签以及从域中移除标签。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
```

对 Amazon OpenSearch 服务域名执行管理操作

如果您需要解决域名问题，Amazon Ser OpenSearch vice 提供了多种管理选项，这些选项可提供精细控制。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。

OpenSearch 服务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稳定。通过在节点上重新启动 OpenSearch进程和重新启动节点本身的管理选项，您可以控制其中一些缓解措施。

您可以使用 AWS Management Console AWS CLI、或 AWS 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。

在节点上重新启动 OpenSearch 进程

在节点上重新启动 OpenSearch 进程

1. 导航到 OpenSearch 服务控制台，网址为https://console.aws.amazon.com/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。

重启数据节点

要重启数据节点

1. 导航到 OpenSearch 服务控制台，网址为https://console.aws.amazon.com/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。

重启控制面板或节点上的 Kibana 流程

要重启控制面板或节点上的 Kibana 流程

1. 导航到 OpenSearch 服务控制台，网址为https://console.aws.amazon.com/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。
- 具有三个或更多数据节点的域支持重启 OpenSearch 和 Elasticsearch 进程以及重启数据节点。
- 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。

- 要在节点上重新启动 OpenSearch 进程或重新启动节点，该域不得处于红色状态，并且所有索引都必须配置副本。

使用亚马逊 OpenSearch 服务使用亚马逊 S3 直接查询

您可以使用亚马逊 OpenSearch 服务直接查询来查询 Amazon S3 中的数据。Amazon Ser OpenSearch vice 提供与 Amazon S3 的直接查询集成，从而无需在服务之间切换即可分析 Amazon S3 中的操作日志和基于 Amazon S3 的数据湖中的操作日志。现在，您可以分析云对象存储中的数据，同时使用 Service 的运营分析和可视化。 OpenSearch

通过使用 Amazon S3 进行直接查询，您不再需要构建复杂的 ETL 管道，也不再需要支付在 OpenSearch 服务和 Amazon S3 存储中复制数据的费用。您还可以安装包含预定义控制面板的常用日志类型模板集成，并配置针对该日志类型量身定制的数据加速。这些模板包括 VPC 流日志、AWS CloudTrail 日志和 Amazon S3 日志。加速包括跳过索引、实体化视图和覆盖索引。

主题

定价

您需要为用于创建和处理直接查询的现有 OpenSearch 服务和 Amazon S3 资源付费。发送到 Amazon S3 的查询使用计费计算并显示为每小时 OpenSearch 计算单位 (OCU)。

使用 Amazon S3 的直接查询有两种类型：交互式查询和加速。交互式查询对 Amazon S3 中的数据进行分析。当您运行新查询时，Ser OpenSearch vice 会启动一个持续至少三分钟的新会话。OpenSearch 服务使会话保持活动状态，以确保后续查询快速运行。加速查询使用计算来维护 OpenSearch 服务中的索引。这些查询通常需要更长的时间，因为它们会将不同数量的数据摄入到 OpenSearch 服务中，从而加快交互式查询的运行速度。

有关更多信息，请参阅 Amazon OpenSearch 服务定价。

限制

以下限制适用于使用 Amazon S3 进行 OpenSearch 服务直接查询。

- 您的 OpenSearch 域名必须是 2.13 或更高版本才能支持 OpenSearch 服务直接查询。
- 在 OpenSearch 无服务器上不可用。
- 您的 OpenSearch 域名和 AWS Glue Data Catalog 必须相同 AWS 账户。您的 Amazon 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 AWS 区域 账户中。
- 某些数据类型不支持。支持的数据类型仅限于 Parquet、CSV 和 JSON。
- OpenSearch 使用 Amazon S3 的服务直接查询仅支持从查询工作台生成的 Spark 表。Spark 流式传输不支持在 AWS Glue Data Catalog 或 Athena 中生成的表，Spark 流式传输是保持加速和更新索引所必需的。
- 在查询之前必须对数据进行扁平化，或者必须使用 SQL in Serv OpenSearch ice 将嵌套列更改为专用列。
- 缺少的列可能需要使用 `COALESCE` SQL 函数返回结果。
- 如果您的数据结构发生变化，则需要更新 AWS Glue 表以及现有的加速。
- OpenSearch 实例类型具有网络有效载荷限制，具体取决于实例类型（10 v 100）。
- AWS CloudFormation 尚不支持模板。

建议

我们建议您在使用直接查询时执行以下操作：

- 使用年、月、日、小时的分区格式将数据提取到 Amazon S3 中，以加快查询速度。
- 对查询设置限制，确保不会提取太多数据。
- 使用索引状态管理（如果适用）来维护实例化视图和覆盖索引的存储。
- 当不再需要加速任务和索引时，将其丢弃。
- 在构建跳过索引时，使用布隆过滤器来获得高基数，使用最小/最大值来表示大范围。建议您使用在高基数字段上设置的值。
- 使用参考指南将数据导出到 Amazon S3。您可以使用诸如CloudFront、和 El astic Load B CloudTrailalancing 之类的 AWS 日志。

配额

您的账户具有以下与 Amazon S3 OpenSearch 服务直接查询相关的配额。每次启动查询时，Ser OpenSearch vice 都会打开一个会话并使其保持活动状态至少十分钟。这可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
每次加速的最大 OCU	20	是
最大临时存储空间	20	是

支持的区域

以下区域可用于通过 Amazon S3 进行 OpenSearch 服务直接查询：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（法兰克福）、欧洲（爱尔兰）、欧洲（斯德哥尔摩）、美国东部（俄亥俄州）和美国西部（俄勒冈）。

创建与亚马逊 S3 的亚马逊 OpenSearch 服务数据源集成

您可以通过 AWS 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 AWS Glue Data Catalog 来管理代表 Amazon S3 存储桶的表。

主题

- 先决条件

- 设置新的直接查询数据来源
- 映射 AWS Glue Data Catalog 角色（如果在创建数据源后启用了细粒度访问控制）
- 后续步骤

先决条件

在创建数据源之前，必须拥有版本为 2.13 或更高版本的 OpenSearch 域。有关设置的说明，请参阅the section called “ 创建 OpenSearch 服务域”。

设置新的直接查询数据来源

您可以使用 AWS Management Console 或 OpenSearch 服务 API 在网域上设置直接查询数据源。

AWS Management Console

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.aws.amazon.com/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。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。
4. 选择创建。
5. 在数据来源创建页面上，输入新数据来源的名称。在数据来源类型下，选择 Amazon S3。选择一个对在 AWS Glue Data Catalog 和 Amazon S3 中可以访问的内容有限制的现有 IAM 角色。
6. 选择创建。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/Admin"
        }
    }
```

```
    "Description": "data-source-description",
    "Name": "my-data-source"
}
```

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，则这些权限包括示例策略中的最低权限权限。

集成需要访问权限才能写入 Amazon S3 和 AWS Glue Data Catalog。对于 Amazon S3，我们需要写入权限才能在构建加速时维护检查点位置。因为 AWS Glue Data Catalog，我们需要写入权限才能从 S OpenSearch ervice 内部管理集成所需的数据库、表和分区。

```
{
   "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"
         ],
         "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:::<checkpoint_bucket_name>",
            "arn:aws:s3:::<checkpoint_bucket_name>/*"
         ]
      }
   ]
}
```

要在不同的账户中支持 Amazon S3 存储桶，您需要在 Amazon S3 策略中加入一个条件并添加相应的账户。

```
"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。

```
{
    "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。新的细粒度访问控制角色的名称将是。AWS OpenSearchDirectQuery *<name of data source>*

默认情况下，该角色只能访问直接查询数据源索引。尽管您可以将角色配置为限制或授予对数据源的访问权限，但建议您不要调整此角色的访问权限。如果您删除数据源，则该角色将被删除。如果任何其他用户被映射到该角色，则这将删除他们的访问权限。

映射 AWS Glue Data Catalog 角色（如果在创建数据源后启用了细粒度访问控制）

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 AWS Glue Data Catalog 访问权限的 IAM 角色才能运行直接查询。要手动创建可映射到 IAM 角色的后端 glue_access 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 仪表板的主菜单中，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>*
   - 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 AWS 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”。

后续步骤

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控制面板以及查询数据。

在 OpenSearch 仪表板中配置数据源

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。

设置访问控制

在数据源的详细信息页面上，找到 “访问控制” 部分，然后选择 “编辑”。如果您安装了安全插件，请选择受限，然后选择要向哪些基于角色的组提供对新数据来源的访问权限。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。

为常用 AWS 日志类型设置集成

OpenSearch 通过控制面板，您可以使用原始日志轻松快速开始使用存储在 Amazon S3 中的常见日志类型，但 Parquet 格式支持的 Amazon VPC 流日志除外。 OpenSearch 仪表板提供的集成功能

可以安装对 AWS Glue Data Catalog 表格、已保存的查询和仪表板等资产的访问权限。这些资产由 OpenSearch 加速功能提供支持，在你安装后会自动更新。您可以从数据源详细信息页面或左侧导航栏中设置集成。要实现此目的，应按照以下步骤进行：

1. 选择要安装的日志类型。确保您安装的日志类型具有 Amazon S3 标签。
2. 如果尚未选择，请选择连接类型作为 Amazon S3 连接。
3. 根据您的用例，选择要安装集成的数据源名称、数据的 Amazon S3 位置、用于保持加速索引状态的检查点以及所需的资产。

> **Note**
>
> 创建 IAM 角色时，您为具有检查点位置写入操作权限的检查点指定了 Amazon S3 资源。您需要引用对检查点位置具有写入权限的 Amazon S3 存储桶位置。否则，集成将安装的加速将失败。

> **Note**
>
> Amazon VPC 流日志集成需要使用 OpenSearch 控制面板安装补丁。填充已安装的仪表板可能需要几分钟。

将数据导出到 Amazon S3 的参考指南

您可以使用以下参考指南将数据导出到 Amazon S3：

源：

- Apache 访问权限
- CloudFront
- CloudTrail

- Elastic Load Balancing
- Amazon S3
- AWS WAF
- 亚马逊 VPC 流程

- NGINX

使用查询工作台创建 Spark 表

从 OpenSearch 服务直接查询到 Amazon S3 使用中的 Spark 表 AWS 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。

要为以 Parquet 格式存储在 S3 中的 VPC 流日志设置表，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
```

加速查询

在数据来源详细信息页面上，选择加速性能选项。为了确保在 Amazon S3 中快速使用数据，您可以设置三种不同的加速方式来将数据编入 OpenSearch 服务索引，即跳过索引、物化视图和覆盖索引。

跳过索引

使用跳过索引，您只能为 Amazon S3 中所存储数据的元数据编制索引。查询带有跳过索引的表时，查询计划程序会引用该索引并重写查询以有效地定位数据，而不是扫描所有分区和文件。这使跳过索引可以快速缩小存储数据具体位置的范围。

在数据源详细信息页面中，选择加速性能，您可以从中选择要加速的数据库和表开始使用。或者，您可以选择自动生成跳过的索引。如果您更喜欢手动添加要加速的字段，则可以通过选择 “添加字段” 按钮来实现。添加字段时，系统会询问您要添加哪种类型的跳过索引。您需要从以下选项中进行选择：

- 分区：使用数据分区详细信息来查找数据（最适合基于分区的列，例如年、月、日、小时）
- MinMax：使用索引列的下限和上限来定位数据（最适合数字列）
- ValueSet：使用唯一值集来定位数据（最适合基数为中低且需要精确匹配的列）
- BloomFilter：使用布隆过滤器来定位数据（最适合基数较高且不需要精确匹配的列）

您也可以使用 Query Workbench 在表上手动创建跳过的索引。只需从数据源下拉列表中选择 S3 数据源并添加以下查询即可：

```
CREATE SKIPPING INDEX
ON datasourcename.gluedatabasename.vpclogstable(
 `srcaddr` BLOOM_FILTER,
 `dstaddr` BLOOM_FILTER,
 `day` PARTITION,
 `account_id`BLOOM_FILTER
 ) WITH (
index_settings = '{"number_of_shards":5,"number_of_replicas":1}',
auto_refresh = true,
checkpoint_location = 's3://accountnum-vpcflow/AWSLogs/checkpoint'
)
```

实体化视图

借助实体化视图，您可以使用复杂的查询（例如聚合）为控制面板可视化提供支持。根据查询的不同，物化视图会将您的少量数据摄入到 OpenSearch ServiceStorage 中。 OpenSearch 然后，服务会从摄取的数据中形成一个索引，您可以将其用于可视化。您可以使用管理实例化视图索引the section called “索引状态管理”，就像管理任何其他 OpenSearch 索引一样。

由于您将指定目标索引，因此系统会要求您命名该索引并添加水印延迟，该延迟定义了数据可以进入并仍被处理的延迟时间。

使用以下查询为您在中创建的 VPC 流日志表创建新的实体化视图：the section called “使用查询工作台创建 Spark 表”

```
CREATE MATERIALIZED VIEW {table_name}__week_live_mview AS
  SELECT
```

```
    cloud.account_uid AS `aws.vpc.cloud_account_uid`,
    cloud.region AS `aws.vpc.cloud_region`,
    cloud.zone AS `aws.vpc.cloud_zone`,
    cloud.provider AS `aws.vpc.cloud_provider`,

    CAST(IFNULL(src_endpoint.port, 0) AS LONG) AS `aws.vpc.srcport`,
    CAST(IFNULL(src_endpoint.svc_name, 'Unknown') AS STRING)  AS `aws.vpc.pkt-src-aws-
service`,
    CAST(IFNULL(src_endpoint.ip, '0.0.0.0') AS STRING)  AS `aws.vpc.srcaddr`,
    CAST(IFNULL(src_endpoint.interface_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-
interface_uid`,
    CAST(IFNULL(src_endpoint.vpc_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-vpc_uid`,
    CAST(IFNULL(src_endpoint.instance_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-
instance_uid`,
    CAST(IFNULL(src_endpoint.subnet_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-
subnet_uid`,

    CAST(IFNULL(dst_endpoint.port, 0) AS LONG) AS `aws.vpc.dstport`,
    CAST(IFNULL(dst_endpoint.svc_name, 'Unknown') AS STRING) AS `aws.vpc.pkt-dst-aws-
service`,
    CAST(IFNULL(dst_endpoint.ip, '0.0.0.0') AS STRING)  AS `aws.vpc.dstaddr`,
    CAST(IFNULL(dst_endpoint.interface_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-
interface_uid`,
    CAST(IFNULL(dst_endpoint.vpc_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-vpc_uid`,
    CAST(IFNULL(dst_endpoint.instance_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-
instance_uid`,
    CAST(IFNULL(dst_endpoint.subnet_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-
subnet_uid`,
    CASE
      WHEN regexp(dst_endpoint.ip, '(10\\..*)|(192\\.168\\..*)|(172\\.1[6-9]\\..*)|
(172\\.2[0-9]\\..*)|(172\\.3[0-1]\\.*)')
        THEN 'ingress'
      ELSE 'egress'
      END AS `aws.vpc.flow-direction`,

    CAST(IFNULL(connection_info['protocol_num'], 0) AS INT) AS
 `aws.vpc.connection.protocol_num`,
    CAST(IFNULL(connection_info['tcp_flags'], '0') AS STRING)  AS
 `aws.vpc.connection.tcp_flags`,
    CAST(IFNULL(connection_info['protocol_ver'], '0') AS STRING)  AS
 `aws.vpc.connection.protocol_ver`,
    CAST(IFNULL(connection_info['boundary'], 'Unknown') AS STRING)  AS
 `aws.vpc.connection.boundary`,
```

```
    CAST(IFNULL(connection_info['direction'], 'Unknown') AS STRING)  AS
 `aws.vpc.connection.direction`,

    CAST(IFNULL(traffic.packets, 0) AS LONG) AS `aws.vpc.packets`,
    CAST(IFNULL(traffic.bytes, 0) AS LONG) AS `aws.vpc.bytes`,

    CAST(FROM_UNIXTIME(time / 1000) AS TIMESTAMP) AS `@timestamp`,
    CAST(FROM_UNIXTIME(start_time / 1000) AS TIMESTAMP) AS `start_time`,
    CAST(FROM_UNIXTIME(start_time / 1000) AS TIMESTAMP) AS `interval_start_time`,
    CAST(FROM_UNIXTIME(end_time / 1000) AS TIMESTAMP) AS `end_time`,
    status_code AS `aws.vpc.status_code`,

    severity AS `aws.vpc.severity`,
    class_name AS `aws.vpc.class_name`,
    category_name AS `aws.vpc.category_name`,
    activity_name AS `aws.vpc.activity_name`,
    disposition AS `aws.vpc.disposition`,
    type_name AS `aws.vpc.type_name`,

    region AS `aws.vpc.region`,
    accountid AS `aws.vpc.account-id`
  FROM
  datasourcename.gluedatabasename.vpclogstable
WITH (
  auto_refresh = true,
  refresh_interval = '15 Minute',
  checkpoint_location = 's3://accountnum-vpcflow/AWSLogs/checkpoint',
  watermark_delay = '1 Minute',
)
```

覆盖索引

使用覆盖索引，您可以从表中的指定列摄取数据。这是三种索引类型中性能最高的一种。由于 S OpenSearch ervice 会从所需列中提取所有数据，因此您可以获得更好的性能并可以执行高级分析。

与实例化视图一样，S OpenSearch ervice 会根据覆盖索引数据创建新索引。您可以将此新索引用于仪表板可视化和其他 OpenSearch 服务功能，例如异常检测或地理空间功能。您可以使用管理覆盖视图索引the section called “索引状态管理”，就像管理任何其他 OpenSearch 索引一样。

使用以下查询为您在中创建的 VPC 流日志表创建新的覆盖索引the section called “使用查询工作台创建 Spark 表”：

```
CREATE INDEX vpc_covering_index
```

```
ON datasourcename.gluedatabasename.vpclogstable (version, account_id, interface_id,
srcaddr, dstaddr, srcport, dstport, protocol, packets,
bytes, start, action, log_status STRING,
`aws-account-id`, `aws-service`, `aws-region`, year,
month, day, hour )
WITH (
  auto_refresh = true,
  refresh_interval = '15 minute',
  checkpoint_location = 's3://accountnum-vpcflow/AWSLogs/checkpoint'
)
```

在 OpenSearch 仪表板中查询数据

设置表并配置所需的可选查询加速后，您现在可以开始对数据执行分析。要查询您的数据，请从 OpenSearch 仪表板的 “发现” 页面或 “可观察性” 页面的下拉菜单中选择数据源。

如果您使用跳过索引或尚未创建索引，则可以使用 SQL 或管道处理语言（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。

SQL

使用以下查询对您在中创建的 VPC 流日志表运行示例 SQL 查询the section called “使用查询工作台创建 Spark 表”：

```
SELECT srcaddr, SUM (CAST(bytes AS LONG)) as total_bytes
 FROM datasourcename.gluedatabasename.vpclogstable GROUP BY srcaddrORDER BY total_bytes
 DESCLIMIT 10;
```

PPL

使用以下查询对您在中创建的 VPC 日志表运行 PPL 查询示例：the section called “使用查询工作台创建 Spark 表”

```
source = datasourcename.gluedatabasename.vpclogstable | fields account_id, srcaddr,
 dstaddr, action | head 10
```

建议

在某些情况下，结果可能未按预期返回。如果您遇到任何问题，我们建议您采取以下措施：

- SELECT* 语句不返回任何结果-请检查您的表以查看它是否有需要分解的嵌套 struc 列。
- 选择多个表时，使用SQL UNION语句来引用多个表。
- 加速设置为使用特定数量的工作线程来执行查询。如果查询返回缓慢，则可以手动分配更多的工作线程来执行查询，以提高性能。
- 在构建跳过索引时，使用布隆过滤器来获得高基数，使用最小/最大值来表示大范围，以节省域上的空间。如果您需要执行精确匹配，建议您在中等基数字段上设置值。
- 有关常用 SQL 查询的更多信息，请参阅AWS 服务日志。

管理数据源

管理数据源是维护直接查询数据源和其他 AWS 解决方案的可靠性、可用性和性能的重要组成部分。AWS 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
- 启用和禁用数据源
- 用 AWS 预算进行监控
- OpenSearch 使用亚马逊 S3 删除亚马逊服务数据源

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。

您还可以设置警报以监控某些阈值，并在达到这些阈值时发送通知或采取行动。有关更多信息，请参阅什么是亚马逊 CloudWatch。

直接查询报告以下指标：

指标	描述
AsyncQueryCreateAPI	向 API 发出的用于创建异步查询的请求总数。 相关统计数据： 平均值、最大值、总和 尺寸：`ClientId`，`DomainName` 频率：60 秒
AsyncQueryGetApiRequestCount	向 API 发出的用于检索异步查询结果的请求总数。 相关统计数据： 平均值、最大值、总和 尺寸：`ClientId`，`DomainName` 频率：60 秒
AsyncQueryCancelApiRequestCount	向 API 发出的取消异步查询的请求总数。 相关统计数据： 平均值、最大值、总和 尺寸：`ClientId`，`DomainName` 频率：60 秒
A syncQueryGet ApiFailed RequestCusErrCount	检索异步查询结果时由于与客户相关的错误（例如，无效的查询 ID）而失败的请求数。 相关统计数据： 平均值、最大值、总和 尺寸：`ClientId`，`DomainName` 频率：60 秒

指标	描述
AsyncQueryCancelApiFailedRequestCusErrCount	检索异步查询结果时由于与客户相关的错误（例如，无效的查询 ID）而失败的请求数。 相关统计数据：平均值、最大值、总和 尺寸：ClientId，DomainName 频率：60 秒
AsyncQueryCancelApiFailedRequestSysErrCount	创建异步查询时由于与客户相关的错误而失败的请求数。 相关统计数据：平均值、最大值、总计 尺寸：ClientId，DomainName 频率：60 秒
A syncQueryGet ApiFailed RequestSysErrCount	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 尺寸：ClientId，DomainName 频率：60 秒

启用和禁用数据源

如果您想停止使用直接查询数据源，则可以选择禁用该数据源。禁用数据源将完成现有查询的执行并停止用户执行所有新查询。

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。数据源在禁用后设置为活动状态后，用户查询将按预期运行。先前设置并设置为手动的加速需要手动配置才能再次按计划运行。

用 AWS 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。客户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）。想要在 DirectQuery OCU（OCU 小时数）使用量达到阈值时收到通知的客户可以创建一个 AWS 预算账户，并根据他们设置的阈值配置提醒。或者，客户可以选择设置一个 Amazon SNS 主题，该主题将在满足阈值标准时关闭数据源。

> **Note**
>
> AWS 预算中的使用数据不是实时的，最多可能会延迟 8 小时。

OpenSearch 使用亚马逊 S3 删除亚马逊服务数据源

当您删除数据源时，亚马逊 OpenSearch 服务会将其从您的域中删除。 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 Amazon S3 中删除，但是 Amazon S3 不会向 OpenSearch 服务发送新数据。

您可以使用 AWS Management Console 或 OpenSearch 服务 API 删除数据源集成。

AWS Management Console

删除数据来源

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.aws.amazon.com/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。

OpenSearch 服务 API

使用 DeleteDataSourceAPI 操作删除您网域中的现有数据源。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource/data-source-name
```

监控 Amazon OpenSearch Service 域

监控是保持 Amazon OpenSearch Service 和您的其他 AWS 解决方案的可靠性、可用性和性能的重要方面。AWS 提供了以下工具来监控您的 OpenSearch Service 资源、报告问题并适时自动采取措施：

Amazon CloudWatch

Amazon CloudWatch 实时监控您的 OpenSearch Service 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 Logs

Amazon CloudWatch Logs 允许您监控、存储和访问您的 OpenSearch 日志文件。CloudWatch Logs 监控日志文件中的信息，并可在达到特定阈值时通知您。有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南。

Amazon EventBridge

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 Service 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时在其他 AWS 服务中触发自动操作。有关更多信息，请参阅 Amazon EventBridge 用户指南。

AWS CloudTrail

AWS CloudTrail 捕获对 OpenSearch Service 进行的配置 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 AWS CloudTrail 用户指南。

主题

使用 Amazon 监控 OpenSearch 集群指标 CloudWatch

亚马逊 OpenSearch 服务会将您的域名中的数据发布到亚马逊 CloudWatch。 CloudWatch 允许您以一组有序的时间序列数据（称为指标）的形式检索有关这些数据点的统计信息。 OpenSearch 服务以 60 秒为间隔向 CloudWatch 发送大多数指标。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。所有累积指标（例如`ThreadpoolSearchRejected`）都在内存中`ThreadpoolWriteRejeceted`，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，指标将重置。有关亚马逊的更多信息 CloudWatch，请参阅亚马逊 CloudWatch 用户指南。

OpenSearch 服务控制台根据来自的原始数据显示一系列图表 CloudWatch。根据您的需求，您可能更喜欢在中查看集群数据， CloudWatch 而不是在控制台中查看图表。该服务会将指标存档两周，然后再丢弃。这些指标不收取额外费用，但创建仪表板和警报 CloudWatch 仍会收费。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 “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 指标首先按服务命名空间分组，然后按每个命名空间内的各种维度组合进行分组。

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。

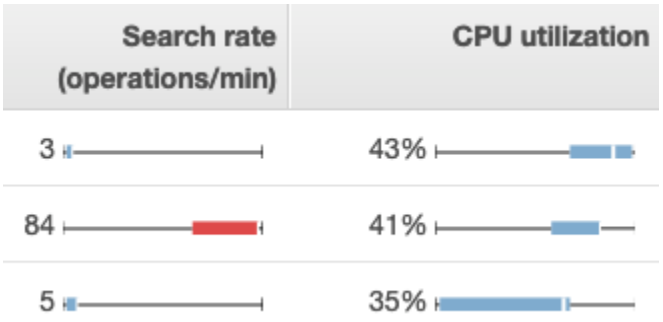
要查看指标列表，请使用 AWS CLI

运行以下命令：

```
aws cloudwatch list-metrics --namespace "AWS/ES"
```

在 S OpenSearch ervice 中解释健康图表

要在 S OpenSearch ervice 中查看指标，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方框图 at-a-glance 来查看每个 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 OpenSearch 服务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
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。

指标	描述
	相关统计数据：总计
Nodes	OpenSearch 服务集群中的节点数量，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。这些文档不再出现在搜索结果中， OpenSearch 只会在段合并期间从磁盘中删除已删除的文档。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 为单位显示它。 **Note** FreeStorageSpace 将始终低于 OpenSearch _cluster/stats 和 _cat/allocation API 提供的值。 OpenSearch Service 会在每个实例上预留一定比例的存储空间用于内部操作。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 为单位显示它。 相关统计数据：Minimum、Maximum
ClusterIndexWritesBlocked	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 JVMMemoryPressure 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 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。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 开发人员指南中的 CPU 组。此指标仅对 T2 实例类型有效。 相关统计数据：Minimum
OpenSearchDashboardsHealthyNodes	OpenSearch 仪表板的运行状况检查。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失败的生成 OpenSearch 仪表板报告的请求数。 相关统计数据：总计
OpensearchDashboardsReportingFailedRequestUserErrCount	由于客户端问题而失败的生成 OpenSearch 仪表板报告的请求数。 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	成功请求生成 OpenSearch 仪表板报告的次数。 相关统计数据：总计
KMSKeyError	值为 1 表示用于加密静态数据的密 AWS KMS 钥已被禁用。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	值为 1 表示用于加密静态数据的 AWS KMS 密钥已被删除或撤销其对 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	向 OpenSearch 集群发出的包含无效（或缺失）主机标头的 HTTP 请求数。有效的请求包括域主机名作为主机标头值。 OpenSearch 对于没有限制性访问策略的公共访问域，Service 会拒绝无效请求。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值很大，请确认您的 OpenSearch 客户端在其请求中包含域主机名（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	表示域上的每秒输入/输出操作数 (IOPS) 是否已受到限制。当数据节点的 IOPS 违反数据节点的 EBS 卷或 EC2 实例的最大允许限制时，就会发生限制。 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

专用主节点指标

Amazon OpenSearch 服务为专用主节点提供以下指标。

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 **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

指标	描述
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 开发人员指南中的 CPU 组。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。它们通常是网络连接问题或 AWS 依赖问题造成的。 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

EBS 卷指标

Amazon OpenSearch 服务为 EBS 卷提供了以下指标。

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

指标	描述
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。 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

指标	描述
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
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

实例指标

Amazon OpenSearch 服务为域中的每个实例提供以下指标。 OpenSearch 服务还会汇总这些实例指标，以深入了解集群的整体运行状况。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 5 x、6.0 和 6.2 使用批量线程池。 OpenSearch 而 Elasticsearch 6.3 及更高版本则使用写线程池。目前， OpenSearch 服务控制台不包含批量线程池的图表。
> 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
ConcurrentSearchRate	对数据节点上的所有分片使用每分钟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

指标	描述
	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	在分钟 N 和分钟 (N-1) 之间的节点中使用并发分段搜索进行的所有搜索所得的总时间差，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位于没有优化实例的OpenSearch域上，则集群中的其他节点也会记录总共四次索引操作。对于具有优化实例的 OpenSearch 域，具有副本的其他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
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

指标	描述
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。您拥有的区段越多，每次搜索所需的时间就越长。 OpenSearch 偶尔会将较小的片段合并为一个较大的片段。 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
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

指标	描述
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	与 OpenSearch仪表板的活跃并发连接数。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 仪表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 仪表板的堆内存量，以 MiB 为单位。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
OpenSearchDashboardsHeapUsed	OpenSearch 仪表板使用的绝对堆内存量，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

指标	描述
OpenSearchDashboardsHeapUtilization	OpenSearch仪表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 仪表板一分钟 CPU 平均负载。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 请求总数。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 仪表板响应请求所需的最大时间（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
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

指标	描述
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
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

指标	描述
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
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

指标	描述
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
ThreadpoolIndexSearcherRejected	索引搜索器线程池中被拒绝的任务数。 相关节点统计数据：Maximum 相关集群统计数据：Sum

指标	描述
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** 由于在 7.1 版本中，默认写入队列大小从 200 增加到 10000，因此该指标不再是服务拒绝的唯一指标。 OpenSearch 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	自上次启动 OpenSearch 服务进程以来，由于索引压力，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	自上次启动 OpenSearch 服务进程以来，由于索引压力，主分片上发生的拒绝总数。 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	自上次启动 OpenSearch 服务进程以来，由于索引压力，副本分片上发生的拒绝总数。 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。

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。平均值表示集群中的所有 UltraWarm 节点。此指标也适用于单个 UltraWarm 节点。 相关统计数据：Maximum、Average

指标	描述
WarmFreeStorageSpace	以 MiB 为单位的可用温存储空间量。因为 UltraWarm使用 Amazon S3 而不是连接的磁盘，所以Sum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
WarmSearchLatency	在 N 分钟到分钟 (N-1) UltraWarm 之间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
WarmSearchRate	UltraWarm 节点上所有分片每分钟搜索请求的总数。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

指标	描述
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
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

指标	描述
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
WarmThreadpoolSearchRejected	UltraWarm 搜索线程池中被拒绝的任务数。如果这个数字持续增长，可以考虑添加更多 UltraWarm节点。 相关节点统计数据：Maximum 相关集群统计数据：Sum
WarmThreadpoolSearchQueue	UltraWarm 搜索线程池中排队的任务数。如果队列大小一直很高，可以考虑添加更多 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 节点用于 “旧一代” 的 Java 堆的最大百分比。 相关统计数据：Maximum

指标	描述
WarmJVMGCYoungCollectionCount	“年轻一代” 垃圾收集在 UltraWarm 节点上运行的次数。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
WarmJVMGCYoungCollectionTime	集群在节点上执行 “年轻一代” 垃圾收集所花费的时间，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	“老一代” 垃圾收集在 UltraWarm 节点上运行的次数。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	使用每分钟并发分段搜索的 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：Sum、Maximum、Average
WarmConcurrentSearchLatency	在分钟 N 和分钟 (N-1) 之间的 UltraWarm 节点中使用并发分段搜索进行的所有搜索所得的总时间差，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

指标	描述
WarmThreadpoolIndexSearcherQueue	UltraWarm 索引搜索器线程池中排队的任务数。 相关节点统计数据：Maximum 相关集群统计数据：Sum、Maximum、Average
WarmThreadpoolIndexSearcherRejected	UltraWarm 索引搜索器线程池中被拒绝的任务数。 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 相关集群统计信息：总和、平均值

冷存储指标

Amazon OpenSearch 服务提供以下冷存储指标。

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

指标	描述
ColdToWarmMigrationSuccessCount	成功从冷到温迁移的总数。 相关统计数据：总计
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

亚马逊 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。这是从上次启动 OpenSearch 服务进程开始计算的。 相关统计数据：总计

提醒指标

Amazon OpenSearch 服务提供以下警报指标。

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 OpenSearch 服务控制台或_nodes/stats 请求查看是否有任何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 OpenSearch 服务提供以下异常检测指标。

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 OpenSearch 服务为异步搜索提供了以下指标。

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

亚马逊 OpenSearch 服务为自动调整提供了以下指标。

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 为带备用模式的多可用区提供了以下指标。

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 为单位显示它。
JVMMemoryPressure	用于集群中所有数据节点的 Java 堆的最大百分比。 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。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

亚马逊 OpenSearch 服务为时间点 (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

亚马逊 OpenSearch 服务为 SQL 支持提供了以下指标。

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，为了响应某些请求，SQL 插件正在返回 5 xx 响应代码或向 OpenSearch传递无效的查询 DSL。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

亚马逊 OpenSearch 服务包含 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 OpenSearch 服务为跨集群搜索提供了以下指标。

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 OpenSearch 服务为跨集群复制提供了以下指标。

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 OpenSearch 服务为学习排名提供了以下指标。

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）。
LTRFeaturesetMemoryUsageInBytes	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 OpenSearch 服务为管道处理语言提供了以下指标。

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
- 搜索请求慢日志
- 分片慢日志
- 审核日志

搜索分片慢日志、索引分片慢日志和错误日志对于解决性能和稳定性问题非常有用。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

主题

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. 转至 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：

```
{
  "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 (AWS 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。

如果您启用了其中一个分片慢日志，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 (AWS 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

以下示例允许发布指定域的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`。

如果您启用了其中一个分片慢日志，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 (AWS SDK)

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 AWS SDK 访问这些操作。

AWS 软件开发工具包（Android 和 iOS 软件开发工具包除外）支持《亚马逊 OpenSearch 服务 API 参考》中定义的所有操作，包括和`--log-publishing-options`选项。`CreateDomain` `UpdateDomainConfig`

如果您启用了其中一个分片慢日志，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 AWS 账户 向该域发出所有 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 AWS 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花费时间配置的。这与分片请求慢日志不同，后者是为单个分片花费时间配置的。

您可以使用集群设置指定搜索请求慢日志。这与分片慢速日志不同，后者是通过索引设置启用的。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 AWS CLI `describe-domain-config`选项，或`DescribeDomainConfig`使用其中一个 SDK 调用。
- OpenSearch 日志阈值是否足够低，以至于您的请求超出了这些阈值？

  要查看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
  ```

  要查看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 AWS CLI 不会停止生成 OpenSearch 日志；它只会停止发布这些日志。如果您不再需要分片慢日志，请务必检查您的索引设置；如果您不再需要搜索请求慢日志，请务必检查您的域设置。

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

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。AWS 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：

```
{
  "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 AWS CLI

以下 AWS 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 AWS 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

类别	描述	REST 可用	可用于传输
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。
COMPLIANCE_INTERNAL_CONFIG_READ	在 `.opendistro_security` 索引中执行读取事件的请求。

类别	描述
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。
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。

名称	后端设置	描述
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

亚马逊 OpenSearch 服务与亚马逊集成 EventBridge ，可通知您某些影响您域名的事件。来自 AWS 服务的事件几乎实时 EventBridge 地传送到。同样的事件也会发送到亚马逊 CloudWatch 的前身 Amazon 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 AWS Lambda
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams
- 激活 AWS 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
- 降级节点停用事件
- 域错误事件
- 教程：监听亚马逊 OpenSearch 服务 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 当发生以下服务软件更新事件之一 EventBridge 时，服务会将事件发送到。

可用服务软件更新

OpenSearch 当服务软件更新可用时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 服务会在安排服务软件更新时发送此事件。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": "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 服务软件更新开始后，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": "Started",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] started.
  }
}
```

服务软件更新已完成

OpenSearch 服务软件更新完成后，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": "Completed",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] completed."
  }
}
```

已取消服务软件更新

OpenSearch 服务软件更新取消后，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 as a
                    newer update is available. Please schedule the latest update."
  }
}
```

已取消计划服务软件更新

OpenSearch 当先前为该域安排的服务软件更新被取消时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 服务在无法启动服务软件更新时会发送此事件。

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": "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 当服务软件更新失败时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 当 Auto-Tune 确定了改善集群性能和可用性的调整建议时，服务会发送此事件。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 当 Auto-Tune 开始对您的域应用新设置时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 当 Auto-Tune 确定了需要计划蓝/绿部署的调整建议时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 由于没有待处理的调整建议而取消了 Auto-Tune 计划时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 当 Auto-Tune 完成蓝/绿部署并且集群在设置了新 JVM 设置的情况下可以运行时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 当 Auto-Tune 被禁用并且应用的更改已回滚时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
```

```
  "version": "0",
  "id": "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 当禁用 Auto-Tune 并且保留了应用的更改时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 EventBridge 当您的集群的运行状况受到威胁时，服务会向发送某些事件。

红色集群恢复已开始

OpenSearch 在您的集群状态持续变为红色超过一个小时后，服务会发送此事件。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 当服务在尝试修复红色群集状态时只能从快照中恢复部分红色索引时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
   "version":"0",
   "id":"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 如果服务在尝试修复红色群集状态时未能恢复任何索引，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 如果服务在您的红色群集状态持续变为红色 14 天后尝试自动修复该状态，但一个或多个索引仍为红色，则服务会发送此事件。再过 7 天（总共 21 天持续变为红色）后，S OpenSearch ervic e 继续删除所有红色索引上未分配的分片。

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 在您的集群状态持续变为红色 21 天后，服务会发送此事件。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 当您的热数据节点的平均分片数超过建议的默认限制（1,000）的 90% 时，服务会发送此事件。尽管更高版本的 Elasticsearch OpenSearch 支持可配置的每个节点的最大分片数限制，但我们建议每个节点的分片数不超过 1,000 个。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 当您的热数据节点的平均分片数超过建议的默认限制 1,000 时，服务会发送此事件。尽管更高版本的 Elasticsearch OpenSearch 支持可配置的每个节点的最大分片数限制，但我们建议每个节点的分片数不超过 1,000 个。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 当集群中的一个或多个节点的可用存储空间少于 25% 或小于 25 GB 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":"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 当集群中所有节点的可用存储空间低于 10% 或小于 10 GB 时，服务会发送此事件。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 突发平衡降至 70% 以下时，服务会发送此事件。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 突发平衡降至 20% 以下时，服务会发送此事件。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 由于您的 EBS 卷或 EC2 实例的吞吐量限制，对您的域的读取和写入请求受到限制时，服务会发送此事件。如果您收到此通知，请考虑按照 AWS 推荐的最佳实践扩展您的卷或实例。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 65GiB 时，服务会发送此事件。为确保最佳的集群性能和稳定性，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 当您的域名的JVMMemoryPressure指标超过 80% 时，服务会发送此事件。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 当最大 JVM 大于 70% 且最大值和最小值之差小于 30% 时，服务会发送此事件。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并且包含带有自定义 index.routing.allocation 设置的索引时，服务会发送此事件，这可能会导致蓝绿色部署卡住。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 当您的域名由于未分配分片而导致运行状况不佳时，服务会发送此事件。[ShardLockObtainFailedException]有关更多信息，请参阅[如何解决 Ama OpenSearch zon Service 中的内存分片锁定异常](#)？

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[AWS PrivateLink 接口端点](#)相关的事件。

VPC 端点创建失败

OpenSearch 服务在无法创建请求的 VPC 终端节点时发送此事件。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 服务在无法删除请求的 VPC 终端节点时发送此事件。

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 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 服务在无法删除请求的 VPC 终端节点时发送此事件。

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 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 当发生以下节点停用事件之一 EventBridge 时，服务会向发送事件。

已计划停用节点

OpenSearch 在安排节点停用时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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 节点停用完成后，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 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 当节点停用失败时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
  "version": "0",
  "id": "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."
```

```
  }
}
```

降级节点停用事件

OpenSearch 当由于节点上的硬件降级而需要更换节点时，服务会发送这些事件。

降级节点停用通知

OpenSearch 当已为你的域安排了停用和替换降级节点的自动操作时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
   "version":"0",
   "id":"db233454-aad1-7676-3b15-10a84b052baa",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2024-01-11T08:16:06Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
   ],
   "detail":{
      "severity":"Medium",
      "description":"An automated action to retire and replace a node has
 been scheduled on your domain. For more information, please see https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/monitoring-events.html.",
      "event":"Degraded Node Retirement Notification",
      "status":"Scheduled"
   }
}
```

降级节点停用已完成

OpenSearch 当降级的节点已停用并替换为新节点时，服务会发送此事件。

示例

以下是该类型的示例事件：

```
{
   "version":"0",
   "id":"7444215c-90f9-a52d-bcda-e85973a9a762",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2024-01-11T10:20:30Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
   ],
   "detail":{
      "severity":"Medium",
      "description":"The node has been retired and replaced with a new node.",
      "event":"Degraded Node Retirement Notification",
      "status":"Completed"
   }
}
```


降级节点停用失败

OpenSearch 如果降级节点停用失败，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 当出现以下域错误之一 EventBridge 时，服务会向发送事件。

域更新验证故障

OpenSearch 如果服务在尝试更新域或对域执行配置更改时遇到一个或多个验证失败，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 服务在无法访问您的 AWS 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 当您的域名变得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，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 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：监听亚马逊 OpenSearch 服务 EventBridge 事件

在本教程中，您设置了一个用于监听 Amazon S OpenSearch ervice 事件并将其写入 CloudWatch 日志流的简单 AWS Lambda 函数。

先决条件

本教程假设您已有 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 函数作为 OpenSearch 服务事件消息的目标。

创建目标 Lambda 函数

1. 打开 AWS 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，用于打印 OpenSearch 服务发送的事件。如果所有配置都正确，则在本教程结束时，事件详细信息将显示在与此 Lambda 函数关联的 CloudWatch 日志流中。

7. 选择部署。

步骤 2：注册事件规则

在此步骤中，您将创建一条 EventBridge 规则，用于捕获来自您的 OpenSearch 服务域的事件。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AWS 服务、Amazon OpenSearch 服务和所有事件。这种模式适用于您的所有 OpenSearch 服务域和每个 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 OpenSearch 服务控制台的 “通知” 部分收到通知时，如果一切配置正确，则会触发您的 Lambda 函数，并将事件数据写入该函数的 CloudWatch 日志流中。

测试配置

1. 打开 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/。

2. 在导航窗格中，选择日志，然后选择 Lambda 函数的日志组（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，您将配置亚马逊 EventBridge 事件规则，该规则用于捕获亚马逊服务中可用服务软件更新的通知，并通过 Amazon Simple Notification Service (A OpenSearch mazon SNS) 向您发送电子邮件通知。

先决条件

本教程假设您已有 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. 通过 https://console.aws.amazon.com/sns/v3/home 打开 Amazon SNS 控制台。
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AWS 服务、亚马逊 OpenSearch 服务和亚马逊 OpenSearch 服务软件更新通知。此模式与来自 OpenSearch 服务的任何服务软件更新事件相匹配。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）。

下次您收到服务部门关于可用 OpenSearch 服务软件更新的通知时，如果一切配置正确，Amazon SNS 应向您发送有关该更新的电子邮件提醒。

使用 AWS CloudTrail 监控 Amazon OpenSearch Service API 调用

Amazon OpenSearch Service 与 AWS CloudTrail 集成，后者是在 OpenSearch Service 中提供用户、角色或 AWS 服务所采取操作的记录的服务。CloudTrail 以事件形式捕获 OpenSearch 服务的所有配置 API 调用。

> **Note**
>
> CloudTrail 仅捕获对配置 API 之外的调用，例如 `CreateDomain` 和 `GetUpgradeStatus`。CloudTrail 不会捕获对 OpenSearch API之外的压缩算法，例如 `_search` 和 `_bulk`。有关这些调用，请参阅 the section called “监控审计日志”。

捕获的调用包括来自 OpenSearch Service 控制台、AWS CLI 或 AWS 开发工具包的调用。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 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history (事件历史记录) 中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 OpenSearch Service 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。

要了解有关 CloudTrail 的更多信息，请参阅《AWS CloudTrail 用户指南》。

CloudTrail 中的 Amazon OpenTrail 信息

在您创建 AWS 账户 时，将在该账户上启用 CloudTrail。当 OpenSearch Service 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在事件历史记录中。您可以在 AWS 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 CloudTrail 事件历史记录查看事件。

要持续记录 AWS 账户 账户中的事件（包括 OpenSearch Service 事件），请创建跟踪。通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service（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AWS 区域。此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Simple Storage Service（Amazon S3）桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- 为您的 AWS 账户 创建跟踪
- AWS 服务与 CloudTrail Logs 的集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

所有 OpenSearch Service 配置 API 操作都由 CloudTrail 记录，并且在 Amazon OpenSearch Service API 参考中正式记载。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 AWS Identity and Access Management (IAM) 用户凭证发出的
- 请求是使用角色还是联合身份用户的临时安全凭证发出的
- 请求是否由其它 AWS 服务发出

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 Amazon OpenSearch Service 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 桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。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 `CreateDomain` 操作。


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/test-user",
    "accountId": "123456789012",
    "accessKeyId": "access-key",
    "userName": "test-user",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-08-21T21:59:11Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2018-08-21T22:00:05Z",
  "eventSource": "es.amazonaws.com",
  "eventName": "CreateDomain",
  "awsRegion": "us-west-1",
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "engineVersion": "OpenSearch_1.0",
    "clusterConfig": {
      "instanceType": "m4.large.search",
      "instanceCount": 1
    },
    "snapshotOptions": {
      "automatedSnapshotStartHour": 0
    },
    "domainName": "test-domain",
    "encryptionAtRestOptions": {},
    "eBSOptions": {
      "eBSEnabled": true,
      "volumeSize": 10,
      "volumeType": "gp2"
    },
    "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}",
```

```
    "advancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    }
  },
  "responseElements": {
    "domainStatus": {
      "created": true,
      "clusterConfig": {
        "zoneAwarenessEnabled": false,
        "instanceType": "m4.large.search",
        "dedicatedMasterEnabled": false,
        "instanceCount": 1
      },
      "cognitoOptions": {
        "enabled": false
      },
      "encryptionAtRestOptions": {
        "enabled": false
      },
      "advancedOptions": {
        "rest.action.multi.allow_explicit_index": "true"
      },
      "upgradeProcessing": false,
      "snapshotOptions": {
        "automatedSnapshotStartHour": 0
      },
      "eBSOptions": {
        "eBSEnabled": true,
        "volumeSize": 10,
        "volumeType": "gp2"
      },
      "engineVersion": "OpenSearch_1.0",
      "processing": true,
      "aRN": "arn:aws:es:us-west-1:123456789012:domain/test-domain",
      "domainId": "123456789012/test-domain",
      "deleted": false,
      "domainName": "test-domain",
      "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":\"es:*\",
\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}"
    }
  },
  "requestID": "12345678-1234-1234-1234-987654321098",
  "eventID": "87654321-4321-4321-4321-987654321098",
```

```
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Amazon OpenSearch 服务中的安全

云安全 AWS 是重中之重。作为 AWS 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 AWS 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 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。 AWS 还为您提供可以安全使用的服务。作为 AWS 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 Amazon OpenSearch 服务的合规计划，请参阅合规计划范围内的AWS 服务。
- 云端安全-您的责任由您使用的 AWS 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 AWS 服务来帮助您监控和保护您的 OpenSearch 服务资源。

主题

Amazon OpenSearch 服务中的数据保护

分 AWS 担责任模式适用于亚马逊 OpenSearch 服务中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础架构上的内容的控制。您还负

责您所使用的 AWS 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。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS 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。我们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）。
- 使用 SSL/TLS 与资源通信。 AWS 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 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 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 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 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 140-2 版》。

我们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、 AWS CLI API 或 AWS SDK AWS 服务 使用 OpenSearch 服务或其他服务时。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 AWS Key Management Service (AWS 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 AWS 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 AWS KMS 密钥加密其 CloudWatch 日志组。有关更多信息，请参阅 Amazon Lo CloudWatch gs 用户指南 AWS KMS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，则必须先将其移至热存储，然后才能禁用 UltraWarm 或冷存储。

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 AWS Key Management Service 开发人员指南中的创建密钥。

无论是否启用了静态加密，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 AWS KMS，例如以下基于身份的策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

如果要使用 AWS 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AWS Key Management Service 开发人员指南中的在 AWS 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 AWS 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 AWS 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。 AWS 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 AWS 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。如需了解更多信息，请参阅 AWS Key Management Service 开发人员指南中的轮换密钥。
- 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。

N 亚马逊 OpenSearch 服务ode-to-node 加密

N ode-to-node 加密在 Amazon OpenSearch 服务的默认功能之上提供了额外的安全层。

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，不会加密 VPC 内的流量。N ode-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、 AWS CLI或配置 API 要求所有进入该域的流量都通过 HTTPS 到达。

如果启用精细访问控制，则需要进行ode-to-node 加密。

启用 node-to-node 加密

N 对新域名进行ode-to-node 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 AWS 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 AWS CLI 或配置 API。有关更多信息，请参阅AWS 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 API。Principal 元素指定账户、用户或允许访问的角色。Resource 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:*)：

```
{
  "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：

```
{
  "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。此策略提供对索引中所有 URI 的 HTTP GET 和 PUT 方法的 power-user-role 访问权限：

```
{
  "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 AWS 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 OpenSearch 索引和 API 的访问权限。

> **Note**
>
> 使用 AWS 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：

```
{
  "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：

```
{
  "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：

```
{
  "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：

```
{
  "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**
>
> 将一个或多个 OpenSearch API 添加到任何基于标签的策略后，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：

```
{
  "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详细信息，以及基于资源的策略和基于身份的策略之间的差异，请参阅 IAM 用户指南。

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，它们仅仅是指定匿名委托人的基于资源的策略，且包含一个特殊的 Condition 元素。

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 OpenSearch 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。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”。

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：

```
{
  "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：

```
{
  "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，则对 OpenSearch API 的请求也必须使用签 AWS 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 AWS SDK。与创建和签署您自己的请求相比，开发工具包可大大简化流程，从而为您节省大量时间。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"
    }
  }
  ```

如果您使用了其中一个开发工具包（如 Boto 3），该开发工具包将自动处理请求签名：

```
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 AWS 软件开发工具包”。

- 要调用 OpenSearch API，您必须签署自己的请求。这 OpenSearch 些 API 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**
>
> 对于用签名版本 4 签署的 HTTP POST 请求，服务会忽略在 URL 中传入的参数。

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 用户指南中的了解 IAM 的工作方式提供了策略评估逻辑的简明摘要：

- 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	Summary
`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 AWS 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： • AWS 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"]}`

JSON 策略元素	Summary
	• 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，请仔细检查以下策略，以确认它们未授予广泛访问权限：
>
> - 附加到关联 AWS 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 AWS 资源的策略（例如 AWS Key Management Service KMS 密钥）

JSON 策略元素	Summary

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：
	
	`{`
	`  "Version": "2012-10-17",`
	`  "Statement": [`
	`  {`
	`  "Effect": "Allow",`
	`  "Action": [`
	`  "es:ESHttpGet",`
	`  "es:DescribeDomain"`
	`  ],`
	`  "Resource": "*"`
	`  }`
	`  ]`
	`}`

JSON 策略元素	Summary
	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AWS 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 OpenSearch 和 API。

JSON 策略元素	Summary

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	Summary
	从本质上讲，如果将子资源 OpenSearch 表示为 URI，则可以使用访问策略来控制对它的访问。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：

```
{
  "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 使用 URL 来控制对网域子资源的访问权限，因此实际上可以使用批量 API 对域名子资源进行`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，则对批量、mget 和 msearch API 的所有在请求正文中指定索引名称的调用都将停止工作。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。若要进行部分补救，您可以将 `rest.action.multi.allow_explicit_index` 保留为 true 或拒绝特定用户访问这些 API 中的一个或多个。

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：

```
{
  "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* 通配符可以匿名访问您的域名。不建议您使用通配符此外，请仔细检查以下策略以确认它们不授予广泛访问权限：

- 附加到关联 AWS 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 AWS 资源的策略（例如 AWS 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《IAM 用户指南》中的创建 IAM policy。

其他示例策略

尽管本章包含许多示例策略，但 AWS 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 AWS 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`

AWS Amazon OpenSearch 服务的托管政策

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 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，AWS 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 AWS 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 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 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）。AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

AmazonOpenSearchDirectQueryGlueCreateAccess

授予亚马逊 OpenSearch 服务直接查询服务访问`CreateDatabase`、`CreatePartition`、`CreateTable`和`BatchCreatePartition` AWS 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 AWS 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 AWS 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。 AWS 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。 AWS 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 AWS 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 AWS 托管策略更新的详细信息。

更改	描述	日期
新增了 OpenSearchIngestionSelfManagedVpcePolicy	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
已添加 AmazonOpenSearchDirectQueryGlueCreateAccess	授予亚马逊 OpenSearch 服务直接查询服务访问CreateDatabase CreatePartition 、CreateTable 、和BatchCreatePartition AWS Glue API。	2024 年 5 月 6 日
更新了AmazonOpenSearchServiceRolePolicy 和 AmazonElasticsearchServiceRolePolicy	添加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
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

更改	描述	日期
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。 AWS 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。 AWS 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

更改	描述	日期
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 AWS 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

更改	描述	日期
更新了 AmazonOpenSearchServiceRolePolicy	增加了对 security-group 资源类型的支持。该策略提供了最低 Amazon EC2 和 Elastic Load Balancing 权限，以便服务相关角色启用 VPC 访问。	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 OpenSearch 完全访问权限。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
• 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* 、和es:Get*）的只读访问权限es:List*，并且不提供对这 OpenSearch 些 API 的 HTTP 方法的访问权限。	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。两个策略提供了最低 Amazon EC2 和 Elastic Load Balancing 权限，以便服务相关角色启用 VPC 访问。	2021 年 9 月 7 日

更改	描述	日期
已开启跟踪更改	Amazon OpenSearch 服务现在可以跟踪 AWS托管策略的更改。	2021 年 9 月 7 日

跨服务混淆代理问题防范

混淆代理问题是一个安全性问题，即不具有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 AWS 中，跨服务模拟可能会导致混淆代理问题。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。为了防止这种情况，AWS 提供可帮助您保护所有服务的服务委托人数据的工具，这些服务委托人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 全局条件上下文键，以限制 Amazon OpenSearch Service 为其它服务提供的资源访问权限。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。

aws:SourceArn 的值必须为 OpenSearch Service 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*:*。

以下示例演示如何使用 OpenSearch Service 中的 aws:SourceArn 和 aws:SourceAccount 全局条件上下文键来防范混淆代理问题。

```
{
   "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

- 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

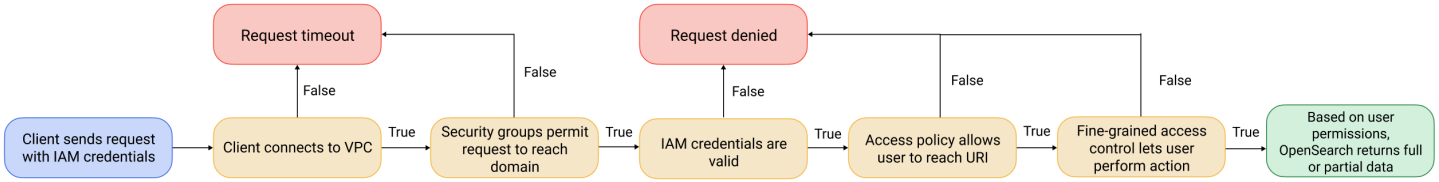
第二个安全层是域访问策略。在请求到达域终端节点后，基于资源的访问策略允许或拒绝请求访问给定 URI。访问策略在域名 “边缘” 的请求到达 OpenSearch自身之前接受或拒绝。

精细访问控制

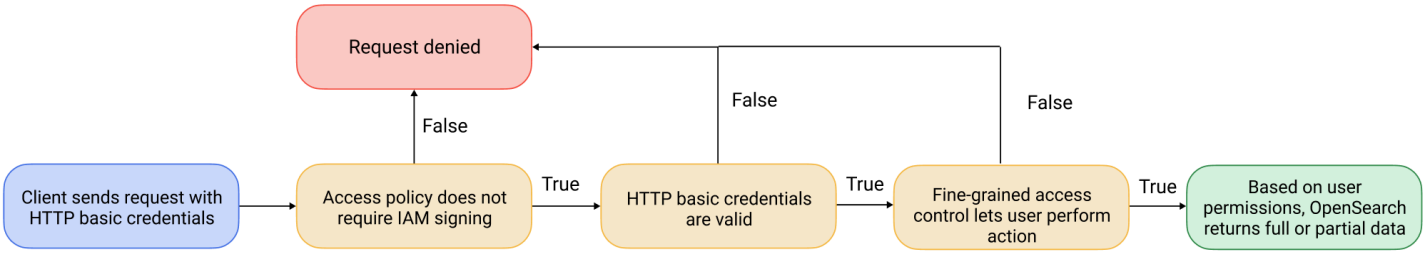
第三个也是最后一个安全层是精细访问控制。在基于资源的访问策略允许请求到达域终端节点后，精细访问控制对用户凭证进行评估，并对用户进行身份验证或拒绝请求。如果精细访问控制对用户进行身份验证，它将获取映射到该用户的所有角色，并使用完整的权限集来确定如何处理请求。

> **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 AWS 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，请考虑以下概念：

- 角色-使用细粒度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，将其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的管理。

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。如果您创建另一个内部用户并且仅将其映射到该`all_access`角色，则该用户无权访问 “安全” 选项卡。您可以通过将主用户显式映射到`all_access`和`security_manager`角色来创建其他主用户。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 委托人，也可以在内部用户数据库中创建主用户。在决定使用哪个时，请考虑以下几点：

- IAM 委托人 — 如果您为主用户选择 IAM 委托人，则必须使用签 AWS 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 用户或角色纯粹用于身份验证。针对该用户或角色的策略与主用户的授权无关。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数据库（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 证书）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 AWS 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、 AWS CLI或配置 API 启用精细访问控制。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 AWS 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。

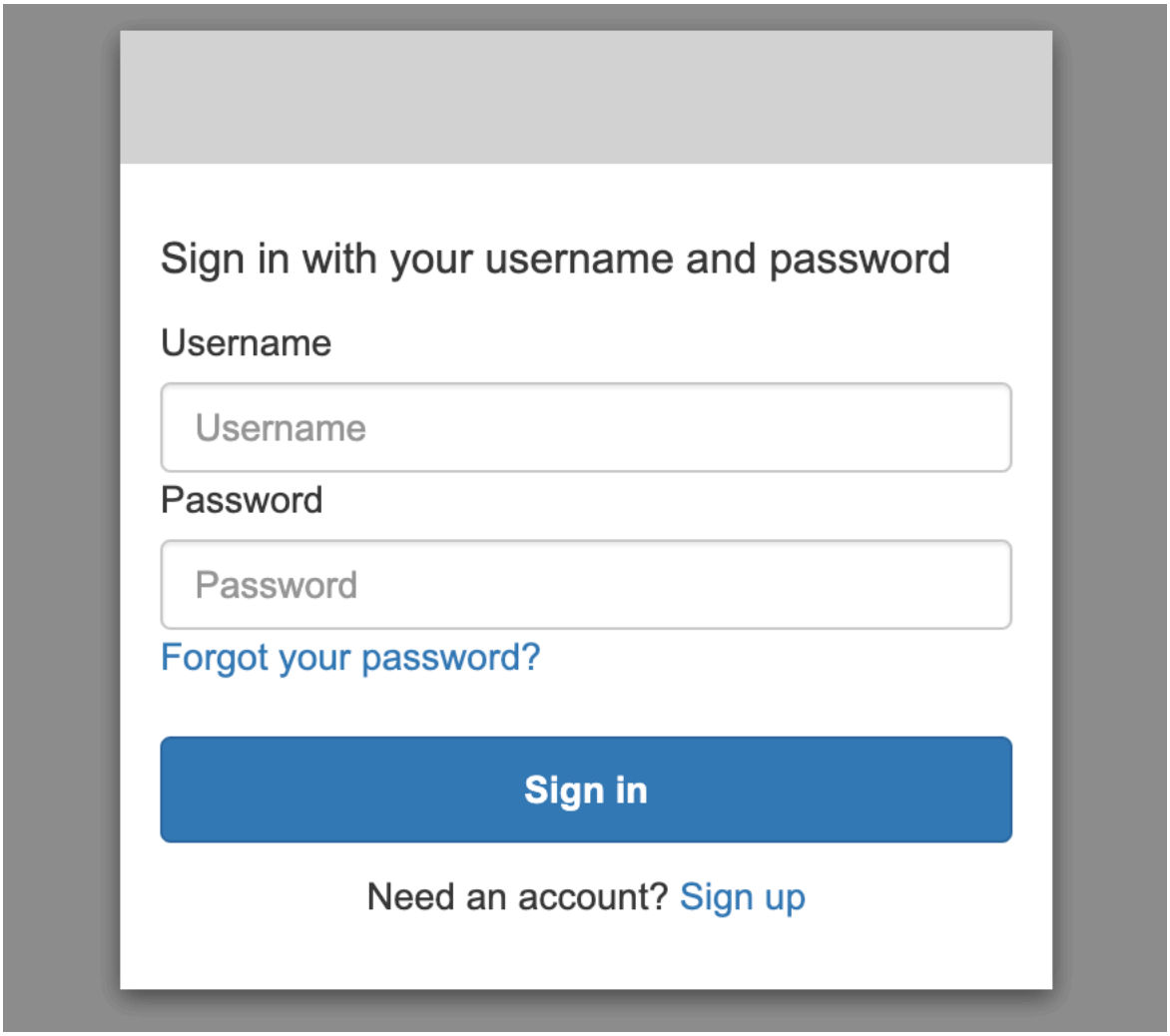
身份验证方法	启用精细访问控制之前	启用精细访问控制之后	管理员任务
			3. 禁用迁移期。来自允许的 IP 地址或 CIDR 块的发送请求且无基本身份验证或 IAM 凭证的用户将失去对该域的访问权限。
开放访问策略	互联网上的所有用户均可访问该域。	在 30 天的迁移期间，互联网上的所有用户均可继续访问该域。	1. 在域上创建角色映射。 2. 根据角色映射配置，更新客户端，以提供基本身份验证凭证或 IAM 凭证。 3. 禁用迁移期。发送请求且无基本身份验证或 IAM 凭证的用户将失去对域的访问权限。

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。

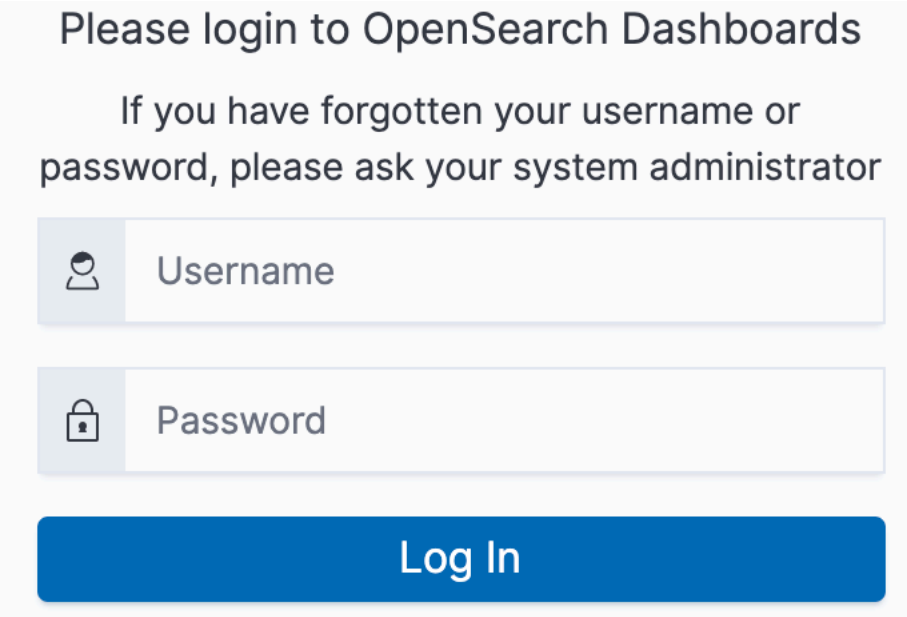
- 如果要使用 IAM 进行用户管理，请使用 the section called “OpenSearch 控制面板的 Amazon Cognito 认证” 访问控制面板。否则，控制面板将显示一个不起作用的登录页面。请参阅 the section called “限制”。

  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。

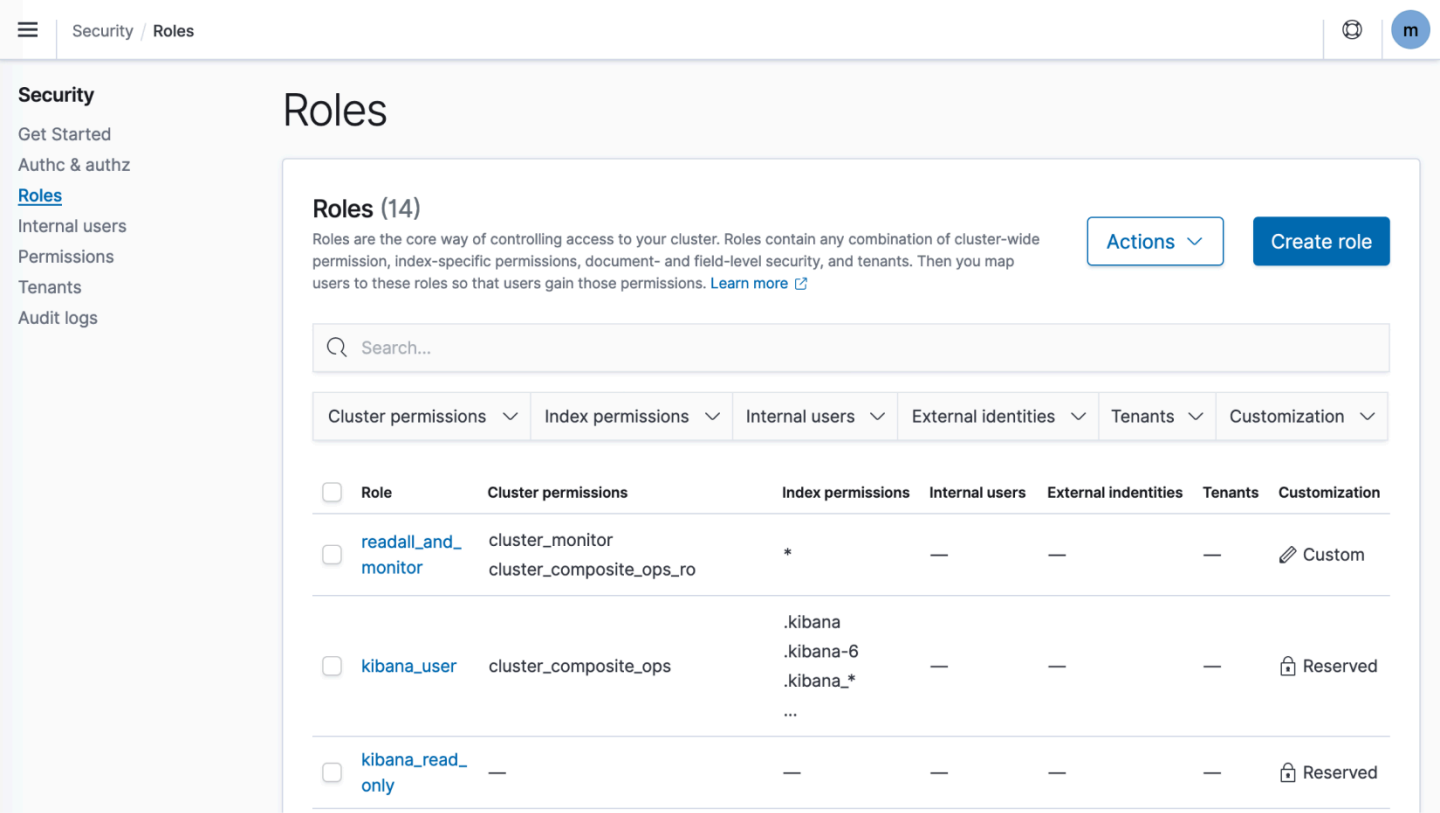
  有关此配置的更多信息，请参阅the section called “教程：使用基本身份验证的内部用户数据库”。



- 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 提供了多个不可用的角色 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。

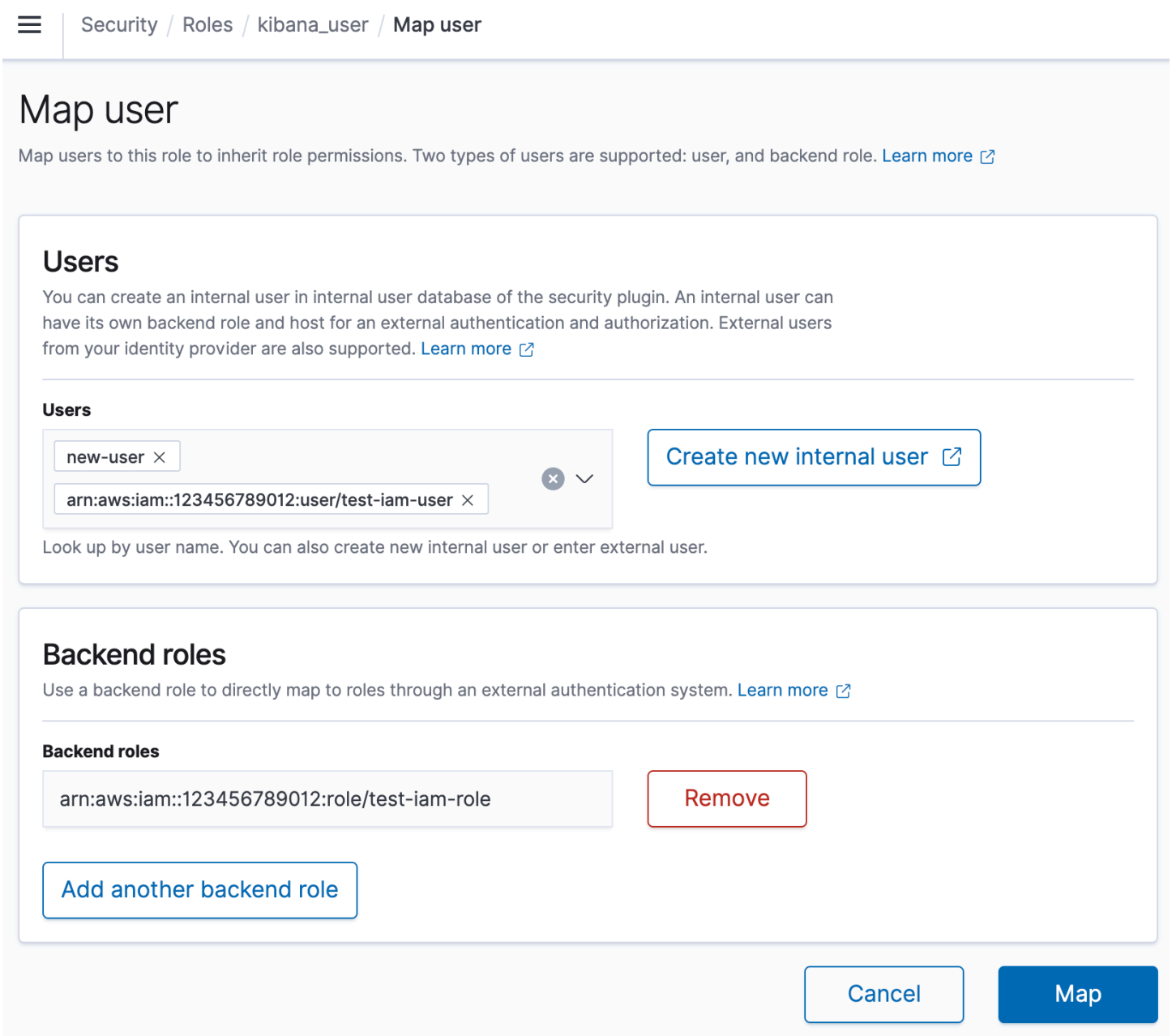
如果您为主用户选择了 IAM，请忽略控制面板的此部分。改为创建 IAM 角色。有关更多信息，请参阅 IAM 用户指南。

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。

- 请在 Users（用户）部分指定用户、用户 ARN 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`*。
- 在 Backend roles (后端角色) 部分中指定后端角色和 IAM 角色 ARN。



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 API，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	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": "es:ESHttp*",`

描述	主用户	域访问策略
		"Resource": " *domain-arn* /*" }] }

使用 IAM 凭证或基本身份验证调用 OpenSearch API。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	{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": " *domain-arn* /*"
		}
		]
		}

使用 IAM 凭证调用 OpenSearch API，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	{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": " *domain-arn* /*"
		}
		]
		}

描述	主用户	域访问策略

使用 IAM 凭证调用 OpenSearch API，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	{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": " *domain-arn* /*"
		},
		{
		"Effect": "Deny",
		"Principal": {
		"AWS": "*"
		},
		"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": {
    "AWS": [
      "*"
    ]
  },
  "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源，则需要自己设置细粒度的访问控制角色。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、 AWS CLI或配置 API 重新配置它。

修改主用户（控制台）

1. 导航到亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home/。
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。

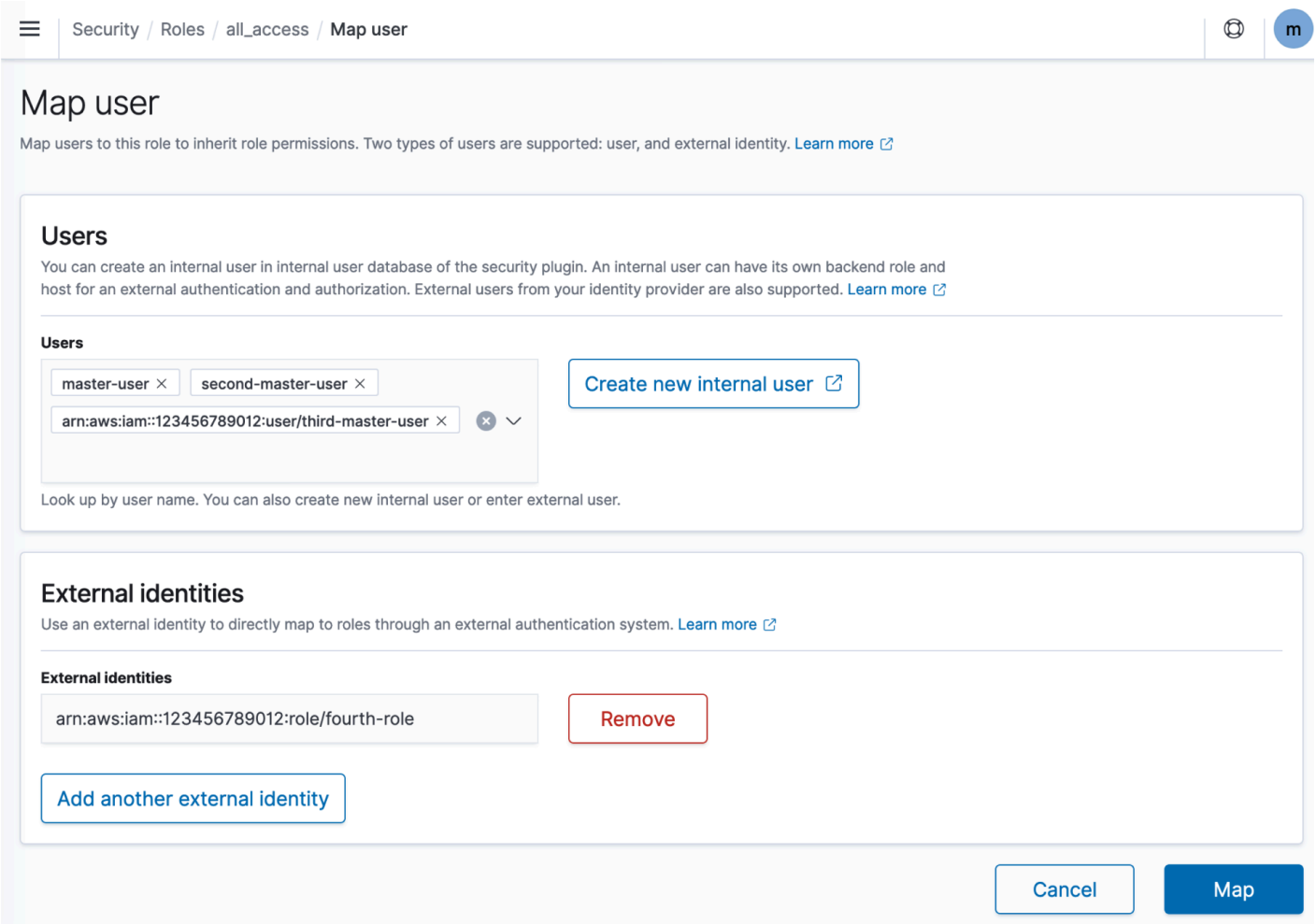
- 从内部用户数据库切换到 IAM 主用户不从内部用户数据库中删除任何用户。相反，它只是禁用 HTTP 基本身份验证。手动从内部用户数据库中删除用户，或保留这些用户，以防您需要重新启用 HTTP 基本身份验证。

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 AWS 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，Firehose 交付流通常使用名为的 IAM 角色。`firehose_delivery_role`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 AWS IoT 规则或 AWS 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

细粒度的访问控制 REST API 略有不同，具体取决于你的 OpenSearch /Elasticsearch 版本。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 AWS 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：

```
{
  "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

通过 https://console.aws.amazon.com/aos/home/ 导航到亚马逊 OpenSearch 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：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "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
- N ode-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 AWS 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

通过 https://console.aws.amazon.com/aos/home/ 导航到亚马逊 OpenSearch 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：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "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
- N ode-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 AWS 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AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 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。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了部署以安全性和合规性为重点 AWS 的基准环境的步骤。
- 在 A@@ mazon Web Services 上构建 HIPAA 安全与合规性 — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

> **Note**
>
> 并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅符合 HIPAA 要求的服务参考。

- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS 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。
- AWS Security Hub— 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控件评估您的 AWS 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件的列表，请参阅 Security Hub 控件参考。
- Amazon GuardDuty — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 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。
- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

Amazon OpenSearch Service 中的恢复能力

AWS全球基础设施围绕AWS 区域和可用区构建。AWS 区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、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故障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设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅 AWS 全球基础设施。

除了 AWS 全球基础设施之外，OpenSearch Service 还提供了多种功能，以帮助支持您的数据弹性和备份需求：

- 多可用区域分片和副本分片
- 自动和手动快照

适用于 Amazon 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 (JWT) 进行身份验证和授权。JWT 是基于 JSON 的访问令牌，用于授予单点登录 (SSO) 访问权限。您可以使用 Service 中的 JWT 创建单点登录令牌来验证对 OpenSearch 服务域的请求。 OpenSearch 要使用 JWT，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、 AWS Command Line Interface (AWS CLI) 或软件开发工具包配置 J AWS SON Web 令牌。

注意事项

在将 JWT 与 Amazon OpenSearch 服务配合使用之前，您必须考虑以下几点：

- 由于 PEM 格式的 RSA 公钥很大，我们建议使用 AWS 控制台配置 JWT 身份验证和授权。
- 在为 JWT 指定主题和角色字段时，必须提供有效的用户和角色，否则，请求将被拒绝。

修改域访问策略

在将您的域配置为使用 JWT 身份验证和授权之前，必须更新您的域访问策略以允许 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推荐域访问策略是：

```
{
  "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

您可以在域名创建过程中或通过更新现有域来启用 JWT 身份验证和授权。根据您选择的选项，设置步骤略有不同。

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. 配置要用于您的域的公钥。为此，您可以上传包含公钥的 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，说明问题。

3. （可选）在 “其他设置” 下，您可以配置以下可选字段

   - 主题密钥 — 您可以将此字段留空以使用您的 JWT 的默认`sub`密钥。
   - 角色密钥 — 您可以将此字段留空，以便使用您的 JWT 的默认`roles`密钥。

   进行更改后，保存您的域名。

使用 JWT 发送测试请求

创建具有指定主题和角色对的新 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 指定了自定义主题密钥或角色密钥，则必须为 JWT 使用正确的声明名称。

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 ()AWS CLI

如果域存在，以下 AWS 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 进行配置）

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

为了为您的 OpenSearch 域配置 JWT，您需要提供隐私增强邮件 (PEM) 格式的公钥。当使用 JWT 时，亚马逊 OpenSearch 服务目前支持两种非对称加密算法：RSA 和 ECDSA。

要使用常用 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，该`publickey.pem`文件包含用于 Amazon OpenSearch 服务的公钥，而`privatekey.pem`包含用于签署发送到该服务的 JWT 的私钥。此外，如果您需要将私钥转换为常用`pkcs8`格式来生成 JWT，则可以选择将其转换为常用格式。

如果您使用上传按钮将 PEM 文件直接添加到控制台，则该文件必须具有`.pem`扩展名，`.key`目前不支持其他文件扩展名，例如`.crt`、`.cert`或。

Amazon OpenSearch 服务中的基础设施安全

作为一项托管服务，Amazon OpenSearch 服务受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用通过网络访问 OpenSearch 服务。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 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

您可以使用 AWS 已发布的 API 调用通过网络访问 OpenSearch 服务配置 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 AWS CLI 命令参考。

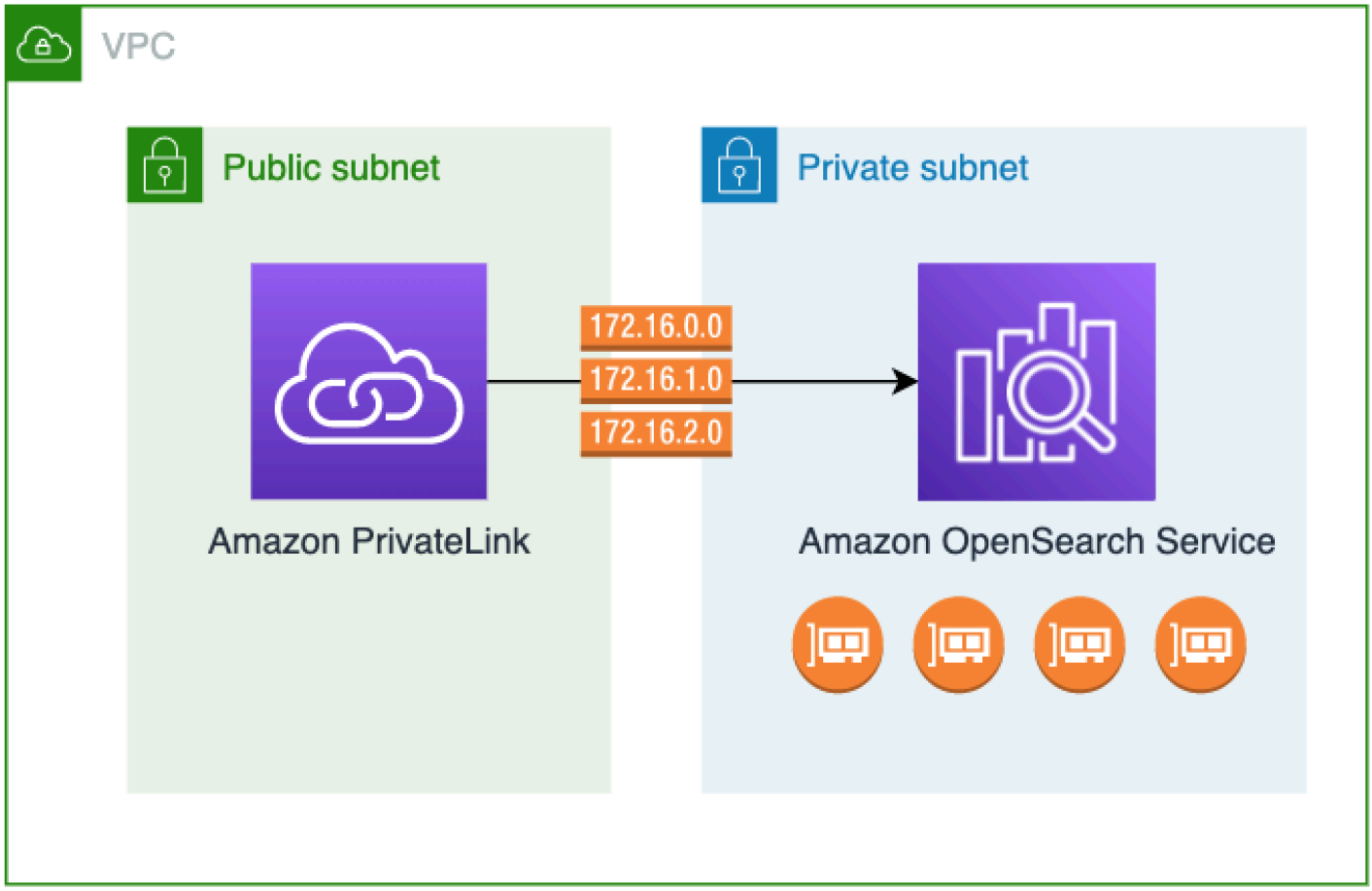
根据您的域配置，您可能还需要签署 OpenSearch API 请求。有关更多信息，请参阅the section called “提出和签署 OpenSearch 服务请求”。

OpenSearch 服务支持公共访问域（可以接收来自任何联网设备的请求）和 VP C 访问域（与公共互联网隔离）。

使用 OpenSearch OpenSearch 服务管理的 VPC 终端节点访问亚马逊服务 ()AWS PrivateLink

您可以通过设置 OpenSearch 服务 OpenSearch 托管 VPC 终端节点（由提供支持 AWS PrivateLink）来访问 Amazon 服务域。这些终端节点在您的 VPC 和 Amazon OpenSearch 服务之间创建私有连接。无需使用互联网网关、NAT 设备、VPN 连接或 AWS Direct Connect 连接，即可像访问您的 VPC OpenSearch 服务域一样访问服务 VPC 域。您的 VPC 中的实例不需要公有 IP 地址即可访问 OpenSearch 服务。

您可以将 OpenSearch 服务域配置为公开在同一 VPC、不同 VPC 或不同的 VPC 中的公有或私有子网上运行的其他终端节点。 AWS 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。下图说明了同一 VPC 中的 OpenSearch 服务托管 VPC 终端节点：



您可以通过创建由提供支持的 OpenSearch 服务管理接口 VPC 终端节点来 AWS 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。这些是服务管理的网络接口，是发往服务的流量的入口点。 OpenSearch 标准AWS 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。 AWS PrivateLink

您可以为运行所有版本 OpenSearch 和旧版 Elasticsearch 的域创建 VPC 终端节点。有关更多信息，请参阅《AWS PrivateLink 指南》中的通过 AWS PrivateLink访问 AWS 服务。

OpenSearch 服务的注意事项和限制

在为 OpenSearch 服务设置接口 VPC 终端节点之前，请查看AWS PrivateLink 指南中的注意事项。

使用 OpenSearch 服务管理的 VPC 终端节点时，请考虑以下几点：

- 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 AWS 区域内的域。
- HTTPS 是 VPC 端点唯一支持的协议。不允许使用 HTTP。
- OpenSearch 服务支持通过接口 VPC 终端节点调用所有支持的 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。
- 您目前无法使用 AWS CloudFormation 创建接口 VPC 终端节点。
- 您只能通过 OpenSearch 服务控制台或使用OpenSearch 服务 API 创建接口 VPC 终端节点。您无法使用 Amazon VPC 控制台为 OpenSearch 服务创建接口 VPC 终端节点。
- OpenSearch 无法通过互联网访问服务管理的 VPC 终端节点。在路由表和安全组允许的情况下，OpenSearch 服务管理的 VPC 终端节点只能在配置了终端节点的 VPC 或与配置终端节点的 VPC 对等的任何 VPC 内进行访问。
- OpenSearch 服务不支持 VPC 终端节点策略。您可以将安全组与终端节点网络接口关联，以控制通过接口 VPC 终端节点流向 OpenSearch 服务的流量。
- 您的服务相关角色必须与您用于创建 VPC 终端节点的 AWS 账户相同。
- 要创建、更新和删除 OpenSearch 服务 VPC 终端节点，除了亚马逊 OpenSearch 服务权限外，您还必须拥有以下 Amazon EC2 权限：
  - `ec2:CreateVpcEndpoint`
  - `ec2:DescribeVpcEndpoints`
  - `ec2:ModifyVpcEndpoint`
  - `ec2:DeleteVpcEndpoints`
  - `ec2:CreateTags`
  - `ec2:DescribeTags`
  - `ec2:DescribeSubnets`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeVpcs`

> **Note**
>
> 目前，您不能将 VPC 终端节点的创建限制为 OpenSearch 服务。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

如果您要访问您的域的 VPC 位于另一个域中 AWS 账户，则需要先通过所有者的账户对其进行授权，然后才能创建接口 VPC 终端节点。

允许另一个 VPC 中的一个 VPC AWS 账户 访问您的域

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home/。
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 endpoints（VPC 端点）选项卡，其中显示有权访问您的域的账户和相应的 VPC。
4. 选择 Authorize principal（为主体授权）。
5. 输入将访问您的域名的账户的 AWS 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 AWS Command Line Interface (AWS 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 终端节点

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home/。
2. 在左侧导航窗格中，选择 VPC endpoints（VPC 端点）。
3. 选择创建端点。
4. 选择是连接当前域 AWS 账户 还是其他域 AWS 账户。
5. 选择您使用此端点连接的域。如果域名在当前域名中 AWS 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，请选择您将从中访问 OpenSearch 服务的 VPC。
7. 对于子网，请选择一个或多个要从中访问服务的 OpenSearch 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 终端节点与 OpenSearch 服务通信的资源与终端节点网络接口通信。

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 API 使用 OpenSearch 服务管理的 VPC 终端节点

使用以下 API 操作创建和管理 OpenSearch 服务管理的 VPC 终端节点。

- 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。 AWS 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 API 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。我们无法为您的确切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 服务提供两个单点登录 URL，分别是 SP 启动的和 IDP 启动的，但您只需要与所需的控制面板登录流程相匹配的单点登录 URL。 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 AWS 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 ???。

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：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "domain-arn/*"
    }
  ]
}
```

要使策略更具限制性，可以在策略中添加 IP 地址条件。此条件限制只能访问指定的 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：

```
{
  "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您的域上启用精细的访问控制，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稳健的密码，那么从安全角度来看，在使用精细访问控制的同时保持策略打开可能是可以接受的。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 URL。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。

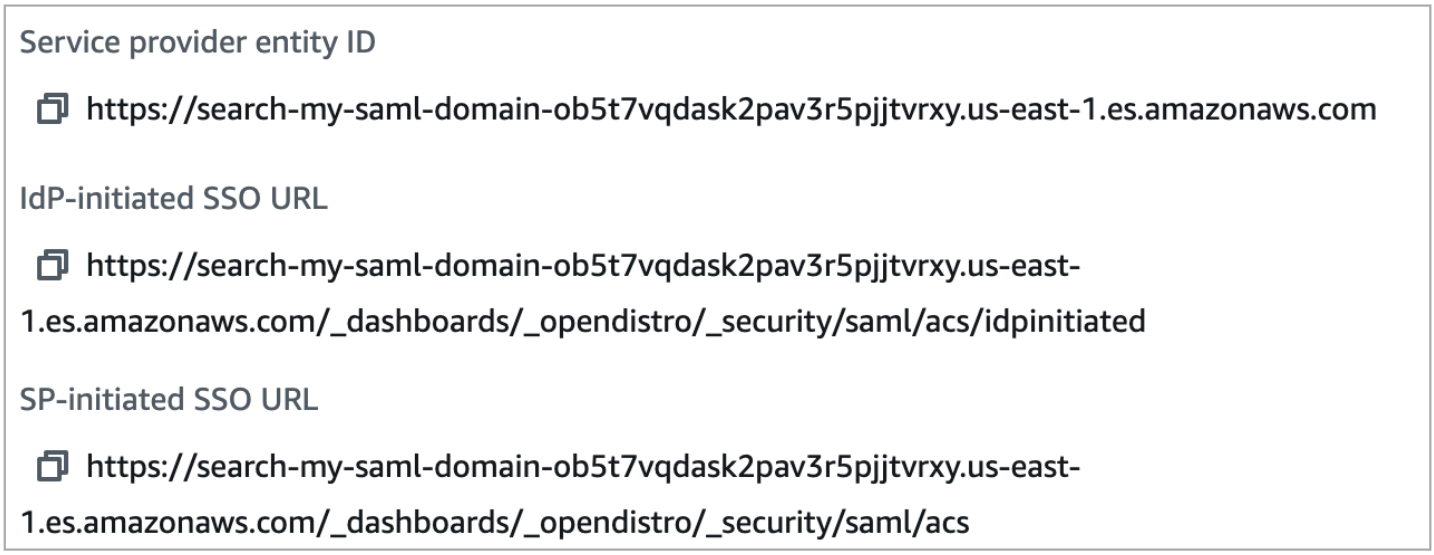
如果您使用的是自定义端点，可以推断出 URL。例如，如果您的自定义端点是 `www.custom-endpoint.com`，则服务提供程序实体 ID 将是 `www.custom-endpoint.com`，IdP 发起的 SSO URL 将是 `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs/idpinitiated`，SP 发起的 SSO URL 将是 `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs`。在创建域之前，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。

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 “步骤 6：更新您的 IdP URL”。

如果您正在编辑现有域

如果您正在现有域上启用 SAML 身份验证，请复制服务提供程序实体 ID 和其中一个 SSO URL。有关使用哪个 URL 的指南，请参阅 the section called “SAML 配置概述”。



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 主后端角色-您指定的用户和/或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。

步骤 6：更新您的 IdP URL

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 URL 才能生成 XML 元数据文件。域状态更改为 `Active` 后，您可以获取正确的 URL 并修改 IdP。

要检索 URL，请选择域，然后选择 Actions（操作）、Edit security configuration（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 网址。复制这些值用于配置身份提供程序，替换您在步骤 2 中提供的临时 URL。

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登录。要继续执行步骤 7 中的示例，请以 jdoe 或 admins 组中 成员登录)。

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 Allow this app to request other SSO URLs（允许此应用程序请求其他 SSO URL）。
4. 在 Requestable SSO URLs（可请求的 SSO URL）下，添加一个或多个 SP 发起的 SSO URL。例如，`https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs`。

配置 SAML 身份验证 (AWS CLI)

以下 AWS 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。有关使用的详细信息 AWS CLI，请参阅《AWS 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
您的请求：'*/some/path* '不允许。	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。
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 AWS 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。 您的系统管理员可以使用像 SAML -Tracer 这样的工具验证您的 SAML 断言的内容，然后使用以下请求检查您的角色映射：

错误	详细信息
	`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。
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。AWS 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。

错误	详细信息
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。
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 URL 匹配的目标字段中输入。 OpenSearch
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
```

禁用控制面板的 SAML 身份验证不删除 SAML 主用户名和/或 SAML 主后端角色的映射。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"
  ]
}
```

为 OpenSearch 控制面板配置 Amazon Cognito 认证

您可以使用 Amazon Cognito 来验证和保护您的 Amazon OpenSearch Service 默认安装的 OpenSearch 控制面板。Amazon Cognito 身份验证是可选的，仅适用于使用 OPEN 搜索或 Elasticsearch 5.1 或更高版本的域。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发生在 Amazon Cognito，但本节提供了配置 Amazon Cognito 资源以使用 OpenSearch Service 域的指南和要求。标准定价适用于所有 Amazon Cognito 资源。

> **Tip**
>
> 首次将域配置为针对 OpenSearch 控制面板使用 Amazon Cognito 身份验证时，我们建议使用控制台。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

- 常见配置问题
- 禁用 OpenSearch 控制面板的 Amazon Cognito 身份认证
- 删除使用 OpenSearch 控制面板的 Amazon Cognito 身份验证的阈。

先决条件

在配置用于 OpenSearch 控制面板的 Amazon Cognito 身份验证之前，您必须满足几个前提条件。OpenSearch Service 控制台有助于简化这些资源的创建，但了解每个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 AWS 区域 中。您可以使用相同的用户池、身份池和 IAM 角色向多个 OpenSearch Service 域添加用于控制面板的 Amazon Cognito 身份验证。要了解更多信息，请参阅 the section called “配额”。

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 开发人员指南中的创建用户池。

在创建要用于 OpenSearch Service 的用户池时，请考虑以下事项：

- 您的 Amazon Cognito 用户池必须拥有域名。OpenSearch Service 使用此域名将用户重定向到登录页面，以便访问控制面板。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 ID 采用以下格式：*region_ID*。如果您计划使用 AWS CLI 或 AWS 软件开发工具包来配置 OpenSearch Service，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 开发人员指南中的身份池。在创建要用于 OpenSearch Service 的身份池时，请考虑以下事项：

- 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 Service 域之后，Amazon Cognito 禁用此设置。
- 您无需向身份池添加外部身份提供商。在配置 OpenSearch Service 以使用 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 ID 采用以下格式：*region*:*ID*-*ID*-*ID*-*ID*-*ID*。如果您计划使用 AWS CLI 或 AWS 软件开发工具包来配置 OpenSearch Service，请记下 ID。

关于 CognitoAccessForAmazonOpenSearch 角色

OpenSearch Service 需要权限来配置 Amazon Cognito 用户和身份池，并使用它们进行身份验证。为此，您可以使用 `AmazonOpenSearchServiceCognitoAccess`，它是一项 AWS 托管策略。`AmazonESCognitoAccess` 是旧式策略，在该服务重命名为 Amazon OpenSearch Service 时，该旧式策略已被 `AmazonOpenSearchServiceCognitoAccess` 取代。两个策略都提供了最低 Amazon Cognito 权限，以便启用 Cognito 身份验证。有关策略 JSON，请参阅 IAM 控制台。

如果您使用控制台来创建或配置 OpenSearch Service 域，它会为您创建一个 IAM 角色并将 `AmazonOpenSearchServiceCognitoAccess` 策略（如果是 Elasticsearch 域，则为 `AmazonESCognitoAccess` 策略）附加到角色。此角色的默认名称为 `CognitoAccessForAmazonOpenSearch`。

角色权限策略 `AmazonOpenSearchServiceCognitoAccess` 和 `AmazonESCognitoAccess` 均允许 OpenSearch Service 对所有身份和用户池完成以下操作：

- 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 AWS CLI 或 AWS 开发工具包之一，则必须创建您自己的角色、附加策略并在配置您的 OpenSearch Service 域时为此角色指定 ARN。角色必须拥有以下信任关系：

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

有关说明，请查看 IAM 用户指南中的创建角色以向 AWS 服务委托权限和附加和分离 IAM 策略。

将域配置为使用 Amazon Cognito 身份验证

满足先决条件后，您可以配置 OpenSearch 服务域，以将 Amazon Cognito 用于控制面板。

> **Note**
>
> Amazon Cognito 并非在所有 AWS 区域 可用。有关受支持的区域的列表，请参阅 AWS 区域和终端节点。您不需要对 Amazon Cognito 使用与 OpenSearch 服务相同的区域。

配置 Amazon Cognito 身份验证（控制台）

由于它会为您创建 CognitoAccessForAmazonOpenSearch 角色，因此，控制台可提供最简单的配置体验。除了标准的 OpenSearch Service 权限之外，您还需要以下一组权限来使用控制台创建一个域，该域对 OpenSearch 控制面板使用 Amazon Cognito 身份验证。

```
{
  "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：

```
{
  "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 https://console.aws.amazon.com/aos/home/ 打开 Amazon OpenSearch Service 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. 选择启用 Amazon Cognito 身份验证。
5. 对于 Region（区域），请选择包含您的 Amazon Cognito 用户群体和身份池的 AWS 区域。
6. 对于 Cognito user pool（Cognito 用户池），选择一个用户池或创建一个。有关操作指南，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。有关操作指南，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。要了解更多信息，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。要了解有关此角色用途的更多信息，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 (AWS CLI)

使用 `--cognito-options` 参数配置您的 OpenSearch Service 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 Amazon Cognito 身份验证（AWS 软件开发工具包）

AWS SDK（除 Android 和 iOS SDK 之外）支持在 Amazon OpenSearch Service API 参考中定义的所有操作，包括用于 `CreateDomain` 和 `UpdateDomainConfig` 操作的 `CognitoOptions` 参数。有关安装和使用 AWS 开发工具包的更多信息，请参阅 AWS 软件开发工具包。

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 the section called “关于身份池” 中的指南配置的经过身份验证的 IAM 角色没有访问 OpenSearch 控制面板所需的必要权限。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。

您可以在基于身份的策略中包含这些权限，但除非您希望经过身份验证的用户有权访问所有 OpenSearch Service 域，否则附加到单个域的基于资源的策略是更好的方法。

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。

```
{
   "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 Service 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

配置域以使用用于控制面板的 Amazon Cognito 身份验证时，OpenSearch Service 会向用户池中添加一个应用程序客户端，并将该用户池作为身份验证提供商添加到身份池。

> **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 开发人员指南中的从用户池使用联合身份验证和为用户池应用程序指定身份提供商设置。

（可选）配置精细访问

您可能已经注意到，默认的身份池设置会分配登录相同 IAM 角色 (`Cognito_`*`identitypool`*`Auth_Role`)，的每个用户，这意味着每个用户都可以访问相同的 AWS 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 开发人员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。有关创建组的信息，请参阅 Amazon Cognito 开发人员指南中的用户组。

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 开发人员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 开发人员指南中的为用户池指定 UI 自定义设置。

（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 Amazon Cognito 开发人员指南中的管理安全性。

测试

对配置感到满意后，请验证用户体验是否符合的预期。

要访问 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. 加载 OpenSearch 控制面板后，至少要配置一个索引模式。控制面板使用这些模式来标识要分析的索引。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。

配额

Amazon Cognito 对于它的许多资源都有软限制。如果要为许多 OpenSearch Service 域启用控制面板身份验证，请查看 Amazon Cognito 中的限制，并根据需要请求提高限制。

每个 OpenSearch Service 域都会向用户池中添加一个应用程序客户端，该客户端会向身份池添加一个身份验证提供商。如果您为 10 个以上域启用 OpenSearch 控制面板身份验证，则可能会遇到“maximum Amazon Cognito user pool providers per identity pool”(每个身份池的最大 Amazon Cognito 用户池提供商) 限制。如果超出限制，则您尝试配置以使用用于控制面板的 Amazon Cognito 身份验证的任何 OpenSearch Service 域都会卡在配置状态 Processing（正在处理）。

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

问题	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 CognitoAccessForAmazonOpenSearch 角色的 `iam:PassRole` 权限。将以下策略附加到您的账户中： `{ "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。

问题	解决方案
`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`	`CognitoAccessForAmazonOpenSearch` 角色的信任关系中未指定 OpenSearch Service。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。检查该角色是否附加了 AWS 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。
`An error occurred (ValidationException) when calling the CreateDomain operation: User pool does not exist`	OpenSearch Service 找不到用户池。确认您已创建一个用户池并具有正确的 ID。要查找该 ID，您可以使用 Amazon Cognito 控制台或以下 AWS CLI 命令： `aws cognito-idp list-user-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateDomain operation: IdentityPool not found`	OpenSearch Service 找不到身份池。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 AWS CLI 命令： `aws cognito-identity list-identity-pools --max-results 60 --region region`

问题	解决方案
`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 AWS CLI 命令配置一个： `aws cognito-idp create-user-pool-domain --domain name --user-pool-id id`

访问 OpenSearch 控制面板

问题	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 the section called “配置身份提供商” 中指定的方式为 OpenSearch Service 应用程序客户端启用了身份提供商。
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。
OpenSearch 控制面板根本不加载或无法正常工作。	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。
当我从某个选项卡注销 OpenSearch 控制面板时，其余选项卡会显示一条消息，指出刷新令牌已被撤销。	如果您在使用 Amazon Cognito 身份验证时注销 OpenSearch 控制面板会话，OpenSearch Service 将运行 AdminUserGlobalSignOut 操作，该操作将使您注销所有处于活动状态的 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： `{`

问题	解决方案

	"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"
	}
	}
	}]
	}

`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 OpenSearch 控制面板的 Amazon Cognito 身份认证

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. 在 https://console.aws.amazon.com/aos/home/ 打开 Amazon OpenSearch Service 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。

删除使用 OpenSearch 控制面板的 Amazon Cognito 身份验证的阈。

要阻止针对控制面板使用 Amazon Cognito 身份验证的域卡在 Processing（正在处理）配置状态，请在删除这些域的关联Amazon Cognito 用户群体和身份池之前，先删除 OpenSearch Service 域。

在 Amazon OpenSearch 服务中使用服务相关角色

亚马逊 OpenSearch 服务使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特的 IAM 角色，直接链接到 OpenSearch 服务。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 AWS 服务所需的所有权限。

与服务相关的角色可以更轻松地设置 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，除非另有定义，否则只有 S OpenSearch ervice 可以担任其角色。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它 IAM 实体。有关服务相关角色和权限策略的更新，请参阅 Amazon OpenSearch 服务的文档历史记录。

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AWS 服务，并查找服务相关角色列中显示为是的服务。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
- 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

亚马逊 OpenSearch 服务使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特的 IAM 角色，直接链接到 OpenSearch 服务。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 AWS 服务所需的所有权限。

OpenSearch 服务使用名为的服务相关角色 AWSServiceRoleForAmazonOpenSearchService，该角色提供该角色启用 VPC 访问域所需的最低 Amazon EC2 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旧版 Elasticsearch 角色，则 OpenSearch 服务会在您首次创建域名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

名为的角色权限策略AmazonOpenSearchServiceRolePolicy允许 S OpenSearch ervice 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：当请求包含标签 OpenSearchManaged=true 时，针对所有 VPC、安全组、子网和路由表以及所有 VPC 端点执行 ec2:CreateVpcEndpoint
- 操作：当请求包含标签 OpenSearchManaged=true 时，针对所有 VPC、安全组、子网和路由表以及所有 VPC 端点执行 ec2:ModifyVpcEndpoint

- 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。当您使用创建支持 VPC 的域时 AWS Management Console，OpenSearch 服务会为您创建服务相关角色。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 服务不允许您编辑`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. 登录 AWS Management Console 并打开 IAM 控制台，网址为 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、API 或 AWS 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 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特的 IAM 角色，直接链接到 OpenSearch 服务。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 AWS 服务所需的所有权限。

OpenSearch Serverless 使用名为的服务相关角色 AWSServiceRoleForAmazonOpenSearchServerless，该角色提供向您的账户发布与无服务器相关的 CloudWatch指标所需的权限。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《AWS 托管策略参考指南》。

Serverless 的服务相关角色权限 OpenSearch

OpenSearch Serverless 使用名为的服务关联角色 AWSServiceRoleForAmazonOpenSearchServerless，该角色允许 OpenSearch Serverless 代表您调用 AWS 服务。

AWSServiceRoleForAmazonOpenSearchServerless 服务相关角色信任以下服务来代入该角色：

- observability.aoss.amazonaws.com

名为的角色权限策略AmazonOpenSearchServerlessServiceRolePolicy允许 OpenSearch Serverless 对指定资源完成以下操作：

- 操作：cloudwatch:PutMetricData对所有 AWS 资源采取行动

> **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 AWS Management Console、或 AWS API 中创建 OpenSearch 无服务器集合时 AWS 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。

您还可以使用 IAM 控制台在 A mazon OpenSearch Serverless 用例中创建服务相关角色。在 AWS CLI 或 AWS 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功能或服务，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 AWS 账户所有 OpenSearch 无服务器集合。

> **Note**
>
> 如果您尝试删除资源时 OpenSearch Serverless 正在使用该角色，则删除可能会失败。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 AWS CLI、或 AWS 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 无服务器服务相关角色支持的区域

OpenSearch Serverless 支持在每个可用 S OpenSearch erverless 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 Serverless 终端节点和配额。AWS 一般参考

使用服务相关角色创建 OpenSearch 摄取管道

Amazon OpenSearch Ingestion 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特的 IAM 角色，直接关联到 OpenSearch Ingestion。服务相关角色由 OpenSearch Ingestion 预定义，包括该服务代表您调用其他 AWS 服务所需的所有权限。

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，它使用名为的服务相关角色。AWSServiceRoleForOpensearchIngestionSelfManagedVpce随附的策略为该角色提供了在您的账户和 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

名为的角色权限策略AmazonOpenSearchIngestionServiceRolePolicy允许 OpenSearch 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

名为的角色权限策略OpenSearchIngestionSelfManagedVpcePolicy允许 OpenSearch 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。当你在、或 AWS API 中创建 OpenSearch 摄取管道时 AWS Management Console， OpenSearch Ingestion 会为你创建服务相关角色。 AWS 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 会再次为您创建服务相关角色。

编辑 Ingestion 的 OpenSearch 服务相关角色

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 的 OpenSearch 服务相关角色

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**
>
> 如果您尝试删除资源时 OpenSearch Ingestion 正在使用该角色，则删除可能会失败。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. 导航至 Amazon OpenSearch 服务控制台，然后选择 Ingesti on。
2. 删除所有管道。有关说明，请参阅the section called “删除管道”。

删除 Ingestion 的 OpenSearch 服务相关角色

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 OpenSearch Service 示例代码

本章包含关于使用 Amazon OpenSearch Service 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 AWS SDK 创建域。

主题

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 OpenSearch Service 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 OpenSearch 的客户端版本。

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

在亚马逊 OpenSearch 服务中压缩 HTTP 请求

您可以使用 gzip 压缩来压缩亚马逊 OpenSearch 服务域中的 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。有些 OpenSearch 客户端内置了对 gzip 压缩的支持，而许多编程语言都有简化该过程的库。

启用 gzip 压缩

不要与类似的设置混淆，`http_compression.enabled`该 OpenSearch 设置特定于 OpenSearch 服务，可在域上启用或禁用 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。

要确认您已成功启用 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，则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 AWS SDK 与 Amazon OpenSearch Service 进行交互

这部分包含关于如何使用 AWS SDK 与 Amazon OpenSearch Service 配置 API 进行交互的示例。这些代码示例演示了如何创建、更新和删除 OpenSearch Service 域。

Java

此部分包含 AWS SDK for Java 的版本 1 和版本 2。

Version 2

此示例使用 AWS SDK for Java 的版本 2 中的 AmazonOpenSearchClientBuilder 构造函数创建 OpenSearch 域、更新其配置和对其进行删除。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 AWS SDK for Java 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。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 AWS SDK for Python (Boto) 中的 OpenSearchService 低级 Python 客户端创建域、更新其配置和对其进行删除。

```
import boto3
import botocore
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-west-2'
)

client = boto3.client('opensearch', config=my_config)

domainName = 'my-test-domain'  # The name of the domain


def createDomain(client, domainName):
    """Creates an Amazon OpenSearch Service domain with the specified options."""
    response = client.create_domain(
        DomainName=domainName,
        EngineVersion='OpenSearch_1.0',
        ClusterConfig={
            'InstanceType': 't2.small.search',
            'InstanceCount': 5,
            'DedicatedMasterEnabled': True,
            'DedicatedMasterType': 't2.small.search',
            'DedicatedMasterCount': 3
        },
        # Many instance types require EBS storage.
        EBSOptions={
            'EBSEnabled': True,
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
```

```
        AccessPolicies="{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-west-2:123456789012:domain/my-test-domain/*
\"}]}",
        NodeToNodeEncryptionOptions={
            'Enabled': True
        }
    )
    print("Creating domain...")
    print(response)


def updateDomain(client, domainName):
    """Updates the domain to use three data nodes instead of five."""
    try:
        response = client.update_domain_config(
            DomainName=domainName,
            ClusterConfig={
                'InstanceCount': 3
            }
        )
        print('Sending domain update request...')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def deleteDomain(client, domainName):
    """Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes."""
    try:
        response = client.delete_domain(
            DomainName=domainName
        )
        print('Sending domain deletion request...')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
```

```
        else:
            raise error


def waitForDomainProcessing(client, domainName):
    """Waits for the domain to finish processing changes."""
    try:
        response = client.describe_domain(
            DomainName=domainName
        )
        # Every 15 seconds, check whether the domain is processing.
        while response["DomainStatus"]["Processing"] == True:
            print('Domain still processing...')
            time.sleep(15)
            response = client.describe_domain(
                DomainName=domainName)

        # Once we exit the loop, the domain is available.
        print('Amazon OpenSearch Service has finished processing changes for your
 domain.')
        print('Domain description:')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def main():
    """Create a new domain, update its configuration, and delete it."""
    createDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    updateDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    deleteDomain(client, domainName)
```

节点

此示例使用适用于 JavaScript in Node.js 的 SDK 版本 3 OpenSearch 客户端创建域、更新其配置和对其进行删除。

```
var {
    OpenSearchClient,
    CreateDomainCommand,
    DescribeDomainCommand,
    UpdateDomainConfigCommand,
    DeleteDomainCommand
} = require("@aws-sdk/client-opensearch");
var sleep = require('sleep');

var client = new OpenSearchClient();

var domainName = 'my-test-domain'

// Create a new domain, update its configuration, and delete it.
createDomain(client, domainName)
waitForDomainProcessing(client, domainName)
updateDomain(client, domainName)
waitForDomainProcessing(client, domainName)
deleteDomain(client, domainName)

async function createDomain(client, domainName) {
    // Creates an Amazon OpenSearch Service domain with the specified options.
    var command = new CreateDomainCommand({
        DomainName: domainName,
        EngineVersion: 'OpenSearch_1.0',
        ClusterConfig: {
        'InstanceType': 't2.small.search',
        'InstanceCount': 5,
        'DedicatedMasterEnabled': 'True',
        'DedicatedMasterType': 't2.small.search',
        'DedicatedMasterCount': 3
        },
        EBSOptions:{
            'EBSEnabled': 'True',
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies: "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/my-test-domain/*
\"}]}",
         NodeToNodeEncryptionOptions:{
            'Enabled': 'True'
```

```
        }
    });
    const response = await client.send(command);
    console.log("Creating domain...");
    console.log(response);
}

async function updateDomain(client, domainName) {
    // Updates the domain to use three data nodes instead of five.
    var command = new UpdateDomainConfigCommand({
        DomainName: domainName,
        ClusterConfig: {
        'InstanceCount': 3
        }
    });
    const response = await client.send(command);
    console.log('Sending domain update request...');
    console.log(response);
}

async function deleteDomain(client, domainName) {
    // Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes.
    var command = new DeleteDomainCommand({
        DomainName: domainName
    });
    const response = await client.send(command);
    console.log('Sending domain deletion request...');
    console.log(response);
}

async function waitForDomainProcessing(client, domainName) {
    // Waits for the domain to finish processing changes.
    try {
        var command = new DescribeDomainCommand({
            DomainName: domainName
        });
        var response = await client.send(command);

        while (response.DomainStatus.Processing == true) {
            console.log('Domain still processing...')
            await sleep(15000) // Wait for 15 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
```

```
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        // Once we exit the loop, the domain is available.
        console.log('Amazon OpenSearch Service has finished processing changes for your
 domain.');
        console.log('Domain description:');
        console.log(response);

    } catch (error) {
        if (error.name === 'ResourceNotFoundException') {
            console.log('Domain not found. Please check the domain name.');
            }
    };
}
```

在 Amazon OpenSearch 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 (URI) 中使用这些名称。服务器和应用程序通常会记录 HTTP 请求，从而会在 URI 包含敏感信息的情况下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 _bulk API 的响应包含相当多的信息。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 S OpenSearch ervice REST API，但对于经常调用的 API 尤其有用，例如_index和 _bulk API：

```
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。

- 您不能将 Zstandard 压缩编解码器（`"index.codec": "zstd"`或`"index.codec": "zstd_no_dict"`）用于 k-nn 索引或安全分析索引。

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，服务会自动将数据传送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 AWS 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 AWS 区域中都可用。有关更多信息，请参阅 AWS Lambda 开发人员指南中的 Lambda 入门和 AWS 一般参考 中的 AWS 服务端点。

主题

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 AWS 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：

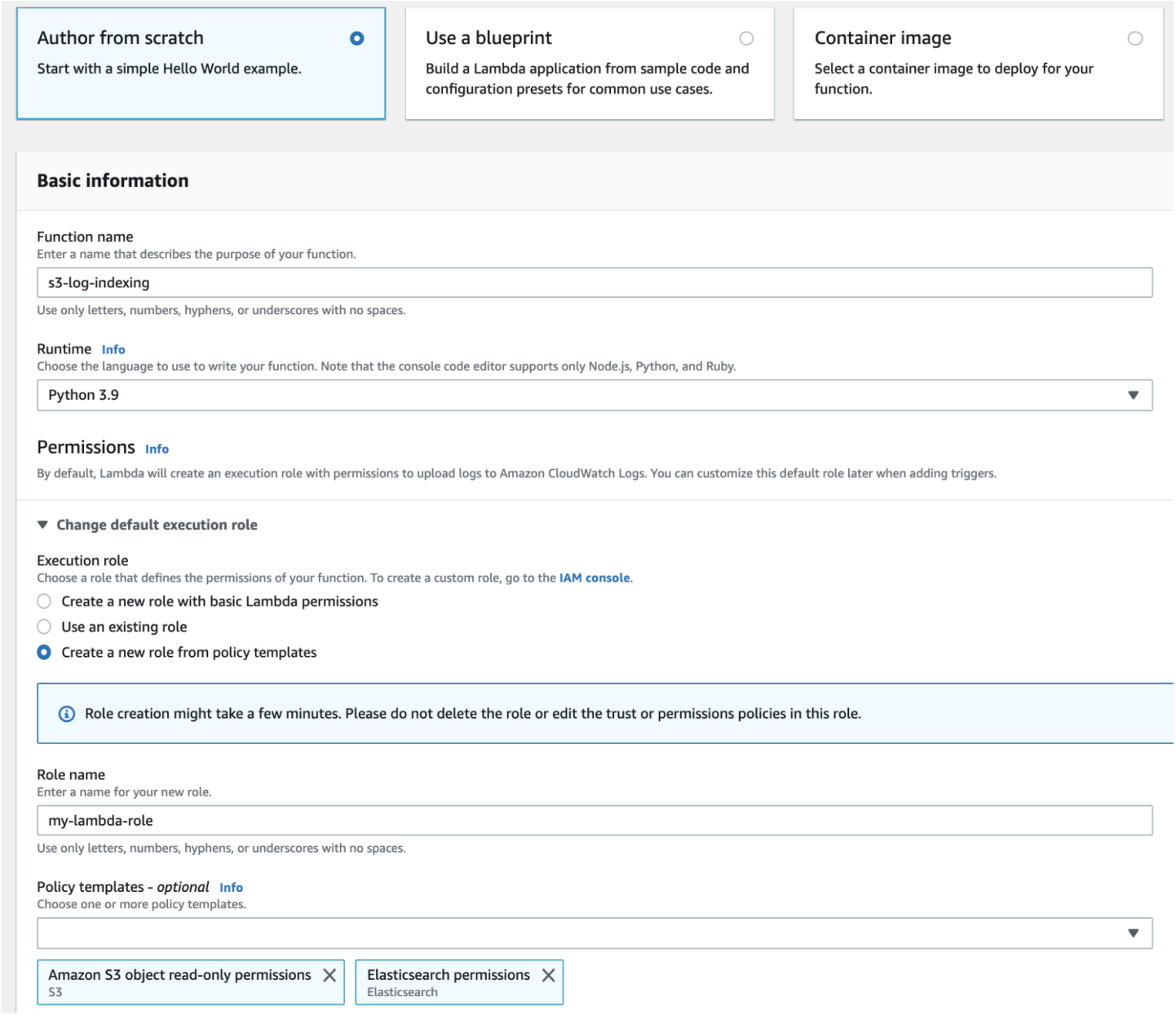
```
cd package
zip -r ../lambda.zip .

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

先决条件	描述

```
      "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：

```
{
  "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 AWS 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-000000000000:495835116157626994950129608214214566865294366804960870 42",
      "_score": 1,
      "_source": {
        "timestamp": 1523648740.051,
        "message": "My test data.",
        "id":
 "shardId-000000000000:495835116157626994950129608214214566865294366804960870 42"
      }
    }
  ]
```

```
}
```

从 Amazon DynamoDB 表中加载流数据

您可以使用 AWS 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： `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "es:ESHttpPost", "es:ESHttpPut", "dynamodb:DescribeStream", "dynamodb:GetRecords", "dynamodb:GetShardIterator", "dynamodb:ListStreams", "logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents" ],`

先决条件	描述

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

角色必须拥有以下信任关系：

```
{
  "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 AWS 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，可为其提供向 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。

从 AWS IoT表中加载流数据

AWS IoT 您可以使用规则发送数据。要了解更多信息，请参阅《AWS IoT 开发者指南》中的OpenSearch操作。

使用 Logstash 将数据加载到 Amazon OpenSearch Service

开源版本的 Logstash (Logstash OSS) 提供了一种使用批量 API 将数据上传到您的 Amazon OpenSearch Service 域的便捷方法。该服务支持所有标准 Logstash 输入插件，包括 Amazon S3 输入插件。OpenSearch Service 支持 logstash-output-opensearch 输出插件，后者支持基本身份验证和 IAM 凭证。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 Service 域使用具有 HTTP 基本身份验证的对访问权限的精细控制，则配置与任何其他 OpenSearch 集群相似。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

如果您的域对主用户使用基于 IAM 的域访问策略或精细访问控制，必须使用 IAM 凭证对所有 OpenSearch Service 请求进行签名。以下基于身份的策略将所有 HTTP 请求授予域的子资源。

```
{
  "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 => "my-s3-bucket"
    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 Service 域位于 VPC 中，则 Logstash OSS 系统必须能够连接到 VPC，并且可以通过 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 Amazon 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 API。某些请求可能不适用于旧 Elasticsearch 版本。

主题

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，请使用 HTTP 请求正文和 OpenSearch域特定语言 (DSL) 进行查询。查询 DSL 允许您指定所有 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 封装在标签中，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}
```

Amazon OpenSearch 服务的定制套餐

Amazon Ser OpenSearch vice 允许您上传自定义词典文件，例如停用词和同义词，还提供了几个预先打包的可选插件，您可以将这些插件与您的域名相关联。这些类型文件的通用术语都是程序包。

字典文件告诉你 OpenSearch 要忽略某些高频单词，或者将 “冷冻蛋奶冻”、“冰淇淋” 和 “冰淇淋” 等词视为等效词，从而改善搜索结果。它们也可以改进词干提取，如在日语 (kuromoji) 分析插件中。

可选插件可以为您的域提供更多功能。例如，您可以使用 Amazon Personalize 插件为您提供个性化的搜索结果。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 “插件（按引擎版本）”。

主题

- 程序包权限要求
- 将程序包上传到 Amazon S3
- 导入和关联程序包
- 将包与一起使用 OpenSearch
- 更新程序包
- 字典的手动索引更新
- 取消程序包关联并移除程序包

程序包权限要求

没有管理员访问权限的用户需要执行某些 AWS Identity and Access Management (IAM) 操作才能管理软件包：

- `es:CreatePackage`-在 OpenSearch 服务区域创建软件包
- `es:DeletePackage`-从 OpenSearch 服务区域删除包裹
- `es:AssociatePackage`-将软件包与域关联
- `es:DissociatePackage`-将程序包与域取消关联

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 “Identity and Access Management”。

将程序包上传到 Amazon S3

因为已预安装了可选的插件包，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。

如果您的字典包含敏感信息，请在上传时使用 S3 托管密钥指定服务器端加密。 OpenSearch 服务无法访问您使用 AWS KMS 密钥保护的 S3 上的文件。

上传文件后，记下其 S3 路径。路径格式为 `s3://`*`bucket-name`*`/`*`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

控制台是将自定义词典导入 S OpenSearch ervice 的最简单方法。当您从 Amazon S3 导入字典时，OpenSearch 服务会存储自己的包副本，并使用 OpenSearch 服务托管密钥使用 AES-256 自动加密该副本。

可选插件已预先安装在 S OpenSearch ervice 中，因此您无需自己上传，但需要将插件与域关联起来。控制台中的程序包屏幕上列出了可用的插件。

导入包并将其与域相关联 AWS Management Console

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. 为自定义字典指定一个描述性名称。
4. 提供文件的 S3 路径，然后选择 Submit（提交）。
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。可选插件将自动可用。
7. 选择关联到域。
8. 选择一个域，然后选择 Associate（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。

或者，也可以使用 AWS CLI、SDK 或配置 API 来导入和关联软件包。有关更多信息，请参阅AWS CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

将包与一起使用 OpenSearch

本节介绍如何使用这两种类型的软件包：自定义字典和可选插件。

使用自定义字典

将文件与域关联后，您可以在创建标记器和标记筛选条件时，在诸如 `synonyms_path`、`stopwords_path` 和 `user_dictionary` 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。

此请求还会在映射中添加一个文本字段 (`description`)，OpenSearch 并告知使用新的分析器作为其搜索分析器。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。

使用可选插件

OpenSearch 服务允许您将预安装的可选 OpenSearch 插件关联到您的域中。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。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 OpenSearch 服务提出请求时引用和使用该插件。

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 “通常会导发蓝/绿部署的更改”。

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，请参阅对搜索结果进行个性化设置。 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。或者，您可以使用更新的数据创建新包，使用此新包创建新索引，将现有索引重新编入新索引，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，您可以通过 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 `TXT-DICTIONARY` 类型的自定义软件包上传，并在其他设置中提供其软件包 ID。

```
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

因为已为您更新了可选的插件包，所以本节仅介绍如何更新自定义字典程序包。将新版本的字典上传到 Amazon S3 不会自动更新亚马逊 OpenSearch 服务上的压缩包。 OpenSearch 服务会存储自己的文件副本，因此，如果您将新版本上传到 S3，则必须手动对其进行更新。

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 中更新软件包，然后应用更新。

使用更新软件包 AWS Management Console

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 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：

   - 如果您的域名正在运行 OpenSearch 或 Elasticsearch 7.8 或更高版本，并且仅使用可更新字段设置为 true 的搜索分析器，则无需采取任何进一步的操作。 OpenSearch 服务使用 _plugins/_refresh_search_ analyzers API 自动更新您的索引。
   - 如果您的域运行的是 Elasticsearch 7.7 或更早版本、使用索引分析器或不使用该`updateable`字段，请参阅。the section called “字典的手动索引更新”

尽管控制台是最简单的方法，但您也可以使用 AWS CLI、软件开发工具包或配置 API 来更新 OpenSearch 服务包。有关更多信息，请参阅AWS CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

使用 AWS SDK 更新软件包

您可以使用 SDK 自动执行更新流程，而不是在控制台中手动更新软件包。以下示例 Python 脚本将新的包文件上传到 Amazon S3，更新 OpenSearch 服务中的软件包，并将新包应用于指定的域。确认更新成功后，它会进行示例调用，以 OpenSearch演示已应用新的同义词。

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 AWS 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')
> ```

字典的手动索引更新

手动索引更新仅适用于自定义词典，不适用于可选插件。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可选插件）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。解除软件包关联后，使用该软件包的现有索引将无法再使用它。必须先将包从任何索引中移除，然后才能将其解除关联，否则解除关联将失败。

控制台是将软件包与域解除关联并将其从 OpenSearch Service 中移除的最简单方法。从 OpenSearch 服务中移除包裹并不会将其从 Amazon S3 上的原始位置移除。

使用 AWS Management Console取消程序包与域的关联

1. 转至 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，选择您的域，然后选择 Packages (程序包) 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 AWS CLI、SDK 或配置 API 来解除关联和移除软件包。有关更多信息，请参阅AWS CLI 命令参考和亚马逊 OpenSearch 服务 API 参考。

使用 SQL 查询您的亚马逊 OpenSearch 服务数据

您可以使用 SQL 来查询您的亚马逊 OpenSearch 服务，而不必使用基于 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。SQL 支持适用于运行 OpenSearch 或 Elasticsearch 6.5 或更高版本的域名。

> **Note**
>
> 本文档描述了 OpenSearch 服务与各种版本的 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的信息，请参阅开源OpenSearch文档。

使用下表查找每个版本 OpenSearch 和 Elasticsearch 版本支持的 SQL 插件版本。

OpenSearch

OpenSearch 版本	SQL 插件版本	显著功能
2.13.0	2.13.0.0	

OpenSearch 版本	SQL 插件版本	显著功能
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**
>
> 如果您的域名运行的是 Elasticsearch 而不是 OpenSearch，则格式为。`_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 仪表板用户界面，允许你运行按需 SQL 查询、将 SQL 转换为 REST 等效项，以及以文本、JSON、JDBC 或 CSV 的形式查看和保存结果。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

Java 数据库连接 (JDBC) 驱动程序允许您将 OpenSearch 服务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。

下表汇总了驱动程序的版本兼容性。

OpenSearch

OpenSearch 版本	JDBC 驱动程序版本
2.13	1.1.0.1
2.1.1	1.1.0.1
2.9	1.1.0.1
2.7	1.1.0.1
2.5	1.1.0.1
2.3	1.1.0.1
1.3	1.1.0.1
1.2	1.1.0.1
1.1	1.1.0.1
1.0	1.1.0.1

Open Distro for Elasticsearch

Elasticsearch 版本	JDBC 驱动程序版本
7.10	1.13.0
7.9	1.11.0
7.8	1.9.0
7.7	1.8.0
7.4	1.4.0
7.1	1.0.0
6.8	0.9.0
6.7	0.9.0
6.5	0.9.0

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 构件页面上下载工作驱动程序文件示例。有关安装驱动程序的信息，请参阅上的 SQL 存储库 GitHub。

在 Amazon 服务中搜索 k 最近邻 (k-nn) OpenSearch

Amazon S OpenSearch ervice 的 k-nn 是其关联的 k 最近邻算法的缩写，它允许您在向量空间中搜索点，并通过欧几里得距离或余弦相似度找到这些点的 “最近邻”。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**
>
> 本文档描述了 OpenSearch 服务与各种版本的 k-nn 插件之间的版本兼容性，以及将插件与托管 OpenSearch 服务一起使用时的限制。有关 k-nn 插件的全面文档，包括简单和复杂的示

例、参数参考以及插件的完整 API 参考，请参阅开源OpenSearch 文档。开源文档还涵盖了性能调整和特定于 K-NN 的集群设置。

使用下表查找在您的亚马逊 OpenSearch 服务域上运行的 k-nn 插件的版本。每个 k-nn 插件版本都对应一个OpenSearch或 Elasticsearch 版本。

OpenSearch

OpenSearch 版本	k-NN 插件版本	显著功能
2.13	2.13.0.0	
2.1.1	2.11.0.0	添加对 k-NN 查询中 `ignore_unmapped` 的支持
2.9	2.9.0.0	使用 Faiss 引擎实现了 k-NN 字节向量和高效过滤
2.7	2.7.0.0	
2.5	2.5.0.0	针对 k-nn 模型系统索引进行了扩展， SystemIndexPlugin 为核心 HybridFS 添加了特定于 Lucene 的文件扩展名
2.3	2.3.0.0	
1.3	1.3.0.0	
1.2	1.2.0.0	增加了对 Faiss 库的支持
1.1	1.1.0.0	
1.0	1.0.0.0	已重命名 REST API，同时支持向后兼容性，将命名空间重命名从 `opendistro` 到 `opensearch`

Elasticsearch

Elasticsearch 版本	k-NN 插件版本	显著功能
7.1	1.3.0.0	欧氏距离

Elasticsearch 版本	k-NN 插件版本	显著功能
7.4	1.4.0.0	
7.7	1.8.0.0	余弦相似性
7.8	1.9.0.0	
7.9	1.11.0.0	热身 API，自定义评分
7.10	1.13.0.0	Hamming 距离、L1 Norm 距离、Painless 脚本

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，根据knn.memory.circuit_breaker.limit统计数据和该实例类型的可用 RAM 检查每个数据节点上的KNNGraphMemoryUsage指标。 OpenSearch 服务将实例内存的一半用于 Java 堆（堆

大小不超过 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。

如果索引使用近似的 k-nn ()，则无法将 k-n n 索引迁移到UltraWarm或冷存储。`"index.knn": true`如果 `index.knn` 已设为 `false`（准确 k-NN），则您可以将索引移至其他存储层。

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 AWS 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 Amazon OpenSearch 服务域相比，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 AWS 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 AWS 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

- 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。
- 您无法连接到自我管理的 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，并且想要升级其中一个或两个域，则必须先删除其中一个连接。
- 不能将自定义字典或 SQL 用于跨集群搜索。
- 您不能 AWS 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
- N 已启用ode-to-node 加密

跨集群搜索定价

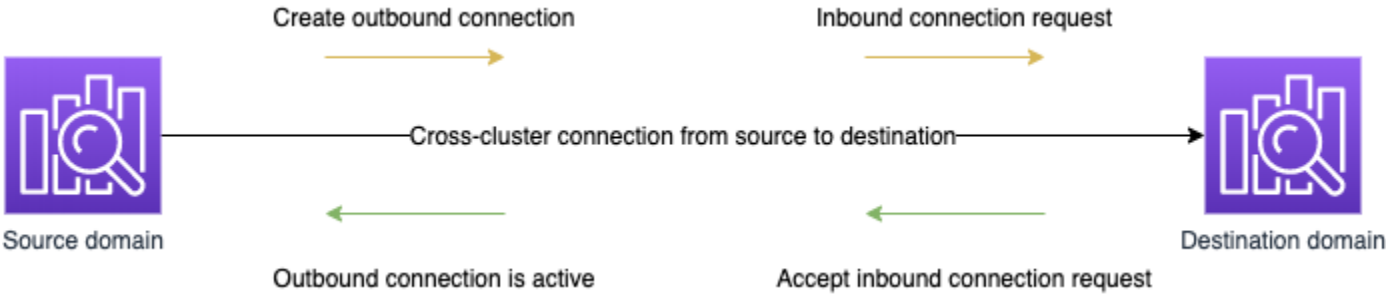
跨域搜索不收取额外的费用。

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 AWS 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 AWS 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 AWS 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进行任何跨集群操作。

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。

```
{
  "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`。

```
{
  "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，除了标准read或相关索引的indices:admin/shards/search_shards权限外，用户还必须拥有search相应权限。

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：

```
{
  "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：

```
{
  "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": "Lets unite the new mutants",
          "likes": 0
        }
      },
      {
        "_index": "cluster_b:b_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 2,
        "_source": {
          "user": "domino",
          "message": "I'm different",
          "likes": 0
        }
      },
      {
        "_index": "cluster_c:c_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 3,
        "_source": {
          "user": "domino",
          "message": "So am I",
          "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。

学习如何使用 Amazon OpenSearch 服务进行排名

OpenSearch 使用名为 BM-25 的概率排名框架来计算相关性分数。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 XGBoost 和 Ranklib 库中的模型来重新评分搜索结果。Elasticsearch LTR 插件最初由 Conn OpenSource ections 开发，维基媒体基金会、Snagajob Engineering、Bonsai 和 Yelp Engineering 做出了重要贡献。该插件的 OpenSearch 版本源自 Elasticsearch LTR 插件。

学习排名需要 OpenSearch 或 Elasticsearch 7.7 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”。

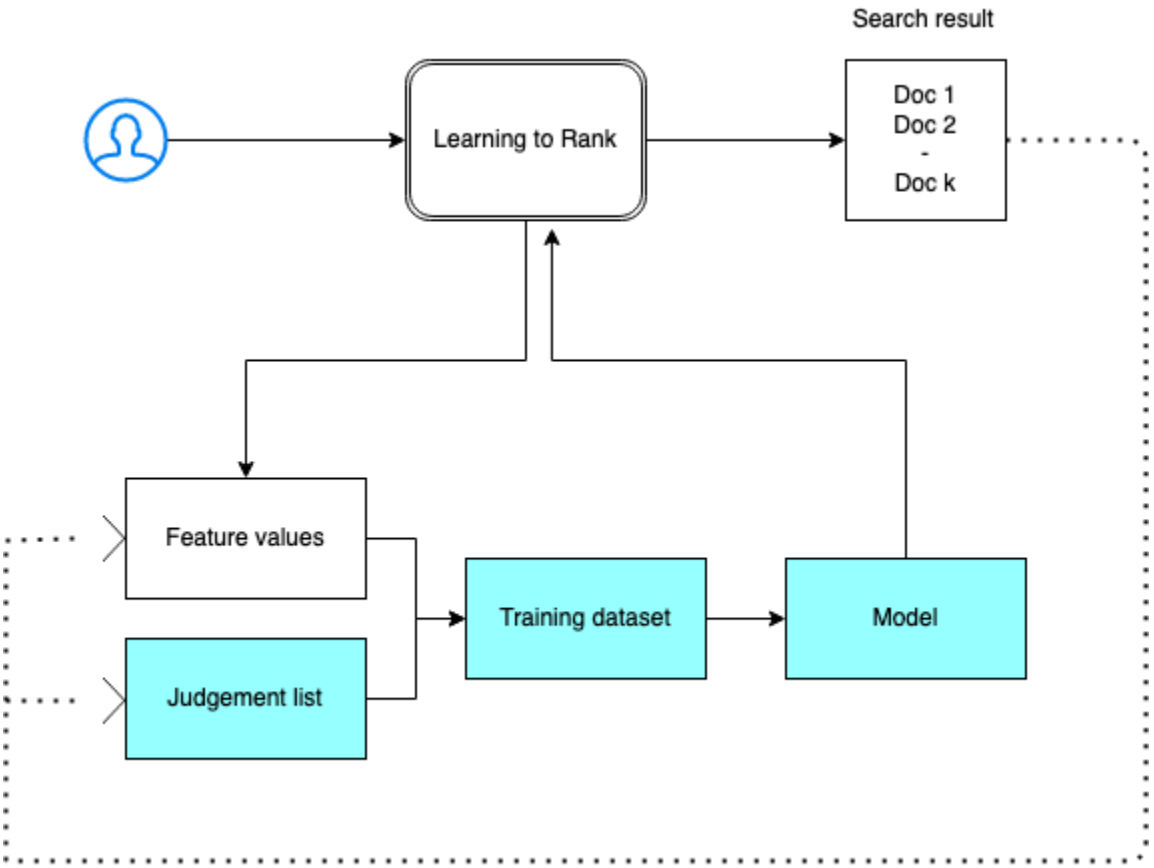
> **Note**
>
> 本文档概述了 Learning to Rank 插件，并帮助您开始使用它。有关完整文档，包括详细步骤和 API 说明，请参阅学习排名文档。

主题

- 学习排名入门
- 学习排名 API

学习排名入门

您需要提供判断清单，准备训练数据集，然后在 Amazon S OpenSearch ervice 之外训练模型。蓝色的部件不在 OpenSearch 服务范围内：



步骤 1：初始化插件

要初始化 Learning to Rank 插件，请向您的 OpenSearch 服务域发送以下请求：

```
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 OpenSearch 服务之外执行此步骤。

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 OpenSearch 服务之外执行此步骤。

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 OpenSearch 服务之外执行此步骤。

训练数据集到位后，下一步是使用 XGBoost 或 Ranklib 库构建模型。XGBoost 和 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，请分别参阅 X GBoost 和文档。RankLib要使用亚马逊 SageMaker 构建 XGBoost 模型，请参阅 X GBoost 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"
        }
      }
    ]
  }
}
```

如果您在不使用 “学习排名” 插件的情况下进行搜索，OpenSearch 则会返回不同的结果：

```
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。

字段名称	描述
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。
stores.model_count	型号数。
status	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
```

亚马逊 OpenSearch 服务中的异步搜索

通过异步搜索 Amazon Ser OpenSearch vice，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 Elasticsearch 7.10 或更高版本。

本文档简要概述了异步搜索。它还讨论了在托管的 Amazon S OpenSearch ervice 域而不是开源 OpenSearch 集群中使用异步搜索的局限性。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 文档中的异步搜索。

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 OpenSearch 版本，请在所有异步搜索`_plugins`请求`_opendistro`中替换为。

您可以指定以下异步搜索选项：

Options	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

Options	描述	默认值	必需
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 OpenSearch 服务中，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" :
 "Fm9pYzJyVG91U19xb0hIQUJnMHJfRFEAAAAAAAAknghQ1OWVBczNZQjVEa2dMYTBXaTdEagAAAAAAAAAB",
  "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 索引进行异步搜索仍然有效。有关更多信息，请参阅 the section called “UltraWarm 存储”。

> **Note**
>
> 您可以在中监控异步搜索统计信息 CloudWatch。有关指标的完整列表，请参阅the section called “异步搜索指标”。

在 Amazon OpenSearch 服务中搜索时间点

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 搜索的主要用途是将其与`search_after`功能相结合。这是中的首选分页方法 OpenSearch，特别是对于深度分页，因为它在时间上冻结的数据集上运行，不绑定到查询，并且支持向前和向后一致的分页。您可以将 PIT 与运行 OpenSearch 版本 2.5 的域一起使用。

> **Note**
>
> 本主题概述了 PIT，以及在托管 Amazon S OpenSearch ervice 域而不是自管理 OpenSearch 集群上使用 PIT 时需要考虑的一些事项。有关 PIT 的完整文档，包括全面的 API 参考，请参阅开源 OpenSearch 文档中的时间点。

注意事项

在配置 PIT 搜索时，请考虑以下事项：

- 如果您要从运行 2.3 OpenSearch 版本的域名升级，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`_search/point_in_time`使用以下格式向发送 HTTP 请求：

```
POST opensearch-domain/my-index/_search/point_in_time?keep_alive=time
```

可以指定以下 PIT 选项：

Options	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

Options	描述	默认值	必需
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 OpenSearch 版本 2.5 的域并且需要精细的访问控制，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 及更高版本的域名，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，您目前无法修改设置。

跨集群搜索

您可以跨集群创建 PIT、使用 PIT ID 搜索、列出 PIT 以及删除 PIT，但有以下几项限制：

- 您只能在源域上列出和删除所有 PIT。
- 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 UltraWarm 索引进行的 PIT 搜索继续有效。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”。

Amazon OpenSearch 服务中的语义搜索

从 2.9 OpenSearch 版开始，您可以使用语义搜索来帮助您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-nearest Ne ighbor (k-nn) 搜索。

借助 OpenSearch 服务，您可以为外部服务设置 AI 连接器。 AWS 服务使用控制台，您还可以使用 AWS CloudFormation 模板创建机器学习模型。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，包括语义搜索的使用 step-by-step 指南，请参阅开源文档中的语义搜索。OpenSearch

在 Amazon OpenSearch 服务中进行并行区段搜索

从 OpenSearch 版本 2.13 开始，您可以使用并行区段搜索来帮助您在查询阶段并行搜索区段。有关并行区段搜索的完整文档，请参阅开源 OpenSearch 文档中的并行区段搜索。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。

当您在 Amazon S OpenSearch ervice 中使用当前区段搜索时，还有一些其他限制适用：

- 您无法在 S OpenSearch ervice 中启用索引级别的并行区段搜索。
- 默认情况下，S OpenSearch ervice 使用最大切片计数机制的 2 个切片计数。

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 OpenSearch 控制面板的安装。 OpenSearch 仪表板在域中的热门数据节点上运行。

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 的域名，网址为。*domain-endpoint*/_plugin/kibana

使用此默认 OpenSearch 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。

以下各节介绍 OpenSearch 仪表板的一些常见用例：

- the section called “控制对 OpenSearch 仪表板的访问权限”
- the section called “将 OpenSearch 仪表板配置为使用 WMS 地图服务器”
- the section called “将本地仪表板服务器连接到 OpenSearch 服务”

控制对 OpenSearch 仪表板的访问权限

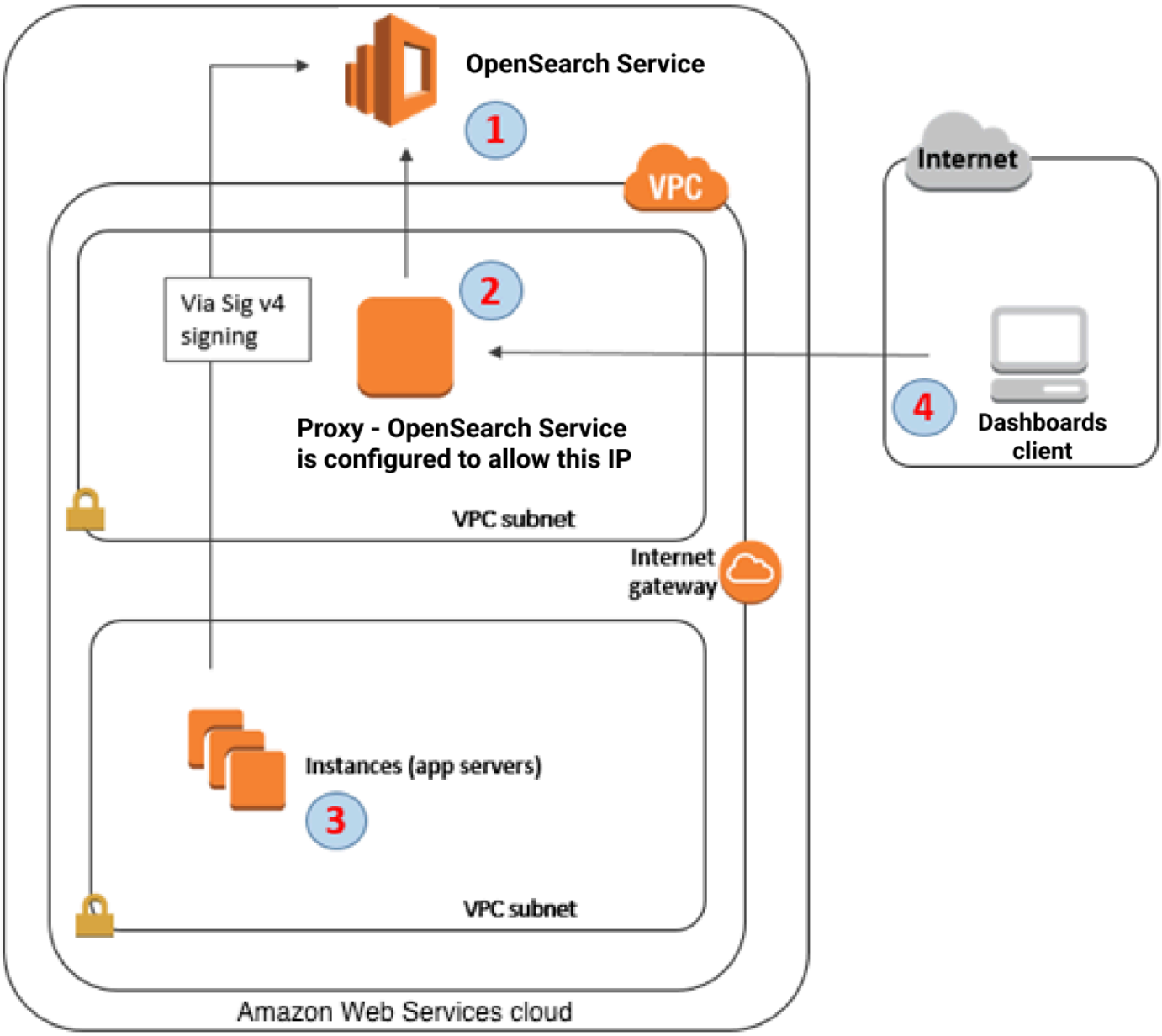
仪表板本身不支持 IAM 用户和角色，但 OpenSearch 服务提供了多种控制仪表板访问权限的解决方案：

- 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 OpenSearch 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 OpenSearch 仪表板的访问权限”。

由于 Dashboards 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 OpenSearch 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. OpenSearch 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：

```
{
```

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

建议使用弹性 IP 地址配置在代理服务器上运行的 EC2 实例。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;
    }
}
```

将 OpenSearch 仪表板配置为使用 WMS 地图服务器

默认安装的 Dash OpenSearch board OpenSearch s for Service 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. 找到 visualization: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 OpenSearch 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 OpenSearch 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'
```

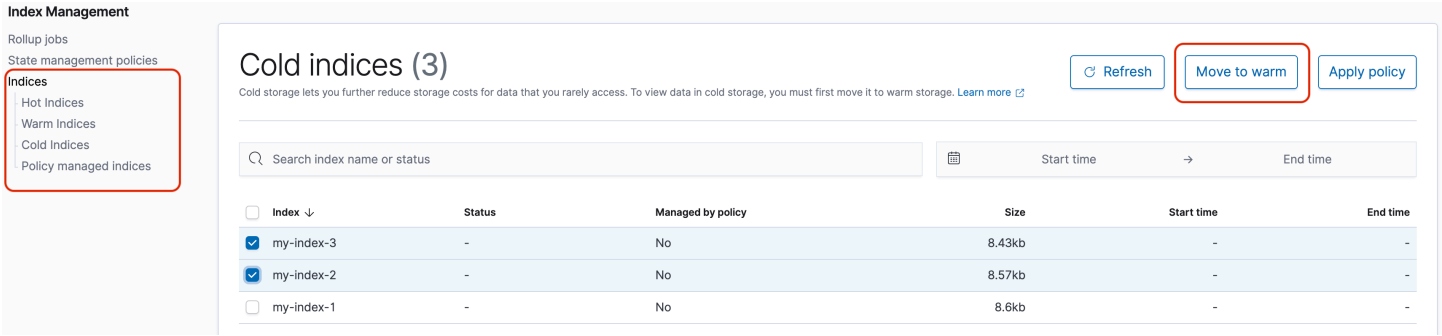
```
opensearch.password: 'password'
opensearch.requestHeadersWhitelist: [authorization, securitytenant,
 security_tenant]
```

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 OpenSearch 仪表板中管理索引

OpenSearch 服务域上安装的 OpenSearch 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 OpenSearch 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

管理 Amazon OpenSearch Service 中的索引

向 Amazon OpenSearch Service 中添加数据后，通常需要对该数据重建索引，使用索引别名，将索引移动到更经济高效的存储中，或者将其完全删除。本章介绍 UltraWarm 存储、冷存储和索引状态管理。有关 Open Search 索引 API 的信息，请参阅 OpenSearch 文档。

主题

UltraWarm 亚马逊 OpenSearch 服务的存储空间

UltraWarm 为在 Amazon OpenSearch 服务上存储大量只读数据提供了一种经济实惠的方式。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 节点不使用附加存储，而是使用 Amazon S3 和复杂的缓存解决方案来提高性能。对于不主动写入、查询频率较低且不需要相同性能的索引， UltraWarm 可以显著降低每 GiB 数据的成本。因为除非将温索引返回到热存储，否则它们 UltraWarm 是只读的，因此最适合存储不可变的数据，例如日志。

在中 OpenSearch，暖索引的行为与任何其他索引一样。您可以使用相同的 API 查询它们，也可以使用它们在 OpenSearch 仪表板中创建可视化效果。

主题

- 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
- 正在禁用 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。
- 使用带备用域的多可用区时，温节点的数量必须是所用可用区数量的倍数。
- 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`)，则您无法将其移至热存储。
- 如果网域使用精细的访问控制，则必须将用户映射到控制 OpenSearch 面板中的`ultrawarm_manager`角色才能进行 UltraWarm API 调用。

> **Note**
>
> 可能无法在某些先前存在的 OpenSearch 服务域上定义该`ultrawarm_manager`角色。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

如果您在先前存在的 OpenSearch 服务域 UltraWarm 上启用，则可能无法在该域上定义该`ultrawarm_manager`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。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. 创建角色后，将其映射到任何将管理 UltraWarm 索引的用户或后端角色。

UltraWarm 存储要求和性能注意事项

如所述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。

由于它使用 Amazon S3，因此不会 UltraWarm 产生任何开销。在计算 UltraWarm 存储需求时，您只考虑主分片的大小。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。通过分配给每种 UltraWarm 实例类型的 CPU 内核数量和 RAM 量，您可以大致了解它们可以同时搜索的分片数量。请注意，虽然在 S3 中只有主分片计入 UltraWarm存储空间，但 OpenSearch 控制面板和`_cat/indices`仍将 UltraWarm 索引大小报告为所有主分片和副本分片的总和。

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。就像任何其他 OpenSearch 工作负载一样，确定是否 UltraWarm 满足您的需求的最重要步骤是使用真实的数据集进行具有代表性的客户测试。

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。

对于 UltraWarm 存储空间，您需要为实际使用量付费。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，您还需要为每个 UltraWarm 节点支付小时费率。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 UltraWarm 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。即使在域状态从 “处理中” 变为 “活动” 之后，也 UltraWarm 可能在几个小时内无法使用。

使用带备用域的多可用区时，温节点的数量必须是所用可用区数量的倍数。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AWS CLI或配置 API 来启用 UltraWarm，特别是中的WarmEnabledWarmCount、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 “配额”。

示例 CLI 命令

以下 AWS CLI 命令创建一个具有三个数据节点、三个专用主节点、六个温节点并启用了细粒度访问控制的域：

```
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 AWS 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 hot 迁移到 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 服务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。

温存储中的索引是只读的，除非您将它们返回到热存储，这 UltraWarm 最适合存储不可变的数据，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

索引迁移到 UltraWarm 存储需要强制合并。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 片段组成。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，将索 UltraWarm 引合并为一个段。

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 按顺序处理队列中的迁移。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 添加了另外两个选项，类似于_all，以帮助管理热索引和温索引。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
```

一次最多可以有 10 个从温存储到热存储的排队迁移。 OpenSearch 服务按排队顺序逐一处理迁移请求。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

除了用于自动快照的标准存储库外，还 UltraWarm 添加了第二个用于温索引的存储库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rename_pattern和rename_replacement之类的选项。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 不在手动快照中包含热索引。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

如果您不经常查询数据 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the section called “将索引迁移到冷存储”。

正在禁用 UltraWarm

控制台是最简单的禁用方法 UltraWarm。选择域、Actions（操作）和 Edit cluster configuration（编辑集群配置）。取消选择 “启用 UltraWarm 数据节点”，然后选择 “保存更改”。还可以在 AWS CLI 和配置 API 中使用 `WarmEnabled` 选项。

在禁用之前 UltraWarm，必须删除所有热索引或将其迁移回热存储。热存储空间为空后，请等待五分钟，然后再尝试禁用 UltraWarm。

适用于 Amazon OpenSearch 服务的冷存储

冷存储允许您在您的 Amazon Ser OpenSearch vice 域中存储任意数量的不经常访问的数据或历史数据，并按需进行分析，成本低于其他存储层。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。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

冷存储具有以下先决条件：

- 冷存储需要 OpenSearch 或 Elasticsearch 版本 7.9 或更高版本。
- 要在 OpenSearch 服务域上启用冷存储，还必须在同一域 UltraWarm 上启用冷存储。
- 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`)，则您无法将其移至冷存储。
- 如果域使用精细的访问控制，则必须将非管理员用户映射到 OpenSearch 仪表板中的`cold_manager`角色才能管理冷索引。

> **Note**
>
> 某些先前存在的 OpenSearch 服务域中可能不存在该`cold_manager`角色。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

如果您在先前存在的 OpenSearch 服务域上启用冷存储，则可能无法在该域上定义该`cold_manager`角色。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，因此不会产生任何热存储开销，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。在 Amazon CloudWatch 中监控该`ColdStorageSpaceUtilization`指标，了解您正在使用多少冷存储空间。

冷存储定价

与 UltraWarm 存储类似，使用冷存储时，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。有关冷存储定价的更多信息，请参阅 Amazon OpenSearch 服务定价。

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，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 AWS CLI 或者配置 API以启用冷存储。

示例 CLI 命令

以下 AWS CLI 命令创建一个包含三个数据节点、三个专用主节点、启用冷存储并启用细粒度访问控制的域：

```
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 AWS 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 OpenSearch 仪表板中管理冷索引

您可以使用 OpenSearch 服务域中现有的仪表板界面管理热索引、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 仪表板中的索引。

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 服务一次将一个索引迁移到冷存储。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，该策略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，以决定要迁移到哪些索引以 UltraWarm 供进一步分析。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范围后，将其迁移回可以查询数据 UltraWarm 的位置，然后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。您无法将已删除的冷索引直接恢复到冷层。 OpenSearch 删除冷索引后，服务会将其保留 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 仪表板不支持 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 AWS 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。

亚马逊 OpenSearch 服务的 OR1 存储空间

OR1 是 Amazon OpenSearch 服务的实例系列，它提供了一种经济实惠的方式来存储大量数据。拥有 OR1 实例的域使用亚马逊弹性区块存储 (Amazon EBS) `gp3` 或`io1`卷作为主存储，数据到达时同步复制到亚马逊 S3。这种存储结构提供更高的索引吞吐量和较高的耐久性。OR1 实例系列还支持在发生故障时自动恢复数据。有关 OR1 实例类选项的信息，请参阅 the section called “当前一代实例类型”。

如果您正在运行索引繁重的运营分析工作负载，例如日志分析、可观察性或安全分析，则可以从 OR1 实例提高的性能和计算效率中受益。此外，OR1 实例提供的自动数据恢复功能可提高域的整体可靠性。

OpenSearch 服务将与存储相关的 OR1 指标发送给 Amazon。 CloudWatch有关可用指标的列表，请参阅 ???。

OR1 实例按需提供，也可以按预留实例定价提供，在 Amazon EBS 和 Amazon S3 中预置的实例和存储按小时费率提供。

主题

限制

为您的域使用 OR1 实例时，请考虑以下限制。

- 您的域名必须运行 2.11 或更高 OpenSearch 版本。
- 您的域名必须启用静态加密。有关更多信息，请参阅 ???。
- 您的域名必须是新域名。您无法修改现有域以使用 OR1 实例。
- 如果您的域使用专用主节点，则它们必须使用 Graviton 实例。有关专用主节点的更多信息，请参阅???。
- OR1 实例上的分片大小必须小于 100 GiB。大于 100 GiB 的分片可能会减慢恢复时间。如果您在 OR1 实例上创建大于 100 GiB 的分片 OpenSearch ，则服务会阻止向该域写入请求。如果您仍想使用大于 100 GiB 的分片，AWS Support请联系申请增加配额。
- OR1 实例上索引的刷新间隔必须为 10 秒或更长。OR1 实例的默认刷新间隔为 10 秒。

OR1 与存储有何不同 UltraWarm

OpenSearch 服务提供经过优化的 UltraWarm 实例，可降低存储热数据的成本。OR1 和 UltraWarm 实例都将数据本地存储在亚马逊 EBS 中，并远程存储在 Amazon S3 中。但是，OR1 和 UltraWarm 实例在几个重要方面有所不同：

- OR1 实例将数据副本保存在本地和远程存储中。 UltraWarm 实例，为了降低存储成本，请将数据主要保存在远程存储中。根据使用模式，他们可能会将其移至本地存储。
- OR1 实例处于活动状态，可以接受读取和写入操作，而在您手动将其移回热存储之前， 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。相比之下，OR1 实例在后台执行复制和恢复。如果出现红色索引，OR1 实例会自动从 Amazon S3 的远程存储中恢复丢失的分片。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???。

使用 OR1 实例

使用、 AWS Command Line Interface (AWS CLI) 或 AWS SDK 创建新域时 AWS Management Console，您可以为数据节点选择 OR1 实例。然后，您可以使用现有工具对数据进行索引和查询。

控制台

1. 导航到亚马逊 OpenSearch 服务控制台，网址为https://console.aws.amazon.com/aos/。
2. 在左侧导航窗格中，选择 域。
3. 选择创建域。
4. 输入域名以及其他首选选项。在实例系列下，选择 OR1。选择创建按钮，开始域创建过程。

AWS CLI

1. 导航到您的 AWS CLI 终端。如果需要安装 AWS CLI，请参阅安装或更新最新版本的 AWS CLI。
2. 要使用 OR1 存储，您必须在创建`InstanceType`域时在字段中提供特定 OR1 实例类型大小的值。您还必须启用静态加密。

   以下示例使用大小为 `2xlarge` 的 OR1 实例创建域。

```
aws opensearch create-domain \
  --domain-name test-domain \
  --engine-version OpenSearch_2.11 \
  --cluster-config
 "InstanceType=or1.2xlarge.search,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedMast
 \
  --ebs-options "EBSEnabled=true,VolumeType=gp3,VolumeSize=200" \
  --encryption-at-rest-options Enabled=true \
  --advanced-security-options
 "Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions={MasterUserName=test-
user,MasterUserPassword=test-password}" \
  --node-to-node-encryption-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":
{"AWS":"*"},"Action":"es:*","Resource":"arn:aws:es:us-east-1:account-
id:domain/test-domain/*"}]}'
```

Amazon OpenSearch 服务中的索引状态管理

Amazon S OpenSearch ervice 中的索引状态管理 (ISM) 允许您定义自动执行日常任务的自定义管理策略，并将其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 OpenSearch 或 Elasticsearch 6.8 或更高版本。

> **Note**
>
> 本文档简要概述了 ISM 和几个策略示例。它还解释了 ISM for Amazon S OpenSearch ervice 域与 ISM 在自我管理 OpenSearch集群上的区别。有关 ISM 的完整文档，包括全面的参数参考、每项设置的描述以及 API 参考，请参阅 OpenSearch 文档中的索引状态管理。

> **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。此更新引入了一项重大更改，该更改会影响使用此设置的现有 CloudFormation 模板。

创建一个 ISM 策略

开始使用索引状态管理

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
2. 选择要为其创建 ISM 策略的域。
3. 在域名的控制面板中，导航到控制 OpenSearch 面板网址，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 仪表板中选择索引，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 OpenSearch Elasticsearch 相比，适用于亚马逊 OpenSearch 服务的 ISM 有几个区别。

ISM 操作

- OpenSearch 服务支持三种独特的 ISM 操作：warm_migrationcold_migration、和cold_delete：
  - 如果您的域已UltraWarm启用，则该warm_migration操作会将索引转换为热存储。
  - 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。
- 如果您的域名运行 OpenSearch 或 Elasticsearch 7.4 或更高版本，则 OpenSearch 服务支持 ISM open 和操作。close
- 如果您的域名运行 OpenSearch 或 Elasticsearch 7.7 或更高版本，则 OpenSearch 服务支持 ISM 操作。snapshot

冷存储 ISM 操作

对于冷索引，您必须在使用以下 ISM API 时指定 ?type=_cold 参数：

- 添加策略
- 移除策略
- 更新策略
- 重试失败的索引
- 解释索引

这些冷索引的 API 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 允许您使用 API 更改所有可用的 `_cluster/settings` ISM 设置。在 Amazon OpenSearch 服务上，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。

Amazon S OpenSearch ervice 中的@@ 索引状态管理 (ISM) 允许您自动执行重复的索引管理活动，因此您可以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，根据索引年限、大小和其他条件自动执行这些操作，所有这些操作都可以在您的 Amazon S OpenSearch ervice 域中完成。

OpenSearch 服务支持三个存储层：用于主动写入和低延迟分析的默认 “热” 状态， UltraWarm 用于高达 3 PB 的只读数据，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，它会在两天后将索引从默认的热状态迁移到 UltraWarm 存储，30 天后将索引从冷存储迁移到冷存储，最后在 60 天后删除索引。

先决条件

- 您的 OpenSearch 服务域必须运行 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。
- 您的用户角色需要足够的权限才能访问 OpenSearch 服务控制台。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 OpenSearch 控制面板中配置 ISM 策略。

1. 在 OpenSearch 服务控制台的域名控制面板中，导航到控制 OpenSearch 面板网址，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 仪表板中，选择添加示例数据，然后将一个或多个示例索引添加到您的域中。
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。

使用索引汇总汇总 Amazon OpenSearch 服务中的索引

借助 Amazon S OpenSearch ervice 中的索引汇总，您可以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 OpenSearch 或 Elasticsearch 7.9 或更高版本。

> **Note**
>
> 本文档可帮助您开始在 Amazon S OpenSearch ervice 中创建索引汇总任务。有关全面的文档，包括所有可用设置的列表和完整的 API 参考，请参阅 OpenSearch 文档中的索引汇总。

创建索引汇总作业

要开始使用，请选择 “ OpenSearch仪表板中的索引管理”。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 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 之前的版本和旧版 Elasticsearch OSS 版本每次搜索仅支持一个汇总索引。

在 Amazon OpenSearch 服务中转换索引

虽然索引汇总作业允许您通过将旧数据汇总到压缩索引来降低数据粒度，但转换作业允许您围绕某些字段创建不同的汇总数据视图，因此您可以用不同的方式对数据进行可视化或分析。

索引转换具有 OpenSearch 仪表板用户界面和 REST API。该功能需要 OpenSearch 1.0 或更高版本。

> **Note**
>
> 本文档简要概述了索引转换，以帮助您开始在 Amazon S OpenSearch ervice 域上使用索引转换。如需全面的文档和 REST API 参考，请参阅开源 OpenSearch 文档中的索引转换。

创建索引转换任务

如果您的集群中没有任何数据，请使用 OpenSearch 仪表板中的示例飞行数据来尝试转换作业。添加数据后，启动 OpenSearch 仪表板。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子集，请选择 “添加数据过滤器”，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 `DestAirportID` 字段添加到 `DestAirportID_terms`字段的目标字段中，并且您可以在转换任务完成后在转换后的索引中找到分组的机场 ID。
- 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 AWS 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 AWS 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

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 AWS 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
- N 已启用ode-to-node 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：

```
{
  "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 AWS 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 AWS 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 AWS 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 AWS 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 的远程索引，但不支持从 OpenSearch 1.0 到 Elasticsearch 7.10.x 的重新索引。

> **Note**
>
> 本文档介绍如何在 Amazon Ser OpenSearch vice 域之间重新编制数据索引。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 AWS 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 AWS 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。您可以通过使用内置的 VPC 终端节点连接功能建立连接 AWS PrivateLink，也可以配置代理来实现此目的。

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 AWS PrivateLink 连接。 AWS CLI AWS 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 AWS 区域。

使用重新索引数据 AWS 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.aws.amazon.com/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####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 OpenSearch 服务之外，例如在自我管理的 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进行基本身份验证。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，请考虑较小的批处理大小和/或更长的超时。有关更多信息，请参阅滚动搜索。

```
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：

Options	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
region	String	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 OpenSearch 服务中的时间序列数据

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 OpenSearch 版本 1.0 或更高版本。

> **Note**
>
> 本文档提供了一些基本步骤，可帮助您开始使用 Amazon S OpenSearch ervice 域上的数据流。有关全面的文档，请参阅 OpenSearch 文档中的数据流。

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。

因为我们有一个与data_stream对象匹配的索引模板，所以 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 API。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到冷存储，则 OpenSearch 会将该索引从数据流中删除。即使将索引移回到 UltraWarm，索引仍保持独立状态，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：在 OpenSearch 仪表板中管理数据流

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 OpenSearch Service 中的数据

通过警报和异常检测功能，主动监控 Amazon OpenSearch Service 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

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。

提醒需要使用 Elasticsearch 6.2 OpenSearch 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

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。此角色授予查看警报、目标和监视器的权限，但不授予确认警报或修改目标或监视器的权限。

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，您的 OpenSearch 服务域必须能够连接到互联网才能通知 Slack 频道或向第三方服务器发送自定义 webhook。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
  }
}
```

以下请求将警报配置为在七天后自动删除历史索引，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，但想要停止创建每日警报索引，请删除所有警报历史索引：

```
DELETE .plugins-alerting-alert-history-*
```

要减少历史索引的分片数，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。

Amazon OpenSearch 服务中的异常检测

Amazon S OpenSearch ervice 中的异常检测使用随机剪辑森林 (RCF) 算法，近乎实时地自动检测 OpenSearch 数据中的异常。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 Al erting 插件配对，以便在检测到异常时立即通知您。

异常检测适用于运行任何 OpenSearch 版本或 Elasticsearch 7.4 或更高版本的域名。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 Amazon OpenSearch 服务环境中的异常检测。有关全面的文档，包括详细步骤、API 参考、所有可用设置的参考以及创建可视化和仪表板的步骤，请参阅开源 OpenSearch 文档中的异常检测。

先决条件

异常检测具有以下先决条件：

- 异常检测需要 OpenSearch 或 Elasticsearch 7.4 或更高版本。
- 异常检测仅支持 Elasticsearch 7.9 及更高版本以及所有版本的精细访问控制。 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。
- 如果您的域使用精细的访问控制，则必须将非管理员用户映射到 OpenSearch 仪表板中的`anomaly_read_access`角色才能查看检测器或`anomaly_full_access`创建和管理检测器。

异常检测入门

要开始使用，请选择 “ OpenSearch 仪表板中的异常检测”。

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**
>
> count()聚合方法仅在 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"
>      }
>   }
> }
> ```

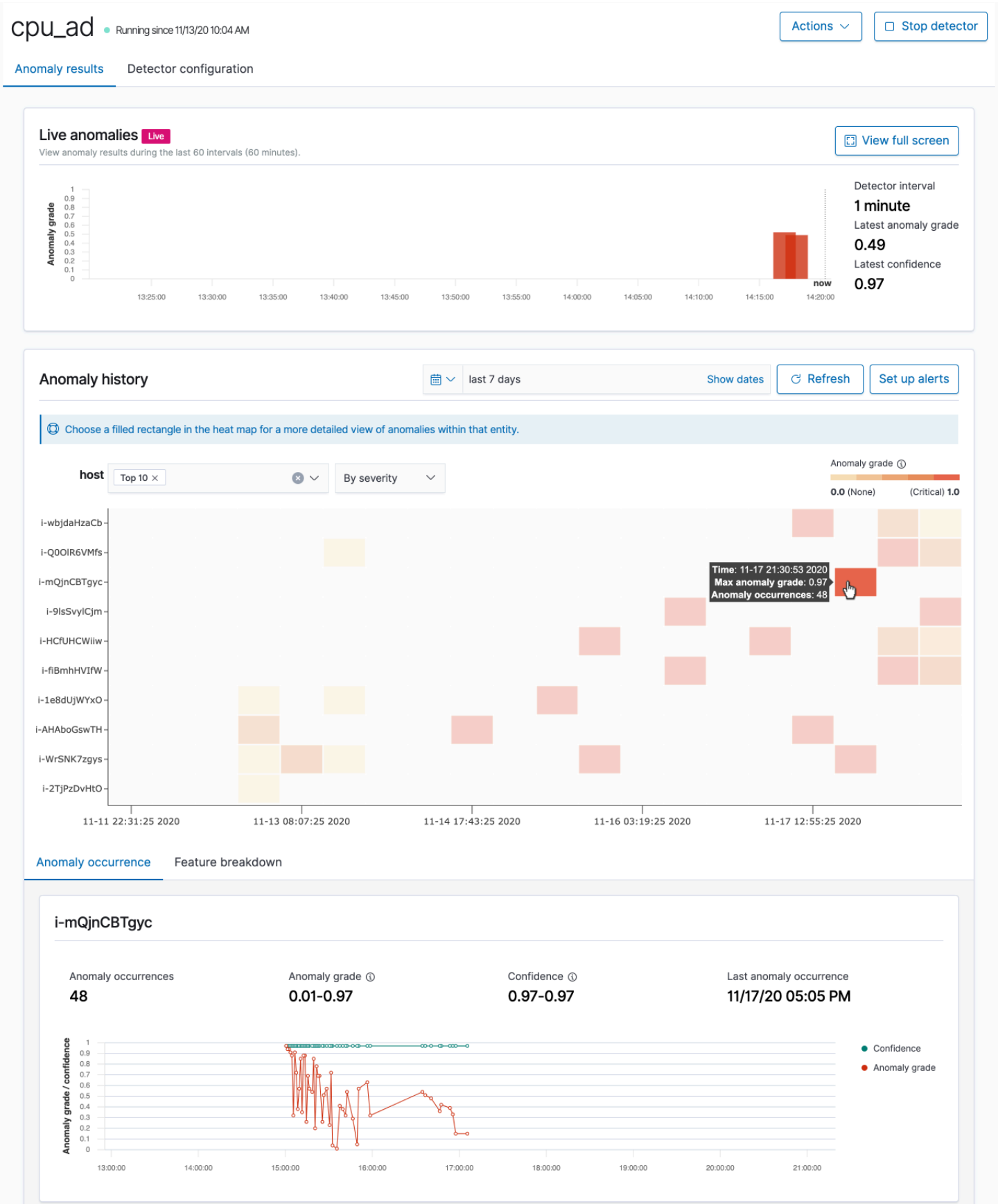
聚合方法决定了是什么构成异常。例如，如果选择 min()，则检测器将着重根据特征的最小值查找异常。如果选择 average()，则检测器将根据特征的平均值查找异常。每个检测器最多可以添加五个特征。

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
13:25:00
13:30:00
13:35:00
13:40:00
13:45:00
13:50:00
13:55:00
14:00:00
14:05:00
14:10:00
14:15:00
14:20:00
now
Detector interval
1 minute
Latest anomaly grade
0.49
Latest confidence
0.97
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-9IsSvylCjm
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
13:00:00
14:00:00
15:00:00
16:00:00
17:00:00
18:00:00
19:00:00
20:00:00
21:00:00
Confidence
Anomaly grade
Anomaly occurrences (48)
1 2 3 4 5
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 服务中创建异常检测器来检测 CPU 使用率过高。您将使用 OpenSearch 仪表板配置检测器来监控 CPU 使用情况，并在您的 CPU 使用率超过指定阈值时生成警报。

> **Note**
>
> 这些步骤适用于的最新版本，可能 OpenSearch 与过去的版本略有不同。

先决条件

- 您必须拥有运行 Elasticsearch 7.4 或更高版本或任何 OpenSearch版本的 OpenSearch 服务域。
- 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 控制面板” 中打开左侧面板菜单，选择 “异常检测”，然后选择 “创建检测器”。
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 “提示”

- the section called “异常检测”
- 异常检测 API

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

Amazon OpenSearch 服务 ML 连接器适用于 AWS 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 AWS 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。 AWS 服务 你可以设置一个连接器来包含 Amazon SageMaker 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

要创建连接器，您必须拥有一个 Amazon SageMaker 域终端节点和一个授予 OpenSearch 服务访问权限的 IAM 角色。

设置 Amazon SageMaker 域名

要部署您的机器学习模型，请参阅亚马逊 SageMaker 开发者指南 SageMaker中的在 Amazon 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 AWS 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 S OpenSearch ervice 访问您的 SageMaker 终端节点。要将策略附加到角色，请参阅添加 IAM 身份权限。

```
{
    "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”：

```
{
    "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 AWS 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：

```
{
  "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。此请求采用以下形式：

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
```

```
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
         "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
```

如果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 AWS SDK for Python (Boto3)、`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 AWS 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 AWS 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 AWS 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。

```
{
    "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”：

```
{
    "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 AWS 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：

```
{
  "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 AWS Secrets Manager

要将您的授权凭证存储在 Secrets Manager 中，请参阅 AWS Secrets Manager 用户指南中的创建 AWS 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。此请求采用以下形式：

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
    "name": "Cohere Connector: embedding",
```

```
    "description": "The connector to cohere embedding model",
    "version": 1,
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}
```

此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 AWS SDK for Python (Boto3)、`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 AWS 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 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AWS CloudFormation模板。 CloudFormation 是一 AWS 服务 款允许您通过将基础设施视为代码来建模、配置 AWS 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

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 AWS 账户权限

您 AWS 账户 必须拥有访问 CloudFormation 和 Lambda 的权限，以及 AWS 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 模板

Amazon SageMaker CloudFormation 模板定义了多种 AWS 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 插件为用于机器学习连接器的远程模型提供服务。

接下来，使用通过 Amazon Sagemaker 与稀疏编码器集成模板创建一个 Lambda 函数，让您的域设置远程推理连接器。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。模板会将您网域中的模型 ID 返回给您，这样您就可以开始搜索了。

使用 Amazon SageMaker CloudFormation 模板

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
2. 在左侧导航窗格中，选择集成。
3. 在每个 Amazon SageMaker 模板下，选择配置域名、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。

亚马逊 Bedrock 模板

与亚马逊 SageMaker 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 AWS 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，该模板创建 Lambda 函数，该函数让域使用 ML Commons 插件创建连接器。 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 Amazon Bedrock 的预先确定的终端节点，并跳过了终端节点配置步骤。

使用 Amazon Bedrock 模板 CloudFormation

1. 打开亚马逊 OpenSearch 服务控制台，网址为 https://console.aws.amazon.com/aos/home。
2. 在左侧导航窗格中，选择集成。
3. 在 “通过 Amazon Bedrock 与 Amazon Titan 文本嵌入集成” 模型下，选择 “配置域”、“配置公共域”。
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`

有关 ML Commons 设置的更多信息，请参阅 ML Commons 集群设置。

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，您可以使用流程框架模板来自动执行机器学习设置任务。Amazon 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。要查看自定义流程框架模板的示例，请参阅 fl ow-f ramework。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。

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 AWS 服务或第三方平台。有关这方面的更多信息，请参阅为第三方 ML 平台创建连接器。Amazon S OpenSearch ervice Flow framework 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：

- 创建一个 Amazon SageMaker 域名。
- 创建一个 IAM 角色。
- 配置通行证角色权限。
- 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 AWS 服务的 Amazon OpenSearch 服务 ML 连接器。 AWS 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。

通过流程框架服务创建连接器

要使用连接器创建流程框架模板，您需要向您的 OpenSearch 服务域终端节点发送POST请求。你可以使用 curl、示例 Python 客户端、Postman 或其他方法来发送已签名的请求。该POST请求采用以下格式：

```
POST /_plugins/_flow_framework/workflow
{
  "name": "Deploy Claude Model",
  "description": "Deploy a model using a connector to Claude",
  "use_case": "PROVISION",
  "version": {
    "template": "1.0.0",
    "compatibility": [
      "2.12.0",
      "3.0.0"
    ]
  },
  "workflows": {
    "provision": {
      "nodes": [
        {
          "id": "create_claude_connector",
          "type": "create_connector",
          "user_inputs": {
            "name": "Claude Instant Runtime Connector",
            "version": "1",
            "protocol": "aws_sigv4",
            "description": "The connector to BedRock service for Claude model",
            "actions": [
              {
                "headers": {
                  "x-amz-content-sha256": "required",
                  "content-type": "application/json"
                },
                "method": "POST",
                "request_body": "{ \"prompt\":\"${parameters.prompt}\",
 \"max_tokens_to_sample\":${parameters.max_tokens_to_sample},
```

```
 \"temperature\":${parameters.temperature},  \"anthropic_version\":
\"${parameters.anthropic_version}\" }",
                "action_type": "predict",
                "url": "https://bedrock-runtime.us-west-2.amazonaws.com/model/
anthropic.claude-instant-v1/invoke"
              }
            ],
            "credential": {
                "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-
role"
             },
            "parameters": {
              "endpoint": "bedrock-runtime.us-west-2.amazonaws.com",
              "content_type": "application/json",
              "auth": "Sig_V4",
              "max_tokens_to_sample": "8000",
              "service_name": "bedrock",
              "temperature": "0.0001",
              "response_filter": "$.completion",
              "region": "us-west-2",
              "anthropic_version": "bedrock-2023-05-31"
            }
          }
        }
      ]
    }
  }
}
```

如果您的域位于虚拟私有云（Amazon VPC）中，则必须连接到 Amazon VPC，请求才能成功创建 AI 连接器。访问 Amazon VPC 因网络配置而异，但通常需要连接到 VPN 或公司网络。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请求更易于自动化，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AWS SDK (Boto3)、适用于人类的 reque sts: HTTP 和 request s-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)
```

预定义的工作流程模板

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许多默认值。您可以在调用 “创建工作流 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程模板，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

此表概述了不同的可用模板、模板的描述以及所需的参数。

模板用例	描述	必需参数
bedrock_titan_embedding_model_deploy	创建和部署 Amazon Bedrock 嵌入模型（默认情况下，titan-embed-text-v1	create_connector.credential.roleArn
bedrock_titan_embedding_model_deploy	创建和部署 Amazon Bedrock 多模式嵌入模型（默认情况下，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_wi	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 ,

模板用例	描述	必需参数
th_cohere_embedding		create_connector.credential.secretArn
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，该处理器为神经查询设置默认模型 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，并使用用于多模态搜索的 text_image_embedding 处理器和 k-nn 索引配置采集管道。您必须提供您的 AWS 凭证。	create_connector.credential.roleArn

> **Note**
>
> 对于所有需要秘密 ARN 的模板，默认设置是将密钥名称为 “key” 的密钥存储在 Secr AWS ets mangager 中。

带有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。

模板用例	描述
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，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，则权限已经到位。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. 为角色命名 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. 创建角色后，将其映射到任何将管理流程框架索引的用户或后端角色。

Amazon OpenSearch 服务的安全分析

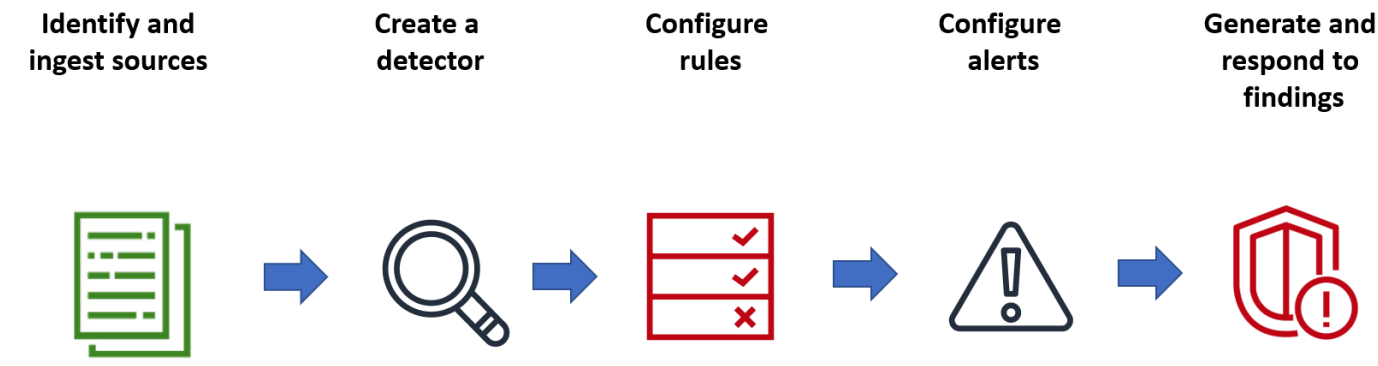
Security Analytics 是一种 OpenSearch 解决方案，可提供对组织基础设施的可见性，监控异常活动，实时检测潜在的安全威胁，并向预先配置的目的地触发警报。您可以通过持续评估安全规则及查看自动生成的安全调查发现来监控安全事件日志中的恶意活动。此外，安全分析还可以生成自动警报，并将其发送到指定通知通道，例如 Slack 或电子邮件。

您可以使用 Security Analytics 插件来检测常见威胁， out-of-the-box 并根据现有的安全事件日志（例如防火墙日志、Windows 日志和身份验证审核日志）生成重要的安全见解。要使用安全分析，您的域名必须运行 OpenSearch 版本 2.5 或更高版本。

> **Note**
>
> 本文档简要概述了 Amazon OpenSearch 服务的安全分析。它定义了关键概念并提供了配置权限的步骤。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 文档中的安全分析。

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，并为每种类型提供 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 API 来创建和使用规则。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。有关调查结果的更多信息，请参阅 OpenSearch 文档中的使用调查结果。

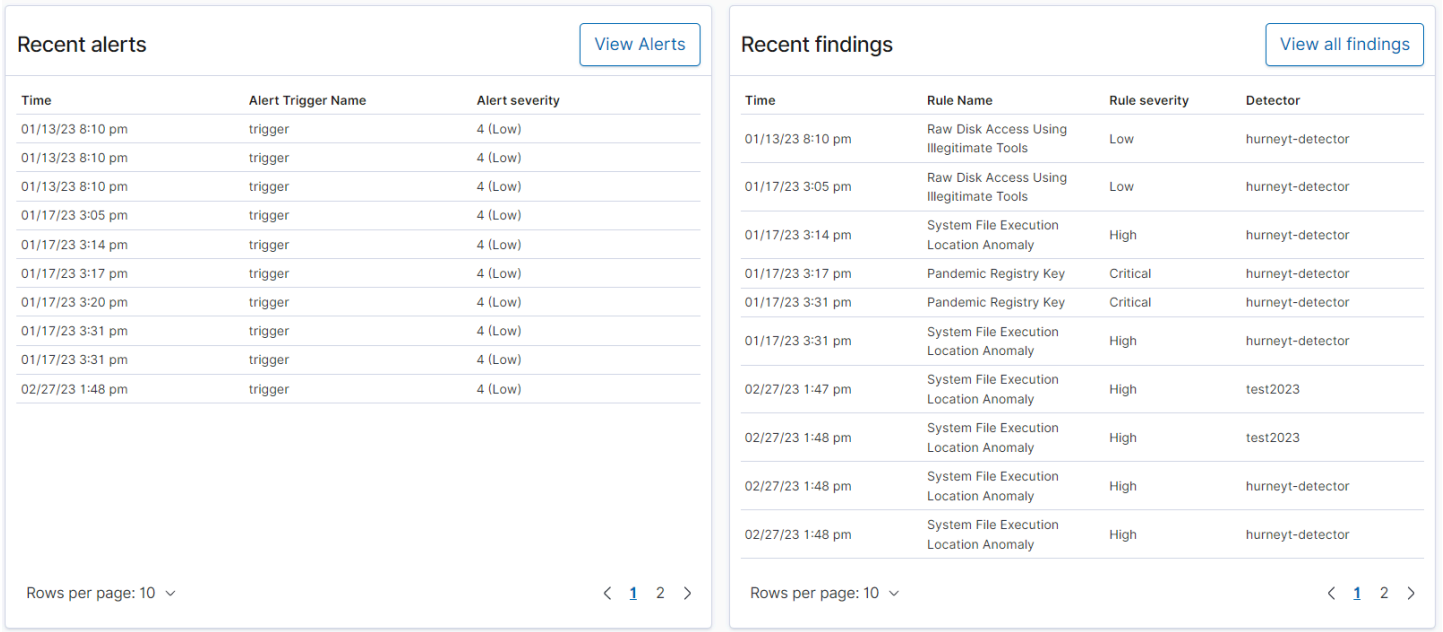
提醒

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。有关警报的更多信息，请参阅 OpenSearch 文档中的处理警报。

探索安全分析

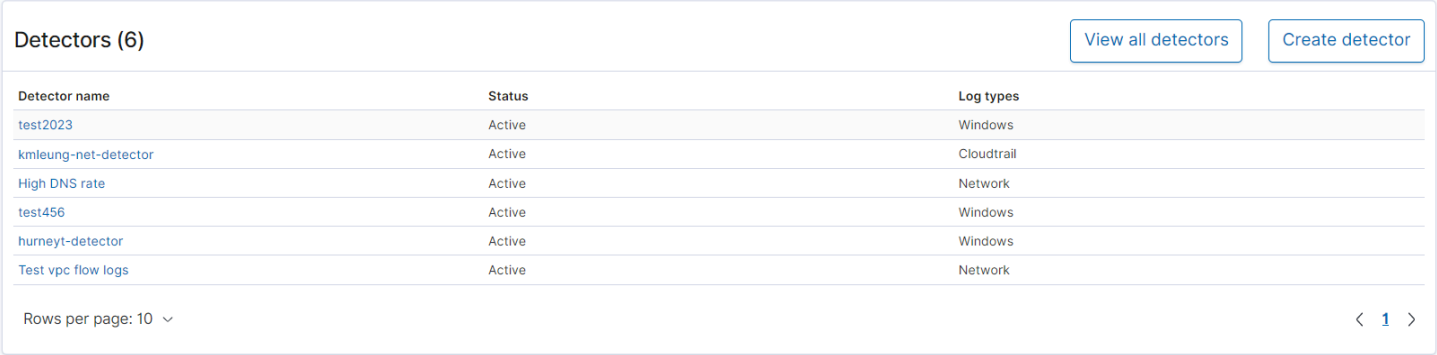
您可以使用 OpenSearch 仪表板对安全分析插件进行可视化和深入了解。概述视图提供诸如发现结果和警报计数、最近的发现和警报、频繁检测规则以及探测器列表等信息。您可以查看由多个可视化效果图构成的摘要视图。例如，下图显示了给定时段内不同日志类型的调查发现和提醒趋势。

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

如果您在先前存在的 OpenSearch 服务域上启用安全分析，则可能无法在该域上定义该`security_analytics_manager`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。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 OpenSearch 服务中的可观察性

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，以便浏览、发现和查询存储在中的数据。OpenSearch该插件需要 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。数据收集和监控集中在一处，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 OpenSearch 服务中的可观察性。有关可观察性插件的全面文档（包括权限），请参阅可观察性。

每个人探索数据的过程都是不同的。如果您不熟悉探索数据和创建可视化效果，我们建议您尝试以下工作流程。

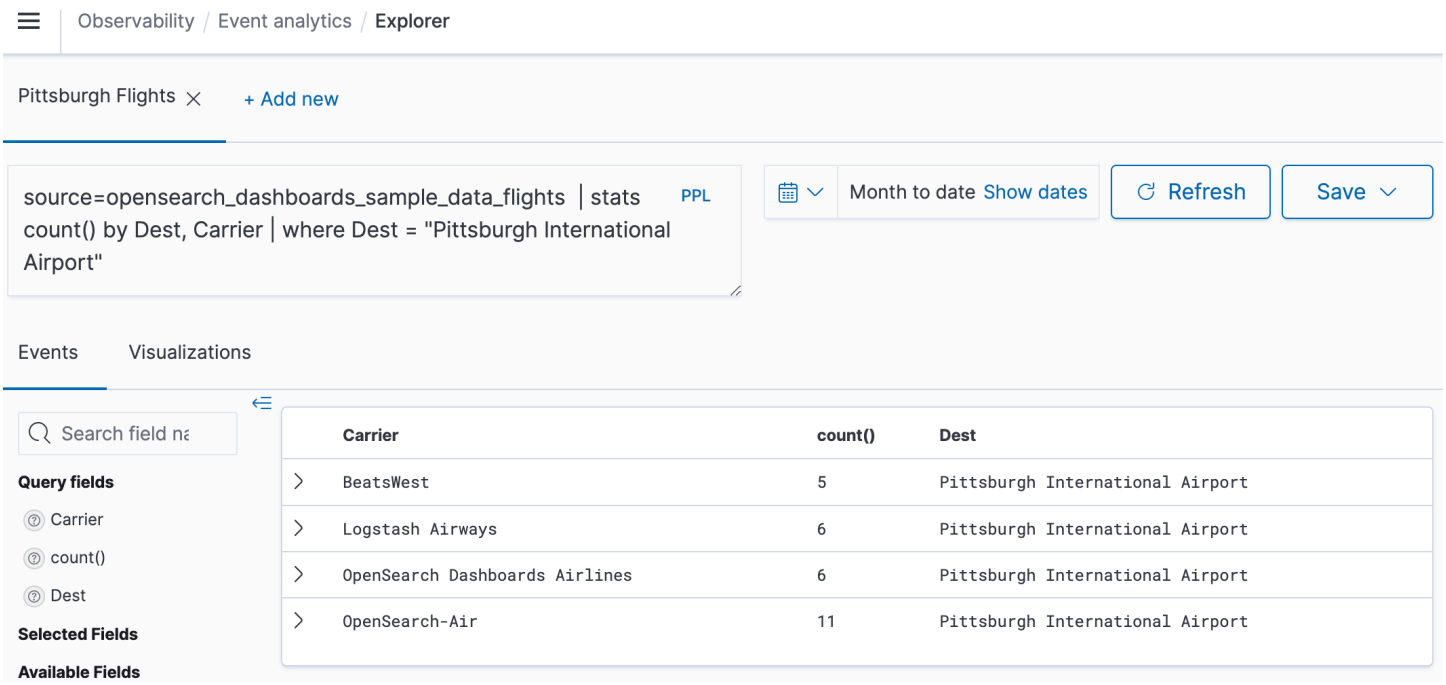
使用事件分析来探索数据

首先，假设您正在 OpenSearch 服务域中收集航班数据，并且您想了解上个月抵达匹兹堡国际机场的航班最多的航空公司。您会编写下面的 PPL 查询：

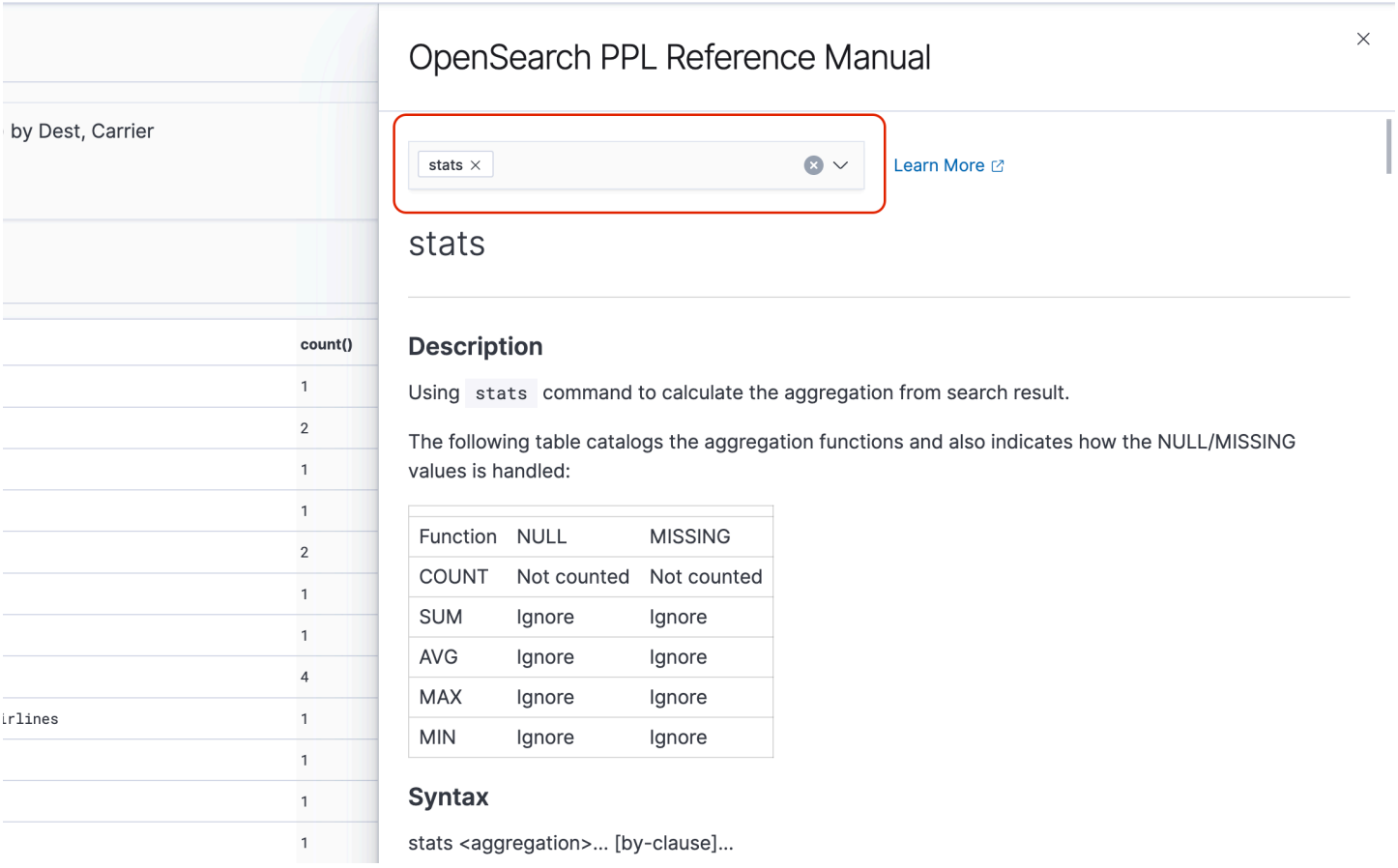
```
source=opensearch_dashboards_sample_data_flights |
    stats count() by Dest, Carrier |
    where Dest = "Pittsburgh International Airport"
```

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：



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：



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的平均延误较少。

	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）。

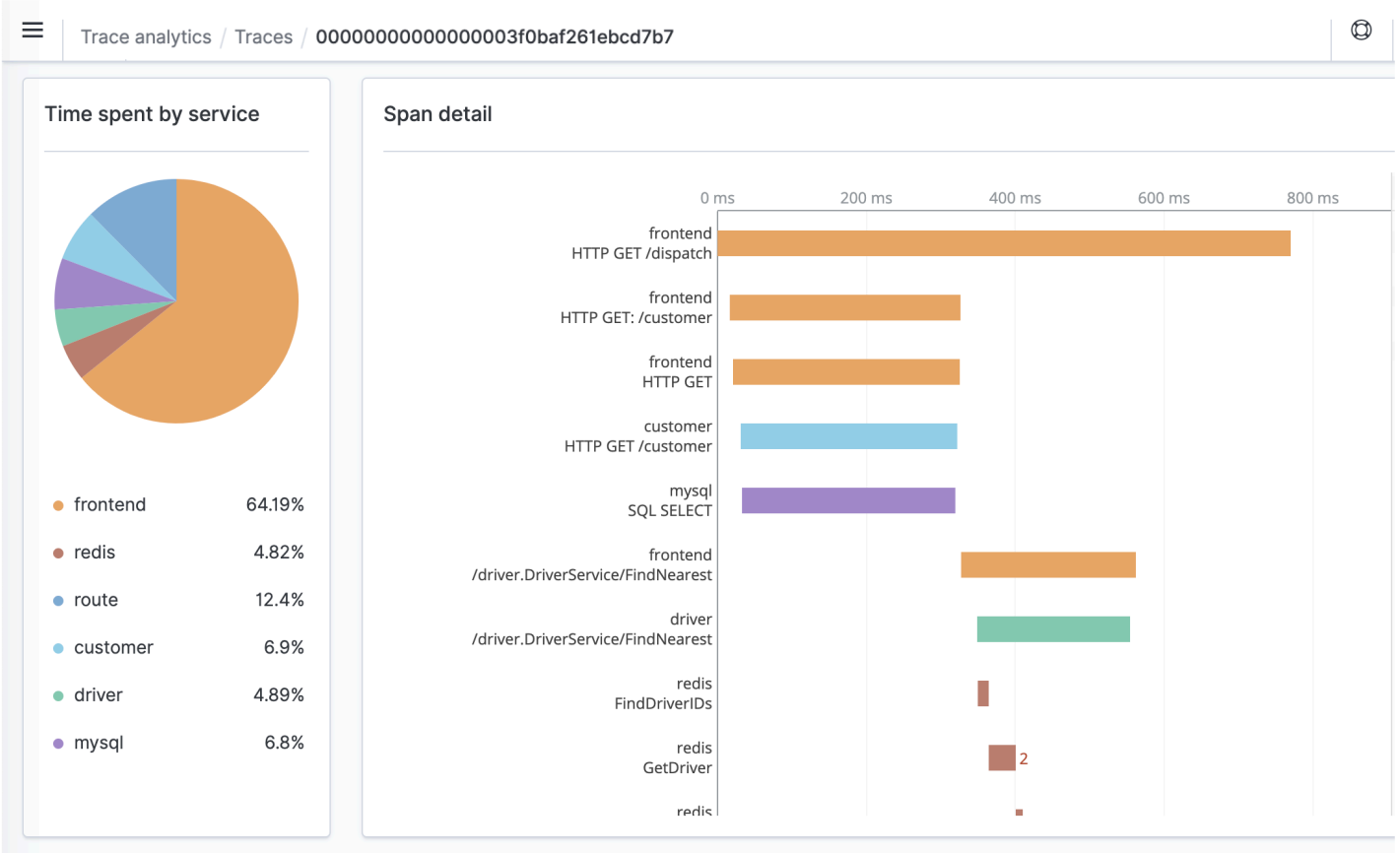
创建可视化

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

Trace Analytics 提供了一种可视化 OpenSearch 数据中事件流的方法，以识别和修复分布式应用程序中的性能问题。



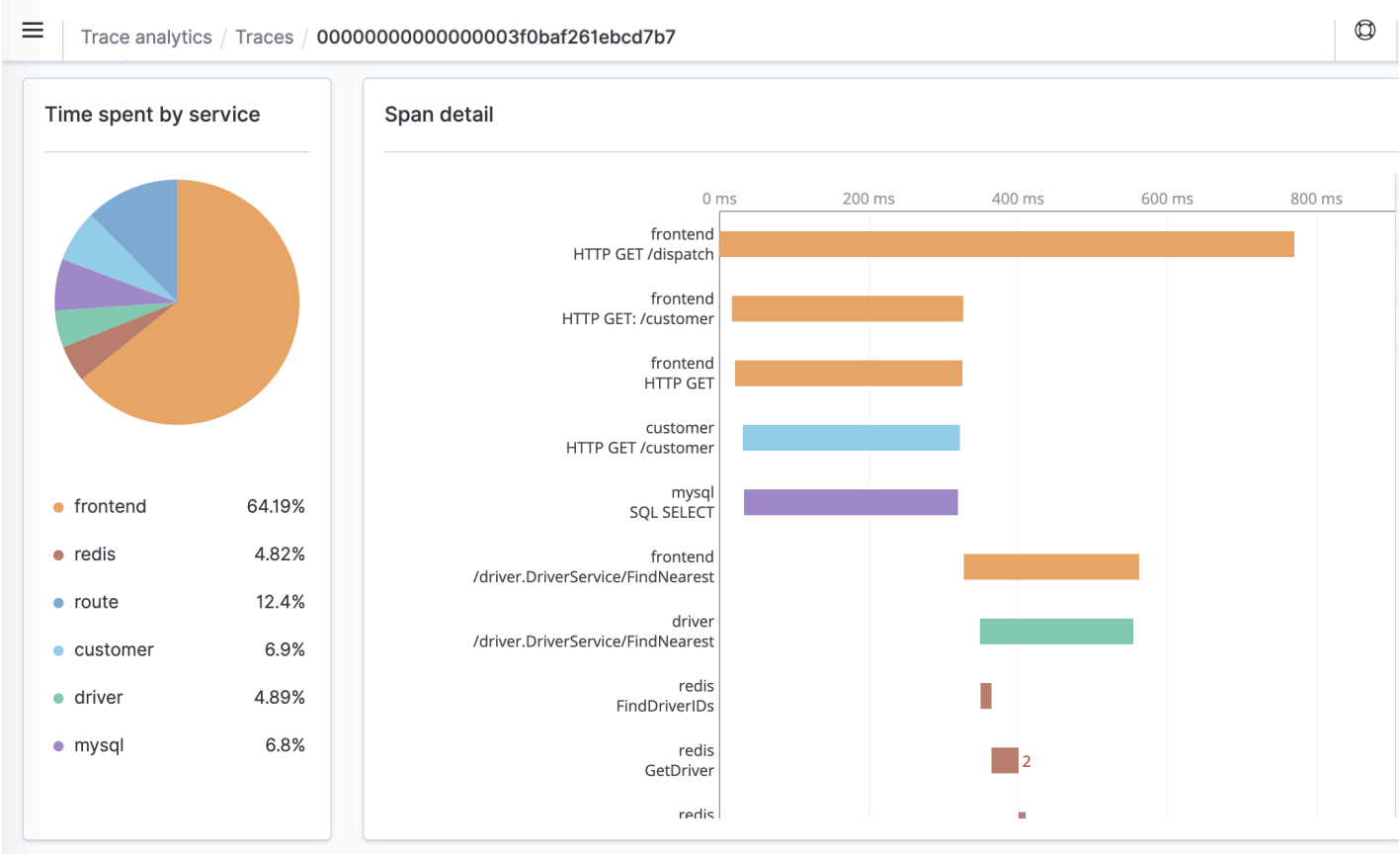
Amazon OpenSearch 服务的追踪分析

您可以使用 Trace Analytics（ OpenSearch 可观察性插件的一部分）来分析来自分布式应用程序的跟踪数据。追踪分析需要 OpenSearch 或 Elasticsearch 7.9 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。如需全面的文档，请参阅开源 OpenSearch 文档中的追踪分析。



先决条件

Trace Analytics 要求您向应用程序添加工具，并使用 OpenTelemetry支持的库（例如 Jaeger 或 Zipkin）生成跟踪数据。此步骤完全在 OpenSearch 服务之外进行。 OpenTelemetry 文档AWS 发行版包含许多编程语言的示例应用程序，可以帮助您入门，包括 Java、Python、Go 和 JavaScript。

向应用程序添加工具后，OpenTelemetry收集器会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。 AWS 的发行版 OpenTelemetry 包括一个接收器. AWS 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 收集器与配合使用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 摄取示例配置

要将跟踪数据发送到 OpenSearch 服务域，请尝试以下 OpenSearch 采集管道配置示例。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:
```

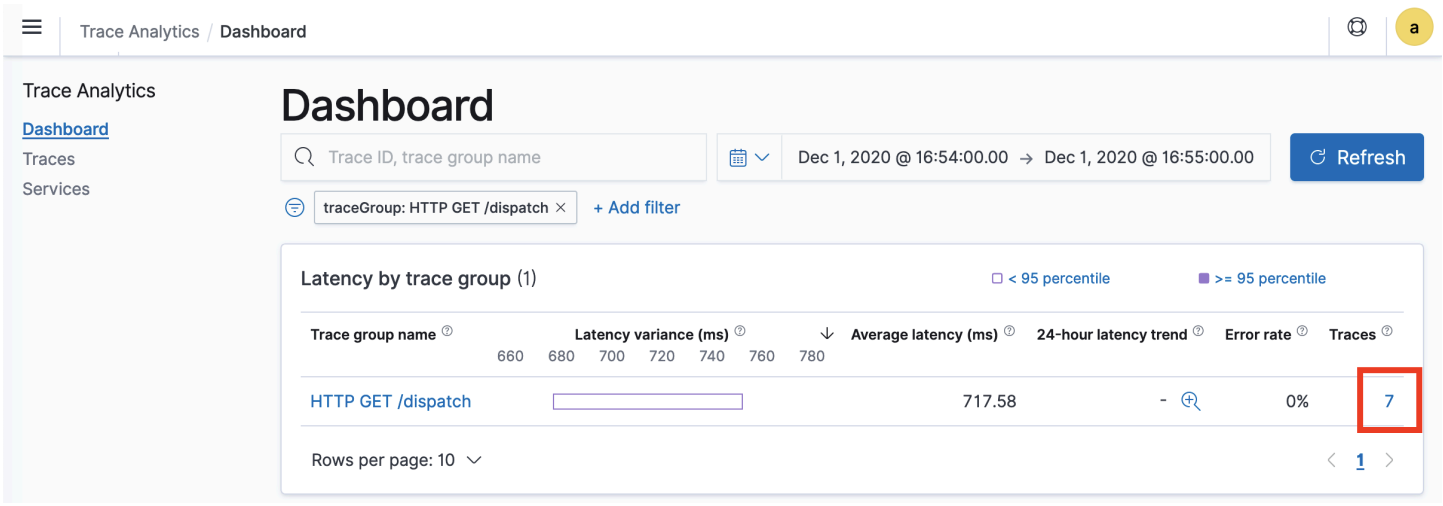
```
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - otel_traces:
  sink:
    - opensearch:
        hosts: ["https://domain-endpoint"]
        index_type: trace-analytics-raw
        aws:
          # IAM role that OpenSearch Ingestion assumes to access the domain sink
          sts_role_arn: "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对接收器的写入权限。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”。

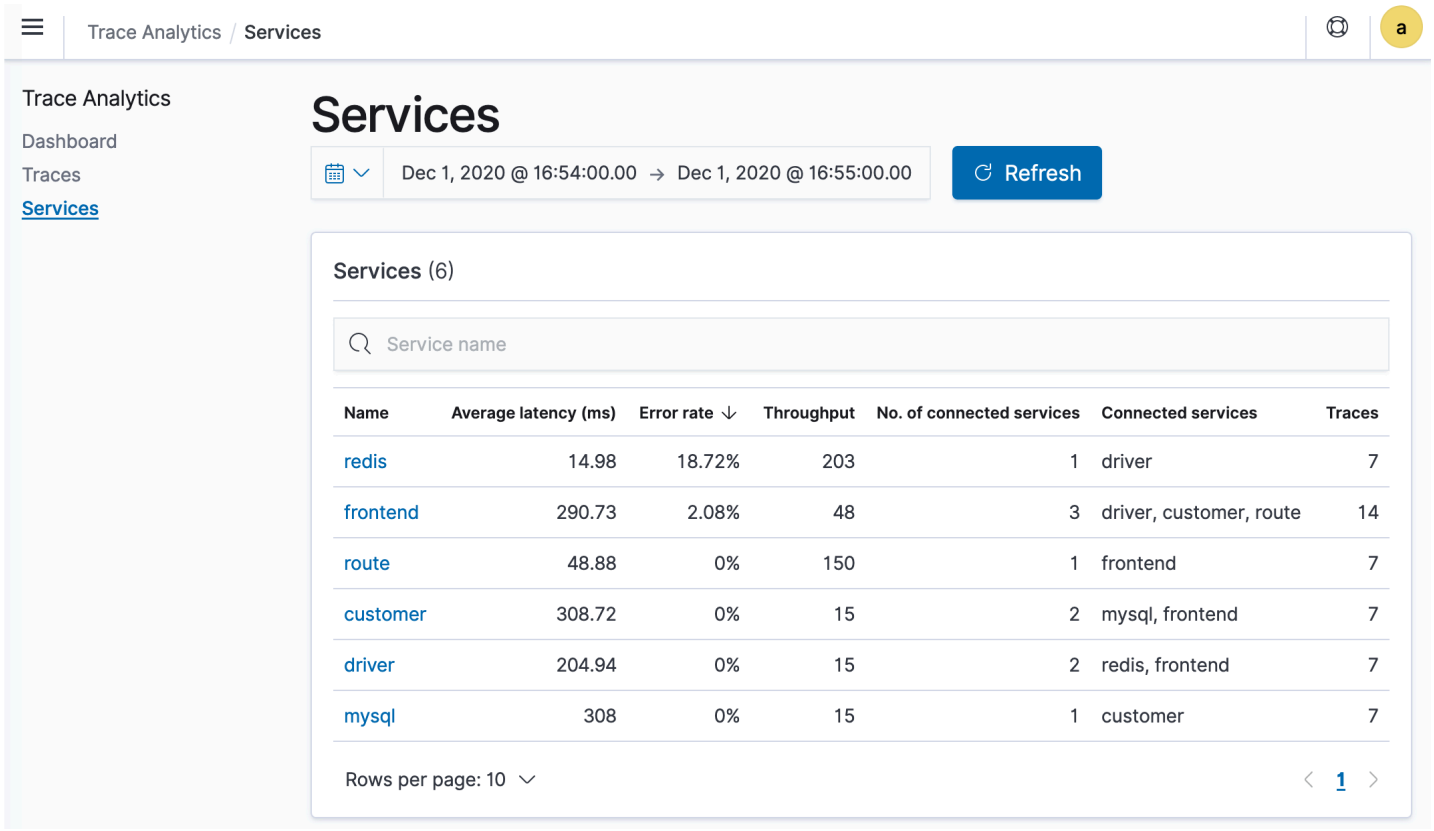
探索跟踪数据

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亚马逊 OpenSearch 服务数据

管道处理语言 (PPL) 是一种查询语言，允许您使用 pipe (|) 语法来查询存储在亚马逊 OpenSearch 服务中的数据。PPL 需要 Elasticsearch 7.9 OpenSearch 或更高版本。

> **Note**
>
> 本文档简要概述了 Amazon OpenSearch 服务的 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

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 服务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 向包含属于该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。如果索引包含多个分片，请尝试将分片数量设为数据节点数量的偶数倍。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工作负载模式而异，每个节点的分片数上限为 1000 个。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。如果您的实例类型具有 8 个 vCPU，则数据节点数量的设置应确保每个节点的分片数量不超过 6 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`。如果您未在特定时间范围（通常为两周）内更新域名，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 AWS 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 S OpenSearch ervice 将副本分片分发到与其对应的主分片不同的可用区。可用区之间的集群通信不会产生跨可用区数据传输费用。

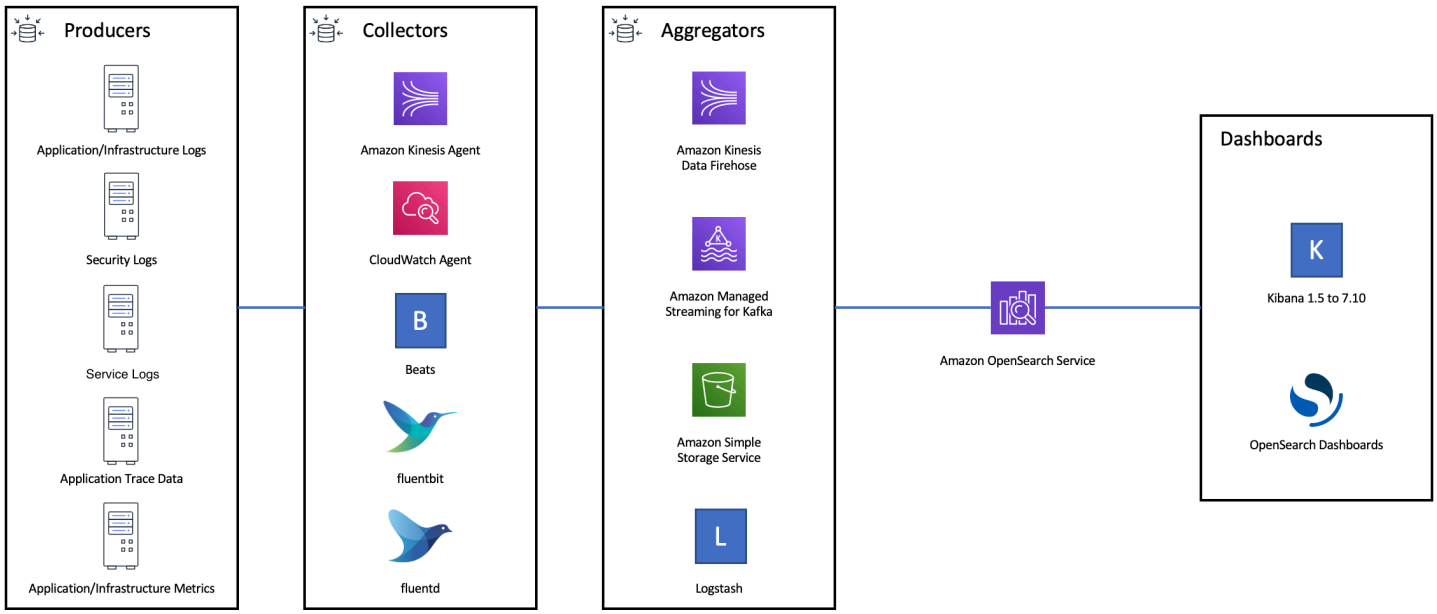
可用区是每个 区域内的隔离位置。使用双可用区配置时，失去一个可用区意味着您将损失一半的域容量。使用三个可用区可进一步减少失去单个可用区的影响。

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 会自动将该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。

Performance

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 AWS 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 AWS Key Management Service (AWS KMS) 来存储和管理您的加密密钥，使用带有 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

N ode-to-node 加密在 S OpenSearch ervice 中的默认安全功能之上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)。 OpenSearchN ode-to-node 加密，任何通过 HTTPS 发送到您的 OpenSearch 服务域的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 AWS 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AWS 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AWS 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的存储空间来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，对索引的查询的执行与对热 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

在确定好性能和计算消耗基线后考虑购买预留实例（RI）。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%。

观察到至少稳定运行 14 天后，在 Cost Explorer 中查看预留实例建议。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，首先要了解您的存储需求、服务及其 OpenSearch 本身，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：每个副本都是一个索引的完整复制，需要同等量的磁盘空间。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 服务会为每个实例预留 20% 的存储空间（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 及更高版本以及的所有版本的 OpenSearch每个节点上限为 1,000 个分片。要调整每个节点的最大分区数，请配置 `cluster.max_shards_per_node` 设置。有关示例，请参阅集群设置。

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 服务中的 PB 级规模 OpenSearch

Amazon OpenSearch 服务域提供高达 3 PB 的附加存储空间。您可以配置包含 200 个 `i3.16xlarge.search` 实例类型的域，每个实例类型都有 15 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。

虽然本节中频繁引用多个 i3.16xlarge.search 实例类型，您可以使用其他实例类型，以达到 1PB 的总域存储。

创建域

这种大小的域超过了每个域 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 200 个实例，请在 AWS 支持中心上打开一个案例。

定价

在创建如此大小的域名之前，请查看亚马逊 OpenSearch 服务定价页面，确保相关费用符合您的预期。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

i3 实例类型设计用于提供快速、本地的非易失性存储规范 (NVMe) 存储空间。由于与 Amazon Elastic Block Store 相比，这种本地存储往往具有性能优势，因此当您在 OpenSearch 服务中选择这些实例类型时，EBS 卷不是一个选项。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

通常的 OpenSearch 指导方针是每个分片的容量不得超过 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。

控制分片的数量很棘手。 OpenSearch 用户通常每天轮换索引，并将数据保留一两个星期。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。

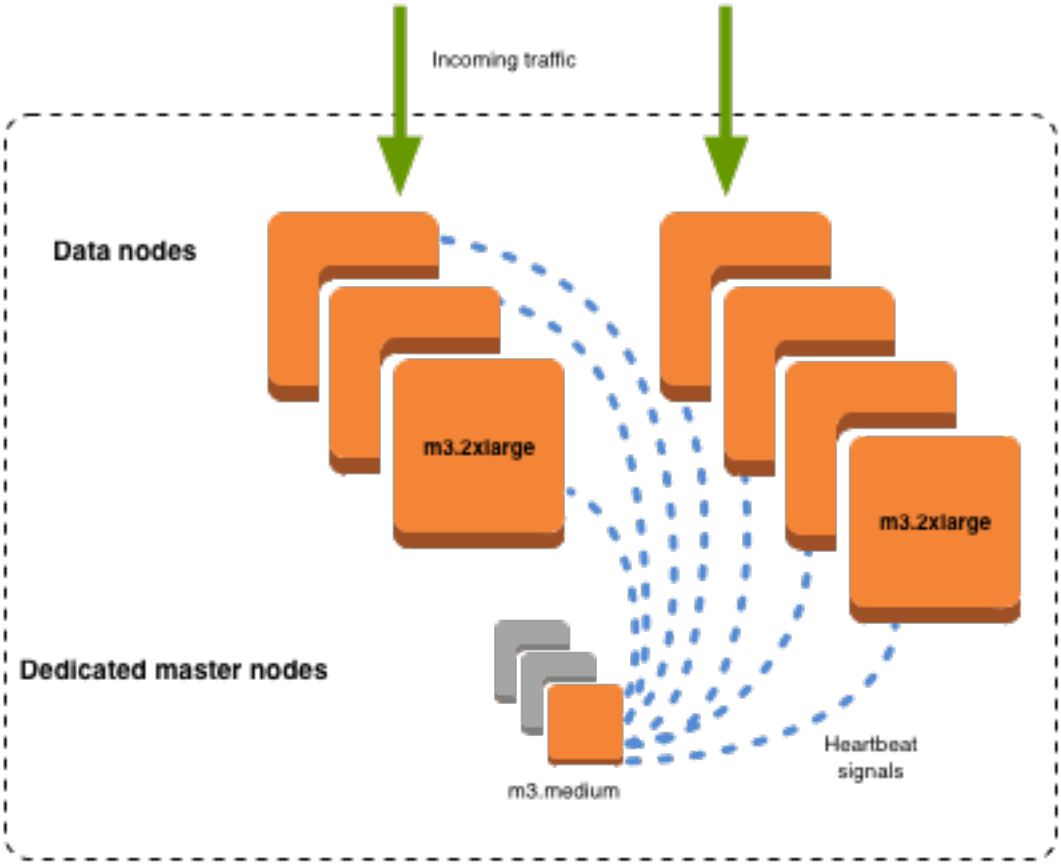
Amazon OpenSearch 服务中的专用主节点

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，两个可用区具有一个专用主节点，一个可用区有两个。如果该 AZ 发生中断，其余两个可用区没有选择新主节点所需的仲裁节数 (3)。
- 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

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。监控专用主节点指标以查看您是否需要使用更大的实例类型。

实例计数	主节点 RAM 大小	支持的最大分片数	推荐的最小专用主实例类型
1–10	8 GiB	10K	`m5.large.search` 或 `m6g.large.search`
11–30	16 GiB	30K	`c5.2xlarge.search` 或 `c6g.2xlarge.search`

实例计数	主节点 RAM 大小	支持的最大分片数	推荐的最小专用主实例类型
31–75	32 GiB	40K	`r5.xlarge.search` 或 `r6g.xlarge.search`
76 — 125	64 GiB	75K	`r5.2xlarge.search` 或 `r6g.2xlarge.search`
126 – 200	128 GiB	75K	`r5.4xlarge.search` 或 `r6g.4xlarge.search`

- 有关特定配置更改如何影响专用主节点的信息，请参阅the section called “配置更改”。
- 有关实例数量限制的说明，请参阅OpenSearch 服务域和实例配额。
- 有关特定实例类型（包括 vCPU、内存和定价）的更多信息，请参阅 A mazon OpenSearch 服务价格。

亚马逊 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 AWS `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 AWS 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	问题
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。
ClusterIndexWritesBlocked 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	问题
值 >= 80% 达到 15 分钟，连续 3 次	
JVMMemory Pressure 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。
OldGenJVM MemoryPressure 最大值 >= 80% 达到 1 分钟，连续 3 次	
MasterCPU Utilization 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。
MasterJVM MemoryPressure 最大值 >= 95% 达到 1 分钟，连续 3 次	
MasterOld GenJVMMem oryPressure 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 AWS 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	问题
KMSKeyInaccessible >= 1 达到 1 分钟，1 次连续时间	用于 AWS 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 最大值为 < 1，持续 5 分钟，连续 1 次	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 AWS 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	问题
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	问题
WarmFreeStorageSpace 是 >= 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	问题
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。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。
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	问题
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
OR1	or1.medium.search or1.large.search or1.xlarge.search	• OR1 实例类型需要 OpenSearch 2.11 或更高版本。 • OR1 实例仅与其他 Graviton 实例类型主节点（C6g、M6g、R6g）兼容。

实例类型	实例	限制
	or1.2xlarge.search or1.4xlarge.search or1.8xlarge.search or1.12xlarge.search or1.16xlarge.search	

实例类型	实例	限制
Im4gn	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。 • Im4gn 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	I3 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 存储卷。
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 • 仅当您的域配置为非备用实例时，您才能使用 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”。 • 只有当您的域的实例数等于 10 或更少时，您才能使用 T3 实例类型。 • T3 实例类型不支持 UltraWarm存储、冷存储或自动调整。

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。我们鼓励您使用当前一代实例类型以获得最佳性能，但我们将继续支持下面的上一代实例类型。

实例类型	实例	限制
C4	c4.large.search c4.xlarge.search c4.2xlarge.search c4.4xlarge.search c4.8xlarge.search	
I2	i2.xlarge.search i2.2xlarge.search	
M3	m3.medium.search m3.large.search m3.xlarge.search m3.2xlarge.search	• M3 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
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 身份验证 OpenSearch		

功能	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
就地升级		
Curator 支持	不包括	5.1
每小时自动快照	1.0	5.3
N ode-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		
提示	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		
远程重建索引		

功能	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
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
搜索管道	2.9	不包括
机器学习连接器	2.9	不包括
多模态语义搜索	2.1.1	不包括
直接查询 Amazon S3 的数据来源	2.1.1	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 S OpenSearch ervice 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
ICU 分析	1.0	包含在所有域中
日语 (kuromoji) 分析		
拼音分析	1.0	2.3
Seunjeon 韩语分析	1.0	5.1
智能中文分析		
Stempel Polish 分析		
Ingest Attachment 处理器		
Ingest 用户代理处理器		

插件	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
映射器 Murmur3		
映射器大小	1.0	5.3
乌克兰语分析		
OpenSearch 警报	1.0	6.2
OpenSearch SQL	1.0	6.5
OpenSearch 安全	1.0	6.7
OpenSearch 索引状态管理	1.0	6.8
OpenSearch k-NN	1.0	7.1
OpenSearch 异常检测	1.0	7.4
IK（中文）分析	1.0	7.7
越南语分析		
泰语分析		
学习排名		
OpenSearch 异步搜索	1.0	7.10

插件	所需的最低 OpenSearch 版本	最低要求的 Elasticsearch 版本
OpenSearch 跨集群复制	1.1	7.10
OpenSearch 可观察性	1.2	不支持
Nori 分析	1.3	不支持
拼音分析	1.3	不支持
ST 转换	1.3	不支持
Sudachi 分析	1.3	不支持
ML Commons	1.3	不支持
OpenSearch 通知	2.3	不支持
安全分析	2.5	不支持
神经搜索	2.9	不支持
亚马逊 Personalize 搜索排名	2.9	不支持
希伯来语分析	2.1.1	不支持
HanLP	2.1.1	不支持

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 AWS Management Console 和将插件 AWS 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。或者，您可以使用更新的数据创建新包，使用此新包创建新索引，将现有索引重新编入新索引，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 “自定义程序包”。

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
- OpenSearch 版本 2.13
- OpenSearch 版本 2.11
- OpenSearch 版本 2.9
- OpenSearch 版本 2.7
- OpenSearch 版本 2.5
- OpenSearch 版本 2.3
- OpenSearch 版本 1.3
- OpenSearch 版本 1.2
- OpenSearch 版本 1.1
- OpenSearch 版本 1.0
- Elasticsearch 7.10 版
- Elasticsearch 7.9 版

- Elasticsearch 7.8 版
- Elasticsearch 7.7 版
- Elasticsearch 7.4 版
- Elasticsearch 7.1 版
- Elasticsearch 6.8 版
- Elasticsearch 6.7 版
- Elasticsearch 6.5 版
- Elasticsearch 6.4 版
- Elasticsearch 6.3 版
- Elasticsearch 6.2 版
- Elasticsearch 6.0 版
- Elasticsearch 5.6 版
- Elasticsearch 5.5 版
- Elasticsearch 5.3 版
- Elasticsearch 5.1 版
- Elasticsearch 2.3 版
- Elasticsearch 1.5 版

值得注意的 API 差异

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 服务的某些设置和统计信息 API 的响应，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
  }
}
```

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 OpenSearchREST 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/_security_analytics
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - cluster.max_shards_per_node
  - cluster.search.request.slowlog.level
  - cluster.search.request.slowlog.threshold.warn
  - cluster.search.request.slowlog.threshold.info
  - cluster.search.request.slowlog.threshold.debug
  - cluster.search.request.slowlog.threshold.trace
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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/_security_analytics
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. 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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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/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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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/_ml
- _plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

1. 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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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/_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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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
- /_plugins/_transforms
- /_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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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”。

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 OpenSearchREST 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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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 7.10 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]：
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_index_template
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_opendistro/_asynchronous_search
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_plugins/_replication
- /_rank_eval
- /_validate

1. 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）
- /_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
- /_index_template [6]
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count

1. 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”。此列表仅提及 S OpenSearch ervice 支持的一般 OpenSearch 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 7.8 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
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile

- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 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
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

1. 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
- /_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
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node

- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval

1. 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

- /_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
  - cluster.max_shards_per_node
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 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
- /_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

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/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks

- /_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.blocks.read_only
- /_opendistro/_alerting
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_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 6.7 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/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
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 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]：
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_sql
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- /_percolate
- /_plugin/kibana
- /_rank_eval

1. 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
- /_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
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 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
- /_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 6.2 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]：
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- /_plugin/kibana
- /_rank_eval

1. 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
- /_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 5.6 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
- /_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 5.5 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]：
- /_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 5.3 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
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_render
- /_rollover
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [4]

- /_all
- /_analyze
- /_bulk
- /_cat（/_cat/nodeattrs 除外）
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- 适用于多个属性的 /_cluster/settings [3]：
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 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
- /_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
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [3]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. 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
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_render
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

- threadpool.percolate.queue_size
- threadpool.search.queue_size
- threadpool.suggest.queue_size

Elasticsearch 1.5 版

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-name* /_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
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_plugin/kibana3
- /_plugin/migration
- /_refresh
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

- `threadpool.suggest.queue_size`

亚马逊 OpenSearch 服务配额

您的 AWS 账户对每项 AWS 服务都有默认配额（以前称为限制）。除非另有说明，否则，每个配额都特定于区域。

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。AWS 一般参考

要在中查看 OpenSearch 服务配额 AWS Management Console，请打开 Serv ice Quotas 控制台。在导航窗格中，选择AWS 服务，然后选择亚马逊 OpenSearch服务。要请求提高限额，请参阅《服务限额用户指南》中的请求提高限额。

主题

- UltraWarm 存储配额
- EBS 卷大小配额
- 网络配额
- 分片大小配额
- Java 进程配额
- 域策略配额

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
`ultrawarm1.medium.search`	1.5 TiB
`ultrawarm1.large.search`	20 TiB

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。

- 如果您在创建域时选择 EBS 卷类型下的磁性存储，除 `t2.small` 和 `t2.medium` 之外的所有实例类型以及所有不支持磁性存储的引力实例（M6g、C6g、R6g 和 R6gd）的最大卷大小为 100GiB。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.search`	10 GiB	35 GiB	不适用
`t2.medium.search`	10 GiB	35 GiB	不适用
`t3.small.search`	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

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
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
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

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
c5.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c5.9xlarge.search	10 GiB	3.5 TiB	3.5 TiB
c5.18xlarge.search	10 GiB	7 TiB	7 TiB
c6g.large.search	10 GiB	256 GiB	256 GiB
c6g.xlarge.search	10 GiB	512 GiB	512 GiB
c6g.2xlarge.search	10 GiB	1 TiB	1 TiB
c6g.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c6g.8xlarge.search	10 GiB	3 TiB	3 TiB
c6g.12xlarge.search	10 GiB	4.5 TiB	4.5 TiB
r3.large.search	10 GiB	512 GiB	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

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
r5.large.search	10 GiB	1 TiB	2 TiB
r5.xlarge.search	10 GiB	1.5 TiB	3 TiB
r5.2xlarge.search	10 GiB	3 TiB	6 TiB
r5.4xlarge.search	10 GiB	6 TiB	12 TiB
r5.12xlarge.search	10 GiB	12 TiB	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

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
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
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
or1.large.search	20 GiB	不适用	1532 GiB
or1.xlarge.search	20 GiB	不适用	3 TiB
or1.2xlarge.search	20 GiB	不适用	6 TiB
or1.4xlarge.search	20 GiB	不适用	12 TiB
or1.8xlarge.search	20 GiB	不适用	16 TiB
or1.12xlarge.search	20 GiB	不适用	24TiB
or1.16xlarge.search	20 GiB	不适用	36 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

实例类型	最小 EBS 容量	最大 EBS 容量（gp2）	最大 EBS 容量（gp3）
im4gn.16xlarge.search	不适用	不适用	不适用

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.search	10 MiB
t2.medium.search	10 MiB
t3.small.search	10 MiB
t3.medium.search	10 MiB
m3.medium.search	10 MiB
m3.large.search	10 MiB
m3.xlarge.search	100 MiB
m3.2xlarge.search	100 MiB
m4.large.search	10 MiB
m4.xlarge.search	100 MiB
m4.2xlarge.search	100 MiB
m4.4xlarge.search	100 MiB
m4.10xlarge.search	100 MiB

实例类型	HTTP 请求负载的最大值
m5.large.search	10 MiB
m5.xlarge.search	100 MiB
m5.2xlarge.search	100 MiB
m5.4xlarge.search	100 MiB
m5.12xlarge.search	100 MiB
m6g.large.search	10 MiB
m6g.xlarge.search	100 MiB
m6g.2xlarge.search	100 MiB
m6g.4xlarge.search	100 MiB
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

实例类型	HTTP 请求负载的最大值
c5.xlarge.search	100 MiB
c5.2xlarge.search	100 MiB
c5.4xlarge.search	100 MiB
c5.9xlarge.search	100 MiB
c5.18xlarge.search	100 MiB
c6g.large.search	10 MiB
c6g.xlarge.search	100 MiB
c6g.2xlarge.search	100 MiB
c6g.4xlarge.search	100 MiB
c6g.8xlarge.search	100 MiB
c6g.12xlarge.search	100 MiB
r3.large.search	10 MiB
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

实例类型	HTTP 请求负载的最大值
r4.xlarge.search	100 MiB
r4.2xlarge.search	100 MiB
r4.4xlarge.search	100 MiB
r4.8xlarge.search	100 MiB
r4.16xlarge.search	100 MiB
r5.large.search	100 MiB
r5.xlarge.search	100 MiB
r5.2xlarge.search	100 MiB
r5.4xlarge.search	100 MiB
r5.12xlarge.search	100 MiB
r6g.large.search	100 MiB
r6g.xlarge.search	100 MiB
r6g.2xlarge.search	100 MiB
r6g.4xlarge.search	100 MiB
r6g.8xlarge.search	100 MiB
r6g.12xlarge.search	100 MiB
r6gd.large.search	100 MiB

实例类型	HTTP 请求负载的最大值
r6gd.xlarge.search	100 MiB
r6gd.2xlarge.search	100 MiB
r6gd.4xlarge.search	100 MiB
r6gd.8xlarge.search	100 MiB
r6gd.12xlarge.search	100 MiB
r6gd.16xlarge.search	100 MiB
i2.xlarge.search	100 MiB
i2.2xlarge.search	100 MiB
i3.large.search	100 MiB
i3.xlarge.search	100 MiB
i3.2xlarge.search	100 MiB
i3.4xlarge.search	100 MiB
i3.8xlarge.search	100 MiB
i3.16xlarge.search	100 MiB
or1.medium.search	10 MiB
or1.large.search	100 MiB

实例类型	HTTP 请求负载的最大值
or1.xlarge.search	100 MiB
or1.2xlarge.search	100 MiB
or1.4xlarge.search	100 MiB
or1.8xlarge.search	100 MiB
or1.12xlarge.search	100 MiB
or1.16xlarge.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

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	100GiB	65 GiB
Im4gn	不适用	65 GiB

要请求提高配额，请联系 AWS Support。

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 OpenSearch Service 中的预留实例

与标准按需实例相比，Amazon OpenSearch Service 预留实例 (RI) 提供大幅折扣。实例本身是相同的；RI 只是对账户中的按需实例所应用的账单折扣。对于使用可预测、长期存在的应用程序，随着时间的推移，RI 可以提供可观的节省时间。

OpenSearch Service RI 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。

RI 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。有关完整详情，请参阅 Amazon OpenSearch Service 定价和常见问题。

主题

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. 转至 https://aws.amazon.com，然后选择 Sign In to the Console (登录控制台)。
2. 在 Analytics 下，选择 Amazon OpenSearch Service。
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 AWS 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。有关操作指南，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 Next（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 (AWS CLI)

AWS CLI 具有查看产品、购买预留实例和查看预留实例的命令。以下命令和示例响应显示指定 AWS 区域 的产品：

```
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。

Field	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。

Field	描述
UsagePrice	旧字段。对于 OpenSearch Service，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。
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 OpenSearch Service 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 AWS CLI 命令引用。

购买预留实例 (AWS SDK)

AWS SDK（除 Android 和 iOS SDK 之外）支持在 Amazon OpenSearch Service API 参考中定义的所有操作，包括：

- `DescribeReservedInstanceOfferings`
- `PurchaseReservedInstanceOffering`
- `DescribeReservedInstances`

此示例脚本使用 AWS SDK for Python (Boto3) 中的 OpenSearchService 低级别 Python 客户端购买预留实例。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 AWS SDK 的更多信息，请参阅 AWS 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。分析此数据有助于确定趋势和了解 RI 是否适合使用案例。如果已有 RI，则可分组 (按 Purchase Option (购买选项)) 并显示摊销费用以将此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 AWS 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。

Amazon OpenSearch Service 教程

本章包含多个关于使用 Amazon OpenSearch Service 的完整教程，包括如何迁移到服务，如何构建简单的搜索应用程序，以及如何在 OpenSearch 中控制面板。

主题

教程：在 Amazon OpenSearch Service 中创建和搜索文档

在本教程中，您将了解如何在 Amazon OpenSearch Service 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：

- 您必须具有 AWS 账户。
- 您必须具有一个活动的 OpenSearch Service 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 Postman、cURL 或 OpenSearch 控制面板控制台。这些示例假定您使用 OpenSearch 控制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 OpenSearch Service 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

OpenSearch Service 可以自动为您的文档生成一个 ID。生成 ID 的命令使用 POST 请求而非 PUT 请求，并且不需要文档 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，响应中的其他 `_id` 字段表示自动创建的 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，OpenSearch Service 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"
  ]
}
```

相关的 资源

有关更多信息，请参阅以下资源：

- 开始使用
- 为数据建立索引
- 搜索数据

教程：迁移至 Amazon OpenSearch Service

要从自行管理的 OpenSearch 集群或遗留的 Elasticsearch 集群迁移到 Amazon OpenSearch Service，的一种常用的方式是使用索引快照。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 Service 域。
3. 授予 OpenSearch Service 访问存储桶的权限，并确保您有权限使用快照。
4. 恢复 OpeSearch Service 域上的快照。

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

尽管可以使用 repository-s3 插件直接将快照生成到 S3，但必须在每个节点上安装此插件，调整 `opensearch.yml`（如果使用的是 Elasticsearch 集群，则需要调整 `elasticsearch.yml`），重新启动每个节点，添加 AWS 凭证，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 AWS CLI 将其上传到 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 AWS 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://bucket-name --region us-west-2
aws s3 sync . s3://bucket-name --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

虽然控制台是创建域的最简单方法，但在这种情况下，您已经打开了终端并安装了 AWS CLI。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。如果需要更高级的安全配置（如 VPC），请使用 OpenSearch Service 控制台。

在发布命令之前，请更改域名、主用户凭证和账号。指定与您用于 S3 存储桶的相同 AWS 区域，以及与您的快照兼容的 OpenSearch/Elasticsearch 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 Elasticsearch 7.x 集群上的 OpenSearch 1.x 集群还原快照，只能从 OpenSearch 1.x 或 2.x 集群还原快照。次要版本也很重要。您无法从 5.3.2 OpenSearch Service 域上自行托管的 5.3.3 集群还原快照。我们建议您选择快照支持的最新版本的 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 AWS 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 AWS服务。将该角色命名为 `OpenSearchSnapshotRole`，以便于查找。

权限

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

```
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

信任关系

```
{
  "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

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
    }
  ]
}
```

映射 OpenSearch 控制面板中的快照角色（如果使用精细访问控制）

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. 然后使用您在创建 OpenSearch Service 域时指定的主用户凭证录入到控制面板。您可以在 OpenSearch Service 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
```

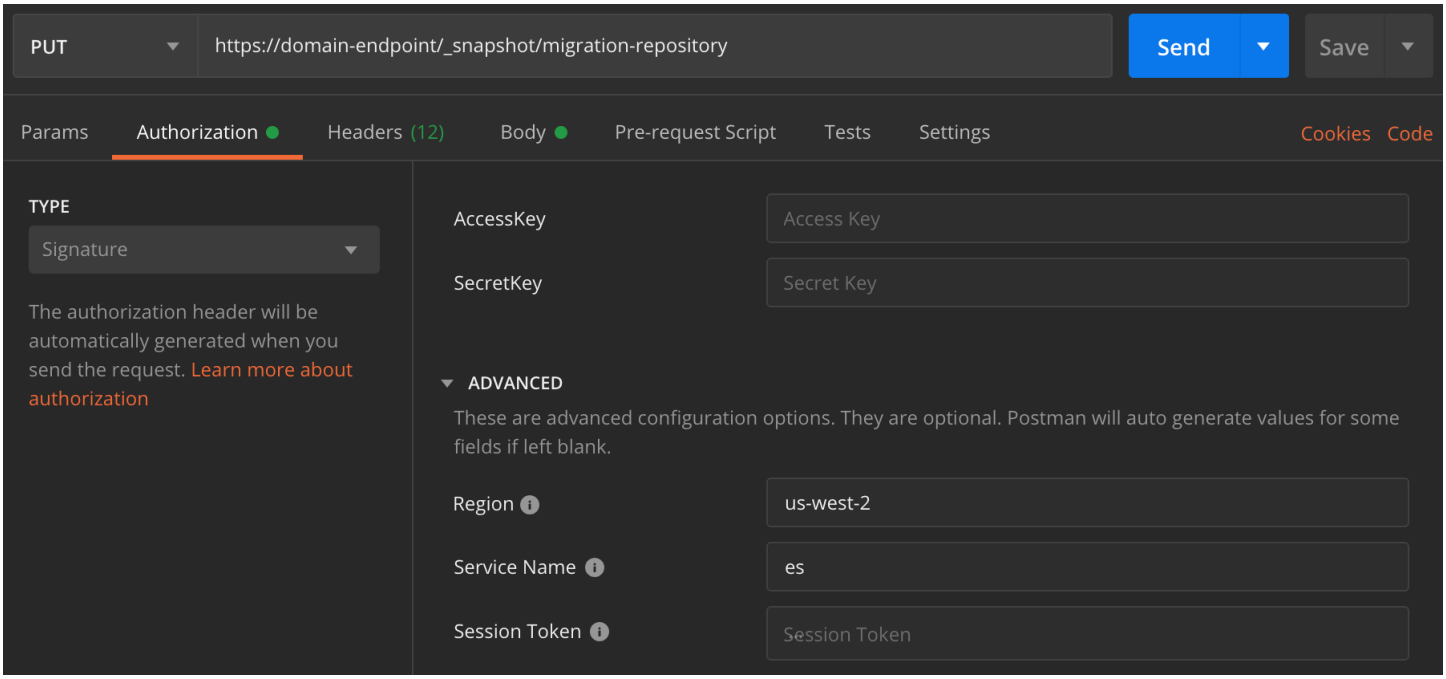
```
arn:aws:iam::123456789123:role/role-name
```

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，可以通过两种方式访问 OpenSearch Service 域：使用您的主用户凭证进行 HTTP 基本身份验证，或者使用 IAM 凭证进行 AWS 身份验证。由于快照使用 Amazon S3（没有主用户的概念），因此必须使用 IAM 凭证向您的 OpenSearch Service 域注册快照存储库。

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": "bucket-name",
    "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

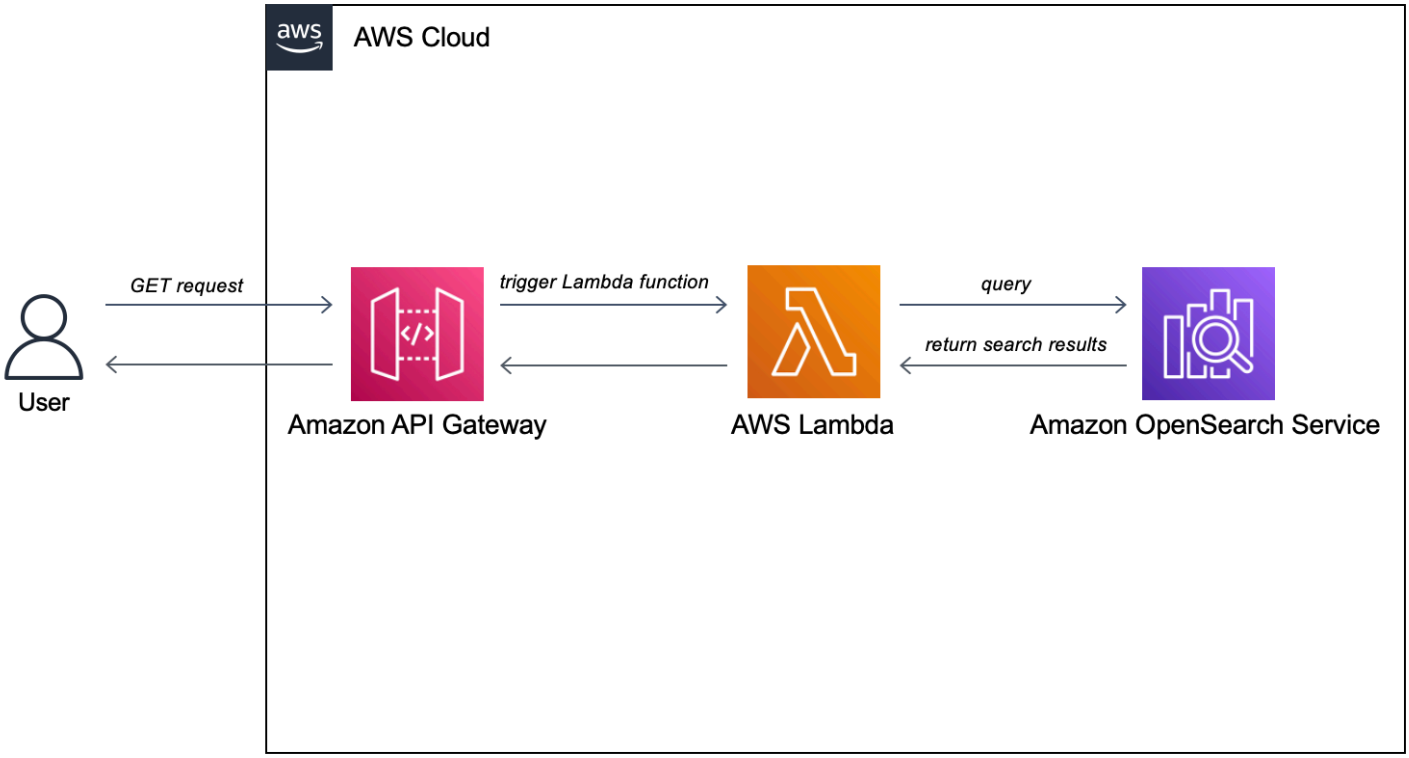
   ```
   curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/_cat/
   indices?v
   ```

此时，迁移已完成。您可以将客户端配置为使用新的 OpenSearch Service 端点，调整域大小以适应您的工作负载，检查索引的分片计数，切换到 IAM 主用户，或开始构建 OpenSearch 控制面板。

教程：使用 Amazon OpenSearch Service 创建搜索应用程序

使用 Amazon OpenSearch Service 创建搜索应用程序的一个常用方法是使用 Web 表单将用户查询发送到服务器。然后，您可以授权服务器直接调用 OpenSearch API 并让服务器向 OpenSearch Service 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。不建议允许对 OpenSearch API 的未签名公有访问权限。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。

本章为您提供了一个解决方案：使用 Amazon API Gateway 将用户限制到一部分 OpenSearch API 和 AWS Lambda，并从 API Gateway 到 OpenSearch Service 签署请求。



> **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

此教程的一个先决条件是 OpenSearch Service 域。如果您还没有域，请按照创建 OpenSearch Service 域中的步骤创建一个域。

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 会将请求传递到 Lambda 函数，该函数将查询 OpenSearch Service 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 AWS 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. 导航到 Lambda 控制台：https://console.aws.amazon.com/lambda/home。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 创建更加受限的 API 简化了与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. 通过以下网址导航到 API Gateway 控制台：https://console.aws.amazon.com/apigateway/home。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 Amazon OpenSearch Service 域的公共 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	可以
Lambda 区域	*us-west-1*
Lambda 函数	opensearch-lambda
使用原定设置超时	可以

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
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
姓名	q
必需	可以

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	可以
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

OpenSearch Service 域必须允许 Lambda 函数向 `movies` 索引发出 GET 请求。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：

```
{
  "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 AWS Identity and Access Management(IAM) 控制台，请选择角色，并搜索 “lambda”。

```
{
  "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访问控制，则还需要在 OpenSearch 控制面板中将角色映射到用户，否则就会看到权限错误。

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）

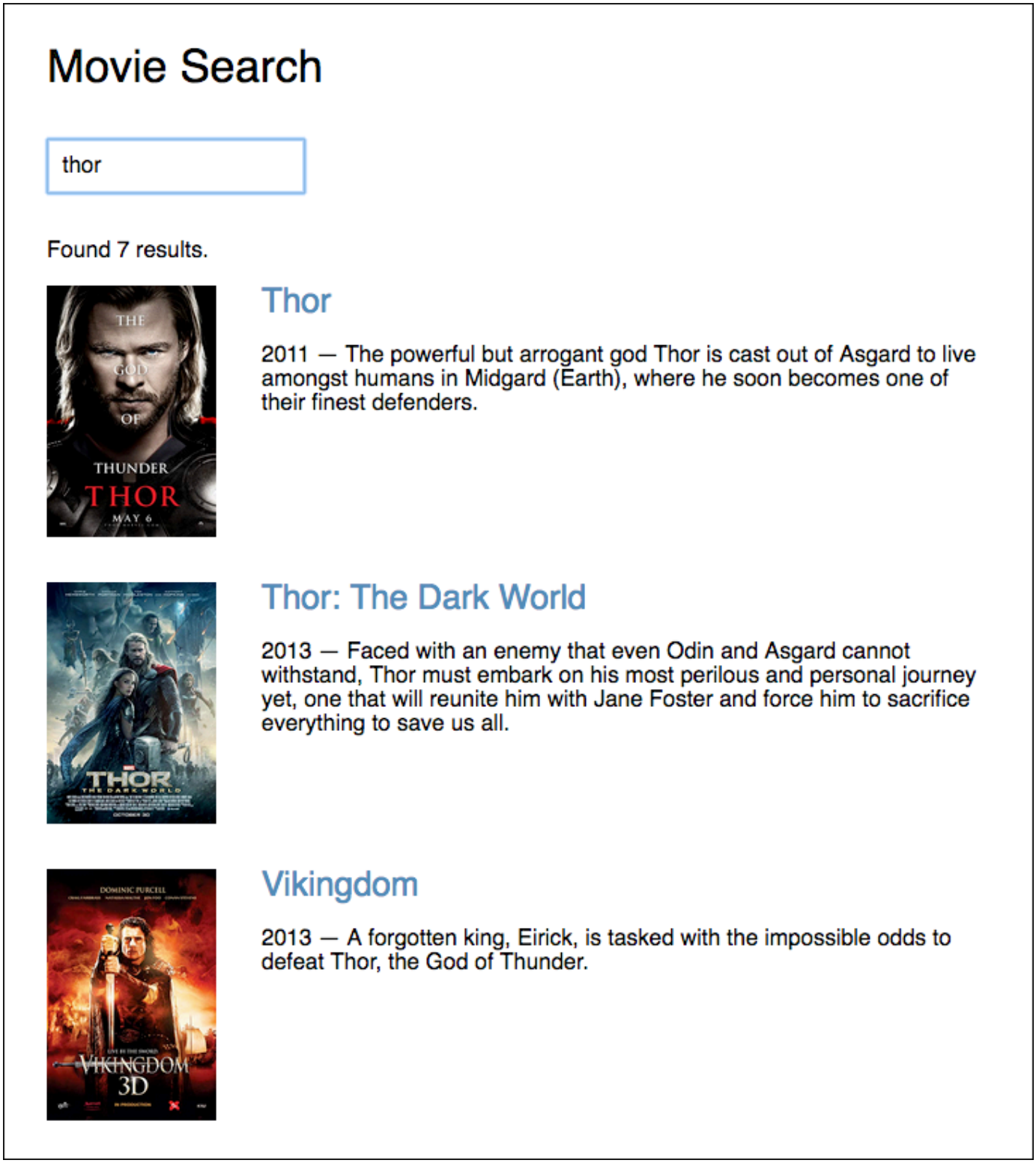
精细访问控制将在您能测试应用程序之前引入一个额外步骤。即使您将 HTTP 基本身份验证用于所有其他目的，也需要将 Lambda 角色映射到用户，否则您将看到权限错误。

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 Service 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 OpenSearch Service 和 OpenSearch 控制面板可视化客户支持呼叫

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 Transcribe、Amazon Comprehend和 Amazon OpenSearch Service ，您只需极少的代码便可自动完成类似流程，并可获得更多数据。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 和 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 OpenSearch Service 中为文档创建索引之前，激发您有关如何丰富 JSON 文档的创意。

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。
- 具有一个 `t2.medium` 实例和可使用几个小时的 10GiB EBS 存储的 OpenSearch Service 域

  要了解更多信息，请参阅Amazon OpenSearch Service 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

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 Service 域	数据的目的地。有关更多信息，请参阅创建 OpenSearch Service 域。

如果还没有这些资源，可以使用以下 AWS 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 AWS 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 与 `call-center.py` 放在同一目录中并运行脚本。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 MP3，但 Amazon Transcribe 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 OpenSearch Service 中为该文档创建索引。

> **Tip**
>
> LibriVox 具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：

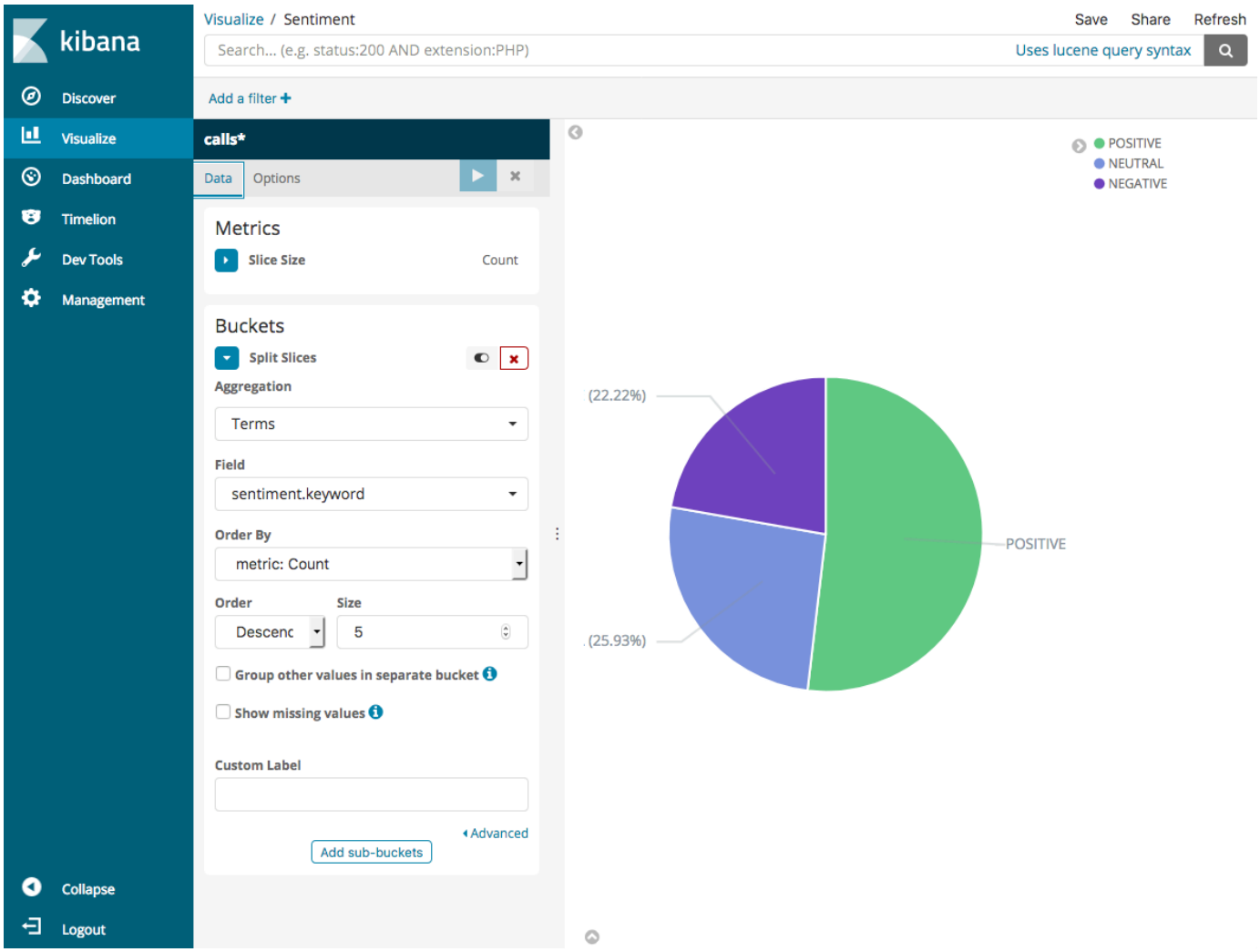
```
$ python bulk-helper.py
{
  "errors": false,
  "items": [
    {
      "index": {
        "_id": "1",
        "_index": "support-calls",
        "_primary_term": 1,
        "_seq_no": 42,
        "_shards": {
          "failed": 0,
          "successful": 1,
          "total": 2
        },
        "_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 OpenSearch Service 中有一些数据，可以使用 OpenSearch 控制面板来可视化这些数据。

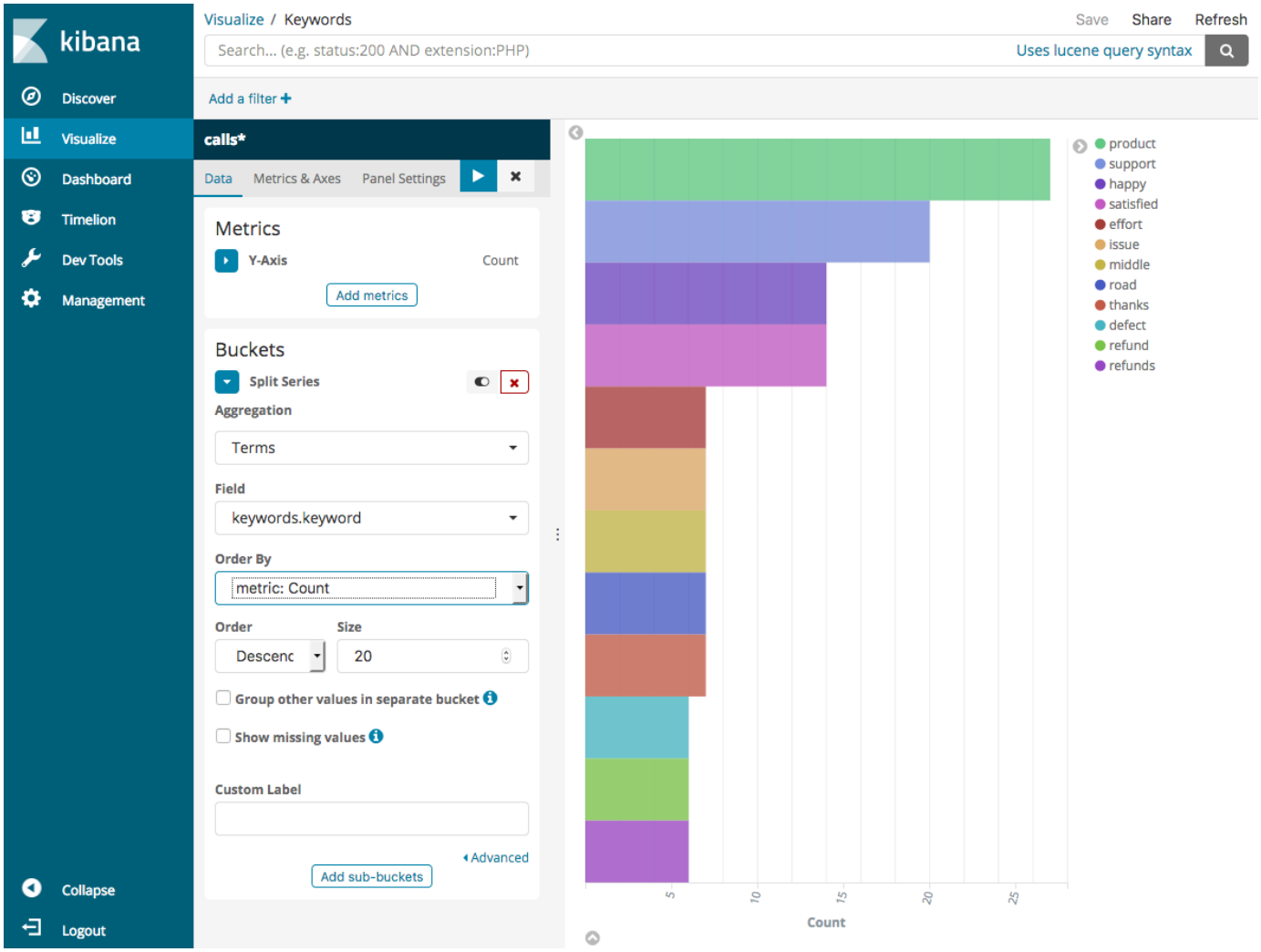
1. 导航到 `https://search-domain.region.es.amazonaws.com/_dashboards`。
2. 在使用 OpenSearch 控制面板之前，您需要索引模式。控制面板使用索引模式来将分析范围缩小到一个或多个索引。要匹配 `call-center.py` 创建的 `support-calls` 索引，转到堆栈管理、索引模式，并定义索引模式`support*`，然后选择下一步。
3. 对于 Time Filter field name (时间筛选字段名称)，选择 timestamp (时间戳)。
4. 现在，您可以开始创建可视化了。选择 Visualize (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)。

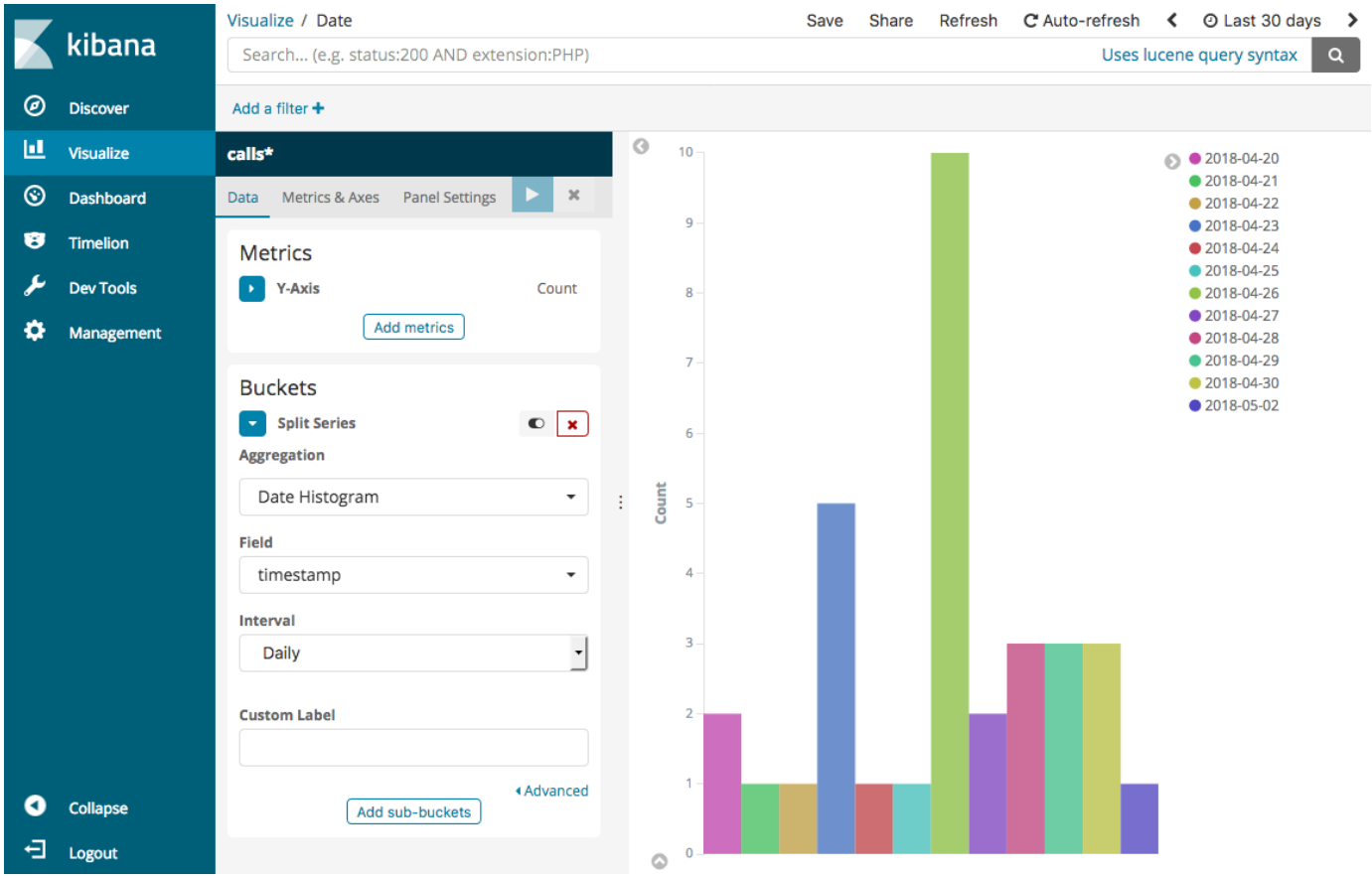


7. 返回到 Visualize (可视化) 页面，然后添加其他可视化。这次请选择水平条形图。
8. 选择 Split Series (拆分序列)。

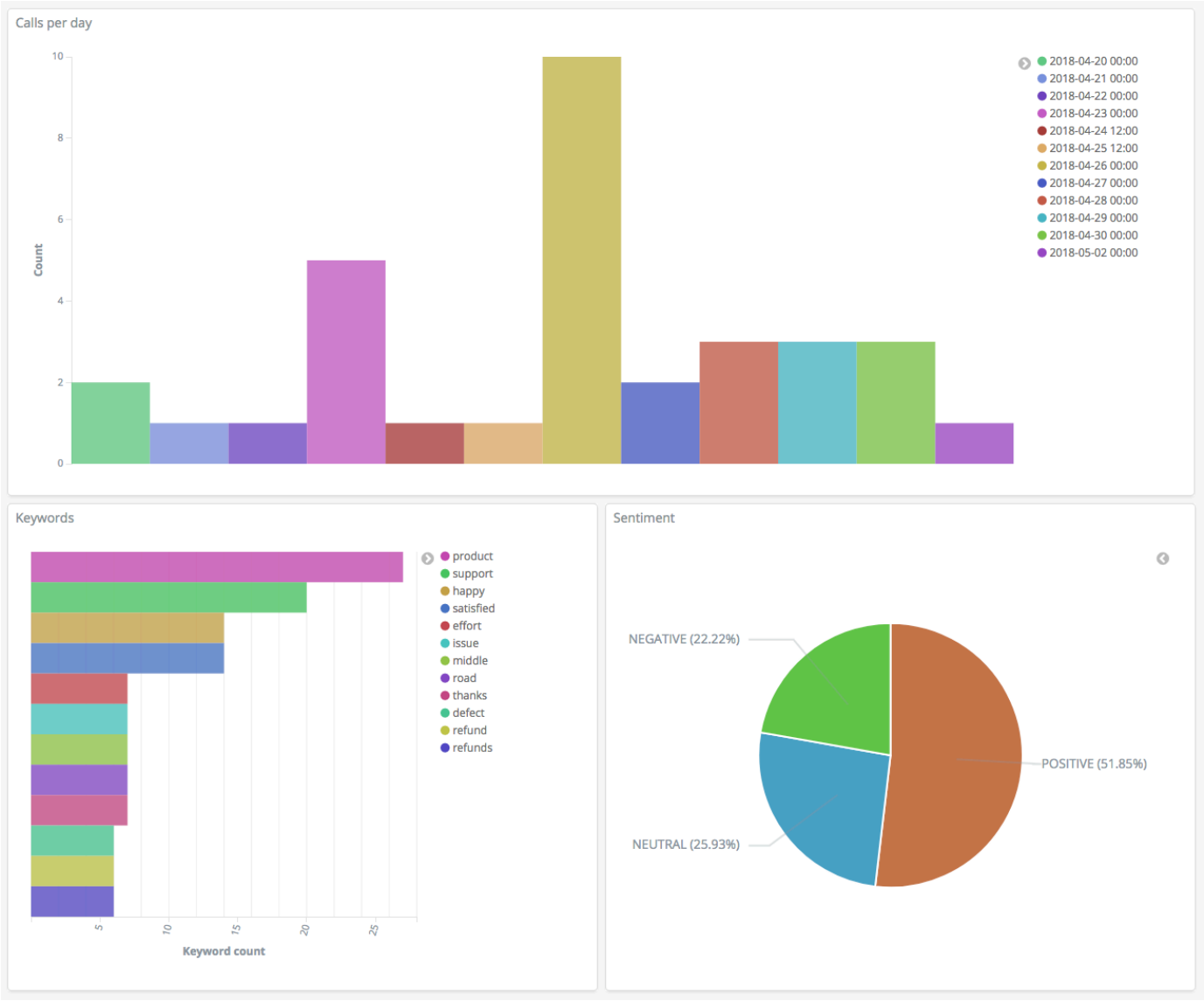
   对于 Aggregation，选择 Terms。对于 Field (字段)，请选择 keywords.keyword，并将 Size (大小) 更改为 20。然后选择 Apply Changes (应用更改) 和 Save (保存)。



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 OpenSearch Service 域。要了解详情，请参阅 Amazon Simple Storage Service 用户指南中的删除存储桶以及本指南中的删除 OpenSearch Service 域。

脚本需要的磁盘空间远远少于 MP3 文件。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 AWS 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 OpenSearch Service 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 OpenSearch Service。OpenSearch Service 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 时才适用。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。

主题

新 API 版本

新版本的 OpenSearch Service 配置 API (2021-01-01) 适用于 OpenSearch 以及传统的 Elasticsearch 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` 已更改为 `CreateDomain`），但是 OpenSearch Service 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。如果您不指定版本，则默认为 OpenSearch 的最新版本。

将 AWS CLI 升级到版本 1.20.40 或更高版本以使用 `aws opensearch ...` 创建和管理于。有关新的 CLI 格式，请参阅 OpenSearch CLI 引用。

重命名的实例类型

Amazon OpenSearch Service 中的实例类型现在采用格式 `<type>.<size>.search`，例如 `m6g.large.search`，而不是 `m6g.large.elasticsearch`。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 (RI)，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 policy

我们建议您更新 IAM 策略以使用重命名的 API 操作。但是，OpenSearch Service 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 (SCP) 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 OpenSearch Service 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 AWS CloudFormation 和 AWS 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 CloudFormation 中定义的域从 Elasticsearch 升级到 OpenSearch 的说明，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 AWS CloudTrail 和 AWS Config 等服务，它引用了在域上运行的软件而不是整个 OpenSearch Service。这些服务现在包含单独的资源类型，用于运行 Elasticsearch (`AWS::Elasticsearch::Domain`) 与运行 OpenSearch (`AWS::OpenSearch::Domain`) 的域。

> **Note**
>
> 在 AWS Config 中，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 OpenSearch 也是如此。

Kibana 重命名为 OpenSearch 控制面板

OpenSearch 控制面板，Kibana 的 AWS 替代，是一种开源可视化工具，专为与 OpenSearch 结合使用而设计。将域从 Elasticsearch 升级到 OpenSearch 后，`/_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` 端点。

如果您使用的是 the section called “仪表板的 SAML 身份验证 OpenSearch ”，在将域升级到 OpenSearch 之前，您需要将身份提供商 (IdP) 中配置的所有 Kibana URL 从 `/_plugin/kibana` 更改为 `/_dashboards`。最常见的 URL 是断言使用者服务 (ACS) URL 和收件人 URL。

OpenSearch 控制面板的默认 `kibana_read_only` 角色已重命名为 `opensearch_dashboards_read_only`，而 `kibana_user` 角色已重命名为 `opensearch_dashboards_user`。该更改适用于所有新创建的、运行服务软件 R20211203 或更高版本的 OpenSearch 1.x 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

运行 OpenSearch 的域的多个 CloudWatch 指标发生了变化。当您将域升级到 OpenSearch 时，指标会自动更改，当前 CloudWatch 警报将中断。在将集群从 Elasticsearch 版本升级到 OpenSearch 版本之前，请确保更新您的 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`

原始指标名称	新名称
KibanaOS1MinuteLoad	OpenSearchDashboardsOS1MinuteLoad
KibanaRequestTotal	OpenSearchDashboardsRequestTotal
KibanaResponseTimesMaxInMillis	OpenSearchDashboardsResponseTimesMaxInMillis
ESReportingFailedRequestSysErrCount	KibanaReportingFailedRequestSysErrCount
ESReportingRequestCount	KibanaReportingRequestCount
ESReportingFailedRequestUserErrCount	KibanaReportingFailedRequestUserErrCount
ESReportingSuccessCount	KibanaReportingSuccessCount
ElasticsearchRequests	OpenSearchRequests

有关 OpenSearch Service 发送给 Amazon CloudWatch 的完整指标列表，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 OpenSearch Service 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 OpenSearch Service。当您的 Elasticsearch 使用率减少并增加 OpenSearch 使用率时，您最初可能会收到警报，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：

Field	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	Amazon OpenSearch Service 实例 Amazon OpenSearch Service 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 Service 发送给 Amazon EventBridge 和 Amazon CloudWatch 的事件格式已更改，特别是 `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 ARN
- 域终端节点

入门：将您的域升级到 OpenSearch 1.x

OpenSearch 1.x 支持从 Elasticsearch 版本 6.8 和 7.x 升级。有关升级域的说明，请参阅 the section called “开始升级（控制台）”。如果您使用的是 AWS CLI 或配置 API 来升级域，则需要指定 `TargetVersion` 作为 `OpenSearch_1.x`。

OpenSearch 1.x 引入了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，因此兼容模式将 OpenSearch 设置为报告其版本为 7.10，以便这些客户端继续工作。

首次创建 OpenSearch 域时，或者从 Elasticsearch 版本升级到 OpenSearch 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"
   }
}
```

要在现有的 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 AWS 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 OpenSearch 仪表板的访问权限”、the section called “关于 VPC 域的访问策略” 和 the section called “Identity and Access Management”。

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AWS 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 AWS Health Dashboard，请参阅《AWS 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。

群集状态为红色的最常见原因是群集节点出现故障，以及由于持续繁重的处理负载而导致 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， AWS Support 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。为了识别导致红色集群状态的索引，有一些 OpenSearch 有用的 API。

- `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，S OpenSearch ervice 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
JVM MemoryPressure	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`	
CPU 利用率	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 无法向其他节点分配副本，因此单节点集群始终以黄色群集状态初始化。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 S OpenSearch ervice 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 JVM MemoryPressure 指标在 30 分钟内超过 92% 时，Serv OpenSearch ice 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 JVM MemoryPressure 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 AWS 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

断路器	描述	集群设置属性
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 实例可能会遇到意外终止并重新启动。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仍处于故障状态。

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 AWS 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。

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 AWS 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 AWS CLI、或 SDK 来配置您的域。您也可以使用 OpenSearch REST API 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，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 AWS 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 AWS 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 AWS 区域的域摘要

以下脚本使用 Amazon EC2 describe- regions AWS 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
- 跨数据节点的 JVM 堆和/或 CPU 利用率不均匀

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 AWS 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 AWS Security Token Service（AWS 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，AWS 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 AWS 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。

```
{
  "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 AWS SDK 使用您计算机上的 CA 证书，因此当您尝试使用 SDK 时，更改 AWS 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：

- 已安装 2005 年 1 月版或更高版本更新的 Microsoft Windows 版本在其信任列表中至少包含一个必需 CA。
- 带 Java for Mac OS X 10.4 版本 5 的 Mac OS X 10.4 (2007 年 2 月版)、Mac OS X 10.5 (2007 年 10 月版) 及更高版本在其信任列表中至少包含一个必需 CA。
- Red Hat Enterprise Linux 5 (2007 年 3 月版)、6 和 7 以及 CentOS 5、6 和 7 在其默认信任 CA 列表中至少包含一个必需 CA。
- Java 1.4.2_12（2006 年 5 月版）、5 Update 2（2005 年 3 月版）以及所有更高版本，包括 Java 6（2006 年 12 月版）、7 和 8 在其默认信任 CA 列表中至少包含一个必需 CA。

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 AWS Certificate Manager 常见问题。

> **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。

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 AWS 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 AWS？

此历史记录的相关日期：

- 当前产品版本—2021-01-01
- 最新产品发布 — 2024 年 6 月 12 日
- 最新文档更新 — 2024 年 6 月 12 日

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	2024 年 6 月 12 日

	终端 Amazon CloudWatch 节点的管道发送指标数据。	
亚马逊 OpenSearch 服务零 ETL 与亚马逊 S3 集成	亚马逊 OpenSearch 服务现在支持直接查询在 Amazon S3 中查询数据。	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 2.12 和 2.13 版本中包含的所有功能。有关更多信息，请参阅 2.12 和 2. 13 发行说明。	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 版本说明。	2024 年 4 月 4 日
AWS 服务 OpenSearch 无服务器集合的私有访问权限	现在 AWS 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
配置变更可见性	现在，您可以在亚马逊 OpenSearch 服务控制台中使用配置 API 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	亚马逊 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在亚马逊 OpenSearch 服务中提供这些数据。	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。此版本还支持所有集合类型的最大允许容量为 200 个 OCU，并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 版本说明。此外，您可以将 Amazon DynamoDB 指定为管道来源。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 OpenSearch ervice 在控制台中提供了一个 AWS 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AmazonOpenSearchServiceRolePolicy 分配和取消分配 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 API 或配置界面来设置时间序列集合的数据留存策略，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，可以进行精细控制。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 Analysis（中文），以及亚马逊个性化搜索排名插件。	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	Amazon 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 AWS 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，您现在可以指定 Amazon Managed Streaming for Apache Kafka (Amazon MSK) 作为管道来源。	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。此版本还对搜索和时间序列集合提供最大允许容量为 100 个 OCU 的支持。	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 指定为管道来源。	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 (AZ) 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，它可以安排需要蓝/绿部署的服务软件更新和自动调整优化。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： • 当集合端点上的流量减少时，容量现在可以缩减到最少的配置 OCU 数。 • 索引和搜索允许的最大 OCU 数从 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蓝/绿部署。	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
AWS 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 终端节点只能在预配置终端节点的 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 对等的任何 VPC 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 会在部署蓝/绿之前执行验证检查，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	亚马逊 OpenSearch 服务增加了对机器学习共享插件的支持，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 OpenSearch API 的访问权限。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， AWS 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`JVMMemory Pressure` 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 `JVMMemory Pressure` 、`MasterJVM MemoryPressure` 和/或 `WarmJVMMemoryPress ure` 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 AWS托管策略	Amazon S OpenSearch ervice 的推出包括新的 AWS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	AWS 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 AWS 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 Elasticse arch 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 Serv OpenSearch ice 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。有关更多信息，请参阅 。	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，并且 AWS 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	`cn-north-1` 中国（北京）区域和`cn-northwest-1` 中国（宁夏）区域现已提供静态数据 node-to-node 加密和加密功能。	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 及更高版本的域创建快照，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 S OpenSearch ervice 现在在许多地区支持三个可用区。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
N ode-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 “OpenSearch 控制面板的 Amazon Cognito 认证”。	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 AWS Support 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

AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 AWS 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。