

云搜索服务

用户指南

文档版本 01
发布日期 2024-08-15



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 CSS 服务权限管理	1
1.1 创建 IAM 用户并授权使用 CSS	1
1.2 示例：为 IAM 用户（子用户）配置创建 CSS 集群的权限	4
2 使用 Elasticsearch 搜索数据	10
2.1 Elasticsearch 使用流程	10
2.2 Elasticsearch 集群规划建议	13
2.3 创建 Elasticsearch 集群	24
2.4 访问 Elasticsearch 集群	31
2.4.1 Elasticsearch 集群访问方式	31
2.4.2 通过 Kibana 登录 Elasticsearch 集群	34
2.4.3 通过 Cerebro 登录 Elasticsearch 集群	37
2.4.4 通过 Curl 命令行接入 Elasticsearch 集群	38
2.4.5 通过 Java 客户端接入 Elasticsearch 集群	39
2.4.5.1 通过 Rest High Level Client 接入 Elasticsearch 集群	39
2.4.5.2 通过 Rest Low Level Client 接入 Elasticsearch 集群	48
2.4.5.3 通过 Transport Client 接入 Elasticsearch 集群	63
2.4.5.4 通过 Spring Boot 接入 Elasticsearch 集群	64
2.4.6 通过 Python 客户端接入 Elasticsearch 集群	70
2.4.7 通过 MRS Hive 客户端接入 Elasticsearch 集群	72
2.4.8 通过 Go 客户端接入 Elasticsearch 集群	78
2.5 导入数据至 Elasticsearch 集群	81
2.5.1 Elasticsearch 集群导入数据方式	81
2.5.2 使用自建 Logstash 导入数据到 Elasticsearch	82
2.5.3 使用开源 Elasticsearch API 导入数据到 Elasticsearch	90
2.5.4 使用 CDM 导入数据到 Elasticsearch	94
2.5.5 增强 Elasticsearch 集群数据导入性能	97
2.6 使用 Elasticsearch 集群搜索数据	100
2.6.1 使用 DSL 语言在 Elasticsearch 中搜索数据	100
2.6.2 使用 SQL 语言在 Elasticsearch 中搜索数据	101
2.7 增强 Elasticsearch 集群搜索能力	105
2.7.1 Elasticsearch 集群搜索增强特性介绍	105
2.7.2 配置 Elasticsearch 集群向量检索	107
2.7.2.1 向量检索特性介绍	107

2.7.2.2 创建向量索引.....	109
2.7.2.3 使用向量索引搜索数据.....	116
2.7.2.4 优化向量检索写入与查询性能.....	120
2.7.2.5 管理向量索引缓存.....	121
2.7.2.6 向量检索的客户端代码示例 (Python)	121
2.7.2.7 向量检索的客户端代码示例 (Java)	123
2.7.2.8 示例：使用 PV_GRAPH 算法实现向量检索.....	127
2.7.3 配置 Elasticsearch 集群存算分离.....	129
2.7.4 配置 Elasticsearch 集群读写流量控制策略 2.0.....	143
2.7.5 配置 Elasticsearch 集群读写流量控制策略 1.0.....	152
2.7.6 配置 Elasticsearch 集群大查询隔离.....	165
2.7.7 配置 Elasticsearch 集群聚合增强.....	170
2.7.8 配置 Elasticsearch 集群读写分离.....	173
2.7.9 切换 Elasticsearch 集群冷热数据.....	179
2.7.10 配置 Elasticsearch 集群索引回收站.....	181
2.8 配置 Elasticsearch 集群网络.....	183
2.8.1 配置公网访问.....	183
2.8.2 配置终端节点服务.....	186
2.8.3 配置独享型负载均衡.....	189
2.9 备份与恢复 Elasticsearch 集群数据.....	198
2.9.1 创建快照备份 Elasticsearch 集群数据.....	198
2.9.2 恢复 Elasticsearch 集群数据.....	204
2.9.3 删除 Elasticsearch 集群快照.....	206
2.10 扩缩容 Elasticsearch 集群.....	207
2.10.1 扩容 Elasticsearch 集群.....	207
2.10.2 缩容 Elasticsearch 集群.....	212
2.11 升级 Elasticsearch 集群版本.....	216
2.12 管理 Elasticsearch 集群.....	221
2.12.1 查看 Elasticsearch 集群信息.....	222
2.12.2 创建 Elasticsearch 集群用户并授权使用集群.....	225
2.12.3 设置 Elasticsearch 集群标签.....	234
2.12.4 配置 Elasticsearch 集群 YML 文件默认参数.....	236
2.12.5 为 Elasticsearch 集群绑定企业项目.....	237
2.12.6 替换 Elasticsearch 集群指定节点.....	238
2.12.7 更改 Elasticsearch 集群安全模式.....	239
2.12.8 切换 Elasticsearch 集群可用区.....	244
2.12.9 配置 Elasticsearch 集群自定义词库.....	246
2.12.9.1 Elasticsearch 集群词库介绍.....	246
2.12.9.2 配置和使用 Elasticsearch 集群的自定义词库.....	248
2.12.10 在 Elasticsearch 集群实现搜索数据的简繁体转换.....	257
2.12.11 重启 Elasticsearch 集群.....	259
2.12.12 删除 Elasticsearch 集群.....	260

2.13 管理 Elasticsearch 集群索引策略.....	261
2.13.1 创建和管理 Elasticsearch 索引策略.....	261
2.13.2 通过索引生命周期实现时间序列数据滚动索引.....	264
2.13.3 通过索引生命周期实现索引存算分离.....	267
2.14 Elasticsearch 集群监控与日志管理.....	269
2.14.1 Elasticsearch 集群支持的监控指标.....	270
2.14.1.1 CES 中 Elasticsearch 集群支持的监控指标.....	270
2.14.1.2 Elasticsearch 集群内核支持的监控指标.....	293
2.14.2 配置 Elasticsearch 集群监控.....	293
2.14.2.1 使用 CES 监控 Elasticsearch 集群.....	294
2.14.2.2 配置 Elasticsearch 集群内核监控.....	296
2.14.2.3 配置 Elasticsearch 集群索引监控.....	298
2.14.3 设置 Elasticsearch 集群 SMN 告警通知.....	308
2.14.4 智能检测 Elasticsearch 集群风险.....	313
2.14.5 查询和管理 Elasticsearch 集群日志.....	315
2.15 查看 Elasticsearch 集群审计日志.....	319
3 使用 OpenSearch 搜索数据.....	321
3.1 OpenSearch 使用流程.....	321
3.2 OpenSearch 集群规划建议.....	323
3.3 创建 OpenSearch 集群.....	333
3.4 访问 OpenSearch 集群.....	340
3.4.1 OpenSearch 集群访问方式.....	340
3.4.2 通过 OpenSearch Dashboards 登录 OpenSearch 集群.....	343
3.4.3 通过 Cerebro 登录 OpenSearch 集群.....	346
3.4.4 通过 Curl 命令行接入 OpenSearch 集群.....	347
3.5 导入数据至 OpenSearch 集群.....	348
3.5.1 OpenSearch 集群导入数据方式.....	349
3.5.2 使用自建 Logstash 导入数据到 OpenSearch.....	350
3.5.3 使用开源 OpenSearch API 导入数据到 OpenSearch.....	357
3.5.4 使用 CDM 导入数据到 OpenSearch.....	360
3.5.5 增强 OpenSearch 集群数据导入性能.....	363
3.6 使用 OpenSearch 集群搜索数据.....	366
3.6.1 使用 DSL 语言在 OpenSearch 中搜索数据.....	366
3.6.2 使用 SQL 语言在 OpenSearch 中搜索数据.....	368
3.7 增强 OpenSearch 集群搜索能力.....	372
3.7.1 OpenSearch 集群搜索增强特性介绍.....	372
3.7.2 配置 OpenSearch 集群存算分离.....	373
3.7.3 切换 OpenSearch 集群冷热数据.....	386
3.8 配置 OpenSearch 集群网络.....	388
3.8.1 配置公网访问.....	388
3.8.2 配置终端节点服务.....	391
3.8.3 配置独享型负载均衡.....	394

3.9 备份与恢复 OpenSearch 集群数据.....	403
3.9.1 创建快照备份 OpenSearch 集群数据.....	403
3.9.2 恢复 OpenSearch 集群数据.....	409
3.9.3 删除 OpenSearch 集群快照.....	411
3.10 扩缩容 OpenSearch 集群.....	412
3.10.1 扩容 OpenSearch 集群.....	412
3.10.2 缩容 OpenSearch 集群的节点数量.....	417
3.11 管理 OpenSearch 集群.....	421
3.11.1 查看 OpenSearch 集群信息.....	421
3.11.2 使用 OpenSearch Dashboard 创建用户并授权.....	425
3.11.3 设置 OpenSearch 集群标签.....	430
3.11.4 配置 OpenSearch 集群 YML 文件默认参数.....	432
3.11.5 为 OpenSearch 集群绑定企业项目.....	433
3.11.6 替换 OpenSearch 集群指定节点.....	434
3.11.7 更改 OpenSearch 集群安全模式.....	435
3.11.8 切换 OpenSearch 集群可用区.....	440
3.11.9 配置 OpenSearch 集群自定义词库.....	442
3.11.9.1 OpenSearch 集群词库介绍.....	442
3.11.9.2 配置和使用 OpenSearch 集群的自定义词库.....	444
3.11.10 在 OpenSearch 集群实现搜索数据的简繁体转换.....	453
3.11.11 重启 OpenSearch 集群.....	455
3.11.12 删除 OpenSearch 集群.....	456
3.12 管理 OpenSearch 集群索引策略.....	457
3.12.1 创建及管理 OpenSearch 索引策略.....	457
3.12.2 通过索引生命周期实现时间序列数据滚动索引.....	460
3.13 OpenSearch 集群监控与日志管理.....	463
3.13.1 CES 中 OpenSearch 集群支持的监控指标.....	463
3.13.2 使用 CES 监控 OpenSearch 集群.....	486
3.13.3 设置 OpenSearch 集群 SMN 告警通知.....	488
3.13.4 智能检测 OpenSearch 集群风险.....	493
3.13.5 查询和管理 OpenSearch 集群日志.....	496
3.14 查看 OpenSearch 集群审计日志.....	500
4 使用 Logstash 迁移数据.....	502
4.1 Logstash 使用流程.....	502
4.2 Logstash 集群规划建议.....	503
4.3 创建 Logstash 集群.....	505
4.4 配置 Logstash 集群路由.....	509
4.5 配置 Logstash 迁移任务.....	510
4.5.1 创建 Logstash 配置文件.....	510
4.5.2 启动 Logstash 迁移任务.....	513
4.5.3 停止 Logstash 迁移任务.....	515
4.5.4 Logstash 配置文件模板介绍.....	515

4.6 扩缩容 Logstash 集群.....	524
4.6.1 扩容 Logstash 集群.....	524
4.6.2 缩容 Logstash 集群.....	527
4.7 管理 Logstash 集群.....	528
4.7.1 查看 Logstash 集群信息.....	528
4.7.2 设置 Logstash 集群标签.....	531
4.7.3 为 Logstash 集群绑定企业项目.....	533
4.7.4 管理 Logstash 集群的 CA 证书.....	534
4.7.5 强制重启 Logstash 集群.....	534
4.7.6 删除 Logstash 集群.....	535
4.8 Logstash 集群监控与日志管理.....	536
4.8.1 CES 中 Logstash 集群支持的监控指标.....	536
4.8.2 使用 CES 监控 Logstash 集群.....	540
4.8.3 查询和管理 Logstash 集群日志.....	542
4.9 查看 Logstash 集群审计日志.....	544

1 CSS 服务权限管理

1.1 创建 IAM 用户并授权使用 CSS

如果您需要对您所拥有的CSS服务进行精细的权限管理，您可以使用[统一身份认证服务](#)（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的华为云账号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用CSS资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将CSS资源委托给更专业、高效的其他华为云账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用CSS服务的其他功能。

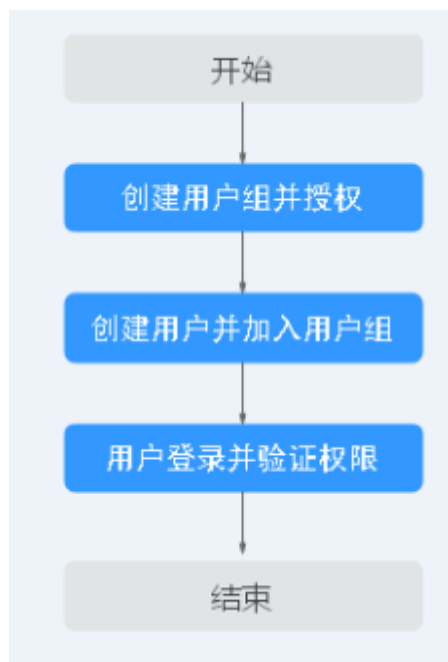
本章节为您介绍对用户授权的方法。

前提条件

给用户组授权之前，请您了解用户组可以添加的CSS权限，并结合实际需求进行选择，CSS支持管理员权限和只读权限，详细的系统权限请参见[权限管理](#)。

示例流程

图 1-1 给用户授权 CSS 权限流程



1. 创建用户组并授权

在IAM控制台创建用户组，并授予云搜索服务只读权限“CSS ReadOnlyAccess”。

2. 创建用户并加入用户组

在IAM控制台创建用户，并将其加入1中创建的用户组。

将用户添加至用户组中，从而使用户拥有对应的CSS权限，一个用户组下面的用户具有相同的权限。

3. 用户登录并验证权限

新创建的用户登录控制台，切换至授权区域，验证权限：

- 在“服务列表”中选择云搜索服务CSS，进入CSS主界面，单击右上角“创建集群”，尝试购买CSS集群，如果无法购买CSS集群（假设当前权限仅包含CSS ReadOnlyAccess），表示“CSS ReadOnlyAccess”已生效。
- 在“服务列表”中选择除云搜索服务外（假设当前策略仅包含CSS ReadOnlyAccess）的任一服务，如果提示权限不足，表示“CSS ReadOnlyAccess”已生效。

CSS 自定义策略样例

如果系统预置的CSS权限，不满足您的授权要求，可以创建自定义策略。自定义策略中可以添加的授权项（Action）请参考[权限策略和授权项](#)。

目前华为云支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。
- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：[创建自定义策略](#)。下面为您介绍常用的CSS自定义策略样例。

📖 说明

为了兼容开源生态，CSS服务的IAM权限和数据面集群权限分开控制，如果要开启数据面的安全能力，需要使用安全模式。

如果是子账号，需要同时设置GetBucketStoragePolicy、GetBucketLocation、ListBucket、ListAllMyBuckets权限，才能看到OBS桶。

- 示例1：授权用户创建集群。

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "css:cluster:create",
        "vpc:securityGroups:get",
        "vpc:securityGroups:create",
        "vpc:securityGroups:delete",
        "vpc:securityGroupRules:get",
        "vpc:securityGroupRules:create",
        "vpc:securityGroupRules:delete",
        "vpc:vpcs:list",
        "vpc:privateIps:list",
        "vpc:ports:get",
        "vpc:ports:create",
        "vpc:ports:update",
        "vpc:ports:delete",
        "vpc:quotas:list",
        "vpc:subnets:get",
        "ecs:cloudServerFlavors:get",
        "ecs:serverInterfaces:use",
        "ecs:cloudServers:addNics",
        "ecs:quotas:get",
        "evs:types:get",
        "evs:quotas:get"
      ],
      "Effect": "Allow"
    }
  ]
}
```

- 示例2：拒绝用户删除集群。

拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循**Deny优先原则**。

如果您给用户授予CSS Admin的系统策略，但不希望用户拥有CSS admin中定义的删除云服务器权限，您可以创建一条拒绝删除云服务的自定义策略，然后同时将CSS Admin和拒绝策略授予用户，根据Deny优先原则，则用户可以对CSS执行除了删除集群外的所有操作。拒绝策略示例如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "css:cluster:delete"
      ]
    }
  ]
}
```

- 示例3：多个授权项策略。

一个自定义策略中可以包含多个授权项，且除了可以包含本服务的授权项外，还可以包含其他服务的授权项，可以包含的其他服务必须跟本服务同属性，即都是项目级服务或都是全局级服务。多个授权语句策略描述如下：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "ecs:cloudServers:resize",
        "ecs:cloudServers:delete",
        "ecs:cloudServers:delete",
        "css:cluster:restart",
        "css:*:get*",
        "css:*:list*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

1.2 示例：为 IAM 用户（子用户）配置创建 CSS 集群的权限

如果您需要对所拥有的云搜索服务CSS进行细颗粒度的权限管理，可以使用统一身份认证服务IAM创建独立的IAM用户（子用户）并给IAM用户组授予策略或角色，便可使用这些策略来控制用户对CSS资源的访问范围。

本章节为您详细介绍如何创建IAM用户，并将IAM用户添加至用户组中，从而使IAM用户拥有创建CSS集群的权限。

- 创建用户组**：A公司的团队分为管理组（admin）和开发人员组。由于系统默认内置了admin组，用于拥有账号所有资源的使用及管理权限，因此A公司的团队只需要在IAM中再创建开发人员组即可。
- 给用户组授权**：A公司的开发人员需要使用的云服务为CSS和OBS，需要为“开发人员组”授予这几个服务的管理员权限。完成用户组的授权后，用户组中的用户才可以使用这些云服务。
- 创建IAM用户**：A公司使用已注册的管理员账号，给公司成员创建IAM用户并加入用户组，使得他们拥有独立的用户和密码，可以独立登录云服务平台并使用权限范围内的资源。
- IAM用户登录并验证权限**：A公司在名为“A-Company”的账号中创建了名为James、Alice、Charlie、Jackson和Emily的IAM用户。完成IAM用户创建后，A公司管理员需要将账号名、IAM用户名及初始密码告知对应的员工，这些员工就可以使用自己的用户名及密码访问云服务平台，并在云搜索服务控制台创建集群验证自己的权限。

创建用户组

步骤1 A公司管理员登录并进入华为云控制台。

步骤2 在“控制台”页面，鼠标移动至右上方的用户名，在下拉列表中选择“统一身份认证”。

步骤3 在统一身份认证服务，左侧导航窗格中，单击“用户组” > “创建用户组”。

图 1-2 创建用户组



步骤4 在“创建用户组”界面，输入“用户组名称”，单击“确定”，完成用户组创建。

图 1-3 输入用户组信息

用户组 / 创建用户组

* 用户组名称 开发人员组

描述 请输入用户组信息。 0/255

确定 取消

----结束

给用户组授权

步骤1 在用户组列表中，单击新建用户组“开发人员组”操作列的“授权”“权限配置”。

步骤2 在用户组“选择策略”页面中，在搜索框中搜索“CSS FullAccess”和“OBS Administrator”，勾选并单击“下一步”。

- 创建CSS集群的权限有“CSS FullAccess”或“Elasticsearch Administrator”。
- 如果还需要查看资源的消费情况，请在同区域选择“BSS Administrator”权限。

步骤3 设置最小授权范围。建议权限范围方案选择“指定区域项目资源”，使用哪个区域的资源就选哪个区域。

此处以仅为“华北-北京四”区域的资源设置权限为例，选择授权范围方案为“指定区域项目资源”，并选择“cn-north-4 [华北-北京四]”区域。

图 1-4 设置最小授权范围

① 选择策略 — ② 设置最小授权范围 — ③ 完成

根据当前您所选的策略，系统推荐以下授权范围方案。建议您勾选最小化授权，可进行选择。了解如何根据您的应用场景选择合适的授权范围方案

选择授权范围方案

所有资源

指定区域项目资源

授权后，用户根据权限使用已选区域项目中的资源。

共10个项目，请选择您想关联的区域项目

项目所属区域	描述
<input type="checkbox"/> cn-north-1 [华北-北京一]	--
<input checked="" type="checkbox"/> cn-north-4 [华北-北京四]	--

步骤4 单击“确定”，完成用户组授权。

----结束

创建 IAM 用户

步骤1 在统一身份认证服务，左侧导航中，单击“用户”>“创建用户”。

步骤2 配置基本信息。在“创建用户”界面填写“用户信息”和“访问方式”。如需一次创建多个用户，可以单击“添加用户”进行批量创建，每次最多可创建10个用户。

图 1-5 配置用户信息

The screenshot shows the 'Create User' interface with the 'Configure User Basic Information' step selected. The form includes the following fields and options:

- 用户名 (必填):** Input field for the user's name.
- 邮件地址 (选填):** Input field for the user's email address.
- 手机号 (选填):** Input field for the user's mobile number, with a dropdown menu for the country code (currently set to +86).
- 描述 (选填):** Input field for a description of the user.
- 外部身份ID (选填):** Input field for an external ID.
- 删除:** A button to delete the user.

At the bottom, there is a '添加用户' (Add User) button and a note: '您本次还可以创建8个用户。' (You can still create 8 users this time.)

说明

- 用户可以使用此处设置的用户名、邮件地址或手机号任意一种方式登录华为云。
- 当用户忘记密码时，可以通过此处绑定的邮箱或手机自行重置密码，如果用户没有绑定邮箱或手机号码，只能由管理员重置密码。

表 1-1 用户信息

用户信息	说明
用户名	必填。IAM用户登录华为云的用户名，此处以“James”和“Alice”为例。
邮件地址	“访问方式”选择“首次登录时设置”时必填，选择其他时选填。IAM用户绑定的邮件地址，可作为子账户的登录凭证，也可由IAM用户自己绑定。
手机号	选填。IAM用户绑定的手机号，可作为子账户的登录凭证，也可由IAM用户自己绑定。
描述	选填。记录IAM用户相关信息。

图 1-6 配置访问方式

* 访问方式

- 编程访问
启用访问密钥或密码，用户仅能通过API、CLI、SDK等开发工具访问华为云服务。
- 管理控制台访问
启用密码，用户仅能登录华为云管理控制台访问云服务。

凭证类型

- 访问密钥
创建用户成功后下载访问密钥。
- 密码

自定义

.....

首次登录时重置密码

自动生成
系统自动生成密码，创建用户完成后可下载。

首次登录时设置
系统通过邮件发一次性登录链接给用户，用户使用该链接登录管理控制台并设置密码。

* 登录保护

- 开启登录保护 (推荐)
- 不开启

- 编程访问：为IAM用户启用访问密钥或密码，支持用户通过API、CLI、SDK等开发工具访问华为云服务。
- 管理控制台访问：为IAM用户启用密码，支持用户登录华为云管理控制台访问云服务。

说明

- 如果IAM用户仅需登录管理控制台访问云服务，建议访问方式选择管理控制台访问，凭证类型为密码。
- 如果IAM用户仅需编程访问华为云服务，建议访问方式选择编程访问，凭证类型为访问密钥。
- 如果IAM用户需要使用密码作为编程访问的凭证（部分API要求），建议访问方式选择编程访问，凭证类型为密码。
- 如果IAM用户使用部分云服务时，需要在其控制台验证访问密钥（由IAM用户输入），建议访问方式选择编程访问和管理控制台访问，凭证类型为密码和访问密钥。例如IAM用户在控制台使用云数据迁移CDM服务创建数据迁移，需要通过访问密钥进行身份验证。

表 1-2 配置凭证类型和登录保护

凭证类型与登录保护		说明
访问密钥		创建用户完成后即可下载本次创建的所有用户的访问密钥（AK/SK）。 一个用户最多拥有两个访问密钥。
密码	自定义	自定义用户密码，并选择用户首次登录时是否需要重置密码。 如果您是用户的使用主体，建议您选择该方式，设置自己的登录密码，且无需勾选首次登录时重置密码。

凭证类型与登录保护		说明
	自动生成	系统自动生成IAM用户的登录密码，创建完用户即可下载 excel形式的密码文件。将密码文件提供给用户，用户使用该密码登录。 仅在创建单个用户时适用。
	首次登录时设置	系统通过邮件发一次性登录链接给用户，用户登录控制台并设置密码。 如果您不是用户的使用主体，建议选择该方式，同时输入用户的邮件地址和手机，用户通过邮件中的一次性链接登录华为云，自行设置密码。该链接 7天内 有效。
登录保护	开启登录保护（推荐）	开启登录保护后，IAM用户登录时，除了在登录页面输入用户名和密码外（第一次身份验证），还需要在登录验证页面输入验证码（第二次身份验证），该功能是一种安全实践，建议开启登录保护，多次身份认证可以提高账号安全性。 您可以选择通过手机、邮箱、虚拟MFA进行登录验证。
	不开启	创建完成后，如需开启登录保护，请参见： 登录保护 。

步骤3 单击“下一步”，将用户加入到用户组（可选）。

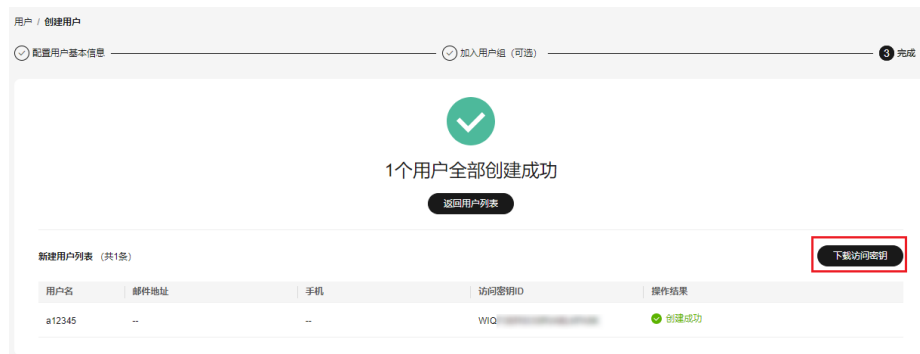
- 将用户加入用户组，用户将具备用户组的权限。
- 如需创建新的用户组，可单击“创建用户组”，填写用户组名称和描述（可选），创建成功后即可将用户加入到新创建的用户组中。

说明

- 如果该用户是管理员，可以将用户加入默认用户组“admin”中。
- 一个用户可以同时加入多个用户组。

步骤4 单击“下一步”，IAM用户创建完成，用户列表中显示新创建的IAM用户。如果勾选了“编程访问”，可在此页面下载访问密钥。

图 1-7 创建成功



----结束

IAM 用户登录并验证权限

步骤1 在登录页面，单击登录下方的“IAM用户”“IAM用户登录”“子用户登录”，在“IAM用户登录”页面，输入“租户名/原华为云账号名”、“IAM用户名/邮件地址”和“密码”“账号名”、“用户名/邮箱”和“密码”。

图 1-8 IAM 用户登录



- 租户名/原华为云账号名：IAM用户所属的账号。
- IAM用户名/邮件地址：在IAM创建用户时，输入的IAM用户名/邮件地址，例如“James”。如果不知道用户名及初始密码，请向管理员获取。
- IAM用户密码：IAM用户的密码，非账号密码。

步骤2 单击“登录”，登录华为云。

步骤3 在“服务列表”中选择云搜索服务，进入云搜索服务CSS控制台。

步骤4 在云搜索服务总览页面右上角，单击“创建集群”按钮，按照[创建集群](#)的步骤创建集群，如果创建成功，则表示权限配置成功。

----结束

2 使用 Elasticsearch 搜索数据

2.1 Elasticsearch 使用流程

表 2-1 Elasticsearch 集群的使用流程

任务分类	操作	参考文档
使用	规划集群	在创建Elasticsearch集群前，需要先完成集群规划。规划时，应考虑是否多可用区部署以提高集群的高可用性，合理配置集群的节点类型与节点存储规格，以及根据业务需求选择适当的集群版本和安全模式，同时注意索引分片的优化，以确保集群的稳定性和性能。 Elasticsearch集群规划建议
	创建集群	基于集群规划，创建合适的Elasticsearch集群。 创建Elasticsearch集群
	访问集群	Elasticsearch集群提供多样化的访问方式，包括Kibana、Cerebro、开源API、Java、Python和Go客户端，以及通过内网和公网的多种网络配置，用户可根据编程语言偏好和网络环境选择最合适的接入方法。 Elasticsearch集群访问方式
	导入数据	Elasticsearch集群提供多种数据导入方式，包括Logstash、开源Elasticsearch API、云数据迁移服务CDM和数据复制服务DRS，支持不同来源和格式的数据导入，以及针对关系型数据库的实时同步，用户可根据业务场景和数据特性选择最佳方法。 Elasticsearch集群导入数据方式

任务分类	操作	参考文档
	搜索数据	<p>CSS服务推荐使用DSL语言在Elasticsearch集群中搜索数据，同时也支持使用SQL语言进行数据搜索。</p> <p>使用DSL语言在Elasticsearch中搜索数据 使用SQL语言在Elasticsearch中搜索数据</p>
	增强集群搜索能力	<p>CSS服务的Elasticsearch集群基于社区版本提供了一系列增强特性，包括向量检索、存算分离、流量控制、大查询隔离、聚合增强、读写分离、冷热数据切换、索引回收站等，以适应不同场景下的性能和成本优化需求，增强集群稳定性和搜索能力。</p> <p>Elasticsearch集群搜索增强特性介绍</p>
运维	备份与恢复	<p>备份与恢复是通过集群快照备份Elasticsearch集群数据，在数据丢失或需要历史数据时快速恢复数据，提升集群的数据可靠性。</p> <p>创建快照备份Elasticsearch集群数据 恢复Elasticsearch集群数据</p>
	扩缩容	<p>CSS服务提供灵活的扩容和缩容功能，支持增加节点数量、升级节点规格、增加节点类型，以及随机或指定缩容节点，以动态调整集群资源，适应不同业务需求并优化成本。</p> <p>扩容Elasticsearch集群 缩容Elasticsearch集群</p>
	升级	<p>Elasticsearch集群支持同版本升级、跨版本升级和跨引擎升级。同版本升级是升级集群的内核补丁，用于修复问题或优化性能；跨版本升级是升级集群的版本，用于功能加强或版本收编；跨引擎升级是将Elasticsearch集群升级至OpenSearch集群。</p> <p>升级Elasticsearch集群版本</p>

任务分类	操作	参考文档
	<p>管理集群</p> <p>CSS服务提供全面的集群管理功能，包括集群信息查看、集群用户授权、标签管理、安全模式更改、节点替换、企业项目绑定、可用区切换、自定义词库配置等，旨在帮助用户高效、灵活地管理Elasticsearch集群，确保集群的安全性、高可用性和性能优化。</p>	<p>查看Elasticsearch集群信息</p> <p>创建Elasticsearch集群用户并授权使用集群</p> <p>设置Elasticsearch集群标签</p> <p>配置Elasticsearch集群YML文件默认参数</p> <p>为Elasticsearch集群绑定企业项目</p> <p>替换Elasticsearch集群指定节点</p> <p>更改Elasticsearch集群安全模式</p> <p>切换Elasticsearch集群可用区</p> <p>配置和使用Elasticsearch集群的自定义词库</p> <p>在Elasticsearch集群实现搜索数据的简繁体转换</p> <p>重启Elasticsearch集群</p> <p>删除Elasticsearch集群</p>
	<p>管理集群索引策略</p> <p>使用Elasticsearch的ISM (Index State Management) 插件创建和管理索引的生命周期策略，包括策略的创建、索引关联、以及策略的管理与变更，旨在优化集群性能和降低存储成本。</p>	<p>创建和管理Elasticsearch索引策略</p>
	<p>监控与日志管理</p> <p>CSS服务提供全面的监控和日志管理功能，包括集群和节点监控指标、告警规则配置、日志备份与采集，以及智能运维工具，帮助用户有效监控、分析和维护Elasticsearch集群，确保集群的稳定性和性能。</p>	<p>Elasticsearch集群支持的监控指标</p> <p>配置Elasticsearch集群监控</p> <p>设置Elasticsearch集群SMN告警通知</p> <p>智能检测Elasticsearch集群风险</p> <p>查询和管理Elasticsearch集群日志</p>
	<p>审计日志</p> <p>通过云审计服务，用户可以记录和查询Elasticsearch集群的关键操作事件，且操作记录在管理控制台保存最近7天，便于后续的审计和回溯。</p>	<p>查看Elasticsearch集群审计日志</p>

2.2 Elasticsearch 集群规划建议

在创建Elasticsearch集群前，需要先完成集群规划。规划时，应考虑是否多可用区部署以提高集群的高可用性，合理配置集群的节点类型与节点存储规格，以及根据业务需求选择适当的集群版本和安全模式，同时注意索引分片的优化，以确保集群的稳定性和性能。

规划集群可用区

为防止数据丢失，并确保在服务中断情况下能降低集群的停机时间，从而增强集群的高可用性，CSS服务支持跨可用区（即多可用区）部署。用户可以在同一个区域内选择两个或三个不同的可用区进行集群部署。

在创建集群时，如果用户选择了两个或三个可用区，CSS服务将自动开启跨AZ的高可用性特性，确保节点在这些可用区内均匀分配。系统均匀分配的节点满足各个AZ之间节点数量的差小于等于1，具体的节点分布情况，包括不同数量的节点如何在各个可用区中分布，可以参考[表2-2](#)。

须知

- 在创建集群时，选择的任意类型的节点数量都要大于等于所选的AZ数量，否则跨可用区部署会失败。
- 部署跨AZ集群时，任意类型的节点都会被均匀的分布在不同的AZ上。

表 2-2 节点数量和 AZ 分布

集群节点个数	单AZ	两AZ		三AZ		
	AZ1	AZ1	AZ2	AZ1	AZ2	AZ3
1个节点	1	不支持		不支持		
2个节点	2	1	1	不支持		
3个节点	3	2	1	1	1	1
4个节点	4	2	2	2	1	1
...

在选择多可用区部署时，建议合理配置副本数量，以更高效地利用跨可用区的高可用性特性。

- 在跨两个可用区的部署中，当其中一个AZ不可用时，剩下的AZ需要继续提供服务，因此索引的副本个数至少为1个。由于Elasticsearch默认副本数为1个，因此如果您对读性能没有特殊要求，可以直接使用默认值。
- 在跨三个可用区部署中，为了保证其中任意一个AZ不可用时，剩余的AZ可以继续提供服务，因此索引的副本数至少要为1个。为了提高集群的查询能力，也可以设置更多的副本。由于Elasticsearch默认的副本数为1个，因此需要用户修改setting配置来实现修改索引副本个数。

可以通过如下命令修改索引的副本个数，如：

```
curl -XPUT http://ip:9200/{index_name}/_settings -d
 '{"number_of_replicas":2}'
```

也可以通过在模板中指定所有索引的副本个数，如：

```
curl -XPUT http://ip:9200/_template/templatename -d '{ "template": "*",
 "settings": {"number_of_replicas": 2}}'
```

其中，“ip”表示集群内网访问地址，“index_name”表示索引名称，“number_of_replicas”表示修改后的索引副本个数，此处以修改为2个索引副本为例。

当采用多可用区部署时，如果某个可用区发生故障，相关的业务故障行为分析及应对策略请参见表2-3。

表 2-3 AZ 故障的业务故障行为分析

可用区数量	主节点个数	业务中断行为及应对建议
2	0	<ul style="list-style-type: none">● 如果节点个数为2的倍数：一半的数据节点故障，需要替换故障可用区中的一个节点，才能继续选择主节点。● 如果节点数为奇数：<ul style="list-style-type: none">- 故障AZ含多一个节点，需要替换故障可用区中一个节点，才能继续选择主节点。相关替换请联系技术支持。- 故障AZ含少一个节点，不中断业务，能够继续选择主节点。
2	3	有50%机会的停机时间。当两个专用主节点分配到一个可用区中，一个主节点分配到另一个可用区中时： <ul style="list-style-type: none">● 如果具有一个专用主节点的可用区遇到中断，则剩余可用区具有两个专用主节点，这两个专用主节点可以选择出主节点。● 如果具有两个专用主节点的可用区遇到中断，剩余可用区只有一个专用主节点，无法选择出主节点，业务中断，需要联系技术支持。
3	0	当您选择3个可用区，节点个数为4，三个可用区的节点分布数为2，1，1，如果节点个数为2的可用区故障，那么此时业务中断，建议您选择三个可用区时避免选择4个节点。 一般不会出现业务中断时间。
3	3	无业务中断时间。

说明

当集群创建完成后，支持切换可用区，具体操作请参见[切换Elasticsearch集群可用区](#)。

切换可用区包含两大场景：可用区高可用改造和可用区平移切换。

- 可用区高可用改造：适用于单AZ改造成两AZ、单AZ改造成三AZ或两AZ改造成三AZ的场景，目的是为了提升集群的高可用性。
- 可用区平移切换：适用于从一个AZ完全迁移到另一个AZ的场景，是为了解决当前可用区资源不足的问题。

规划集群版本

选择Elasticsearch集群版本时，建议综合考虑业务需求、特性支持、性能改进、安全性更新和长期支持等因素，以确保选择的版本能够满足当前和未来的业务发展，同时提供稳定和安全的运行环境。

- 当首次使用CSS服务的Elasticsearch集群时，建议选择最新版本。
- 当需要将自建或第三方Elasticsearch集群迁移到CSS服务，且仅迁移集群不改造集群时，建议版本号和源集群一致。
- 当需要将自建或第三方Elasticsearch集群迁移到CSS服务，且需要对集群进行代码改造时，建议选择7.10.2或7.6.2版本。

表 2-4 集群版本支持情况

特性	Elasticsearch 7.6.2	Elasticsearch 7.10.2	相关文档
向量检索	√	√	配置Elasticsearch集群向量检索
存算分离	√	√	配置Elasticsearch集群存算分离
流量控制2.0	√	√	配置Elasticsearch集群读写流量控制策略2.0
流量控制1.0	√	√	配置Elasticsearch集群读写流量控制策略1.0
大查询隔离	√	√	配置Elasticsearch集群大查询隔离
聚合增强	x	√	配置Elasticsearch集群聚合增强
读写分离	√	√	配置Elasticsearch集群读写分离
切换冷热数据	√	√	切换Elasticsearch集群冷热数据
索引回收站	x	√	配置Elasticsearch集群索引回收站
导入性能增强	x	√	增强Elasticsearch集群数据导入性能

特性	Elasticsearch 7.6.2	Elasticsearch 7.10.2	相关文档
集群内核监控增强	√	√	配置Elasticsearch集群内核监控
索引监控	√	√	配置Elasticsearch集群索引监控

规划节点类型

在Elasticsearch集群中，合理规划不同节点类型对于优化性能和资源利用率至关重要。在创建集群时，应根据业务需求、查询负载、数据增长模式和性能目标来确定添加哪些类型的节点，以实现合适的集群性能和资源管理。[表2-5](#)是介绍了不同节点类型的适用场景，建议用户根据具体的业务需求和性能预期来选择是否启用该类节点。

说明

- 如果创建集群时未启用Master节点或Client节点，当业务运行一段时间后，发现数据节点压力太大时，支持单独添加Master节点或Client节点，具体操作请参见[添加Master或Client节点](#)。
- 如果创建集群时未启用冷数据节点，则集群创建完成后不支持单独添加冷数据节点，请在创建集群时合理选择是否启用冷数据节点。

表 2-5 各类节点的适用场景

节点类型	节点功能描述	适用场景
数据节点 (ess)	数据节点用于存储数据，当集群没有Master节点和Client节点时，数据节点会同时兼顾这两类节点的功能。	集群必配的节点类型。 <ul style="list-style-type: none">当集群未启用Master节点和Client节点时，数据节点将同时承担集群管理、存储数据、提供接入集群和分析数据的职责。此时，为保证集群中数据的稳定性，建议设置节点数量大于等于3个。当集群启用了Master节点但未启用Client节点时，数据节点将用于存储数据并提供接入集群和分析数据的功能。当集群未启用Master节点但启用了Client节点时，数据节点将用于存储数据并提供集群管理功能。当集群同时启用了Master节点和Client节点时，数据节点将仅用于存储数据。

节点类型	节点功能描述	适用场景
Master节点 (ess-master)	Master节点负责管理集群中所有节点任务，如元数据管理、索引创建与删除、分片分配等。在大规模集群的元数据管理、节点管理、稳定性保障和集群操作控制中发挥着至关重要的作用。	<ul style="list-style-type: none"> ● 集群规模较大：当集群的节点数超过16个时，为了更有效地管理集群状态和元数据，建议添加专用的Master节点。 ● 拥有高索引和分片数量：如果索引数量或分片数超过1万个，Master节点可以处理更复杂的集群管理任务，避免对数据节点的性能造成影响。 ● 单独管理集群节点：Master节点负责维护集群的元数据，包括索引映射、设置和别名等，对于复杂的集群结构，专用的Master节点可以提供更好的管理。包括节点加入、退出以及故障检测等，Master节点在集群节点管理中扮演核心角色。 ● 提升集群稳定性和可靠性：专用的master节点可以提高集群的稳定性和可靠性，因为它减少了对同时承担数据存储和查询任务的节点的依赖。 ● 优化数据节点性能：通过将集群管理任务从数据节点分离到Master节点，可以优化数据节点的性能，使其专注于数据操作。
Client节点 (ess-client)	Client节点负责接收并协调外部请求，如search和write请求，在处理高负载查询、复杂聚合、大量分片管理以及优化集群扩展性方面发挥着重要作用。	<ul style="list-style-type: none"> ● 存在高查询QPS：当集群面临高查询每秒查询率（QPS）时，独立的Client节点可以更均匀地分发查询请求，减轻数据节点的负担，提高整体查询性能。 ● 存在复杂的聚合查询：对于需要大量计算资源的复杂聚合查询，Client节点可以专门处理聚合结果，从而提升聚合查询的效率和响应速度。 ● 集群分片数量多：在分片数量较多的集群中，Client节点可以有效地协调和管理对各个分片的查询请求，提高请求的转发和处理效率。 ● 可以减轻数据节点压力：Client节点负责解析search请求，确定索引分片的位置，并协调分片节点执行查询。可以减轻数据节点的负载，使它们更专注于数据的存储和索引。 ● 提高集群扩展性：增加client节点可以提供更好的集群扩展性和灵活性，支持更大规模的数据集和更复杂的查询需求。

节点类型	节点功能描述	适用场景
冷数据节点 (ess-cold)	冷数据节点用于存储对查询时延要求不高, 但数据量较大的历史数据, 是管理大规模数据集和优化存储成本的有效方式。	<ul style="list-style-type: none"> ● 需要存储大量历史数据: 当需要存储大量不常访问但对分析有用的历史数据时, 使用冷数据节点可以提供成本效益较高的存储解决方案。 ● 需要优化热数据性能: 通过将旧数据迁移到冷数据节点, 可以减少对热数据节点的存储压力, 从而优化热数据的查询和写入性能。 ● 对查询时延要求不高: 对于那些查询频率不高且可以容忍较高查询时延的数据, 冷数据节点是合适的存储选择。 ● 追求成本效益: 冷数据节点通常采用具有较大磁盘容量的规格, 这有助于降低存储成本, 同时满足大规模数据存储需求。

规划节点存储

- **规划节点机型**

CSS服务支持多种节点机型, 每种机型适用于不同的业务场景, 建议用户根据具体的业务需求和性能预期来选择最合适的机型, 以优化存储性能和成本效益。

表 2-6 节点机型适用场景

节点机型	磁盘类型	规格说明	适用场景
计算密集型	云盘	CPU:内存=1:2	推荐场景 , 用于数据量较少 (单节点 <100GB) 的搜索场景。
通用计算型	云盘	CPU:内存=1:4	通用场景 , 用于单节点数据量在 100-1000GB间的搜索与分析场景, 例如中等规模的电商搜索、社交搜索、日志搜索等场景。
内存优化型	云盘	CPU:内存=1:8	通用场景 , 用于单节点数据量在 100-2000GB间的搜索与分析场景。 适合向量检索场景 , 大内存有利于提升集群的性能与稳定性。
磁盘增强型	本地盘	挂载HDD盘	适合日志场景 , 用于存储冷数据, 对冷数据的数据查询性能要求低, 并且数据需要更新的场景。
超高I/O型 (CPU架构为鲲鹏计算)	本地盘	挂载SSD盘	适用大型日志场景 , 用于存储热数据。

节点机型	磁盘类型	规格说明	适用场景
超高I/O型 (CPU架构为X86计算)	本地盘	挂载SSD盘	适用大型搜索与分析场景, 场景对计算或磁盘I/O均有较高要求, 例如舆情分析、专利检索、以及部分数据库加速场景。

● 规划节点规格

在规划节点规格时, 推荐优先考虑高配置但节点数量较少的方案。例如, 一个由3个节点组成的集群, 每个节点配置为32核CPU和64GB内存, 通常比一个由12个节点组成的集群, 每个节点配置为8核CPU和16GB内存, 在集群的稳定性和扩展性方面更具优势。

优势主要体现在如下方面。

- 集群稳定性: 高配置节点通常能提供更强的处理能力和更大的内存空间, 从而提高集群的整体稳定性。
- 扩容便捷性: 当高配置集群遇到性能瓶颈时, 可以通过横向扩展轻松解决, 即简单地向集群中添加更多具有相同高配置的节点。这种扩展方式简单直接, 易于实施。
- 维护简便: 较少的节点数量意味着更少的维护工作和更低的管理复杂性。

相比之下, 低配置集群在需要扩容时, 往往需要进行纵向扩展, 即提升单个节点的配置。这不仅可能涉及更复杂的迁移和升级过程, 还可能增加额外的维护成本和技术挑战。

因此, 在规划集群时, 应综合考虑性能、成本、维护和扩展性, 选择最适合业务需求的节点规格。

● 规划存储容量

在规划CSS集群的存储容量时, 应考虑数据量、副本因子、数据膨胀率和磁盘使用率等多个关键因素。以下是一个推荐的计算方法, 用以确定所需的集群存储容量。

存储容量=源数据×(1+副本数量)×(1+数据膨胀率)×(1+预留空间比例)

- 源数据: 首先确定预期存储的原始数据量。
- 副本数量: 设置副本因子, 默认建议值为1, 以保证数据的高可用性。
- 数据膨胀率: 集群在索引过程中可能会产生额外的数据膨胀, 通常建议按照25%的膨胀率进行计算。
- 磁盘空间使用率: 考虑到操作系统和文件系统本身占用的空间, 以及留出一定的空间以优化磁盘性能和冗余, 建议将磁盘使用率控制在70%, 即预留空间比例为30%。

将具体数值代入公式: 存储容量=源数据×2×1.25×1.3

简化计算, 如果源数据量已知, 最终的存储容量大约是源数据的3.25倍。这个计算方法提供了一个基础的估算, 但实际配置时还需要根据具体业务场景和增长预期进行调整。

规划节点数量

创建集群时, 集群的节点数量应当基于业务性能需求和预期负载进行规划。[表2-7](#)提供了计算方式用以确定合适的节点数量。通过这个计算方式可以更科学地规划集群的节点数量, 以满足业务需求并保证集群的性能和稳定性。

表 2-7 节点数量的计算方式

节点	性能基线	节点数量计算方式	示例
写入节点	<ul style="list-style-type: none"> 对于挂载云盘的节点，其单核写入性能基线为1MB/s。 对于超高IO型的节点，其单核写入性能基线为1.5MB/s。 	写入节点数=业务峰值时的流量÷单节点的核数÷单核写入性能基线×副本数	业务峰值写入100MB/s，使用16u64g的节点，预计需要 $100 \div 16 \div 1 \times 2 = 12$ 个节点。
查询节点	相同节点，不同业务场景下的性能差异非常大，单节点的性能基线难以评估。这里以业务 平均查询响应时间 （单位为秒）作为查询的性能基线进行测算。	查询节点数=QPS÷（单节点的核数×3÷2÷平均查询响应时间）×分片数量	查询QPS要求1000，平均查询响应时间100ms，索引规划3个分片，使用16u64g的节点，预计需要 $1000 \div (16 \times 3 \div 2 \div 0.1) \times 3 = 12$ 个节点。
总节点数量	不涉及	总节点数量=写入节点数+查询节点数	总节点数=写入节点数+查询节点数=24个节点数。
说明 这里计算的总节点数量表示数据节点和冷数据节点的数量之和。			

当一个集群包含的节点类型不同时，各节点类型支持的节点数量不同，设置节点数量时可以参考[表2-8](#)。

表 2-8 集群的节点数量取值说明

一个集群包含的节点类型	节点数量的取值范围
ess	ess: 1~32
ess、ess-master	ess: 1~200 ess-master: 3~9的奇数
ess、ess-client	ess: 1~32 ess-client: 1~32
ess、ess-cold	ess: 1~32 ess-cold: 1~32
ess、ess-master、ess-client	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32
ess、ess-master、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-cold: 1~32

一个集群包含的节点类型	节点数量的取值范围
ess、ess-client、ess-cold	ess: 1~32 ess-client: 1~32 ess-cold: 1~32
ess、ess-master、ess-client、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32 ess-cold: 1~32
说明 <ul style="list-style-type: none">• ess: 数据节点，即创建集群时必配的节点类型，其他3种节点类型都是基于业务需要可选的类型。• ess-master: Master节点。• ess-client: Client节点。• ess-cold: 冷数据节点。	

规划虚拟私有云和子网

CSS服务支持在非共享VPC和共享VPC内创建集群。

共享VPC和非共享VPC相比，有如下优势：

- 在一个账号内统一创建资源，并将资源共享给其他账号，其他账号无需创建重复资源，可以精简资源数量以及网络架构，提升管理效率并节约成本。
比如不同账号下的VPC网络互通需要建立对等连接，使用共享VPC后，则不同账号用户可以在同一个VPC创建资源，免去了对等连接配置，有效的简化组网结构。
- 在一个账号内统一管理运维资源，便于企业集中配置业务安全策略，并且利于对资源使用情况的监控和审计，支撑业务的安全诉求。

当选择使用共享VPC创建集群时，需要提前完成共享VPC子网的创建，操作流程请参见[表2-9](#)。有关共享VPC的使用限制及详细操作指导等信息可以参考[共享VPC概述](#)。

表 2-9 共享子网创建流程说明

方法	说明	操作指导
方法A	<ol style="list-style-type: none">通过RAM管理控制台，所有者创建共享，将子网共享给使用者。配置如下：<ol style="list-style-type: none">选择共享子网。为共享子网选择权限，即指定使用者对该共享子网具备的权限。 CSS服务在共享VPC中创建集群需要选择权限：default vpc subnet statement。指定共享子网的使用者，可以指定多个。共享创建完成后，通过RAM管理控制台，使用者可以选择接受或者拒绝共享申请。<ul style="list-style-type: none">使用者接受共享申请，子网共享成功。如果后续使用者不再需要使用该共享子网，可以退出该共享。使用者拒绝共享申请，子网共享失败。	<ol style="list-style-type: none">创建共享接受/拒绝共享邀请 退出共享
方法B	<ol style="list-style-type: none">通过RAM管理控制台，所有者创建共享，将子网共享给使用者。配置如下：<ol style="list-style-type: none">选择共享子网。为共享子网选择权限，即指定使用者对该共享子网具备的权限。 CSS服务在共享VPC中创建集群需要选择权限：default vpc subnet statement。指定共享子网的使用者，可以指定多个。通过VPC管理控制台，选择上一步中已创建的共享，将子网加入到该共享内。共享创建完成后，通过RAM管理控制台，使用者可以选择接受或者拒绝共享申请。<ul style="list-style-type: none">使用者接受共享申请，子网共享成功。如果后续使用者不再需要使用该共享子网，可以退出该共享。使用者拒绝共享申请，子网共享失败。	<ol style="list-style-type: none">创建共享将VPC子网共享给其他账号接受/拒绝共享邀请 退出共享

规划集群安全模式

表 2-10 集群类型介绍

集群类型		集群描述	适用场景
非安全集群	非安全模式的集群	非安全模式的集群无需安全认证即可访问，采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	适合内网业务，用于测试场景。 <ul style="list-style-type: none">• 优点：简单，接入集群容易。• 缺点：安全性差，任何人都可以访问集群。
安全集群	安全模式+HTTP协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	可以实现用户权限隔离，适用于对集群性能敏感的场景。 <ul style="list-style-type: none">• 优点：访问集群需要安全认证，提升了集群安全性。通过HTTP协议访问集群又能保留集群的高性能。• 缺点：无法公网访问集群。
	安全模式+HTTPS协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTPS协议进行通信加密，使数据更安全。	有非常高的安全要求，且需要公网访问集群的场景。 <ul style="list-style-type: none">• 优点：访问集群需要安全认证，提升了集群安全性，且HTTPS协议的通讯加密可以实现集群公网访问功能。• 缺点：通过HTTPS协议访问集群，集群的读取性能相对HTTP协议来说，会下降20%左右。

当访问安全模式的集群时，需要输入用户名和密码通过安全认证才能访问。CSS服务支持以下两类用户的安全认证：

- 集群的管理员：管理员账户名默认为**admin**，密码为创建集群时设置的管理员密码。
- 集群的用户：集群的管理员通过Kibana创建集群的用户和密码。创建方式请参见[创建Elasticsearch集群用户并授权使用集群](#)。

说明

当集群创建完成后，支持切换安全模式，具体操作请参见[更改Elasticsearch集群安全模式](#)。

切换安全模式包含三大场景：非安全模式切换为安全模式、安全模式切换为非安全模式、安全模式的协议切换。

规划索引分片数

在使用集群的过程时，特别是在进行数据导入操作之前，建议根据具体的业务需求，提前对集群的数据结构和分布进行规划。这包括合理设计索引和确定分片数量。为了确保集群在性能和可扩展性方面达到最佳状态，以下是一些建议。

- **单个分片大小**：建议将每个分片的大小控制在10GB到50GB之间。这有助于在存储效率和查询性能之间取得平衡。
- **集群总分片数量**：为了管理方便和避免过度扩展，建议将集群的总分片数量控制在3万以内。这有助于保持集群的稳定性和响应速度。
- **内存与分片比例**：在资源分配上，建议每1GB的内存空间放置20到30个分片。这样可以确保每个分片都有足够的内存资源进行索引和查询操作。
- **单节点分片数**：为了避免单点过载，建议每个节点上的分片数量不超过1000个。这有助于避免节点资源竞争，确保节点的稳定运行。
- **索引分片与节点数的关系**：对于单个索引，建议其分片数与集群的节点数保持一致，或者设置为节点数的整数倍。这有助于实现负载均衡，优化查询和索引的性能。

通过以上建议，可以更有效地规划和管理CSS集群的索引分片，从而提升集群的整体性能和可维护性。

2.3 创建 Elasticsearch 集群

本文介绍如何创建Elasticsearch集群。

场景描述

当创建的集群类型不同时，需要关注如[表2-11](#)所示的关键参数的配置。

表 2-11 集群创建差异

集群类型	“安全模式”	“HTTPS访问”	“公网访问”	“Kibana公网访问”
非安全模式的集群	关闭	不涉及	不支持启用	不支持启用
安全模式+HTTP协议的集群	开启	关闭	不支持启用	支持启用
安全模式+HTTPS协议的集群	开启	开启	支持启用	支持启用

前提条件

已经参考[Elasticsearch集群规划建议](#)完成待创建的Elasticsearch集群规划。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在总览页面单击右上角的“创建集群”，进入“创建集群”页面。

或者左侧导航栏单击“集群管理 > Elasticsearch”，单击右上角的“创建集群”，进入“创建集群”页面。

3. 在“基础配置”页面，完成Elasticsearch集群的基本信息和资源配置。

表 2-12 Elasticsearch 集群的基础配置

参数	说明
计费模式	集群支持包年/包月和按需计费两种模式。 <ul style="list-style-type: none">● 包年/包月：根据集群购买时长，一次性支付集群费用。最短时长为1个月，最长时长为3年。如果购买时长超过9个月，建议包年购买，价格更优惠。一年计费为购买10个月得12个月。● 按需计费：按实际使用时长计费，计费周期为一小时，不足一小时按一小时计费。
订购周期	选择包年/包月模式后，需要选择购买时长。 您可以根据需求，选择是否需要“自动续费”。
当前区域	选择集群的所在区域。 不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
可用区	选择集群工作区域下关联的可用区。 最多支持配置3个“可用区”，多可用区的使用建议请参见 规划集群可用区 。
集群类型	选择“Elasticsearch”。
集群版本	选择所需的集群版本，支持的版本以界面可选项为准。
集群名称	自定义集群名称，可输入的字符范围为4~32个字符，只能包含数字、字母、中划线和下划线，且必须以字母开头。
节点数量	集群中的数据节点个数。可选节点数为1~32，建议节点数为3或3以上，以提升集群可用性。 <ul style="list-style-type: none">● 当集群未启用Master节点和Client节点时，数据节点将同时承担集群管理、存储数据、提供接入集群和分析数据的职责。此时，为保证集群中数据的稳定性，建议设置节点数量大于等于3个。● 当集群启用了Master节点但未启用Client节点时，数据节点将用于存储数据并提供接入集群和分析数据的功能。● 当集群未启用Master节点但启用了Client节点时，数据节点将用于存储数据并提供集群管理功能。● 当集群同时启用了Master节点和Client节点时，数据节点将仅用于存储数据。
CPU架构	支持“X86计算”和“鲲鹏计算”两种类型。具体支持的类型由实际区域环境决定。

参数	说明
节点规格	集群中的节点规格。您可以根据需求，选择对应的规格。每个集群只能选择一个规格，规格的详细说明可参考 弹性云服务器的实例类型与规格 。
节点存储	选择存储类型，支持普通I/O、高I/O、超高I/O、极速型SSD。 说明 如果当前节点的存储类型不支持，则界面不显示。
节点存储容量	设置节点的存储空间大小，其取值范围与“节点规格”关联，不同的规格允许的取值范围不同。 节点存储容量只支持配置为20的倍数。 集群创建完成后，不支持扩容节点存储容量，请基于业务量合理选择容量。
启用Master节点	Master节点负责管理集群中所有节点任务，如元数据管理、索引创建与删除、分片分配等。在大规模集群的元数据管理、节点管理、稳定性保障和集群操作控制中发挥着至关重要的作用。 启用Master节点后，在下方选择对应的“节点规格”、“节点数量”和“节点存储”。“节点数量”必须是不小于3的奇数，最多设置9个节点。“节点存储”的存储容量为固定值，存储类型可以根据实际情况选择。
启用Client节点	Client节点负责接收并协调外部请求，如search和write请求，在处理高负载查询、复杂聚合、大量分片管理以及优化集群扩展性方面发挥着重要作用。 启用Client节点后，在下方选择对应的“节点规格”、“节点数量”和“节点存储”。“节点数量”可设置为1~32任意数值。“节点存储”的存储容量为固定值，存储类型可以根据实际情况选择。
启用冷数据节点	冷数据节点用于存储对查询时延要求不高，但数据量较大的历史数据，是管理大规模数据集和优化存储成本的有效方式。 启用冷数据节点后，在下方选择对应的“节点规格”、“节点数量”和“节点存储”。“节点数量”可设置为1~32任意数值。“节点存储”的存储类型和存储容量可以根据实际情况选择。 开启冷数据节点之后，支持切换集群的冷热数据，详情请参见 切换Elasticsearch集群冷热数据 。
企业项目	如果开通了“企业项目”，在创建集群时可以给集群绑定一个企业项目。 在下拉框中选择企业项目，单击“查看项目管理”跳转到“企业项目管理”管理控制台，查看已有的企业项目。

4. 单击“下一步：网络配置”。
5. 在“网络配置”页面，完成Elasticsearch集群的网络和安全模式配置。

表 2-13 Elasticsearch 集群的网络配置

参数	说明
虚拟私有云	<p>指定集群节点使用的虚拟专用网络，实现不同业务的网络隔离。</p> <p>单击“查看虚拟私有云”跳转到虚拟私有云列表，查看已创建或共享至当前账号下的VPC名称和ID。</p> <p>如果没有合适的VPC，建议联系CSS服务管理员新建VPC，具体请参见创建虚拟私有云和子网。</p> <p>说明 此处选择的VPC必须包含网段（CIDR），否则集群将无法创建成功。新建的VPC默认包含网段（CIDR）。</p>
子网	<p>集群使用子网实现与其他网络的隔离，并独享所有网络资源，以提高网络安全。</p> <p>选择当前虚拟私有云下集群需要的子网。支持选择共享VPC下的子网。</p>
安全组	<p>安全组起着虚拟防火墙的作用，为集群提供安全的网络访问控制策略。</p> <p>选择集群需要的安全组，单击“查看安全组”跳转到安全组列表，可以了解安全组详情。</p> <p>说明</p> <ul style="list-style-type: none">• 为了确保您能够正常访问集群，需要放通安全组9200规则。• 如果创建的集群为7.6.2及以上版本，则需要确保同安全组内节点之间的端口全放通。如果无法放通同安全组内节点之间的全部端口，请至少确保9300端口的通信。• 放开9300端口通信后，如果集群磁盘使用率较高，可清理过期数据，释放磁盘存储空间，请参考如何使用Elasticsearch清理过期数据，释放磁盘存储空间？。
安全模式	<p>选择是否开启集群安全模式。</p> <ul style="list-style-type: none">• 默认开启，则创建的是安全模式的集群。安全模式的集群会对集群进行通讯加密和安全认证。因此必须配置集群的“管理员账户名”和“管理员密码”。<ul style="list-style-type: none">- 管理员账户名默认为admin。- 设置并确认管理员密码。要记住设置的密码，后续访问集群需要输入密码。• 不开启，则创建的是非安全模式的集群。非安全模式的集群无需安全认证即可访问，并且采用HTTP明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。

参数	说明
HTTPS访问	<p>只有开启集群的安全模式才可以启用HTTPS访问，开启HTTPS访问后，访问集群将进行通讯加密。</p> <p>说明 安全集群使用HTTPS通信，相比非安全集群使用HTTP通信在读取性能上会降低，预期相对HTTP集群在大并发压力下有20%的性能劣化。如果想要读取性能快，又想要使用安全集群所提供的用户权限隔离资源（索引、文档、字段等）的功能，则可以关闭HTTPS访问。关闭HTTPS访问后，会使用HTTP协议与集群通信，无法保证数据安全性，并且无法开启公网访问功能。</p>
公网访问	<p>仅当集群开启了“安全模式”和“HTTPS访问”时，才可以选择是否配置“公网访问”。配置公网访问后，用户可以获得一个公网访问的IP地址，通过这个IP地址可以在公网访问该安全集群，详细配置请参考配置公网访问。</p>

6. 单击“下一步：高级配置”。
7. 在“高级配置”页面，完成Elasticsearch集群的快照和其他高级配置。
 - a. 设置集群快照。

系统默认打开集群快照开关，如果您不需要启用自动快照，可以在“集群快照开关”右侧关闭。自动快照会创建委托访问对象存储服务OBS，快照存储在OBS标准存储中需额外计费。

表 2-14 集群快照基础配置

参数	说明
OBS桶	<p>在下拉框中选择存储快照的OBS桶。也可以单击右侧的“创建桶”新建OBS。详细操作步骤请参见创建桶。</p> <p>创建或者已存在的OBS桶需满足如下条件：</p> <ul style="list-style-type: none">• “存储类别”为“标准存储”。• “区域”须与创建的集群所在区域相同。
备份路径	<p>快照在OBS桶中的存放路径。</p> <p>备份路径配置规则：</p> <ul style="list-style-type: none">• 备份路径不能使用符号“\.*?"<> ”。• 备份路径不能以“/”开头。• 备份路径不能以“.”开头或结尾。• 备份路径的总长度不能超过1023个字符。

参数	说明
IAM委托	<p>指当前账号授权云搜索服务访问或维护存储在OBS中数据。如果没有合适的委托可以联系CSS服务管理员新建IAM委托。详细操作步骤请参见如何创建委托。</p> <p>所选的IAM委托需满足如下条件：</p> <ul style="list-style-type: none">“委托类型”选择“云服务”。“云服务”选择“Elasticsearch”或者“云搜索服务CSS”。必选策略：“OBS Administrator”

表 2-15 设置自动创建快照

参数	说明
快照名称前缀	快照名称前缀的长度为1~32个字符，只能包含小写字母、数字、中划线和下划线，且必须以小写字母开头。快照名称由快照名称前缀加上时间戳组成，例如自动生成的快照名称为“snapshot-1566921603720”。
时区	指备份时间对应的时区，不支持修改。基于此时区选择备份开始时间。
备份开始时间	指每天自动开始备份的时间，只能指定整点时间，如00:00、01:00，取值范围为00:00~23:00。请在下拉框中选择时间。
保留快照个数	<p>自定义设置自动快照保留的个数，范围是1~90。系统在半点时刻会自动删除超过保留个数的快照（过期删除策略只针对与当前自动创建快照策略相同执行频次的自动快照）。</p> <p>说明</p> <p>保存快照个数与自动创建快照策略中设置的执行频次和索引有关，当执行频次间隔时间短或索引数量最大，保留自动快照可能会达不到设置的个数，请谨慎选择。</p>

- b. 配置集群高级功能。根据需要选择“默认配置”或“自定义”。
- “默认配置”：默认关闭“终端节点服务”、“Kibana公网访问”和“标签”功能，在集群创建完成后，如果有需要也可以人工启用这些功能。
 - “自定义”：根据需要选择开启“终端节点服务”、“Kibana公网访问”和“标签”功能。

终端节点服务

终端节点服务（VPC Endpoint Service）通过专属网关，可以将VPC中的服务方便的提供给其他VPC中的资源使用，实现跨VPC的访问，而不必暴露服务端相关的网络信息，使您的访问更加安全、可靠。在开启终端节点服务时，系统会默认给用户创建一个终端节点，内网域名由用户自己选择是否创建，通过节点IP或者是内网域名，用户可以跨VPC访问该集群。

说明

- 当集群的网络配置中，“虚拟私有云”选择的是共享VPC，“子网”选择的是共享VPC下的子网，则该集群不支持配置终端节点服务。
- 集群开启终端节点服务之后，终端节点将按需进行收费，终端节点的费用将由用户进行支付，详细的计费方式请参考[终端节点计费说明](#)。

表 2-16 配置终端节点服务

参数	说明
创建内网域名	勾选“创建内网域名”，系统将会自动为用户创建一个内网域名，通过这个域名可以在同一个VPC内访问该集群。
终端节点服务白名单	<p>在“终端节点服务白名单”中添加允许通过节点IP或内网域名访问集群的账号。</p> <ul style="list-style-type: none">单击“添加”输入授权账号ID。授权账号ID配置成“*”，则表示允许全部用户访问该集群。单击操作列的“删除”，可以删除不允许访问的账号ID。 <p>说明 “授权账号ID”可以在“我的凭证”中进行查看“账号ID”获取。</p>

Kibana公网访问

只有开启“安全模式”的集群，才能配置Kibana公网访问。开启Kibana公网访问后，用户可以获得一个Kibana公网访问地址，通过这个地址可以在公网访问Kibana。

表 2-17 配置 Kibana 公网访问

参数	说明
带宽	设置公网访问的带宽。 取值范围：1-100。 单位：Mbit/s。
访问控制开关	如果关闭访问控制开关，则允许任何IP通过公网IP访问集群Kibana。如果开启访问控制开关，则只允许白名单列表中的IP通过公网IP访问集群Kibana。
白名单	设置允许访问的IP地址或网段，中间用英文逗号隔开。仅当打开“访问控制开关”时才需要配置。 建议开启白名单。 说明 Kibana公网访问配置的白名单依赖ELB的白名单能力。更新白名单后，白名单对新建的连接是实时生效的，但对于已存在的长连接，可能会出现去掉的白名单IP地址还能访问Kibana的场景，这是因为要等长连接断开后才生效，预计1分钟左右。

标签

为集群添加标签，方便用户识别和管理拥有的集群资源。此处您可以选择“标签管理服务”中已定义好的“预定义标签”，也可以自己定义标签。

说明

如您的组织已经设定云搜索服务的相关标签策略，则需按照标签策略规则为集群添加标签。标签不符合标签策略的规则，则可能会导致集群创建失败，请联系组织管理员了解标签策略详情。

表 2-18 标签命名规则

参数	说明
标签键	<ul style="list-style-type: none">对于同一个集群，标签键值唯一。长度不超过64个字符。只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @”。不能以空格开头和结尾。不能为空。
标签值	<ul style="list-style-type: none">长度不超过64个字符。只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @/”。不能以空格开头和结尾。不能为空。

- 单击“下一步：确认配置”，确认完成后单击“立即创建”开始创建集群。
- 单击“返回集群列表”，系统将跳转到“集群管理”页面。您创建的集群将展现在集群列表中，且集群状态为“创建中”，创建成功后集群状态会变为“可用”。

如果集群创建失败，请根据界面提示重新创建集群。

后续操作

Elasticsearch集群创建完成后，建议参考[优化集群性能](#)进行集群的性能优化，便于提高集群的写入和查询性能，提升使用效率。

2.4 访问 Elasticsearch 集群

2.4.1 Elasticsearch 集群访问方式

Elasticsearch集群支持多种访问方式，方便根据业务使用的编程语言自行选择接入方式。各种访问方式兼容多种网络配置方式，不同网络配置方式获取到的集群访问地址有差异，您可以根据自己的网络环境选择合适的网络配置方式。

访问方式

表 2-19 集群的访问方式

访问方式	适用场景	相关文档
Kibana（推荐方式）	<ul style="list-style-type: none">图形化操作与展示。监控实例。	通过Kibana登录Elasticsearch集群
Cerebro	<ul style="list-style-type: none">管理数据。 不限制访问语言。	通过Cerebro登录Elasticsearch集群
开源Elasticsearch API	通过Curl命令访问与管理Elasticsearch集群。	通过Curl命令行接入Elasticsearch集群
Java客户端	通过Java语言访问与管理Elasticsearch集群。 建议使用对应Elasticsearch集群版本的Java客户端，否则可能存在兼容性问题。	通过Rest High Level Client接入Elasticsearch集群 通过Rest Low Level Client接入Elasticsearch集群 通过Transport Client接入Elasticsearch集群 通过Spring Boot接入Elasticsearch集群
Python客户端	通过Python语言访问与管理Elasticsearch集群。	通过Python客户端接入Elasticsearch集群
MRS Hive客户端	通过MRS的ES-Hadoop实现Hive工具访问与管理Elasticsearch集群。	通过MRS Hive客户端接入Elasticsearch集群
Go客户端	通过Go语言访问与管理Elasticsearch集群。	通过Go客户端接入Elasticsearch集群

网络配置

配置Elasticsearch集群的网络，确保服务器与集群的网络是互通的。同时，根据选择的网络获取集群的访问地址，除了Kibana和Cerebro一键访问集群不需要填写集群访问地址，其他访问方式接入集群时都需要输入集群访问地址。

表 2-20 集群的网络配置

网络环境	访问方式	场景介绍	相关文档
内网	内网IP地址	<p>集群基本信息中获取各个节点的IP地址，通过直连节点IP地址群访问集群。</p> <p>获取方式：</p> <ol style="list-style-type: none"> 1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。 2. 在集群管理列表页面，选择需要访问的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。 <p>当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。</p>	默认方式
内网	终端节点IP地址或内网域名	<p>通过配置终端节点实现内网场景下跨VPC访问集群。适用于对性能要求不高的场景。</p> <p>获取方式：</p> <ol style="list-style-type: none"> 1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。 2. 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。 3. 在左侧菜单栏选择“终端节点服务”，获取并记录终端节点服务的“节点IPv4”或“内网域名”。 	配置终端节点服务
内网	独享型负载均衡实例的私有IP地址	<p>通过独享型负载均衡实例分配集群的节点数据。适用于对性能要求高的场景。</p> <p>获取方式：</p> <ol style="list-style-type: none"> 1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。 2. 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。 3. 在左侧菜单栏选择“负载均衡”，获取并记录负载均衡实例的“私有IP”。 	配置独享型负载均衡

网络环境	访问方式	场景介绍	相关文档
公网	公网IP地址	通过配置集群的公网访问白名单实现公网访问集群。仅适用于启用了HTTPS访问的安全集群。 获取方式： 1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。 2. 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。获取并记录“公网访问”。 说明 仅安全模式+HTTPS协议的集群才支持启用公网访问。	配置公网访问
公网	独享型负载均衡实例的公网IP地址	通过独享型负载均衡实例分配集群的节点数据。适用于对性能要求高的场景。 1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。 2. 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。 3. 在左侧菜单栏选择“负载均衡”，获取并记录负载均衡实例的“公网IP”。	配置独享型负载均衡

2.4.2 通过 Kibana 登录 Elasticsearch 集群

CSS服务的Elasticsearch集群默认提供Kibana，无需安装部署，即可一键访问Kibana，同时CSS服务的Elasticsearch集群也兼容了开源Kibana的可视化展现和Elasticsearch统计分析能力。

Kibana支持多种访问方式，不同访问方式登录Elasticsearch集群的操作有差异，具体请参见[表2-21](#)。

表 2-21 通过 Kibana 登录 Elasticsearch 集群的方式

Kibana访问方式	使用约束	参考文档
控制台一键访问 Kibana	无。	通过控制台访问 Kibana登录 Elasticsearch集群
公网地址访问 Kibana	<ul style="list-style-type: none"> 仅安全模式的集群支持通过Kibana公网访问地址访问Kibana。 	通过公网地址访问 Kibana登录 Elasticsearch集群

Kibana访问方式	使用约束	参考文档
内网地址访问 Kibana	同一VPC下的服务器才能通过Elasticsearch集群的内网访问地址访问Kibana。	通过内网地址访问 Kibana登录 Elasticsearch集群
自建Kibana	<ul style="list-style-type: none">本地环境需要支持外网访问。通过同vpc下ECS服务搭建Kibana时，本地公网访问Kibana即可。只支持OSS版本的Kibana镜像连接到云搜索服务的Elasticsearch。	自建Kibana如何对接CSS服务的Elasticsearch集群?

Kibana 使用限制

Kibana中可以自定义用户名、角色名、租户名等，不能包含中文字符。

通过控制台访问 Kibana 登录 Elasticsearch 集群

- 登录云搜索服务管理控制台。
- 在“集群管理”页面选择需要登录的集群，单击“操作”列中的“Kibana”进入Kibana登录界面。
 - 非安全模式的集群：将直接进入Kibana操作界面。
 - 安全模式的集群：需要在登录页面输入用户名和密码，单击“Log In”进入Kibana操作界面。用户名默认为admin，密码为创建集群时设置的管理员密码。
- 登录成功后，可在Kibana界面进行相关操作访问集群。

通过公网地址访问 Kibana 登录 Elasticsearch 集群

📖 说明

- 仅安全模式的集群支持通过Kibana公网访问地址访问Kibana。
- 登录云搜索服务管理控制台。
 - 开启Elasticsearch集群的Kibana公网访问。支持在创建集群的时候就配置Kibana公网访问，或者在集群创建完之后再开启Kibana公网访问。
 - 在创建集群时配置Kibana公网访问：操作指导请参见[创建Elasticsearch集群](#)。
 - 集群创建完之后再开启Kibana公网访问：
 - 在集群管理页面，单击需要配置Kibana公网访问的集群名称，进入集群基本信息页面。
 - 左侧菜单栏选择“Kibana公网访问”，在“Kibana公网访问”右侧单击开关，打开Kibana公网访问功能。
 - 在开启Kibana公网访问页面，配置相关参数。如果集群已配置Kibana公网访问，此时也支持修改相关配置。

表 2-22 配置 Kibana 公网访问

参数	说明
带宽	设置公网访问的带宽。 取值范围：1-100。 单位：Mbit/s。
访问控制开关	如果关闭访问控制开关，则允许任何IP通过公网IP访问集群Kibana。如果开启访问控制开关，则只允许白名单列表中的IP通过公网IP访问集群Kibana。
白名单	设置允许访问的IP地址或网段，中间用英文逗号隔开。仅当打开“访问控制开关”时才需要配置。 建议开启白名单。 说明 Kibana公网访问配置的白名单依赖ELB的白名单能力。更新白名单后，白名单对新建的连接是实时生效的，但对于已存在的长连接，可能会出现去掉的白名单IP地址还能访问Kibana的场景，这是因为要等长连接断开后才生效，预计1分钟左右。

- iv. 配置完成后，单击“确定”。
3. 待集群的Kibana公网访问开通以后，在Kibana公网访问页面获取“kibana公网访问地址”。

图 2-1 获取 Kibana 公网访问地址



4. 在浏览器中，输入“kibana公网访问地址”即可进入Kibana登录界面。
在登录页面输入用户名和密码，单击“Log In”进入Kibana操作界面。用户名默认为admin，密码为创建集群时设置的管理员密码。
5. 登录成功后，可在Kibana界面进行相关操作访问Elasticsearch集群。

通过内网地址访问 Kibana 登录 Elasticsearch 集群

📖 说明

同一VPC下的服务器才能通过Elasticsearch集群的内网访问地址访问Kibana。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击集群名称，进入集群基本信息页面。
3. 在集群基本信息页面，获取集群的内网访问地址。

图 2-2 获取内网地址



4. 将集群的内网访问地址的端口号从“9200”改为“5601”，即为Kibana的内网地址。例如，集群的“内网访问IPv4地址”是“192.168.0.***:9200”，则Kibana的内网地址为“192.168.0.***:5601”。
5. 在服务器中，输入Kibana的内网地址即可进入Kibana登录界面。
 - 非安全模式的集群：将直接进入Kibana操作界面。
 - 安全模式的集群：需要在登录页面输入用户名和密码，单击“Log In”进入Kibana操作界面。用户名默认为admin，密码为创建集群时设置的管理员密码。
6. 登录成功后，可在Kibana界面进行相关操作访问Elasticsearch集群。

2.4.3 通过 Cerebro 登录 ELasticsearch 集群

CSS服务的ELasticsearch和OpenSearch集群默认提供Cerebro，无需安装部署，即可一键访问Cerebro，同时CSS服务的ELasticsearch和OpenSearch集群也完全兼容开源Cerebro。

通过 CSS 控制台的 Cerebro 访问集群

CSS服务创建的ELasticsearch和OpenSearch集群自带Cerebro组件，支持一键打开Cerebro，快速访问ELasticsearch和OpenSearch集群。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要登录的集群，单击“操作”列中的“更多 > Cerebro”进入Cerebro登录页面。

- 非安全模式的集群：单击Cerebro登录页面的集群名称即可进入Cerebro操作界面。
 - 安全模式的集群：单击Cerebro登录页面的集群名称，再输入用户名和密码，单击“Authenticate”进入Cerebro操作界面。用户名默认为admin，密码为创建集群时设置的管理员密码。
3. 登录成功后，可在Cerebro界面进行相关操作访问集群。

通过自建 Cerebro 访问集群

📖 说明

使用自建Cerebro访问集群时，需要确保自建Cerebro与集群的网络是互通的。

1. 获取集群的访问地址。不同网络配置的访问地址获取方式请参见[网络配置](#)。
2. 启动自建Cerebro，填写集群的访问地址对接CSS集群。
 - 安全模式的集群填写：https://访问地址:9200
安全模式的集群要输入集群的用户名和密码才能登录。
 - 非安全模式的集群填写：http://访问地址:9200

2.4.4 通过 Curl 命令行接入 Elasticsearch 集群

开源Elasticsearch提供了一系列RESTful风格的API，通过Curl命令可以在Kibana、Postman等工具中使用这些API。本文介绍如何通过Curl命令接入Elasticsearch和OpenSearch集群。

前提条件

- CSS服务的Elasticsearch和OpenSearch集群处于可用状态。
- 已创建一个ECS服务，且和CSS服务的集群在同一个虚拟私有云和安全组中。
 - 如果ECS和CSS服务的集群不在同一安全组中，请修改ECS安全组或者配置ECS安全组的出入规则，允许集群所有安全组的访问。修改操作请参见[配置安全组规则](#)。
 - ECS的使用请参见[弹性云服务器ECS的操作导航](#)。

操作步骤

1. 获取集群的内网访问地址。访问集群时，需要输入内网访问地址。

本文以内网IP地址方式访问集群为例，当使用其他网络配置访问集群时，仅访问地址有差异，不同访问地址的获取方式请参见[网络配置](#)。

 - a. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。
 - b. 在集群管理列表页面，选择需要访问的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。

当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。
2. 在ECS中执行如下命令，访问集群。集群的安全模式不同，访问命令也不同。
 - 非安全模式的集群

```
curl "http://<host>:<port>"
```

- 安全模式+HTTP协议的集群
`curl -u <user>:<password> "http://<host>:<port>"`
- 安全模式+HTTPS协议的集群
`curl -u <user>:<password> -k "https://<host>:<port>"`

表 2-23 变量说明

变量名	说明
<host>	集群中各节点的IP地址，当集群包含多个节点时，会存在多个IP地址，可以任选其中一个发送。
<port>	集群节点的访问端口号，一般为9200。
<user>	访问集群的用户名。
<password>	用户名对应的密码。 当密码中存在特殊字符时，请将用户名和密码加上英文双引号，例如“curl -u "user:password!" "http://<host>:<port>"”。

访问示例如下：

```
curl "http://10.62.176.32:9200"
```

返回结果如下：

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8
content-length: 513

{
  "name": "xxx-1",
  "cluster_name": "xxx",
  "cluster_uuid": "xxx_uuid",
  "version": {
    "number": "7.10.2",
    "build_flavor": "oss",
    "build_type": "tar",
    "build_hash": "unknown",
    "build_date": "unknown",
    "build_snapshot": true,
    "lucene_version": "8.7.0",
    "minimum_wire_compatibility_version": "6.7.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

📖 说明

更多命令，请参见[Elasticsearch官方文档](#)。

2.4.5 通过 Java 客户端接入 Elasticsearch 集群

2.4.5.1 通过 Rest High Level Client 接入 Elasticsearch 集群

Elasticsearch官方提供了SDK（Rest High level Client）方式连接集群，Rest Client客户端对Elasticsearch的API进行了封装，用户只需要构造对应的结构即可对ES集群进行

访问。Rest Client的详细使用请参考官方文档：<https://www.elastic.co/guide/en/elasticsearch/client/java-api-client/master/index.html>

本文介绍通过Rest High level Client访问CSS集群的配置说明。Rest High level Client接入集群有3种方式：

- **通过Rest High Level Client连接非安全集群**：适用于非安全模式的集群
- **通过Rest High Level Client连接安全集群（不使用安全证书）**：适用于安全模式+HTTP协议的集群、安全模式+HTTPS协议的集群（忽略证书）
- **通过Rest High Level Client连接安全集群（使用安全证书）**：适用于安全模式+HTTPS协议的集群

约束限制

建议Rest High Level Client的版本和Elasticsearch的版本保持一致，例如需要访问的ES集群版本是7.6.2，则使用的Rest High Level Client客户端版本建议也是7.6.2。如果您使用相比Elasticsearch集群更高版本的Java Rest High Level Client且存在少量请求的兼容性问题，您可以使用“RestHighLevelClient.getLowLevelClient()”方式直接获取Low Level Client，实现自定义的Elasticsearch请求内容。

前提条件

- CSS集群处于可用状态。
- 确保运行Java代码的服务器与CSS集群的网络是互通的。
- 根据集群选择的网络配置方式，获取集群的访问地址，具体操作请参见[网络配置](#)。
- 确认服务器已安装JDK1.8，JDK1.8官网下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>。
- 服务器已引入Java依赖。

其中7.6.2为Elasticsearch Java客户端的版本号。

- Maven方式引入：

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>7.6.2</version>
</dependency>
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch</artifactId>
  <version>7.6.2</version>
</dependency>
```

- Gradle方式引入：

```
compile group: 'org.elasticsearch.client', name: 'elasticsearch-rest-high-level-client', version: '7.6.2'
```

通过 Rest High Level Client 连接非安全集群

通过Rest High Level Client连接非安全集群，并查询test索引是否存在。代码示例如下：

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;
```

```
import java.io.IOException;
import java.util.Arrays;
import java.util.List;

/**
 * Rest Hive Level 连接非安全集群
 */
public class Main {
    public static void main(String[] args) throws IOException {
        List<String> host = Arrays.asList("x.x.x.x", "x.x.x.x");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        GetIndexRequest indexRequest = new GetIndexRequest("test");
        boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);
        System.out.println(exists);
        client.close();
    }

    /**
     * constructHttpHosts函数转换host集群节点ip列表。
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

其中，*host*为集群访问地址，当存在多个IP地址时，中间用“,”隔开；*test*为查询的索引名称。

通过 Rest High Level Client 连接安全集群（不使用安全证书）

该场景适用于连接2种集群：安全模式+HTTP协议的集群、安全模式+HTTPS协议的集群（忽略证书）。

代码示例如下：

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;
import org.elasticsearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
```



```
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

/**
 * Rest High Level连接安全集群（不使用证书）
 */
public class Main {
    /**
     * 创建客户端的类，定义create函数用于创建客户端。
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int connectTimeout,
int connectionRequestTimeout, int socketTimeout, String username, String password) throws IOException{
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(username,
password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig -> requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {} ", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("es rest client health response {} ", response);
        return client;
    }

    /**
     * constructHttpHosts函数转换host集群节点ip列表。
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }

    /**
     * trustAllCerts忽略证书配置。
     */
    public static TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            @Override
            public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException
            {
            }

            @Override
            public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
            }

            @Override
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        }
    }
}
```

```
};

private static final Logger logger = LogManager.getLogger(Main.class);

static class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;
    /**
     * The {@link SSLIOStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOStrategy sslStrategy;
    /**
     * Create a new {@link SecuredHttpClientConfigCallback}.
     *
     * @param credentialsProvider The credential provider, if a username/password have been supplied
     * @param sslStrategy The SSL strategy, if SSL / TLS have been supplied
     * @throws NullPointerException if {@code sslStrategy} is {@code null}
     */
    SecuredHttpClientConfigCallback(final SSLIOStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }
    /**
     * Get the {@link CredentialsProvider} that will be added to the HTTP client.
     *
     * @return Can be {@code null}.
     */
    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }
    /**
     * Get the {@link SSLIOStrategy} that will be added to the HTTP client.
     *
     * @return Never {@code null}.
     */
    SSLIOStrategy getSSLStrategy() {
        return sslStrategy;
    }
    /**
     * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
    credential provider},
     *
     * @param httpClientBuilder The client to configure.
     * @return Always {@code httpClientBuilder}.
     */
    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
        // enable SSL / TLS
        httpClientBuilder.setSSLStrategy(sslStrategy);
        // enable user authentication
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }
}

public static class NullHostNameVerifier implements HostnameVerifier {
    @Override
    public boolean verify(String arg0, SSLSession arg1) {
        return true;
    }
}

/**
 * main函数参考如下，调用上面的create函数创建客户端，查询“test”索引是否存在。
 */
```

```
public static void main(String[] args) throws IOException {  
    RestHighLevelClient client = create(Arrays.asList("x.x.x.x", "x.x.x.x"), 9200, "https", 1000, 1000, 1000,  
    "username", "password");  
    GetIndexRequest indexRequest = new GetIndexRequest("test");  
    boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);  
    System.out.println(exists);  
    client.close();  
}  
}
```

表 2-24 函数中的变量说明

参数	描述
host	Elasticsearch集群访问地址，当存在多个IP地址时，中间用“,”隔开。
port	Elasticsearch集群的连接端口，默认是“9200”。
protocol	连接协议，“http”或者“https”，根据集群实际情况填写。
connectTimeout	socket连接超时时间。
connectionRequestTimeout	socket连接请求超时时间。
socketTimeout	socket请求超时时间。
username	访问集群的用户名。
password	用户名对应的密码。

通过 Rest High Level Client 连接安全集群（使用安全证书）

该场景适用于使用安全证书连接安全模式+HTTPS协议的集群。

1. 获取安全证书（CloudSearchService.cer）。
 - a. 登录云搜索服务控制台。
 - b. 选择“集群管理”进入集群列表。
 - c. 单击对应集群的名称，进入集群基本信息页面。
 - d. 在“基本信息”页面，单击“HTTPS访问”后面的“下载证书”。

图 2-3 下载证书

配置信息

区域	
可用区	
虚拟私有云	vpc-
子网	subnet
安全组	dws 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	-- 绑定
HTTPS访问	开启 下载证书
内网访问IPv4地址	192

2. 转换安全证书（CloudSearchService.cer）。将下载的安全证书上传到客户端机器上，使用keytool工具将“.cer”证书转换成Java可以读取的“.jks”证书格式。

- 在Linux系统中，执行如下命令转换证书。

```
keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
```

- 在Windows系统中，执行如下命令转换证书。

```
keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
```

其中，*newname*是由用户自定义的证书名称。

该命令执行后，会提示设置证书密码，并确认密码。请保存该密码，后续接入集群会使用。

3. 接入集群。代码示例如下：

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
```

```
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.GetIndexRequest;
import org.elasticsearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

/**
 * Rest Hive Level连接安全集群（使用https证书）
 */
public class Main {
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int
connectTimeout, int connectionRequestTimeout, int socketTimeout, String username, String password,
String cerFilePath,
String cerPassword) throws IOException {

        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(username, password));
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(cerFilePath, cerPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //也可以使用SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig ->
requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestHighLevelClient client = new RestHighLevelClient(builder);
        logger.info("es rest client build success {}", client);

        ClusterHealthRequest request = new ClusterHealthRequest();
        ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
        logger.info("es rest client health response {}", response);
        return client;
    }

    /**
     * constructHttpHosts函数转换host集群节点ip列表。
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
```

```
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
}

/**
 * SecuredHttpClientConfigCallback类定义。
 */
static class SecuredHttpClientConfigCallback implements
RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;

    private final SSLIOSessionStrategy sslStrategy;

    SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
    @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }

    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }

    SSLIOSessionStrategy getSSLStrategy() {
        return sslStrategy;
    }

    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder
httpClientBuilder) {
        httpClientBuilder.setSSLStrategy(sslStrategy);
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }
}

private static final Logger logger = LogManager.getLogger(Main.class);

public static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String cerFilePath, String cerPassword) throws Exception {
        File file = new File(cerFilePath);
        if (!file.isFile()) {
            throw new Exception("Wrong Certification Path");
        }
        System.out.println("Loading KeyStore " + file + "...");
        InputStream in = new FileInputStream(file);
        KeyStore ks = KeyStore.getInstance("JKS");
        ks.load(in, cerPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
    }

    @Override
    public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {

```

```
    }  
  
    @Override  
    public void checkServerTrusted(X509Certificate[] chain, String authType) throws  
CertificateException {  
  
    }  
  
    @Override  
    public X509Certificate[] getAcceptedIssuers() {  
        return new X509Certificate[0];  
    }  
}  
  
/**  
 * main函数参考如下，调用上面的create函数创建客户端，查询“test”索引是否存在。  
 */  
public static void main(String[] args) throws IOException {  
    RestHighLevelClient client = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200,  
"https", 1000, 1000, 1000, "username", "password", "cerFilePath", "cerPassword");  
    GetIndexRequest indexRequest = new GetIndexRequest("test");  
    boolean exists = client.indices().exists(indexRequest, RequestOptions.DEFAULT);  
    System.out.println(exists);  
    client.close();  
}  
}
```

表 2-25 函数中的参数说明

参数	描述
host	Elasticsearch集群访问地址，当存在多个IP地址时，中间用“,”隔开。
port	Elasticsearch集群的连接端口，默认是“9200”。
protocol	连接协议，此处填写“https”。
connectTimeout	socket连接超时时间。
connectionRequestTimeout	socket连接请求超时时间。
socketTimeout	socket请求超时时间。
username	访问集群的用户名。
password	用户名对应的密码。
cerFilePath	证书路径。
cerPassword	证书密码。

2.4.5.2 通过 Rest Low Level Client 接入 Elasticsearch 集群

High Level Client是在Low Level Client基础上进行封装的，如果High Level Client中的方法调用（例如“.search”，“.bulk”）不能满足使用需求，或存在兼容性问题，可以选择使用Low Level Client方式，甚至可以使用“HighLevelClient.getLowLevelClient()”方式直接获取Low Level Client。使用Low Level Client发送请求时需要自定义请求结构，使用上更为灵活，能满足所有Elasticsearch支持的请求格式，例如GET、POST、DELETE、HEAD等。

本文介绍通过Rest Low Level Client访问CSS集群的配置说明。Rest Low Level Client接入集群有3种方式，每一种方式又分为直接创建Rest Client（即Rest Low Level Client）和通过创建High Level Client再调用getLowLevelClient()获取Low Level Client。

- **通过Rest Low Level Client连接非安全集群**：适用于非安全模式的集群
- **通过Rest Low Level Client连接安全集群（不使用安全证书）**：适用于安全模式+HTTP协议的集群、安全模式+HTTPS协议的集群（忽略证书）
- **通过Rest Low Level Client连接安全集群（使用安全证书）**：适用于安全模式+HTTPS协议的集群

注意事项

建议Rest Low Level Client的版本和Elasticsearch的版本保持一致，例如需要访问的ES集群版本是7.6.2，则使用的Rest Low Level Client客户端版本建议也是7.6.2。

准备工作

- CSS集群处于可用状态。
- 确保运行Java代码的服务器与CSS集群的网络是互通的。
- 根据集群选择的网络配置方式，获取集群的访问地址，具体操作请参见[网络配置](#)。
- 确认服务器已安装JDK1.8，JDK1.8官网下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>。
- 通过Maven方式引入apache版本。代码示例以7.6.2版本为例。
其中7.6.2为Elasticsearch Java客户端的版本号。

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-client</artifactId>
  <version>7.6.2</version>
</dependency>
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch</artifactId>
  <version>7.6.2</version>
</dependency>
```

通过 Rest Low Level Client 连接非安全集群

- **方式一：直接创建Rest Low Level Client**

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;

public class Main {

    public static void main(String[] args) throws IOException {
        List<String> host = Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        /**
         * 创建Rest Low Level Client。
         */
    }
}
```



```
RestClient lowLevelClient = builder.build();
/**
 * 查询“test”索引是否存在。当索引存在时返回200，不存在时返回404。
 */
Request request = new Request("HEAD", "/test");
Response response = lowLevelClient.performRequest(request);
System.out.println(response.getStatusLine().getStatusCode());
lowLevelClient.close();
}

/**
 * constructHttpHosts函数转换host集群节点ip列表。
 */
public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
}
}
```

- **方式二：创建High Level Client再调用getLowLevelClient()获取Low Level Client**

```
import org.apache.http.HttpHost;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;

import java.io.IOException;
import java.util.Arrays;
import java.util.List;

public class Main {

    public static void main(String[] args) throws IOException {
        List<String> host = Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx");
        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, 9200, "http"));
        final RestHighLevelClient restHighLevelClient = new RestHighLevelClient(builder);
        /**
         * 创建High Level Client再调用getLowLevelClient()获取Low Level Client，即client创建仅下面这
         * 一行代码存在差别。
         */
        final RestClient lowLevelClient = restHighLevelClient.getLowLevelClient();
        /**
         * 查询“test”索引是否存在。当索引存在时返回200，不存在时返回404。
         */
        Request request = new Request("HEAD", "/test");
        Response response = lowLevelClient.performRequest(request);
        System.out.println(response.getStatusLine().getStatusCode());
        lowLevelClient.close();
    }

    /**
     * constructHttpHosts函数转换host集群节点ip列表。
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
    }
}
```

其中，*host*为集群的访问地址，当存在多个IP地址时，中间用“,”隔开；*test*为查询的索引名称。

通过 Rest Low Level Client 连接安全集群（不使用安全证书）

- **方式一：直接创建Rest Low Level Client**

```
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
import org.apache.http.auth.AuthScope;
```

```
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.client.DefaultConnectionKeepAliveStrategy;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.http.protocol.HttpContext;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;import javax.net.ssl.X509TrustManager;

public class Main {

    /**
     * 创建客户端的类，定义create函数用于创建客户端。
     */
    public static RestClient create(List<String> host, int port, String protocol, int connectTimeout, int
connectionRequestTimeout, int socketTimeout, String username, String password) throws
IOException {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(username, password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig ->
requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestClient client = builder.build();
        logger.info("es rest client build success {}", client);
        return client;
    }

    /**
     * constructHttpHosts函数转换host集群节点ip列表。
     */
}
```

```
public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
}

/**
 * trustAllCerts忽略证书配置。
 */
public static TrustManager[] trustAllCerts = new TrustManager[] {
    new X509TrustManager() {
        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }

        @Override
        public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }

        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    }
};

/**
 * CustomConnectionKeepAliveStrategy函数设置连接的保活时间，主要应对大量短连接的情况和数据
请求不高的场景。
 */
public static class CustomConnectionKeepAliveStrategy extends
DefaultConnectionKeepAliveStrategy {
    public static final CustomConnectionKeepAliveStrategy INSTANCE = new
CustomConnectionKeepAliveStrategy();

    private CustomConnectionKeepAliveStrategy() {
        super();
    }

    /**
     * 最大keep alive的时间（分钟）
     * 这里默认为10分钟，可以根据实际情况设置。可以观察客户端机器状态为TIME_WAIT的TCP连接
数，如果太多，可以增大此值。
     */
    private final long MAX_KEEP_ALIVE_MINUTES = 10;

    @Override
    public long getKeepAliveDuration(HttpResponse response, HttpContext context) {
        long keepAliveDuration = super.getKeepAliveDuration(response, context);
        // <0 为无限期keepalive
        // 将无限期替换成一个默认的时间
        if (keepAliveDuration < 0) {
            return TimeUnit.MINUTES.toMillis(MAX_KEEP_ALIVE_MINUTES);
        }
        return keepAliveDuration;
    }
}

private static final Logger logger = LogManager.getLogger(Main.class);

static class SecuredHttpClientConfigCallback implements
RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;
    /**
     * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOSessionStrategy sslStrategy;
    /**
```

```
* Create a new {@link SecuredHttpClientConfigCallback}.
*
* @param credentialsProvider The credential provider, if a username/password have been
supplied
* @param sslStrategy The SSL strategy, if SSL / TLS have been supplied
* @throws NullPointerException if {@code sslStrategy} is {@code null}
*/
SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
    @Nullable final CredentialsProvider credentialsProvider) {
    this.sslStrategy = Objects.requireNonNull(sslStrategy);
    this.credentialsProvider = credentialsProvider;
}
/**
* Get the {@link CredentialsProvider} that will be added to the HTTP client.
*
* @return Can be {@code null}.
*/
@Nullable
CredentialsProvider getCredentialsProvider() {
    return credentialsProvider;
}
/**
* Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
*
* @return Never {@code null}.
*/
SSLIOSessionStrategy getSSLStrategy() {
    return sslStrategy;
}
/**
* Sets the {@linkplain
HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider) credential provider},
*
* @param httpClientBuilder The client to configure.
* @return Always {@code httpClientBuilder}.
*/
@Override
public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder
httpClientBuilder) {
    // enable SSL / TLS
    httpClientBuilder.setSSLStrategy(sslStrategy);
    // enable user authentication
    if (credentialsProvider != null) {
        httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
    }
    return httpClientBuilder;
}
}

public static class NullHostNameVerifier implements HostnameVerifier {
    @Override
    public boolean verify(String arg0, SSLSession arg1) {
        return true;
    }
}
}

/**
* main函数参考如下，调用create函数创建Rest Low Level Client客户端，查询“test”索引是否存在。
*/
public static void main(String[] args) throws IOException {
    RestClient lowLevelClient = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200, "http",
1000, 1000, 1000, "username", "password");
    Request request = new Request("HEAD", "/test");
    Response response = lowLevelClient.performRequest(request);
    System.out.println(response.getStatusLine().getStatusCode());
    lowLevelClient.close();
}
}
```

- **方式二：创建High Level Client再调用getLowLevelClient()获取Low Level Client**

```
import org.apache.http.HttpHost;
import org.apache.http.HttpResponse;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.client.DefaultConnectionKeepAliveStrategy;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.apache.http.protocol.HttpContext;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;

import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;import javax.net.ssl.X509TrustManager;

import org.elasticsearch.client.RestHighLevelClient;

public class Main13 {

    /**
     * 创建客户端的类，定义create函数用于创建客户端。
     */
    public static RestHighLevelClient create(List<String> host, int port, String protocol, int
connectTimeout, int connectionRequestTimeout, int socketTimeout, String username, String
password) throws IOException {

        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(username, password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig ->
requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
```

```
final RestHighLevelClient client = new RestHighLevelClient(builder);
logger.info("es rest client build success {}", client);
return client;
}

/**
 * constructHttpHosts函数转换host集群节点ip列表。
 */
public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
}

/**
 * trustAllCerts忽略证书配置。
 */
public static TrustManager[] trustAllCerts = new TrustManager[] {
    new X509TrustManager() {
        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }

        @Override
        public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }

        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    }
};

/**
 * CustomConnectionKeepAliveStrategy函数设置连接的保活时间，主要应对大量短连接的情况和数据
请求不高的场景。
 */
public static class CustomConnectionKeepAliveStrategy extends
DefaultConnectionKeepAliveStrategy {
    public static final CustomConnectionKeepAliveStrategy INSTANCE = new
CustomConnectionKeepAliveStrategy();

    private CustomConnectionKeepAliveStrategy() {
        super();
    }

    /**
     * 最大keep alive的时间（分钟）
     * 这里默认为10分钟，可以根据实际情况设置。可以观察客户端机器状态为TIME_WAIT的TCP连接
数，如果太多，可以增大此值。
     */
    private final long MAX_KEEP_ALIVE_MINUTES = 10;

    @Override
    public long getKeepAliveDuration(HttpResponse response, HttpContext context) {
        long keepAliveDuration = super.getKeepAliveDuration(response, context);
        // <0 为无限期keepalive
        // 将无限期替换成一个默认的时间
        if (keepAliveDuration < 0) {
            return TimeUnit.MINUTES.toMillis(MAX_KEEP_ALIVE_MINUTES);
        }
        return keepAliveDuration;
    }
}

private static final Logger logger = LogManager.getLogger(Main.class);

static class SecuredHttpClientConfigCallback implements
```

```
RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;
    /**
     * The {@link SSLIOStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOStrategy sslStrategy;
    /**
     * Create a new {@link SecuredHttpClientConfigCallback}.
     *
     * @param credentialsProvider The credential provider, if a username/password have been
    supplied
     * @param sslStrategy The SSL strategy, if SSL / TLS have been supplied
     * @throws NullPointerException if {@code sslStrategy} is {@code null}
     */
    SecuredHttpClientConfigCallback(final SSLIOStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }
    /**
     * Get the {@link CredentialsProvider} that will be added to the HTTP client.
     *
     * @return Can be {@code null}.
     */
    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }
    /**
     * Get the {@link SSLIOStrategy} that will be added to the HTTP client.
     *
     * @return Never {@code null}.
     */
    SSLIOStrategy getSSLStrategy() {
        return sslStrategy;
    }
    /**
     * Sets the {@linkplain
    HttpClientBuilder#setDefaultCredentialsProvider(CredentialsProvider) credential provider},
     *
     * @param httpClientBuilder The client to configure.
     * @return Always {@code httpClientBuilder}.
     */
    @Override
    public HttpClientBuilder customizeHttpClient(final HttpClientBuilder
    httpClientBuilder) {
        // enable SSL / TLS
        httpClientBuilder.setSSLStrategy(sslStrategy);
        // enable user authentication
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }
}

public static class NullHostNameVerifier implements HostnameVerifier {
    @Override
    public boolean verify(String arg0, SSLSession arg1) {
        return true;
    }
}

/**
 * main函数参考如下，调用create函数创建High Level Client再调用getLowLevelClient()获取Low
    Level Client，并查询“test”索引是否存在。
 */
public static void main(String[] args) throws IOException {
```

```
RestHighLevelClient client = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200,
"http", 1000, 1000, 1000, "username", "password");
RestClient lowLevelClient = client.getLowLevelClient();
Request request = new Request("HEAD", "test");
Response response = lowLevelClient.performRequest(request);
System.out.println(response.getStatusLine().getStatusCode());
lowLevelClient.close();
}
}
```

表 2-26 函数中的变量说明

参数	描述
host	Elasticsearch集群访问地址，当存在多个IP地址时，中间用“,”隔开。
port	Elasticsearch集群的连接端口，默认是“9200”。
protocol	连接协议，“http”或者“https”，根据集群实际情况填写。
connectTimeout	socket连接超时时间。
connectionRequestTimeout	socket连接请求超时时间。
socketTimeout	socket请求超时时间。
username	访问集群的用户名。
password	用户名对应的密码。

通过 Rest Low Level Client 连接安全集群（使用安全证书）

- 方式一：直接创建Rest Low Level Client

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;
```



```
import javax.net.ssl.SSLContext;import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main13 {

    private static final Logger logger = LogManager.getLogger(Main.class);

    /**
     * 创建客户端的类，定义create函数用于创建客户端。
     */
    public static RestClient create(List<String> host, int port, String protocol, int connectTimeout, int
connectionRequestTimeout, int socketTimeout, String username, String password, String cerFilePath,
String cerPassword) throws IOException {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(username, password));
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(cerFilePath, cerPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //也可以使用SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
            .setRequestConfigCallback(requestConfig ->
requestConfig.setConnectTimeout(connectTimeout)
                .setConnectionRequestTimeout(connectionRequestTimeout)
                .setSocketTimeout(socketTimeout))
            .setHttpClientConfigCallback(httpClientConfigCallback);
        final RestClient client = builder.build();
        logger.info("es rest client build success {} ", client);
        return client;
    }

    /**
     * constructHttpHosts函数转换host集群节点ip列表。
     */
    public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
        return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);}

    static class SecuredHttpClientConfigCallback implements
RestClientBuilder.HttpClientConfigCallback {
        @Nullable
        private final CredentialsProvider credentialsProvider;

        private final SSLIOSessionStrategy sslStrategy;

        SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
            @Nullable final CredentialsProvider credentialsProvider) {
            this.sslStrategy = Objects.requireNonNull(sslStrategy);
            this.credentialsProvider = credentialsProvider;
        }

        @Nullable
        CredentialsProvider getCredentialsProvider() {
            return credentialsProvider;
        }
    }
}
```

```
SSLIOSessionStrategy getSSLStrategy() {
    return sslStrategy;
}

@Override
public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder
httpClientBuilder) {
    httpClientBuilder.setSSLStrategy(sslStrategy);
    if (credentialsProvider != null) {
        httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
    }
    return httpClientBuilder;
}

public static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String cerFilePath, String cerPassword) throws Exception {
        File file = new File(cerFilePath);
        if (!file.isFile()) {
            throw new Exception("Wrong Certification Path");
        }
        System.out.println("Loading KeyStore " + file + "...");
        InputStream in = new FileInputStream(file);
        KeyStore ks = KeyStore.getInstance("JKS");
        ks.load(in, cerPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
    }

    @Override
    public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {

    }

    @Override
    public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {

    }

    @Override
    public X509Certificate[] getAcceptedIssuers() {
        return new X509Certificate[0];
    }
}

/**
 * main函数参考如下，调用create函数创建Rest Low Level Client，查询“test”索引是否存在。
 */
public static void main(String[] args) throws IOException {
    RestClient lowLevelClient = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200,
"https", 1000, 1000, 1000, "username", "password", "cerFilePath", "cerPassword");
    Request request = new Request("HEAD", "test");
    Response response = lowLevelClient.performRequest(request);
    System.out.println(response.getStatusLine().getStatusCode());
    lowLevelClient.close();
}
}
```

- **方式二：创建High Level Client再调用getLowLevelClient()获取Low Level Client**

```
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthRequest;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.common.Nullable;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;
import java.util.Objects;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;

public class Main {

    private static final Logger logger = LogManager.getLogger(Main.class);

    /**
     * 创建客户端的类，定义create函数用于创建客户端。
     */
    public static RestClient create(List<String> host, int port, String protocol, int
connectTimeout, int connectionRequestTimeout, int socketTimeout, String username, String password,
String cerFilePath, String cerPassword) throws IOException {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(username, password));
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(cerFilePath, cerPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            //也可以使用SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOStrategy sessionStrategy = new SSLIOStrategy(sc, new
NoopHostnameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);

        RestClientBuilder builder = RestClient.builder(constructHttpHosts(host, port, protocol))
.setRequestConfigCallback(requestConfig ->
```

```
requestConfig.setConnectTimeout(connectTimeout)
    .setConnectionRequestTimeout(connectionRequestTimeout)
    .setSocketTimeout(socketTimeout)
    .setHttpClientConfigCallback(httpClientConfigCallback);
final RestHighLevelClient client = new RestHighLevelClient(builder);
logger.info("es rest client build success {}", client);

ClusterHealthRequest request = new ClusterHealthRequest();
ClusterHealthResponse response = client.cluster().health(request, RequestOptions.DEFAULT);
logger.info("es rest client health response {}", response);
return client;
}

/**
 * constructHttpHosts函数转换host集群节点ip列表。
 */
public static HttpHost[] constructHttpHosts(List<String> host, int port, String protocol) {
    return host.stream().map(p -> new HttpHost(p, port, protocol)).toArray(HttpHost[]::new);
}

static class SecuredHttpClientConfigCallback implements
RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;

    private final SSLIOSessionStrategy sslStrategy;

    SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }

    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }

    SSLIOSessionStrategy getSSLStrategy() {
        return sslStrategy;
    }

    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder
httpClientBuilder) {
        httpClientBuilder.setSSLStrategy(sslStrategy);
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }
}

public static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;

    MyX509TrustManager(String cerFilePath, String cerPassword) throws Exception {
        File file = new File(cerFilePath);
        if (!file.isFile()) {
            throw new Exception("Wrong Certification Path");
        }
        System.out.println("Loading KeyStore " + file + "...");
        InputStream in = new FileInputStream(file);
        KeyStore ks = KeyStore.getInstance("JKS");
        ks.load(in, cerPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
            }
        }
    }
}
```

```

        return;
    }
}
throw new Exception("Couldn't initialize");
}

@Override
public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {

}

@Override
public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {

}

@Override
public X509Certificate[] getAcceptedIssuers() {
    return new X509Certificate[0];
}
}

/**
 * main函数参考如下，调用create函数创建High Level Client再调用getLowLevelClient()获取Low
 * Level Client，并查询“test”索引是否存在。
 */
public static void main(String[] args) throws IOException {
    RestHighLevelClient client = create(Arrays.asList("xxx.xxx.xxx.xxx", "xxx.xxx.xxx.xxx"), 9200,
    "https", 1000, 1000, 1000, "username", "password", "cerFilePath", "cerPassword");
    RestClient lowLevelClient = client.getLowLevelClient();
    Request request = new Request("HEAD", "test");
    Response response = lowLevelClient.performRequest(request);
    System.out.println(response.getStatusLine().getStatusCode());
    lowLevelClient.close();
}
}

```

表 2-27 函数中的参数说明

参数	描述
host	Elasticsearch集群访问地址，当存在多个IP地址时，中间用“,”隔开。
port	Elasticsearch集群的连接端口，默认是“9200”。
protocol	连接协议，此处填写“https”。
connectTimeout	socket连接超时时间。
connectionRequestTimeout	socket连接请求超时时间。
socketTimeout	socket请求超时时间。
username	访问集群的用户名。
password	用户名对应的密码。
cerFilePath	证书路径。
cerPassword	证书密码。

2.4.5.3 通过 Transport Client 接入 Elasticsearch 集群

本文介绍通过 Transport Client 访问 CSS 服务非安全集群的配置说明。如果是安全模式的集群，建议[通过 Rest High Level Client 接入 Elasticsearch 集群](#)。

注意事项

- 建议 Transport Client 的版本和 Elasticsearch 的版本保持一致，例如需要访问的 Elasticsearch 集群版本是 7.6.2，则使用的 Transport Client 客户端版本建议也是 7.6.2。
- 由于该方案只适用于接入非安全集群，因此只支持通过内网 IP 地址的方式访问集群。

准备工作

- CSS 集群处于可用状态。
- 确保运行 Java 代码的服务器与 CSS 集群的网络是互通的。
- 确认服务器已安装 JDK 1.8，JDK 1.8 官网下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>。
- 引入 Java 依赖：
其中 7.6.2 为 Elasticsearch Java 客户端的版本号。

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>transport</artifactId>
  <version>7.6.2</version>
</dependency>
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch</artifactId>
  <version>7.6.2</version>
</dependency>
```

操作步骤

以下介绍 Transport Client 连接 Elasticsearch 集群并查询 *test* 索引是否存在的代码示例。

```
import org.elasticsearch.action.ActionFuture;
import org.elasticsearch.action.admin.indices.exists.indices.IndicesExistsRequest;
import org.elasticsearch.action.admin.indices.exists.indices.IndicesExistsResponse;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.transport.TransportAddress;
import org.elasticsearch.transport.client.PreBuiltTransportClient;

import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.concurrent.ExecutionException;

public class Main {
    public static void main(String[] args) throws ExecutionException, InterruptedException,
UnknownHostException {
        String cluster_name = "xxx";
        String host1 = "x.x.x.x";
        String host2 = "y.y.y.y";
        Settings settings = Settings.builder()
            .put("client.transport.sniff", false)
            .put("cluster.name", cluster_name)
            .build();
        TransportClient client = new PreBuiltTransportClient(settings)
            .addTransportAddress(new TransportAddress(InetAddress.getByName(host1), 9300))
```

```
.addTransportAddress(new TransportAddress(InetAddress.getByName(host2), 9300));
IndicesExistsRequest indicesExistsRequest = new IndicesExistsRequest("test");
ActionFuture<IndicesExistsResponse> exists = client.admin().indices().exists(indicesExistsRequest);
System.out.println(exists.get().isExists());
}
}
```

其中，*cluster_name*为集群的名称；*host1*、*host2*为集群节点的内网IP地址，可通过GET `_cat/nodes`命令查看节点的IP地址。

2.4.5.4 通过 Spring Boot 接入 Elasticsearch 集群

本文介绍通过Spring Boot访问CSS集群的配置说明。Spring Boot接入集群有3种方式：

- **通过Spring Boot接入HTTP集群**：适用于非安全模式的集群、安全模式+HTTP协议的集群
- **使用Spring Boot接入HTTPS集群（不使用安全证书）**：适用于安全模式+HTTPS协议的集群
- **使用Spring Boot接入HTTPS集群（使用安全证书）**：适用于安全模式+HTTPS协议的集群

📖 说明

Spring Boot的具体使用方式请参见官方文档：<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>。

注意事项

- 建议Elasticsearch Rest High Level Client的版本和Elasticsearch的版本保持一致，例如需要访问的Elasticsearch集群版本是7.10.2，则使用的Elasticsearch Rest High Level Client客户端版本建议也是7.10.2。
- 本章节以2.5.5版本Spring Boot为例介绍Spring Boot接入集群的方式，对应的spring data elasticsearch版本是4.2.x。

准备工作

- CSS集群处于可用状态。
- 确保运行Java代码的服务器与CSS集群的网络是互通的。
- 根据集群选择的网络配置方式，获取集群的访问地址，具体操作请参见[网络配置](#)。
- 确认服务器已安装JDK1.8，JDK1.8官网下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>。
- 创建SpringBoot项目。
- 引入Java依赖。

其中7.10.2为Elasticsearch Java客户端的版本号。

- Maven方式引入：

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.5.5</version>
</parent>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-elasticsearch</artifactId>
</dependency>
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>7.10.2</version>
</dependency>
</dependencies>
```

通过 Spring Boot 接入 HTTP 集群

该场景适用于连接非安全模式的集群或是安全模式+HTTP协议的集群。

配置文件：

```
elasticsearch.url=host1:9200,host2:9200
//非安全集群不用配置如下两行。
elasticsearch.username=username
elasticsearch.password=password
```

表 2-28 参数说明

参数	描述
host	Elasticsearch集群的访问地址。
username	访问集群的用户名。
password	用户名对应的密码。

配置代码：

📖 说明

- com.xxx为项目目录，例如com.company.project。
- com.xxx.repository为仓库目录，通过extends org.springframework.data.elasticsearch.repository.ElasticsearchRepository进行具体定义。

```
package com.xxx.configuration;

import org.elasticsearch.client.RestHighLevelClient;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.elasticsearch.client.ClientConfiguration;
import org.springframework.data.elasticsearch.client.RestClients;
import org.springframework.data.elasticsearch.config.AbstractElasticsearchConfiguration;
import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;

@Configuration
@EnableElasticsearchRepositories(basePackages = "com.xxx.repository")
@ComponentScan(basePackages = "com.xxx")
public class Config extends AbstractElasticsearchConfiguration {

    @Value("${elasticsearch.url}")
    public String elasticsearchUrl;
```



```
//非安全集群不用配置如下两个参数。
@Value("${elasticsearch.username}")
public String elasticsearchUsername;

@Value("${elasticsearch.password}")
public String elasticsearchPassword;

@Override
@Bean
public RestHighLevelClient elasticsearchClient() {
    final ClientConfiguration clientConfiguration = ClientConfiguration.builder()
        .connectedTo(StringHostParse(elasticsearchUrl))
        //非安全集群无需配置withBasicAuth。
        .withBasicAuth(elasticsearchUsername, elasticsearchPassword)
        .build();

    return RestClients.create(clientConfiguration).rest();
}

private String[] StringHostParse(String hostAndPorts) {
    return hostAndPorts.split(",");
}
}
```

使用 Spring Boot 接入 HTTPS 集群（不使用安全证书）

该场景适用于不使用安全证书连接安全模式+HTTPS协议的集群。

配置文件：

```
elasticsearch.url=host1:9200,host2:9200
elasticsearch.username=username
elasticsearch.password=password
```

表 2-29 参数说明

参数	描述
host	Elasticsearch集群的访问地址。
username	访问集群的用户名。
password	用户名对应的密码。

配置代码：

说明

- com.xxx为项目目录，例如com.company.project。
- com.xxx.repository为仓库目录，通过extends org.springframework.data.elasticsearch.repository.ElasticsearchRepository进行具体定义。

```
package com.xxx.configuration;
import org.elasticsearch.client.RestHighLevelClient;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.elasticsearch.client.ClientConfiguration;
import org.springframework.data.elasticsearch.client.RestClients;
import org.springframework.data.elasticsearch.config.AbstractElasticsearchConfiguration;
import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;
```

```
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
@Configuration
@EnableElasticsearchRepositories(basePackages = "com.xxx.repository")
@ComponentScan(basePackages = "com.xxx")
public class Config extends AbstractElasticsearchConfiguration {
    @Value("${elasticsearch.url}")
    public String elasticsearchUrl;
    @Value("${elasticsearch.username}")
    public String elasticsearchUsername;
    @Value("${elasticsearch.password}")
    public String elasticsearchPassword;
    @Override
    @Bean
    public RestHighLevelClient elasticsearchClient() {
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        final ClientConfiguration clientConfiguration = ClientConfiguration.builder()
            .connectedTo(StringHostParse(elasticsearchUrl))
            .usingSsl(sc, new NullHostNameVerifier())
            .withBasicAuth(elasticsearchUsername, elasticsearchPassword)
            .build();
        return RestClients.create(clientConfiguration).rest();
    }
    private String[] StringHostParse(String hostAndPorts) {
        return hostAndPorts.split(",");
    }
    public static TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            @Override
            public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException
        {
            }
            @Override
            public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
            }
            @Override
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        }
    };
    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }
}
```

使用 Spring Boot 接入 HTTPS 集群（使用安全证书）

该场景适用于使用安全证书连接安全模式+HTTPS协议的集群。

1. 获取安全证书（CloudSearchService.cer）。
 - a. 登录云搜索服务控制台。
 - b. 选择“集群管理”进入集群列表。
 - c. 单击对应集群的名称，进入集群基本信息页面。
 - d. 在“基本信息”页面，单击“HTTPS访问”后面的“下载证书”。

图 2-4 下载证书

配置信息	
区域	
可用区	
虚拟私有云	vpc-
子网	subnet
安全组	dws 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	-- 绑定
HTTPS访问	开启 下载证书
内网访问IPv4地址	192

2. 转换安全证书（CloudSearchService.cer）。将下载的安全证书上传到客户端机器上，使用keytool工具将“.cer”证书转换成Java可以读取的“.jks”证书格式。

- 在Linux系统中，执行如下命令转换证书。

```
keytool -import -alias newname -keystore ./truststore.jks -file ./CloudSearchService.cer
```

- 在Windows系统中，执行如下命令转换证书。

```
keytool -import -alias newname -keystore .\truststore.jks -file .\CloudSearchService.cer
```

其中，*newname*是由用户自定义的证书名称。

该命令执行后，会提示设置证书密码，并确认密码。请保存该密码，后续接入集群会使用。

3. application.properties配置文件：

```
elasticsearch.url=host1:9200,host2:9200  
elasticsearch.username=username  
elasticsearch.password=password
```

表 2-30 参数说明

参数	描述
host	Elasticsearch 集群的访问地址。
username	访问集群的用户名。
password	用户名对应的密码。

4. 配置代码：

 说明

- com.xxx 为项目目录，例如 com.company.project。
- com.xxx.repository 为仓库目录，通过 extends org.springframework.data.elasticsearch.repository.ElasticsearchRepository 进行具体定义。

```
package com.xxx.configuration;
import org.elasticsearch.client.RestHighLevelClient;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.elasticsearch.client.ClientConfiguration;
import org.springframework.data.elasticsearch.client.RestClients;
import org.springframework.data.elasticsearch.config.AbstractElasticsearchConfiguration;
import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.TrustManagerFactory;
import javax.net.ssl.X509TrustManager;
@Configuration
@EnableElasticsearchRepositories(basePackages = "com.xxx.repository")
@ComponentScan(basePackages = "com.xxx")
public class Config extends AbstractElasticsearchConfiguration {
    @Value("${elasticsearch.url}")
    public String elasticsearchUrl;
    @Value("${elasticsearch.username}")
    public String elasticsearchUsername;
    @Value("${elasticsearch.password}")
    public String elasticsearchPassword;
    @Override
    @Bean
    public RestHighLevelClient elasticsearchClient() {
        SSLContext sc = null;
        try {
            TrustManager[] tm = {new MyX509TrustManager(cerFilePath, cerPassword)};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            sc.init(null, tm, new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }
        final ClientConfiguration clientConfiguration = ClientConfiguration.builder()
            .connectedTo(StringHostParse(elasticsearchUrl))
            .usingSsl(sc, new NullHostNameVerifier())
```

```
.withBasicAuth(elasticsearchUsername, elasticsearchPassword)
    .build();
return RestClients.create(clientConfiguration).rest();
}

private String[] StringHostParse(String hostAndPorts) {
    return hostAndPorts.split(",");
}

public static class MyX509TrustManager implements X509TrustManager {
    X509TrustManager sunJSSEX509TrustManager;
    MyX509TrustManager(String cerFilePath, String cerPassword) throws Exception {
        File file = new File(cerFilePath);
        if (!file.isFile()) {
            throw new Exception("Wrong Certification Path");
        }
        System.out.println("Loading KeyStore " + file + "...");
        InputStream in = new FileInputStream(file);
        KeyStore ks = KeyStore.getInstance("JKS");
        ks.load(in, cerPassword.toCharArray());
        TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509", "SunJSSE");
        tmf.init(ks);
        TrustManager[] tms = tmf.getTrustManagers();
        for (TrustManager tm : tms) {
            if (tm instanceof X509TrustManager) {
                sunJSSEX509TrustManager = (X509TrustManager) tm;
                return;
            }
        }
        throw new Exception("Couldn't initialize");
    }
    @Override
    public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
    }
    @Override
    public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
    }
    @Override
    public X509Certificate[] getAcceptedIssuers() {
        return new X509Certificate[0];
    }
}

public static class NullHostNameVerifier implements HostnameVerifier {
    @Override
    public boolean verify(String arg0, SSLSession arg1) {
        return true;
    }
}
}
```

其中，*cerFilePath*和*cerPassword*是生成的.jks证书的存放路径及其密码。

2.4.6 通过 Python 客户端接入 Elasticsearch 集群

本文介绍通过Python语言访问CSS集群的配置说明。

准备工作

- CSS集群处于可用状态。
- 确保运行Python代码的服务器与CSS集群的网络是互通的。

操作步骤

1. 安装Elasticsearch Python客户端，建议和Elasticsearch的版本保持一致，例如需要访问的集群版本是7.6.2，则安装7.6的Elasticsearch Python客户端。

```
pip install Elasticsearch==7.6
```

2. 创建Elasticsearch客户端并查看是否存在索引“test”。根据集群安全模式参考对应的示例代码。

- 非安全模式的集群

```
from elasticsearch import Elasticsearch

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password))
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch

es = ElasticFactory(["xxx.xxx.xxx.xxx"], "9200", None, None).create()
print(es.indices.exists(index='test'))
```

- 安全模式+HTTP协议的集群

```
from elasticsearch import Elasticsearch

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        addrs = []
        for host in self.host:
            addr = {'host': host, 'port': self.port}
            addrs.append(addr)

        if self.username and self.password:
            elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password))
        else:
            elasticsearch = Elasticsearch(addrs)
        return elasticsearch

es = ElasticFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(es.indices.exists(index='test'))
```

- 安全模式+HTTPS协议的集群

```
from elasticsearch import Elasticsearch
import ssl

class ElasticFactory(object):

    def __init__(self, host: list, port: str, username: str, password: str):
        self.port = port
        self.host = host
        self.username = username
        self.password = password

    def create(self) -> Elasticsearch:
        context = ssl.create_unverified_context()
```

```
    addrs = []
    for host in self.host:
        addr = {'host': host, 'port': self.port}
        addrs.append(addr)

    if self.username and self.password:
        elasticsearch = Elasticsearch(addrs, http_auth=(self.username, self.password),
        scheme="https", ssl_context=context)
    else:
        elasticsearch = Elasticsearch(addrs)
    return elasticsearch

es = ElasticFactory(["xxx.xxx.xxx.xxx"], "9200", "username", "password").create()
print(es.indices.exists(index='test'))
```

表 2-31 函数中的变量说明

参数	描述
host	ES集群的访问地址，当存在多个IP地址时，中间用“,” 隔开。
port	ES集群的连接端口，填写“9200”。
username	访问集群的用户名。
password	用户名对应的密码。

3. 通过Elasticsearch客户端创建集群索引。

```
mappings = {
    "settings": {
        "index": {
            "number_of_shards": number_of_shards,
            "number_of_replicas": 1,
        },
    },
    "mappings": {
        properties
    }
}
result = es.indices.create(index=index, body=mappings)
```

4. 通过Elasticsearch客户端查询上一步创建的索引。

```
body = {
    "query": {
        "match": {
            "查询字段": "查询内容"
        }
    }
}
result = es.search(index=index, body=body)
```

2.4.7 通过 MRS Hive 客户端接入 Elasticsearch 集群

Elasticsearch-Hadoop (ES-Hadoop) 连接器将Hadoop海量的数据存储和深度加工能力与Elasticsearch实时搜索和分析功能结合在一起。它能够让您快速深入了解大数据，并让您在Hadoop生态系统中更好地开展工作。

本文通过MRS的ES-Hadoop与CSS集群连接作为示例，你可以配置其他任何需要使用ES集群的应用。如有需要，也可以参考本文在其他服务中使用Elasticsearch，前提是要保证客户端与Elasticsearch集群网络连通。

准备工作

- CSS集群处于可用状态。
- 确保客户端与CSS集群的网络是互通的。
- 确保CSS集群和MRS集群在同一个区域、可用区、虚拟私有云和子网。

图 2-5 CSS 集群的区域等信息

配置信息

区域	
可用区	
虚拟私有云	vpc-
子网	subnet-
安全组	dws 更改安全组

操作步骤

1. 获取集群的内网访问地址。访问集群时，需要输入内网访问地址。
本文以内网IP地址方式访问集群为例，当使用其他网络配置访问集群时，仅访问地址有差异，不同访问地址的获取方式请参见[网络配置](#)。
 - a. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。
 - b. 在集群管理列表页面，选择需要访问的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。
当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。
2. 登录MRS集群节点，操作步骤请参见[登录集群节点](#)。
3. 在MRS集群节点上通过curl命令检查网络连通性，需要确保MRS集群的每个节点都能连通CSS集群。
 - 非安全模式的集群

```
curl -X GET http://<host>:<port>
```
 - 安全模式+HTTP协议的集群

```
curl -X GET http://<host>:<port> -u <user>:<password>
```
 - 安全模式+HTTPS协议的集群

```
curl -X GET https://<host>:<port> -u <user>:<password> -ik
```


表 2-32 变量说明

变量名	说明
<host>	集群中各节点的IP地址，当集群包含多个节点时，会存在多个IP地址，可以任选其中一个发送。
<port>	集群节点的访问端口号，一般为9200。
<user>	访问集群的用户名。
<password>	用户名对应的密码。 当密码中存在特殊字符时，请将用户名和密码加上英文双引号，例如“curl -u "user:password!" "http://<host>:<port>"”。

4. 下载ES-Hadoop的lib包，并解压zip包获取“elasticsearch-hadoop-x.x.x.jar”文件。版本需要与CSS集群版本一致，例如CSS集群是7.6.2版本，则建议下载elasticsearch-hadoop-7.6.2.zip。
5. 下载httpclient依赖包commons-httpclient:commons-httpclient-3.1.jar，其中3.1为版本号，建议用户根据实际需要选择。
6. 安装MRS客户端，如果已经安装可以跳过该步骤，未安装的请参见[安装客户端（3.x及之后版本）](#)。
7. 登录MRS客户端，将下载的ES-Hadoop和httpclient的jar依赖包上传到MRS客户端。
8. 在MRS客户端创建HDFS目录，将ES-Hadoop lib包和httpclient依赖包上传到该目录下。

```
hadoop fs -mkdir /tmp/hadoop-es
hadoop fs -put elasticsearch-hadoop-x.x.x.jar /tmp/hadoop-es
hadoop fs -put commons-httpclient-3.1.jar /tmp/hadoop-es
```
9. 从MRS客户端登录到Hive客户端，具体操作请参见[Hive客户端使用实践](#)。
10. 在Hive客户端，添加ES-Hadoop lib包和httpclient依赖包。该命令只对当前会话有效。

输入beeline或hive进入到执行界面，执行如下命令：

```
add jar hdfs:///tmp/hadoop-es/commons-httpclient-3.1.jar;
add jar hdfs:///tmp/hadoop-es/elasticsearch-hadoop-x.x.x.jar;
```

11. 在Hive客户端，创建Hive外表。
 - 非安全模式的集群

```
CREATE EXTERNAL table IF NOT EXISTS student(
  id BIGINT,
  name STRING,
  addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
  'es.nodes' = 'xxx.xxx.xxx.xxx:9200',
  'es.port' = '9200',
  'es.net.ssl' = 'false',
  'es.nodes.wan.only' = 'false',
  'es.nodes.discovery'='false',
  'es.input.use.sliced.partitions'='false',
  'es.resource' = 'student/_doc'
);
```
 - 安全模式+HTTP协议的集群

```
CREATE EXTERNAL table IF NOT EXISTS student(
  id BIGINT,
```

```
name STRING,  
addr STRING  
)  
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'  
TBLPROPERTIES(  
  'es.nodes' = 'xxx.xxx.xxx.xxx:9200',  
  'es.port' = '9200',  
  'es.net.ssl' = 'false',  
  'es.nodes.wan.only' = 'false',  
  'es.nodes.discovery'='false',  
  'es.input.use.sliced.partitions'='false',  
  'es.nodes.client.only'='true',  
  'es.resource' = 'student/_doc',  
  'es.net.http.auth.user' = 'username',  
  'es.net.http.auth.pass' = 'password'  
);
```

- 安全模式+HTTPS协议的集群

i. 获取安全证书（CloudSearchService.cer）。

- 1) 登录云搜索服务控制台。
- 2) 选择“集群管理”进入集群列表。
- 3) 单击对应集群的名称，进入集群基本信息页面。
- 4) 在“基本信息”页面，单击“HTTPS访问”后面的“下载证书”。

图 2-6 下载证书

配置信息

区域	
可用区	
虚拟私有云	vpc-
子网	subnet
安全组	dws 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	-- 绑定
HTTPS访问	开启 下载证书
内网访问IPv4地址	192

- ii. 转换安全证书（CloudSearchService.cer）。将下载的安全证书上传到客户端机器上，使用keytool工具将“.cer”证书转换成Java可以读取的“.jks”证书格式。

- 在Linux系统中，执行如下命令转换证书。
keytool -import -alias *newname* -keystore ./truststore.jks -file ./CloudSearchService.cer
- 在Windows系统中，执行如下命令转换证书。
keytool -import -alias *newname* -keystore .\truststore.jks -file .\CloudSearchService.cer

其中，*newname*是由用户自定义的证书名称。

该命令执行后，会提示设置证书密码，并确认密码。请保存该密码，后续接入集群会使用。

- iii. 将“.jks”文件分发到MRS集群的每个节点的相同路径，如“/tmp”，可以使用scp命令进行文件传输。同时，要确保omm用户有权限读取该文件，设置权限可以参考如下命令：

```
chown -R omm truststore.jks
```

- iv. 创建Hive外表。

```
CREATE EXTERNAL table IF NOT EXISTS student(
  id BIGINT,
  name STRING,
  addr STRING
)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES(
  'es.nodes' = 'https://xxx.xxx.xxx.xxx:9200',
  'es.port' = '9200',
  'es.net.ssl' = 'true',
  'es.net.ssl.truststore.location' = 'cerFilePath',
  'es.net.ssl.truststore.pass' = 'cerPassword',
  'es.nodes.wan.only' = 'false',
  'es.nodes.discovery'='false',
  'es.nodes.client.only'='true',
  'es.input.use.sliced.partitions'='false',
  'es.resource' = 'student/_doc',
  'es.net.http.auth.user' = 'username',
  'es.net.http.auth.pass' = 'password'
);
```

表 2-33 ES-Hadoop 参数说明

参数	默认值	描述
es.nodes	localhost	CSS集群的访问地址，可以集群列表页面查看“内网访问地址”。
es.port	9200	集群的访问端口号，一般为“9200”。
es.nodes.wan.only	false	是否进行节点嗅探。
es.nodes.discovery	true	是否禁用节点发现。

参数	默认值	描述
es.input.use.sliced.partitions	true	是否使用slice分区： <ul style="list-style-type: none"> • true：使用。 • false：不使用。 说明 设置为true，可能会导致索引在预读阶段的时间明显变长，有时会远远超出查询数据所耗费的时间。建议设置为false，以提高查询效率。
es.resource	NA	指定要读写的Index和type。
es.net.http.auth.user	NA	访问集群的用户名，只有启用了安全模式才需要配置此值。
es.net.http.auth.pass	NA	用户名所对应的密码，只有启用了安全模式才需要配置此值。
es.net.ssl	false	是否启用SSL，启用后需要配置安全证书信息。
es.net.ssl.truststore.location	NA	“.jks”证书文件的路径，如“file:///tmp/truststore.jks”。
es.nodes.client.only	false	是否配置了独立的Client节点的IP地址给es.nodes（即创建Elasticsearch集群时是否“启用Client节点”）。如果是，则需要将该值改为“true”，否则会报错找不到data节点。
es.net.ssl.truststore.pass	NA	“.jks”证书文件的密码。

更多ES-Hadoop配置项说明请参见[官方配置说明](#)。

- 在Hive客户端，插入数据。

```
INSERT INTO TABLE student VALUES (1, "Lucy", "address1"), (2, "Lily", "address2");
```

- 在Hive客户端，执行查询。

```
select * from student;
```

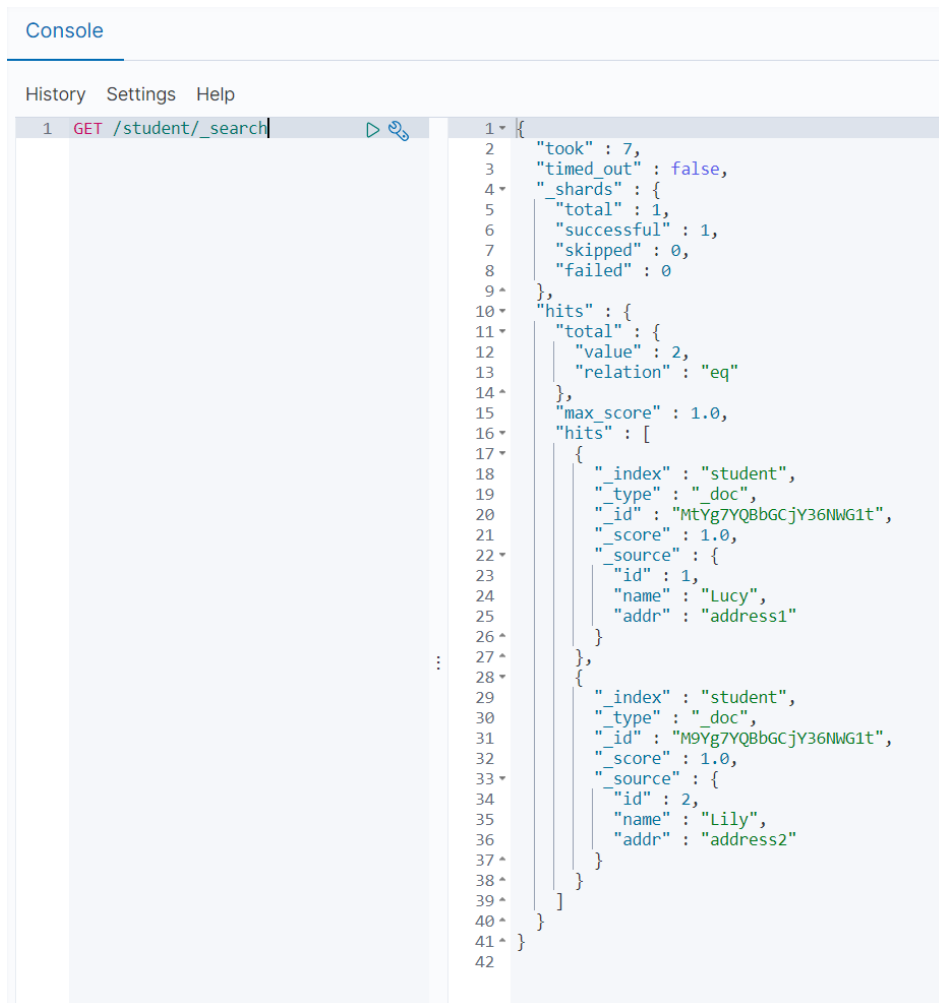
查询结果如下：

```
+-----+-----+-----+
| student.id | student.name | student.addr |
+-----+-----+-----+
| 1          | Lucy         | address1     |
| 2          | Lily        | address2     |
+-----+-----+-----+
2 rows selected (0.116 seconds)
```

- 登录CSS控制台，在“集群管理”页面，单击集群操作列“Kibana”，登录Kibana界面。
- 进入Kibana的“Dev Tools”页面执行查询命令，查看查询结果。

```
GET /student/_search
```

图 2-7 kibana 查询结果



```
1 GET /student/_search
2 {
3   "took" : 7,
4   "timed_out" : false,
5   "shards" : {
6     "total" : 1,
7     "successful" : 1,
8     "skipped" : 0,
9     "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 2,
14      "relation" : "eq"
15    },
16    "max_score" : 1.0,
17    "hits" : [
18      {
19        "_index" : "student",
20        "_type" : "_doc",
21        "_id" : "MtYg7YQBbGCjY36NWG1t",
22        "_score" : 1.0,
23        "_source" : {
24          "id" : 1,
25          "name" : "Lucy",
26          "addr" : "address1"
27        }
28      },
29      {
30        "_index" : "student",
31        "_type" : "_doc",
32        "_id" : "M9Yg7YQBbGCjY36NWG1t",
33        "_score" : 1.0,
34        "_source" : {
35          "id" : 2,
36          "name" : "Lily",
37          "addr" : "address2"
38        }
39      }
40    ]
41  }
42 }
```

2.4.8 通过 Go 客户端接入 Elasticsearch 集群

本文介绍通过Go语言访问CSS集群的配置说明。

准备工作

- CSS集群处于可用状态。
- 确保运行Go代码的服务器与CSS集群的网络是互通的。
- 确认服务器已安装Go，Go官网下载地址：<https://go.dev/dl/>。

连接非安全集群

连接非安全集群，示例代码如下：

```
package main

import (
    "github.com/elastic/go-elasticsearch/v7"
    "log"
)

func main() {
    cfg := elasticsearch.Config{
```

```
Addresses: []string{
    "http://HOST:9200/",
},
}

es, _ := elasticsearch.NewClient(cfg)
log.Println(es.Info())
}
```

其中，HOST为集群节点的内网IP地址。

连接安全集群

- 连接未开启https的安全集群，示例如下：

```
package main

import (
    "github.com/elastic/go-elasticsearch/v7"
    "log"
)

func main() {
    cfg := elasticsearch.Config{
        Addresses: []string{
            "http://HOST:9200/",
        },
        Username: "USERNAME",
        Password: "PASSWORD",
    }

    es, _ := elasticsearch.NewClient(cfg)
    log.Println(es.Info())
}
```

- 连接开启https的安全集群，不使用证书，示例代码如下：

```
package main

import (
    "crypto/tls"
    "github.com/elastic/go-elasticsearch/v7"
    "log"
    "net/http"
)

func main() {
    cfg := elasticsearch.Config{
        Addresses: []string{
            "https://HOST:9200/",
        },
        Username: "USERNAME",
        Password: "PASSWORD",
        Transport: &http.Transport{
            TLSClientConfig: &tls.Config{
                InsecureSkipVerify: true,
            },
        },
    }

    es, _ := elasticsearch.NewClient(cfg)
    log.Println(es.Info())
}
```

- 连接开启https的安全集群，使用证书，示例代码如下：

```
package main

import (
    "crypto/tls"
    "crypto/x509"
    "flag"
)
```

```
"github.com/elastic/go-elasticsearch/v7"
"io/ioutil"
"log"
"net"
"net/http"
"time"
)

func main() {
    insecure := flag.Bool("insecure-ssl", false, "Accept/Ignore all server SSL certificates")
    flag.Parse()

    // Get the SystemCertPool, continue with an empty pool on error
    rootCAs, _ := x509.SystemCertPool()
    if rootCAs == nil {
        rootCAs = x509.NewCertPool()
    }

    // Read in the cert file
    certs, err := ioutil.ReadFile("/tmp/CloudSearchService.cer")
    if err != nil {
        log.Fatalf("Failed to append %q to RootCAs: %v", "xxx", err)
    }

    // Append our cert to the system pool
    if ok := rootCAs.AppendCertsFromPEM(certs); !ok {
        log.Println("No certs appended, using system certs only")
    }

    config := elasticsearch.Config{
        Addresses: []string{
            "https://HOST:9200/",
        },
        Username: "USERNAME",
        Password: "PASSWORD",
        Transport: &http.Transport{
            MaxIdleConnsPerHost: 10,
            ResponseHeaderTimeout: time.Second,
            DialContext: (&net.Dialer{
                Timeout: 30 * time.Second,
                KeepAlive: 30 * time.Second,
            }).DialContext,
            TLSClientConfig: &tls.Config{
                InsecureSkipVerify: *insecure,
                RootCAs: rootCAs,
            },
        },
    }
    es, _ := elasticsearch.NewClient(config)
    log.Println(elasticsearch.Version)
    log.Println(es.Info())
}
```

表 2-34 函数中的变量说明

参数	描述
HOST	ES集群的访问地址，当存在多个IP地址时，中间用“，”隔开。
USERNAME	访问集群的用户名。
PASSWORD	用户名对应的密码。

运行代码

根据集群类型将以上适配的代码写入到“EsTest.gc”文件并存放到一个单独的目录，在该目录执行以下命令运行代码。

```
go env -w GO111MODULE=on
go env -w GOPROXY=https://repo.huaweicloud.com/repository/goproxy/
go env -w GONOSUMDB=*

go mod init test
go mod tidy
go run EsTest.go
```

2.5 导入数据至 Elasticsearch 集群

2.5.1 Elasticsearch 集群导入数据方式

Elasticsearch 集群导入数据方式介绍

Elasticsearch 集群支持通过多种方式导入数据，基于业务使用场景可以自行选择导入数据的方式，具体请参见表2-35。在导入数据前可以根据需要选择是否对Elasticsearch 集群进行导入性能增强，具体操作请参见[增强Elasticsearch 集群数据导入性能](#)。

表 2-35 Elasticsearch 集群导入数据的方式

导入数据方式	适用场景	支持数据格式	相关文档
数据处理管道 Logstash	Logstash开源服务器端实时数据处理管道，支持多个来源采取数据。适用场景多，比如日志数据、监控数据、metric数据等流数据。	JSON、CSV、文本等多种格式	使用自建Logstash导入数据到Elasticsearch
开源Elasticsearch API	使用开源Elasticsearch API导入数据，使用灵活，适用于开发自己写的应用代码场景。	JSON	使用开源Elasticsearch API导入数据到Elasticsearch
云数据迁移服务 CDM	CDM向导式页面。适用于批数据迁移，比如数据存放在OBS或者Oracle数据库中，推荐使用CDM比较方便。	JSON	使用CDM导入数据到Elasticsearch
数据复制服务DRS	DRS用于数据库在线迁移和实时同步数据的云服务。	关系型数据库	使用DRS从数据库导入数据至Elasticsearch

使用 DRS 从数据库导入数据至 Elasticsearch

数据复制服务（Data Replication Service，简称DRS）是一种易用、稳定、高效、用于数据库在线迁移和数据库实时同步的云服务。实时同步是指在不同的系统之间，将数据通过同步技术从一个数据源复制到其他数据库，并保持一致，实现关键业务的数据实时流动。

使用DRS支持将多种关系型数据库的数据导入至Elasticsearch集群，源数据库和目标集群的版本支持情况请参见表2-36。

表 2-36 使用 DRS 从数据库导入数据至 Elasticsearch

数据导入场景	源数据库	目标Elasticsearch集群	参考文档
将RDS for MySQL数据库的数据导入到CSS Elasticsearch集群	RDS for MySQL 5.5、5.6、5.7、8.0 版本	ElasticSearch 5.5、6.2、6.5、7.1、7.6、7.9、7.10版本	将MySQL同步到CSS/ES
将GaussDB(for MySQL)数据库的数据导入到CSS Elasticsearch集群	GaussDB(for MySQL)主备实例	ElasticSearch 5.5、6.2、6.5、7.1、7.6、7.9、7.10版本	将GaussDB(for MySQL)同步到CSS/ES
将自建MySQL数据库的数据导入到CSS Elasticsearch集群	本地或ECS自建MySQL数据库 5.5、5.6、5.7、8.0 版本	ElasticSearch 5.5、6.2、6.5、7.1、7.6、7.9、7.10版本	将MySQL同步到CSS/ES

2.5.2 使用自建 Logstash 导入数据到 Elasticsearch

云搜索服务支持使用自建Logstash将其收集的数据迁移到Elasticsearch和OpenSearch中，方便用户通过CSS搜索引擎高效管理和获取数据。数据文件支持JSON、CSV等格式。

Logstash是开源的服务器端数据处理管道，能够同时从多个来源采集数据、转换数据，然后将数据发送到Elasticsearch或OpenSearch中。Logstash的官方文档请参见：<https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html>。

数据导入分为如下2种场景：

- [Logstash部署在外网时导入数据](#)
- [Logstash部署在弹性云服务器上时导入数据](#)

前提条件

- 为方便操作，建议采用Linux操作系统的机器部署Logstash。
- Logstash的下载路径为：<https://www.elastic.co/cn/downloads/logstash-oss>

📖 说明

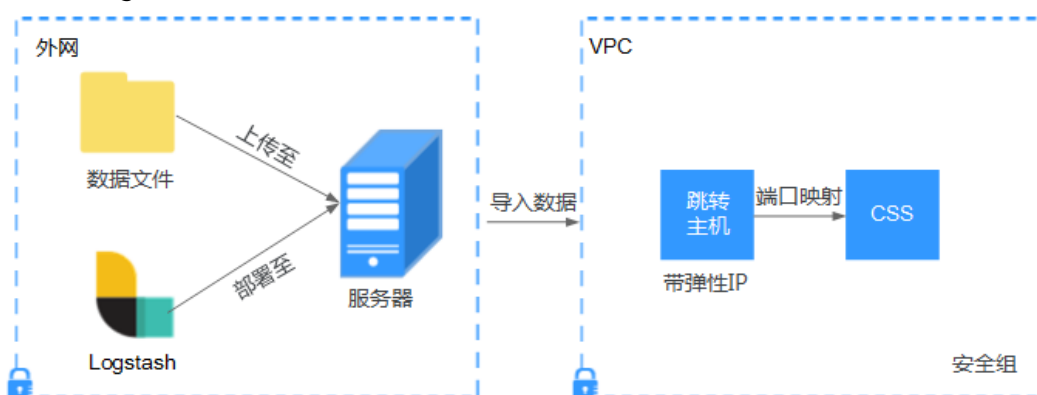
Logstash要求使用OSS版本，选择和CSS一致版本。

- 安装Logstash之前，需要先安装JDK。在Linux操作系统中，您可以执行**yum -y install java-1.8.0**命令直接安装1.8.0版本JDK。在Windows操作系统中，您可以访问[JDK官网](#)，下载符合操作系统版本的JDK，并根据指导安装。
- 安装完Logstash后，再根据如下步骤导入数据。安装Logstash的操作指导，请参见：<https://www.elastic.co/guide/en/logstash/current/installing-logstash.html>
- 在“**Logstash部署在弹性云服务器上时导入数据**”场景中，请确保此弹性云服务器与接入的Elasticsearch集群在同一个VPC下。

Logstash 部署在外网时导入数据

当Logstash部署在外网时，导入数据的流程说明如**图2-8**所示。

图 2-8 Logstash 部署在外网时导入数据示意图



1. 创建一个跳转主机，并按如下要求进行配置。
 - 跳转主机为一台Linux操作系统的弹性云服务器，且已绑定弹性IP。
 - 跳转主机与CSS集群在同一虚拟私有云下。
 - 已开放跳转主机的本地端口，用于SSH转发，能够从本地端口转发至CSS集群某一节点的9200端口。
 - 关于跳转主机的本地端口转发配置，请参见[SSH官方文档](#)。
2. 使用PuTTY，通过弹性IP登录已创建的跳转主机。
3. 执行如下命令进行端口映射，将发往跳转主机对外开放端口的请求转发到待导入数据的集群中。

```
ssh -g -L <跳转主机的本地端口:节点的内网访问地址和端口号> -N -f root@<跳转主机的私网IP地址>
```

说明

- <跳转主机的本地端口>：为步骤1中的端口。
- <节点的内网访问地址和端口号>：为集群中某一节点的内网访问地址和端口号。当该节点出现故障时，将导致命令执行失败。如果集群包含多个节点，可以将<节点的内网访问地址和端口号>替换为集群中另一节点的内网访问地址和端口号；如果集群只包含一个节点，则需要将该节点修复之后再次执行命令进行端口映射。
- <跳转主机的私网IP地址>：打开弹性云服务器管理控制台，从“IP地址”列中获取标有“私网”对应的IP地址。

例如：跳转主机对外开放的端口号为9200，节点的内网访问地址和端口号为192.168.0.81:9200，跳转主机的私网IP地址为192.168.0.227，需要执行如下命令进行端口映射。

```
ssh -g -L 9200:192.168.0.81:9200 -N -f root@192.168.0.227
```

4. 登录部署了Logstash的服务器，将需要进行操作的数据文件存储至此服务器中。
例如，需要导入的数据文件“access_20181029_log”，文件存储路径为“/tmp/access_log/”，此数据文件中包含的数据如下所示：

📖 说明

文件存储路径中的access_log文件夹如果不存在，用户可以自建。

All	Heap used for segments		18.6403	MB
All	Heap used for doc values		0.119289	MB
All	Heap used for terms		17.4095	MB
All	Heap used for norms		0.0767822	MB
All	Heap used for points		0.225246	MB
All	Heap used for stored fields		0.809448	MB
All	Segment count		101	
All	Min Throughput	index-append	66232.6	docs/s
All	Median Throughput	index-append	66735.3	docs/s
All	Max Throughput	index-append	67745.6	docs/s
All	50th percentile latency	index-append	510.261	ms

5. 在部署Logstash的服务器中，执行如下命令在Logstash的安装目录下新建配置文件logstash-simple.conf。

```
cd /<Logstash的安装目录>/
vi logstash-simple.conf
```

6. 在配置文件logstash-simple.conf中输入如下内容。

```
input {
  <数据所在的位置>
}
filter {
  <数据的相关处理>
}
output {
  elasticsearch {
    hosts => "<跳转主机的公网IP地址>:<跳转主机对外开放的端口号>"
  }
}
```

- input: 指定数据的来源。实际请根据用户的具体情况来设置。input参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/input-plugins.html>。
- filter: 指定对数据进行处理的方式。例如，对日志进行了提取和处理，将非结构化信息转换为结构化信息。filter参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/filter-plugins.html>。
- output: 指定数据的目的地。output参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/output-plugins.html>。<跳转主机的公网IP地址>请从弹性云服务器管理控制台的“IP地址”列中获取标有“弹性公网”对应的IP地址。<跳转主机对外开放的端口号>即为步骤1中的端口，例如：9200。

以步骤4中“/tmp/access_log/”的数据文件为例，输入数据文件从首行开始，且过滤条件保持为空，即不做任何数据处理操作。跳转主机的公网IP和端口号为“192.168.0.227:9200”。导入数据的索引名称为“myindex”。配置文件的示例如下所示，配置文件按实际数据情况修改完成后，输入“:wq”保存。

```
input {
  file {
    path => "/tmp/access_log/*"
    start_position => "beginning"
  }
}
filter {
}
```

```
output {
  elasticsearch {
    hosts => "192.168.0.227:9200"
    index => "myindex"
  }
}
```

📖 说明

如果在使用中出现license相关的报错，可以尝试设置ilm_enabled => false。
如果集群开启了安全模式，则需要先下载证书。

- 在云搜索服务管理控制台。
- 在“集群管理”页面，单击需要下载证书的集群名称，进入集群基本信息页面。
- 在集群基本信息页面下载证书。

图 2-9 下载证书

配置信息

区域	
可用区	
虚拟私有云	vpc-
子网	subne
安全组	dws 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	-- 绑定
HTTPS访问	开启 下载证书
内网访问IPv4地址	

- 将下载的证书存放到部署logstash服务器中。
- 修改配置文件logstash-simple.conf。
以步骤4中“/tmp/access_log/”的数据文件为例，输入数据文件从首行开始，且过滤条件保持为空，即不做任何数据处理操作。跳转主机的公网IP和

端口号为“192.168.0.227:9200”。导入数据的索引名称为“myindex”，证书存放路径为“/logstash/logstash6.8/config/CloudSearchService.cer”。配置文件的示例如下所示，配置文件按实际数据情况修改完成后，输入“:wq”保存。

```
input{
  file {
    path => "/tmp/access_log/*"
    start_position => "beginning"
  }
}
filter {
}
output{
  elasticsearch{
    hosts => ["https://192.168.0.227:9200"]
    index => "myindex"
    user => "admin"
    password => "*****"
    cacert => "/logstash/logstash6.8/config/CloudSearchService.cer"
    manager_template => false
    ilm_enabled => false
    ssl => true
    ssl_certificate_verification => false
  }
}
```

📖 说明

password：登录安全集群的密码。

7. 执行如下命令将Logstash收集的数据导入到集群中。

```
./bin/logstash -f logstash-simple.conf
```

📖 说明

此命令需要在存放logstash-simple.conf文件的目录下执行。例如，logstash-simple.conf文件存放在/root/logstash-7.1.1/，则需要先进入该路径，再执行此命令。

8. 登录云搜索服务管理控制台。
9. 在左侧导航栏中，对应的集群类型，进入集群管理列表页面。
10. 在集群列表页面中，单击待导入数据的集群“操作”列的“Kibana”。
11. 在Kibana的左侧导航中选择“Dev Tools”，进入Console界面。
12. 在已打开的Kibana的Console界面，通过搜索获取已导入的数据。

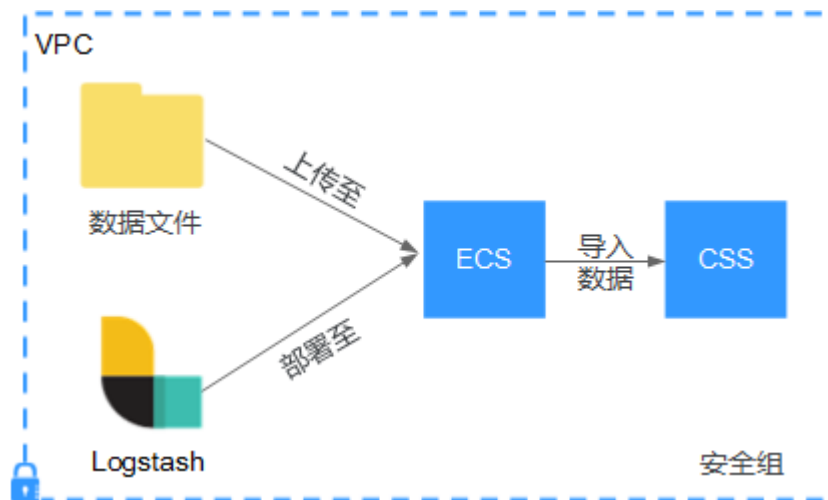
在Kibana控制台，输入如下命令，搜索数据。查看搜索结果，如果数据与导入数据一致，表示数据文件的数据已导入成功。

```
GET myindex/_search
```

Logstash 部署在弹性云服务器上时导入数据

当Logstash部署在同一VPC的弹性云服务时，导入数据的流程说明如[图2-10](#)所示。

图 2-10 Logstash 部署在弹性云服务器上时导入数据示意图



1. 确保已部署Logstash的弹性云服务器与待导入数据的集群在同一虚拟私有云下，已开放安全组的9200端口的的外网访问权限，且弹性云服务器已绑定弹性IP。

说明

- 如果同一个VPC内有多台服务器，只要其中一台绑定了弹性IP，其他的服务器可以不需要绑定弹性IP。通过绑定弹性IP的节点跳转到部署Logstash的节点即可。
- 如果有专线或者VPN，也不需要绑定弹性IP。

2. 使用PuTTY登录弹性云服务器。

例如此服务器中存储了需要导入的数据文件“access_20181029_log”，文件存储路径为“/tmp/access_log/”，此数据文件中包含的数据如下所示：

All	Heap used for segments		18.6403	MB
All	Heap used for doc values		0.119289	MB
All	Heap used for terms		17.4095	MB
All	Heap used for norms		0.0767822	MB
All	Heap used for points		0.225246	MB
All	Heap used for stored fields		0.809448	MB
All	Segment count		101	
All	Min Throughput	index-append	66232.6	docs/s
All	Median Throughput	index-append	66735.3	docs/s
All	Max Throughput	index-append	67745.6	docs/s
All	50th percentile latency	index-append	510.261	ms

3. 执行如下命令在Logstash的安装目录下新建配置文件logstash-simple.conf。

```
cd /<Logstash的安装目录>
vi logstash-simple.conf
```

在配置文件logstash-simple.conf中输入如下内容。

```
input {
  数据所在的位置
}
filter {
  数据的相关处理
}
output {
  elasticsearch{
    hosts => "<节点的网内网访问地址和端口号>"
  }
}
```

- input: 指定数据的来源。实际请根据用户的具体情况来设置。input参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/input-plugins.html>。

- filter: 对日志进行了提取和处理, 将非结构化信息转换为结构化信息。filter 参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/filter-plugins.html>。
- output: 指定数据的目的地。output 参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/output-plugins.html>。<节点的内网访问地址和端口号>为集群中节点的内网访问地址和端口号。

当集群包含多个节点时, 为了避免节点故障, 建议将上述命令中<节点的内网访问地址和端口号>替换为该集群中多个节点的内网访问地址和端口号, 多个节点的内网访问地址和端口号之间用英文逗号隔开, 填写格式请参见如下示例。

```
hosts => ["192.168.0.81:9200","192.168.0.24:9200"]
```

当集群只包含一个节点时, 填写格式请参见如下示例。

```
hosts => "192.168.0.81:9200"
```

以步骤2中“/tmp/access_log/”的数据文件为例, 输入数据文件从首行开始, 且过滤条件保持为空, 即不做任何数据处理操作。需导入数据的集群, 其节点内网访问地址和端口号为“192.168.0.81:9200”。导入数据的索引名称为“myindex”。配置文件的示例如下所示, 配置文件按实际数据情况修改完成后, 输入“:wq”保存。

```
input {
  file {
    path => "/tmp/access_log/*"
    start_position => "beginning"
  }
}
filter {
}
output {
  elasticsearch {
    hosts => "192.168.0.81:9200"
    index => "myindex"
  }
}
```

如果集群开启了安全模式, 则需要先下载证书。

- a. 在集群基本信息页面下载证书。

图 2-11 下载证书

配置信息

区域	
可用区	
虚拟私有云	vpc-
子网	subn
安全组	dws 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	-- 绑定
HTTPS访问	开启 下载证书
内网访问IPv4地址	

- 将下载的证书存放部署logstash服务器中。
- 修改配置文件logstash-simple.conf。

以步骤2中“/tmp/access_log/”的数据文件为例，输入数据文件从首行开始，且过滤条件保持为空，即不做任何数据处理操作。跳转主机的公网IP和端口号为“192.168.0.227:9200”。导入数据的索引名称为“myindex”，证书存放路径为“/logstash/logstash6.8/config/CloudSearchService.cer”。配置文件的示例如下所示，配置文件按实际数据情况修改完成后，输入“:wq”保存。

```
input{
  file {
    path => "/tmp/access_log/"
    start_position => "beginning"
  }
}
filter {
}
output{
  elasticsearch{
    hosts => ["https://192.168.0.227:9200"]
    index => "myindex"
    user => "admin"
    password => "*****"
    cacert => "/logstash/logstash6.8/config/CloudSearchService.cer"
  }
}
```



```
manager_template => false
ilm_enabled => false
ssl => true
ssl_certificate_verification => false
}
}
```

说明

password: 登录安全集群的密码。

4. 执行如下命令将Logstash收集的弹性云服务器的数据导入到集群中。
./bin/logstash -f logstash-simple.conf
5. 登录云搜索服务管理控制台。
6. 在左侧导航栏中，选择对应的集群类型，进入集群管理列表页面。
7. 在集群列表页面中，单击待导入数据的集群“操作”列的“Kibana”。
8. 在Kibana的左侧导航中选择“Dev Tools”，进入Console界面。
9. 在已打开的Kibana的Console界面，通过搜索获取已导入的数据。
在Kibana控制台，输入如下命令，搜索数据。查看搜索结果，如果数据与导入数据一致，表示数据文件的数据已导入成功。

```
GET myindex/_search
```

2.5.3 使用开源 Elasticsearch API 导入数据到 Elasticsearch

云搜索服务支持在Kibana或者ECS服务器上使用开源Elasticsearch API将数据导入到Elasticsearch集群中，数据文件支持JSON等格式。

- [在Kibana上使用开源Elasticsearch API导入数据](#)：使用POST命令导入数据。
- [在ECS服务器上使用开源Elasticsearch API导入数据](#)：使用Curl命令导入数据。

在 Kibana 上使用开源 Elasticsearch API 导入数据

在Kibana上支持通过POST命令使用开源Elasticsearch API导入单条数据。

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择“集群管理 > Elasticsearch”，进入集群管理列表页面。
3. 选择待导入数据的集群，单击操作列“Kibana”，登录Kibana。
4. 单击左侧导航栏的“Dev Tools”进入操作页面。
5. 在Kibana操作页面，执行命令查看集群是否存在索引。

```
GET _cat/indices?v
```

- 当待导入数据的集群已存在可用的索引时，则不需要再创建索引，直接执行步骤7。
- 当待导入数据的集群不存在可用的索引时，则需要执行下一步创建索引。

6. 在Kibana执行命令，创建待导入数据的索引，并指定自定义映射来定义数据类型。

例如执行如下命令，创建索引“my_store”。

Elasticsearch版本<7.x的命令

```
PUT /my_store
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "products": {
```

```
    "properties": {
      "productName": {
        "type": "text"
      },
      "size": {
        "type": "keyword"
      }
    }
  }
}
```

Elasticsearch版本≥7.x的命令

```
PUT /my_store
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "productName": {
        "type": "text"
      },
      "size": {
        "type": "keyword"
      }
    }
  }
}
```

7. 在Kibana执行命令导入数据，以导入一条数据为例，执行如下命令。

Elasticsearch版本<7.x的命令

```
POST /my_store/products/_bulk
{"index":{}}
{"productName":"Latest art shirts for women in 2017 autumn","size":"L"}
```

Elasticsearch版本≥7.x的命令

```
POST /my_store/_bulk
{"index":{}}
{"productName":"Latest art shirts for women in 2017 autumn","size":"L"}
```

返回结果如[图2-12](#)所示，当返回结果信息中“errors”字段的值为“false”时，表示导入数据成功。

图 2-12 返回消息

```
1 {
2   "took": 42,
3   "errors": false,
4   "items": [
5     {
6       "index": {
7         "_index": "my_store",
8         "_type": "products",
9         "_id": "AWTGbHT7BwpN-hb3LKau",
10        "_version": 1,
11        "result": "created",
12        "_shards": {
13          "total": 2,
14          "successful": 2,
15          "failed": 0
16        },
17        "created": true,
18        "status": 201
19      }
20    }
21  ]
22 }
```

在 ECS 服务器上使用开源 Elasticsearch API 导入数据

在ECS服务器上支持通过Curl命令使用开源Elasticsearch API导入JSON数据文件。

本案例以非安全模式的集群为例介绍Curl命令导入数据，如果是安全模式的集群访问集群的命令请参见[通过Curl命令行接入Elasticsearch集群](#)。

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择“集群管理 > Elasticsearch”，进入集群管理列表页面。
3. 在集群管理列表页面，选择待导入数据的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>和端口<port>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。

当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。

4. 登录即将接入集群的弹性云服务器ECS。

弹性云服务器的使用指导请参见[快速购买和使用Linux ECS](#)。

5. 将JSON数据文件上传至ECS。

例如，将如下数据保存为json格式的文件，上传到ECS。

Elasticsearch版本<7.x的数据

```
{"index": {"_index": "my_store", "_type": "products"}}
{"productName": "2019秋装新款文艺衬衫女装", "size": "M"}
{"index": {"_index": "my_store", "_type": "products"}}
{"productName": "2019秋装新款文艺衬衫女装", "size": "L"}
```

Elasticsearch版本≥7.x或Opensearch版本的数据

```
{"index": {"_index": "my_store"}}
{"productName": "2019秋装新款文艺衬衫女装", "size": "M"}
```

```
{"index": {"_index": "my_store"}}
{"productName": "2019秋装新款文艺衬衫女装", "size": "L"}
```

6. 在ECS中存放JSON数据文件的路径下，执行如下命令将JSON数据文件导入到Elasticsearch集群。

```
curl -X PUT "http://{{Private network address and port number of the node}}/_bulk" -H 'Content-Type: application/json' --data-binary @test.json
```

其中，**{Private network address and port number of the node}**需替换为集群中节点的内网访问地址和端口号。当该节点出现故障时，将导致命令执行失败，如果集群包含多个节点，可以将**{Private network address and port number of the node}**替换为集群中另一节点的内网访问地址和端口号，如果集群只包含一个节点，则需要将该节点修复之后再次执行命令进行导入数据。“test.json”为导入数据的json文件。

📖 说明

其中，-X参数的参数值为命令，如“-X PUT”，-H参数的参数值为消息头，如“-H 'Content-Type: application/json' --data-binary @test.json”。添加的-k参数时，请勿将-k参数放置在参数与参数值之间。

示例：将“test.json”数据文件中的数据导入至Elasticsearch集群，此集群未进行通信加密，其中一个节点内网访问地址为“192.168.0.90”，端口号为“9200”。

- a. 执行如下命令，创建“my_store”索引。

Elasticsearch版本<7.x的命令

```
curl -X PUT http://192.168.0.90:9200/my_store -H 'Content-Type: application/json' -d '{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "products": {
      "properties": {
        "productName": {
          "type": "text"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
}'
```

Elasticsearch版本≥7.x的命令

```
curl -X PUT http://192.168.0.90:9200/my_store -H 'Content-Type: application/json' -d '{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "productName": {
        "type": "text"
      },
      "size": {
        "type": "keyword"
      }
    }
  }
}'
```

- b. 执行如下命令，导入“test.json”文件中的数据。

```
curl -X PUT "http://192.168.0.90:9200/_bulk" -H 'Content-Type: application/json' --data-binary @test.json
```

本案例回显如下信息，表示数据导入成功。

```
{
  "took": 204,
  "errors": false,
  "items": [
    {
      "index": {
        "_index": "my_store",
        "_type": "_doc",
        "_id": "DJQkBlwBbJvUd2769Wi-",
        "_version": 1,
        "result": "created",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0,
          "_seq_no": 0,
          "_primary_term": 1,
          "status": 201
        }
      }
    },
    {
      "index": {
        "_index": "my_store",
        "_type": "_doc",
        "_id": "DZQkBlwBbJvUd2769Wi_",
        "_version": 1,
        "result": "created",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0,
          "_seq_no": 1,
          "_primary_term": 1,
          "status": 201
        }
      }
    }
  ]
}
```

2.5.4 使用 CDM 导入数据到 Elasticsearch

云搜索服务支持通过云数据迁移服务CDM的向导式界面，将存储在Oracle数据库或对象存储服务OBS中的数据导入到Elasticsearch或OpenSearch集群中，数据文件支持JSON格式。

表 2-37 使用 CDM 导入数据至 CSS 服务

数据导入场景	源数据	目标集群
将Oracle数据库的数据导入到CSS服务	本地或第三方Oracle数据库	ElasticSearch 5.5、6.2、6.5、7.1、7.6、7.9、7.10 版本 OpenSearch 1.3.6版本
将OBS的数据导入到CSS服务	OBS桶中的JSON数据	ElasticSearch 5.5、6.2、6.5、7.1、7.6、7.9、7.10 版本 OpenSearch 1.3.6版本

准备工作

1. 确认源数据。

例如，源数据为如下json格式的文件。

Elasticsearch版本<7.x的数据

```
{
  "index": {
    "_index": "my_store",
    "_type": "products"
  }
  "productName": "2019秋装新款文艺衬衫女装",
  "size": "M"
},
{
  "index": {
    "_index": "my_store",
    "_type": "products"
  }
  "productName": "2019秋装新款文艺衬衫女装",
  "size": "L"
}
```

Elasticsearch版本≥7.x或Opensearch版本的数据

```
{
  "index": {
    "_index": "my_store"
  }
  "productName": "2019秋装新款文艺衬衫女装",
  "size": "M"
},
{
  "index": {
    "_index": "my_store"
  }
  "productName": "2019秋装新款文艺衬衫女装",
  "size": "L"
}
```

2. 获取源数据信息。

- 当源数据是Oracle数据库时，需要获取Oracle数据库的IP、数据库名、用户名和密码。
- 当源数据是OBS桶中的JSON数据时，需要获取OBS的访问域名、端口，以及AK、SK。

3. 当源数据是Oracle数据库时，需要确保Oracle可通过公网IP访问，或者已经建立好了企业内部数据中心到华为云的VPN通道或专线。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择对应集群类型，进入集群管理列表页面。
3. 在集群管理列表页面，选择待导入数据的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>和端口<port>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。

当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。

4. 在集群列表，单击待导入数据的集群操作列的“Kibana”，登录Kibana。
5. 在Kibana的左侧导航中选择“Dev Tools”，进入操作页面。
6. 在Kibana执行命令查看集群是否存在索引。
GET _cat/indices?v
 - 当待导入数据的集群已存在可用的索引时，则不需要再创建索引，直接执行步骤8。
 - 当待导入数据的集群不存在可用的索引时，则需要执行下一步创建索引。
7. 在Kibana执行命令创建待导入数据的索引，并指定自定义映射来定义数据类型。例如：执行如下命令，创建索引“demo”。

Elasticsearch版本<7.x的命令

```
PUT /demo
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "products": {
      "properties": {
        "productName": {
          "type": "text",
          "analyzer": "ik_smart"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
}
```

Elasticsearch版本≥7.x或Opensearch版本的命令

```
PUT /demo
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "productName": {
        "type": "text",
        "analyzer": "ik_smart"
      },
      "size": {
        "type": "keyword"
      }
    }
  }
}
```

执行成功后显示如下：

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "demo"
}
```

8. 登录云数据迁移CDM管理控制台，通过CDM集群将Oracle或OBS的数据导入Elasticsearch或OpenSearch集群。
 - 当源数据是Oracle数据库时，请参考[Oracle数据迁移到云搜索服务](#)完成数据导入。
 - 当源数据是OBS桶中的JSON数据时，请参考[OBS数据迁移到云搜索服务](#)完成数据导入。
9. 当数据迁移完成后，重新进入Elasticsearch或OpenSearch集群的Kibana操作页面，搜索已导入的数据。

执行如下命令，搜索数据。查看搜索结果，如果数据与源数据一致，表示数据导入成功。

```
GET demo/_search
```

“demo”为创建的索引名称，需根据实际情况填写。

执行成功后显示如下：

```
{
  "took": 18,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 2,
    "max_score": 1,
    "hits": [
      {
        "_index": "demo",
        "_type": "products",
        "_id": "g6UepnEBuvdFwWkRmn4V",
        "_score": 1,
        "_source": {
          "size": """"size": "L""""",
          "productName": """"{"productName": "2019秋装新款文艺衬衫女装""""
        }
      },
      {
        "_index": "demo",
        "_type": "products",
        "_id": "hKUepnEBuvdFwWkRmn4V",
        "_score": 1,
        "_source": {
          "size": """"size": "M""""",
          "productName": """"{"productName": "2019秋装新款文艺衬衫女装""""
        }
      }
    ]
  }
}
```

2.5.5 增强 Elasticsearch 集群数据导入性能

特性介绍

数据导入性能增强为云搜索服务自主研发的特性，通过优化Bulk路由、文本索引加速、分词加速等方式，有效提升导入性能以及降低写入拒绝的情况。适用于索引分片较多、文本索引量大、导入吞吐量高的场景。

表 2-38 数据导入性能优化方式

优化方式	功能描述	相关文档
Bulk路由优化	<p>根据Elasticsearch默认的路由规则，Bulk请求中的每一条数据会被路由到不同的shard，当索引分片较多时，会产生大量的内部转发请求，在大规模写入场景下容易触发写拒绝。同时，在大规模集群中，长尾效应会导致Bulk请求时延较高。</p> <p>通过指定配置项“index.bulk_routing”可以开启集群的Bulk路由优化，该优化可以减少内部转发的请求数量，在shard数较多的场景下，能够有效提升写入性能以及减少写入拒绝。</p> <p>说明</p> <p>开启Bulk路由优化后（即“index.bulk_routing”设置为“pack”或“ocal_pack”），数据写入不再根据“_id”进行路由，与路由的相关功能使用会受限，例如根据“_id”进行文档GET请求可能失败。</p>	Bulk路由优化
Bulk聚合优化	<p>通过指定配置项“index.aggr_perf_batch_size”可以开启集群的Bulk聚合优化。Bulk聚合优化是通过批量导入将Bulk请求中的doc从单个依次写入变为批量写入，该方案可以有效减少内存申请、锁申请、及其他调用开销，从而提升数据导入性能。</p>	Bulk聚合优化
文本索引加速	<p>通过指定配置项“index.native_speed_up”可以开启文本索引加速。索引加速功能通过优化索引流程以及内存使用等方式实现，对于文本字段（text、keyword）能够极大提升索引构建的性能。当开启文本索引加速时，支持通过指定配置项“index.native_analyzer”同时开启分词加速。对于需要分词的文本字段（text），当无特殊分词需求时可以开启分词加速提升分词性能。</p> <p>说明</p> <ul style="list-style-type: none">仅当开启文本索引加速（即“index.native_speed_up”设置为“true”）时，才支持开启分词加速（即“index.native_analyzer”设置为“true”），否则分词加速不生效。包含“nested”字段的索引不支持开启文本索引加速。	文本索引加速

优化方式	功能描述	相关文档
索引merge任务优化	开启以上三种数据导入性能优化后，集群的索引merge任务会增加，通过指定配置项“index.merge.scheduler.max_thread_count”可以降低索引merge任务开销对导入性能的影响。索引merge任务优化可以增加shard的合并线程数，减少合并对数据导入的限流。	索引merge任务优化

约束限制

仅Elasticsearch 7.10.2集群和OpenSearch 1.3.6集群支持数据导入性能增强。

前提条件

待增强数据导入性能的集群处于“可用”状态。

Bulk 路由优化

通过指定配置项“index.bulk_routing”可以开启集群的Bulk路由优化，该优化可以减少内部转发的请求数量，在shard数较多的场景下，能够有效提升写入性能以及减少写入拒绝。

📖 说明

开启Bulk路由优化后（即“index.bulk_routing”设置为“pack”或“ocal_pack”），数据写入不再根据“_id”进行路由，与路由的相关功能使用会受限，例如根据“_id”进行文档GET请求可能失败。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。
2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令开启Bulk路由优化。

```
PUT my_index
{
  "settings": {
    "index.bulk_routing": "local_pack"
  }
}
```

配置项“index.bulk_routing”的取值范围如下所示。

- “default”：缺省值，使用集群默认的路由机制，Bulk请求中的每一条记录会拆分后独立路由。
- “pack”：单个Bulk请求的数据会被随机路由到同一个shard中。
- “ocal_pack”：单个Bulk请求的数据会被路由到接收该Bulk请求的数据节点的本地shard中，如果该节点不包含对应index的shard，则会进行随机路由到其他包含该索引shard节点上。该方案依赖客户端Bulk请求的随机打散和主shard的均衡分布。

Bulk 聚合优化

通过指定配置项 “index.aggr_perf_batch_size” 可以开启集群的Bulk聚合优化。Bulk聚合优化是通过批量导入将Bulk请求中的doc从单个依次写入变为批量写入，该方案可以有效减少内存申请、锁申请、及其他调用开销，从而提升数据导入性能。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。
2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令开启Bulk聚合优化。

```
PUT my_index
{
  "settings": {
    "index.aggr_perf_batch_size": "128"
  }
}
```

配置项 “index.aggr_perf_batch_size” 的取值范围为[1, Integer.MAX_VALUE]。缺省值为1，表示关闭Bulk聚合优化。当取值大于1时，表示打开Bulk聚合优化且批量取值为MIN(bulk_doc_size, aggr_perf_batch_size)。

文本索引加速

通过指定配置项 “index.native_speed_up” 可以开启文本索引加速。索引加速功能通过优化索引流程以及内存使用等方式实现，对于文本字段（text、keyword）能够极大提升索引构建的性能。当开启文本索引加速时，支持通过指定配置项 “index.native_analyzer” 同时开启分词加速。对于需要分词的文本字段（text），当无特殊分词需求时可以开启分词加速提升分词性能。

📖 说明

- 仅当开启文本索引加速（即 “index.native_speed_up” 设置为 “true” ）时，才支持开启分词加速（即 “index.native_analyzer” 设置为 “true” ），否则分词加速不生效。
- 包含 “nested” 字段的索引不支持开启文本索引加速。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。
2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令开启文本索引加速。

```
PUT my_index
{
  "settings": {
    "index.native_speed_up": true,
    "index.native_analyzer": true
  }
}
```

配置项 “index.native_speed_up” 和 “index.native_analyzer” 的取值范围是true和false，缺省值是false。

索引 merge 任务优化

开启以上三种数据导入性能优化后，集群的索引merge任务会增加，通过指定配置项 “index.merge.scheduler.max_thread_count” 可以降低索引merge任务开销对导入性能的影响。索引merge任务优化可以增加shard的合并线程数，减少合并对数据导入的限流。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。

2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令启动索引merge任务优化。

```
PUT my_index
{
  "settings": {
    "index.merge.scheduler.max_thread_count": 8
  }
}
```

配置项“index.merge.scheduler.max_thread_count”的取值范围是[1, node.processors/2]，缺省值是4，建议设置为8。

2.6 使用 Elasticsearch 集群搜索数据

2.6.1 使用 DSL 语言在 Elasticsearch 中搜索数据

DSL语言是Elasticsearch和OpenSearch查询域的特定语言，是客户端与Elasticsearch和OpenSearch集群交互的最佳语言。Elasticsearch DSL是基于JSON格式的语言，其他语言如SQL本质上也是先转译为Elasticsearch DSL再与Elasticsearch和OpenSearch集群交互。

DSL 使用示例

在Kibana的DevTools中编写请求内容的JSON并执行搜索请求。

例如，执行如下命令，在“test”索引中匹配所有文档。

```
GET /test/_search
{
  "query": {
    "match_all": {}
  }
}
```

查询结果返回的也是JSON格式的数据。

常用的 DSL 查询语句

下面列举了常用的DSL查询语句，全量的DSL查询语句请参见《[Elasticsearch指南](#)》。

- **设置查询条件过滤**，等同于SQL语言中的where。

如下命令中，查询未在“_search”前面过滤索引，所以是查询所有索引。bool为条件句，filter强制过滤“status”字段为“published”并且“publish_date”字段为“2015-01-01”以后的文档，must规定“title”字段含有“Search”并且“content”字段含有“Search”。

📖 说明

must和filter字段的区别在于filter等同于SQL的where会过滤字段但不参与ES搜索的打分机制，must也属于过滤必须满足的条件，但是会根据查询的匹配程度对搜索到的文档进行打分，在返回的结果中越匹配的文档会越靠前显示。

```
GET /_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
```

```
    "title": "Search"
  }
},
{
  "match": {
    "content": "search"
  }
}
],
"filter": [
  {
    "term": {
      "status": "published"
    }
  },
  {
    "range": {
      "publish_date": {
        "gte": "2015-01-01"
      }
    }
  }
]
}
}
```

- **聚合查询**，近似于SQL语言中的Group by。

该查询是根据genre这个字段进行聚合，会根据test索引里的title字段进行分类统计。如果title为text（含keyword）类型，需要使用“title.keyword”进行聚合，默认情况下Elasticsearch和OpenSearch无法直接对text类型的字段进行聚合。其中titles仅为聚合的命名，可以根据需要命名为titles或者titleaggs等。

```
GET /test/_search
{
  "aggs": {
    "titles": {
      "terms": {
        "field": "title.keyword"
      }
    }
  }
}
```

以上聚合查询的例子包含了test索引的所有文档，即match_all查询条件为隐式条件。可以配合[设置查询条件过滤](#)中的查询条件过滤特定的文档进行聚合。

2.6.2 使用 SQL 语言在 Elasticsearch 中搜索数据

在Elasticsearch集群 6.5.4及之后版本中提供Open Distro for Elasticsearch SQL插件允许您使用SQL而不是Elasticsearch查询域特定语言（DSL）编写查询。

熟悉SQL语言的用户，可以使用SQL语言在Elasticsearch中搜索数据。

SQL 使用示例

- 在Kibana中使用SQL语言搜索数据（推荐）

在Kibana的DevTools中将请求发送到“_opendistro/_sqlURI”，可以使用请求参数或请求正文。

例如，执行如下命令，从“my-index”索引中搜索出50条数据。

```
POST _opendistro/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

默认情况下，查询结果返回的是JSON格式的数据。当需要返回CSV格式的数据时，则需要在命令中对format参数进行如下设置：

```
POST _opendistro/_sql?format=csv
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

查询结果返回CSV格式的数据时，每行对应一个文档，每列对应一个字段。

- 在ECS中使用Curl命令执行SQL搜索数据

例如，执行如下命令，从“kibana_sample_data_flights”索引中搜索出10条数据。

```
curl -XPOST https://localhost:9200/_opendistro/_sql -u username:password -k -d '{"query": "SELECT * FROM kibana_sample_data_flights LIMIT 10"}' -H 'Content-Type: application/json'
```

localhost表示集群的访问地址，username和password分别表示安全模式集群的用户名和密码。

支持的 SQL 操作

支持的SQL操作包括声明、条件、聚合函数、Include和Exclude、常用函数、连接join和展示等操作。

- 声明statements

表 2-39 声明 statements

Statement	Example
Select	SELECT * FROM my-index
Delete	DELETE FROM my-index WHERE _id=1
Where	SELECT * FROM my-index WHERE ['field']='value'
Order by	SELECT * FROM my-index ORDER BY _id asc
Group by	SELECT * FROM my-index GROUP BY range(age, 20,30,39)
Limit	SELECT * FROM my-index LIMIT 50 (default is 200)
Union	SELECT * FROM my-index1 UNION SELECT * FROM my-index2
Minus	SELECT * FROM my-index1 MINUS SELECT * FROM my-index2

说明

与任何复杂查询一样，大型UNION和MINUS语句可能会使集群资源紧张甚至崩溃。

- 条件Conditions

表 2-40 条件 Conditions

Condition	Example
Like	SELECT * FROM my-index WHERE name LIKE 'j%'
And	SELECT * FROM my-index WHERE name LIKE 'j%' AND age > 21
Or	SELECT * FROM my-index WHERE name LIKE 'j%' OR age > 21
Count distinct	SELECT count(distinct age) FROM my-index
In	SELECT * FROM my-index WHERE name IN ('alejandro', 'carolina')
Not	SELECT * FROM my-index WHERE name NOT IN ('jane')
Between	SELECT * FROM my-index WHERE age BETWEEN 20 AND 30
Aliases	SELECT avg(age) AS Average_Age FROM my-index
Date	SELECT * FROM my-index WHERE birthday='1990-11-15'
Null	SELECT * FROM my-index WHERE name IS NULL

- 聚合函数 Aggregation

表 2-41 聚合函数 Aggregation

Aggregation	Example
avg()	SELECT avg(age) FROM my-index
count()	SELECT count(age) FROM my-index
max()	SELECT max(age) AS Highest_Age FROM my-index
min()	SELECT min(age) AS Lowest_Age FROM my-index
sum()	SELECT sum(age) AS Age_Sum FROM my-index

- Include和Exclude字段

表 2-42 Include 和 Exclude

Pattern	Example
include()	SELECT include('a*'), exclude('age') FROM my-index

Pattern	Example
exclude()	SELECT exclude('*name') FROM my-index

- 函数 Functions

表 2-43 函数 Functions

Function	Example
floor	SELECT floor(number) AS Rounded_Down FROM my-index
trim	SELECT trim(name) FROM my-index
log	SELECT log(number) FROM my-index
log10	SELECT log10(number) FROM my-index
substring	SELECT substring(name, 2,5) FROM my-index
round	SELECT round(number) FROM my-index
sqrt	SELECT sqrt(number) FROM my-index
concat_ws	SELECT concat_ws(' ', age, height) AS combined FROM my-index
/	SELECT number / 100 FROM my-index
%	SELECT number % 100 FROM my-index
date_format	SELECT date_format(date, 'Y') FROM my-index

📖 说明

必须在文档映射中启用 field_data 才能使大多数字符串函数正常工作。

- 连接操作 Joins

表 2-44 连接操作 Joins

Join	Example
Inner join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s JOIN school sc ON sc.name = s.school_name WHERE s.age > 20
Left outer join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s LEFT JOIN school sc ON sc.name = s.school_name

Join	Example
Cross join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s CROSS JOIN school sc

相关约束和限制，参考[连接操作Joins](#)。

- 展示Show
展示show操作与索引模式匹配的索引和映射。您可以使用*或%使用通配符。

表 2-45 展示 show

Show	Example
Show tables like	SHOW TABLES LIKE logs-*

连接操作 Joins

Open Distro for Elasticsearch SQL支持inner joins, left outer joins,和cross joins。
Join操作有许多约束：

- 您只能加入两个参数。
- 您必须为索引使用别名（例如people p）。
- 在ON子句中，您只能使用AND条件。
- 在WHERE语句中，不要将包含多个索引的树组合在一起。例如，以下语句有效：
WHERE (a.type1 > 3 OR a.type1 < 0) AND (b.type2 > 4 OR b.type2 < -1)
- 以下声明无效：
WHERE (a.type1 > 3 OR b.type2 < 0) AND (a.type1 > 4 OR b.type2 < -1)
- 您不能使用GROUP BY或ORDER BY来获得结果。
- LIMIT和OFFSET不支持一起使用（例如LIMIT 25 OFFSET 25）。

JDBC 驱动

Java数据库连接（JDBC）驱动程序允许您将Open Distro for Elasticsearch与您的商业智能（BI）应用程序集成。

有关下载和使用JAR文件的信息，请参阅[GitHub仓库](#)。

2.7 增强 Elasticsearch 集群搜索能力

2.7.1 Elasticsearch 集群搜索增强特性介绍

云搜索服务的Elasticsearch集群在开源Elasticsearch基础上增加了许多增强特性，增强特性的功能和支持的集群版本请参见[表2-46](#)。

表 2-46 Elasticsearch 集群搜索增强特性列表

增强特性	特性描述	支持的集群版本	相关文档
向量检索	向量检索支持对图像、视频、语料等非结构化数据提取的特征向量数据进行最近邻或近似近邻检索。	Elasticsearch 7.6.2、 Elasticsearch 7.10.2	配置 Elasticsearch 集群向量检索
存算分离	存算分离支持将集群新写入的数据存储在SSD来达到最佳的查询检索性能，将历史数据存储到OBS中降低数据的存储成本。 存算分离和切换冷热数据比，更适用于对搜索性能要求不高的场景，冷数据存储到OBS中，存储成本更低一些。	Elasticsearch 7.6.2、 Elasticsearch 7.10.2、 OpenSearch 1.3.6	配置 Elasticsearch 集群存算分离
流量控制 2.0	流量控制2.0提供节点级别的流量控制功能，可提供单个节点基于黑白名单的访问限制、HTTPS并发连接数限制、HTTP最大连接数限制，基于节点内存的客户端写入流量反压控制，一键断流能力。同时也提供节点访问IP统计和URI的统计能力。开启客户端写入流量反压控制功能，会在节点堆内存过大时阻止大请求涌入节点占用内存，避免节点崩溃，减少节点不可用的风险。 说明 2023年02月及之后创建的Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制2.0版本，之前创建Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制1.0版本。	Elasticsearch 7.6.2、 Elasticsearch 7.10.2	配置 Elasticsearch 集群读写流量控制策略 2.0
流量控制 1.0	流量控制1.0提供节点级别的流量控制功能，可提供单个节点基于黑白名单的访问限制、HTTP并发连接数限制、HTTP最大连接数限制、基于请求Path的堆内存最大使用量流控能力、基于CPU最大占用率流控能力，一键断流能力，同时也提供节点访问IP统计和URI的采样统计能力。开启流控功能会使请求在入口处直接阻塞，可以缓解节点高并发场景下的集群压力，降低P99时延，减少节点不可用的风险。 说明 2023年02月及之后创建的Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制2.0版本，之前创建Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制1.0版本。	Elasticsearch 7.6.2、 Elasticsearch 7.10.2	配置 Elasticsearch 集群读写流量控制策略 1.0

增强特性	特性描述	支持的集群版本	相关文档
大查询隔离	大查询隔离支持对查询请求进行独立管理，将高内存、长耗时的查询请求进行隔离，保证节点内存安全。在节点堆内存使用率过高时，触发中断控制程序，根据选择的中断策略将其中一条大查询请求进行中断，取消其正在运行的查询任务。大查询隔离同时支持全局查询超时配置，用户可实时配置所有查询请求的超时时间，中断超时查询请求。	Elasticsearch 7.6.2、 Elasticsearch 7.10.2	配置 Elasticsearch 集群大查询隔离
聚合增强	聚合增强在数据聚簇的情况下，利用向量化技术，批量处理数据，从而提升聚合性能，优化可观测性业务的聚合分析能力。	Elasticsearch 7.10.2	配置 Elasticsearch 集群聚合增强
读写分离	读写分离支持将写入主集群（Leader）的数据自动同步到从集群（Follower），实现主集群承担写入任务，从集群承担查询任务。读写分离可以将读写压力分开，提高查询的性能，还能支持在主集群无法提供服务时降级，使用从集群提供写入和查询服务。	Elasticsearch 7.6.2、 Elasticsearch 7.10.2	配置 Elasticsearch 集群读写分离
切换冷热数据	通过切换冷热数据，可以将部分现查要求秒级返回的数据放在高性能机器上面作为热数据，对于历史数据要求分钟级别返回的数据放在大容量低规格节点作为冷数据。冷热数据切换可以减低存储成本，提升搜索效率。 切换冷热数据和存算分离比，更适用于对搜索性能要求高的场景，冷数据存储于集群本地的冷数据节点中，存储的数据量大小依赖冷数据节点数和磁盘容量，存储成本也会比OBS高一些。	只要集群有冷数据节点就支持。	切换 Elasticsearch 集群冷热数据
索引回收站	索引回收站支持将删除的索引存放到回收站中，且支持从回收站中还原索引，防止误操作导致数据被删除，进而提升集群的数据可靠性。	Elasticsearch 7.10.2	配置 Elasticsearch 集群索引回收站

2.7.2 配置 Elasticsearch 集群向量检索

2.7.2.1 向量检索特性介绍

向量检索支持对图像、视频、语料等非结构化数据提取的特征向量数据进行最近邻或近似近邻检索。

原理说明

向量检索从本质上讲，其思维框架和传统的检索方法没有区别。为了提升向量检索的性能，通常需要解决以下两个问题：

- **减少候选向量集**

和传统的文本检索类似，向量检索也需要某种索引结构来避免在全量的数据上做匹配，传统文本检索是通过倒排索引来过滤掉无关文档，而向量检索是通过对向量建立索引结构来绕过不相关的向量，减小需要考察的范围。

- **降低单个向量计算的复杂度**

向量检索支持漏斗模型，先对所有向量进行量化和近似计算，筛选出一定量接近检索目标的数据集，然后基于筛选的数据集进行精细的计算和排序。本方法不需要对所有向量都进行复杂的计算，可以有效提高检索效率。

向量检索即在一个给定的向量数据集中，按照某种度量方式，检索出与查询向量相近的K个向量（K-Nearest Neighbor, KNN），但由于KNN计算量过大，通常只关注近似近邻（Approximate Nearest Neighbor, ANN）问题。

功能介绍

云搜索服务的向量检索引擎集成了暴力检索、图索引（HNSW）、乘积量化、IVF-HNSW等多种向量索引，支持欧式、内积、余弦、汉明等多种相似度计算方式，召回率和检索性能均优于开源引擎。能够满足高性能、高精度、低成本、多模态等多种应用场景及需求。

向量检索支持原生Elasticsearch的所有能力，包括分布式、多副本、错误恢复、快照、权限控制等；兼容所有原生Elasticsearch生态，包括集群监测工具Cerebro，可视化工具Kibana，实时数据采集工具Logstash等；提供Python/Java/Go/C++等多种客户端语言支持。

约束限制

- 仅Elasticsearch 7.6.2、Elasticsearch 7.10.2集群支持向量检索。
- 向量检索插件涉及较高的内存计算，内存要求比普通索引高，建议集群选择“内存优化型”的计算规格。

向量检索的集群规格规划

向量检索的索引构建与查询均使用堆外内存，所以集群容量与索引类型、总堆外内存大小等因素相关。通过预估全量索引所需的堆外内存大小，可以选择合适的集群规格。由于向量索引内存占用较高，CSS对于内存规格为8GB及以下的集群默认禁用了向量检索插件。

不同类型的索引所需堆外内存大小的预估方式不同，计算公式如下：

- **GRAPH类索引**

$$\text{mem_needs} = (\text{dim} \times \text{dim_size} + \text{neighbros} \times 4) \times \text{num} + \text{delta}$$

 **说明**

如果有实时更新索引的需求，还需要考虑向量索引构建和自动merge所需的堆外内存开销，保守估计需要1.5~2倍 mem_needs 。

- **PQ类索引**

$$\text{mem_needs} = \text{frag_num} \times \text{frag_size} \times \text{num} + \text{delta}$$

- **FALT、IVF索引**

$$\text{mem_needs} = \text{dim} \times \text{dim_size} \times \text{num} + \text{delta}$$

表 2-47 参数说明

参数	说明
dim	向量维度。
neighbors	图节点邻居数，默认值为64。
dim_size	每一维度值所需的字节数，默认为float类型，需要4字节。
num	向量总条数。
delta	元数据大小，该项通常可以忽略。
frag_num	量化编码时的向量分段数，创建索引时如果未配置该值，则由向量维度“dim”决定。 <pre> if dim <= 256: frag_num = dim / 4 elif dim <= 512: frag_num = dim / 8 else : frag_num = 64 </pre>
frag_size	量化编码时中心点编码的size，默认为1，当“frag_num”大于256时，该值等于2。

基于上述计算方法，可预估出完整向量索引所需堆外内存的大小。选择集群规格时，还需考虑每个节点的堆内存开销。

节点的堆内存分配策略：每个节点的堆内存大小为节点物理内存的一半，且最大不超过31GB。

例如，基于SIFT10M数据集创建GRAPH索引，其“dim”为“128”，“dim_size”为“4”，“neighbors”采用默认值“64”，“num”为“1000万”，将各值代入上述公式得到GRAPH索引所需堆外内存大小约为7.5GB，计算公式为“ $\text{mem_needs} = (128 \times 4 + 64 \times 4) \times 10000000 \approx 7.5$ ”。

同时考虑到堆内存的开销，单台“8U 16G”规格的机器可以满足该场景的需求。如果实际场景还有实时写入或更新的需求，则需要考虑申请更大的内存规格。

2.7.2.2 创建向量索引

创建向量索引的操作流程如下所示：

1. **（可选）准备工作**：根据业务需要，完成集群高级配置。
2. **（可选）预构建与注册中心点向量**：当选择使用“IVF_GRAPH”或“IVF_GRAPH_PQ”索引算法创建向量索引时，则需要对中心点向量进行预构建和注册。
3. **创建向量索引**：基于业务场景创建向量索引。
4. **导入向量数据**：将向量数据导入集群。
5. **使用向量索引搜索数据**：进行向量查询。

前提条件

已经参考[向量检索的集群规格规划](#)完成集群创建，且集群版本是Elasticsearch 7.6.2、Elasticsearch 7.10.2或OpenSearch 1.3.6。

（可选）准备工作

在创建向量索引前，请根据业务场景，完成集群高级配置。

- 在离线导入数据场景下，为了提高批量写入性能，建议将索引的“refresh_interval”参数设置为“-1”，即关闭自动刷新索引。
- 建议将备份数“number_of_replicas”设置为“0”，当离线数据导入完成后，再设置为需要的值。
- 其他高级功能的参数配置说明请参见[表2-48](#)。

表 2-48 集群高级配置参数说明

参数	说明
native.cache.circuit_breaker.enabled	是否开启堆外内存熔断。 默认值：true。
native.cache.circuit_breaker.cpu.limit	向量索引堆外内存使用上限。 假设使用128GB内存的机器且堆内存大小为31GB，默认堆外内存使用上限为 $(128 - 31) * 45\% = 43.65\text{GB}$ ，堆外内存使用量超过该值将会触发写入熔断。 默认值：45%。
native.cache.expire.enabled	是否开启缓存超时设置。开启时，如果某些缓存项长时间没有被访问过将会被清除。 取值范围：true、false。 默认值：false。
native.cache.expire.time	超时时长。 默认值：24h。
native.vector.index_threads	创建底层索引时所使用的线程数，每个shard均会使用多个构建线程。该值建议不要设置过大，避免产生过多的构建线程抢占查询资源。 默认值：4。

（可选）预构建与注册中心点向量

当创建向量索引时选择使用“IVF_GRAPH”和“IVF_GRAPH_PQ”的索引算法，则需要对中心点向量进行预构建和注册。

在向量索引加速算法中，“IVF_GRAPH”和“IVF_GRAPH_PQ”适用于超大规模场景。这两种算法需要通过对于子空间的切割缩小查询范围，子空间的划分通常采用聚类或者随机采样的方式。在预构建之前，需要通过聚类或者随机采样得到所有的中心点向量。通过预构建和注册将中心点向量预构建成为GRAPH或者GRAPH_PQ索引，同时注

册到CSS集群内，实现在多个节点间共享此索引文件。中心点索引在shard间复用能够有效减少训练的开销、中心点索引的查询次数，提升写入以及查询的性能。

1. 选择启用向量检索的集群，单击操作列“Kibana”，登录Kibana界面。
2. 单击左侧导航栏的“Dev Tools”，进入操作界面。
3. 创建中心点索引表。
 - 创建的索引命名为my_dict，注意该索引的number_of_shards数必须设置为1，否则无法注册。
 - 当需要使用IVF_GRAPH索引时，中心点索引的algorithm设置为GRAPH。
 - 当需要使用IVF_GRAPH_PQ索引时，中心点索引的algorithm设置为GRAPH_PQ。

```
PUT my_dict
{
  "settings": {
    "index": {
      "vector": true
    },
    "number_of_shards": 1,
    "number_of_replicas": 0
  },
  "mappings": {
    "properties": {
      "my_vector": {
        "type": "vector",
        "dimension": 2,
        "indexing": true,
        "algorithm": "GRAPH",
        "metric": "euclidean"
      }
    }
  }
}
```

4. 写入中心点向量数据。
参考[导入向量数据](#)将采样或者聚类得到的中心点向量写入上述创建的my_dict索引中。
5. 调用注册接口。
将上述创建的my_dict索引注册具有全局唯一标识名称（dict_name）的Dict对象。

```
PUT _vector/register/my_dict
{
  "dict_name": "my_dict"
}
```

6. 创建IVF_GRAPH或IVF_GRAPH_PQ索引。
在创建IVF_GRAPH或者IVF_GRAPH_PQ索引时，不再需要指定dimension以及metric信息，只需指定之前注册好的dict名称即可。

```
PUT my_index
{
  "settings": {
    "index": {
      "vector": true
    }
  },
  "mappings": {
    "properties": {
      "my_vector": {
        "type": "vector",
        "indexing": true,
        "algorithm": "IVF_GRAPH",

```

```
"dict_name": "my_dict",  
  "offload_ivf": false  
}  
}  
}
```

表 2-49 Field mappings 参数

参数	说明
dict_name	指定依赖的中心点索引名称。该索引字段的向量维度和度量方式将与dict索引保持一致，不再需要额外指定。
offload_ivf	将底层索引实现的IVF倒排索引卸载到ES端实现，可以减少堆外内存的使用，以及减少写入/合并的性能开销，但是查询的性能也有一定的损失。采用默认值即可。 取值范围：true、false。 默认值：false。

创建向量索引

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，选择需要启用向量检索的集群，单击操作列“Kibana”，登录Kibana界面。
3. 单击左侧导航栏的“Dev Tools”，执行如下命令创建向量索引。

创建一个名为“my_index”的索引，该索引包含一个名为“my_vector”的向量字段和一个名为“my_label”的文本字段。其中，向量字段创建了GRAPH图索引，并使用欧式距离作为相似度度量。

```
PUT my_index  
{  
  "settings": {  
    "index": {  
      "vector": true  
    }  
  },  
  "mappings": {  
    "properties": {  
      "my_vector": {  
        "type": "vector",  
        "dimension": 2,  
        "indexing": true,  
        "algorithm": "GRAPH",  
        "metric": "euclidean"  
      },  
      "my_label": {  
        "type": "text"  
      }  
    }  
  }  
}
```

表 2-50 创建索引参数说明

类型	参数	说明
Index settings 参数	vector	当需要使用向量索引加速时，需要设置该值为 true。
Field mappings 参数	type	字段类型，“vector”表示该字段为向量字段。
	dimension	向量数据维度。取值范围：[1, 4096]。
	indexing	是否开启向量索引加速。 可选值： <ul style="list-style-type: none">• false：表示关闭向量索引加速，向量数据仅写入 docvalues，只支持使用 ScriptScore 以及 Rescore 进行向量查询。• true：表示开启向量索引加速，系统将创建额外的向量索引，索引算法由 "algorithm" 字段指定，写入数据后可以使用 VectorQuery 进行查询。 默认值：false。

类型	参数	说明
	algorithm	<p>索引算法。仅当“indexing”为“true”时生效。</p> <p>可选值：</p> <ul style="list-style-type: none"> ● FLAT：暴力计算，目标向量依次和所有向量进行距离计算，此方法计算量大，召回率100%。适用于对召回准确率要求极高的场景。 ● GRAPH：图索引，内嵌深度优化的HNSW算法，主要应用在对性能和精度均有较高要求且单shard中文档数量在千万个以内的场景。 ● GRAPH_PQ：将HNSW算法与PQ算法进行了结合，通过PQ降低原始向量的存储开销，能够使HNSW轻松支撑上亿规模的检索场景。 ● IVF_GRAPH：算法将IVF与HNSW结合，对全量空间进行划分，每一个聚类中心向量代表了一个子空间，极大地提升检索效率，同时会带来微小的检索精度损失。适用于数据量在上亿以上同时对检索性能要求较高的场景。 ● IVF_GRAPH_PQ：PQ算法与IVF-HNSW的结合，PQ可以通过配置选择与HNSW结合和IVF结合，进一步提升系统的容量并降低系统开销，适用于shard中文档数量在十亿级以上同时对检索性能要求较高的场景。 ● PV_GRAPH：图索引，同属内嵌优化的HNSW算法，主要应用在性能和精度均有较高要求，且单shard中文档数量在千万个以内，可用于内存资源充足的场景。同时该算法支持联合标量过滤，目前仅限关联keyword类型字段，相较于后置过滤和布尔查询，可大幅提升返回结果的填充率和查询性能。仅Elasticsearch集群7.10.2版本支持PV_GRAPH索引。使用PV_GRAPH算法实现向量检索可以参考案例示例：使用PV_GRAPH算法实现向量检索。 <p>默认值：GRAPH。</p> <p>说明 当选择IVF_GRAPH或者IVF_GRAPH_PQ索引时，需要额外进行预构建中心点索引以及注册等步骤，具体内容请参考（可选）预构建与注册中心点向量。</p>
	其他可选参数	<p>当使用向量索引加速时（即“indexing”为“true”时），为了获得更高的查询性能以及查询精度，CSS提供了与向量索引相关的可选参数配置，参数说明请参见表2-51。</p>

类型	参数	说明
	metric	计算向量之间距离的度量方式。 可选值： <ul style="list-style-type: none"> euclidean：欧式距离。 inner_product：内积距离。 cosine：余弦距离。 hamming：汉明距离，仅支持设置 "dim_type" 为 "binary" 时使用。 默认值：euclidean。
	dim_type	向量维度值的类型。 可选值：binary、float（默认）。
	sub_fields	新增参数字段 sub_fields 用于定义向量附属标量字段，仅支持 keyword 类型，对应新增支持联合过滤功能，仅支持 PV_GRPAAH 类型。

表 2-51 可选参数说明

类型	参数	说明
GRAPH类索引配置参数	neighbors	图索引中每个向量的邻居数，默认值为64，值越大查询精度越高。索引越大，构建速度以及后续的查询速度也会变慢。 取值范围：[10, 255]。
	shrink	构建hnswh时的裁边系数，默认值1.0f。 取值范围：(0.1, 10)。
	scaling	构建hnswh时上层图节点数的缩放比例，默认值50。 取值范围：(0, 128]。
	efc	构建hnswh时考察邻居节点的队列大小，默认值为200，值越大精度越高，构建速度将会变慢。 取值范围：(0, 100000]。
	max_scan_num	扫描节点上限，默认值为10000，值越大精度越高，索引速度变慢。 取值范围：(0, 1000000]。
PQ类索引配置参数	centroid_num	每一段的聚类中心点数目，默认值为255。 取值范围：(0, 65535]。
	fragment_num	段数，默认值为0，插件自动根据向量长度设置合适的段数。 取值范围：[0, 4096]。

导入向量数据

执行如下命令，导入向量数据。向“my_index”索引中写入向量数据时，需要指定向量字段名称和向量数据。

- 向量数据输入格式为逗号分隔的浮点型数组时：

```
POST my_index/_doc
{
  "my_vector": [1.0, 2.0]
}
```

- 向量数据输入格式为小端字节序编码的Base64字符串时：

在写入二值向量，或向量维度较高、数值有效位较多时，使用Base64编码格式传输、解析更加高效。

```
POST my_index/_doc
{
  "my_vector": "AACAPwAAAAEA="
}
```

- 当写入大规模数据时，建议使用Bulk操作：

```
POST my_index/_bulk
{"index": {}}
{"my_vector": [1.0, 2.0], "my_label": "red"}
{"index": {}}
{"my_vector": [2.0, 2.0], "my_label": "green"}
{"index": {}}
{"my_vector": [2.0, 3.0], "my_label": "red"}
```

2.7.2.3 使用向量索引搜索数据

使用向量索引搜索数据支持多种方式。

- [标准查询](#)
- [复合查询](#)
- [ScriptScore查询](#)
- [重打分查询](#)
- [Painless语法扩展查询](#)

标准查询

针对创建了向量索引的向量字段，提供了标准向量查询语法。下述查询命令将会返回所有数据中与查询向量最近的size（topk）条数据。

```
POST my_index/_search
{
  "size":2,
  "_source": false,
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1, 1],
        "topk":2
      }
    }
  }
}
```

表 2-52 标准查询的参数说明

参数	说明
vector (第一个)	表示该查询类型为VectorQuery。
my_vector	指定了需要查询的向量字段名称。
vector (第二个)	指定查询向量的具体值，支持数组形式以及Base64编码形式的输入。
topk	topk的值通常与size保持一致。
其他可选的查询参数	通过调整不同索引的查询参数，可以获得更高的查询性能或者查询精度，其他参数请参见表2-53。

表 2-53 可选的查询参数说明

参数	子参数	说明
GRAPH类索引配置参数	ef	查询时考察邻居节点的队列大小。值越大查询精度越高，查询速度会变慢。默认值为200。 取值范围：(0, 100000]。
	max_scan_num	扫描节点上限。值越大精度越高，查询速度变慢。默认值为10000。 取值范围：(0, 1000000]。
IVF类索引配置参数	nprobe	查询考察中心点的数目。值越大精度越高，查询速度变慢。默认值为100。 取值范围：(0, 100000]。

复合查询

向量检索支持与其他ES子查询组合进行复合查询，比如布尔查询、后置过滤等。

以下两个示例的查询结果：首先查询top10条与查询向量距离最近的结果，filter作为过滤条件将仅保留my_label字段为“red”的结果。

- 布尔查询示例

```
POST my_index/_search
{
  "size": 10,
  "query": {
    "bool": {
      "must": {
        "vector": {
          "my_vector": {
            "vector": [1, 2],
            "topk": 10
          }
        }
      }
    },
    "filter": {
      "term": { "my_label": "red" }
    }
  }
}
```

```
}  
}  
}  
• 后置过滤示例  
GET my_index/_search  
{  
  "size": 10,  
  "query": {  
    "vector": {  
      "my_vector": {  
        "vector": [1, 2],  
        "topk": 10  
      }  
    }  
  },  
  "post_filter": {  
    "term": { "my_label": "red" }  
  }  
}
```

ScriptScore 查询

写入向量数据后，针对向量字段可以使用ScriptScore进行最近邻查询，查询语法如下所示。

前置过滤条件可以为任意查询，script_score仅针对前置过滤的结果进行遍历，计算向量相似度并排序返回。此种查询方式的性能取决于前置过滤后中间结果集的大小，当前置过滤条件为"match_all"时，相当于全局暴力检索。

```
POST my_index/_search  
{  
  "size":2,  
  "query": {  
    "script_score": {  
      "query": {  
        "match_all": {}  
      },  
      "script": {  
        "source": "vector_score",  
        "lang": "vector",  
        "params": {  
          "field": "my_vector",  
          "vector": [1.0, 2.0],  
          "metric": "euclidean"  
        }  
      }  
    }  
  }  
}
```

表 2-54 script_score 参数说明

参数	说明
source	script脚本描述，使用向量相似度打分时为固定值"vector_score"。
lang	script语法描述，使用固定值"vector"。
field	向量字段名称。
vector	查询向量数据。

参数	说明
metric	度量方式，可选值为：euclidean、inner_product、cosine、hamming。 默认值：euclidean。

重打分查询

当使用GRAPH_PQ索引或者IVF_GRAPH_PQ索引时，查询结果是根据PQ计算的非对称距离进行排序。CSS支持Rescore的方式对查询结果进行重打分精排，提升召回率。

假设my_index是PQ类型的索引，Rescore示例如下：

```
GET my_index/_search
{
  "size": 10,
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1.0, 2.0],
        "topk": 100
      }
    }
  },
  "rescore": {
    "window_size": 100,
    "vector_rescore": {
      "field": "my_vector",
      "vector": [1.0, 2.0],
      "metric": "euclidean"
    }
  }
}
```

表 2-55 Rescore 参数说明

参数	说明
window_size	向量检索将会返回topk条结果，仅取前window_size条结果精排。
field	向量字段名称。
vector	查询向量数据。
metric	度量方式，可选值为：euclidean、inner_product、cosine、hamming。 默认值：euclidean。

Painless 语法扩展查询

CSS扩展实现了多种向量距离计算函数，可在自定义的painless脚本中直接使用，用以构建灵活的重打分公式。

示例如下：

```
POST my_index/_search
{
  "size": 10,
  "query": {
    "script_score": {
      "query": {
        "match_all": {}
      },
      "script": {
        "source": "1 / (1 + euclidean(params.vector, doc[params.field]))",
        "params": {
          "field": "my_vector",
          "vector": [1, 2]
        }
      }
    }
  }
}
```

表 2-56 支持的距离计算函数

函数签名	说明
euclidean(Float[], DocValues)	欧式距离函数。
cosine(Float[], DocValues)	余弦相似度函数。
innerproduct(Float[], DocValues)	内积函数。
hamming(String, DocValues)	汉明距离函数。只支持"dim_type"为"binary"的向量字段，输入的查询向量需要为Base64编码字符串格式。

2.7.2.4 优化向量检索写入与查询性能

写入性能优化

- 关闭副本，待数据导入完成后再开启副本，减少副本构建的开销。
- 调整“refresh_interval”为120s或者更大，避免频繁刷新索引生成大量小的segments，同时减少merge带来的向量索引构建开销。
- 适当调大“native.vector.index_threads”的值（默认为4），增加向量索引构建的线程数。

```
PUT _cluster/settings
{
  "persistent": {
    "native.vector.index_threads": 8
  }
}
```

查询性能优化

- 在批量导入场景下，数据写入完成后，执行forcemerge操作能有效提升查询效率。
POST index_name/_forcemerge?max_num_segments=1
- 如果向量索引所需堆外内存超过了熔断线，查询时索引的缓存管理器会控制索引的换进换出，导致查询变慢，此时可适当调大熔断线的配置。

```
PUT _cluster/settings
{
  "persistent": {
    "native.cache.circuit_breaker.cpu.limit": "75%"
  }
}
```

- 如果查询的fetch阶段开销较大，可通过配置“_source”减小fdt文件的大小，从而降低fetch开销。

```
PUT my_index
{
  "settings": {
    "index": {
      "vector": "true"
    },
    "index.soft_deletes.enabled": false
  },
  "mappings": {
    "_source": {
      "excludes": ["my_vector"]
    },
    "properties": {
      "my_vector": {
        "type": "vector",
        "dimension": 128,
        "indexing": true,
        "algorithm": "GRAPH",
        "metric": "euclidean"
      }
    }
  }
}
```

2.7.2.5 管理向量索引缓存

CSS的向量检索引擎使用C++实现，使用的是堆外内存，该插件提供了接口对向量索引的缓存进行管理。

- **查看缓存统计信息**

```
GET /_vector/stats
```

在向量插件实现中，向量索引与Lucene其他类型索引一样，每一个segment构造并存储一份索引文件，在查询时，该索引文件会被加载到堆外内存中。插件使用缓存机制对这些堆外内存进行管理。上述API能够查询当前堆外内存使用量、缓存命中次数、加载次数等信息。

- **预加载向量索引**

```
PUT /_vector/warmup/{index_name}
```

使用上述接口能将指定index_name的向量索引预加载至堆外内存供查询使用。

- **清除缓存**

```
PUT /_vector/clear/cache
```

```
PUT /_vector/clear/cache/index_name
```

在使用向量索引时，缓存机制会限制堆外内存使用量。当总索引大小超出缓存大小限制时，将会发生索引项的换进换出，此时将会影响查询的性能。通过清除缓存API能够将不再使用的索引缓存清空，保证热数据索引的查询性能。

2.7.2.6 向量检索的客户端代码示例（Python）

Elasticsearch提供了标准的REST接口，以及Java、Python等语言编写的客户端。

本节提供一份创建向量索引、导入向量数据和查询向量数据的Python代码示例，介绍如何使用客户端实现向量检索。

前提条件

客户端已经安装python依赖包。如果未安装可以执行如下命令安装：

```
# 根据集群实际版本填写，此处以7.6举例  
pip install elasticsearch==7.6
```

代码示例

```
from elasticsearch import Elasticsearch  
from elasticsearch import helpers  
  
# 创建Elasticsearch客户端  
def get_client(hosts: list, user: str = None, password: str = None):  
    if user and password:  
        return Elasticsearch(hosts, http_auth=(user, password), verify_certs=False, ssl_show_warn=False)  
    else:  
        return Elasticsearch(hosts)  
  
# 创建索引表  
def create(client: Elasticsearch, index: str):  
    # 索引mapping信息  
    index_mapping = {  
        "settings": {  
            "index": {  
                "vector": "true", # 开启向量特性  
                "number_of_shards": 1, # 索引分片数，根据实际需求设置  
                "number_of_replicas": 0, # 索引副本数，根据实际需求设置  
            }  
        },  
        "mappings": {  
            "properties": {  
                "my_vector": {  
                    "type": "vector",  
                    "dimension": 2,  
                    "indexing": True,  
                    "algorithm": "GRAPH",  
                    "metric": "euclidean"  
                }  
            }  
            # 可根据需求添加其他字段  
        }  
    }  
    res = client.indices.create(index=index, body=index_mapping)  
    print("create index result: ", res)  
  
# 写入数据  
def write(client: Elasticsearch, index: str, vecs: list, bulk_size=500):  
    for i in range(0, len(vecs), bulk_size):  
        actions = [  
            {  
                "_index": index,  
                "my_vector": vec,  
                # 可根据需求添加其他字段  
            }  
            for vec in vecs[i:i+bulk_size]  
        ]  
        success, errors = helpers.bulk(client, actions, request_timeout=3600)  
        if errors:  
            print("write bulk failed with errors: ", errors) # 根据需求进行错误处理  
        else:  
            print("write bulk {} docs success".format(success))  
        client.indices.refresh(index=index, request_timeout=3600)  
  
# 查询向量索引  
def search(client: Elasticsearch, index: str, query: list, size: int):  
    # 查询语句，可根据需求选择合适的查询方式  
    query_body = {  
        "size": size,
```

```
"query": {
  "vector": {
    "my_vector": {
      "vector": query,
      "topk": size
    }
  }
}
}
}
res = client.search(index=index, body=query_body)
print("search index result: ", res)

# 删除索引
def delete(client: Elasticsearch, index: str):
    res = client.indices.delete(index=index)
    print("delete index result: ", res)

if __name__ == '__main__':
    # 对于非安全集群, 使用:
    es_client = get_client(hosts=['http://x.x.x.x:9200'])

    # 对于开启了https的安全集群, 使用:
    # es_client = get_client(hosts=['https://x.x.x.x:9200', 'https://x.x.x.x:9200'], user='xxxxx', password='xxxxx')

    # 对于未开启https的安全集群, 使用:
    # es_client = get_client(hosts=['http://x.x.x.x:9200', 'http://x.x.x.x:9200'], user='xxxxx', password='xxxxx')

    # 测试索引名称
    index_name = "my_index"

    # 创建索引
    create(es_client, index=index_name)

    # 写入数据
    data = [[1.0, 1.0], [2.0, 2.0], [3.0, 3.0]]
    write(es_client, index=index_name, vecs=data)

    # 查询索引
    query_vector = [1.0, 1.0]
    search(es_client, index=index_name, query=query_vector, size=3)

    # 删除索引
    delete(es_client, index=index_name)
```

2.7.2.7 向量检索的客户端代码示例 (Java)

Elasticsearch提供了标准的REST接口, 以及Java、Python等语言编写的客户端。

本节提供一份创建向量索引、导入向量数据和查询向量数据的Java代码示例, 介绍如何使用客户端实现向量检索。

前提条件

根据集群实际版本添加如下Maven依赖, 此处以7.10.2举例。

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>7.10.2</version>
</dependency>
```

代码示例

```
package org.example;

import org.apache.http.HttpEntity;
```

```
import org.apache.http.HttpHost;
import org.apache.http.HttpStatus;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.elasticsearch.action.admin.indices.delete.DeleteIndexRequest;
import org.elasticsearch.action.admin.indices.refresh.RefreshRequest;
import org.elasticsearch.action.bulk.BulkRequest;
import org.elasticsearch.action.bulk.BulkResponse;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.search.SearchResponse;
import org.elasticsearch.action.support.master.AcknowledgedResponse;
import org.elasticsearch.client.Request;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.client.indices.CreateIndexRequest;
import org.elasticsearch.client.indices.CreateIndexResponse;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.DeprecationHandler;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.common.xcontent.XContentType;

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import java.io.IOException;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.X509Certificate;
import java.util.Arrays;
import java.util.List;

public class ClientExample {
    private final RestHighLevelClient client;

    public ClientExample(RestHighLevelClient client) {
        this.client = client;
    }

    // 创建非安全集群访问客户端
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme) {
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
        return new RestHighLevelClient(RestClient.builder(httpHosts));
    }

    // 创建安全集群访问客户端
    public static RestHighLevelClient getClient(List<String> hosts, int port, String scheme, String user, String
password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(user,
password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException | NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        HttpHost[] httpHosts = hosts.stream().map(p -> new HttpHost(p, port,
scheme)).toArray(HttpHost[]::new);
```

```
final SSLIOStrategy sessionStrategy = new SSLIOStrategy(sc,
NoopHostnameVerifier.INSTANCE);
RestClientBuilder builder = RestClient.builder(httpHosts).setHttpClientConfigCallback(httpClientBuilder
-> {
    httpClientBuilder.disableAuthCaching();
    httpClientBuilder.setSSLStrategy(sessionStrategy);
    return httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
});
return new RestHighLevelClient(builder);
}

public static TrustManager[] trustAllCerts = new TrustManager[] {
    new X509TrustManager() {
        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) {
        }
        @Override
        public void checkServerTrusted(X509Certificate[] chain, String authType) {
        }
        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    }
};

// 创建索引
public void create(String index) throws IOException {
    CreateIndexRequest request = new CreateIndexRequest(index);
    request.settings(Settings.builder()
        .put("index.vector", true) // 开启向量特性
        .put("index.number_of_shards", 1) // 索引分片数, 根据实际需求设置
        .put("index.number_of_replicas", 0) // 索引副本数, 根据实际需求设置
    );
    String mapping =
        "{\n" +
        "  \"properties\": {\n" +
        "    \"my_vector\": {\n" +
        "      \"type\": \"vector\",\n" +
        "      \"indexing\": \"true\",\n" +
        "      \"dimension\": \"2\",\n" +
        "      \"metric\": \"euclidean\",\n" +
        "      \"algorithm\": \"GRAPH\"\n" +
        "    }\n" +
        "  }\n" +
        "}";
    request.mapping(mapping, XContentType.JSON);
    CreateIndexResponse response = client.indices().create(request, RequestOptions.DEFAULT);
    if (response.isAcknowledged()) {
        System.out.println("create " + index + " success");
    }
}

// 写入数据, 一批数据的大小建议控制在500条以内
public void write(String index, List<float[]> vectors) throws IOException {
    BulkRequest request = new BulkRequest();
    for (float[] vec : vectors) {
        request.add(new IndexRequest(index).source("my_vector", vec));
    }

    BulkResponse response = client.bulk(request, RequestOptions.DEFAULT);
    if (response.hasFailures()) {
        System.out.println(response.buildFailureMessage());
    } else {
        System.out.println("write bulk to " + index + " success");
    }
}

// 可选, Elasticsearch会默认刷新
client.indices().refresh(new RefreshRequest(index), RequestOptions.DEFAULT);
```

```
}

// 查询向量
public void search(String index, float[] query, int size) throws IOException {
    String queryFormat =
        "{\n" +
        "  \"size\": %d,\n" +
        "  \"query\": {\n" +
        "    \"vector\": {\n" +
        "      \"my_vector\": {\n" + // 查询向量字段名称
        "        \"vector\": %s,\n" +
        "        \"topk\": %d\n" +
        "      }\n" +
        "    }\n" +
        "  }\n" +
        "}";
    String body = String.format(queryFormat, size, Arrays.toString(query), size);
    Request request = new Request("POST", index + "/" + "_search");
    request.setJsonEntity(body);
    Response response = client.getLowLevelClient().performRequest(request);
    if (response.getStatusLine().getStatusCode() != HttpStatus.SC_OK) {
        System.out.println(response.getEntity()); // 根据业务需求处理查询错误
        return;
    }
    // 根据业务需求处理正常返回结果
    HttpEntity entity = response.getEntity();
    XContentType xContentType = XContentType.fromMediaTypeOrFormat("application/json");
    XContentParser parser = xContentType.xContent().createParser(NamedXContentRegistry.EMPTY,
        DeprecationHandler.IGNORE_DEPRECATIONS, entity.getContent());
    SearchResponse searchResponse = SearchResponse.fromXContent(parser);
    System.out.println(searchResponse);
}

// 删除索引
public void delete(String index) throws IOException {
    DeleteIndexRequest request = new DeleteIndexRequest(index);
    AcknowledgedResponse response = client.indices().delete(request, RequestOptions.DEFAULT);
    if (response.isAcknowledged()) {
        System.out.println("delete " + index + " success");
    }
}

public void close() throws IOException {
    client.close();
}

public static void main(String[] args) throws IOException {
    // 对于非安全集群，使用：
    RestHighLevelClient client = getClient(Arrays.asList("x.x.x.x"), 9200, "http");

    /*
     * 对于开启了https的安全集群，使用：
     * RestHighLevelClient client = getClient(Arrays.asList("x.x.x.x", "x.x.x.x"), 9200, "https", "user_name",
    "password");
     * 对于未开启https的安全集群，使用：
     * RestHighLevelClient client = getClient(Arrays.asList("x.x.x.x", "x.x.x.x"), 9200, "http", "user_name",
    "password");
     */

    ClientExample example = new ClientExample(client);
    String indexName = "my_index";

    // 创建索引
    example.create(indexName);

    // 写入数据
    List<float[]> data = Arrays.asList(new float[]{1.0f, 1.0f}, new float[]{2.0f, 2.0f}, new float[]{3.0f, 3.0f});
    example.write(indexName, data);
}
```

```
// 查询索引
float[] queryVector = new float[]{1.0f, 1.0f};
example.search(indexName, queryVector, 3);

// 删除索引
example.delete(indexName);

// 关闭客户端
example.close();
}
```

2.7.2.8 示例：使用 PV_GRAPH 算法实现向量检索

向量字段支持的PV_GRAPH算法属于深度优化的HNSW算法，可支持向量标量联合过滤特性。使用联合标量过滤特性时，相较于后置过滤和布尔查询，可大幅提升结果的填充率和查询性能。

前提条件

已经参考[向量检索的集群规格规划](#)完成集群创建，且集群版本是Elasticsearch 7.10.2。

创建索引

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，选择需要启用向量检索的集群，单击操作列“Kibana”，登录Kibana界面。
3. 单击左侧导航栏的“Dev Tools”，执行如下命令创建向量索引。

创建一个名为“my_index”的索引，该索引包含一个名为“my_vector”的向量字段和两个子标签字段“country”和“category”。

```
PUT my_index
{
  "settings": {
    "index": {
      "vector": true
    }
  },
  "mappings": {
    "properties": {
      "my_vector": {
        "type": "vector",
        "dimension": 2,
        "indexing": true,
        "algorithm": "PV_GRAPH",
        "metric": "euclidean",
        "sub_fields": ["country", "category"]
      }
    }
  }
}
```

创建索引参数说明请见[表2-50](#)。

📖 说明

PV_GRAPH索引算法“metric”参数仅支持取值为“euclidean”和“inner_product”。

导入向量和标量数据

针对使用了PV_GRAPH索引算法和“sub_fields”的字段，支持如下数据写入语法，其中联合标量字段仅支持写入keyword类型数据，且支持多值、单值场景。

```
# 写入单条数据
POST my_index/_doc
{
  "my_vector": {
    "data": [1.0, 1.0],
    "country": "cn",
    "category": ["1", "2"]
  }
}

# 批量写入多条数据
POST my_index/_bulk
{"index": {}}
{"my_vector": {"data": [1.0, 2.0], "country": "cn", "category": "1"}}
{"index": {}}
{"my_vector": {"data": [2.0, 2.0], "country": "cn", "category": ["1", "2"]}}
{"index": {}}
{"my_vector": {"data": [2.0, 3.0], "country": "eu", "category": "2"}}
```

查询向量数据

基于现有的Elasticsearch接口，在vector下新增filter参数支持向量标量联合过滤特性。标签过滤使用的过滤字段，当前支持json格式，支持should、must、must_not、term、terms查询，语法与Elasticsearch查询语法一致。具体的限制如下。

当前过滤嵌套深度最大为4层，其中：

- must_not不支持嵌套和被嵌套。
- 第一级必须只有一个查询关键词（must等），不支持多个关键词并列。

涉及指定联合过滤的标量字段，即创建索引时sub_fields所定义的字段，仅在向量索引类型为PV_GRAPH类型生效。当指定的过滤字段不存在时，过滤请求失效，当作无过滤条件处理。

```
# 单标签单值匹配查询示例
GET my_index/_search
{
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1.0, 1.0],
        "topk": 10,
        "filter": {
          "term": { "country": "cn" }
        }
      }
    }
  }
}

# 单标签多值匹配查询示例
GET my_index/_search
{
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1.0, 1.0],
        "topk": 10,
        "filter": {
          "terms": { "country": ["cn", "eu"] }
        }
      }
    }
  }
}
```

```
# 多标签匹配查询示例
GET my_index/_search
{
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1.0, 1.0],
        "topk": 10,
        "filter": {
          "must": [
            {
              "term": {"country": "cn"}
            },
            {
              "terms": {"category": ["1", "2"]}
            }
          ]
        }
      }
    }
  }
}

# must_not匹配查询示例
GET my_index/_search
{
  "query": {
    "vector": {
      "my_vector": {
        "vector": [1.0, 1.0],
        "topk": 10,
        "filter": {
          "must_not": [
            {
              "term": {"country": "eu"}
            }
          ]
        }
      }
    }
  }
}
```

向量查询参数说明请见[表2-52](#)。

2.7.3 配置 Elasticsearch 集群存算分离

存算分离支持将集群新写入的数据存储在SSD来达到最佳的查询检索性能，将历史数据存储在OBS中降低数据的存储成本。

场景描述

存算分离是通过冻结索引将历史数据转储到OBS，以此降低数据的存储成本。

对于有海量数据写入和存储的场景，数据有明显的冷热区分，新写入的数据一般属于热数据，存储在SSD中方便快速被搜索，随着时间的推移，历史数据不再写入，查询QPS也降低，这时候历史数据就算冷数据，通过存算分离将冷数据转储到OBS，热数据依旧存放在SSD中，方便快速搜索。

约束限制

- 仅Elasticsearch 7.6.2、Elasticsearch 7.10.2、OpenSearch 1.3.6集群支持存算分离。

- 在执行冻结索引前，系统会将待冻结的索引配置为只读状态，即便在索引数据转储到OBS后，索引也是只读状态，不支持写入数据。
- 在冻结索引过程中，查询请求不受影响。在冻结完成后，会将索引先Close再Open，在这段时间内，索引不可查询，集群可能短暂出现Red状态，索引Open结束后集群恢复正常。
- 将历史索引数据转储到OBS的过程中会占用网络带宽。
- 冻结索引后，索引会被转储到OBS中，同时本地磁盘中的索引数据会被删除。当查询被冻结的索引时，时延会增加，在聚合时，由于查询复杂、数据读取多，时延变长会更明显。
- 已转储到OBS中的索引不支持解冻，即不可回退为可写的索引。
- 由于存算分离的特性依赖OBS，所以使用过程中要遵守OBS的“带宽”和“每秒请求数（QPS）”的使用限制，详细请参见[OBS使用限制](#)。当超过限制时，集群中涉及到OBS查询的性能都会下降，例如恢复分片的速度变慢、查询数据时变慢等。

访问集群

本文通过Kibana访问集群，介绍如何使用存算分离。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，选择需要进行存算分离的集群，单击操作列“Kibana”，登录Kibana页面。
3. 选择左侧导航栏的“Dev Tools”，进入操作页面。

冻结索引

执行如下命令，将指定索引转储到OBS中。

```
POST ${index_name}/_freeze_low_cost
```

表 2-57 请求参数说明

参数名	说明
index_name	需要冻结的索引名称。

返回结果如下：

```
{
  "freeze_uuid": "pdsRgUtSTymVDWR_HoTGFw"
}
```

表 2-58 返回参数说明

参数名	说明
freeze_uuid	冻结索引任务的ID，在提交冻结请求后会启动一个异步任务，请求返回任务ID，使用该ID可以查询冻结索引任务的进度。

查看冻结索引任务

执行如下命令，查看冻结索引任务的进度。

```
GET _freeze_low_cost_progress/${freeze_uuid}
```

表 2-59 请求参数说明

参数名	说明
freeze_uuid	冻结索引任务的ID，该ID由 冻结索引 时获取。

返回结果如下：

```
{
  "stage": "STARTED",
  "shards_stats": {
    "INIT": 0,
    "FAILURE": 0,
    "DONE": 0,
    "STARTED": 3,
    "ABORTED": 0
  },
  "indices": {
    "data1": [
      {
        "uuid": "7OS-G1-tRke2jHZPlckexg",
        "index": {
          "name": "data1",
          "index_id": "4b5PHXJITLaS6AurlmfQ9A",
          "shard": 2
        },
        "start_ms": 1611972010852,
        "end_ms": -1,
        "total_time": "10.5s",
        "total_time_in_millis": 10505,
        "stage": "STARTED",
        "failure": null,
        "size": {
          "total_bytes": 3211446689,
          "finished_bytes": 222491269,
          "percent": "6.0%"
        },
        "file": {
          "total_files": 271,
          "finished_files": 12,
          "percent": "4.0%"
        },
        "rate_limit": {
          "paused_times": 1,
          "paused_nanos": 946460970
        }
      },
      {
        "uuid": "7OS-G1-tRke2jHZPlckexg",
        "index": {
          "name": "data1",
          "index_id": "4b5PHXJITLaS6AurlmfQ9A",
          "shard": 0
        },
        "start_ms": 1611972010998,
        "end_ms": -1,
        "total_time": "10.3s",
        "total_time_in_millis": 10359,
        "stage": "STARTED",
```

```

"failure" : null,
"size" : {
  "total_bytes" : 3221418186,
  "finished_bytes" : 272347118,
  "percent" : "8.0%"
},
"file" : {
  "total_files" : 372,
  "finished_files" : 16,
  "percent" : "4.0%"
},
"rate_limit" : {
  "paused_times" : 5,
  "paused_nanos" : 8269016764
}
},
{
  "uuid" : "7OS-G1-tRke2jHZPlckexg",
  "index" : {
    "name" : "data1",
    "index_id" : "4b5PHXJITLaS6AurlmfQ9A",
    "shard" : 1
  },
  "start_ms" : 1611972011021,
  "end_ms" : -1,
  "total_time" : "10.3s",
  "total_time_in_millis" : 10336,
  "stage" : "STARTED",
  "failure" : null,
  "size" : {
    "total_bytes" : 3220787498,
    "finished_bytes" : 305789614,
    "percent" : "9.0%"
  },
  "file" : {
    "total_files" : 323,
    "finished_files" : 14,
    "percent" : "4.0%"
  },
  "rate_limit" : {
    "paused_times" : 3,
    "paused_nanos" : 6057933087
  }
}
]
}
}

```

表 2-60 返回参数说明

参数名	说明
stage	当前所处状态。取值包括： <ul style="list-style-type: none"> INIT：刚启动或者正在初始化。 FAILURE：失败。 DONE：完成。 STARTED：已启动。 ABORTED：取消，预留字段。
shards_stats	处在各个状态的shard个数。
indices	每个索引的状态详情。

表 2-61 indices 返回值说明

参数名	说明
uuid	freeze的uuid。
index	索引信息和shard信息。
start_ms	开始时间。
end_ms	结束时间，如果没有结束则显示为-1。
total_time	已花费时间。
total_time_in_millis	已花费时间毫秒数。
stage	当前shard所处的状态。
failure	失败原因，如果没有失败则显示为null。
size.total_bytes	总共需要冻结的文件的字节数。
size.finished_bytes	已经完成冻结的字节数。
size.percent	已经完成冻结的字节数百分比。
file.total_bytes	总共需要冻结的文件个数。
file.finished_bytes	已经完成冻结的文件个数。
file.percent	已经完成冻结的文件个数百分比。
rate_limit.paused_times	达到限速导致冻结暂停的次数。
rate_limit.paused_nanos	达到限速导致冻结暂停的时间纳秒数。

冻结完成的索引会增加以下settings，可参考表2-62。

表 2-62 冻结索引 settings

参数	说明
index.frozen_low_cost	标识该索引为冻结索引。取值为true。
index.blocks.write	冻结后的索引禁止写入。取值为true。
index.store.type	标识该索引的存储类型为obs。取值为obs。

基于冻结状态查询索引列表

执行如下命令，基于冻结状态查询索引列表。

```
GET _cat/freeze_indices?stage=${STAGE}
```

表 2-63 请求参数说明

参数名	说明
STAGE	索引的冻结状态，可选值如下： <ul style="list-style-type: none"> start：开始冻结但是还未完成的索引列表 done：已经完成冻结的索引列表 unfreeze：没有冻结的索引列表 空或者其他值：正在冻结过程中或者已经冻结完成的所有索引列表

返回结果如下：

```
green open data2 0bNtxWDtRbOSkS4JYaUgMQ 3 0 5 0 7.9kb 7.9kb
green open data3 oYMLvw31QnyasqUNuyP6RA 3 0 51 0 23.5kb 23.5kb
```

📖 说明

此命令的参数和返回值与开源Elasticsearch的_cat/indices一致。

修改存储在 OBS 中冷数据的缓存配置

索引数据转储到OBS后，为了尽可能的减少对OBS的访问请求，并提升集群的查询性能，系统会缓存部分数据。第一次查询冷数据时，集群会直接访问OBS，获取到的数据会被缓存在集群内存中，后续查询时会先检查是否有缓存数据。

集群访问不同的文件，访问的模式是不一样的，缓存系统支持多级缓存，会使用不同大小的block来缓存不同的文件，例如fdx和tip文件会使用大量的小block缓存，对fdt文件会使用较少的大block缓存。缓存配置支持基于业务情况进行修改，配置项请参见[表 2-64](#)。

表 2-64 缓存的配置项

配置项	类型	说明
low_cost.obs.blockcache.names	Array	缓存系统支持多级缓存，分别用来缓存不同访问粒度的数据。此配置列出所有缓存的名字，即使不配置，系统也会默认有一个缓存，名字为default。如果自定义配置，请确保有一个名字为default的缓存，其他名字任意。 默认值：default。
low_cost.obs.blockcache.<NAME>.type	ENUM	缓存的类型，支持memory和file。 当使用memory类型的缓存时，会占用一定的内存大小。当使用file类型的缓存时，会使用磁盘作为缓存。建议使用超高IO型的磁盘提升缓存性能。 默认值：memory。

配置项	类型	说明
low_cost.obs.blockcache.<NAME>.blockshift	Integer	缓存每个block的大小，为字节左移数，即 2^x 字节。如配置为16，表示block大小为 2^{16} 字节，等于65536字节，即64K。 默认值：13（即8K）。
low_cost.obs.blockcache.<NAME>.bank.count	Integer	缓存分区数。 默认值：1。
low_cost.obs.blockcache.<NAME>.number.blocks.perbank	Integer	每个缓存分区中包含的block数。 默认值：8192。
low_cost.obs.blockcache.<NAME>.exclude.file.types	Array	不缓存的文件后缀名。如果某些后缀既不包含在exclude列表，也不包含在include列表，则会使用default缓存。
low_cost.obs.blockcache.<NAME>.file.types	Array	缓存的文件后缀名。如果某些后缀既不包含在exclude列表，也不包含在include列表，则会使用default缓存。
index.frozen.obs.max_bytes_per_sec	String	冻结过程中往OBS上传文件最大限速。动态配置，修改后立即生效。 默认值：150MB。
low_cost.obs.index.upload.threshold.use.multipart	String	冻结过程中文件大小超过此配置会使用OBS的分段上传。 默认值：1GB。
index.frozen.reader.cache.expire.duration.seconds	Integer	此参数设置超时时间。 为了减少冻结后的索引占用的堆内存，在索引shard启动后，reader会缓存一段时间，超时后关闭。 默认值：300s。
index.frozen.reader.cache.max.size	Integer	配置缓存最大值。 默认值：100。

下面是一个常见的缓存配置，该配置使用两级缓存，名字分别为“default”和“large”。其中，“default”缓存使用大小为64K的block，一共有 30×4096 个block，“default”缓存用于缓存除fdt后缀的其他文件，“large”缓存使用大小为2M的block，一共有 5×1000 个block，“large”缓存用于缓存fdx、dvd和tip后缀的文件。

```
low_cost.obs.blockcache.names: ["default", "large"]
low_cost.obs.blockcache.default.type: file
low_cost.obs.blockcache.default.blockshift: 16
low_cost.obs.blockcache.default.number.blocks.perbank: 4096
low_cost.obs.blockcache.default.bank.count: 30
low_cost.obs.blockcache.default.exclude.file.types: ["fdt"]
```

```
low_cost.obs.blockcache.large.type: file
low_cost.obs.blockcache.large.blockshift: 21
```

```
low_cost.obs.blockcache.large.number.blocks.perbank: 1000
low_cost.obs.blockcache.large.bank.count: 5
low_cost.obs.blockcache.large.file.types: ["fdx", "dvd", "tip"]
```

查询存储在 OBS 中冷数据的缓存状态

冻结索引之后，当冷数据第一次被查询时，系统会自动进行数据缓存，集群支持查询存储在OBS中冷数据的缓存状态。当需要调试集群性能时，也可以重置缓存状态。

1. 查询节点中存储在OBS中冷数据的缓存状态。

- 执行如下命令，查询所有节点中冷数据的缓存信息。

```
GET _frozen_stats
```

- 执行如下命令，查询指定节点中冷数据的缓存信息。

```
GET _frozen_stats/${node_id}
```

表 2-65 请求参数说明

参数名	说明
node_id	节点ID。

返回结果如下：

```
{
  "_nodes" : {
    "total" : 3, // 所有节点数量
    "successful" : 3, // 成功的节点数量
    "failed" : 0 // 失败的节点数量
  },
  "cluster_name" : "css-zzz1", // 集群名称
  "nodes" : {
    "7uwKO38RRoaON37YsXhCYw" : {
      "name" : "css-zzz1-ess-esn-2-1", // 节点名称
      "transport_address" : "10.0.0.247:9300", // 节点transport连接地址
      "host" : "10.0.0.247", // 节点host
      "ip" : "10.0.0.247", // 节点IP
      "block_cache" : {
        "default" : {
          "type" : "memory", // cache类型，memory表示内存cache
          "block_cache_capacity" : 8192, // cache容量
          "block_cache_blocksize" : 8192, // cache中单个block大小，单位为bytes，如此处表示8k
          "block_cache_size" : 12, // cache已被占用的大小
          "block_cache_hit" : 14, // cache命中数
          "block_cache_miss" : 0, // cache未命中数
          "block_cache_eviction" : 0, // cache被逐出次数
          "block_cache_store_fail" : 0 // 如果cache满了，可能导致新的cache存储失败，此处表示存储失败
            的次数
        }
      },
    },
    "obs_stats" : {
      "list" : {
        "obs_list_count" : 17, // 调用OBS list接口的次数
        "obs_list_ms" : 265, // 调用OBS list接口的总耗时
        "obs_list_avg_ms" : 15 // 调用OBS list接口的平均耗时
      },
      "get_meta" : {
        "obs_get_meta_count" : 79, // 调用OBS 获取元数据接口的次数
        "obs_get_meta_ms" : 183, // 调用OBS 获取元数据接口的总耗时
        "obs_get_meta_avg_ms" : 2 // 调用OBS 获取元数据接口的平均耗时
      },
      "get_obj" : {
        "obs_get_obj_count" : 12, // 调用OBS getObject接口的次数
        "obs_get_obj_ms" : 123, // 调用OBS getObject接口的总耗时
        "obs_get_obj_avg_ms" : 10 // 调用OBS getObject接口的平均耗时
      }
    }
  }
}
```

```
},
"put_obj" : {
  "obs_put_obj_count" : 12, // 调用OBS putObject接口的总次数
  "obs_put_obj_ms" : 2451, // 调用OBS putObject接口的总耗时
  "obs_put_obj_avg_ms" : 204 // 调用OBS putObject接口的平均耗时
},
"obs_op_total" : {
  "obs_op_total_ms" : 3022, // 调用OBS接口的总耗时
  "obs_op_total_count" : 120, // 调用OBS接口的总次数
  "obs_op_avg_ms" : 25 // 调用OBS接口的平均耗时
}
},
"reader_cache" : {
  "hit_count" : 0,
  "miss_count" : 1,
  "load_success_count" : 1,
  "load_exception_count" : 0,
  "total_load_time" : 291194714,
  "eviction_count" : 0
}
},
"73EDpEqoQES749umJqxOzQ" : {
  "name" : "css-zzz1-ess-esn-3-1",
  "transport_address" : "10.0.0.201:9300",
  "host" : "10.0.0.201",
  "ip" : "10.0.0.201",
  "block_cache" : {
    "default" : {
      "type" : "memory",
      "block_cache_capacity" : 8192,
      "block_cache_blocksize" : 8192,
      "block_cache_size" : 12,
      "block_cache_hit" : 14,
      "block_cache_miss" : 0,
      "block_cache_eviction" : 0,
      "block_cache_store_fail" : 0
    }
  },
  "obs_stats" : {
    "list" : {
      "obs_list_count" : 17,
      "obs_list_ms" : 309,
      "obs_list_avg_ms" : 18
    },
    "get_meta" : {
      "obs_get_meta_count" : 79,
      "obs_get_meta_ms" : 216,
      "obs_get_meta_avg_ms" : 2
    },
    "get_obj" : {
      "obs_get_obj_count" : 12,
      "obs_get_obj_ms" : 140,
      "obs_get_obj_avg_ms" : 11
    },
    "put_obj" : {
      "obs_put_obj_count" : 12,
      "obs_put_obj_ms" : 1081,
      "obs_put_obj_avg_ms" : 90
    },
    "obs_op_total" : {
      "obs_op_total_ms" : 1746,
      "obs_op_total_count" : 120,
      "obs_op_avg_ms" : 14
    }
  },
  "reader_cache" : {
    "hit_count" : 0,
    "miss_count" : 1,
    "load_success_count" : 1,
```



```
"load_exception_count" : 0,
"total_load_time" : 367179751,
"eviction_count" : 0
}
},
"EF8WoLCUQbqJl1Pkqo9-OA" : {
  "name" : "css-zzz1-ess-esn-1-1",
  "transport_address" : "10.0.0.18:9300",
  "host" : "10.0.0.18",
  "ip" : "10.0.0.18",
  "block_cache" : {
    "default" : {
      "type" : "memory",
      "block_cache_capacity" : 8192,
      "block_cache_blocksize" : 8192,
      "block_cache_size" : 12,
      "block_cache_hit" : 14,
      "block_cache_miss" : 0,
      "block_cache_eviction" : 0,
      "block_cache_store_fail" : 0
    }
  },
  "obs_stats" : {
    "list" : {
      "obs_list_count" : 17,
      "obs_list_ms" : 220,
      "obs_list_avg_ms" : 12
    },
    "get_meta" : {
      "obs_get_meta_count" : 79,
      "obs_get_meta_ms" : 139,
      "obs_get_meta_avg_ms" : 1
    },
    "get_obj" : {
      "obs_get_obj_count" : 12,
      "obs_get_obj_ms" : 82,
      "obs_get_obj_avg_ms" : 6
    },
    "put_obj" : {
      "obs_put_obj_count" : 12,
      "obs_put_obj_ms" : 879,
      "obs_put_obj_avg_ms" : 73
    },
    "obs_op_total" : {
      "obs_op_total_ms" : 1320,
      "obs_op_total_count" : 120,
      "obs_op_avg_ms" : 11
    }
  },
  "reader_cache" : {
    "hit_count" : 0,
    "miss_count" : 1,
    "load_success_count" : 1,
    "load_exception_count" : 0,
    "total_load_time" : 235706838,
    "eviction_count" : 0
  }
}
}
```

2. 执行如下命令，可以重置缓存状态。

```
POST _frozen_stats/reset
```

说明

此命令主要用于调试性能问题，例如重置缓存状态后再次执行查询可以清晰看到本次查询的缓存命令情况。在业务运行阶段不需要执行此命令。

返回结果如下：

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "Es-0325-007_01",
  "nodes": {
    "mqTdk2YRSPyOSXfesREFSg": {
      "result": "ok"
    }
  }
}
```

提升冷数据的查询性能

须知

仅2023年02月后创建的Elasticsearch 7.6.2、Elasticsearch 7.10.2和OpenSearch 1.3.6集群支持提升冷数据的查询性能。

在Kibana的Discover页面首次查询冷数据时，由于此时无任何缓存，导致所有数据均需要从OBS上获取，当命中的数据较多时，需要耗费大量的时间从OBS上获取对应的时间字段以及文件元数据。如果将这一部分数据直接缓存在本地，即可大幅提升查询性能，解决Discover页面首次查询慢的问题。云搜索服务就是通过冷数据的本地缓存，实现冷数据的查询性能提升。本地缓存配置是预置的，用户可以基于业务场景修改配置，也可以查询、了解本地缓存信息。

1. 修改冷数据的本地缓存配置。

表 2-66 本地缓存配置项说明

配置项	类型	scope	是否可动态修改	说明
low_cost.local_cache.max.capacity	Integer	node	是	<p>节点上，能够打开的冷数据缓存的最大数量。（每个shard对应一个缓存对象）</p> <p>取值范围：10~5000</p> <p>默认值：500</p> <p>说明</p> <ul style="list-style-type: none">• 当堆内存使用率一直很高时，可以尝试降低该值。• 当查询冷数据本地缓存的相关统计指标中load_overflow_count数值一直持续快速增加时，建议调大该值。

配置项	类型	scope	是否可动态修改	说明
index.low_cost.local_cache.threshold	Integer	index	是	<p>启用冷数据本地缓存的阈值。</p> <ul style="list-style-type: none"> 当date类型字段的占比小于此值时，启用冷数据本地缓存date类型字段。否则不使用。 如果当前索引的date类型字段占据当前索引的绝大部分数据量，则不建议使用此功能。 <p>单位：% 取值范围：0~100 默认值：50</p>
index.low_cost.local_cache.evict_time	字符串	index	是	<p>冷数据的本地缓存的淘汰时间，当缓存时间大于该值，缓存将被删除。根据index.frozen_date（冻结成功的时间）判定，当索引无index.frozen_date时，则根据索引创建时间判定。</p> <p>单位：天 取值范围：1d~365d 默认值：30d</p> <p>说明 建议根据磁盘用量调整淘汰时长，节省磁盘空间。</p>

- 执行如下命令，修改“low_cost.local_cache.max.capacity”。

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.local_cache.max.capacity": 1000
  }
}
```

- 执行如下命令，修改“index.low_cost.local_cache.threshold”。

```
PUT es_write_pref2-00000000021/_settings
{
  "index.low_cost.local_cache.threshold": 20
}
```

- 执行如下命令，修改“index.low_cost.local_cache.evict_time”。

```
PUT es_write_pref2-00000000021/_settings
{
  "index.low_cost.local_cache.evict_time": "7d"
}
```

2. 查询冷数据的本地缓存信息。

- 执行如下命令，查询所有节点中冷数据的本地缓存相关指标。

```
GET _/frozen_stats/local_cache
```

- 执行如下命令，查询指定节点中冷数据的本地缓存相关指标。

```
GET _/frozen_stats/local_cache/{nodeId}
```

“{nodeId}”为节点ID。

返回结果如下:

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "elasticsearch",
  "nodes": {
    "6by3lPy1R3m55Dcq3lik8Q": {
      "name": "node-1",
      "transport_address": "127.0.0.1:9300",
      "host": "127.0.0.1",
      "ip": "127.0.0.1",
      "local_cache": {
        "get_stats": {
          "get_total_count": 562,           //从冷数据本地缓存查询数据的总次数
          "get_hit_count": 562,           //从冷数据本地缓存查询数据命中的次数
          "get_miss_count": 0,           //从冷数据本地缓存查询数据未命中的次数
          "get_total_ns": 43849200,      //从冷数据本地缓存查询数据的总时长
          "get_avg_ns": 78023            //从冷数据本地缓存查询数据的平均时长
        },
        "load_stats": {
          "load_count": 2,                //加载冷数据本地缓存的次数
          "load_total_ms": 29,           //加载冷数据本地缓存的总时长
          "load_avg_ms": 14,            //加载冷数据本地缓存的平均时长
          "load_fail_count": 0,         //加载冷数据本地缓存的失败次数
          "load_overflow_count": 0      //加载冷数据本地缓存时超过缓存池的次数
        },
        "reload_stats": {
          "reload_count": 0,             //重新生成冷数据本地缓存的次数
          "reload_total_ms": 0,         //重新生成冷数据本地缓存的总时长
          "reload_avg_ms": 0,          //重新生成冷数据本地缓存的平均时长
          "reload_fail_count": 0        //重新生成冷数据本地缓存的失败次数
        },
        "init_stats": {
          "init_count": 0,              //初始化冷数据本地缓存的次数
          "init_total_ms": 0,          //初始化冷数据本地缓存的总时长
          "init_avg_ms": 0,            //初始化冷数据本地缓存的平均时长
          "init_fail_count": 0         //初始化冷数据本地缓存的失败次数
        }
      }
    }
  }
}
```

查询存储冷数据的 OBS 实时速率

须知

仅2023年02月后创建的Elasticsearch 7.6.2、Elasticsearch 7.10.2和OpenSearch 1.3.6 集群支持提升冷数据的查询性能。

为了更清晰的了解到存算分离的插件在OBS中的使用情况，CSS服务新增了OBS实时速率的统计接口，并且将实时速率记录到系统索引“.freeze_obs_rate-YYYY.mm.dd”中。

OBS实时速率的计算方式：每5秒计算一次，查询前5秒内的平均OBS操作速率。

系统索引“.freeze_obs_rate-YYYY.mm.dd”用于存放OBS操作实时速率和OBS操作数据，便于了解存储冷数据的OBS的操作趋势。该索引的默认保留时间是30天。

1. 查询存储冷数据的OBS实时速率。

- 执行如下命令，查询所有节点中存储冷数据的OBS实时速率。

```
GET _frozen_stats/obs_rate
```

- 执行如下命令，查询指定节点中存储冷数据的OBS实时速率。

```
GET _frozen_stats/obs_rate/{nodeId}
```

“{nodeId}”为节点ID。

响应示例：

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "elasticsearch",
  "nodes": {
    "df1DvcSwTJ-fkillT2zE3A": {
      "name": "node-1",
      "transport_address": "127.0.0.1:9300",
      "host": "127.0.0.1",
      "ip": "127.0.0.1",
      "update_time": 1671777600482, // 当前统计值的更新时间。
      "obs_rate": {
        "list_op_rate": 0.0, // obs list操作的速率，单位：次/秒。
        "get_meta_op_rate": 0.0, // obs get meta操作的速率，单位：次/秒。
        "get_obj_op_rate": 0.0, // obs get操作的速率，单位：次/秒。
        "put_op_rate": 0.0, // obs put操作的速率，单位：次/秒。
        "obs_total_op_rate": 0.0, // obs所有操作的速率，单位：次/秒。
        "obs_upload_rate": "0.0 MB/s", // obs上传数据的速率，MB/秒。
        "obs_download_rate": "0.0 MB/s" // obs下载数据的速率，MB/秒。
      }
    }
  }
}
```

2. 修改存储OBS实时速率的索引“.freeze_obs_rate-YYYY.mm.dd”的保留时间。索引的默认保留时间是30天。

执行如下命令，将索引保留时间改成7天。

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.obs_rate_index.evict_time": "7d"
  }
}
```

表 2-67 配置项说明

配置项	类型	scope	是否可动态修改	说明
low_cost.obs_rate_index.evict_time	String	node	是	用于控制索引“.freeze_obs_rate-YYYY.mm.dd”的保留时间。 <ul style="list-style-type: none"> 取值范围：1d~365d。 默认值：30d。 单位：天。

2.7.4 配置 Elasticsearch 集群读写流量控制策略 2.0

场景描述

流量控制2.0提供节点级别的流量控制功能，可提供单个节点基于黑白名单的访问限制、HTTPS并发连接数限制、HTTP最大连接数限制，基于节点内存的客户端写入流量反压控制，一键断流能力。同时也提供节点访问IP统计和URI的统计能力。开启客户端写入流量反压控制功能，会在节点堆内存过大时阻止大请求涌入节点占用内存，避免节点崩溃，减少节点不可用的风险。

表 2-68 流量控制策略

策略	描述	相关文档
HTTP/HTTPS流控	<ul style="list-style-type: none">• HTTP/HTTPS黑白名单设置IP和子网控制客户端IP访问，如果节点IP在黑名单中，则该客户端的连接将直接中断，节点不会处理任何请求。白名单规则优先于黑名单规则，如果客户端IP在黑白名单中都出现，客户端请求将不会被拒绝。• HTTP/HTTPS并发连接数流控通过限制节点每秒中的HTTP连接总数来限制节点流量。• HTTP/HTTPS新建连接数流控通过限制节点新建的连接数来限制节点流量。	开启HTTP/HTTPS节点流控
内存流控	基于节点堆内存使用量限制写入流量，将请求需要读取的内容反压在客户端，暂停请求的接收。同时尽力触发垃圾回收，按堆内存的空闲量继续读取请求。	开启内存流控
请求采样统计	可以记录客户端IP的访问和客户端的请求类型，用户可以基于统计值识别客户端IP的访问流量，分析当前客户端的写入和查询访问量。	开启请求采样统计
一键断流	可以切断节点的所有客户端连接，不包括Kibana访问和CSS后台运维、监控类接口。用于应对突发流量场景下的集群异常，达到快速恢复集群的目的。	开启一键断流
流量控制	提供单独的流量统计接口，记录当前客户端连接数以及客户端反压连接数，用户可以基于统计值评估流控配置阈值和衡量集群压力。	查看流量控制信息
访问日志	可以记录一段时间内节点接收的HTTP/HTTPS请求URL和Body，用户可以基于访问日志信息分析当前的流量压力。	开启并查看访问日志
访问日志记录	客户端访问集群节点，会记录为{集群名_access_log.log}访问日志文件，通过日志备份功能可以在OBS详细查看访问日志。	开启访问日志记录到文件

约束限制

- 2023年02月及之后创建的Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制2.0版本，之前创建Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制1.0版本。
- 开启流量控制功能会消耗部分节点性能。
- 开启流量控制会直接拒绝超过阈值的用户请求。
- 开启内存流控会消耗部分请求性能，开启内存流控会导致Kibana的部分search请求失败。
- 开启访问日志会影响集群性能。
- 内存流控是基于请求Path的流控，Path长度和个数不应该配置过多，否则影响集群性能。

开启 HTTP/HTTPS 节点流控

1. 执行如下命令，开启HTTP/HTTPS节点流控。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": true,
    "flowcontrol.http.allow": ["192.168.0.1/24", "192.168.2.1/24"],
    "flowcontrol.http.deny": "192.168.1.1/24",
    "flowcontrol.http.concurrent": 1000,
    "flowcontrol.http.newconnect": 1000,
    "flowcontrol.http.warmup_period": 0
  }
}
```

表 2-69 HTTP/HTTPS 流控的配置项说明

配置项	类型	说明
flowcontrol.http.enabled	Boolean	HTTP/HTTPS流控开关，默认关闭，开启会影响节点访问性能。 取值范围：true、false 默认值：false
flowcontrol.http.allow	List<String>	IP地址访问白名单。 支持配置多个IP地址和掩码，或者IP地址列表形式，中间用英文逗号隔开。例如“xx.xx.xx.xx/24,xx.xx.xx.xx/24”或“xx.xx.xx.xx,xx.xx.xx.xx”形式。 默认值为空。
flowcontrol.http.deny	List<String>	IP访问黑名单。 支持配置多个IP和掩码，或者IP列表形式，中间用英文逗号隔开。 默认值为空。
flowcontrol.http.concurrent	Integer	HTTP/HTTPS请求的并发连接数阈值。 默认值：节点可用核数x400

配置项	类型	说明
flowcontrol.http.newconnect	Integer	HTTP/HTTPS请求的每秒可以创建的新建连接数阈值。 默认值：节点可用核数x200
flowcontrol.http.warmup_period	Integer	HTTP/HTTPS新建连接数达到最大速率的所需的时间，如果“flowcontrol.http.newconnect”配置为“100”且“flowcontrol.http.warmup_period”配置为“5000ms”，表示5s以后系统的新建连接数才可以达到每秒100。 取值范围：0~10000 单位：ms 默认值：0

📖 说明

当所有参数指设置为“null”时，表示恢复配置默认值。

2. 执行如下命令，关闭HTTP/HTTPS节点流控。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": false
  }
}
```

开启内存流控

1. 执行如下命令，开启内存流控。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.memory.enabled": true,
    "flowcontrol.memory.heap_limit": "80%"
  }
}
```

表 2-70 内存流控的配置项说明

配置项	类型	说明
flowcontrol.memory.enabled	Boolean	内存流控开关，开启后持续监控内存使用情况。 取值范围： <ul style="list-style-type: none">• true• false（默认值）

配置项	类型	说明
flowcontrol.memory.heap_limit	String	<p>节点堆内存的阈值，限制节点全局堆内存的使用率，超过此值将进行流量反压。</p> <p>取值范围：10%-100%</p> <p>默认值：90%</p> <p>说明</p> <ul style="list-style-type: none">默认值90%是一个比较保守的阈值，即堆内存大于90%使用量时会停止读取客户端超过64KB的大请求，直至堆内存下降。如堆内存下降到85%，会开始允许最多一次读取“5%×堆内存最大值”的客户端数据量。如果堆内存持续超过90%，则无法放开客户端连接的请求读取，此时会尝试触发GC算法进行垃圾回收，直到堆内存低于所设定的阈值。日常使用时可以将“flowcontrol.memory.heap_limit”阈值设置为80%或以下，保证节点有一定的堆内存余量，供写入内存以外的行为使用，比如：Elasticsearch查询、Segment merge等。
flowcontrol.holding.in_flight_factor	Float	<p>反压释放因子，原理类似于熔断器“network.breaker.inflight_requests.overhead”参数。内存达到限制值时，该值越大反压越强，写入流量将受限。</p> <p>取值范围：≥0.5</p> <p>默认值：1.0</p>
flowcontrol.holding.max	TimeValue	<p>每个请求最长的延迟时间，当延迟超过此值时可以设置断开该请求反压或断开请求链路。详见“flowcontrol.holding.max_strategy”配置。</p> <p>取值范围：≥15s</p> <p>默认值：60s</p>
flowcontrol.holding.max_strategy	String	<p>超过最大延迟时间后的运行策略。</p> <p>取值范围：</p> <ul style="list-style-type: none">keep（默认值）：如果堆内存仍在高位，选择继续反压 - 何时执行请求仍由服务器根据实时内存自主决定。soft：如果堆内存仍在高位，也必须执行该请求。执行/拒绝权力交给inFlight熔断器。hard：如果堆内存仍在高位，丢弃该请求，同时断开该请求的客户端连接。

配置项	类型	说明
flowcontrol.memory.once_free_max	String	被暂停的请求队列一次性最大重新打开的内存，防止强压场景下短暂的低内存现象一次性冲挂集群。 取值范围：1%-50% 默认值：10%
flowcontrol.memory.nudges_gc	Boolean	写入压力过大时（1s检查一次反压连接池，所有现有连接均被阻塞无法放开新的写入请求），是否尽力触发垃圾回收，保证写入稳定性。取值： <ul style="list-style-type: none">• true（默认值）• false

📖 说明

当所有参数指设置为“null”时，表示恢复配置默认值。

2. 执行如下命令，关闭集群内存流控。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.memory.enabled": false
  }
}
```

开启请求采样统计

1. 执行如下命令，开启请求采样统计。

```
PUT _cluster/settings
{
  "transient": {
    "flowcontrol.log.access.enabled": true
  }
}
```

表 2-71 请求采样统计的配置项说明

配置项	类型	说明
flowcontrol.log.access.enabled	Boolean	请求采样统计开关，统计最近访问Elasticsearch集群的客户端IP地址及其请求数量，会统计bulk写入和search、msearch查询请求的个数。 取值范围： <ul style="list-style-type: none">• true• false（默认值）

配置项	类型	说明
flowcontrol.log.access.count	Integer	统计最近IP地址访问集群的客户端来源个数。IP统计分为请求类型统计和日志记录开关。 取值范围：0-100 默认值：10

📖 说明

当所有参数指设置为“null”时，表示恢复配置默认值。

2. 执行如下命令，关闭请求采样统计。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.access.enabled": false
  }
}
```

开启一键断流

1. 执行如下命令，开启一键断流。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": true
  }
}
```

2. 执行如下命令，关闭一键断流。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": false
  }
}
```

查看流量控制信息

- 查看所有节点的流量控制情况
GET /_nodes/stats/filter/v2
- 查看所有节点的流量控制详细情况
GET /_nodes/stats/filter/v2?detail
- 查看某个具体节点的流量控制情况
GET /_nodes/{nodeId}/stats/filter/v2

“{nodeId}”为需要查看流量控制的节点ID。

响应示例：

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-xxxx",
  "nodes" : {
    "d3qnVlpPTtSoadkV0LQEKA" : {
      "name" : "css-xxxx-ess-esn-1-1",
```

```
"host" : "192.168.x.x",
"timestamp" : 1672236425112,
"flow_control" : {
  "http" : {
    "current_connect" : 52,
    "rejected_concurrent" : 0,
    "rejected_rate" : 0,
    "rejected_black" : 0,
    "rejected_breaker" : 0
  },
  "access_items" : [
    {
      "remote_address" : "10.0.0.x",
      "search_count" : 0,
      "bulk_count" : 0,
      "other_count" : 4
    }
  ],
  "holding_requests" : 0
}
}
```

表 2-72 响应参数说明

参数名	说明
current_connect	节点实际的HTTP连接数据信息，没有开启流控这个配置也会记录，等同于GET /_nodes/stats/http接口current_open值，可以看到节点当前的客户端连接数。
rejected_concurrent	HTTP流控开启生效，关闭后不清零，开启流控期间拒绝的并发连接数。
rejected_rate	HTTP流控开启生效，关闭后不清零，开启流控期间拒绝的新建连接数。
rejected_black	HTTP流控开启生效，关闭后不清零，配置黑名单拒绝的请求数。
rejected_breaker	一键断流开启，拒绝的新建连接数。
remote_address	IP地址统计，基于配置值统计节点访问的IP地址和请求数量。
search_count	该客户端以_search、_msearch访问的次数。
bulk_count	该客户端以_bulk访问的次数。
other_count	该客户端其他请求的访问次数。

开启并查看访问日志

1. 执行如下命令，开启访问日志。
 - 开启集群所有节点的访问日志
PUT /_access_log?duration_limit=30s&capacity_limit=1mb
 - 开启集群中某一节点的访问日志
PUT /_access_log/{nodeId}?duration_limit=30s&capacity_limit=1mb
“{nodeId}”为节点ID。

表 2-73 访问日志的配置项说明

配置项	类型	说明
duration_limit	String	访问日志记录时间。 取值范围：10~120 单位：s 默认值：30
capacity_limit	String	访问日志记录大小。统计开启访问日志后记录的请求大小，当统计的大小大于该配置值，访问日志记录终止。 取值范围：1~5 单位：MB 默认值：1

说明

- “duration_limit”和“capacity_limit”只要有一个参数达到阈值，访问日志记录就会停止。
 - 当所有参数指设置为“null”时，表示恢复配置默认值。
2. 执行如下命令，查看访问日志。

– 查看集群所有节点的访问日志API

```
GET /_access_log
```

– 查看集群中某一节点的访问日志API

```
GET /_access_log/{nodeId}
```

“{nodeId}”为节点ID。

响应示例：

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "css-flowcontroller",
  "nodes": {
    "8x-ZHu-wTemBQwpcGivFKg": {
      "name": "css-flowcontroller-ess-esn-1-1",
      "host": "10.0.0.98",
      "count": 2,
      "access": [
        {
          "time": "2021-02-23 02:09:50",
          "remote_address": "/10.0.0.98:28191",
          "url": "/_access/security/log?pretty",
          "method": "GET",
          "content": ""
        },
        {
          "time": "2021-02-23 02:09:52",
          "remote_address": "/10.0.0.98:28193",
          "url": "/_access/security/log?pretty",
          "method": "GET",
          "content": ""
        }
      ]
    }
  }
}
```

```
}  
}  
}
```

表 2-74 响应参数说明

参数名	说明
name	节点名称。
host	节点对应的IP地址。
count	统计周期内，访问节点的请求数量。
access	统计周期内，访问节点的请求详情。参数说明请参见表 2-75。

表 2-75 access

参数名	说明
time	记录请求时间。
remote_address	请求对应的源IP地址和端口。
url	请求的原始URL
method	对应请求Path的方法。
content	对应请求的内容。

3. 执行命令删除访问日志。

- 删除集群所有节点的访问日志API
DELETE /_access_log
- 删除集群中某一节点的访问日志API
DELETE /_access_log/{nodeId}
“{nodeId}”为节点ID。

开启访问日志记录到文件

📖 说明

- 开启访问日志记录到文件后，客户端访问集群节点，会记录为{集群名_access_log.log}文件，通过日志备份功能可以在OBS详细查看访问日志。
- 该功能一般用于定位问题，问题解决后，建议关闭此开关。

1. 执行如下命令，开启访问日志记录到文件。

```
PUT /_cluster/settings  
{  
  "persistent": {  
    "flowcontrol.log.file.enabled": true  
  }  
}
```

表 2-76 访问日志记录到文件的配置项说明

参数	类型	说明
flowcontrol.log.file.enabled	Boolean	是否将每个请求的日志详情记录到后台日志文件。 取值范围： <ul style="list-style-type: none">truefalse (默认值)

2. 执行如下命令，关闭访问日志记录到文件。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.file.enabled": false
  }
}
```

2.7.5 配置 Elasticsearch 集群读写流量控制策略 1.0

场景描述

流量控制1.0提供节点级别的流量控制功能，可提供单个节点基于黑白名单的访问限制、HTTP并发连接数限制、HTTP最大连接数限制、基于请求Path的堆内存最大使用量流控能力、基于CPU最大占用率流控能力，一键断流能力，同时也提供节点访问IP统计和URI的采样统计能力。开启流控功能会使请求在入口处直接阻塞，可以缓解节点高并发场景下的集群压力，降低P99时延，减少节点不可用的风险。

表 2-77 流量控制策略

策略	描述	相关文档
HTTP/HTTPS流控	<ul style="list-style-type: none">• HTTP/HTTPS黑白名单设置IP和子网控制客户端IP访问，如果节点IP在黑名单中，则该客户端的连接将直接中断，节点不会处理任何请求。白名单规则优先于黑名单规则，如果客户端IP在黑白名单中都出现，客户端请求将不会被拒绝。• HTTP/HTTPS并发连接数流控通过限制节点每秒中的HTTP连接总数来限制节点流量。• HTTP/HTTPS新建连接数流控通过限制节点新建的连接数来限制节点流量。	开启HTTP/HTTPS节点流控
内存流控	基于节点堆内存使用量限制请求Path，支持设置内存流控白名单、全局内存使用阈值和基于单个Path设置堆内存使用阈值。全局内存流控阈值优先于单个Path内存阈值，白名单配置Path不参与内存流控。	开启内存流控

策略	描述	相关文档
Path全局免流 控白名单	可以根据客户需要，设置Path白名单全局免流控，当用户需要自定义插件时，可适当配置。	添加Path全局免流控白名单
请求采样统计	可以记录客户端IP的访问数量和采样用户的请求Path，用户可以基于统计值识别客户端IP的访问流量和分析哪些请求Path访问量	开启请求采样统计
一键断流	可以切断节点的所有访问流量，不包括 kibana访问和elasticsearch monitor类接口。	开启一键断流
流量控制	提供单独的流量统计查看接口，记录触发流控的数量，用户可以基于统计值评估流控配置阈值和衡量集群压力。	查看流量控制信息
访问日志	可以记录一段时间内节点接收的HTTP/HTTPS请求URL和Body，用户可以基于访问日志信息分析当前的流量压力。	开启并查看访问日志
访问日志记录	客户端访问集群节点，会记录为{集群名_access_log.log}访问日志文件，通过日志备份功能可以在OBS详细查看访问日志。	开启访问日志记录到文件
CPU流控	基于节点配置的最大CPU占用率来限制节点访问流量。	开启CPU流控

约束限制

- 2023年02月及之后创建的Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制2.0版本，之前创建Elasticsearch 7.6.2和Elasticsearch 7.10.2集群仅支持流量控制1.0版本。
- 开启流量控制功能会消耗部分节点性能。
- 开启流量控制会直接拒绝超过阈值的用户请求。
- 开启内存流控会消耗部分请求性能，开启内存流控会导致Kibana的部分search请求失败。
- 开启访问日志会影响集群性能。
- 内存流控和CPU流控都是基于请求Path的流控，Path长度和个数不应该配置过多，否则影响集群性能。

开启 HTTP/HTTPS 节点流控

1. 执行如下命令，开启HTTP/HTTPS节点流控。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.http.enabled": true,
    "flowcontrol.http.allow": ["192.168.0.1/24", "192.168.2.1/24"],
    "flowcontrol.http.deny": "192.168.1.1/24",
    "flowcontrol.http.concurrent": 1000,
    "flowcontrol.http.newconnect": 1000,
    "flowcontrol.http.warmup_period": 0
  }
}
```



```
}  
}
```

表 2-78 HTTP/HTTPS 流控的配置项说明

配置项	类型	说明
flowcontrol.http.enabled	Boolean	HTTP/HTTPS流控开关，默认关闭，开启会影响节点访问性能。 取值范围：true、false 默认值：false
flowcontrol.http.allow	List<String>	IP地址访问白名单。 支持配置多个IP地址和掩码，或者IP地址列表形式，中间用英文逗号隔开。例如“xx.xx.xx.xx/24,xx.xx.xx.xx/24”或“xx.xx.xx.xx,xx.xx.xx.xx”形式。 默认值为空。
flowcontrol.http.deny	List<String>	IP访问黑名单。 支持配置多个IP和掩码，或者IP列表形式，中间用英文逗号隔开。 默认值为空。
flowcontrol.http.concurrent	Integer	HTTP/HTTPS请求的并发连接数阈值。 默认值：节点可用核数x400
flowcontrol.http.newconnect	Integer	HTTP/HTTPS请求的每秒可以创建的新建连接数阈值。 默认值：节点可用核数x200
flowcontrol.http.warmup_period	Integer	HTTP/HTTPS新建连接数达到最大速率的需要的的时间，如果“flowcontrol.http.newconnect”配置为“100”且“flowcontrol.http.warmup_period”配置为“5000ms”，表示5s以后系统的新建连接数才可以达到每秒100。 取值范围：0~10000 单位：ms 默认值：0

说明

当所有参数指设置为“null”时，表示恢复配置默认值。

2. 执行如下命令，关闭HTTP/HTTPS节点流控。

```
PUT /_cluster/settings  
{  
  "persistent": {  
    "flowcontrol.http.enabled": false
```

```
}  
}
```

开启内存流控

📖 说明

Elasticsearch 5.5.1版本的集群开启内存流控会导致“_mget”请求被拦截，Kibana访问异常，可以通过把“_mget”请求加入请求白名单规避。

1. 开启内存流控。

- 执行如下命令，开启内存流控。

```
PUT /_cluster/settings  
{  
  "persistent": {  
    "flowcontrol.memory.enabled": true,  
    "flowcontrol.memory.allow_path": "/index/_search",  
    "flowcontrol.memory.heap_limit": "85%"  
  }  
}
```

- 执行如下命令，开启单个请求Path的内存流控。

当基于单个索引和请求Path设置堆内存使用阈值时，可以基于此规则做优先级调度。

```
PUT /_cluster/settings  
{  
  "persistent": {  
    "flowcontrol.memory.enabled": true,  
    "flowcontrol.memory": {  
      "flowcontrol_search": {  
        "filter_path": "index1/_search",  
        "heap_limit": "50%"  
      },  
      "flowcontrol_bulk": {  
        "filter_path": "index*/_bulk",  
        "heap_limit": "50%"  
      }  
    }  
  }  
}
```

表 2-79 内存流控的配置项说明

配置项	类型	说明
flowcontrol.memory.enabled	Boolean	内存流控开关，默认关闭，开启内存流控对节点性能有些许影响。 取值范围：true、false 默认值：false

配置项	类型	说明
flowcontrol.memory.allow_path	List<String>	<p>内存流控白名单Path。配置的路径不参与内存流控，可以支持通配符配置。集群控制的查询类接口默认放通，不参与内存流控，避免内存达到阈值，不能查询集群信息。</p> <p>例如：</p> <ul style="list-style-type: none"> • "flowcontrol.memory.allow_path": "/"index/_search", • "flowcontrol.memory.allow_path": "/"index*/_search", • "flowcontrol.memory.allow_path": ["/index/_search", "/index1/_bulk"], <p>支持最大配置10个Path，每个Path最大长度限制小于32。 默认值为空。</p>
flowcontrol.memory.heap_limit	String	<p>限制节点全局堆内存的最大使用率。不能低于堆内存的10%。</p> <p>取值范围：10%-100% 默认值：90%</p>

配置项	类型	说明
flowcontrol.memory.*.filter_path	String	<p>配置需要进行内存流控的访问Path，控制单个请求Path的堆内存使用阈值。</p> <p>默认值 ""，表示匹配所有的路径。如果只配置了单路径 “flowcontrol.memory.heap_limit”，没有配置“flowcontrol.memory.*.filter_path”，表示除去白名单外的所有path都影响。白名单规则优先于单路径规则，如果一个路径同时配置了“flowcontrol.memory.allow_path”和“flowcontrol.memory.*.filter_path”，此请求路径会被允许。</p> <p>例如同时配置了“flowcontrol.memory.allow_path”和“flowcontrol.memory.*.filter_path”，其中flowcontrol.memory.*.filter_path="abc/_search"且flowcontrol.memory.allow_path="abc/_search"，此种情况abc/_search将不被流控。</p> <p>最大长度：32</p>
flowcontrol.memory.*.heap_limit	String	<p>配置请求Path的堆内存阈值，堆内存超过阈值触发流控。</p> <p>取值范围：0%-100%</p> <p>默认值：90%</p>

📖 说明

当所有参数指设置为“null”时，表示恢复配置默认值。

2. 执行如下命令，关闭集群内存流控。

- 执行如下命令，删除单个请求Path的内存流控。

关闭内存流控时，要先删除单个请求Path的内存流控。

```
PUT /_cluster/settings
{
```

```
"persistent": {  
  "flowcontrol.memory.enabled": true,  
  "flowcontrol.memory": {  
    "flowcontrol_search": {  
      "filter_path": null,  
      "heap_limit": null  
    }  
  }  
}
```

- 执行如下命令，关闭内存流控。

```
PUT /_cluster/settings  
{  
  "persistent": {  
    "flowcontrol.memory.enabled": false  
  }  
}
```

添加 Path 全局免流控白名单

执行如下命令，添加Path全局免流控白名单。

```
PUT /_cluster/settings  
{  
  "persistent": {  
    "flowcontrol.path.white_list": "xxxx"  
  }  
}
```

表 2-80 Path 全局免流控白名单的配置项说明

配置项	类型	说明
flowcontrol.path.white_list	List<String>	<p>Path全局免流控白名单，配置的路径不参与内存、CPU流控和一键断流，IP流控除外。</p> <p>支持最大配置10个Path，每个Path最大长度限制小于32。</p> <p>默认值为空。</p> <p>说明 一般不建议配置此值，仅在自定义插件时根据业务需求配置。</p>

📖 说明

当所有参数指设置为“null”时，表示恢复配置默认值。

开启请求采样统计

1. 执行如下命令，开启请求采样统计。

```
PUT /_cluster/settings  
{  
  "persistent": {  
    "flowcontrol.statics.enabled": true,  
    "flowcontrol.statics.threshold": 100,  
    "flowcontrol.statics.sample_frequency": 50  
  }  
}
```

表 2-81 请求采样统计的配置项说明

配置项	类型	说明
flowcontrol.statics.enabled	Boolean	请求采样统计开关。开启请求采样统计对节点性能会有影响。 取值范围：true、false 默认值：false
flowcontrol.statics.threshold	Integer	统计最近时间访问的请求数量。配置为100，表示会统计出最近访问最多的100个IP地址和基于采样统计的访问最多的100个URL。 最小值：10 最大值：1000 默认值：100 说明 <ul style="list-style-type: none">IP统计和URL采样统计基于访问时间缓存策略，节点会记录最近访问的IP和请求URL，如果缓存空间达到设置的阈值（即“flowcontrol.statics.threshold”配置值），访问时间距离现在最久的记录将被清除掉。URL采样统计当前基于URL hash值确认访问Path的一致性。
flowcontrol.statics.sample_frequency	Integer	Path采样频率。配置为100，表示每100个请求采样统计一次。 最小值：50 默认值：100

📖 说明

当所有参数指设置为“null”时，表示恢复配置默认值。

2. 执行如下命令，关闭请求采样统计。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.statics.enabled": false
  }
}
```

开启一键断流

1. 执行如下命令，开启一键断流。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.break.enabled": true
  }
}
```

2. 执行如下命令，关闭一键断流。

```
PUT /_cluster/settings
{
```

```
"persistent": {
  "flowcontrol.break.enabled": false
}
```

查看流量控制信息

- 查看所有节点的流量控制情况

```
GET /_nodes/stats/filter
```

- 查看某个具体节点的流量控制情况

```
GET /_nodes/{nodeId}/stats/filter
```

“{nodeId}”为需要查看流量控制的节点ID。

响应示例：

📖 说明

返回值以Node级别分开，http记录并发和新建连接数据统计，memory记录内存流控统计，ip_address记录最近最多访问的客户端IP，url_sample记录采样的最近最多请求URL。cpu记录CPU流控统计。

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "css-flowcontroller",
  "nodes": {
    "EIBRNCMbTj6L1C-Wke-Dnw": {
      "name": "css-flowcontroller-ess-esn-1-1",
      "host": "10.0.0.133",
      "timestamp": 1613979513747,
      "flow_control": {
        "transport": {
          "concurrent_req": 0,
          "rejected_concurrent": 0,
          "rejected_new": 0,
          "rejected_deny": 0
        },
        "http": {
          "concurrent_req": 0,
          "rejected_concurrent": 0,
          "rejected_new": 0,
          "rejected_deny": 0
        },
        "memory": {
          "memory_allow": 41,
          "memory_rejected": 0
        },
        "cpu": {
          "rejected_cpu": 0
        }
      },
      "ip_address": [
        {
          "ip": "/10.0.0.198",
          "count": 453
        },
        {
          "ip": "/198.19.49.1",
          "count": 42
        }
      ],
      "url_sample": [
        {
          "url": "/*/_search?pretty=true",
          "method": "GET",

```

```

    "remote_address" : "/10.0.0.198:16763",
    "count" : 1
  }
]
}
}
}

```

表 2-82 响应参数说明

参数名	说明
concurrent_req	节点实际的TCP连接数据信息，没有开启流控这个配置也会记录，参考GET /_nodes/stats/http接口current_open值，但是会比这个值小，这里忽略了白名单IP和内部节点IP。
rejected_concurrent	HTTP流控开启生效，关闭后不清零，开启流控期间拒绝的并发连接数。
rejected_new	HTTP流控开启生效，关闭后不清零，开启流控期间拒绝的新建连接数。
rejected_deny	HTTP流控开启生效，关闭后不清零，配置黑名单拒绝的请求数。
memory_allow	内存流控开启生效，关闭后不清零，内存流控允许的请求数，触发内存流控后允许的请求数量，allow_path白名单中通过请求不会被记录，如果allow_path配置为“**”，所有请求都不会被记录。
memory_rejected	内存流控开启生效，关闭后不清零，内存流控拒绝的请求数，触发内存流控后拒绝的请求数量，allow_path白名单中通过请求不会被记录，如果allow_path配置为“**”，所有请求都不会被记录。
rejected_cpu	CPU流控开启生效，关闭后不清零，超过CPU流控阈值拒绝的请求数。
ip_address	IP地址统计，基于配置值统计节点访问的IP地址和请求数量。参数说明请参见表2-83表2-83。
url_sample	请求Path采样统计，基于配置时间和采样间隔统计相同请求URL数量。参数说明请参见表2-84。

表 2-83 ip_address

参数名	说明
ip	访问节点的源IP地址。
method	对应IP地址的访问次数统计。

表 2-84 url_sample

参数名	说明
url	请求的采样统计，记录访问节点的请求URL。
method	对应请求Path的方法。
remote_address	请求对应的源IP地址和端口。
count	对应请求Path的采样统计次数。

开启并查看访问日志

1. 执行如下命令，开启访问日志。

- 开启集群所有节点的访问日志
`PUT /_access_log?duration_limit=30s&capacity_limit=1mb`
- 开启集群中某一节点的访问日志
`PUT /_access_log/{nodeId}?duration_limit=30s&capacity_limit=1mb`
“{nodeId}”为节点ID。

表 2-85 访问日志的配置项说明

配置项	类型	说明
duration_limit	String	访问日志记录时间。 取值范围：10~120 单位：s 默认值：30
capacity_limit	String	访问日志记录大小。统计开启访问日志后记录的请求大小，当统计的大小大于该配置值，访问日志记录终止。 取值范围：1~5 单位：MB 默认值：1

📖 说明

- “duration_limit”和“capacity_limit”只要有一个参数达到阈值，访问日志记录就会停止。
 - 当所有参数指设置为“null”时，表示恢复配置默认值。
- ### 2. 执行如下命令，查看访问日志。
- 查看集群所有节点的访问日志API
`GET /_access_log`
 - 查看集群中某一节点的访问日志API
`GET /_access_log/{nodeId}`
“{nodeId}”为节点ID。

响应示例：

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "css-flowcontroller",
  "nodes" : {
    "8x-ZHu-wTemBQwpcGivFKg" : {
      "name" : "css-flowcontroller-ess-esn-1-1",
      "host" : "10.0.0.98",
      "count" : 2,
      "access" : [
        {
          "time" : "2021-02-23 02:09:50",
          "remote_address" : "/10.0.0.98:28191",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        },
        {
          "time" : "2021-02-23 02:09:52",
          "remote_address" : "/10.0.0.98:28193",
          "url" : "/_access/security/log?pretty",
          "method" : "GET",
          "content" : ""
        }
      ]
    }
  }
}
```

表 2-86 响应参数说明

参数名	说明
name	节点名称。
host	节点对应的IP地址。
count	统计周期内，访问节点的请求数量。
access	统计周期内，访问节点的请求详情。参数说明请参见 表 2-87 。

表 2-87 access

参数名	说明
time	记录请求时间。
remote_address	请求对应的源IP地址和端口。
url	请求的原始URL
method	对应请求Path的方法。
content	对应请求的内容。

3. 执行命令删除访问日志。
 - 删除集群所有节点的访问日志API
DELETE /_access_log
 - 删除集群中某一节点的访问日志API
DELETE /_access_log/{nodeId}
“{nodeId}”为节点ID。

开启访问日志记录到文件

📖 说明

- 开启访问日志记录到文件后，客户端访问集群节点，会记录为{集群名_access_log.log}文件，通过日志备份功能可以在OBS详细查看访问日志。
- 该功能一般用于定位问题，问题解决后，建议关闭此开关。

1. 执行如下命令，开启访问日志记录到文件。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.file.enabled": true
  }
}
```

表 2-88 访问日志记录到文件的配置项说明

参数	类型	说明
flowcontrol.log.file.enabled	Boolean	是否将每个请求的日志详情记录到后台日志文件。 取值范围： <ul style="list-style-type: none">• true• false（默认值）

2. 执行如下命令，关闭访问日志记录到文件。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.log.file.enabled": false
  }
}
```

开启 CPU 流控

1. 执行如下命令，开启CPU流控。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.cpu.enabled": true,
    "flowcontrol.cpu.percent_limit": 80,
    "flowcontrol.cpu.allow_path": ["index/_search"]
  }
}
```

表 2-89 访问日志的配置项说明

配置项	类型	说明
flowcontrol.cpu.enabled	Boolean	是否开启CPU流控。 取值范围： <ul style="list-style-type: none">• true• false（默认值）
flowcontrol.cpu.percent_limit	Integer	流控的阈值，在CPU使用率超过此阈值时触发流控。 默认值：90
flowcontrol.cpu.allow_path	List<String>	CPU流控白名单Path。 配置的路径不参与CPU流控，支持通配符配置。集群控制的查询类接口默认放通，不参与CPU流控，避免CPU达到阈值导致不能查询集群信息。 例如： <ul style="list-style-type: none">• "flowcontrol.memory.allow_path": "/index/_search",• "flowcontrol.memory.allow_path": "/index*/_search",• "flowcontrol.memory.allow_path": ["/index/_search", "/index1/_bulk"], 支持最大配置10个Path，每个Path最大长度限制小于32。 默认值为空。

📖 说明

当所有参数指设置为“null”时，表示恢复配置默认值。

2. 执行如下命令，关闭CPU流控。

```
PUT /_cluster/settings
{
  "persistent": {
    "flowcontrol.cpu.enabled": false
  }
}
```

2.7.6 配置 Elasticsearch 集群大查询隔离

场景描述

大查询隔离支持对查询请求进行独立管理，将高内存、长耗时的查询请求进行隔离，保证节点内存安全。在节点堆内存使用率过高时，触发中断控制程序，根据选择的中断策略将其中一条大查询请求进行中断，取消其正在运行的查询任务。大查询隔离同时支持全局查询超时配置，用户可实时配置所有查询请求的超时时间，中断超时查询请求。

大查询隔离可以有效解决以下问题，提升集群的搜索能力。

- 部分查询占用了很高的节点堆内存，导致机器频繁Garbage Collect，甚至引发OOM异常。
- 频繁Garbage Collect导致节点脱离，查询迟迟无法响应甚至失败。
- 查询量过大导致CPU爆满，线上业务受到影响。

约束限制

仅Elasticsearch 7.6.2、Elasticsearch 7.10.2集群支持大查询隔离。

操作步骤

大查询隔离特性和全局超时特性默认关闭，用户可根据需要实时配置，配置后立即生效。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，选择待配置大查询隔离的集群，单击操作列“Kibana”，登录Kibana界面。
3. 在Kibana的左侧导航中选择“Dev Tools”，执行如下命令开启大查询隔离和全局查询超时的特性开关。

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.enabled": true,
    "search.isolator.time.enabled": true
  }
}
```

以上两个开关是独立功能，当开关打开后，支持配置的参数请参见[表2-90](#)。

表 2-90 大查询隔离和全局超时的参数配置

特性开关	配置参数	参数说明
search.isolator.enabled	search.isolator.memory.task.limit search.isolator.time.management	单个分片查询任务被定义为大查询任务的阈值。
	search.isolator.memory.pool.limit search.isolator.memory.heap.limit search.isolator.count.limit	触发隔离池内查询任务中断的阈值。 说明 参数“search.isolator.memory.heap.limit”定义了节点实际堆内存的使用限制，包括写入和查询等操作，超过限制时将选取隔离池内的大查询任务进行中断。
	search.isolator.strategy search.isolator.strategy.ratio	中断隔离池中某一条查询任务的选取策略。
search.isolator.time.enabled	search.isolator.time.limit	全局查询任务超时设置。

4. 配置大查询隔离。

- 执行如下命令，设置单个分片查询任务被定义为大查询任务的阈值。

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.memory.task.limit": "50MB",
    "search.isolator.time.management": "10s"
  }
}
```

表 2-91 参数说明

参数名	数据类型	说明
search.isolator.memory.task.limit	String	查询任务用于聚合等操作向节点申请的大内存，申请内存超过此阈值将进行隔离观察。 <ul style="list-style-type: none"> 取值范围：0b~节点最大堆内存 默认值：50MB 说明 可以通过如下命令查询集群堆内存使用情况和最大值。 GET _cat/nodes?&h=id,ip,port,r,ramPercent,ramCurrent,heapMax,heapCurrent
search.isolator.time.management	String	查询任务创建至今的时长（即开始使用集群资源进行查询），超过阈值将被隔离观察。 <ul style="list-style-type: none"> 取值范围：≥ 0ms 默认值：10s

- 执行如下命令，触发隔离池内查询任务中断的阈值。

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.memory.pool.limit": "50%",
    "search.isolator.memory.heap.limit": "90%",
    "search.isolator.count.limit": 1000
  }
}
```

表 2-92 参数说明

参数名	数据类型	说明
search.isolator.memory.pool.limit	String	当前节点最大堆内存百分比，当隔离池所有大查询任务申请的内存超过此阈值将触发中断控制程序，取消执行隔离池其中一条大查询任务。 <ul style="list-style-type: none"> 取值范围：0.0~100.0% 默认值：50%

参数名	数据类型	说明
search.isolator.memory.heap.limit	String	当前节点堆内存的实际使用阈值，当节点堆内存使用超过阈值百分比时触发中断控制程序，取消执行隔离池其中一条大查询任务。 <ul style="list-style-type: none">取值范围：0.0~100.0%默认值：90%
search.isolator.count.limit	Integer	当前节点隔离池的大查询任务数阈值，被观察的查询任务数超过此阈值将触发中断控制程序，不再接受新的大查询。如果继续触发大查询请求，则直接取消该请求。 <ul style="list-style-type: none">取值范围：10~50000默认值：1000

📖 说明

根据业务设置 “search.isolator.memory.pool.limit”，“search.isolator.count.limit” 参数时，可结合 “search.isolator.memory.task.limit”，“search.isolator.time.management” 两个参数控制查询任务进入到隔离池的数量。

- 执行如下命令，中断隔离池中某一条查询任务的选取策略。

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.strategy": "fair",
    "search.isolator.strategy.ratio": "0.5%"
  }
}
```

参数名	数据类型	说明
search.isolator.strategy	String	<p>触发中断控制程序时大查询选取的策略。根据策略选取一条查询进行中断。</p> <p>说明 大查询隔离池每秒检查一次，直至堆内存下降到安全范围。</p> <p>取值范围：fair、mem-first、time-first</p> <ul style="list-style-type: none"> mem-first策略是指触发中断时，选取隔离池中堆内存使用最大的一条查询任务进行中断。 time-first策略是指触发中断时，选取隔离池中已运行时间最长的一条查询任务进行中断。 fair策略是综合考虑内存和时间两种因素，如果分片查询的堆内存申请大小相差不超过“最大堆内存乘 search.isolator.strategy.ratio”的大小，则认为时间较长的查询更应该中断。否则认为堆内存使用较大的查询更应该中断。 <p>默认值：fair</p>
search.isolator.strategy.ratio	String	<p>fair策略的阈值，仅当“search.isolator.strategy”值为“fair”生效。综合考虑大查询的运行时间及内存，当大查询任务内存相差不超过此阈值时，考虑选取运行时间长的大查询进行中断。当查询内存相差超过此阈值时，选取大内存查询任务进行中断。</p> <ul style="list-style-type: none"> 取值范围：0.0-100.0% 默认值：1%

5. 配置全局查询超时。

执行如下命令，设置全局查询任务超时时间。

```
PUT _cluster/settings
{
  "persistent": {
    "search.isolator.time.limit": "120s"
  }
}
```

参数名	数据类型	说明
search.isolator.time.limit	String	<p>当全局查询超时功能开启时，所有已创建的查询任务超过此时长将被取消执行。</p> <ul style="list-style-type: none"> 取值范围：≥ 0ms 默认值：120s

2.7.7 配置 Elasticsearch 集群聚合增强

场景描述

聚合增强在数据聚簇的情况下，利用向量化技术，批量处理数据，从而提升聚合性能，优化可观测性业务的聚合分析能力。

在大规模数据的集聚合分析场景下，耗时主要集中在对数据的分组聚合。

提升分组聚合能力依赖排序键和聚簇键。

- 排序键：数据按照排序键顺序存储。
- 聚簇键：是排序键的前缀子集，数据按照聚簇键聚簇在一起。

聚合增强常见的使用场景请参见[表2-93](#)。

表 2-93 聚合增强的使用场景

场景	描述	相关文档
低基字段聚合	表示对具有较少不同值的列进行聚合，如对一个存储类别的列进行聚合。	低基字段分组聚合
高基字段聚合	表示对具有较多不同值的列进行聚合，如对一个存储时间的列进行按天的时间聚合。	高基字段直方图聚合
低基字段和高基字段混合聚合	表示对以上两种类型的聚合混合在一起，如先对低基字段做分组聚合，然后再对高基字段做直方图聚合。	低基字段分组聚合混合高基字段直方图聚合

约束限制

仅Elasticsearch 7.10.2集群支持聚合增强。

低基字段分组聚合

低基字段一般采用分组聚合，在排序的情况下具备较好的数据聚簇性，利于向量化批量处理数据。

例如，查询语句如下所示。

```
POST testindex/_search
{
  "size": 0,
  "aggs": {
    "groupby_region": {
      "terms": {
        "field": "region"
      },
      "aggs": {
        "groupby_host": {
          "terms": {
            "field": "host"
          },
          "aggs": {
            "avg_cpu_usage": {
```

```
      "avg": {
        "field": "cpu_usage"
      }
    }
  }
}
}
```

当需要对“region”和“host”这两个低基字段进行聚合增强，则需要对索引进行如下设置。

```
// 索引设置
"settings": {
  "index": {
    "search": {
      "turbo": {
        "enabled": "true" // 开启优化
      }
    },
    "sort": { // 指定排序键
      "field": [
        "region",
        "host",
        "other"
      ]
    },
    "cluster": {
      "field": [ // 指定聚簇键
        "region",
        "host"
      ]
    }
  }
}
```

📖 说明

聚簇键必须是排序键的前缀子集。

高基字段直方图聚合

高基字段一般采用直方图分组聚合，利于处理某个区间内的数据。

例如，查询语句如下所示。这条查询是对时间字段timestamp做直方图分组，然后求score的平均值。

```
POST testindex/_search?pretty
{
  "size": 0,
  "aggs": {
    "avg_score": {
      "avg": {
        "field": "score"
      },
    },
    "aggs": {
      "groupbytime": {
        "date_histogram": {
          "field": "timestamp",
          "calendar_interval": "day"
        }
      }
    }
  }
}
```

当需要对典型的高基字段“timestamp”进行聚合增强时，则需要对索引进行如下设置。

```
// 索引设置
"settings": {
  "index": {
    "search": {
      "turbo": {
        "enabled": "true" // 开启优化
      }
    },
    "sort": { // 指定排序键
      "field": [
        "timestamp"
      ]
    }
  }
}
```

低基字段分组聚合混合高基字段直方图聚合

在低基字段和高基字段混合的场景下，一般先对低基字段做分组聚合，然后再对高基字段做直方图聚合。

假如，查询语句如下所示。

```
POST testindex/_search
{
  "size": 0,
  "aggs": {
    "groupby_region": {
      "terms": {
        "field": "region"
      },
      "aggs": {
        "groupby_host": {
          "terms": {
            "field": "host"
          },
          "aggs": {
            "groupby_timestamp": {
              "date_histogram": {
                "field": "timestamp",
                "interval": "day"
              },
              "aggs": {
                "avg_score": {
                  "avg": {
                    "field": "score"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

当需要先对“region”低基字段进行分组，再对“host”低基字段进行分组，最后对“timestamp”高基字段进行直方图分组时，聚合增强的索引设置如下。

```
// 索引设置
"settings": {
  "index": {
    "search": {
      "turbo": {
```

```
    "enabled": "true" // 开启优化
  }
},
"sort": { // 指定排序键
  "field": [
    "region",
    "host",
    "timestamp",
    "other"
  ]
},
"cluster": {
  "field": [ // 指定聚簇键
    "region",
    "host"
  ]
}
}
```

说明

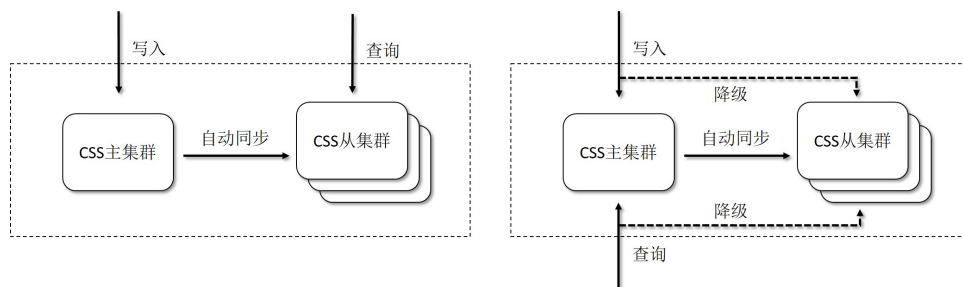
- 聚簇键必须是排序键的前缀子集。
- 高基字段必须在排序键中，且高基字段必须在最后一个低基字段后面。

2.7.8 配置 Elasticsearch 集群读写分离

场景描述

读写分离支持将写入主集群（Leader）的数据自动同步到从集群（Follower），实现主集群承担写入任务，从集群承担查询任务。读写分离可以将读写压力分开，提高查询的性能，还能支持在主集群无法提供服务时降级，使用从集群提供写入和查询服务。

图 2-13 读写分离特性的两种使用场景



场景一（图左）：主集群写入，从集群查询，分离读写压力。

场景二（图右）：主集群故障时，从集群升级，正常提供服务，避免业务中断。

约束限制

- 仅Elasticsearch 7.6.2、Elasticsearch 7.10.2集群支持读写分离。
- 主集群和从集群的版本必须保持一致，否则服务可能异常。

前提条件

已创建两个版本一致的集群，一个作为主集群，另一个作为从集群。从集群必须能访问主集群的REST接口（默认端口为9200）。

连接主从集群

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择从集群，单击操作列“Kibana”，登录Kibana页面。
3. 单击左侧导航栏的“Dev Tools”进入操作页面
4. 执行如下命令，在从集群配置主集群的信息。

```
PUT /_cluster/settings
{
  "persistent" : {
    "cluster" : {
      "remote.rest" : {
        "leader1" : {
          "seeds" : [
            "http://10.0.0.1:9200",
            "http://10.0.0.2:9200",
            "http://10.0.0.3:9200"
          ],
          "username": "elastic",
          "password": "*****"
        }
      }
    }
  }
}
```

表 2-94 请求体参数说明

参数名	说明
<i>leader1</i>	主集群配置任务的名称，由用户自定义，在后续读写分离配置时会用到该名称。
seeds	主集群的访问地址。当集群开启HTTPS访问时，URI schema需要对应使用https。
username	主集群的用户名，仅主集群是安全集群时才需要配置。
password	主集群的密码，仅主集群是安全集群时才需要配置。

返回示例：

```
{
  "acknowledged" : true, // 是否成功
  "persistent" : {
    "cluster" : {
      "remote" : {
        "rest" : {
          "leader1" : {
            "seeds" : [
              "http://10.0.0.1:9200",
              "http://10.0.0.2:9200",
              "http://10.0.0.3:9200"
            ]
          }
        }
      }
    }
  },
  "transient" : { }
}
```

- 配置完成后，在从集群执行如下命令确认从集群和主集群的连接状况。

```
GET _remote/rest/info
```

返回示例：

```
{
  "leader1": {
    "connected": true //表示连接正常
  }
}
```

索引同步

索引同步有2种方式：指定索引同步和匹配索引同步。

开启索引同步后，从集群的索引会变为只读状态，并且定时与主集群索引同步。同步周期默认30秒，如果需要修改请参见[修改同步周期](#)。

指定索引同步

- 在从集群执行如下命令，将主集群的单个索引同步到从集群，不修改索引配置。

```
PUT start_remote_sync
{
  "remote_cluster": "leader1",
  "remote_index": "data1_leader",
  "local_index": "data1_follower"
}
```

- 在从集群执行如下命令，将主集群的单个索引同步到从集群，并修改部分索引配置。

```
PUT start_remote_sync
{
  "remote_cluster": "leader1",
  "remote_index": "data1_leader",
  "local_index": "data1_follower",
  "settings": {
    "number_of_replicas": 4
  }
}
```

📖 说明

以下索引配置不支持修改：“number_of_shards”、“version.created”、“uuid”、“creation_date”、“soft_deletes.enabled”。

表 2-95 请求体参数说明

参数名	说明
remote_cluster	主集群配置任务的名称，在 连接主从集群 时设置，例如本文设置的是“leader1”。
remote_index	主集群待同步索引名称。
local_index	同步到从集群的索引名称。
settings	同步索引的索引设置。

匹配索引同步

- 在从集群执行如下命令，创建匹配模式同步索引，将主集群中匹配的索引同步到从集群。

```
PUT auto_sync/pattern/${PATTERN}
{
  "remote_cluster": "leader1",
  "remote_index_patterns": "log*",
  "local_index_pattern": "{{remote_index}}-sync",
  "apply_exist_index": true
}
```

- 在从集群执行如下命令，创建匹配模式同步索引，将主集群中匹配的索引同步到从集群，并修改部分索引配置。

```
PUT auto_sync/pattern/${PATTERN}
{
  "remote_cluster": "leader1",
  "remote_index_patterns": "log*",
  "local_index_pattern": "{{remote_index}}-sync",
  "apply_exist_index": true,
  "settings": {
    "number_of_replicas": 4
  }
}
```

📖 说明

以下索引配置不支持修改：“number_of_shards”、“version.created”、“uuid”、“creation_date”、“soft_deletes.enabled”。

表 2-96 请求体参数说明

参数名	说明
PATTERN	同步集群配置的pattern名字，用于区分多个不同的匹配模式。
remote_cluster	主集群配置任务的名称，通过 连接主从集群 设置，具体值如上文的“leader1”。
remote_index_patterns	主集群待同步索引模式，支持通配符“*”。
local_index_pattern	同步到从集群的索引模式，支持模板替换。例如取值为“{{remote_index}}-sync”时，待同步索引为“log1”，同步过来的索引为“log1-sync”。
apply_exist_index	是否同步主集群上已存在的索引，默认为“true”。
settings	同步索引的索引设置。

停止索引同步

在从集群执行如下命令，终止索引名称对应的同步任务，后续主集群对该索引的修改不会被同步到从集群，并且从集群中该索引的只读状态解除，支持新数据的写入。

```
PUT log*/stop_remote_sync
```

“log*”表示索引名称，支持指定多个索引（多个索引之间用英文逗号分隔）和使用通配符，此例是停止以“log”开头的索引的同步任务。

查询并删除集群匹配模式

1. 在从集群执行如下命令，查询已创建的匹配模式。

- 列表查询
GET auto_sync/pattern
- 指定名称查询
GET auto_sync/pattern/{PATTERN}

接口返回示例如下：

```
{
  "patterns" : [
    {
      "name" : "pattern1",
      "pattern" : {
        "remote_cluster" : "leader",
        "remote_index_patterns" : [
          "log*"
        ],
        "local_index_pattern" : "{{remote_index}}-sync",
        "settings" : { }
      }
    }
  ]
}
```

2. 在从集群执行如下命令，指定模式名称删除模式。

```
DELETE auto_sync/pattern/{PATTERN}
```

开启强制同步

- **开启强制同步**

插件默认根据主集群索引的文档数是否发生变化来决定是否要同步元数据，当主集群只对文档进行更新且文档数不变时，插件不会同步该更新到从集群。支持修改配置，开启后，每个同步周期从集群都会强制同步主机群的索引元数据。

请求示例如下（开启强制同步）：

```
PUT _cluster/settings
{
  "persistent": {
    "remote_sync.force_synchronize": true
  }
}
```

修改同步周期

- **修改同步周期**

同步周期默认30秒，支持修改。

请求示例如下（修改同步周期为2秒）：

```
PUT {index_name}/_settings
{
  "index.remote_sync.sync_interval": "2s"
}
```

查询同步状态

- **获取自动同步状态**

该接口可以获取匹配索引同步的状态。

请求示例如下：

```
GET auto_sync/stats
```

接口返回示例如下：


```
{
  "success_count": 3,
  "failed_count": 0,
  "failed_remote_cluster_state_requests_count": 0,
  "last_fail_exception": { },
  "last_fail_remote_cluster_requests_exception": { }
}
```

- **获取指定索引的自动同步状态**

请求示例如下：

```
GET {index_name}/sync_stats
```

接口返回示例如下：

```
{
  "indices": {
    "data1_follower": {
      "shards": {
        "0": [
          {
            "primary": false,
            "total_synced_times": 27,
            "total_empty_times": 25,
            "total_synced_files": 4,
            "total_synced_bytes": 3580,
            "total_paused_nanos": 0,
            "total_paused_times": 0,
            "current": {
              "files_count": 0,
              "finished_files_count": 0,
              "bytes": 0,
              "finished_bytes": 0
            }
          }
        ],
      },
    },
    {
      "primary": true,
      "total_synced_times": 28,
      "total_empty_times": 26,
      "total_synced_files": 20,
      "total_synced_bytes": 17547,
      "total_paused_nanos": 0,
      "total_paused_times": 0,
      "current": {
        "files_count": 0,
        "finished_files_count": 0,
        "bytes": 0,
        "finished_bytes": 0
      }
    }
  ]
}
}
```

主从切换

当主集群出现故障时，支持通过主从切换保证业务正常进行。

1. 判断主从集群的索引同步方式。查询从集群是否有索引同步的匹配模式，命令参考[查询并删除集群匹配模式](#)。
 - 如果没有匹配模式，则表示主从集群是指定索引同步关系，则执行**3**。
 - 如果有匹配模式，则表示主从集群是模式匹配索引同步关系，则执行**2**。
2. 删除从集群的索引同步的匹配模式，命令参考[查询并删除集群匹配模式](#)。
3. 从集群**停止索引同步**，将读写流量切换到从集群。如果主从集群是匹配索引同步关系，停止索引同步命令使用通配符匹配索引。

4. 当主集群恢复后，通过[索引同步](#)将从集群数据向主集群同步，完成主从切换。

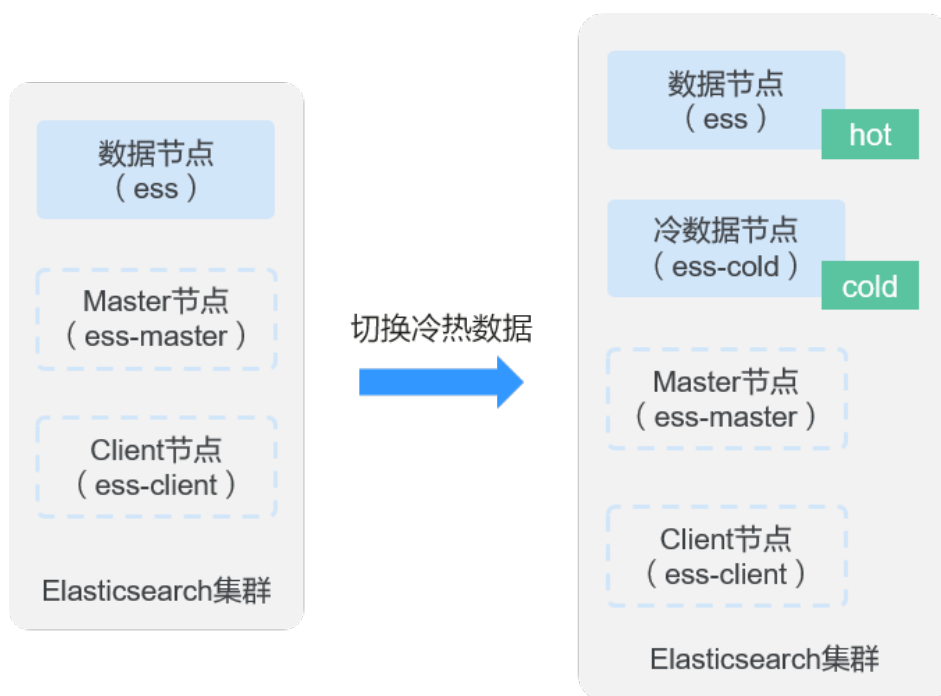
2.7.9 切换 Elasticsearch 集群冷热数据

云搜索服务支持给Elasticsearch集群配置冷数据节点，支持对索引数据进行冷热切换。

场景描述

通过切换冷热数据，可以将部分现查要求秒级返回的数据放在高性能机器上面作为热数据，对于历史数据要求分钟级别返回的数据放在大容量低规格节点作为冷数据。冷热数据切换可以减低存储成本，提升搜索效率。

图 2-14 切换冷热数据原理



在创建集群时，启用的冷数据节点会被打上“cold”标签，用来存放冷数据，数据节点会被打上“hot”标签，用于存放热数据。用户可以通过配置指定索引，将数据分配到冷数据节点上。只有集群存在冷数据节点时，数据节点才会被打上热标签。

冷数据节点只有在创建集群时可以选择启用，已建集群不支持添加冷数据节点，如果集群没有冷数据节点又想切换冷热数据，可以使用存算分离功能，具体操作请参见[配置Elasticsearch集群存算分离](#)。

冷数据节点支持节点数量扩缩容和磁盘存储容量的扩容，具体操作请参见[扩缩容Elasticsearch集群](#)。

约束限制

仅启用了冷数据节点的集群才支持进行冷热数据切换。只有在创建集群时可以选择启用冷数据节点，已建集群不支持添加冷数据节点。

操作步骤

1. 登录云搜索服务管理控制台。
2. 确认集群是否已启用冷数据节点。
在“集群管理”页面，选择需要进行存算分离的集群，单击集群名称进入集群详情页，查看“节点信息”中是否有冷数据节点的信息。

图 2-15 查看冷数据节点

节点信息	
节点规格	ess.s 8 GB
节点存储	40 G
节点数量	1
冷数据节点规格	ess.s 8 GB
冷数据节点存储	40 G
冷数据节点数	3

- 是，集群存在冷数据节点，则执行下一步。
 - 否，集群不存在冷数据节点，则不支持切换集群的冷热数据。如果想将历史数据定义为冷数据，减低存储成本，提升搜索效率，可以使用存算分离功能，具体操作请参见[配置Elasticsearch集群存算分离](#)。
3. 单击集群操作列“Kibana”，登录Kibana页面。
 4. 选择左侧导航栏的“Dev Tools”，进入操作页面。
 5. 在Kibana操作页面设置template，通过设置集群模板将相应的索引数据存储到冷热数据节点上。

例如，执行如下命令设置template将以“myindex”开头的索引存储到冷数据节点上。

```
- Elasticsearch版本<6.x的创建模板命令
PUT _template/test
{
  "order": 1,
  "template": "myindex*",
  "settings": {
    "index": {
      "refresh_interval": "30s",
      "number_of_shards": "3",
      "number_of_replicas": "1",
      "routing.allocation.require.box_type": "cold"
    }
  }
}
```

- Elasticsearch版本≥6.x的创建模板命令

```
PUT _template/test
{
  "order": 1,
  "index_patterns": "myindex*",
  "settings": {
    "refresh_interval": "30s",
    "number_of_shards": "3",
    "number_of_replicas": "1",
    "routing.allocation.require.box_type": "cold"
  }
}
```

也支持直接将已有索引存储到冷热数据节点上。

例如，执行如下命令将“myindex”索引存储到冷数据节点上。

```
PUT myindex/_settings
{
  "index.routing.allocation.require.box_type": "cold"
}
```

“myindex”表示索引名称。“cold”可以基于需要改成“hot”。

6. 当需要回退冷热数据时，可以执行如下命令，删除冷热数据配置。这时候索引数据会被当成普通索引，随机均衡分布在冷热数据节点中。

```
PUT myindex/_settings
{
  "index.routing.allocation.require.box_type": null
}
```

“myindex”表示索引名称。

2.7.10 配置 Elasticsearch 集群索引回收站

Elasticsearch集群在执行删除索引的操作时，会直接将索引删除，没有回收站的能力。为了防止客户误操作导致数据被删除，CSS提供了索引回收站功能，支持将删除的索引存放到回收站中，且支持从回收站中还原索引，进而提升集群的数据可靠性。

约束限制

- 仅Elasticsearch 7.10.2集群支持索引回收站功能。
- 当开启回收站之后，使用DELETE API将索引放入回收站时，集群的状态可能会出现短暂的RED，这是由于索引存放至回收站时会将索引CLOSE，在Elasticsearch中CLOSE索引时会出现短暂的RED状态。
- 当使用还原API从回收站中还原索引时，集群的状态可能会出现短暂的RED，这是由于还原索引时需要重新OPEN索引，在Elasticsearch中OPEN索引时会重新初始化分片，导致短暂的RED状态。
- 存放在回收站中的索引仍然是集群元数据的一部分，因此集群无法新建和回收站中的同名的索引。

开启索引回收站

执行如下命令，启用索引回收站功能。

```
PUT _cluster/settings
{
  "persistent": {
    "index.trash.enabled": true
  }
}
```

表 2-97 配置项说明

配置项	说明
“index.trash.enabled”	索引回收站开关，缺省值是“false”，表示不启用索引回收站功能，集群与开源Elasticsearch集群的能力兼容。当设置成“true”时，表示开启索引回收站功能，执行删除索引的操作后，索引将被存放至回收站中，需要再次执行删除索引的操作才能彻底删除索引数据。
“indices.trash.keep.time”	索引在回收站中的存放时长，缺省值是“1d”，表示索引在回收站中存放1天后将被系统彻底删除。最小值为“1d”。

📖 说明

执行如下命令，删除索引。

```
DELETE {INDEX_NAME}
```

其中“INDEX_NAME”为需要删除的索引名称，支持通配符。

查看回收站中的索引

执行如下命令，查看索引回收站中的索引。

```
GET _cat/trash?v=true&s=index
```

表 2-98 参数说明

参数名称	说明
v	当返回值格式是表格形式时，设置是否显示表头。 <ul style="list-style-type: none">• true: 显示表头。• false: 不显示表头。 缺省值是“false”。
s	指定索引的排序字段。可选值为“index”、“uuid”、“pri”、“rep”、“trash.ts”、“delete.time”。
format	设置返回值的呈现格式，默认为表格形式，支持json、yaml、cobr、smile，其中cobr和smile为二进制格式。
h	设置需要显示的列名，默认全部显示，当只需要显示部分列名时，可以通过逗号隔开列名，例如“h=index,uuid,delete.time”。

返回值如下所示，此处是示例，仅供参考，请以实际返回值为准。

```
index  uuid          pri rep  trash.ts delete.time
index1 CMD3FCLzTOyTg4RUekWNNA 1 1 1714465116615 23.6h
index1 6ATijuu6SfqamVI-WMyOKg 1 1 1714466233898 23.9h
```

表 2-99 返回值的参数说明

列名	说明
index	索引名称。
uuid	索引的uuid。
pri	索引的分片数。
rep	索引的副本数。
trash.ts	索引存放至回收站的时间。
delete.time	索引在回收站中剩余存放时长，当变成0时，回收站中的索引将被彻底删除。

从回收站中还原索引

执行如下命令，从回收站中还原索引。

```
POST /trash/recover/{INDEX_NAME}
```

其中“INDEX_NAME”为需要还原的索引名称，支持通配符。

清空回收站

执行如下命令，清空回收站。

```
POST trash/empty
```

执行该命令后，将彻底删除回收站中的索引数据，不可恢复，请谨慎操作。

2.8 配置 Elasticsearch 集群网络

2.8.1 配置公网访问

当集群开启了“安全模式”和“HTTPS访问”时，可以选择是否配置“公网访问”。配置公网访问后，用户可以获得一个公网访问的IP地址，通过这个IP地址可以在公网访问该安全集群。同时，还支持配置访问控制，设置允许公网访问集群的IP地址或网段。

Elasticsearch和OpenSearch集群的公网访问是通过共享型负载均衡器实现外网访问，共享型负载均衡器资源与其他实例共享资源，如果业务追求更优的负载均衡体验，推荐使用独享型负载均衡器接入集群，配置方案请参见[配置独享型负载均衡](#)。

约束限制

- CSS开启公网访问后，会使用到EIP和带宽资源，涉及相关资源费用。
- 仅Elasticsearch集群版本≥6.5.4和OpenSearch集群才支持开启“安全模式”。
- 公网访问和终端节点服务使用的是同一个负载均衡。当集群开启了公网访问白名单，由于白名单是作用在负载均衡上面，将会同时限制公网访问集群和内网通过VPCEP访问集群的IP地址。此时需要在公网访问白名单中添加一个网络白名单“198.19.128.0/17”，通过该白名单用来放通经过VPCEP的流量。

创建集群时配置公网访问

1. 登录云搜索服务管理控制台。
2. 在创建集群页面，开启“安全模式”。设置管理员密码，并启用HTTPS访问。
3. “公网访问”选择“自动绑定”，配置公网访问相关参数。

图 2-16 创建集群时配置公网访问

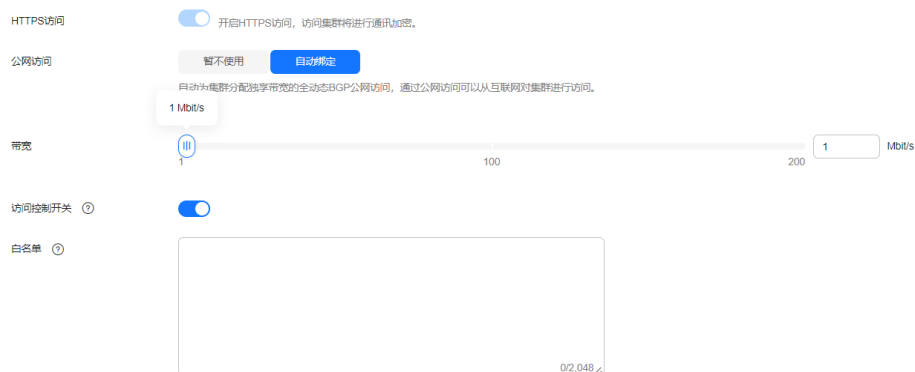


表 2-100 公网访问参数说明

参数	说明
带宽	设置公网访问的带宽。
访问控制开关	如果关闭访问控制开关，则允许任何IP通过公网IP访问集群。如果开启访问控制开关，则只允许白名单列表中的IP通过公网IP访问集群。
白名单	设置允许访问的IP地址或网段，中间用英文逗号隔开。仅当打开“访问控制开关”时才需要配置。

已有集群配置公网访问

您可以对已经创建集群的公网访问进行修改，查看，解绑，也可以配置公网访问。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击需要配置公网访问的集群名称，进入集群基本信息页面，管理公网访问相关配置。

图 2-17 修改公网访问相关配置

配置信息	
区域	
可用区	
虚拟私有云	vpc
子网	subnet
安全组	dw: 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	解绑
访问控制	未开启 设置
带宽	5 Mbit/s 修改
HTTPS访问	开启 下载证书
内网访问IPv4地址	

- 配置公网访问
如果创建安全集群时，开启了HTTPS访问但未配置公网访问，集群创建成功后，可以在集群基本信息页面配置公网访问。
单击“公网访问”参数右侧的“绑定”，设置访问带宽后，单击“确定”。
如果绑定失败，用户可以等待几分钟后，再次尝试重新绑定公网访问。
- 修改
对已经配置了公网访问的集群，可以通过单击“带宽”参数右侧的“修改”，修改带宽大小，也可以通过单击“访问控制”右侧的“设置”，设置访问控制开关和访问白名单。
- 查看
在“基本信息”页面，可以查看当前集群绑定的公网IP地址。
- 解绑
对于已经绑定的公网IP地址，可以通过单击“公网访问”参数右侧的“解绑”，解绑公网IP地址。

通过公网 IP 地址接入集群

公网访问配置完成后，集群将会获得一个“公网访问”的IP地址，用户可以通过公网IP地址和端口接入集群。

例如，启用HTTPS的安全集群的公网访问地址为“10.62.179.32”，端口为“9200”，则执行Curl命令查看集群中的索引信息。

```
curl -u username:password -k 'https://10.62.179.32:9200/_cat/indices'
```

其中username和password表示安全集群的用户名和密码。

2.8.2 配置终端节点服务

终端节点服务（VPC Endpoint Service）通过专属网关，可以将VPC中的服务方便的提供给其他VPC中的资源使用，实现跨VPC的访问，而不必暴露服务端相关的网络信息，使您的访问更加安全、可靠。在开启终端节点服务时，系统会默认给用户创建一个终端节点，内网域名由用户自己选择是否创建，通过节点IP或者是内网域名，用户可以跨VPC访问该集群。

终端节点服务通过共享型负载均衡器实现内网访问，共享型负载均衡器资源与其他实例共享资源，如果业务追求更优的负载均衡体验，推荐使用独享型负载均衡器接入集群，配置方案请参见[配置独享型负载均衡](#)。

约束限制

- 集群开启终端节点服务之后，终端节点将按需进行收费，终端节点的费用将由用户进行支付，详细的计费方式请参考[终端节点计费说明](#)。
- 当集群的网络配置中，“虚拟私有云”选择的是共享VPC，“子网”选择的是共享VPC下的子网，则该集群不支持配置终端节点服务。
- 创建终端节点需要有相关的权限，请参考[VPC终端节点权限管理](#)。
- 公网访问和终端节点服务使用的是同一个负载均衡。当集群开启了公网访问白名单，由于白名单是作用在负载均衡上面，将会同时限制公网访问集群和内网通过VPC访问集群的IP地址。此时需要在公网访问白名单中添加一个网络白名单“198.19.128.0/17”，通过该白名单用来放通过VPC的流量。

创建集群时开启终端节点服务

1. 登录云搜索服务管理控制台。
2. 在右上方单击“创建集群”。
3. 在创建集群页面，“高级配置”选择“自定义”后，开启终端节点服务。

图 2-18 开启终端节点服务

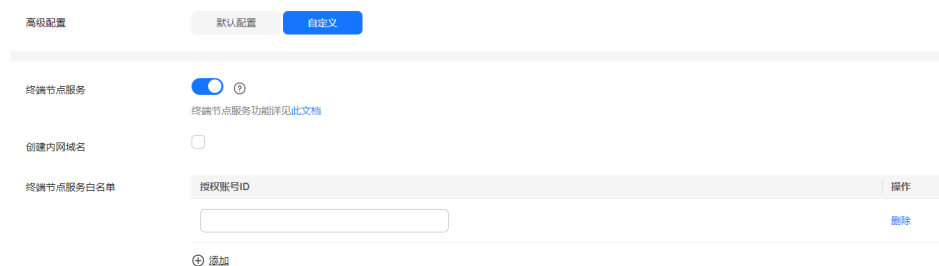


表 2-101 配置终端节点服务

参数	说明
创建内网域名	勾选“创建内网域名”，系统将会自动为用户创建一个内网域名，通过这个域名可以在同一个VPC内访问该集群。
终端节点服务白名单	<p>在“终端节点服务白名单”中添加允许通过节点IP或内网域名访问集群的账号。</p> <ul style="list-style-type: none">单击“添加”输入授权账号ID。授权账号ID配置成“*”，则表示允许全部用户访问该集群。单击操作列的“删除”，可以删除不允许访问的账号ID。 <p>说明 “授权账号ID”可以在“我的凭证”中进行查看“账号ID”获取。</p>

已有集群终端节点服务管理

如果创建集群时未开启终端节点服务，集群创建成功后，可以通过如下步骤进行开启。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击需要开启终端节点服务的集群名称，进入集群基本信息页面。
3. 选择“终端节点服务”，在“终端节点服务”右侧单击开关，打开集群的终端节点服务功能。

表 2-102 配置终端节点服务

参数	说明
创建内网域名	勾选“创建内网域名”，系统将会自动为用户创建一个内网域名，通过这个域名可以在同一个VPC内访问该集群。
终端节点服务白名单	<p>在“终端节点服务白名单”中添加允许通过节点IP或内网域名访问集群的账号。</p> <ul style="list-style-type: none">单击“添加”输入授权账号ID。授权账号ID配置成“*”，则表示允许全部用户访问该集群。单击操作列的“删除”，可以删除不允许访问的账号ID。 <p>说明 “授权账号ID”可以在“我的凭证”中进行查看“账号ID”获取。</p>

4. 管理终端节点。
在终端节点服务页面下，显示所有连接当前终端节点服务的终端节点。

图 2-19 管理终端节点

终端节点ID	状态	最大连接数	拥有者	创建时间	操作
a24dc31c	已接受	3000		2024/03/15 16:14:20 GMT+08:00	接受 拒绝

单击操作列的“接受”或者“拒绝”可以修改节点的“状态”。如果对某个终端节点“拒绝”操作之后，其生成的内网域名将不能再访问到当前集群。

通过节点 IP 或内网域名访问集群

1. 获取集群的内网域名或者节点IP。

登录云搜索服务控制台，进入集群列表，单击集群名称，进入集群“基本信息”页面，选择“终端节点服务”，查看“节点IP”和“内网域名”。

图 2-20 查看节点 IP 和内网域名信息

基本信息	
终端节点服务名称	cn
节点IP	192
内网域名	vpcep- .com
终端节点服务白名单	-- 修改

2. 在弹性云服务器中，直接通过Curl命令调用Elasticsearch API即可使用集群。

弹性云服务器需要满足如下要求：

- 为弹性云服务分配足够的磁盘空间。
- 此弹性云服务器的VPC需要与集群在同一个VPC中，开通终端节点服务后，可以实现跨VPC访问。
- 此弹性云服务器的安全组需要和集群的安全组相同。
如果不同，请修改弹性云服务器安全组或配置弹性云服务器安全组的出入规则允许集群所有安全组的访问。修改操作请参见[配置安全组规则](#)。
- 待接入的CSS集群，其安全组的出方向和入方向需允许TCP协议及9200端口，或者允许端口范围包含9200端口。

例如，执行如下Curl命令，查看集群中的索引信息，集群中的内网访问地址为“vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com”，端口为“9200”。

- 如果接入集群未启用安全模式，接入方式为：

```
curl 'http://vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com:9200/_cat/indices'
```
- 如果接入集群已启用安全模式，则需要使用https方式访问，并附加用户名和密码，在curl命令中添加-u选项。

```
curl -u username:password -k 'https://vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com:9200/_cat/indices'
```

2.8.3 配置独享型负载均衡

云搜索服务集成了共享型负载均衡器的能力，支持绑定公网访问和开启终端节点服务。相比共享型负载均衡器，独享型负载均衡器功能更丰富、性能更强，本章主要介绍如何配置集群的独享型负载均衡。

场景描述

使用独享型负载均衡器接入集群具有如下优点：

- 非安全模式集群也支持集成弹性负载均衡服务的能力。
- 支持用户使用自定义证书进行HTTPS双向认证。
- 支持7层流量监控及告警配置，方便用户随时查看监控情况。

不同安全模式的集群对接独享型负载均衡器共有8种不同的服务形态，不同服务形态对应的ELB能力如表2-103所示，8种组合的配置概览如表2-104所示。

表 2-103 不同集群对应的 ELB 能力

集群安全模式	ELB最终对外提供的服务形态	ELB负载均衡	ELB流量监控	ELB双向认证
非安全	无认证	支持	支持	不支持
	单向认证	支持	支持	支持
	双向认证			
安全+HTTP	密码认证	支持	支持	不支持
	单向认证+密码认证	支持	支持	支持
	双向认证+密码认证			
安全+HTTPS	单向认证+密码认证 双向认证+密码认证	支持	支持	支持

表 2-104 不同集群对接 ELB 的配置概览

集群安全模式	ELB最终对外提供的服务形态	ELB监听器			后端服务器组		
		前端协议	前端端口	SSL解析方式	后端协议	健康检查端口	健康检查路径
非安全	无认证	HTTP	9200	无认证	HTTP	9200	/
	单向认证	HTTPS	9200	单向认证	HTTP	9200	
	双向认证	HTTPS	9200	双向认证	HTTP	9200	

安全+HT TP	密码认证	HTTP	9200	无认证	HTTP	9200	/_opendis tro/ _security /health
	单向认证+密 码认证	HTTPS	9200	单向认 证	HTTP	9200	
	双向认证+密 码认证	HTTPS	9200	双向认 证	HTTP	9200	
安全+HT TPS	单向认证+密 码认证	HTTPS	9200	单向认 证	HTTPS	9200	
	双向认证+密 码认证	HTTPS	9200	双向认 证	HTTPS	9200	

CSS集群对接独享型负载均衡的操作步骤如下：

1. 如果规划ELB监听器的前端协议是HTTPS，则需要准备签名证书并上传至ELB管理控制台：[准备并上传自签名证书](#)
2. 在ELB管理控制台创建独享型负载均衡器：[创建独享型负载均衡器](#)
3. Elasticsearch或OpenSearch集群开启负载均衡：[集群对接负载均衡器](#)
4. 通过独享型负载均衡实例接入集群：[通过Curl命令接入集群](#)

本文还提供了通过独享型负载均衡实例接入集群的Java客户端代码示例：[通过负载均衡器的HTTPS双向认证方式接入集群的代码示例（Java）](#)

约束限制

- 如果ELB绑定了公网，则不推荐接入非安全模式的集群。非安全模式的集群使用HTTP通信且不需要安全认证即可访问，如果对接公网访问存在安全风险。
- 开启了HTTPS访问的安全模式的集群不支持HTTP类型的前端协议认证，如果需要使用HTTP类型的前端协议，需要将集群的“安全模式”变更为“非安全模式”，具体操作请参见[更改Elasticsearch集群安全模式](#)。在变更安全模式前，请先关闭“负载均衡”，安全模式变更完成后再开启“负载均衡”。

准备并上传自签名证书

当规划的ELB监听器的前端协议为HTTPS时，需要参考本节准备自签名证书，作为服务器证书或CA证书上传至ELB。

说明

建议使用云证书管理服务CCM购买的证书，或者其他权威机构颁发的证书。

1. 登录到任意一台安装有OpenSSL工具和JDK的Linux客户端。
2. 执行如下命令制作自签名证书。

```
mkdir ca
mkdir server
mkdir client

#使用OpenSSL制作CA证书
cd ca
#创建CA证书的openssl配置文件ca_cert.conf
cat >ca_cert.conf <<EOF
[ req ]
```

```
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
EOF
#创建CA证书私钥文件ca.key
openssl genrsa -out ca.key 2048
#创建CA证书的csr请求文件ca.csr
openssl req -out ca.csr -key ca.key -new -config ./ca_cert.conf
#创建自签名的CA证书ca.crt
openssl x509 -req -in ca.csr -out ca.crt -sha1 -days 5000 -signkey ca.key
#把ca证书格式转为p12格式
openssl pkcs12 -export -clcerts -in ca.crt -inkey ca.key -out ca.p12
#把ca证书格式转为jks格式
keytool -importkeystore -srckeystore ca.p12 -srcstoretype PKCS12 -deststoretype JKS -destkeystore
ca.jks

#使用CA证书签发服务器证书
cd ../server
#创建服务器证书的openssl配置文件server_cert.conf，CN字段根据需求改为服务器对应的域名、IP地址
cat >server_cert.conf <<EOF
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
CN = 127.0.0.1
EOF
#创建服务器证书私钥文件server.key
openssl genrsa -out server.key 2048
#创建服务器证书的csr请求文件server.csr
openssl req -out server.csr -key server.key -new -config ./server_cert.conf
#使用CA证书签发服务器证书server.crt
openssl x509 -req -in server.csr -out server.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -
CAkey ../ca/ca.key
#把服务端证书格式转为p12格式
openssl pkcs12 -export -clcerts -in server.crt -inkey server.key -out server.p12
#把服务证书格式转为jks格式
keytool -importkeystore -srckeystore server.p12 -srcstoretype PKCS12 -deststoretype JKS -destkeystore
server.jks

#使用CA证书签发客户端证书
cd ../client
#创建客户端证书的openssl配置文件client_cert.conf，CN字段根据需求改为服务器对应的域名、IP地址
cat >client_cert.conf <<EOF
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
CN = 127.0.0.1
EOF
#创建客户端证书私钥文件client.key
openssl genrsa -out client.key 2048
#创建客户端证书的csr请求文件client.csr
openssl req -out client.csr -key client.key -new -config ./client_cert.conf
#使用CA证书签发客户端证书client.crt
openssl x509 -req -in client.csr -out client.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -
CAkey ../ca/ca.key
#把客户端证书格式转为浏览器可识别的p12格式
openssl pkcs12 -export -clcerts -in client.crt -inkey client.key -out client.p12
#把客户端证书格式转为jks格式
keytool -importkeystore -srckeystore client.p12 -srcstoretype PKCS12 -deststoretype JKS -destkeystore
client.jks
```

3. 上传自签名证书，具体操作请参见[创建证书](#)。

创建独享型负载均衡器

1. 登录弹性负载均衡管理控制台。
2. 参考[创建独享型负载均衡器](#)，创建独享型负载均衡器。CSS集群对接独享型负载均衡器所需要关注的参数如表2-105所示，其他参数请根据实际需要填写。

表 2-105 独享型负载均衡器的配置说明

参数	配置说明	取值样例
实例类型	选择“独享型”。	独享型
计费模式	性能独享型负载均衡器的收费类型。	按需计费
区域	选择CSS集群所在的区域。	-
跨VPC后端	开启跨VPC后端才能连接CSS集群。	开启
网络类型	负载均衡器对外提供服务所使用的网络类型。	IPv4私网
所属VPC	所属虚拟私有云。无论选择哪种网络类型，均需配置此项。 需要选择和CSS集群同一VPC。	-
子网	选择创建负载均衡实例的子网。无论选择哪种网络类型，均需配置此项。 需要选择和CSS集群同一子网。	-
规格	建议选择功能和性能更优的应用型规格。	应用型(HTTP/HTTPS) “小型1”

集群对接负载均衡器

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要对接负载均衡器的集群，单击集群名称进入集群“基本信息”页面。
3. 在左侧菜单栏，选择“负载均衡”，打开负载均衡开关，在弹窗中配置负载均衡基础信息。

表 2-106 配置负载均衡

参数	说明
负载均衡器	选择前面创建的独享型负载均衡器。CSS集群属于托管资源，所选的负载均衡器必须开启“跨VPC后端”能力才能正常使用。
委托	授权CSS服务访问和使用ELB资源。所选委托必须包含策略“ELB Administrator”或“ELB FullAccess”。

图 2-21 开启负载均衡

基础配置

负载均衡器 [创建负载均衡器](#) [?](#)

委托 [创建委托](#) [?](#)

[取消](#) [确定](#)


- 单击“确定”，开启负载均衡。
- 在“监听器配置”区域，您可以单击右侧  配置监听器的相关信息。

表 2-107 配置监听器

参数	配置说明
前端协议	客户端与负载均衡监听器建立流量分发连接的协议。支持选择“HTTP”或“HTTPS”。 根据实际业务需要选择协议。
前端端口	客户端与负载均衡监听器建立流量分发连接的端口。 根据实际业务需要自定义。
SSL解析方式	客户端到服务器端认证方式。仅“前端协议”选择“HTTPS”才需要配置。 根据实际业务需要选择解析方式。
服务器证书	服务器证书用于SSL握手协商，需提供证书内容和私钥。仅“前端协议”选择“HTTPS”才需要配置服务器证书。 选择 准备并上传自签名证书 创建的服务器证书。
CA证书	CA证书又称客户端CA公钥证书，用于验证客户端证书的签发者。仅“SSL解析方式”选择“双向认证”时，才需要配置CA证书。 选择 准备并上传自签名证书 创建的CA证书。 在开启HTTPS双向认证功能时，只有当客户端能够出具指定CA签发的证书时，HTTPS连接才能成功。

图 2-22 配置监听器

监听器配置

前端协议	HTTP	HTTPS
前端端口	<input type="text"/>	
SSL解析方式	单向认证	双向认证
服务器证书	server-new	<input type="button" value="Q"/>
CA证书	ca-new	<input type="button" value="Q"/>

- （可选）在监听器配置区域，单击“访问控制”后的“设置”跳转到负载均衡器的监听器列表，单击监听器访问控制列的“设置”，配置允许通过负载均衡实例访问集群的IP地址组，不设置的话默认允许所有的IP地址访问。
- 在健康检查区域，您可以查看各个节点IP的健康检查结果。

表 2-108 健康检查结果状态说明

健康检查结果	说明
正常	节点IP连接正常。
异常	节点IP连接，不可用

通过 Curl 命令接入集群

- 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。
- 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。
- 在左侧菜单栏选择“负载均衡”，获取并记录负载均衡实例的“私有IP”或“公网IP”。
- 在ECS服务器中执行如下Curl命令测试能否通过独享型负载均衡实例接入集群。

表 2-109 不同集群的接入命令

集群安全模式	ELB最终对外提供的服务形态	接入集群的Curl命令
非安全	无认证	curl http://IP9200
	单向认证	curl -k --cert ./client.crt --key ./client.key https://IP9200

集群安全模式	ELB最终对外提供的服务形态	接入集群的Curl命令
	双向认证	<code>curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:9200</code>
安全+HTTP	密码认证	<code>curl http://IP:9200 -u user:pwd</code>
	单向认证+密码认证	<code>curl -k --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>
	双向认证+密码认证	<code>curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>
安全+HTTPS	单向认证+密码认证	<code>curl -k --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>
	双向认证+密码认证	<code>curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>

表 2-110 变量说明

变量名	说明
IP	负载均衡实例的“私有IP”或“公网IP”。
user	集群的用户名，仅安全集群需要配置。
pwd	用户名对应的密码，仅安全集群需要配置。

当可以正常返回集群信息时，表示连接成功。

通过负载均衡器的 HTTPS 双向认证方式接入集群的代码示例（Java）

在Java客户端，通过负载均衡器的HTTPS双向认证方式接入集群的代码示例。

ESSecuredClientWithCerDemo 代码示例

```
import org.apache.commons.io.IOUtils;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.HttpHost;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.elasticsearch.action.search.SearchRequest;
import org.elasticsearch.action.search.SearchResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.index.query.QueryBuilders;
import org.elasticsearch.search.SearchHit;
import org.elasticsearch.search.SearchHits;
import org.elasticsearch.search.builder.SearchSourceBuilder;
import java.io.InputStream;
```

```
import java.io.IOException;
import java.security.KeyStore;
import java.security.SecureRandom;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManagerFactory;
public class ESSecuredClientWithCerDemo {
    private static final String KEY_STORE_PWD = "";
    private static final String TRUST_KEY_STORE_PWD = "";
    private static final String CA_JKS_PATH = "ca.jks";
    private static final String CLIENT_JKS_PATH = "client.jks";
    private static final String ELB_ADDRESS = "127.0.0.1";
    private static final int ELB_PORT = 9200;
    private static final String CSS_USERNAME = "user";
    private static final String CSS_PWD = "";
    public static void main(String[] args) {
        // 建立客户端
        RestHighLevelClient client = initESClient(ELB_ADDRESS, CSS_USERNAME, CSS_PWD);
        try {
            // 查询 match_all, 相当于 {"query": {"match_all": {}}}
            SearchRequest searchRequest = new SearchRequest();
            SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
            searchSourceBuilder.query(QueryBuilders.matchAllQuery());
            searchRequest.source(searchSourceBuilder);
            // query
            SearchResponse searchResponse = client.search(searchRequest, RequestOptions.DEFAULT);
            System.out.println("query result: " + searchResponse.toString());
            SearchHits hits = searchResponse.getHits();
            for (SearchHit hit : hits) {
                System.out.println(hit.getSourceAsString());
            }
            System.out.println("query success");
            Thread.sleep(2000L);
        } catch (InterruptedException | IOException e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(client);
        }
    }
    private static RestHighLevelClient initESClient(String clusterAddress, String userName, String password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(userName,
password));
        SSLContext ctx = null;
        try {
            KeyStore ks = getKeyStore(CLIENT_JKS_PATH, KEY_STORE_PWD, "JKS");
            KeyManagerFactory kmf = KeyManagerFactory.getInstance("SunX509");
            kmf.init(ks, KEY_STORE_PWD.toCharArray());
            KeyStore tks = getKeyStore(CA_JKS_PATH, TRUST_KEY_STORE_PWD, "JKS");
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509");
            tmf.init(tks);
            ctx = SSLContext.getInstance("SSL", "SunJSSE");
            ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(), new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(ctx, new HostnameVerifier() {
            @Override
            public boolean verify(String arg0, SSLSession arg1) {
                return true;
            }
        });
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);
        RestClient.Builder builder = RestClient.builder(new HttpHost(clusterAddress, ELB_PORT, "https"))
.setHttpClientConfigCallback(httpClientConfigCallback);
```

```
RestHighLevelClient client = new RestHighLevelClient(builder);
return client;
}
private static KeyStore getKeyStore(String path, String pwd, String type) {
    KeyStore keyStore = null;
    FileInputStream is = null;
    try {
        is = new FileInputStream(path);
        keyStore = KeyStore.getInstance(type);
        keyStore.load(is, pwd.toCharArray());
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        IOUtils.closeQuietly(is);
    }
    return keyStore;
}
}
```

SecuredHttpClientConfigCallback 代码示例

```
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;
import java.util.Objects;
class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;
    /**
     * The {@link SSLIOStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOStrategy sslStrategy;
    /**
     * Create a new {@link SecuredHttpClientConfigCallback}.
     *
     * @param credentialsProvider The credential provider, if a username/password have been supplied
     * @param sslStrategy The SSL strategy, if SSL / TLS have been supplied
     * @throws NullPointerException if {@code sslStrategy} is {@code null}
     */
    SecuredHttpClientConfigCallback(final SSLIOStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }
    /**
     * Get the {@link CredentialsProvider} that will be added to the HTTP client.
     *
     * @return Can be {@code null}.
     */
    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }
    /**
     * Get the {@link SSLIOStrategy} that will be added to the HTTP client.
     *
     * @return Never {@code null}.
     */
    SSLIOStrategy getSSLStrategy() {
        return sslStrategy;
    }
    /**
     * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
    credential provider},
     *
     * @param httpClientBuilder The client to configure.
     * @return Always {@code httpClientBuilder}.
     */
}
```

```
*/
@Override
public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
    // enable SSL / TLS
    httpClientBuilder.setSSLStrategy(sslStrategy);
    // enable user authentication
    if (credentialsProvider != null) {
        httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
    }
    return httpClientBuilder;
}
}
```

pom.xml 代码示例

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>1</groupId>
    <artifactId>ESClient</artifactId>
    <version>1.0-SNAPSHOT</version>
    <name>ESClient</name>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <elasticsearch.version>7.10.2</elasticsearch.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>transport</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch</groupId>
            <artifactId>elasticsearch</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>elasticsearch-rest-high-level-client</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>commons-io</groupId>
            <artifactId>commons-io</artifactId>
            <version>2.11.0</version>
        </dependency>
    </dependencies>
</project>
```

2.9 备份与恢复 Elasticsearch 集群数据

2.9.1 创建快照备份 Elasticsearch 集群数据

本文介绍CSS集群快照备份的两种方式：自动创建快照和手动创建快照。

约束限制

- 备份与恢复索引功能上线之前（即2018年3月10日之前）创建的集群，无法创建快照。
- 第一次备份时，建议将所有索引数据进行备份。
- 集群快照会导致CPU、磁盘IO上升等影响，建议在业务低峰期进行操作。
- 修改存储快照的OBS桶之后，集群将会从新的OBS桶中获取快照数据。
- 当集群处于“不可用”状态时，快照功能中除了恢复快照功能外，其他快照信息或功能只能查看，无法进行编辑。
- 备份与恢复过程中，支持集群扩容（如果是恢复至本集群，则本集群不支持扩容）、访问Kibana、查看监控、删除其他快照的操作。不支持重启此集群、删除此集群、删除正在创建或恢复的快照、再次创建或恢复快照的操作。当此集群正在进行创建快照或者恢复快照过程中，同时的自动创建快照任务将被取消。
- CSS集群第一次快照是全量，后面再备份快照是在之前的快照基础上增量，CSS是增量快照逻辑，快照之间的文件会相互依赖。

前提条件

- 已创建用于存储快照的OBS桶。快照存储在OBS桶中需要额外收费，具体收费请参见[OBS计费说明](#)。
- 确认账号具有创建或恢复快照功能的权限。可以在IAM服务的“权限管理 > 权限”查看是否有以下权限，如果没有请联系服务管理员添加。
 - “CSS Administrator”角色。
 - “OBS Administrator”策略

设置自动创建快照


1. 登录云搜索服务管理控制台。
2. 开启集群自动创建快照。支持在创建集群的时候就打开集群自动创建快照，操作指导请参见[7](#)，或者在集群创建完成之后再开启集群自动创建快照。
 - 如果在创建集群时已经打开集群自动创建快照，请执行[7](#)。
 - 如果是集群创建完成之后开启集群自动创建快照，请执行下一步。
3. 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
4. 在“集群快照”管理页面，单击“集群快照开关”右侧开关，打开集群快照功能。如果已打开集群快照功能，则跳过该步骤。
打开集群快照功能后，CSS服务会自动为客户创建OBS桶和IAM委托，用于存储快照。自动创建的OBS桶和IAM委托将直接展示在界面中。如果您不希望使用自动创建的OBS桶和IAM委托，您可以在“基础配置”右侧单击进行配置。

表 2-111 集群快照基础配置

参数	说明
OBS桶	在下拉框中选择存储快照的OBS桶。也可以单击右侧的“创建桶”新建OBS。详细操作步骤请参见 创建桶 。 创建或者已存在的OBS桶需满足如下条件： <ul style="list-style-type: none">“存储类别”为“标准存储”。“区域”须与创建的集群所在区域相同。
备份路径	快照在OBS桶中的存放路径。 备份路径配置规则： <ul style="list-style-type: none">备份路径不能使用符号“\:*?\"<> ”。备份路径不能以“/”开头。备份路径不能以“.”开头或结尾。备份路径的总长度不能超过1023个字符。
IAM委托	指当前账号授权云搜索服务访问或维护存储在OBS中数据。如果没有合适的委托可以联系CSS服务管理员新建IAM委托。详细操作步骤请参见 如何创建委托 。 所选的IAM委托需满足如下条件： <ul style="list-style-type: none">“委托类型”选择“云服务”。“云服务”选择“Elasticsearch”或者“云搜索服务 CSS”。必选策略：“OBS Administrator”


5. 在“自动创建快照”右侧，单击开关开启自动创建快照功能，弹出“创建快照策略”页面。如果已启用自动创建快照功能，也可以在开关右侧单击进行快照策略修改。

表 2-112 设置自动创建快照

参数	说明
快照名称前缀	快照名称前缀的长度为1~32个字符，只能包含小写字母、数字、中划线和下划线，且必须以小写字母开头。快照名称由快照名称前缀加上时间戳组成，例如自动生成的快照名称为“snapshot-1566921603720”。
时区	指备份时间对应的时区，不支持修改。基于此时区选择备份开始时间。
备份开始时间	指每天自动开始备份的时间，只能指定整点时间，如00:00、01:00，取值范围为00:00~23:00。请在下拉框中选择时间。

参数	说明
保留快照个数	<p>自定义设置自动快照保留的个数，范围是1~90。系统在半点时刻会自动删除超过保留个数的快照（过期删除策略只针对与当前自动创建快照策略相同执行频次的自动快照）。</p> <p>说明 保存快照个数与自动创建快照策略中设置的执行频次和索引有关，当执行频次间隔时间短或索引数量最大，保留自动快照可能会达不到设置的个数，请谨慎选择。</p>
索引	<p>填写索引名称，支持选择索引进行备份。索引名称不能包含空格和大写字母，且不能包含“\< >/?”特殊字符，多个索引之间使用英文逗号隔开。如果不填写，则默认备份集群中所有索引。支持使用“*”匹配多个索引，例如“index*”，表示备份名称前缀是index的所有索引的数据。</p> <p>说明 在Kibana中使用GET /_cat/indices命令，可以查询集群中所有索引的名称。</p>

图 2-23 自动创建快照

创建快照策略

快照名称前缀

时区 GMT+08:00

备份开始时间

保留快照个数

索引

如果您确定要设置当前选择的自动备份策略，请输入CONFIRM

6. 确定要设置当前选择的自动备份策略，请在输入框输入“CONFIRM”，设置完成后，单击“确定”保存快照策略。
7. 按照策略自动创建的快照将呈现在快照管理列表中。快照列表同时展示自动创建和手动创建的快照，您可以通过快照类型参数进行区分。在快照列表右上角，您可以输入快照名称或快照ID的关键字进行查找。

📖 说明

如果快照创建失败，可以在快照列表的右侧单击查看“失败任务数”，了解失败原因。失败任务数最多显示20条，当快照功能关闭或集群删除时，失败记录也会被清空。

8. （可选）关闭自动创建快照功能。

关闭自动创建快照功能后，系统将停止继续自动创建快照。如果系统正在根据策略自动创建快照，而快照列表还未呈现正在创建的快照时，无法关闭自动创建快照功能。如果您单击了关闭按钮，系统将提示您无法关闭。建议等快照自动创建成功后，即快照列表已出现最新创建的快照时，再单击关闭按钮，关闭自动创建快照功能。

关闭自动创建快照功能时，您可以在弹出窗口中通过“删除自动创建的快照”选项，选择是否立即删除之前已自动创建的快照，默认不勾选。

- 不勾选：表示不会删除关闭此功能前已自动创建的快照。如果不删除，后续还可以在快照列表中通过删除按钮手动删除，详细操作指导请参见[删除 Elasticsearch 集群快照](#)。如果未手动删除，且之后用户又重新开启了自动创建快照功能，那么此集群中所有“快照类型”为自动创建的快照（包含开启自动创建快照功能前已存在的自动创建的快照）都无法手动删除，只会被系统自动删除。系统会基于重新开启自动创建快照功能时的配置策略进行自动删除，例如此策略中定义的保留快照个数为10个，那么当快照个数超过10个，系统在半点时刻会自动删除超过保留个数的快照。
- 勾选：表示删除此集群快照列表中所有“快照类型”为自动创建的快照。

📖 说明

快照功能关闭时，不会主动清理快照数据。如果需要清理快照数据，可前往OBS控制台存储快照的OBS桶对快照数据进行主动管理。

手动创建快照

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
3. 在“集群快照”管理页面，单击“集群快照开关”右侧开关，打开集群快照功能。如果已打开集群快照功能，则跳过该步骤。


打开集群快照功能后，CSS服务会自动为客户创建OBS桶和IAM委托，用于存储快照。自动创建的OBS桶和IAM委托将直接展示在界面中。如果您不希望使用自动创建的OBS桶和IAM委托，您可以在“基础配置”右侧单击进行配置。

表 2-113 集群快照基础配置

参数	说明
OBS桶	在下拉框中选择存储快照的OBS桶。也可以单击右侧的“创建桶”新建OBS。详细操作步骤请参见 创建桶 。 创建或者已存在的OBS桶需满足如下条件： <ul style="list-style-type: none">• “存储类别”为“标准存储”。• “区域”须与创建的集群所在区域相同。

参数	说明
备份路径	快照在OBS桶中的存放路径。 备份路径配置规则： <ul style="list-style-type: none">• 备份路径不能使用符号“\:*?*<> ”。• 备份路径不能以“/”开头。• 备份路径不能以“.”开头或结尾。• 备份路径的总长度不能超过1023个字符。
IAM委托	指当前账号授权云搜索服务访问或维护存储在OBS中数据。如果没有合适的委托可以联系CSS服务管理员新建IAM委托。详细操作步骤请参见 如何创建委托 。 所选的IAM委托需满足如下条件： <ul style="list-style-type: none">• “委托类型”选择“云服务”。• “云服务”选择“Elasticsearch”或者“云搜索服务 CSS”。• 必选策略：“OBS Administrator”

4. 完成基础配置后，单击下方“创建快照”可手动创建快照。

表 2-114 创建快照配置

参数	说明
快照名称	手动创建的快照名称，4~64个字符，只能包含小写字母、数字、中划线和下划线，且必须以字母开头。与自动创建不同，手动创建的快照名称按照用户设置的名称，不会自动加上时间信息。
索引	填写索引名称，支持选择索引进行备份。索引名称不能包含空格和大写字母，且不能包含“\< >/?”特殊字符，多个索引之间使用英文逗号隔开。如果不填写，则默认备份集群中所有索引。支持使用“*”匹配多个索引，例如“index*”，表示备份名称前缀是index的所有索引的数据。 说明 在Kibana中使用GET /_cat/indices命令，可以查询集群中所有索引的名称。
快照描述	创建的快照描述信息。0~256个字符，不能包含“<>”字符。

图 2-24 手动创建快照

创建快照

快照名称

索引 0/1,024

快照描述 0/256

5. 单击“确定”开始创建快照。

快照创建完成后，将直接呈现在快照管理列表中，快照状态为“可用”表示快照创建成功。快照列表同时展示自动创建和手动创建的快照，您可以通过快照类型参数进行区分。在快照列表右上角，您可以输入快照名称或快照ID的关键字进行查找。

说明

如果快照创建失败，可以在快照列表的右侧单击查看“失败任务数”，了解失败原因。失败任务数最多显示20条，当快照功能关闭或集群删除时，失败记录也会被清空。

2.9.2 恢复 Elasticsearch 集群数据

将已有的快照，通过恢复快照功能，将备份的集群快照恢复到本集群或其他满足要求的集群中。

约束限制

- 备份与恢复索引功能上线之前（即2018年3月10日之前）创建的集群，无法创建快照。
- 集群快照会导致CPU、磁盘IO上升等影响，建议在业务低峰期进行操作。
- 当集群处于“不可用”状态时，快照功能中，除了恢复快照功能外，其他快照信息或功能只能查看，无法进行编辑。
- 备份与恢复过程中，支持集群扩容（如果是恢复至本集群，则本集群不支持扩容）、访问Kibana、查看监控、删除其他快照的操作。不支持重启此集群、删除此集群、删除正在创建或恢复的快照、再次创建或恢复快照的操作。当此集群正在进行创建快照或者恢复快照过程中，同时的自动创建快照任务将被取消。
- 快照在恢复过程中不可查询集群数据。
- 当两个集群的shard不一样时，则无法恢复集群数据。
- 用于恢复的目标集群版本不得低于源集群版本，一般保持版本一致即可。

前提条件

- 快照管理列表中有“快照状态”为“可用”的快照。快照存储在OBS桶中需要额外收费，具体收费请参见[OBS计费说明](#)。
- 确认账号具有创建或恢复快照功能的权限。可以在IAM服务的“权限管理 > 权限”查看是否有以下权限，如果没有请联系服务管理员添加。
 - “CSS Administrator”角色。
 - “OBS Administrator”策略

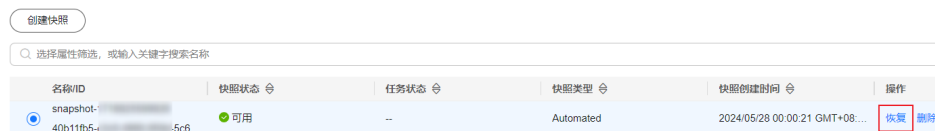
恢复数据

须知

恢复数据将覆盖集群中当前的数据，请谨慎操作。

- 登录云搜索服务管理控制台。
- 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
- 在快照管理列表中，选择需要恢复的快照，单击“操作”列的“恢复”。

图 2-25 选择恢复快照



- 在“恢复”页面配置集群的恢复参数。

表 2-115 恢复配置

参数	说明
索引	指定需要进行恢复的索引名称，默认为空。如保持默认值，即不指定索引名称，则表示恢复所有的索引数据。支持0~1024个字符，不能包含空格和大写字母，且不能包含“\< >/?”特殊字符。支持使用“*”匹配多个索引，比如index*，表示恢复快照中名称前缀是index的所有索引。
索引名称匹配模式	在恢复时，可以根据文本框中定义的过滤条件去恢复符合条件的索引，过滤条件请使用正则表达式。默认值“index_(.)+”表示所有的索引。支持0~1024个字符，不能包含空格和大写字母，且不能包含“\< >/?”特殊字符。 说明 “索引名称匹配模式”和“索引名称替换模式”需要同时设置才会生效。

参数	说明
索引名称替换模式	<p>索引重命名的规则。默认值“restored_index_\$1”表示在所有恢复的索引名称前面加上“restored_”。支持0~1024个字符，不能包含空格和大写字母，且不能包含“\< >/?,”特殊字符。</p> <p>说明 “索引名称匹配模式”和“索引名称替换模式”需要同时设置才会生效。</p>
集群	<p>选择需要进行恢复的集群名称，可选择当前集群或者其他集群。</p> <ul style="list-style-type: none">只能选择处于“可用”状态的集群，如果快照所属的集群处于“不可用”状态，那么也无法将快照恢复到本集群。恢复到其他集群时，目标集群的版本不低于本集群的版本。选择“是否覆盖目标集群同名同shard结构索引”，建议不覆盖，即不勾选。通过快照恢复数据，是以快照文件覆盖的形式进行数据恢复，覆盖目标集群同名的索引后，可能导致目标集群同名索引数据丢失，请谨慎勾选操作。

- 单击“确定”开始恢复。恢复成功，快照列表中“任务状态”将变更为“恢复成功”，索引数据将根据快照信息重新生成。

2.9.3 删除 Elasticsearch 集群快照

当快照信息不需要使用时，您可以删除快照释放存储资源。

当自动创建快照功能开启时，自动创建的快照无法手动删除，系统会按照设置的策略在半点时刻自动删除超过“保留时间”的快照。

当自动创建快照功能开启又关闭了，关闭时，之前已自动创建的快照并未同步删除，这时快照列表中自动创建的快照可以手动删除。如果未手动删除，且之后用户又重新开启了自动创建快照功能，那么此集群中所有“快照类型”为自动创建的快照（包含开启自动创建快照功能前已存在的自动创建的快照）都无法手动删除，只会被系统自动删除。

说明

快照信息删除后，数据将无法恢复，请谨慎操作。

手动删除快照

- 登录云搜索服务管理控制台。
- 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
- 在快照管理列表中，选择需要删除的快照。

- 单击“操作”列的“删除”，在弹窗中确认要删除的快照信息后，单击“确定”删除快照。

2.10 扩缩容 Elasticsearch 集群

2.10.1 扩容 Elasticsearch 集群

当Elasticsearch或OpenSearch集群的数据面业务变化，需要动态调整集群节点的数量、容量、规格和类型时，可以参考本文扩容集群。

场景描述

CSS服务支持多种扩容场景，如[表2-116](#)所示。

表 2-116 扩容场景

场景	描述	相关文档
增加节点数量和容量	<p>只支持基于节点类型扩容集群的“节点数量”和“节点存储容量”。适用于如下场景：</p> <ul style="list-style-type: none">当集群数据节点（ess）的写入与查询压力大、响应时间过长时，可以通过扩容数据节点的“节点存储容量”保证数据的持久性。如果因数据量过大或操作不当导致数据节点状态异常时，可以扩容“节点数量”保证集群的可用性。冷数据节点（ess-cold）主要用于分担数据节点（ess）的压力，当发现冷数据有丢失的风险时，可以扩容冷数据节点的“节点存储容量”保证冷数据的持久性，同时也支持扩容节点个数保证集群的可用性。	扩容节点数量和节点存储容量

场景	描述	相关文档
升级节点规格	<p>只支持基于节点类型变更集群的“节点规格”。一般用于小规格变更为大规格，常用于如下场景：</p> <ul style="list-style-type: none">当新增索引或分片分配的处理时间过长，或管理集群各个节点的协调、调度不足时，可以升级Master节点（ess-master）的“节点规格”保证集群的正常使用。当数据节点任务分发量、结果汇聚量过大时，可以升级Client节点（ess-client）的“节点规格”。当数据的写入与查询突然变得缓慢时，可以升级数据节点（ess）的“节点规格”提高数据节点的查询与写入效率。当存在冷数据查询缓慢时，可以升级冷数据节点（ess-cold）的“节点规格”提高对数据查询的效率。	变更节点规格
增加节点类型	<p>当集群本身没有Master节点（ess-master）和Client节点（ess-client）时，随着数据面业务增长，需要动态调整集群形态时，可以选择增加“Master节点”或“Client节点”。</p>	添加Master或Client节点

约束限制

- 集群的节点存储容量只支持扩容不支持缩容，请基于业务量合理选择节点存储容量。
- 包周期集群不支持同时扩容“节点数量”和“节点存储容量”。
- 在集群数据量比较大的情况下，变更节点规格耗时比较长，建议在业务低峰期更改节点规格，利于更快完成规格更改。
- 变更节点规格也支持将大规格更改为小规格，但是此变更会降低集群的处理性能，进而影响业务能力，请谨慎使用。
- 当集群包含多种节点类型时，一次只支持变更一种类型的节点规格，且变更完成后只生效所选类型的节点规格。
- 变更规格过程中，Kibana不可用。
- 变更规格过程中，会依次对节点进行关机，完成更改后再依次开机，是一个滚动的变更过程。在这过程中，Kibana不可用。
- 当一个集群包含的节点类型不同时，各节点类型支持的节点数量不同，扩容节点数量时请参考[表2-117](#)。

表 2-117 集群的节点数量取值说明

一个集群包含的节点类型	节点数量的取值范围
ess	ess: 1~32
ess、ess-master	ess: 1~200 ess-master: 3~9的奇数
ess、ess-client	ess: 1~32 ess-client: 1~32
ess、ess-cold	ess: 1~32 ess-cold: 1~32
ess、ess-master、ess-client	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32
ess、ess-master、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-cold: 1~32
ess、ess-client、ess-cold	ess: 1~32 ess-client: 1~32 ess-cold: 1~32
ess、ess-master、ess-client、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32 ess-cold: 1~32
说明	<ul style="list-style-type: none">● ess: 数据节点，即创建集群时必配的节点类型，其他3种节点类型都是基于业务需要可选的类型。● ess-master: Master节点。● ess-client: Client节点。● ess-cold: 冷数据节点。

前提条件

- 确认集群处于“可用”状态，且无正在进行的任务。
- 确认CSS服务有足够的配额支持集群扩容。

扩容节点数量和节点存储容量

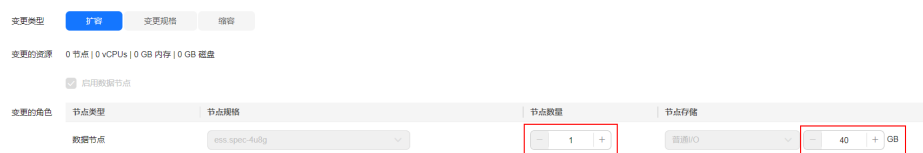
1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。

4. 选择“更改集群规格”页签。
5. 在更改集群规格页面，设置扩容参数。

表 2-118 扩容节点数量和节点存储容量

参数	说明
变更类型	选择“扩容”。
变更的资源	显示本次扩容的资源变化量。
变更的角色	<p>在“节点数量”列和“节点存储”列增加节点数量和节点存储容量。支持多种节点类型同时变更。</p> <ul style="list-style-type: none">● 节点数量的取值范围请参见表2-117。● 节点存储容量的取值范围由“节点规格”决定，且只支持配置为20的倍数。 <p>说明 包周期集群不支持同时扩容“节点数量”和“节点存储容量”。</p>

图 2-26 集群扩容



6. 单击“下一步”。
7. 确认变更信息后，单击“提交申请”。
8. 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列显示为“扩容”，表示集群正在扩容。当集群状态变为“可用”，则表示扩容成功。

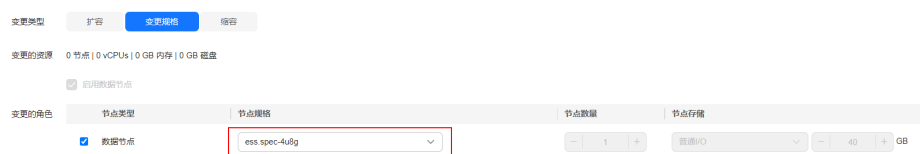
变更节点规格

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 确认业务数据是否都有副本，确保变更规格过程中不会中断业务。
 - a. 在集群列表选择目标集群，单击操作列的Kibana，登录Kibana。
 - b. 选择左侧导航栏的“Dev Tools”，进入操作页面。
 - c. 在Kibana中执行命令`GET _cat/indices?v`。
 - 当返回的“rep”值大于“0”时，则表示有副本，请继续执行下一步。
 - 当返回的“rep”值等于“0”时，则表示没有副本，建议先为集群[手动创建快照](#)再变更规格。
4. 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
5. 选择“更改集群规格”页签。
6. 在更改集群规格页面，设置变更规格的参数。

表 2-119 变更节点规格

参数	说明
变更类型	选择“变更规格”。
变更的资源	显示本次变更规格的资源变化量。
变更的角色	先在“节点类型”列选择本次要变更的节点类型，只能选择一种类型。再在“节点规格”列选择新规格。

图 2-27 集群变更规格



- 单击“下一步”。
- 确认变更信息后，单击“提交申请”。
- 在弹出的窗口确认是否勾选“进行索引副本校验”和“检测集群状态”，单击“确认”启动集群规格变更。
 - 索引副本校验：索引副本校验用于保证索引正常提供服务。如果跳过校验，变更规格操作将不会要求每一个索引都有副本，变更过程中可能会影响业务。
 - 没有Master节点的集群更改节点规格时，如果选择进行索引副本校验，则要求所有索引至少有1个副本，且“节点数量”总和不小于3。
 - 有Master节点的集群更改节点规格时，如果选择进行索引副本校验，则要求所有索引至少有1个副本。
 - 检测集群状态：默认检测集群状态，变更规格为滚动变更，变更过程中，为了保证变更成功率以及数据安全，会确保每个节点进程拉起来后继续后续节点操作。当集群负载过高业务故障，无法正常下发变更请求，依赖更多的资源才能恢复的紧急情况下可忽略检测集群状态，忽略后，变更过程中因为忽略集群状态检测可能会导致集群故障并中断业务，请谨慎跳过。
- 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“规格修改”，表示集群正在更改规格。当集群状态变为“可用”，则表示规格变更成功。

添加 Master 或 Client 节点

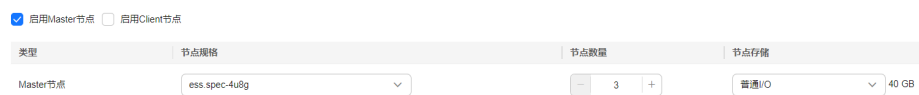
- 登录云搜索服务管理控制台。
- 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
- 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
- 选择“添加Master/Client节点”页签。

当集群同时拥有Master节点和Client节点时，形态变更页面将不存在“添加Master/Client节点”页签。
- 在添加Master/Client节点页面，勾选需要添加的节点，并配置“节点规格”、“节点数量”和“节点存储”。

表 2-120 添加 Master 或 Client 节点

参数	说明
启用节点类型	勾选本次要添加的节点类型。 <ul style="list-style-type: none">只能选择一种节点类型，当需要同时添加Master和Client节点时，就需要执行2次“添加Master/Client节点”任务。当集群本身已经有Master节点或Client节点时，则此处只显示另一种节点类型供选择。
节点规格	根据需求选择节点规格。
节点数量	设置该节点类型的节点数量，取值范围请参见表 2-117。
节点存储	设置该节点类型的存储类型和容量，节点存储容量的取值范围由“节点规格”决定，且只支持配置为20的倍数。

图 2-28 添加 Master/Client 节点



- 单击“下一步”。
- 确认变更信息后，单击“提交申请”。
返回集群列表页面，集群的“任务状态”列中显示为“扩容中”。
 - 如果添加Master节点，当“集群状态”变为“可用”时表示节点添加成功。

须知

如果集群版本小于7.x，当“集群状态”变为“可用”时，还需要重启集群的所有数据节点和冷数据节点，使新增节点的配置生效。如果不重启使新增节点配置生效，使用集群时因为上报机制问题可能显示不可用状态（集群业务并未不可用）。重启操作请参见[重启Elasticsearch集群](#)。

- 如果添加Client节点，当“集群状态”变为“可用”时表示节点添加成功。可选择是否重启数据节点和冷数据节点，下线数据节点上的Cerebro和Kibana的进程。

2.10.2 缩容 Elasticsearch 集群

当集群有充足的能力处理当前数据时，为节省资源可以参考本文缩容集群的节点数量。

场景描述

CSS服务支持多种缩容场景，如表2-116所示。

表 2-121 缩容场景

场景	描述	相关文档
缩容随机节点	缩容随机节点是基于节点类型随机下线节点减少集群的“节点数量”，减少集群占用的资源。	缩容随机节点
缩容指定节点	缩容指定节点是通过选择下线具体哪个节点减少集群的“节点数量”，减少集群占用的资源。	缩容指定节点

约束限制

- 缩容是通过下线节点减少节点数量，为了降低业务影响，建议在业务低峰期进行缩容操作。
- 缩容过程会涉及数据迁移，将要下线的节点数据迁移到其他节点上，数据迁移的超时阈值为5小时。当超过5小时数据还未迁移完成，那么缩容会失败。建议在集群数据量较大的情况下，分多次进行缩容。
- 要确保缩容之后的磁盘使用量小于80%，且集群每个节点类型中每个AZ的节点数至少为1。
- 关于跨AZ的集群，在不同AZ中同类型节点个数的差值要小于等于1。
- 当集群没有Master节点时，每次缩容的数据节点和冷数据节点个数之和要小于缩容前数据节点和冷数据节点个数之和的一半，缩容后的数据节点和冷数据节点个数之和要大于索引的最大副本个数。
- 当集群有Master节点时，每次缩容的Master节点个数要小于当前Master节点总数的一半，缩容后的Master节点个数必须是奇数且不小于3。
- 当集群没有Master节点时，数据节点数量和冷数据节点数量之和大于等于3时才支持缩容。当集群有Master节点时，数据节点数量大于等于2时才支持缩容。当集群不满足缩容条件却想减少集群节点时，建议新建集群，[备份与恢复Elasticsearch 集群数据](#)迁移集群数据。
- 当一个集群包含的节点类型不同时，各节点类型支持的节点数量不同，缩容节点数量时请参考[表2-122](#)。

表 2-122 集群的节点数量取值说明

一个集群包含的节点类型	节点数量的取值范围
ess	ess: 1~32
ess、ess-master	ess: 1~200 ess-master: 3~9的奇数
ess、ess-client	ess: 1~32 ess-client: 1~32
ess、ess-cold	ess: 1~32 ess-cold: 1~32

一个集群包含的节点类型	节点数量的取值范围
ess、ess-master、ess-client	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32
ess、ess-master、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-cold: 1~32
ess、ess-client、ess-cold	ess: 1~32 ess-client: 1~32 ess-cold: 1~32
ess、ess-master、ess-client、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32 ess-cold: 1~32
说明 <ul style="list-style-type: none"> • ess: 数据节点，即创建集群时必配的节点类型，其他3种节点类型都是基于业务需要可选的类型。 • ess-master: Master节点。 • ess-client: Client节点。 • ess-cold: 冷数据节点。 	

前提条件

集群处于“可用”状态，且无正在进行的任务。

缩容随机节点

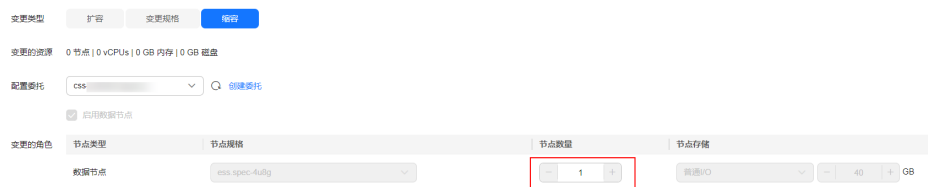
1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
4. 选择“更改集群规格”页签。
5. 在更改集群规格页面，设置缩容参数。

表 2-123 缩容随机节点

参数	说明
变更类型	选择“缩容”。
变更的资源	显示本次缩容的资源变化量。

参数	说明
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平面时才显示该参数。</p> <ul style="list-style-type: none">删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。
变更的角色	<p>在“节点数量”列减少节点数量。支持多种节点类型同时变更。</p> <p>节点数量的取值范围请参见表2-122。</p>

图 2-29 集群缩容



- 单击“下一步”。
- 确认变更信息后，单击“提交申请”。
- 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“缩容中”，表示集群正在缩容。当集群状态变为“可用”，则表示缩容成功。

缩容指定节点

- 登录云搜索服务管理控制台。
- 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
- 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
- 选择“缩容指定节点”页签。
- 在缩容指定节点页面，设置缩容参数。

表 2-124 缩容指定节点

参数	说明
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平面时才显示该参数。</p> <ul style="list-style-type: none">删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。
节点类型	展开本次要变更的节点类型，此处会显示当前节点类型拥有的节点，勾选要下线的节点。

- 单击“下一步”。
- 确认变更信息后，单击“提交申请”，在数据迁移弹窗中确认是否进行数据迁移（建议勾选“进行数据迁移”避免数据丢失），单击“确认”提交申请。

数据迁移过程中，系统会把待下线节点中的所有数据分片移动到剩余节点，待数据迁移完成后再进行节点下线操作。当待下线节点上的数据在剩余节点存在副本时，可跳过数据迁移，缩短集群变更时间，减少数据迁移带来的负载压力。
- 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“缩容中”，表示集群正在缩容。当集群状态变为“可用”，则表示缩容成功。

2.11 升级 Elasticsearch 集群版本

Elasticsearch集群的版本升级功能支持同版本升级、跨版本升级和跨引擎升级。

场景描述

升级场景

- 同版本升级是升级集群的内核补丁，用于修复问题或优化性能。
- 跨版本升级是升级集群的版本，用于功能加强或版本收编。
- 跨引擎升级是支持Elasticsearch集群升级至OpenSearch集群。

升级原理

升级过程采用的是one-by-one的方式，不会中断业务。升级时，先下线一个节点，将该节点的数据迁移到其余节点，再创建一个目标版本的新节点，将已下线节点的网卡port挂载到新节点，以此保留节点IP地址，待新节点加入到集群后，再依次将其余节点进行替换。当集群的数据量很大时，升级耗时将依赖数据迁移耗时。

升级流程

- 步骤1 进行升级前检查：[升级前检查](#)**

升级前检查大部分支持系统检查，少部分需要人工检查。

步骤2 创建快照备份全量索引数据：[手动创建快照](#)

升级配置中支持选择进行全量索引快照备份检测，避免升级失败数据丢失。

步骤3 创建升级任务，启动升级：[创建升级任务](#)

----结束

版本升级能力

待升级的集群版本不同，支持升级的目标版本也不同，具体请参见[表2-125](#)。

表 2-125 版本升级能力

当前版本	目标版本
Elasticsearch: 6.2.3	Elasticsearch: 6.5.4, 6.8.23
Elasticsearch: 6.5.4	Elasticsearch: 6.8.23
Elasticsearch: 6.8.23	Elasticsearch: 7.6.2, 7.10.2
Elasticsearch: 7.1.1	Elasticsearch: 7.6.2, 7.10.2
Elasticsearch: 7.6.2	Elasticsearch: 7.10.2
Elasticsearch: 7.9.3	Elasticsearch: 7.10.2
Elasticsearch: 7.10.2	OpenSearch: 1.3.6

说明：

- ELasticsearch集群以7.6.2 和7.10.2为主力版本，建议其他版本的集群收编到该版本。实际支持的目标版本请以升级页面中“目标镜像”的可选值为准。
- 5.x版本的ELasticsearch集群不支持跨版本升级，6.2.3和6.5.4版本Elasticsearch集群可先升级到6.8.23版本，再升级到7.x.x版本。
- 当前仅支持7.10.2版本Elasticsearch集群跨引擎升级至1.3.6版本OpenSearch集群。

约束限制

- 最多同时支持20个集群升级，建议在业务低峰期进行升级操作。
- 待升级的集群不能存在正在进行中的任务。
- 升级任务一旦启动就无法中止，直到升级任务的“任务状态”显示“失败”或“成功”才结束。
- 升级过程中，存在节点替换的过程，此时如果有请求发送到替换中的节点，请求可能会失败。为防止该情况出现，建议通过终端节点服务或者独享型负载均衡器接入集群进行访问。
- 升级过程中会重建Kibana和Cerebro组件，重建过程中，Kibana和Cerebro可能会访问失败。而且，由于不同Kibana版本相互不兼容，升级过程中Kibana还可能因为版本不兼容而无法访问。访问失败的问题，待集群升级成功后会自动恢复。

升级前检查

为了保证升级成功，需要做升级前检查，升级前检查主要包括如下事项：

表 2-126 升级前检查项

检查项	检查方式	描述	正常状态
集群状态	系统检查	升级任务启动后，系统会自动检查集群状态。集群状态为green或者yellow，表示集群可以正常提供服务，不存在未分配的主分片。	“集群状态”为“可用”。
节点个数	系统检查	升级任务启动后，系统会自动检查节点个数。为了保证业务的连续性，集群中数据节点和冷数据节点的个数之和要不小于3。	集群的数据节点和冷数据节点个数之和 ≥ 3
磁盘容量	系统检查	升级任务启动后，系统会自动检查磁盘容量。升级过程中会逐一下线节点再新建节点，需要保证单节点下线后，其余节点的磁盘容量能够接纳该节点的所有数据。	下线单节点后，剩余节点能够包含集群的所有数据。
数据副本	系统检查	检测集群中索引的最大主备分片数是否能够分配到剩余数据节点和冷数据节点中，防止升级过程中出现节点下线后副本无法分配的情况。	索引的主备分片数的最大值 $+1 \leq$ 升级前的数据节点数和冷数据节点之和
数据备份	系统检查	升级前要做好数据备份，防止升级故障数据丢失。在提交升级任务时，可以选择是否需要系统检查全量索引的快照备份。	确认是否存在数据备份。
资源充足	系统检查	升级任务启动后，系统会自动检查资源。升级过程中会新建资源，需要保证有资源可用。	资源可用且配额充足。

检查项	检查方式	描述	正常状态
自定义插件	系统检查+人工检查	<p>历史版本的集群装有自定义插件才需要进行该项检查。如果集群装有自定义插件，则需要在插件管理界面上上传目标版本的所有插件包，升级过程中才能将自定义插件转入新节点中，否则集群升级成功后，自定义插件将丢失。升级任务启动后，系统会自动检查是否已经上传自定义插件包，但是上传的插件包是否正确需要人工检查。</p> <p>说明</p> <p>如果上传的插件包不正确或者不兼容将导致升级过程中无法自动安装插件包，升级任务会失败。如果需要恢复集群，可以终止升级任务，执行指定节点替换操作，修复当前升级失败的节点。</p> <p>升级完成后，自定义插件对应的“插件状态”将会重置为“已上传”状态。</p>	装有自定义插件的集群已经把对应版本的插件包上传到升级插件列表中。
自定义配置	系统检查	升级过程中，系统会自动同步集群配置文件“elasticsearch.yml”中的内容。	如果集群有自定义的参数配置，升级完成后，配置都未丢失。
非标操作	人工检查	确认是否存在非标操作。非标操作指的是没有被记录下来的手动操作，这些操作在升级过程中无法自动传递，比如“Kibana.yml”配置文件修改、系统配置、回程路由等。	有些非标操作是兼容的，例如安全插件的修改，可以通过元数据保留下来；系统配置修改，可以通过镜像的操作保留下来。但是有些非标操作例如“kibana.yml”文件修改就无法保留，需要提前备份。
兼容性检测	系统检查+人工检查	跨版本升级的升级任务启动后，系统会自动检测集群升级前后两个版本是否有不兼容配置。如果集群装有自定义插件，则自定义插件的版本兼容性需要人工检测。	跨版本升级时，升级前后没有不兼容的配置。

检查项	检查方式	描述	正常状态
集群负载检测	系统检查 +人工检查	当集群负载过高时，升级任务大概率会卡住或失败，建议升级前进行集群负载检测，选择在低负载的时候进行升级。 升级配置中也支持选择进行“集群负载检测”。	<ul style="list-style-type: none"> 最大search队列排队数小于1000 (nodes.thread_pool.search.queue < 1000) 最大write队列排队数小于200 (nodes.thread_pool.write.queue < 200) 最大cpu使用率小于90% (nodes.process.cpu.percent < 90) 最大load相对核数占比小于80% (nodes.os.cpu.load_average / cpu核数 < 80%)

创建升级任务

1. 登录云搜索服务管理控制台。
2. 左侧导航栏选择“集群管理”，进入集群列表页面，单击目标集群名称，进入集群基本信息页面。
3. 选择“版本升级”。
4. 在升级页面，配置升级参数。

表 2-127 升级参数说明

参数	描述
升级类型	<ul style="list-style-type: none"> 同版本升级：升级集群的内核补丁，集群版本号不变。 跨版本升级：升级集群的版本，集群版本号升级。 跨引擎升级：Elasticsearch集群升级至OpenSearch集群，当前仅支持7.10.2版本Elasticsearch集群升级至1.3.6版本OpenSearch集群。
目标镜像	<p>选择目标版本的镜像。选中镜像后，下方会显示镜像名称和目标版本的详细说明。</p> <p>实际支持的目标版本请以升级页面中“目标镜像”的可选值为准。如果无法选择目标镜像，有如下几个原因：</p> <ul style="list-style-type: none"> 当前集群已是最新版本集群。 当前集群是23年之前创建的旧集群，且存在向量索引。 当前局点暂未录入新版本镜像。

参数	描述
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平时才显示该参数。</p> <ul style="list-style-type: none">删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。

5. 配置完成后，单击“确认提交”。确认是否进行“全量索引快照备份检测”和“集群负载检测”，单击“确定”启动升级任务。

当集群负载过高时，升级任务大概率会卡住或失败。增加“集群负载检测”可以有效避免失败。

检测中如果出现以下四种情况，请等待或者主动降低负载。如果需求紧急且已了解升级失败风险，也可以不做“集群负载检测”。“集群负载检测”检测项包含：

- 最大search队列排队数小于1000（`nodes.thread_pool.search.queue < 1000`）
- 最大write队列排队数小于200（`nodes.thread_pool.write.queue < 200`）
- 最大cpu使用率小于90%（`nodes.process.cpu.percent < 90`）
- 最大load相对核数占比小于80%（`nodes.os.cpu.load_average / cpu核数 < 80%`）

6. 在“任务记录”列表，显示当前升级任务。当“任务状态”为“运行中”时，可以展开任务列表，单击“查看进度”查看详细的升级进度。

当“任务状态”为“失败”时，可以重试任务或者直接终止任务。

- 重试升级：在任务列表的操作列，单击“重试”，重新升级。
- 终止升级：在任务列表的操作列，单击“终止”，结束升级。

须知

- 同版本升级：当升级的“任务状态”为“失败”即可终止升级任务。
- 跨版本升级：当升级的“任务状态”为“失败”时，且没有任何节点升级成功的，才可以终止升级任务。

当升级任务终止后，集群的“任务状态”将清空“升级失败”的状态，相当于集群回退到升级前状态，不影响集群进行其他任务。

2.12 管理 Elasticsearch 集群

2.12.1 查看 Elasticsearch 集群信息

在Elasticsearch/OpenSearch集群的列表页和基本信息页，可以获取集群的状态、版本、节点等信息。

集群列表介绍

集群列表会展示当前账号下同一集群类型的所有集群。当集群数量较多时，会翻页显示，支持查看任何状态下的集群。

集群列表默认按时间顺序排列，时间最近的集群显示在最前端。

在集群列表页单击“导出”可以下载全量的集群列表信息。

表 2-128 集群列表说明


参数	描述
名称/ID	展示集群的名称和ID。单击集群名称可进入集群“基本信息”页面，展现了集群的基本信息。集群ID是系统自动生成的，是集群在服务中的唯一标示。
集群状态	展示集群的运行状态。 <ul style="list-style-type: none">● 可用：正常运行的集群。● 创建中：正在创建中的集群。● 处理中：重启中、扩容中、备份中或恢复中的集群。● 异常：创建失败或不可用的集群。 说明 当集群处于“不可用”状态时，支持删除集群，也支持将集群正常状态时创建的快照恢复至其他集群。但是无法扩容集群、访问Kibana、创建快照或将快照恢复至此集群。建议不要执行导入数据的操作，避免数据丢失。您可以查看监控或重启集群处理异常，当依旧执行失败时，请及时联系技术支持。
任务状态	展示重启集群、扩容集群、缩容集群等任务的状态。
版本	展示集群版本号和镜像版本号。 说明 当集群的创建时间比较早时，数据库可能未记录其镜像信息，导致页面无法显示集群的镜像版本号。
创建时间	展示集群的创建时间。
企业项目	展示集群所归属的企业项目。
内网访问地址	展示集群的内网访问地址和端口号，您可以使用此参数接入集群。集群有多个节点时，此处显示多个节点的内网访问地址和端口号。
计费模式	展示集群的计费模式，是“按需计费”还是“包年/包月”。
操作	展示集群可执行的操作入口，包含监控信息、重启、删除等其他更多操作。当某一操作无法执行时，按钮将置灰。

查看集群基本信息

在集群的基本信息页面，可以获取集群的内网访问地址、公网访问地址、版本、节点等信息。

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群管理页面。
3. 在集群列表，单击集群名称进入集群“基本信息”页面，查看集群的基本信息。

表 2-129 基本信息的参数说明

类别	参数	描述
基本信息	集群名称	集群名称。支持自定义名称。 单击右侧  可以修改集群名称。
	ID	集群的唯一标识，是系统自动生成的。 同一个区域下，集群ID是唯一的。
	集群版本	集群的版本信息。 如果需要升级集群版本，请参见 升级Elasticsearch集群版本 。
	集群状态	集群当前的状态。
	任务状态	集群当前的任务状态，如果没有进行中的任务则显示“--”。
	创建时间	集群创建的时间。
	集群存储容量 (GB)	集群设置的存储容量。
	集群存储使用量 (GB)	集群已使用的存储容量。
配置信息	区域	集群所在区域。
	可用区	集群所在的可用区。
	虚拟私有云	集群所属的虚拟私有云。
	子网	集群所属的子网。
	安全组	集群所属的安全组。 单击右侧的“更改安全组”可以修改集群的安全组信息。 须知 进行安全组切换前，请确保业务接入时需要的9200端口已经放开，错误的安全组配置可能会导致业务无法访问，请谨慎操作。

类别	参数	描述
	安全模式	<p>集群的安全模式。</p> <ul style="list-style-type: none">• 启用：表示当前集群是安全模式的集群。• 未开启：表示当前集群是非安全集群。 <p>如果需要切换集群的安全模式，请参见更改Elasticsearch集群安全模式。</p>
	重置密码	<p>仅安全模式的集群显示。</p> <p>单击“重置”可以修改安全集群的管理员账户admin的密码。</p> <p>说明</p> <p>管理员密码的规则：</p> <ul style="list-style-type: none">• 可输入的字符串长度为8~32个字符。• 密码至少包含大写字母、小写字母、数字和特殊字符四类中的三类。其中支持的特殊字符有“~!@#%&*()-_+=\ []{};,<.>/?”。• 不能与管理员账户名或倒序的管理员账户名相同。• 建议定期修改密码。
	企业项目	<p>集群所属的企业项目。</p> <p>单击项目名称可以跳转到项目管理页面查看企业项目的基本信息。</p>
	公网访问	<p>公网访问信息，仅安全模式的集群显示。</p> <ul style="list-style-type: none">• 启用公网访问的安全集群，此处显示公网访问地址，通过该地址可以在公网访问安全集群。• 未启用公网访问的安全集群，此处显示“--”。 <p>使用公网地址访问集群时，建议开启访问控制并配置访问白名单，提高集群安全性。公网访问配置的详细信息，请参见配置公网访问。</p>
	访问控制	<p>启用公网访问的集群是否设置访问控制，仅启用公网访问的集群显示。</p> <ul style="list-style-type: none">• 已开启：启用访问控制开关，只允许白名单列表中的IP地址通过公网访问集群。• 未开启：为启用访问控制开关，允许任何IP地址通过公网访问集群。 <p>单击“设置”，可以更新访问控制开关和白名单。</p>

类别	参数	描述
	带宽	公网访问的带宽，仅启用公网访问的集群显示。 单击“修改”，可以更新带宽大小。
	HTTPS访问	集群是否启用HTTPS访问协议。 <ul style="list-style-type: none">关闭：表示集群未启用HTTPS访问，集群使用HTTP访问协议。开启：表示集群启用了HTTPS访问协议，仅安全模式的集群支持开启HTTPS访问。启用HTTPS访问的安全集群可以单击“下载证书”获取CER安全证书，用于接入安全模式的集群。安全证书暂不支持在公网环境下使用。 如果安全模式的集群需要切换访问方式，请参见 切换安全模式下的协议 。
	内网访问IPv4地址	集群的内网IP地址和端口号，使用此参数可以接入集群。如果集群只有一个节点，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；如果集群有多个节点，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。
节点信息	节点规格	集群中节点的规格信息。
	节点存储	集群中节点的存储容量和存储类型。
	节点数量	集群中节点的个数。

2.12.2 创建 Elasticsearch 集群用户并授权使用集群

CSS服务会对开启安全模式的集群进行访问控制，只有通过身份认证的用户才能访问安全集群。在创建安全集群时，必须配置一个管理员账号。该管理员能够使用Kibana为集群添加新用户，并授予相应的安全集群使用权限。本文将以7.6.2版本的Kibana为例，指导如何使用Kibana为安全集群配置用户授权。

背景信息

云搜索服务用opendistro_security安全插件对外提供安全集群能力，opendistro_security安全插件是基于RBAC（Role-Based Access Control）模型构建。RBAC包括三个重要核心概念：用户（User）、权限（Action）、角色（Role）。RBAC简化了用户和权限的关系，降低了权限管理的难度，方便权限扩展易于维护。三者之前的关系如[图2-30](#)所示。

图 2-30 用户、权限和角色



表 2-130 概念介绍

参数	描述
User	用户：可以向Elasticsearch集群发出操作请求。用户具有凭证（例如，用户名和密码）、零个或多个后端角色以及零个或多个自定义属性。
Role	角色：定义为权限或操作组的组合，包括对集群，索引，文档或字段的操作权限。
Permission	权限：单个动作，例如创建索引（例如indices:admin/create）。
Role mapping	角色映射：用户在成功进行身份验证后会担任角色。角色映射，就是将角色映射到用户（或后端角色）。例如，kibana_user（角色）到jdoe（用户）的映射意味着John Doe在获得kibana_user身份验证后获得了所有权限。同样，all_access（角色）到admin（后端角色）的映射意味着具有后端角色admin（来自LDAP / Active Directory服务器）的任何用户都获得了all_access身份验证后的所有权限。您可以将每个角色映射到许多用户和/或后端角色。
Action group	操作组：一组权限。例如，预定义的SEARCH操作组授权角色使用_search和_msearchAPI。

除了RBAC模型，Elasticsearch还有Tenant概念。RBAC用于解决用户级别的授权问题，而Tenant则处理不同租户间的信息共享问题。通过配置Tenant空间，租户能够在该空间内共享Dashboard、index_pattern等信息。

默认情况下，用户仅能查看其Private Tenant空间中的index_pattern、dashboard等信息。当添加新用户“test”时，系统会自动生成一个名为“.kibana_xxx_test”的索引，该用户Private空间的内容将存储于此。同理，管理员账号的Private Tenant空间内容则存储在“.kibana_xxx_admin”索引中。当需要与其他租户共享index_pattern或Dashboard信息时，可以选择在Global Tenant空间中创建它们，其他用户只需切换至Global Tenant，即可访问共享信息。

在Kibana界面的Security菜单中，您可以控制用户在Elasticsearch集群中的权限，实现集群、索引、文档和字段四个级别的细粒度访问权限控制。

支持添加或删除集群的用户，并能够将用户映射到角色，实现用户关联角色权限。

角色映射功能允许配置角色的成员，通过用户名、后端角色和主机名将用户分配给相应的角色。支持为每种角色配置集群访问权限、索引和文档的访问权限，以及Kibana的使用权限。

有关安全集群的更多安全配置信息以及详细的操作指导，可以参考Elasticsearch的[安全模式官方介绍](#)。

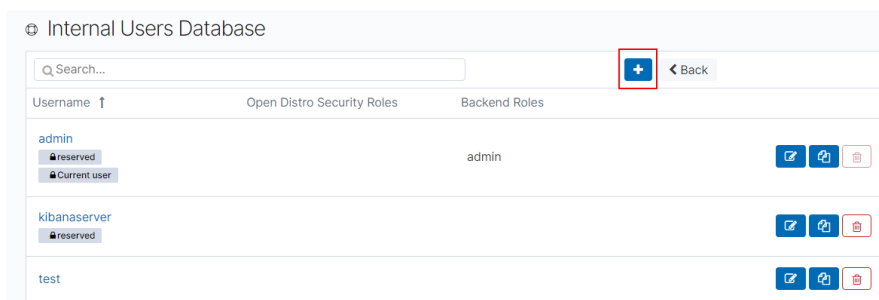
约束限制

- Kibana中可以自定义用户名、角色名、租户名等，但不支持中文字符。
- 不同版本的Kibana界面可能有所差异，请以实际操作环境为准。本文以Kibana 7.6.2版本作为示例。

操作步骤

1. 登录云搜索服务控制台。
2. 在集群管理列表，选择对应集群，单击操作列的“Kibana”。
3. 使用管理员账号登录Kibana页面。
 - 账户名：admin（默认管理员账户名）
 - 密码：创建安全模式的集群时，设置的管理员密码。
4. 登录成功后，在Kibana操作界面的左侧导航栏选择“Security”，进入“Security”页面。
5. 给安全集群添加一个新用户“test”。
 - a. 选择“Authentication Backends” > “Internal Users Database”，进入创建用户页面。
 - b. 在“Internal Users Database”页面，单击“+”，进入添加用户信息页面。

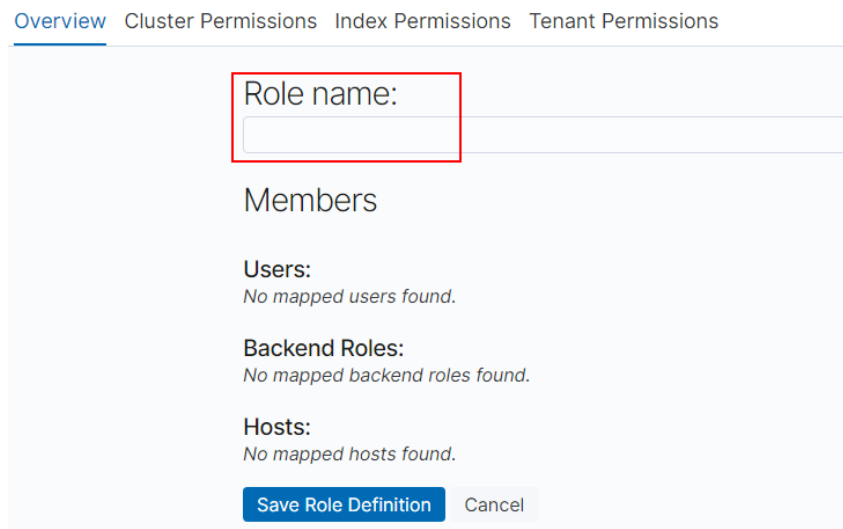
图 2-31 添加用户



- c. 在创建用户页面，输入“Username”、“Password”和“Repeatpassword”，单击“Submit”。

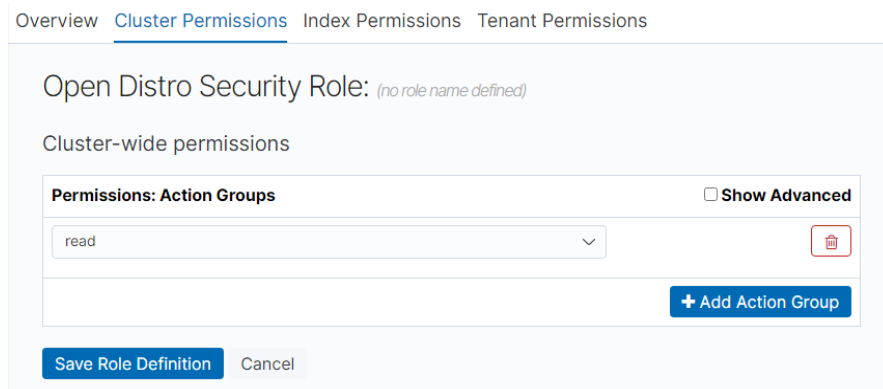
此处以用户名“test”为例。创建成功后，可以在列表中看到新创建的用户。
6. 创建角色“role_test”，并为角色授权。
 - a. 在“Security”中选择“Roles”，进入Open Distro Security Roles页面。
 - b. 在Open Distro Security Roles页面，单击“+”添加角色权限。
 - c. 在Overview页签设置角色名“Role name”，例如“role_test”。

图 2-32 添加角色名称



- d. 在“Cluster Permissions”页签设置集群权限。根据业务需要选择相应的集群权限，不配置时表示角色不具有任何集群级别的权限。
- “Permissions: Action Groups”：单击“Add Action Group”，可以设置集群权限。例如，集群只读权限选择read，表示仅可查看集群状态、集群节点等信息。
 - “Permissions: Single Permissions”：勾选Show Advanced后，单击“Add Single Permission”可以针对集群设置更精细的权限。例如设置为“indices:data/read”，表示仅指定索引的读取权限。

图 2-33 Cluster Permissions 页面



- e. 在“Index Permissions”页面设置索引权限。
- “Index patterns”：单击“Add index permissions”配置为需要设置权限的索引名称，例如，索引模板名称为“my_store”。

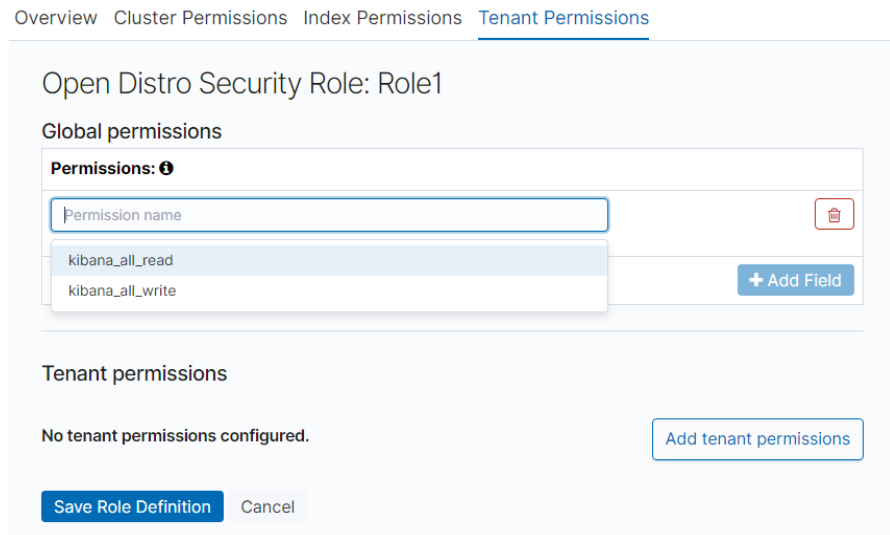
说明

建议索引名称和创建的用户名不要相同。

- “Permissions: Action Groups”：单击“Add Action Group”，根据需要开通的权限设置。例如，只读权限选择“Search”。

- f. “Tenant Permissions” 页面设置租户权限。
 - “Global permissions”：单击 “Add Field”，设置角色的kibana读写权限，例如 “kibana_all_read” 或 “kibana_all_write” 权限。
 - “Tenant permissions”：单击 “Add tenant pattern”，自行添加租户模式，并且为新增的租户模式设置 “kibana_all_read” 或 “kibana_all_write” 权限。

图 2-34 Tenant Permissions 页面



- g. 单击 “Save Role Definition”，保存角色设置，即可看到设置的角色。
7. 将角色映射到用户，使用户具有角色的权限。
- a. 在 “Security” 中选择 “Role Mappings”，进入Role Mappings页面，进行角色映射。
 - b. 在Role Mappings页面，单击 “+”，选择角色 “Role”，添加用户 “Users” 。
 - “Role”：选择待映射的角色名称，例如 “role_test”。
 - “Users”：单击 “Add User”，选择被映射角色的用户名称，例如 “test”。

图 2-35 用户和角色映射

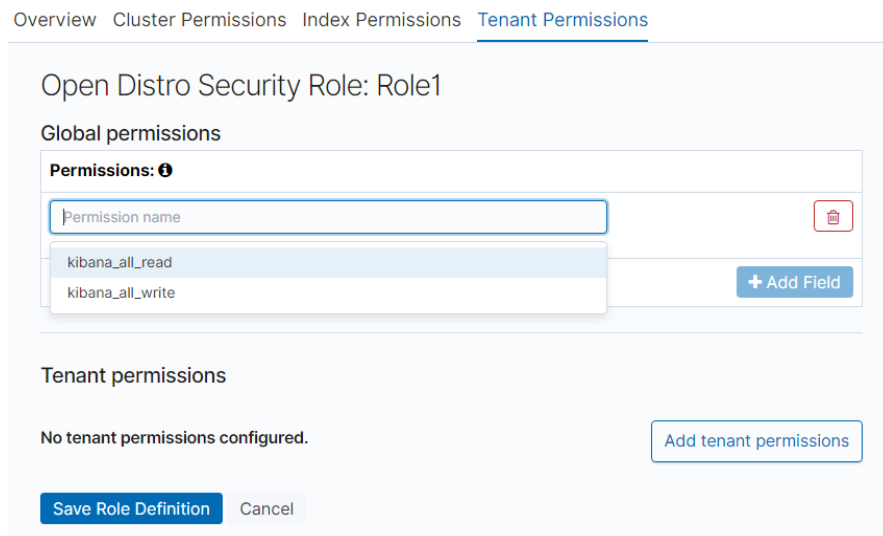
The screenshot shows a configuration form for user and role mapping. It includes a 'Role' dropdown, a 'Users' table with a 'username' entry and an '+ Add User' button, a 'Backend roles' section with an '+ Add Backend Role' button, and a 'Hosts' section with an '+ Add Host' button. At the bottom, there are 'Submit' and 'Cancel' buttons.

- c. 添加完成后，单击“Submit”。
8. 配置完成后，可以使用“test”用户登录Kibana，验证用户权限是否生效。

典型示例：通过管理员账户给用户分配 Kibana 的访问权限

1. 使用管理员账户admin登录安全集群。
2. 在“Security”中选择“Roles”，进入Open Distro Security Roles页面。
3. 在“Open Distro Security Roles”页面，单击“+”添加角色权限。
4. 在“Overview”页签，设置角色名。例如“kibana_user”。
5. 在“Tenant Permissions”页签设置租户权限。
 - “Global permissions”：单击“Add Field”，设置角色的kibana读写权限，例如“kibana_all_read”或“kibana_all_write”权限。
 - “Tenant permissions”：单击“Add tenant pattern”，自行添加租户模式，并且为新增的租户模式设置“kibana_all_read”或“kibana_all_write”权限。

图 2-36 Tenant Permissions 页面

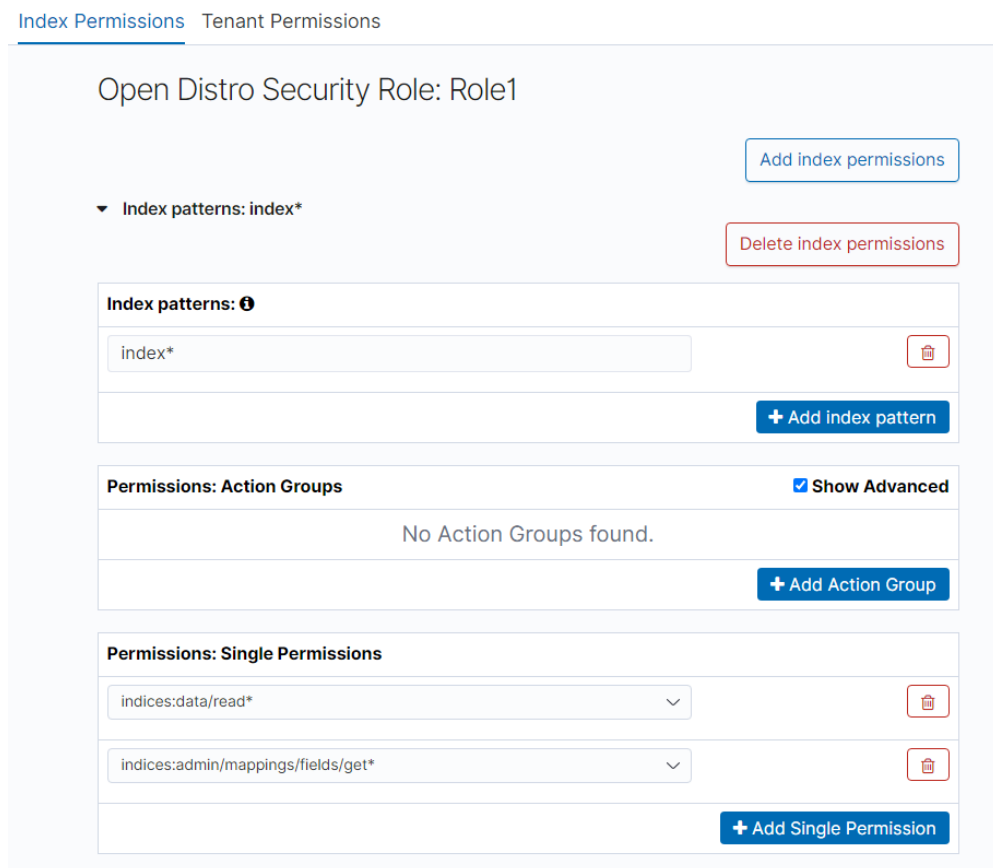


6. 单击“Save Role Definition”，保存角色设置，即可看到设置的角色“kibana_user”。
7. 在Security中选择“Role Mappings”，进入Role Mappings页面。
8. 在“Role Mappings”页面，单击“+”添加用户“test”和角色“kibana_user”映射。
9. 添加完成后，单击“Submit”。
配置完成后，“test”用户即可拥有Kibana的读写权限。

典型示例：通过管理员账户给用户分配指定索引“index*”的Read权限

1. 使用管理员账户admin登录安全集群。
2. 在“Security”中选择“Roles”，进入Open Distro Security Roles页面。
3. 在“Open Distro Security Roles”页面，单击“+”添加角色权限。
4. 在“Overview”页签，设置角色名。例如“Role1”。
5. 在“Index Permissions”页签，单击“Add index permissions”设置索引权限。

图 2-37 设置权限



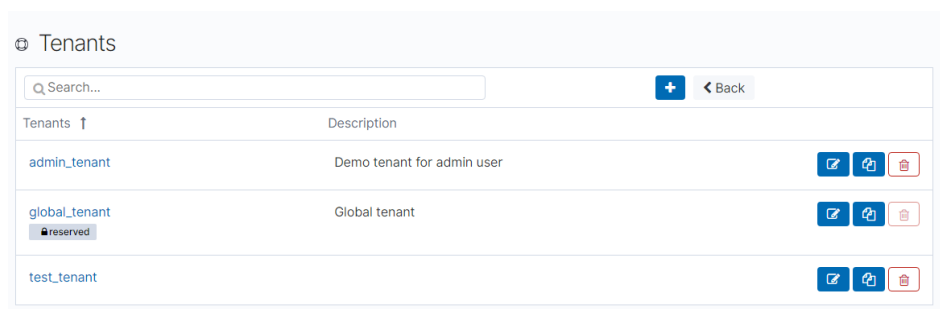
- “Index patterns”：输入“index*”，表示需要为所有index开头的索引设置权限。
 - “Permissions: Action Groups”：选择需要的权限。当需要查询权限时，选择“read”，当需求写入权限时，选择“write”。具体的操作对应着底层的action，可以在Kibana界面的permission模块获得详细介绍。此处以“read”为例，则选择“indices:data/read*”和“indices:admin/mappings/fields/get*”权限。“indices:data/read*”包含“indices:data/read/”下所有的权限，比如“indices:data/read/get”、“indices:data/read/mget”、“indices:data/read/search”等。
6. 单击“Save Role Definition”，保存角色设置，即可看到设置的角色“Role1”。
 7. 在Security中选择“Role Mappings”，进入Role Mappings页面。
 8. 在“Role Mappings”页面，单击“+”添加用户“test”和角色“Role1”映射。
 9. 添加完成后，单击“Submit”。
- 配置完成后，“test”用户即可拥有“index*”的Read权限。

典型示例：通过管理员账户给用户分配其他 Tenant 的访问权限

新添加的test用户可以访问Kibana，并且可以使用Kibana的index_pattern、Discover、Dashboards等权限，但是这并不意味着test用户能看到任意的“.kibana”信息。test用户默认只能看到Private Tenant和Global Tenant空间的数据，如果需要访问其他用户的Tenant空间，则需要通过管理员账户在test用户的Role中定义其他空间的index_pattern、dashboard等信息的访问权限。

1. 使用管理员账户admin在集群的Global Tenant中创建index_pattern和 Dashboards。
- Global Tenant是所有用户都可以直接访问的，当部门过多时，可以通过给用户分Tenant组进行管理。
2. 在“Security”中选择“Tenants”。单击“+”建立以部门为粒度的Tenant，例如“test_tenant”。

图 2-38 创建 Tenants



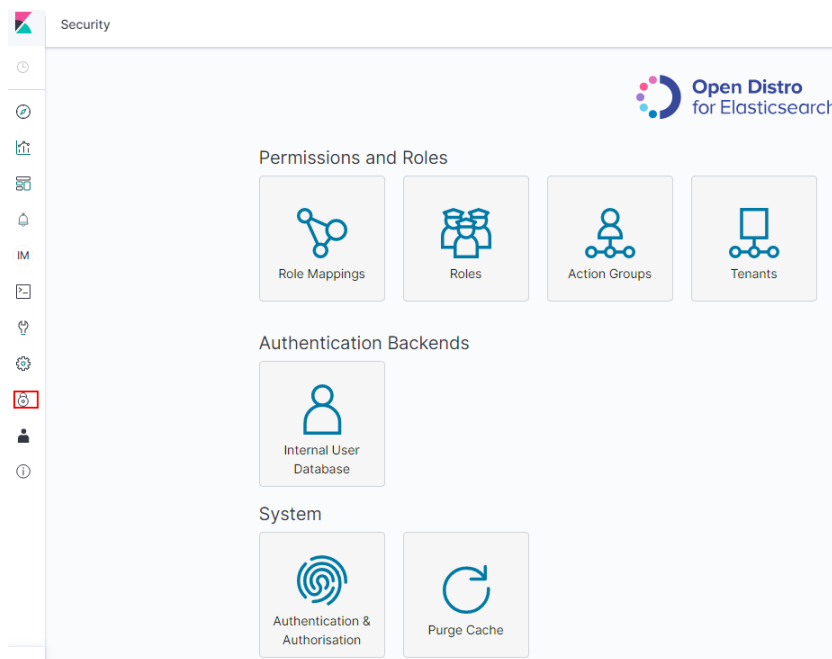
3. 在“test_tenant”下创建本部门支持访问的index_pattern、Dashboards信息。
4. 在“Security”中选择“Roles”，单击test用户对应的角色“Role1”，在“Tenant Permissions”页签，给“Role1”分配“test_tenant”。
5. 保存后使用test用户访问集群，确认能否访问test_tenant空间内容。

创建用户

给安全集群添加一个新用户“test”。

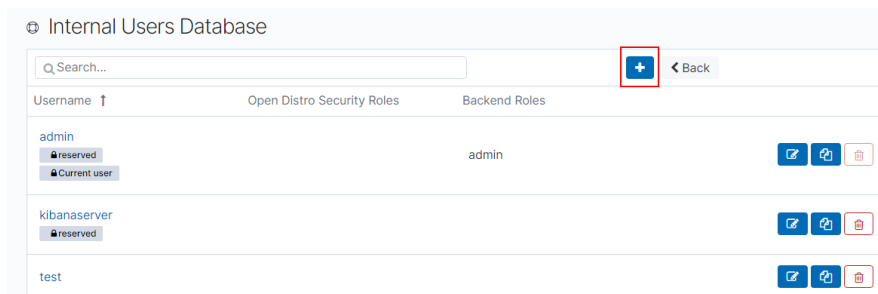
1. 登录成功后，在Kibana操作界面的左侧导航栏选择“Security”，进入“Security”页面。

图 2-39 进入 Security 页面



2. 选择“Authentication Backends” > “Internal Users Database”，进入创建用户页面。
3. 在“Internal Users Database”页面，单击“+”，进入添加用户信息页面。

图 2-40 添加用户（2）



4. 在创建用户页面，输入“Username”、“Password”和“Repeatpassword”，单击“Submit”。

此处以用户名“test”为例。创建成功后，可以在列表中看到新创建的用户。

2.12.3 设置 Elasticsearch 集群标签

标签是集群的标识。为集群添加标签，可以方便用户识别和管理拥有的集群资源。

您可以在创建集群时添加标签，也可以在集群创建完成后，在集群的详情页添加标签。

约束限制

如您的组织已经设定云搜索服务的相关标签策略，则需按照标签策略规则为集群添加标签。标签如果不符合标签策略的规则，则可能会导致集群创建失败，请联系组织管理员了解标签策略详情。

新建集群的标签管理

1. 登录云搜索服务管理控制台。
2. 单击右上角的“创建集群”，进入创建集群页面。
3. 在创建集群页面，“高级配置”选择“自定义”后，为集群添加标签。

您可以选择预定义标签，并为此标签设置“标签值”。您可以单击“查看预定义标签”，进入“标签管理服务”，了解此用户下已有的标签。

您也可以自定义“标签键”和“标签值”。

图 2-41 创建集群时添加标签



云搜索服务的每个集群最多可以设置20个标签。当设置不正确时，可单击标签右侧的“删除”按钮，删除此标签。当不设置标签时，可保持为空。

表 2-131 标签命名规则

参数	说明
标签键	<ul style="list-style-type: none">• 对于同一个集群，标签键值唯一。• 长度不超过64个字符。• 只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @”。不能以空格开头和结尾。• 不能为空。
标签值	<ul style="list-style-type: none">• 长度不超过64个字符。• 只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @/”。不能以空格开头和结尾。• 不能为空。

已有集群的标签管理

您可以对已经创建的集群的标签进行修改，删除，也可以添加标签。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击待管理标签的集群名称。
系统跳转至该集群“基本信息”页面。
3. 左侧菜单栏选择“标签”，在此可以对集群标签进行添加，修改，删除操作。
 - 查看
在“标签”页，可以查看当前集群的标签详情，包括标签个数，以及每个标签的键和值。
 - 添加
单击左上角的“添加标签”，在弹出的“添加标签”窗口，输入新添加标签的键和值，并单击“确定”。
 - 修改
只能修改已有标签的标签值。
单击标签所在行“操作”列下的“编辑”，在弹出的“编辑标签”窗口，输入修改后标签值，并单击“确定”。
 - 删除
单击标签所在行“操作”列下的“删除”，如果确认删除，在弹出的“删除标签”窗口，单击“确定”。

通过标签搜索集群

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击集群列表上方的搜索栏，选择标签键和标签值进行搜索。
标签键和标签值仅支持从下拉列表中选择，当标签键和标签值全匹配时，系统可以自动查询到目标集群。当有多个标签条件时，会取各个标签的交集，进行集群查询。

系统最多支持10个不同标签的组合搜索。

2.12.4 配置 ELasticsearch 集群 YML 文件默认参数

云搜索服务支持用户修改elasticsearch.yml文件。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要修改参数配置的集群名称，进入集群基本信息页面。
3. 选择“参数配置”，单击“编辑”，根据需求修改对应模块的参数值。

表 2-132 模块参数信息说明

模块名称	参数名称	说明
跨域访问	http.cors.allow-credentials	跨域访问是否返回头部的Access-Control-Allow-Credentials。 取值范围：true、false。 默认值：false。
	http.cors.allow-origin	允许跨域访问的IP，配置样例如“122.122.122.122:9200”。
	http.cors.max-age	浏览器默认缓存时间。如果超过设置的时间后，缓存将自动清除。 单位：秒。 默认值：1728000。
	http.cors.allow-headers	跨域访问允许的headers，包括X-Requested-With、Content-Type和Content-Length，中间用英文逗号和空格分开。
	http.cors.enabled	是否允许跨域访问。 取值范围：true、false。 默认值：false。
	http.cors.allow-methods	跨域访问允许的方法，包括OPTIONS、HEAD、GET、POST、PUT、DELETE，中间用英文逗号和空格分开。
集群索引重建	reindex.remote.whitelist	配置该参数可以将本集群数据通过reindex接口迁移到配置的集群，配置样例如“122.122.122.122:9200”。
自定义缓存	indices.queries.cache.size	查询阶段的缓存大小。 取值范围：1-100。 单位：%。 默认值：10%。

模块名称	参数名称	说明
线程池队列大小	thread_pool.force_merge.size	用来做forcemerge的队列大小。输入的参数值为整数类型。 默认值：1。
自定义	用户可以根据实际情况，添加相关参数名称。	自定义参数的取值。 说明 <ul style="list-style-type: none">如果自定义参数有多个取值，则取值的输入格式为[value1, value1, value1...]取值之间用英文逗号和空格隔开。自定义参数值中不能包含冒号。

4. 修改完成后，单击上方的“提交”弹出“提交配置”窗口，确认参数无误后勾选“参数修改后需要手动重启才能生效”，单击“确定”。
当下方的参数修改列表显示“作业状态”为“成功”时，表示修改保存成功。系统最多显示20条修改记录。
5. 返回集群列表，单击集群操作列的“更多 > 重启”重启集群，使修改的配置生效。
 - 如果修改了参数配置，未重启集群，则在“集群管理”页面的“任务状态”栏显示为“配置未更新”。
 - 如果修改后重启集群，“任务状态”显示“配置错误”，则表示修改参数配置文件失败。

2.12.5 为 Elasticsearch 集群绑定企业项目

企业可以根据组织架构规划企业项目，将企业分布在不同区域的资源按照企业项目进行统一管理，同时可以为每个企业项目设置拥有不同权限的用户组和用户。本章节为您介绍CSS集群如何绑定、修改企业项目。

前提条件

在使用企业项目前，需账号开通权限才可以进行操作。开通权限可以通过提交[工单](#)进行申请。

在绑定企业项目前，您已在“企业项目管理控制台”[创建企业项目](#)。

绑定企业项目

在创建集群时，可以在“企业项目”绑定已创建的企业项目，也可以单击“查看项目管理”，前往企业项目管理控制台，新建企业项目和查看已有的企业项目。

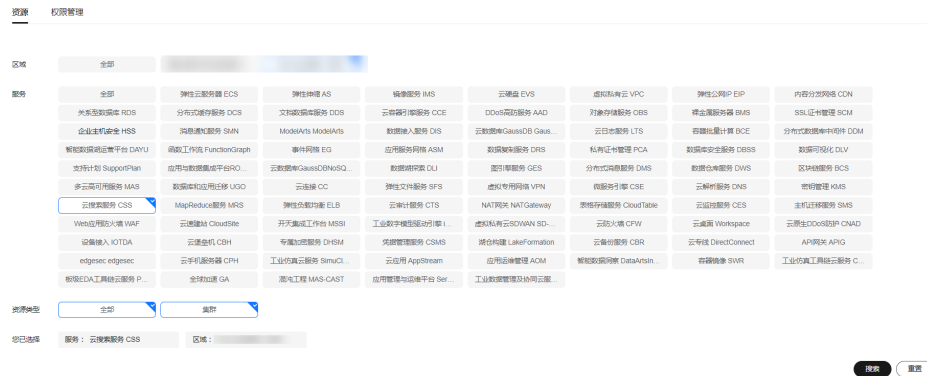
修改企业项目

针对之前已创建的集群，其绑定的企业项目可根据实际情况进行修改。

1. 登录在云搜索服务管理控制台，
2. 在左侧导航栏，选择对应的集群类型，进入集群管理页面。
3. 在集群列表中，单击集群名称进入集群“基本信息”页面。

4. 在集群“基本信息”页面，单击“企业项目”右侧的企业项目名称，进入项目管理页面。
5. 在“资源”页签下，“区域”选项中选择当前集群所在的区域，“服务”选项中选择“云搜索服务 CSS”。此时，资源列表将筛选出对应的CSS集群。

图 2-42 筛选 CSS 集群



6. 勾选需要修改企业项目的集群，然后单击“迁出”。
7. 在“迁出资源”页面，选择“迁出方式”，再选择“请选择要迁入的企业项目”，然后单击“确定”。
8. 迁出完成后，可以在云搜索服务管理控制台集群管理页面，查看修改后的集群企业项目信息。

2.12.6 替换 Elasticsearch 集群指定节点

当集群中的节点发生故障时，可以执行“替换指定节点”任务。通过删除故障节点，重建一个一样的新节点，实现节点替换。替换指定节点过程中会将替换节点的数据提前转移，不会造成数据丢失。

前提条件

集群处于“可用”状态，且无正在进行的任务。

约束限制

- 一次只能选择一个节点进行替换。
- 节点替换过程会按照原节点的ID、IP地址、规格、AZ等信息重建节点。
- 节点替换过程中不会保留手动操作。例如被替换的节点手动加过回程路由，那么节点替换完成后，需要重新添加回程路由。
- 当替换的是数据节点或冷数据节点时，需要确认集群或节点是否满足以下条件。
 - a. 替换数据节点或冷数据节点的过程中，会先将被替换节点的数据迁移到其他数据节点，因此集群中每个索引的副本数和主分片数之和的最大值要小于集群的数据节点数量和冷数据节点数量之和。替换过程耗时跟数据迁移到其他节点的耗时强相关。
 - b. 版本号在7.6.2以前的集群，不能有close的索引，否则不支持替换数据节点或冷数据节点。
 - c. 被替换的数据节点或冷数据节点所在的AZ需要有两个及以上的数据节点或冷数据节点。

- d. 当替换的数据节点或冷数据节点所在集群不存在Master节点时，则集群中可用的数据节点数量和冷数据节点数量之和要大于等于3。
- e. 当替换的是Master节点或Client节点时，则不受以上四条约束。
- f. 当替换的是故障节点时，不管什么节点类型都不受以上四条约束。因为故障节点不包含在“_cat/nodes”中。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
4. 选择“替换指定节点”页签。
5. 在替换指定节点页面，设置参数。

表 2-133 替换指定节点

参数	说明
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平面时才显示该参数。</p> <ul style="list-style-type: none">● 删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。● 当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。● 当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。● 建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。
节点类型	展开本次要变更的节点类型，此处会显示当前节点类型拥有的节点，勾选要替换的节点。

6. 单击“提交申请”。在数据迁移弹窗中确认是否进行数据迁移（建议勾选“进行数据迁移”避免数据丢失），单击“确认”提交申请。

数据迁移过程中，系统会把待替换节点中的所有数据分片移动到剩余节点，待数据迁移完成后再进行节点替换操作。当待替换节点上的数据在剩余节点存在副本时，可跳过数据迁移，缩短集群变更时间，减少数据迁移带来的负载压力。
7. 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“节点替换中”，表示集群正在替换节点。当集群状态变为“可用”，则表示节点替换成功。

2.12.7 更改 Elasticsearch 集群安全模式

集群创建完成后，可以参考本文更改集群的安全模式。

场景描述

CSS服务支持创建多种安全模式的集群，不同集群的差异请参见[表2-134](#)。

表 2-134 集群类型介绍

集群类型		集群描述	适用场景
非安全集群	非安全模式的集群	非安全模式的集群无需安全认证即可访问，采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	<p>适合内网业务，用于测试场景。</p> <ul style="list-style-type: none"> • 优点：简单，接入集群容易。 • 缺点：安全性差，任何人都可以访问集群。
安全集群	安全模式+HTTP协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	<p>可以实现用户权限隔离，适用于对集群性能敏感的场景。</p> <ul style="list-style-type: none"> • 优点：访问集群需要安全认证，提升了集群安全性。通过HTTP协议访问集群又能保留集群的高性能。 • 缺点：无法公网访问集群。
	安全模式+HTTPS协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTPS协议进行通信加密，使数据更安全。	<p>有非常高的安全要求，且需要公网访问集群的场景。</p> <ul style="list-style-type: none"> • 优点：访问集群需要安全认证，提升了集群安全性，且HTTPS协议的通讯加密可以实现集群公网访问功能。 • 缺点：通过HTTPS协议访问集群，集群的读取性能相对HTTP协议来说，会下降20%左右。

同时，CSS服务支持多种安全模式的更改场景，请参见[表2-135](#)。

表 2-135 安全模式的更改场景

场景	相关文档
“非安全模式的集群”切换为“安全模式+HTTP协议的集群”	非安全模式切换为安全模式
“非安全模式的集群”切换为“安全模式+HTTPS协议的集群”	
“安全模式+HTTP协议的集群”切换为“非安全模式的集群”	安全模式切换为非安全模式

场景	相关文档
“安全模式+HTTPS协议的集群”切换为“非安全模式的集群”	
“安全模式+HTTP协议的集群”切换为“安全模式+HTTPS协议的集群”	切换安全模式下的协议
“安全模式+HTTPS协议的集群”切换为“安全模式+HTTP协议的集群”	

前提条件

- 建议更改集群安全模式前，先完成数据备份。
- 集群必须处于“可用”状态，且无正在进行的任务。
- 确认集群是否启用“负载均衡”，如果启用了，则先关闭集群的“负载均衡”，待安全模式变更完成后再开启“负载均衡”。避免变更过程中造成通过负载均衡实例访问集群异常。

约束限制

- 2022年11月之后创建的集群（且集群版本不小于6.5.4）才支持切换安全模式。
- 在更改集群安全模式的过程中，集群会自动重启。重启会导致业务中断，并且重启后调用集群的认证方式也会发生改变，客户端需要做相应调整。
- 对于已打开过Kibana会话框的集群，在更改集群安全模式后，Kibana会提示Session错误，此时需要清理缓存再打开Kibana才能正常访问。

非安全模式切换为安全模式

介绍集群从“非安全模式”切换为“安全模式+HTTP协议”或“安全模式+HTTPS协议”的操作。当集群从非安全模式变更为安全模式后，访问集群将需要进行安全认证。

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择“集群管理>Elasticsearch”，进入Elasticsearch集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
4. 选择“更改安全模式”页签。
5. 在更改安全模式页面，打开“安全模式”的开关，输入并确认集群的管理员密码。

图 2-43 非安全模式切换为安全模式

安全模式 开启安全模式，访问集群需要安全认证。

管理员账户名 admin

管理员密码

确认密码

HTTPS访问

- 选择集群是否启用“HTTPS访问”。
 - 打开开关：实现“非安全模式”切换为“安全模式+HTTPS协议”。集群通讯使用HTTPS协议，通讯数据将进行加密，且可以启用集群的公网访问功能。
 - 关闭开关：实现“非安全模式”切换为“安全模式+HTTP协议”。集群通讯使用HTTP协议，无法启用集群的公网访问功能。
- 单击“提交申请”，确认后将返回集群列表页面。

集群的“任务状态”列中显示为“安全模式更改中”，当“集群状态”变为“可用”表示变更成功。

安全模式切换为非安全模式

介绍集群从“安全模式+HTTP协议”或“安全模式+HTTPS协议”切换为“非安全模式”的操作。当集群从安全模式变更为非安全模式后，访问集群将不再需要进行安全认证。

须知

- 非安全模式的集群将无需安全认证即可访问，且采用的是HTTP协议传输数据，所以请确保集群访问环境的安全性，勿将访问接口暴露到公网环境上。
- 在安全模式切换为非安全模式的过程中，变更任务会删除原安全模式的集群用到的索引。切换前请做好数据备份，以免数据丢失。
- 如果集群已绑定公网IP，则无法将安全模式切换为非安全模式，需要先解绑公网IP，才能切换。
- 如果集群已启用Kibana公网访问，则无法将安全模式切换为非安全模式，需要先关闭Kibana公网访问功能，才能切换。

- 登录云搜索服务管理控制台。
- 左侧导航栏选择“集群管理”，进入集群列表页面，选择需要更改安全模式的集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
- 选择“更改安全模式”页签。
- 在更改安全模式页面，关闭“安全模式”的开关。

图 2-44 安全模式切换为非安全模式

安全模式 关闭安全模式后，集群将无需安全认证即可访问，并且采用HTTP明文传输数据，所以请确保访问环境的安全性，勿将访问接口暴露到公网环境上。

- 单击“提交申请”，确认后将返回集群列表页面。
集群的“任务状态”列中显示为“安全模式更改中”，当“集群状态”变为“可用”表示变更成功。

切换安全模式下的协议

介绍安全模式的集群切换访问协议的操作，包括“安全模式+HTTP协议”切换为“安全模式+HTTPS协议”、“安全模式+HTTPS协议”切换为“安全模式+HTTP协议”。

须知

如果集群已绑定公网IP，则无法将协议从HTTPS切换到HTTP，需要先解绑公网IP，才能切换安全模式下的协议。

- 登录云搜索服务管理控制台。
- 左侧导航栏选择“集群管理”，进入集群列表页面，选择需要更改安全模式的集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
- 选择“更改安全模式”页签。
- 在更改安全模式页面，修改“HTTPS访问”的开关切换安全模式下的集群访问协议。

图 2-45 切换协议

安全模式	<input checked="" type="checkbox"/> 开启安全模式，访问集群需要安全认证。
管理员账户名	admin
管理员密码	<input type="password"/>
确认密码	<input type="password"/>
HTTPS访问	<input checked="" type="checkbox"/> 开启HTTPS访问，访问集群将进行通讯加密。

- 打开开关：实现“安全模式+HTTP协议”切换为“安全模式+HTTPS协议”。切换为HTTPS协议后，集群通讯将进行加密，且可以启用集群的公网访问功能。
 - 关闭开关：实现“安全模式+HTTPS协议”切换为“安全模式+HTTP协议”。关闭开关会弹出告警提示，单击“确定”后关闭开关。
切换为HTTP协议后，集群通讯将不再加密，且无法启用集群的公网访问功能。
- 单击“提交申请”，确认后将返回集群列表页面。
集群的“任务状态”列中显示为“安全模式更改中”，当“集群状态”变为“可用”表示变更成功。

2.12.8 切换 Elasticsearch 集群可用区

CSS服务支持跨可用区的能力，在集群使用过程中遇到可用区资源不足或需要提升集群的高可用性时，可以通过切换可用区实现。

场景描述

切换可用区包含两大场景：可用区高可用改造和可用区平移切换。

- 可用区高可用改造：适用于单AZ改造成两AZ、单AZ改造成三AZ或两AZ改造成三AZ的场景，目的是为了提升集群的高可用性。
- 可用区平移切换：适用于从一个AZ完全迁移到另一个AZ的场景，是为了解决当前可用区资源不足的问题。

前提条件

- 确保存在资源充足的可用区。
- 集群处于“可用”状态，且无正在进行的任务。
- 确认集群不存在非标操作。当集群进行过修改回程路由、系统参数、Kibana配置等这些非标操作时，这些操作无法被记录到系统中，因此在切换过程中将无法继承，切换后可能会影响业务。

约束限制

- 为了保证业务的连续性，集群中数据节点和冷数据节点的个数之和要不小于3。
- 切换过程中会逐一下线节点再新建节点，需要保证单节点下线后，其余节点的磁盘容量能够接纳该节点的所有数据。
- 集群中索引的最大主备分片数能够被分配到剩余数据节点和冷数据节点中，防止切换过程中出现节点下线后副本无法分配的情况。即“索引的主备分片数的最大值+1 ≤ 切换前的数据节点数和冷数据节点之和”。
- 切换前建议做好数据备份，防止升级故障数据丢失。
- 当集群处于切换可用区过程中，部分节点的AZ属性可能已经更改，此时节点的可用区可能会显示出中间状态，等待任务结束后，集群的可用区显示将恢复正常状态。
- **可用区高可用改造**是基于原可用区进行扩展，切换时必须保留原可用区。如果是单AZ改造成两AZ或单AZ改造成三AZ的场景，必须所有节点一起改造；如果是两AZ改造成三AZ的场景，则既支持同时操作集群中所有类型的节点，也支持单独操作集群中某一类型的节点（例如，在已有两AZ的集群中，可以只对Master节点进行切换可用区，将Master节点形变为三AZ，而其他节点依旧是两AZ）。高可用改造会尽可能挪动最小的节点重建集群，改造完成后，因未挪动节点的yml配置更新，需要重启集群才能生效。
- **可用区平移切换**一次只能迁移一个可用区，切换时选择的可用区只能有一个不同。平移切换支持同时操作集群中所有类型的节点，或者单独操作集群中某一类型的节点（例如，在已有两AZ的集群中，可以只对Master节点进行平移切换，将Master节点从当前AZ迁移到另一个AZ，而其他节点依旧在当前AZ）。除了单AZ平移切换场景，其他的多AZ平移切换场景均需要重启集群生效。

操作步骤

1. 登录云搜索服务管理控制台。

2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
4. 选择“切换可用区”页签。
5. 在切换可用区页面，配置切换参数。

表 2-136 切换可用区的参数说明

参数	描述
切换类型	<ul style="list-style-type: none">• 可用区高可用改造：支持单AZ改造成两AZ、单AZ改造成三AZ或两AZ改造成三AZ。 高可用改造会尽可能挪动最小的节点重建集群，改造完成后，因未挪动节点的yaml配置更新，需要重启集群才能生效。• 可用区平移切换：支持从一个AZ迁移到另一个AZ。 除了单AZ平移切换场景，其他的多AZ平移切换场景均需要重启集群生效。
当前节点类型	选择进行AZ切换的节点类型。每次可选择一种节点类型进行操作，当集群有多种节点类型时，可选择“所有节点”进行全部切换。 说明 如果是单AZ改造成两AZ或单AZ改造成三AZ的场景，只支持所有节点一起改造，因此只能选择“所有节点”。
当前AZ集合	显示当前集群所在的可用区。
目标AZ集合	选择需要切换的目标可用区。 <ul style="list-style-type: none">• 可用区高可用改造：高可用区改造是基于原可用区进行扩展，因此“目标AZ集合”必须包含“当前AZ集合”且至少再选择一个可用区，最多选择3个可用区。• 可用区平移切换：一次切换任务只能迁移一个AZ，因此“目标AZ集合”只能有一个可用区和“当前AZ集合”不同。
配置委托	选择IAM委托，授权当前账号切换可用区的操作权限。 如果没有合适的委托，可以单击“创建委托”跳转到IAM控制台新建委托。 说明 选择的委托必须授权了“Tenant Administrator”或“VPC Administrator”策略。

6. 配置完成后，单击“提交申请”。确认是否进行全量索引快照备份检测，单击“确定”启动切换任务。

图 2-46 全量索引快照备份检测

全量索引快照备份检测

建议您在切换前进行全量索引快照备份检测。
如果您已备份，请检查备份时间是否超过1个月，若超过1个月，建议您备份最新数据。
当前检测只能针对索引名称进行匹配，无法校验具体内容以及备份时间。

进行全量索引快照备份检测

取消

确定

7. 在“任务记录”列表，显示当前切换任务。当“任务状态”为“运行中”时，可以展开任务列表，单击“查看进度”查看详细的切换进度。

当“任务状态”为“失败”时，可以重试任务或者直接终止任务。

- 重试切换任务：在任务列表的操作列，单击“重试”，重新切换可用区。
- 终止切换任务：在任务列表的操作列，单击“终止”，结束切换可用区。

当任务终止后，如果原节点还未切换成功，可以通过[替换Elasticsearch集群指定节点](#)尝试恢复节点。

📖 说明

如果存在部分节点已经完成切换，此时集群的AZ形态已经发生变化，此时终止切换任务，可能无法再次按照上一次的切换请求进行命令下发，请谨慎执行终止切换任务。

2.12.9 配置 Elasticsearch 集群自定义词库

2.12.9.1 Elasticsearch 集群词库介绍

背景信息

云搜索服务的词库用于对文本进行分词，使得一些特殊词语在分词的时候能够被识别出来，便于根据关键词搜索文本数据。例如，根据公司名称来查询，如“华为”；或者根据网络流行词来查询，如“喜大普奔”。也支持基于同义词词库，根据同义词搜索文本数据。

CSS服务使用的分词器包括IK分词器和同义词分词器。IK分词器配备主词词库和停词词库；同义词分词器配备同义词词库。其中，IK分词器包含ik_max_word和ik_smart分词策略。同义词分词器使用的是ik_synonym分词策略。

- ik_max_word：会将文本做最细粒度的拆分，比如会将“昨夜西风吹折干林梢”拆分为“昨夜西风,昨夜,西风,吹折干林梢,吹折,干林梢,干,林,折干林,干林,吹”，会穷尽各种可能的分词组合。
- ik_smart：会做最粗粒度的拆分，比如会将“昨夜西风吹折干林梢”拆分为“昨夜西风,吹折干林梢”。

CSS服务给集群预置了静态主词词库、静态停词词库、Extra主词词库和Extra停词词库这四个词库。

- 当这些预置词库已满足集群业务的分词需求时，则集群无需配置自定义词库即可直接实现关键词搜索。
- 当预置词库不满足集群业务分词需求时，可以给集群添加主词词库、停词词库或同义词词库，亦或者是修改预置的四个词库，使集群能够实现关键词或同义词搜索。

CSS服务的各类词库介绍请参见表2-137。

表 2-137 词库介绍

词库类型		说明	文件要求
自定义	主词词库	主词为用户希望进行分词的特殊词语，例如“智能手机”和“喜大普奔”。主词词库则是用户自定义的特殊词语的集合。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，主词文件最大支持100M。如果涉及单词，必须改成小写字母。
	停词词库	停词为用户不希望进行分词或者关注的词语，例如“的”、“什么”、“怎么”等。停词词库是用户自定义的停词词语的集合。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，停词文件最大支持100M。
	同义词词库	同义词为意义相同的一组词语，例如“开心”和“高兴”。同义词词库是用户自定义的同义词词语的集合。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一组同义词对，同义词对中的同义词间逗号隔开，同义词文件最大支持100MB。
预置	静态主词词库	静态主词词库是CSS服务预置的常用主词集合，当需要查看静态主词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/main.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，静态主词文件最大支持100M。
	静态停词词库	静态停词词库是CSS服务预置的常用停词集合，当需要查看静态停词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/stopword.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，静态停词文件最大支持100M。
	Extra主词词库	Extra主词词库是CSS服务预置的生僻主词集合，当需要查看Extra主词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/extra_main.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，Extra主词文件最大支持100M。

词库类型	说明	文件要求
Extra停词词库	Extra停词词库是CSS服务预置的生僻停词集合，当需要查看Extra停词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/extra_stopword.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，Extra停词文件最大支持100M。

约束限制

- 自定义词库功能上线之前（即2018年3月10日之前）创建的集群，无法使用自定义词库功能。
- 自定义词库一般用于中文分词，如果用于英文分词，会按照除“#&+-.@_”外的特殊符号进行切分。
- 词库文件要满足要求，不同词库的词库文件要求请参见[表2-137](#)。
- 操作账号必须具有配置自定义词库的权限。可以在IAM服务的“权限管理 > 权限”查看是否有以下权限，如果没有请联系服务管理员添加。
 - “CSS Administrator”角色。
 - “OBS Administrator”策略

2.12.9.2 配置和使用 Elasticsearch 集群的自定义词库

前提条件

已经准备好集群和自定义词库，且词库文件已上传至OBS桶。

- 集群和词库文件满足[约束限制](#)。
- 上传的OBS桶必须和集群在相同“区域”。上传文件至OBS桶的操作步骤请参见[上传文件](#)。

配置自定义词库

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表页面。
3. 在“集群管理”页面，单击需要配置自定义词库的集群名称，进入集群基本信息页面。
4. 选择“自定义词库”。
5. 在“自定义词库”页面，配置集群的自定义词库或修改预置词库。
 - a. 当需要配置用户自定义的词库时，参考[表2-138](#)完成配置。

表 2-138 配置自定义词库

参数	说明
OBS桶	选择词库文件存储的OBS位置。 单击“创建桶”可以跳转到创建桶页面新建OBS桶，新建OBS桶必须和集群在相同“区域”，且“默认存储类别”只支持“标准存储”或“低频访问存储”。
主词词库	主词词库是用户自定义的词库，初始状态为空。默认选择“不更新”表示不配置该词库。 <ul style="list-style-type: none">当需要添加自定义的主词词库时，单击“更新”，选择txt格式的词库文件。当无需添加自定义的主词词库时，单击“不使用此词库”，删除词库。
停词词库	停词词库是用户自定义的词库，初始状态为空。默认选择“不更新”表示不配置该词库。 <ul style="list-style-type: none">当需要添加自定义的停词词库时，单击“更新”，选择txt格式的词库文件。当无需添加自定义的停词词库时，单击“不使用此词库”，删除词库。
同义词词库	同义词词库是用户自定义的词库，初始状态为空。默认选择“不更新”表示不配置该词库。 <ul style="list-style-type: none">当需要添加自定义的同义词词库时，单击“更新”，选择txt格式的词库文件。当无需添加自定义的同义词词库时，单击“不使用此词库”，删除词库。

- b. 当需要修改预置词库时，单击打开“修改预置词库”右侧的开关，修改对应的预置词库。

说明

如果界面不存在静态词库和Extra词库这4个词库时，表示该集群版本不支持删除或修改这4个预置词库。如果想要使用该功能，建议升级集群版本，或者新建集群并进行数据迁移。

表 2-139 配置预置词库

参数	说明
静态主词词库	静态主词词库是预置的常用词语的主词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的静态主词词库时，单击“更新”，选择txt格式的词库文件。当不使用静态主词库时，单击“不使用此词库”，删除词库。

参数	说明
静态停词词库	静态停词词库是预置的常用词语的停词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的静态停词词库时，单击“更新”，选择txt格式的词库文件。当不使用静态停词库时，单击“不使用此词库”，删除词库。
Extra主词词库	Extra主词词库是预置的生僻词语的主词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的Extra主词词库时，单击“更新”，选择txt格式的的词库文件。当不使用Extra主词库时，单击“不使用此词库”，删除词库。
Extra停词词库	Extra停词词库是预置的生僻词语的Extra停词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的Extra停词词库时，单击“更新”，选择txt格式的的词库文件。当不使用Extra停词库时，单击“不使用此词库”，删除词库。

- 单击“保存”，在弹窗中单击“确定”。词库信息在下方呈现，此时词库状态为“更新中”。请耐心等待1分钟左右，当词库配置完成后，词库状态变更为“成功”。
- 当删除或更新静态词库和Extra词库这4个词库时，需要重启集群才能使配置的词库生效；其他词库的更新为动态更新，无需重启集群。重启集群的操作指导请参见[重启Elasticsearch集群](#)。

使用示例

通过给集群配置自定义词库，将“智能手机”设置为主词，“是”设置为停词，“开心”和“高兴”设置为同义词。使用配置好的集群，对文本内容“智能手机是很好用”进行关键词搜索，查看关键词查询效果；对文本内容“我今天获奖了我很开心”进行同义词搜索，查看同义词查询效果。

步骤1 配置自定义词库，并查看分词效果。当预置词库即可满足分词需求时，可以跳过此步骤。

- 准备词库文件（UTF-8无BOM格式编码的文本文件），上传到对应OBS路径下。主词词库文件中包含词语“智能手机”；停词词库文件中包含词语“是”；同义词词库文件中包含一组同义词“开心”和“高兴”。

说明

由于系统预置的静态停词词库包含了“是”、“的”等常用词，如果集群未删除或更新预置词库，则此类停用词可以不用上传。

- 参考[配置自定义词库](#)，完成词库配置。
- 待词库配置信息生效后，返回集群列表。单击集群操作列的“Kibana”接入集群。

4. 在Kibana界面，单击左侧导航栏的“Dev Tools”，进入操作页面。
5. 执行如下命令，查看自定义词库的ik_smart分词策略和ik_max_word分词策略的分词效果。

- 使用ik_smart分词策略对文本内容“智能手机是很好用”进行分词。

示例代码：

```
POST /_analyze
{
  "analyzer":"ik_smart",
  "text":"智能手机是很好用"
}
```

运行结束后，查看分词效果：

```
{
  "tokens": [
    {
      "token": "智能手机",
      "start_offset": 0,
      "end_offset": 4,
      "type": "CN_WORD",
      "position": 0
    },
    {
      "token": "很好用",
      "start_offset": 5,
      "end_offset": 8,
      "type": "CN_WORD",
      "position": 1
    }
  ]
}
```

- 使用ik_max_word分词策略对文本内容“智能手机是很好用”进行分词。

示例代码：

```
POST /_analyze
{
  "analyzer":"ik_max_word",
  "text":"智能手机是很好用"
}
```

运行结束后，查看分词效果：

```
{
  "tokens": [
    {
      "token": "智能手机",
      "start_offset": 0,
      "end_offset": 4,
      "type": "CN_WORD",
      "position": 0
    },
    {
      "token": "智能",
      "start_offset": 0,
      "end_offset": 2,
      "type": "CN_WORD",
      "position": 1
    },
    {
      "token": "智",
      "start_offset": 0,
      "end_offset": 1,
      "type": "CN_WORD",
      "position": 2
    },
    {
      "token": "能手",
      "start_offset": 1,

```

```
"end_offset" : 3,
"type" : "CN_WORD",
"position" : 3
},
{
  "token" : "手机",
  "start_offset" : 2,
  "end_offset" : 4,
  "type" : "CN_WORD",
  "position" : 4
},
{
  "token" : "机",
  "start_offset" : 3,
  "end_offset" : 4,
  "type" : "CN_WORD",
  "position" : 5
},
{
  "token" : "很好用",
  "start_offset" : 5,
  "end_offset" : 8,
  "type" : "CN_WORD",
  "position" : 6
},
{
  "token" : "很好",
  "start_offset" : 5,
  "end_offset" : 7,
  "type" : "CN_WORD",
  "position" : 7
},
{
  "token" : "好用",
  "start_offset" : 6,
  "end_offset" : 8,
  "type" : "CN_WORD",
  "position" : 8
},
{
  "token" : "用",
  "start_offset" : 7,
  "end_offset" : 8,
  "type" : "CN_WORD",
  "position" : 9
}
]
}
```

步骤2 创建索引并配置分词策略，导入数据后使用关键词搜索数据。

Elasticsearch 7.x之前的版本和之后的版本，操作命令有差别，所以分开举例。

Elasticsearch版本<7.x的代码示例

1. 创建索引“book”，配置分词策略。

示例中“analyzer”和“search_analyzer”可以根据实际需要“ik_max_word”或“ik_smart”分词策略，此处以“ik_max_word”为例。

```
PUT /book
{
  "settings": {
    "number_of_shards": 2,
    "number_of_replicas": 1
  },
  "mappings": {
    "type1": {
      "properties": {
        "content": {
```

```
        "type": "text",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word"
      }
    }
  }
}
```

2. 导入数据，将文本信息导入“book”索引中。

```
PUT /book/type1/1
{
  "content": "智能手机是很好用"
}
```

3. 使用关键词“智能手机”进行文本搜索，并查看搜索结果。

```
GET /book/type1/_search
{
  "query": {
    "match": {
      "content": "智能手机"
    }
  }
}
```

搜索结果：

```
{
  "took" : 20,
  "timed_out" : false,
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.1507283,
    "hits" : [
      {
        "_index" : "book",
        "_type" : "type1",
        "_id" : "1",
        "_score" : 1.1507283,
        "_source" : {
          "content" : "智能手机是很好用"
        }
      }
    ]
  }
}
```

Elasticsearch版本≥7.x和OpenSearch版本的代码示例

1. 创建索引“book”，配置分词策略。

示例中“analyzer”和“search_analyzer”可以根据实际需要“ik_max_word”或“ik_smart”分词策略，此处以“ik_max_word”为例。

```
PUT /book
{
  "settings": {
    "number_of_shards": 2,
    "number_of_replicas": 1
  },
  "mappings": {
    "properties": {
      "content": {
        "type": "text",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word"
      }
    }
  }
}
```

```
    }  
  }  
}
```

2. 导入数据，将文本信息导入“book”索引中。

```
PUT /book/_doc/1  
{  
  "content": "智能手机是很好用"  
}
```

3. 使用关键词“智能手机”进行文本搜索，并查看搜索结果。

```
GET /book/_doc/_search  
{  
  "query": {  
    "match": {  
      "content": "智能手机"  
    }  
  }  
}
```

搜索结果:

```
{  
  "took": 16,  
  "timed_out": false,  
  "_shards": {  
    "total": 2,  
    "successful": 2,  
    "skipped": 0,  
    "failed": 0  
  },  
  "hits": {  
    "total": {  
      "value": 1,  
      "relation": "eq"  
    },  
    "max_score": 1.7260926,  
    "hits": [  
      {  
        "_index": "book",  
        "_type": "_doc",  
        "_id": "1",  
        "_score": 1.7260926,  
        "_source": {  
          "content": "智能手机是很好用"  
        }  
      }  
    ]  
  }  
}
```

步骤3 创建索引并配置同义词策略，导入数据后使用同义词搜索数据。

Elasticsearch 7.x之前的版本和之后的版本，操作命令有差别，所以分开举例。

Elasticsearch版本<7.x的代码示例

1. 创建索引“myindex”，配置分词策略。

```
PUT myindex  
{  
  "settings": {  
    "analysis": {  
      "filter": {  
        "my_synonym": {  
          "type": "dynamic_synonym"  
        }  
      },  
      "analyzer": {  
        "ik_synonym": {  
          "filter": [  

```

```
    "my_synonym"
  ],
  "type": "custom",
  "tokenizer": "ik_smart"
}
}
},
"mappings": {
  "mytype": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ik_synonym"
      }
    }
  }
}
```

2. 导入数据，将文本信息导入“myindex”索引中。

```
PUT /myindex/mytype/1
{
  "desc": "我今天获奖了我很开心"
}
```

3. 使用同义词“高兴”进行文本搜索，并查看搜索结果。

```
GET /myindex/_search
{
  "query": {
    "match": {
      "desc": "高兴"
    }
  }
}
```

搜索结果：

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 0.49445358,
    "hits" : [
      {
        "_index" : "myindex",
        "_type" : "mytype",
        "_id" : "1",
        "_score" : 0.49445358,
        "_source" : {
          "desc" : "我今天获奖了我很开心"
        }
      }
    ]
  }
}
```

Elasticsearch版本≥7.x和OpenSearch版本的代码示例

1. 创建索引“myindex”，配置分词策略。

```
PUT myindex
{
  "settings": {
    "analysis": {
```

```
    "filter": {
      "my_synonym": {
        "type": "dynamic_synonym"
      }
    },
    "analyzer": {
      "ik_synonym": {
        "filter": [
          "my_synonym"
        ],
        "type": "custom",
        "tokenizer": "ik_smart"
      }
    }
  },
  "mappings": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ik_synonym"
      }
    }
  }
}
```

2. 导入数据，将文本信息导入“myindex”索引中。

```
PUT /myindex/_doc/1
{
  "desc": "我今天获奖了我很开心"
}
```

3. 使用同义词“高兴”进行文本搜索，并查看搜索结果。

```
GET /myindex/_search
{
  "query": {
    "match": {
      "desc": "高兴"
    }
  }
}
```

搜索结果：

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.1519955,
    "hits" : [
      {
        "_index" : "myindex",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.1519955,
        "_source" : {
          "desc" : "我今天获奖了我很开心"
        }
      }
    ]
  }
}
```

```
}  
}
```

----结束

2.12.10 在 Elasticsearch 集群实现搜索数据的简繁体转换

本文介绍如何在Elasticsearch集群实现搜索数据的简繁体转换。

场景介绍

简繁体转换插件是一款可以使中文简体和中文繁体相互转换的插件。通过该插件的转换，用户可以使用中文繁体关键字搜索出包含对应中文简体的索引数据，也可以使用中文简体关键字搜索出包含对应中文繁体的索引数据。

云搜索服务默认安装了简繁体转换插件，用户无需自行安装。

简繁体转换插件通常可以当做analyzer、tokenizer、token-filter或char-filter来使用。

简繁体转换插件的转换类型包含如下两种：

- s2t：将中文简体转换为中文繁体。
- t2s：将中文繁体转换为中文简体。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择“集群管理”，进入集群列表页面。
3. 在集群列表中，单击需要使用的集群对应“操作”列的“Kibana”。
如果开启了安全模式，需要输入创建集群时设置的管理员账户名和密码。
4. 在Kibana的左侧导航中选择“Dev Tools”，进入Console界面。
5. 在Console界面，执行如下命令，创建索引“stconvert”，并指定自定义映射来定义数据类型。

Elasticsearch低于7.x版本的代码示例

```
PUT /stconvert  
{  
  "settings": {  
    "number_of_shards": 1,  
    "number_of_replicas": 0,  
    "analysis": {  
      "analyzer": {  
        "ts_ik": {  
          "tokenizer": "ik_smart",  
          "char_filter": [  
            "tsconvert",  
            "stconvert"  
          ]  
        }  
      },  
      "char_filter": {  
        "tsconvert": {  
          "type": "stconvert",  
          "convert_type": "t2s"  
        },  
        "stconvert": {  
          "type": "stconvert",  
          "convert_type": "s2t"  
        }  
      }  
    }  
  }  
}
```



```
    }  
  },  
  "mappings": {  
    "type": {  
      "properties": {  
        "desc": {  
          "type": "text",  
          "analyzer": "ts_ik"  
        }  
      }  
    }  
  }  
}
```

Elasticsearch 大于或等于 7.x 版本和 OpenSearch 集群的代码示例

```
PUT /stconvert  
{  
  "settings": {  
    "number_of_shards": 1,  
    "number_of_replicas": 0,  
    "analysis": {  
      "analyzer": {  
        "ts_ik": {  
          "tokenizer": "ik_smart",  
          "char_filter": [  
            "tsconvert",  
            "stconvert"  
          ]  
        }  
      },  
      "char_filter": {  
        "tsconvert": {  
          "type": "stconvert",  
          "convert_type": "t2s"  
        },  
        "stconvert": {  
          "type": "stconvert",  
          "convert_type": "s2t"  
        }  
      }  
    }  
  },  
  "mappings": {  
    "properties": {  
      "desc": {  
        "type": "text",  
        "analyzer": "ts_ik"  
      }  
    }  
  }  
}
```

返回结果如下所示。

```
{  
  "acknowledged" : true,  
  "shards_acknowledged" : true,  
  "index" : "stconvert"  
}
```

6. 在 Console 界面，执行如下命令，导入数据到 “stconvert” 索引中。

Elasticsearch 低于 7.x 版本的代码示例

```
POST /stconvert/type/1  
{  
  "desc": "國際電視臺"  
}
```

Elasticsearch 大于或等于 7.x 版本和 OpenSearch 集群的代码示例

```
POST /stconvert/_doc/1  
{
```

```
"desc": "國際電視臺"}
}
```

当返回结果信息中“failed”字段的值为“0”时，表示数据导入成功。

7. 在Console界面，执行如下命令，搜索关键字“国际”，并查看搜索结果。

```
GET /stconvert/_search
{
  "query": {
    "match": {
      "desc": "国际"
    }
  }
}
```

搜索结果如下所示。

```
{
  "took": 15,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.5753642,
    "hits": [
      {
        "_index": "stconvert",
        "_type": "type",
        "_id": "1",
        "_score": 0.5753642,
        "_source": {
          "desc": "國際電視臺"
        }
      }
    ]
  }
}
```

2.12.11 重启 Elasticsearch 集群

集群停止工作时，您可通过重启集群恢复运行。

场景描述

重启集群支持快速重启和滚动重启，两种方案的使用说明请参见表2-140。建议在业务空闲时执行重启操作。

表 2-140 重启方案

方案	使用说明	相关文档
快速重启	<ul style="list-style-type: none">所有集群都支持。当选择“节点类型”快速重启时，所选类型的所有节点会一起重启。当选择“节点名称”快速重启时，一次只能重启一个节点。快速重启过程中，集群不可用。	快速重启

方案	使用说明	相关文档
滚动重启	<ul style="list-style-type: none">仅当集群的节点数量（含Master节点、Client节点和冷数据节点）大于等于3时，才支持滚动重启。滚动重启只支持根据“节点类型”进行重启。选择节点类型滚动重启时，所选类型的节点会依次重启。滚动重启过程中，只有正在重启的节点不可用，不在重启过程中的节点可以正常提供服务。当数据量比较大时，滚动重启耗时较长。	滚动重启

前提条件

- 确认集群的“任务状态”没有正在执行中的任务，且集群未被冻结。
- 当集群处于可用状态时，确认集群已停止处理业务数据（如导入数据、搜索数据），否则重启集群时可能导致数据丢失等。

快速重启

- 登录云搜索服务管理控制台。
- 在左侧导航栏，选择对应的集群类型，进入集群管理列表界面。
- 在对应集群的“操作”列中单击“更多>重启”。
- 在“重启集群”页面，选择“快速重启”。
快速重启支持根据“节点类型”或者“节点名称”重启。如果选择“节点类型”，则支持选择多种节点类型同时进行快速重启。如果选择“节点名称”，则一次只能快速重启一个节点。
- 重启集群后，请刷新页面，观察集群状态。重启过程中，集群状态为“处理中”，任务状态为“重启中”。如果集群状态变更为“可用”，表示集群已重启成功。

滚动重启

- 登录云搜索服务管理控制台。
- 在左侧导航栏，选择对应的集群类型，进入集群管理列表界面。
- 在对应集群的“操作”列中单击“更多>重启”。
- 在“重启集群”页面，选择“滚动重启”。
滚动重启支持根据“节点类型”进行重启。如果只需要重启集群中的某些类型的节点时，可以选择需要重启的节点类型。
- 重启集群后，请刷新页面，观察集群状态。重启过程中，集群状态为“处理中”，任务状态为“重启中”。如果集群状态变更为“可用”，表示集群已重启成功。

2.12.12 删除 Elasticsearch 集群

当用户已完成数据搜索业务，无需继续使用某一集群时，可删除集群释放资源。

约束限制

- 删除集群时，会清理集群业务数据，请谨慎操作。
- 当集群启用过快照功能，且OBS桶中创建的快照并未被删除，此时删除集群时，并不会释放这部分备份数据。如果有需要，可以通过OBS桶中存储的快照信息恢复集群，具体参见[集群被删除后是否还能恢复？](#)。

删除按需计费的集群

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表界面。
3. 在对应集群的“操作”列中单击“更多>删除”。
4. 在弹出的确认提示框中，输入需要删除的集群名称，单击“确定”完成集群删除。

删除包年/包月的集群

包年包月计费模式的集群需要通过退订包周期删除集群。集群退订后，资源和数据将会被清空，且无法恢复。请务必确认完成数据备份后再提交退订。

1. 在集群管理页面，选择需要退订的集群。
2. 在操作列单击“更多”>“退订/释放”。
3. 输入需要退订/释放的集群名称，单击“确定”。
进入退订资源页面，可以在该页面核对资源信息以及退费金额。
4. 填写退订原因，勾选相关协议后，单击“退订”。
在弹出确认退订提示信息后，再次单击“退订”。

说明

当集群处于生效中状态时，则走退订流程，此时会产生一个订单进行退费，然后删除集群。当集群处于已过期或者已冻结状态时，则走释放流程，直接删除集群。集群退订的使用说明请参考[费用中心](#)相关描述。

2.13 管理 Elasticsearch 集群索引策略

2.13.1 创建和管理 Elasticsearch 索引策略

本文介绍如何创建和管理Elasticsearch集群的生命周期索引策略。

背景信息

Elasticsearch的ISM (Index State Management) 是一个索引状态管理插件，支持通过索引使用期限、索引大小或文档数等信息的变化来自动触发周期性的管理操作。通过ISM插件可以自定义索引策略，实现自动处理索引的滚动或删除，以优化集群搜索性能或降低存储成本。ISM的使用流程：

1. **创建生命周期策略**：在Kibana创建索引的生命周期策略Policy。
2. **索引关联生命周期策略**：将索引关联上创建的生命周期策略，生成索引策略。
3. **管理索引策略**：支持对索引策略进行策略修改、重启和变更。

ISM更多功能请参见[索引管理官方介绍](#)。

创建的索引生命周期策略常用于以下场景：

- [通过索引生命周期实现时间序列数据滚动索引](#)
- [通过索引生命周期实现索引存算分离](#)

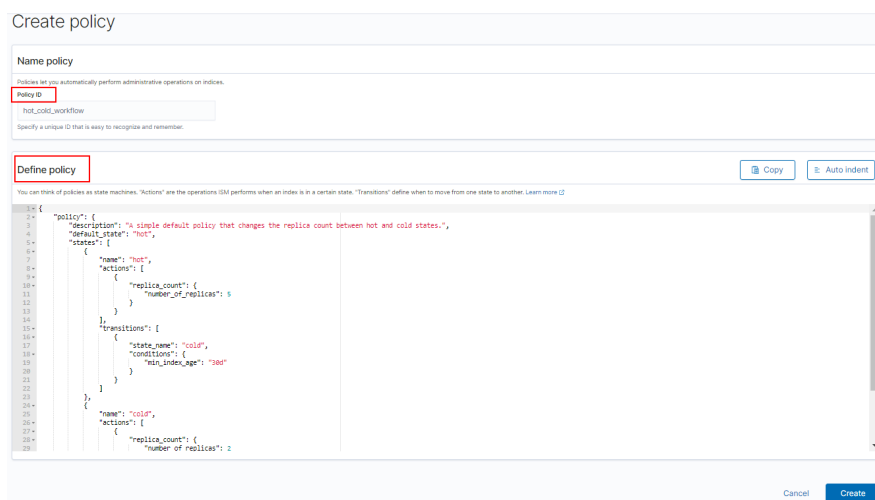
约束限制

- 仅Elasticsearch集群版本≥7.6.2和OpenSearch集群才支持索引状态管理ISM。
- Kibana中可以自定义策略名称，但不支持中文字符。
- 不同版本的Kibana界面可能有所差异，请以实际操作环境为准。本文以Kibana 7.6.2版本作为示例。

创建生命周期策略

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要登录的集群，单击“操作”列中的“Kibana”，登录Kibana页面。
3. 在菜单栏选择“IM”或“Index Management”，进入索引管理页面。
4. 单击“Create policy”，配置生命周期策略。
 - “Policy ID”：自定义策略名称。
 - “Define policy”：基于策略示例，配置自定义策略。

图 2-47 配置策略



5. 单击“Create”，完成策略的创建。

索引关联生命周期策略

生命周期策略创建完成后，可以将此策略关联到一个或多个索引中，匹配该索引模板创建出的索引都将被执行该策略。

索引关联生命周期策略有如下2种方式，任选一种方式配置即可。

- **方式1：Kibana命令行**

在Kibana的“Dev Tools”页面，执行如下命令在索引模板中关联策略ID。

```
PUT _template/<template_name>
{
```

```
"index_patterns": ["index_name-*"],
"settings": {
  "opendistro.index_state_management.policy_id": "policy_id"
}
}
```

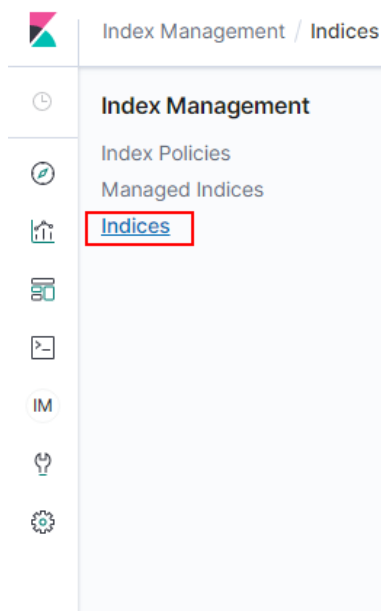
- `template_name`: 需要替换为创建的索引模板名。
- `policy_id`: 需要替换为自定义的策略ID, 即“Policy ID”。

更多创建索引模板的说明可参考[索引模板](#)。

- **方式2: Kibana控制台**

- 在Kibana菜单栏选择“IM”或“Index Management”, 进入索引管理页面。
- 左侧选择“Indices”。

图 2-48 选择 Indices



- 在Indices列表中勾选要关联策略的一个或多个索引。
- 单击右上角的“Apply policy”, 在弹窗中选择要关联的“Policy ID”。

图 2-49 添加索引策略

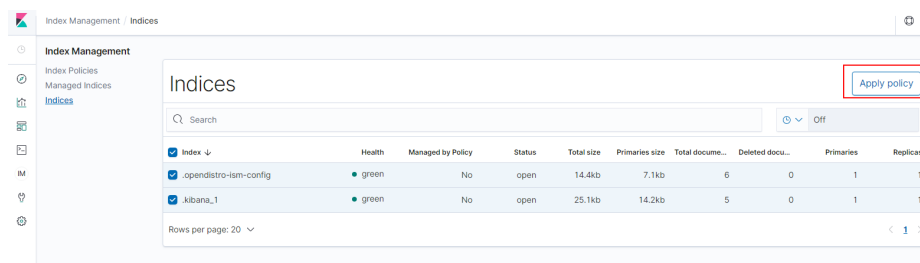
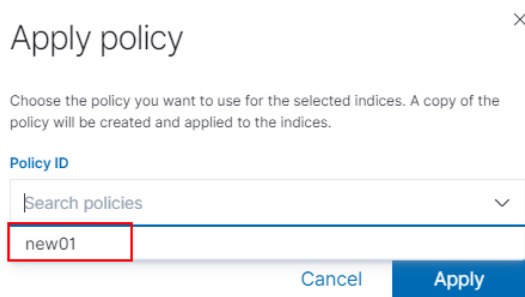


图 2-50 选择索引策略



- e. 单击“Apply”，完成索引关联策略。
当索引关联策略后，ISM会默认创建一个每5分钟运行一次的作业，用于执行策略操作、检查条件并将索引转换为不同的状态。

管理索引策略

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要登录的集群，单击“操作”列中的“Kibana”，登录Kibana页面。
3. 在菜单栏选择“IM”或“Index Management”，进入索引管理页面。
4. 左侧选择“Managed Indices”，进入索引策略管理列表。
此处会展示当前集群已配置的索引策略。
5. 在索引策略管理列表，管理策略。
 - 当索引的策略状态异常时，可以勾选状态异常的索引，单击“Retry policy”，可以重试策略。
 - 当需要更新索引的策略时，可以勾选索引，单击“Change policy”进入变更策略页面，更新策略后，单击“Change”变更策略。
 - 当需要解除索引的策略时，可以勾选索引，单击“Remove policy”，在弹窗确认后单击“Remove”改索引将被解除策略。

具体使用可参考[索引管理官方介绍](#)。

2.13.2 通过索引生命周期实现时间序列数据滚动索引

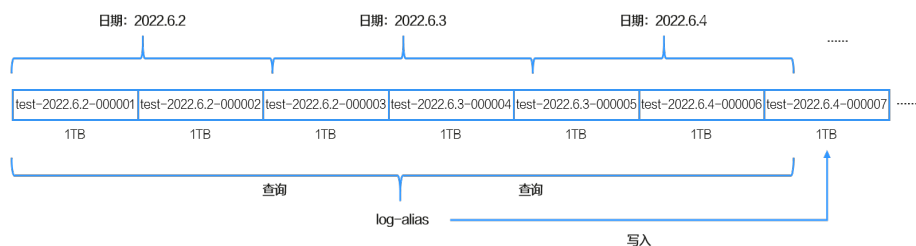
方案概述

对于时间序列数据，随着时间推移数据持续写入，索引会越来越大，通过生命周期管理来定期将数据滚动到新索引，并将历史老索引删除，实现自动滚动索引。

本案例通过配置生命周期策略，当索引的大小达到1TB或索引创建超过1天时，自动滚动生成新索引；当索引创建7天后，关闭数据副本；当索引创建30天后，删除该索引。

假设某个索引，其每天约产生2.4TB的数据，索引别名“log-alias”，其数据在Elasticsearch中的组织形态如下图所示。查询时，指向所有test开头的索引；写入时，指向最新的索引。

图 2-51 log-alias 的组织形态



说明

索引的滚动条件1天是以索引的创建时间来计算的，并不是完整自然日区分的。

前提条件

- CSS集群处于可用状态。
- 使用Elasticsearch 7.6.2或更高版本的集群。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表页面。
3. 在集群列表页面中，单击集群操作列的“Kibana”登录Kibana页面。
4. 在Kibana的左侧导航中选择“Dev Tools”，进入命令执行页面。
5. 创建Rollover生命周期策略“rollover_workflow”。

策略定义：当索引的大小达到1TB或索引创建超过1天时，自动进行滚动；当索引创建7天后，关闭数据副本；当索引创建30天后，删除该索引。

```
PUT _opendistro/_ism/policies/rollover_workflow
```

```
{
  "policy": {
    "description": "rollover test",
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [
          {
            "rollover": {
              "min_size": "1tb",
              "min_index_age": "1d"
            }
          }
        ]
      },
      {
        "state_name": "warm",
        "conditions": {
          "min_index_age": "7d"
        }
      }
    ]
  },
  {
    "name": "warm",
    "actions": [
      {
        "replica_count": {
```



```

        "number_of_replicas": 0
      }
    ],
    "transitions": [
      {
        "state_name": "delete",
        "conditions": {
          "min_index_age": "30d"
        }
      }
    ]
  },
  {
    "name": "delete",
    "actions": [
      {
        "delete": {}
      }
    ]
  }
]
}
}

```

当生命周期策略创建完成后，执行如下命令可以查询策略详情：

```
GET _opendistro/_ism/policies/rollover_workflow
```

6. 新建索引模板 “template_test”。

模板定义：新建的所有 “test” 开头的索引自动关联上Rollover生命周期策略 “rollover_workflow”，并且Rollover时使用 “log_alias” 作为别名。

```
PUT _template/template_test
```

```

{
  "index_patterns": "test*",
  "settings": {
    "number_of_replicas": 1,
    "number_of_shards": 1,
    "opendistro.index_state_management.policy_id": "rollover_workflow",
    "index.opendistro.index_state_management.rollover_alias": "log_alias"
  },
  "mappings": {
    "properties": {
      "name": {
        "type": "text"
      }
    }
  }
}
}

```

表 2-141 参数说明

参数	说明
number_of_shards	索引分片数
number_of_replicas	索引分片副本数
opendistro.index_state_management.policy_id	生命周期的策略名
index.opendistro.index_state_management.rollover_alias	rollover的索引别名

当索引模板创建完成后，可以通过如下命令查询模板详情：

```
GET _template/template_test
```

- 新建一个索引，指定“aliases”，并配置“is_write_index”为“true”。该索引会自动应用索引模板“template_test”，并通过索引模板的配置与生命周期策略“rollover_workflow”相关联，实现当索引的大小达到1TB或索引创建超过1天时，自动进行滚动；当索引创建7天后，关闭数据副本；当索引创建30天后，删除该索引。

如下索引是<test-{now/d}-000001>的URL编码，其创建时默认会带上当天的时间，例如当天日期是“2022.6.02”，创建出来的索引名称为“test-2022.06.02-000001”。

```
PUT %3Ctest-%7Bnow%2Fd%7D-000001%3E
```

```
{
  "aliases": {
    "log_alias": {
      "is_write_index": true
    }
  }
}
```

- 使用别名“log_alias”写入数据，且写入时“log_alias”始终指向最后一个索引。

```
POST log_alias/_bulk
```

```
{ "index": {} }
{ "name": "name1" }
{ "index": {} }
{ "name": "name2" }
{ "index": {} }
{ "name": "name3" }
{ "index": {} }
{ "name": "name4" }
{ "index": {} }
{ "name": "name5" }
{ "index": {} }
{ "name": "name6" }
```

- 查询数据，确认数据是否实现滚动索引。

- 在索引创建一天后查看“test”开头的索引：

```
GET _cat/indices/test*s=i
```

正常情况下会显示至少有两个索引，如下所示：

```
green open test-<日期>-000001 r8ab5NX6T3Ox_hoGUanogQ 1 1 6 0 416b 208b
green open test-<日期>-000002 sfwkVgy8RSSEw7W-xYjM2Q 1 1 0 0 209b 209b
```

其中，“test-<日期>-000001”为7创建的索引，“test-<日期>-000002”为滚动生成的索引。

- 查询别名“log_alias”关联的索引情况：

```
GET _cat/aliases/log_alias?v
```

正常情况下会显示该别名指向多个索引：

alias	index	filter	routing.index	routing.search	is_write_index
log_alias	test-<日期>-000001	-	-	-	false
log_alias	test-<日期>-000002	-	-	-	true

2.13.3 通过索引生命周期实现索引存算分离

方案概述

CSS支持存算分离，即将索引冻结到OBS来降低冷数据的存储成本。本文介绍如何使用索引生命周期管理，在特定的时间自动冻结索引，实现存算分离。

本案例通过配置生命周期策略，实现索引在创建3天后，自动被冻结，数据转储到OBS中；当索引创建7天后，删除该索引。

前提条件

- CSS集群处于可用状态。
- 使用Elasticsearch 7.6.2或更高版本的集群。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表页面。
3. 在集群列表页面中，单击集群操作列的“Kibana”登录Kibana页面。
4. 在Kibana的左侧导航中选择“Dev Tools”，进入命令执行页面。
5. 创建生命周期策略“hot_warm_policy”。

策略定义：当索引创建3天后，自动调用冻结索引API将数据转储到OBS；索引创建7天后，删除该索引。

```
PUT _opendistro/_ism/policies/hot_warm_policy
{
  "policy": {
    "description": "hot warm delete workflow",
    "error_notification": null,
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [],
        "transitions": [
          {
            "state_name": "warm",
            "conditions": {
              "min_index_age": "3d"
            }
          }
        ]
      },
      {
        "name": "warm",
        "actions": [
          {
            "freeze_low_cost": {}
          }
        ],
        "transitions": [
          {
            "state_name": "delete",
            "conditions": {
              "min_index_age": "7d"
            }
          }
        ]
      },
      {
        "name": "delete",
        "actions": [
          {
            "delete": {}
          }
        ],
        "transitions": []
      }
    ]
  }
}
```

6. 新建索引模板“template_hot_warm”。

模板定义：新建的所有“data”开头的索引会自动关联上生命周期策略“hot_warm_policy”。

```
PUT _template/template_hot_warm
{
  "index_patterns": "data*",
  "settings": {
    "number_of_replicas": 5,
    "number_of_shards": 1,
    "opendistro.index_state_management.policy_id": "hot_warm_policy"
  },
  "mappings": {
    "properties": {
      "name": {
        "type": "text"
      }
    }
  }
}
```

表 2-142 参数说明

参数	说明
number_of_shards	索引分片数
number_of_replicas	索引分片副本数
opendistro.index_state_management.policy_id	生命周期的策略名

- 新建一个索引“data-2022-06-06”，该索引会自动应用索引模板“template_hot_warm”，并通过索引模板的配置与生命周期策略“hot_warm_policy”相关联，实现索引在创建3天后冻结，7天后删除。

```
POST data-2022-06-06/_bulk
{"index":{}}
{"name":"name1"}
{"index":{}}
{"name":"name2"}
{"index":{}}
{"name":"name3"}
{"index":{}}
{"name":"name4"}
{"index":{}}
{"name":"name5"}
{"index":{}}
{"name":"name6"}
```

- 查询数据，确认数据是否实现自动存算分离。

- 在索引创建三天后查看冻结的索引：

```
GET _cat/freeze_indices?s=i&v
```

正常情况下会显示3天以前的索引已经被冻结：

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size
green	open	data-2022-06-06	x8ab5NX6T3Ox_xoGUanogQ	1	1	6	0	
7.6kb		3.8kb						

- 在索引创建7天后，查看索引情况，正常情况下7天以前的索引已经被删除。

2.14 Elasticsearch 集群监控与日志管理

2.14.1 Elasticsearch 集群支持的监控指标

2.14.1.1 CES 中 Elasticsearch 集群支持的监控指标

云监控服务CES支持实时监控云搜索服务集群的核心指标，方便用户掌握集群的指标信息，以便及时处理集群的异常状况。

功能说明

云监控服务CES支持实时监控云搜索服务集群的核心指标，方便用户掌握集群的指标信息，以便及时处理集群的异常状况。本文定义了云搜索服务上报云监控服务的集群和节点监控指标的命名空间，监控指标列表和维度定义。用户可以通过云监控服务提供的管理控制台或[API接口](#)来检索云搜索服务产生的监控指标和告警信息。

命名空间

SYS.ES

集群监控指标列表

- 监控的**指标ID**、**指标名称**、**指标含义**以及**取值范围**参见[表2-143](#)。
- 监控的**测量对象**：CSS集群（本文介绍的是Elasticsearch集群和OpenSearch集群的监控指标列表）
- 监控的**监控周期**（原始指标）：1分钟

说明

累计值：从节点启动时开始叠加数值，当节点重启后清零重新累计。

表 2-143 云搜索服务支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
status	集群健康状态	该指标用于统计测量监控对象的状态。	0,1,2,3 <ul style="list-style-type: none"> 0: 集群是100%可用的。 1: 数据是完整的, 部分副本缺失。高可用性在某种程度上弱化, 存在风险, 请及时关注集群情况。 2: 数据缺失, 集群使用时将出现异常。 3: 没有获取到集群状态。 	CSS集群	1分钟
disk_util	磁盘使用率	该指标用于统计测量对象的磁盘使用率。 单位: 百分比	0-100%	CSS集群	1分钟
max_jvm_heap_usage	最大JVM堆使用率	CSS集群中各个节点的JVM堆使用率的最大值。 单位: 百分比。	0-100%	CSS集群	1分钟
max_jvm_young_gc_time	最大JVM Young GC耗时	CSS集群中各个节点的JVM Young GC耗时累计值的最大值。 单位: ms	≥ 0 ms	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_jvm_young_gc_count	最大JVM Young GC次数	CSS集群中各个节点的JVM Young GC次数累计值的最大值。	≥ 0	CSS集群	1分钟
max_jvm_old_gc_time	最大JVM Old GC耗时	CSS集群中各个节点的JVM Old GC耗时累计值的最大值。 单位: ms	≥ 0 ms	CSS集群	1分钟
max_jvm_old_gc_count	最大JVM Old GC次数	CSS集群中各个节点的JVM Old GC次数累计值的最大值。	≥ 0	CSS集群	1分钟
total_fs_size	文件系统总大小	CSS集群的文件系统总大小。 单位: byte	≥ 0 bytes	CSS集群	1分钟
free_fs_size	文件系统可用大小	CSS集群的文件系统可用大小。 单位: byte	≥ 0 bytes	CSS集群	1分钟
max_cpu_usage	最大CPU利用率	CSS集群中各个节点的CPU利用率的最大值。 单位: 百分比	0-100%	CSS集群	1分钟
max_cpu_time_of_jvm_process	最大JVM进程使用的CPU时间	CSS集群中各个节点JVM进程使用CPU的时间累计值的最大值。 单位: ms	≥ 0 ms	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_virtual_memory_size_of_jvm_process	最大JVM进程使用的虚拟内存大小	CSS集群中各个节点JVM进程可使用的虚拟内存大小的最大值。 单位: byte	≥ 0 bytes	CSS集群	1分钟
max_current_opened_http_count	最大当前打开的HTTP连接数	CSS集群中各个节点打开且尚未关闭的HTTP连接数的最大值。	≥ 0	CSS集群	1分钟
max_total_opened_http_count	最大全部打开的HTTP连接数	CSS集群中各个节点打开过的HTTP连接数累计值的最大值。	≥ 0	CSS集群	1分钟
indices_count	索引数量	CSS集群的索引数量。	≥ 0	CSS集群	1分钟
total_shards_count	分片数量	CSS集群的分片数量。	≥ 0	CSS集群	1分钟
primary_shards_count	主分片数量	CSS集群的主分片数量。	≥ 0	CSS集群	1分钟
docs_count	文档数量	CSS集群的文档数量。	≥ 0	CSS集群	1分钟
docs_deleted_count	被删除的文档数量	CSS集群的被删除的文档数量。	≥ 0	CSS集群	1分钟
nodes_count	节点数量	CSS集群的节点数量。	≥ 0	CSS集群	1分钟
data_nodes_count	数据节点数量	CSS集群的数据节点数量。	≥ 0	CSS集群	1分钟
coordinating_nodes_count	协调节点数量	CSS集群的协调节点数量。	≥ 0	CSS集群	1分钟
master_nodes_count	Master节点数量	CSS集群的Master节点数量。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
ingest_node_count	Client节点数量	CSS集群的Client节点数量。	≥ 0	CSS集群	1分钟
max_load_average	最大节点Load值	CSS集群中各个节点在操作系统中1分钟平均排队任务数的最大值。	≥ 0	CSS集群	1分钟
avg_cpu_usage	平均CPU使用率	CSS集群中各节点CPU利用率的平均值。 单位：百分比	0-100%	CSS集群	1分钟
avg_load_average	平均节点Load值	CSS集群中各节点在操作系统中1分钟平均排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_jvm_heap_usage	平均JVM堆使用率	CSS集群中各节点JVM堆内存使用率的平均值。 单位：百分比	0-100%	CSS集群	1分钟
max_open_file_descriptors	已打开的最大文件描述符数	CSS集群中各个节点已打开的文件描述符数的最大值。	≥ 0	CSS集群	1分钟
avg_open_file_descriptors	已打开的平均文件描述符数	CSS集群中各节点已打开的文件描述符数的平均值。	≥ 0	CSS集群	1分钟
sum_max_file_descriptors	最大允许的文件描述符数	CSS集群中各节点最大允许的文件描述符数之和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_open_file_descriptors	已打开的文件描述符数	CSS集群中各节点已打开的文件描述符数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_write_queue	Write队列中总排队任务数	写入线程池中的排队任务数。	≥ 0	CSS集群	1分钟
sum_thread_pool_search_queue	Search队列中总排队任务数	CSS集群中各节点在搜索线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_force_merge_queue	ForceMerge队列中总排队任务数	CSS集群中各节点在强制合并线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_write_rejected	Write队列中总的已拒绝任务数	CSS集群中各节点在写入线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_search_rejected	Search队列中总的已拒绝任务数	CSS集群中各节点在搜索线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_force_merge_rejected	Forcemerge队列中总的已拒绝任务数	CSS集群中各节点在强制合并线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_search_queue	Search队列中最大排队任务数	CSS集群中各个节点在搜索线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_thread_pool_force_merge_queue	ForceMerge队列中最大排队任务数	CSS集群中各个节点在强制合并线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_write_threads	Write线程池总大小	CSS集群中各节点写入线程池的大小之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_search_threads	Search线程池总大小	CSS集群中各节点搜索线程池的大小之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_force_merge_threads	ForceMerge线程池总大小	CSS集群中各节点强制合并线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_write_queue	Write队列中平均排队任务数	CSS集群中各节点在写入线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_search_queue	Search队列中平均排队任务数	CSS集群中各节点在搜索线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_force_merge_queue	ForceMerge队列中平均排队任务数	CSS集群中各节点在强制合并线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_search_threads	Search线程池平均大小	CSS集群中各节点搜索线程池的大小的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_write_threads	Write线程池平均大小	CSS集群中各节点写入线程池的大小的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_thread_pool_force_merge_threads	ForceMerge线程池平均大小	CSS集群中各节点强制合并线程池的大小的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_write_rejected	Write队列中平均已拒绝任务数	CSS集群中各节点写入线程池中的已拒绝任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_flush_queue	Flush队列中总排队任务数	CSS集群中各个节点在Flush线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_flush_rejected	Flush队列中总的已拒绝任务数	CSS集群中各节点在Flush线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_flush_queue	Flush队列中最大排队任务数	CSS集群中各个节点在Flush线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_flush_threads	Flush线程池总大小	CSS集群中各节点Flush线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_flush_queue	Flush队列中平均排队任务数	CSS集群中各节点在Flush线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_flush_threads	Flush线程池平均大小	CSS集群中各节点在Flush线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_generic_queue	Generic队列中总排队任务数	CSS集群中各个节点在Generic线程池中的排队任务数只和。	≥ 0	CSS集群	1分钟
sum_thread_pool_generic_rejecte	Generic队列中总的已拒绝任务数	CSS集群中各个节点在Generic线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_generic_queue	Generic队列中最大排队任务数	CSS集群中各个节点在Generic线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_generic_threads	Generic线程池总大小	CSS集群中各个节点Generic线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_generic_queue	Generic队列中平均排队任务数	CSS集群中各个节点在Generic线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_generic_threads	Generic线程池平均大小	CSS集群中各个节点在Generic线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_management_queue	Management队列中总排队任务数	CSS集群中各个节点在Management线程池中的排队任务数只和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_management_rejected	Management队列中总的已拒绝任务数	CSS集群中各节点在Management线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_management_queue	Management队列中最大排队任务数	CSS集群中各个节点在Management线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_management_threads	Management线程池总大小	CSS集群中各节点Management线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_management_queue	Management队列中平均排队任务数	CSS集群中各节点在Management线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_management_threads	Management线程池平均大小	CSS集群中各节点在Management线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_refresh_queue	Refresh队列中总排队任务数	CSS集群中各个节点在Refresh线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_refresh_rejected	Refresh队列中总的已拒绝任务数	CSS集群中各节点在Refresh线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_thread_pool_refresh_queue	Refresh队列中最大排队任务数	CSS集群中各个节点在Refresh线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_refresh_threads	Refresh线程池总大小	CSS集群中各节点Refresh线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_refresh_queue	Refresh队列中平均排队任务数	CSS集群中各节点在Refresh线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_refresh_threads	Refresh线程池平均大小	CSS集群中各节点在Refresh线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_searcher_queue	OBS Searcher队列中总排队任务数	CSS集群中各个节点在OBS Searcher线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_searcher_rejected	OBS Searcher队列中总的已拒绝任务数	CSS集群中各节点在OBS Searcher线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_obs_searcher_queue	OBS Searcher队列中最大排队任务数	CSS集群中各个节点在OBS Searcher线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_obs_searcher_threads	OBS Searcher线程池总大小	CSS集群中各节点OBS Searcher线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_searcher_queue	OBS Searcher队列中平均排队任务数	CSS集群中各节点在OBS Searcher线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_searcher_threads	OBS Searcher线程池平均大小	CSS集群中各节点在OBS Searcher线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_queue	OBS队列中总排队任务数	CSS集群中各个节点在OBS线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_rejected	OBS队列中总的已拒绝任务数	CSS集群中各节点在OBS线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_obs_queue	OBS队列中最大排队任务数	CSS集群中各个节点在OBS线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_threads	OBS线程池总大小	CSS集群中各节点OBS线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_queue	OBS队列中平均排队任务数	CSS集群中各节点在OBS线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_thread_pool_obs_threads	OBS线程池平均大小	CSS集群中各节点在OBS线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_upload_queue	OBS Upload队列中总排队任务数	CSS集群中各个节点在OBS Upload线程池中的排队任务数只和。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_upload_rejected	OBS Upload队列中总的已拒绝任务数	CSS集群中各节点在OBS Upload线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_obs_upload_queue	OBS Upload队列中最大排队任务数	CSS集群中各个节点在OBS Upload线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_upload_threads	OBS Upload线程池总大小	CSS集群中各节点OBS Upload线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_upload_queue	OBS Upload队列中平均排队任务数	CSS集群中各节点在OBS Upload线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_upload_threads	OBS Upload线程池平均大小	CSS集群中各节点在OBS Upload线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_obs_download_queue	OBS Download 队列中总排队任务数	CSS集群中各个节点在 OBS Download线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_download_rejected	OBS Download 队列中总的已拒绝任务数	CSS集群中各节点在 OBS Download线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_obs_download_queue	OBS Download 队列中最大排队任务数	CSS集群中各个节点在 OBS Download线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_download_threads	OBS Download 线程池总大小	CSS集群中各节点 OBS Download线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_download_queue	OBS Download 队列中平均排队任务数	CSS集群中各节点在 OBS Download线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_download_threads	OBS Download 线程池平均大小	CSS集群中各节点在 OBS Download线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
min_free_fs_size	最小可用存储空间	CSS集群中各个节点可用存储空间的最小值。 单位: byte	≥ 0 bytes	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_jvm_ol d_gc_count	JVM老年代 平均GC次 数	CSS集群中各 个节点“老 年代”垃圾 回收的运行 次数的累计 值的平均 值。	≥ 0	CSS集 群	1分钟
avg_jvm_ol d_gc_time	JVM老年代 平均GC时 间	CSS集群中各 个节点执行 “老年代” 垃圾回收所 花费的时间 累计值的平 均值。 单位: ms	≥ 0 ms	CSS集 群	1分钟
avg_jvm_yo ung_gc_cou nt	JVM年轻代 平均GC次 数	CSS集群中各 个节点“年 轻代”垃圾 回收的运行 次数的累计 值的平均 值。	≥ 0	CSS集 群	1分钟
avg_jvm_yo ung_gc_tim e	JVM年轻代 平均GC时 间	CSS集群中各 个节点执行 “年轻代” 垃圾回收所 花费的时间 累计值的平 均值。 单位: ms	≥ 0 ms	CSS集 群	1分钟
avg_max_fi le_descript ors	最大允许的 文件描述符 数-平均值	CSS集群中各 节点最大允 许的文件描 述符数的平 均值。	≥ 0	CSS集 群	1分钟
avg_mem_f ree_in_byte s	平均可用内 存空间	CSS集群中各 节点未使用 的内存容量 的平均值。 单位: byte	≥ 0 bytes	CSS集 群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_mem_free_percent	平均可用内存比例	CSS集群中各节点未使用的内存比例的平均值。 单位：百分比	0-100%	CSS集群	1分钟
avg_mem_used_in_bytes	平均已用内存空间	CSS集群中各节点已使用的内存容量的平均值。 单位：byte	≥ 0 bytes	CSS集群	1分钟
avg_mem_used_percent	平均已用内存比例	CSS集群中各节点已使用的内存比例的平均值。 单位：百分比	0-100%	CSS集群	1分钟
max_mem_free_in_bytes	最大可用内存空间	CSS集群中各个节点未使用的内存容量的最大值。 单位：byte	≥ 0 bytes	CSS集群	1分钟
max_mem_free_percent	最大可用内存比例	CSS集群中各个节点未使用的内存比例的最大值。 单位：百分比	0-100%	CSS集群	1分钟
max_mem_used_in_bytes	最大已用内存空间	CSS集群中各个节点已使用的内存容量的最大值。 单位：byte	≥ 0 bytes	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_mem_used_percent	最大已用内存比例	CSS集群中各个节点已使用的内存比例的最大值。 单位：百分比	0-100%	CSS集群	1分钟
sum_jvm_oldest_gc_count	JVM老年代总GC次数	CSS集群中各个节点“老年代”垃圾回收的运行次数的累计值之和。	≥ 0	CSS集群	1分钟
sum_jvm_oldest_gc_time	JVM老年代总GC时间	CSS集群中各个节点执行“老年代”垃圾回收所花费的时间累计值之和。 单位：ms	≥ 0 ms	CSS集群	1分钟
sum_jvm_young_gc_count	JVM年轻代总GC次数	CSS集群中各个节点“年轻代”垃圾回收的运行次数的累计值之和。	≥ 0	CSS集群	1分钟
sum_jvm_young_gc_time	JVM年轻代总GC时间	CSS集群中各个节点执行“年轻代”垃圾回收所花费的时间累计值之和。 单位：ms	≥ 0 ms	CSS集群	1分钟
sum_current_opened_http_count	当前已打开HTTP连接数	CSS集群中各个节点打开且尚未关闭的HTTP连接数之和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_total_opened_http_count	历史已打开HTTP连接数	CSS集群中各个节点打开过的HTTP连接数累计值之和。	≥ 0	CSS集群	1分钟
IndexingLatency	平均索引延迟	分片完成索引操作所需的平均时间。 单位: ms	≥ 0 ms	CSS集群	1分钟
IndexingRate	平均索引速率	入库TPS, 集群每秒平均索引操作数。	≥ 0	CSS集群	1分钟
SearchLatency	平均查询延迟	分片完成搜索操作所需的平均时间。 单位: ms。	≥ 0 ms	CSS集群	1分钟
SearchRate	平均查询速率	查询QPS, 集群每秒平均查询操作数。	≥ 0	CSS集群	1分钟
task_max_running_time	最大Task运行时长	该指标为集群运行的所有task中, 运行时长最长的task的耗时	≥ 0 ms	CSS集群	1分钟
number_of_pending_tasks	Pending Task排队任务数	CSS集群中待Master处理的task的排队任务数。	≥ 0	CSS集群	1分钟

节点监控指标列表

- 监控的**指标ID**、**指标名称**、**指标含义**以及**取值范围**参见[表2-144](#)。
- 监控的**测量对象**: CSS集群 - 云服务节点
- 监控的**监控周期** (原始指标): 1分钟

说明

累计值: 从节点启动时开始叠加数值, 当节点重启后清零重新累计。

表 2-144 云搜索服务节点支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
jvm_heap_usage	JVM堆使用率	节点JVM堆内存使用率。 单位：百分比	0-100 %	CSS集群 - 云服务节点	1分钟
cpu_usage	CPU利用率	CPU利用率。 单位：百分比	0-100 %	CSS集群 - 云服务节点	1分钟
load_average	节点Load值	操作系统中1分钟平均排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
open_file_descriptors	已打开的文件描述符数	节点已打开的文件描述符数。	≥ 0	CSS集群 - 云服务节点	1分钟
max_file_descriptors	最大允许的文件描述符数	最大允许的文件描述符数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_write_queue	Write队列中总排队任务数	写入线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_search_queue	Search队列中总排队任务数	搜索线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_force_merge_queue	ForceMerge队列中总排队任务数	强制合并线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_write_rejected	Write队列中总的已拒绝任务数	写入线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_search_rejected	Search队列中总的已拒绝任务数	搜索线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
thread_pool_force_merge_rejected	ForceMerge队列中总的已拒绝任务数	强制合并线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_write_threads	Write线程池总大小	写入线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_search_threads	Search线程池总大小	搜索线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_force_merge_threads	ForceMerge线程池总大小	强制合并线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_flush_queue	Flush队列中总排队任务数	Flush线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_flush_rejected	Flush队列中总的已拒绝任务数	Flush线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_flush_threads	Flush线程池总大小	Flush线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_generic_queue	Generic队列中总排队任务数	Generic线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_generic_rejected	Generic队列中总的已拒绝任务数	Generic线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_generic_threads	Generic线程池总大小	Generic线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
thread_pool_management_queue	Management队列中总排队任务数	Management线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_management_rejected	Management队列中总的已拒绝任务数	Management线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_management_threads	Management线程池总大小	Management线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_refresh_queue	Refresh队列中总排队任务数	Refresh线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_refresh_rejected	Refresh队列中总的已拒绝任务数	Refresh线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_refresh_threads	Refresh线程池总大小	Refresh线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_searcher_queue	OBS Searcher队列中总排队任务数	OBS Searcher线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_searcher_rejected	OBS Searcher队列中总的已拒绝任务数	OBS Searcher线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_searcher_threads	OBS Searcher线程池总大小	OBS Searcher线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_queue	OBS队列中总排队任务数	OBS线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
thread_pool_obs_rejected	OBS队列中总的已拒绝任务数	OBS线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_threads	OBS线程池总大小	OBS线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_upload_queue	OBS Upload队列中总排队任务数	OBS Upload线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_upload_rejected	OBS Upload队列中总的已拒绝任务数	OBS Upload线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_upload_threads	OBS Upload线程池总大小	OBS Upload线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_download_queue	OBS Download队列中总排队任务数	OBS Download线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_download_rejected	OBS Download队列中总的已拒绝任务数	OBS Download线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_download_threads	OBS Download线程池总大小	OBS Download线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
free_fs_size	文件系统可用大小	CSS集群的文件系统可用大小。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟
total_fs_size	文件系统总大小	CSS集群的文件系统总大小。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
jvm_old_gc_count	JVM老年代总GC次数	“老年代”垃圾回收的运行次数。	≥ 0	CSS集群 - 云服务节点	1分钟
jvm_old_gc_time	JVM老年代总GC时间	执行“老年代”垃圾回收所花费的时间。 单位: ms	≥ 0 ms	CSS集群 - 云服务节点	1分钟
jvm_young_gc_count	JVM年轻代总GC次数	“年轻代”垃圾回收的运行次数。	≥ 0	CSS集群 - 云服务节点	1分钟
jvm_young_gc_time	JVM年轻代GC时间	执行“年轻代”垃圾回收所花费的时间。 单位: ms	≥ 0 ms	CSS集群 - 云服务节点	1分钟
mem_free_in_bytes	可用内存空间	节点未使用的内存容量。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟
mem_free_percent	可用内存比例	节点未使用的内存比例。	≥ 0	CSS集群 - 云服务节点	1分钟
mem_used_in_bytes	已用内存空间	节点已使用的内存容量。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟
current_opened_http_count	当前已打开HTTP连接数	节点当前打开的HTTP连接数。	≥ 0	CSS集群 - 云服务节点	1分钟
total_opened_http_count	全部打开的HTTP连接数	节点全部打开的HTTP连接数。	≥ 0	CSS集群 - 云服务节点	1分钟

维度

表 2-145 维度说明

Key	Value
cluster_id	CSS集群
cluster_id,instance_id	CSS集群节点

2.14.1.2 Elasticsearch 集群内核支持的监控指标

表 2-146 集群内核支持的监控指标

指标名称	描述	支持的集群版本	相关文档
search请求的P99时延	开源Elasticsearch在监控search请求时仅提供了平均延迟指标，这不足以准确反映集群的实际搜索性能。为了改进这一点，CSS服务新增了P99延迟监控特性，能够有效地监控集群search请求的第99百分位延迟。	Elasticsearch 7.6.2、Elasticsearch 7.10.2	监控P99时延
集群HTTP状态码	通过HTTP对Elasticsearch的访问会收到响应和相应的状态码，但开源的Elasticsearch原生不支持对这些状态码进行统计，导致无法精确掌握接口调用的具体状况，用户也无法通过监控了解集群的整个请求情况。为了改进这一点，CSS服务新增了HTTP状态码监控功能，能够获取HTTP状态码统计信息。	Elasticsearch 7.6.2、Elasticsearch 7.10.2	监控HTTP状态码
集群索引	索引监控功能用以监测集群索引的运行状态和变化趋势，评估业务使用情况，并能够协助及时发现并处理潜在风险，确保集群稳定运行。索引监控会收集索引的stats信息，并将其存储在集群的监控索引（索引名称是“monitoring-eye-css-[yyyy-mm-dd]”）。默认情况下，这些监控索引将保留一周时间，过期会直接删除。	Elasticsearch 7.6.2、Elasticsearch 7.10.2	配置Elasticsearch集群索引监控

2.14.2 配置 Elasticsearch 集群监控

2.14.2.1 使用 CES 监控 Elasticsearch 集群

云搜索服务支持通过云监控服务CES对已创建成功的集群进行日常监控。配置集群监控后，就可以在CES管理控制台直观查看集群的监控指标数据。

配置集群监控的操作流程：

1. **配置告警规则**：根据实际业务需要对监控指标设置自定义告警规则，当监控指标超过设置的阈值时，会以邮箱、HTTP、HTTPS等方式通知您。
2. **配置监控对象**：为集群或集群中某个节点配置监控指标。
3. **查看监控指标**：您可以选择不同的监控时间周期，查看监控指标数据变化情况。

前提条件

- 集群处于“可用”或“处理中”状态。
- 集群正常运行时长大于10分钟。

推荐配置的监控指标

- 监控集群的**cpu、jvm使用情况**，推荐重点配置如下监控指标：平均JVM堆使用率、最大JVM堆使用率、平均CPU使用率、最大CPU利用率。
- 监控集群的**写入、查询延迟和吞吐量情况**，推荐重点配置如下监控指标：平均索引延迟、平均索引速率、平均查询延迟、平均查询速率。
- 监控集群的**写入、查询的排队队列和拒绝情况**，推荐重点配置如下监控指标：Write队列中总排队任务数、Search队列中总排队任务数、Write队列中总的已拒绝任务数、Search队列中总的已拒绝任务数。

配置告警规则

1. 登录云监控服务CES管理控制台。
2. 左侧导航栏选择“告警 > 告警规则”，进入告警规则列表页面。
3. 在“资源类型”列，筛选“云搜索服务”，查看是否有满足要求的告警规则。

图 2-52 查看告警规则

<input type="checkbox"/>	名称/ID	资源类型 	监控对象
<input type="checkbox"/>	alarm- al1654480875465a...	云搜索服务	CSS集群 指定资源

如果没有，请参考云监控服务CES的[创建告警规则和通知](#)，新建CSS服务的告警规则。其中，“资源类型”和“维度”参数的填写说明请参见[表2-147](#)，其他参数可以根据CES服务的参数说明自定义。

表 2-147 告警内容的配置说明

参数	参数解释	配置说明
资源类型	配置告警规则监控的服务名称。	选择云搜索服务。
维度	用于指定告警规则对应指标的维度名称。	CSS支持2个维度，根据实际需要选择维度。 <ul style="list-style-type: none">● CSS集群：以集群维度指定告警规则。● CSS集群 - 云服务节点：以集群中的某个节点维度指定告警规则。

配置监控对象

1. 参考云监控服务CES的[创建监控面板](#)，创建一个监控面板。如果已有监控面板，可以跳过该步骤。
2. 参考云监控服务CES的[添加监控视图](#)，添加CSS监控视图。

其中，“资源类型”和“维度”参数的填写说明请参见表2-148，其他参数可以根据CES服务的参数说明自定义。

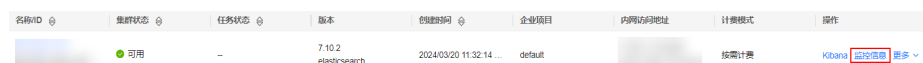
表 2-148 监控视图的配置说明

参数	参数解释	配置说明
资源类型	添加监控视图的服务名称。	选择云搜索服务。
维度	指定监控的维度名称。	CSS支持2个维度，根据实际需要选择维度。 <ul style="list-style-type: none">● CSS集群：以集群维度监控。● CSS集群 - 云服务节点：以集群中的某个节点维度监控。

查看监控指标

1. 登录云搜索服务管理控制台。
2. 选择目标集群，单击操作列“监控信息”查看监控指标。

图 2-53 查看监控信息



名称ID	集群状态	任务状态	版本	创建时间	企业项目	内网访问地址	计费模式	操作
	可用	-	7.10.2 elasticsearch	2024/03/20 11:32:14...	default		按需计费	Kibana 监控信息 更多

3. 选择待查看的时间段页签。
4. 查看监控指标数据。

2.14.2.2 配置 Elasticsearch 集群内核监控

场景描述

表 2-149 集群内核监控介绍

集群内核监控增强	描述	支持的集群版本	相关文档
P99时延监控	开源Elasticsearch在监控search请求时仅提供了平均延迟指标，这不足以准确反映集群的实际搜索性能。为了改进这一点，CSS服务新增了P99延迟监控特性，能够有效地监控集群search请求的第99百分位延迟。	Elasticsearch 7.6.2、Elasticsearch 7.10.2、	监控P99时延
HTTP状态码	通过HTTP对Elasticsearch的访问会收到响应和相应的状态码，但开源的Elasticsearch原生不支持对这些状态码进行统计，导致无法精确掌握接口调用的具体状况，用户也无法通过监控了解集群的整个请求情况。为了改进这一点，CSS服务新增了HTTP状态码监控功能，能够获取HTTP状态码统计信息。	Elasticsearch 7.6.2、Elasticsearch 7.10.2、	监控HTTP状态码

访问集群

本文通过Kibana访问集群，介绍如何使用存算分离。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，选择需要进行存算分离的集群，单击操作列“Kibana”，登录Kibana页面。
3. 选择左侧导航栏的“Dev Tools”，进入操作页面。

监控 P99 时延

执行以下命令，获取当前集群的P99时延。

```
GET /search/stats/percentile
```

返回示例如下：

```
{
  "overall" : {
    "1.0" : 2.0,
    "5.0" : 2.0,
    "25.0" : 6.5,
    "50.0" : 19.5,
    "75.0" : 111.0,
    "95.0" : 169.0,
    "99.0" : 169.0,
    "max" : 169.0,
  }
}
```

```
"min" : 2.0
},
"last_one_day" : {
  "1.0" : 2.0,
  "5.0" : 2.0,
  "25.0" : 6.5,
  "50.0" : 19.5,
  "75.0" : 111.0,
  "95.0" : 169.0,
  "99.0" : 169.0,
  "max" : 169.0,
  "min" : 2.0
},
"latest" : {
  "1.0" : 26.0,
  "5.0" : 26.0,
  "25.0" : 26.0,
  "50.0" : 26.0,
  "75.0" : 26.0,
  "95.0" : 26.0,
  "99.0" : 26.0,
  "max" : 26.0,
  "min" : 26.0
}
}
```

表 2-150 返回参数说明

参数	说明
overall	表示集群从启动到当前时间的统计数值。
last_one_day	表示最近一天的统计数值。
latest	表示从上次重置到当前时间的统计数值。

📖 说明

- P99时延的计算是近似值，不提供精确值，越靠近两端的统计值越准确，即99%的时延比50%的时延更准确。
- 当重启集群时，P99时延的数据将被清空，会从集群重启成功后重新计算。

监控集群P99时延的命令还可以设置其他配置项。

- 可以自定义监控的百分比数值。

例如执行如下命令，显示1%、50%和90%的时延数据。

```
GET /search/stats/percentile
{
  "percents": [1, 50, 90]
}
```

- 支持手动重置latest统计值。

执行以下命令，重置latest统计值。

```
POST /search/stats/reset
```

如下所示，返回“ok”，表示重置成功。

```
{
  "nodes" : {
    "css-c9c8-ess-esn-1-1" : "ok"
  }
}
```


监控 HTTP 状态码

根据集群版本选择监控HTTP状态码的命令。

- Elasticsearch 7.6.2集群执行如下命令，获取HTTP状态码统计。

```
GET /_nodes/http_stats
```

返回样例:

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0 },
  "cluster_name" : "css-8362",
  "nodes" : {
    "F9IFdQPARaOJI7oL7HOXtQ" : {
      "http_code" : {
        "200" : 114,
        "201" : 5,
        "429" : 0,
        "400" : 7,
        "404" : 0,
        "405" : 0
      }
    }
  }
}
```

- Elasticsearch 7.10.2集群执行如下命令，获取HTTP状态码统计。

```
GET _nodes/stats/http
```

返回样例:

```
{
  .....
  "cluster_name" : "css-2985",
  "nodes" : {
    .....
    "omvR9_W-TsGApraMApREjA" : {
      .....
      "http" : {
        "current_open" : 4,
        "total_opened" : 37,
        "http_code" : {
          "200" : 25,
          "201" : 7,
          "429" : 0,
          "400" : 3,
          "404" : 0,
          "405" : 0
        }
      }
    }
  }
}
```

2.14.2.3 配置 Elasticsearch 集群索引监控

场景描述

索引监控功能用以监测集群索引的运行状态和变化趋势，评估业务使用情况，并能够协助及时发现并处理潜在风险，确保集群稳定运行。索引监控会收集索引的stats信息，并将其存储在集群的监控索引（索引名称是“monitoring-eye-css-[yyyy-mm-dd]”）。默认情况下，这些监控索引将保留一周时间，过期会直接删除。

约束限制

- 仅Elasticsearch 7.6.2和Elasticsearch 7.10.2集群支持索引监控。
- 索引监控不会监控以“monitoring-eye-css-*”开头的索引，避免使用的该索引名称匹配到监控索引。
- 索引监控不允许删除“monitoring-eye-css-*”的index pattern，否则会导致监控图表异常。

访问集群

本文通过Kibana访问集群，介绍如何使用存算分离。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，选择需要进行存算分离的集群，单击操作列“Kibana”，登录Kibana页面。
3. 选择左侧导航栏的“Dev Tools”，进入操作页面。

启用索引监控

1. 执行如下命令，打开索引监控开关。

```
PUT _cluster/settings
{
  "persistent": {
    "css.monitoring.index.enabled": "true"
  }
}
```

2. 当需要监控单个索引时，可以执行如下命令。

```
PUT _cluster/settings
{
  "persistent": {
    "css.monitoring.index.enabled": "true",
    "css.monitoring.index.interval": "30s",
    "css.monitoring.index.indices": ["index_name"],
    "css.monitoring.history.duration": "3d"
  }
}
```

表 2-151 配置项说明

配置项	类型	说明
css.monitoring.index.enabled	Boolean	索引监控的控制开关，设置为true将打开集群索引监控功能。 默认值：false
css.monitoring.index.interval	Time	索引监控采集数据的时间间隔。 最小值：1s 默认值：10s

配置项	类型	说明
css.monitoring.index.indices	String	索引监控的索引名称，默认监控所有索引，可以配置监控单个索引，也可以配置通配符监控某一类索引。 例如： <ul style="list-style-type: none"> “css.monitoring.index.indices”: [“index_name”]”表示只监控“index_name”索引。 “css.monitoring.index.indices”: [“log_*”]”表示监控以“log_”开头的索引。 “css.monitoring.index.indices”: [“index1”, “index2”]”表示监控“index1”和“index2”两个索引。 默认值：*（表示监控所有索引）
css.monitoring.history.duration	Time	监控数据存储的索引保留时间，默认保存一周。 最小值：1d 默认值：7d

查看索引读写流量

当集群启用了索引监控后，支持查询一段时间内的索引读写流量。

- 执行如下命令，查看所有索引读写流量。
GET /_cat/monitoring
- 执行如下命令，查看某一索引的读写流量。
GET /_cat/monitoring/{indexname}
{indexName}为需要查看读写流量的索引名称。
- 执行如下命令，查看索引不同时间段的读写流量。
GET _cat/monitoring?begin=1650099461000
GET _cat/monitoring?begin=2022-04-16T08:57:41
GET _cat/monitoring?begin=2022-04-16T08:57:41&end=2022-04-17T08:57:41

表 2-152 配置项说明

配置项	是否必选	说明
indexname	否	索引名称。 说明 不支持查看系统索引，以“.”开头的索引是系统索引。
begin	否	查看监控的起始时间，UTC时间，默认是当前时间的前5分钟。 支持时间格式：strict_date_optional_time epoch_millis 默认值：当前时间减去5分钟。

配置项	是否必选	说明
end	否	查看监控的结束时间，UTC时间，默认是当前时间。 支持时间格式：strict_date_optional_time epoch_millis 默认值：当前时间。

返回信息示例：

```
index begin end status pri rep init unassign docs.count docs.deleted store.size
pri.store.size delete.rate indexing.rate search.rate
test 2022-03-25T09:46:53.765Z 2022-03-25T09:51:43.767Z yellow 1 1 0 1 9 0 5.9kb
5.9kb 0/s 0/s 0/s
```

表 2-153 返回信息的参数说明

参数	说明
index	索引名称。
begin	查看监控数据的起始时间。
end	查看监控数据的结束时间。
status	查询监控时间间隔内的索引状态。
pri	查询监控时间间隔内的索引的shard数量。
rep	查询监控时间间隔内的索引副本数量。
init	查询监控时间间隔内的索引的初始化数量。
unassign	查询监控时间间隔内的索引的未分配数量。
docs.count	查询监控时间间隔内的文档数量。
docs.deleted	查询监控时间间隔内的文档删除数量。
store.size	查询监控时间间隔内存储的索引大小。
pri.store.size	查询监控时间间隔内的索引主分片的大小。
delete.rate	监控时间间隔内的索引每秒删除数量。
indexing.rate	监控时间间隔内的索引每秒写入数量。
search.rate	监控时间间隔内的索引每秒查询数量。

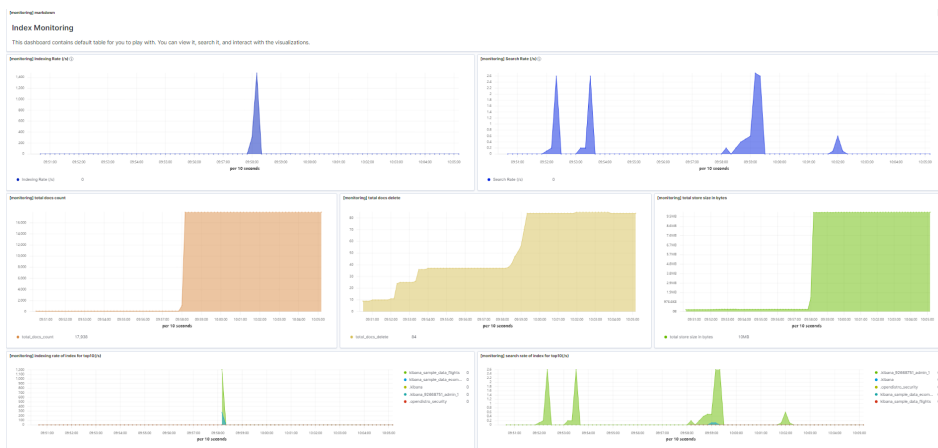
通过 Kibana 图表查看索引监控

为了方便查看索引的监控信息，CSS服务的索引监控预置了Kibana的Dashboard和Visualizations图表。

1. 登录云搜索服务管理控制台。

2. 在“集群管理”页面，选择已创建的集群，单击操作列“Kibana”，登录Kibana界面。
3. 通过**预置Dashboard**图表查看索引监控。
 - a. 单击Kibana左侧导航栏的“Dashboard”，打开Dashboard界面。
 - b. 单击 “[Monitoring] Index monitoring Dashboard” 查看预置Dashboard。

图 2-54 预置 dashboard 图表



预置dashboard展示集群的每秒读写数量和每秒读写数量前10的索引情况。

表 2-154 预置图表说明

图表名称	说明
[monitoring] markdown	Markdown图表，简单说明dashboard表达的内容。
[monitoring] Indexing Rate (/s)	集群每秒写入文档数。
[monitoring] Search Rate (/s)	集群每秒查询次数。
[monitoring] indexing rate of index for top10	每秒写入文档数最多的Top10索引。
[monitoring] search rate of index for top10	每秒查询次数最多的Top10索引。
[monitoring] total docs count	集群总文档数量变化。
[monitoring] total docs delete	集群总的删除文档数量变化。
[monitoring] total store size in bytes	集群总文档所占用的存储空间变化。
[monitoring] indices store_size for top10	占用存储空间最多的Top10索引。

图表名称	说明
[monitoring] indices docs_count for top10	文档数量最多的Top10索引。
[monitoring] indexing time in millis of index for top10(ms)	单位时间内文档写入时延最大的Top10索引 (ms) 。
[monitoring] search query time in millis of index for top10(ms)	单位时间内索引查询时间最大的Top10索引 (ms) 。
[monitoring] segment count of index for top10	索引Segment数量最多的Top10索引。
[monitoring] segment memory in bytes of index for top10	索引Segment占用堆内存最大的Top10索引。

4. 通过自定义Visualizations图表查看索引监控。

索引监控定期把index/stats信息存储到monitoring-eyes索引中，通过使用kibana图表功能可以绘制自定义的图表。

下面以文档数据为例介绍通过图表查看文档数量的变化趋势的操作步骤。

- 单击Kibana左侧导航栏的“Visualize”进入图形列表。
- 单击“Create visualization”，选择“TSVB”，进入TSVB绘图页面。
- 设置图表参数，查看Visualizations。

如图所示，在“Data”页签，配置参数。

- “Field”选择“index_stats primaries.docs.count”表示主分片的文档数量。
- “Aggregation”选择“Derivative”表示聚合桶之间的差异，“Units”设置为“1s”表示使用每秒代替速率。
- “Aggregation”选择“Positive Only”避免数字重置后出现负数。
- 当需要区分不同索引的情况时，则将“Group by”设置为“Terms”，“By”设置为“index_stats.index”，最终得到的聚合结果将按照索引名称区分。
- 当需要查看不同时间段的数据时，则需要将时间聚合间隔设置好，否则将导致数据显示不完整。在“Panel options”页签，将“Interval”设置为“1m”或“30m”，即可调整“timestamp”的时间间隔。

图 2-55 TSVB 绘图页面

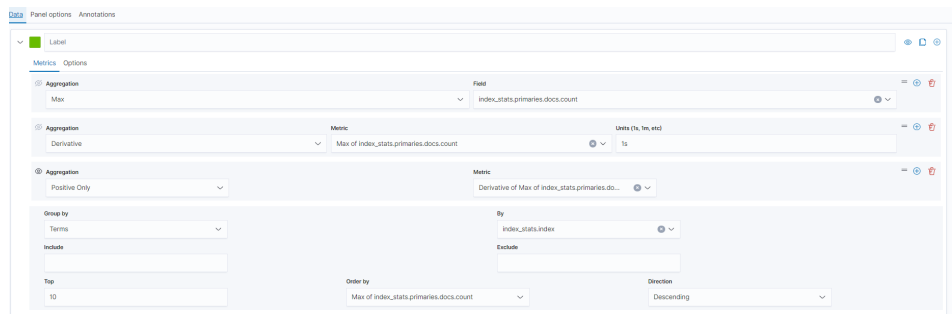
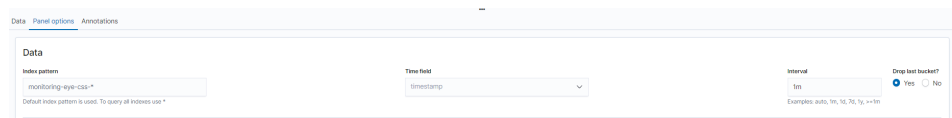
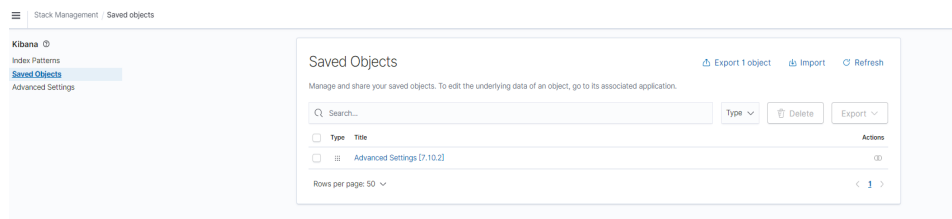


图 2-56 设置时间间隔



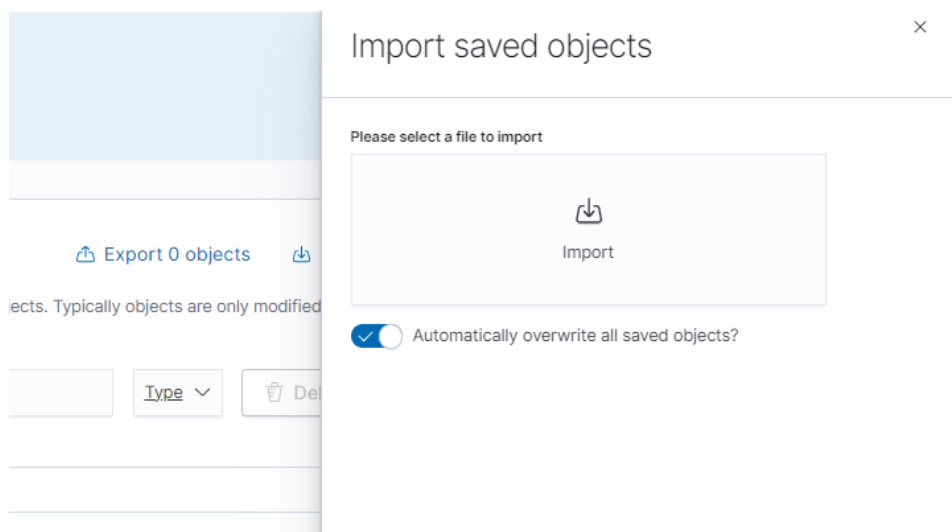
5. 当出现索引监控图表不显示时，可以在Kibana重新导入图表加载监控视图。
 - a. 参考[kibana-monitor配置文件](#)创建“monitoring-kibana.ndjson”文件。
 - b. 在Kibana页面，选择“Management > Stack Management > Saved objects”。

图 2-57 选择 Saved objects



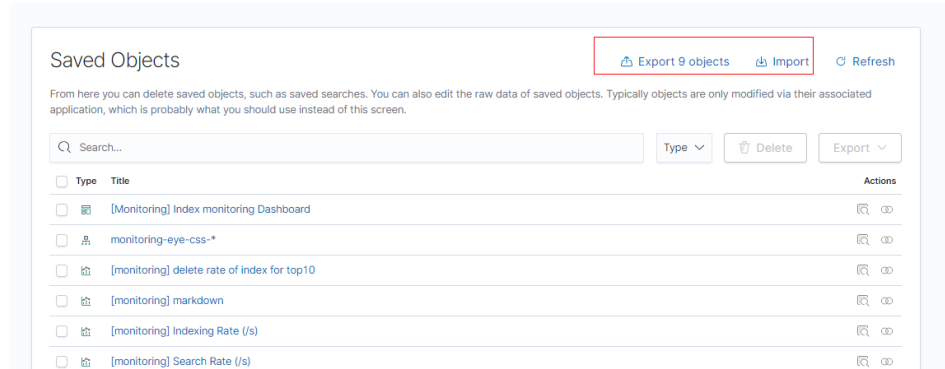
- c. 单击“Import”，选择上传5.a中创建的“monitoring-kibana.ndjson”文件。

图 2-58 上传文件



d. 上传完成选择“done”，索引监控图表即导入成功。

图 2-59 索引监控图表导入成功



kibana-monitor 配置文件

kibana-monitor 配置文件内容如下，建议保存为“monitoring-kibana.ndjson”文件。

```
{
  "attributes": {
    "description": "",
    "kibanaSavedObjectMeta": {
      "searchSourceJSON": {}
    },
    "title": "[monitoring] segment memory in bytes of index for top10",
    "uiStateJSON": {},
    "version": 1,
    "visState": {
      "title": "[monitoring] segment memory in bytes of index for top10",
      "type": "timeseries",
      "series": [
        {
          "id": "61ca57f0-469d-11e7-af02-69e470af7417",
          "color": "#68BC00",
          "split_mode": "terms",
          "split_color_mode": "kibana",
          "metrics": [
            {
              "id": "61ca57f2-469d-11e7-af02-69e470af7417",
              "type": "max",
              "field": "index_stats.total.segments.memory_in_bytes",
              "separate_axis": 0,
              "axis_position": "right",
              "formatter": "bytes",
              "chart_type": "line",
              "line_width": 1,
              "point_size": 1,
              "fill": 0.5,
              "stacked": "none",
              "label": "segments memory in bytes",
              "type": "timeseries",
              "terms_field": "index_stats.index",
              "terms_order_by": "61ca57f2-469d-11e7-af02-69e470af7417",
              "time_field": "timestamp",
              "index_pattern": "monitoring-eye-css-*",
              "interval": "",
              "axis_position": "left",
              "axis_formatter": "number",
              "axis_scale": "normal",
              "show_legend": 1,
              "show_grid": 1,
              "tooltip_mode": "show_all",
              "default_timefield": "timestamp",
              "isModelInvalid": false
            }
          ],
          "id": "3ae5d820-6628-11ed-8cd7-973626cf6f70",
          "references": []
        }
      ],
      "type": "visualization",
      "updated_at": "2022-12-01T12:41:01.165Z",
      "version": "WzlwNiwYXQ=="
    }
  },
  "attributes": {
    "description": "",
    "kibanaSavedObjectMeta": {
      "searchSourceJSON": {}
    },
    "title": "[monitoring] segment count of index for top10",
    "uiStateJSON": {},
    "version": 1,
    "visState": {
      "aggs": [],
      "params": {
        "monitoring-eye-css-*": {
          "default_timefield": "timestamp",
          "filter": {
            "language": "kuery",
            "query": ""
          },
          "id": "61ca57f0-469d-11e7-af02-69e470af7417",
          "index_pattern": "monitoring-eye-css-*",
          "interval": "",
          "isModelInvalid": false,
          "series": [
            {
              "axis_position": "right",
              "chart_type": "line",
              "color": "rgba(231,102,76,1)",
              "fill": 0.5,
              "formatter": "number",
              "id": "61ca57f1-469d-11e7-af02-69e470af7417",
              "label": "segment count of index for top10",
              "line_width": 1,
              "metrics": [
                {
                  "field": "index_stats.total.segments.count",
                  "id": "61ca57f2-469d-11e7-af02-69e470af7417",
                  "type": "max",
                  "point_size": 1,
                  "separate_axis": 0,
                  "split_color_mode": "kibana",
                  "split_mode": "terms",
                  "stacked": "none",
                  "terms_field": "index_stats.index",
                  "terms_order_by": "61ca57f2-469d-11e7-af02-69e470af7417",
                  "type": "timeseries",
                  "show_grid": 1,
                  "show_legend": 1,
                  "time_field": "timestamp",
                  "tooltip_mode": "show_all",
                  "type": "timeseries"
                }
              ],
              "title": "[monitoring] segment count of index for top10",
              "type": "metrics"
            }
          ],
          "id": "45d571c0-6626-11ed-8cd7-973626cf6f70",
          "references": []
        }
      ],
      "type": "visualization",
      "updated_at": "2022-12-01T12:41:01.165Z",
      "version": "WzlwNywyXQ=="
    }
  },
  "attributes": {
    "description": "",
    "kibanaSavedObjectMeta": {
      "searchSourceJSON": {}
    },
    "title": "[monitoring] markdown",
    "uiStateJSON": {},
    "version": 1,
    "visState": {
      "title": "[monitoring] markdown",
      "type": "markdown",
      "params": {
        "fontSize": 12,
        "openLinksInNewTab": false,
        "markdown": "### Index Monitoring\n\nThis dashboard contains default table for you to play with. You can view it, search it, and interact with the visualizations."
      },
      "aggs": [],
      "id": "b2811c70-a5f1-11ec-9a68-ada9d754c566",
      "references": []
    },
    "type": "visualization",
    "updated_at": "2022-12-01T12:41:01.165Z",
    "version": "WzlwOCwyXQ=="
  },
  "attributes": {
    "description": "number of document being indexing for primary and replica shards",
    "kibanaSavedObjectMeta": {
      "searchSourceJSON": {}
    },
    "title": "[monitoring] Indexing Rate (/s)",
    "uiStateJSON": {},
    "version": 1,
    "visState": {
      "title": "[monitoring] Indexing Rate (/s)",
      "type": "metrics",
      "params": {
        "id": "61ca57f0-469d-11e7-af02-69e470af7417",
        "type": "timeseries",
        "series": [
          {
            "id": "61ca57f1-469d-11e7-af02-69e470af7417",
            "color": "rgba(0,32,188,1)",
            "split_mode": "everything",
            "metrics": [
              {
                "id": "61ca57f2-469d-11e7-af02-69e470af7417",
                "type": "max",
                "field": "indices_stats_all.total.indexing_index_total",
                "unit": "1s",
                "id": "fed72db0-a5f8-11ec-aa10-992297d21a2e",
                "type": "derivative",
                "field": "61ca57f2-469d-11e7-af02-69e470af7417",
                "unit":

```



```
\\":{},"id":"14b66420-a5f9-11ec-aa10-992297d21a2e\\",\\"type\\":\\"positive_only\\",\\"field\\":\\"fed72db0-a5f8-11ec-aa10-992297d21a2e\\",\\"separate_axis\\":0,\\"axis_position\\":\\"right\\",\\"formatter\\":\\"number\\",\\"chart_type\\":\\"line\\",\\"line_width\\":1,\\"point_size\\":1,\\"fill\\":0.5,\\"stacked\\":\\"none\\",\\"label\\":\\"Indexing Rate (/s)\\",\\"type\\":\\"timeseries\\",\\"split_color_mode\\":\\"rainbow\\",\\"hidden\\":false}},\\"time_field\\":\\"timestamp\\",\\"index_pattern\\":\\"monitoring-eye-css-*\\",\\"interval\\":\\"\\",\\"axis_position\\":\\"left\\",\\"axis_formatter\\":\\"number\\",\\"axis_scale\\":\\"normal\\",\\"show_legend\\":1,\\"show_grid\\":1,\\"default_index_pattern\\":\\"monitoring-eye-css-*\\",\\"default_timefield\\":\\"timestamp\\",\\"isModelInvalid\\":false,\\"legend_position\\":\\"bottom\\",\\"aggs\\":[]}},\\"id\\":\\"de4f8ab0-a5f8-11ec-9a68-ada9d754c566\\",\\"references\\":[]},\\"type\\":\\"visualization\\",\\"updated_at\\":\\"2022-12-01T12:41:01.165Z\\",\\"version\\":\\"WzlwOSwyXQ==\\"}{\\"attributes\\":{\\"description\\":\\"number of search request being executed in primary and replica shards\\",\\"kibanaSavedObjectMeta\\":{\\"searchSourceJSON\\":\\"\\"},\\"title\\":\\"[monitoring] Search Rate (/s)\\",\\"uiStateJSON\\":\\"\\"},\\"version\\":1,\\"visState\\":{\\"title\\":\\"[monitoring] Search Rate (/s)\\",\\"type\\":\\"metrics\\",\\"params\\":{\\"id\\":\\"61ca57f0-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"timeseries\\",\\"series\\":{\\"id\\":\\"61ca57f1-469d-11e7-af02-69e470af7417\\",\\"color\\":\\"rgba(0,33,224,1)\\",\\"split_model\\":\\"everything\\",\\"metrics\\":{\\"id\\":\\"61ca57f2-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"max\\",\\"field\\":\\"indices_stats_all.total.search.query_total\\",\\"unit\\":\\"1s\\",\\"id\\":\\"b1093ac0-a5f7-11ec-aa10-992297d21a2e\\",\\"type\\":\\"derivative\\",\\"field\\":\\"61ca57f2-469d-11e7-af02-69e470af7417\\",\\"unit\\":\\"\\",\\"id\\":\\"c17db930-a5f7-11ec-aa10-992297d21a2e\\",\\"type\\":\\"positive_only\\",\\"field\\":\\"b1093ac0-a5f7-11ec-aa10-992297d21a2e\\",\\"separate_axis\\":0,\\"axis_position\\":\\"right\\",\\"formatter\\":\\"number\\",\\"chart_type\\":\\"line\\",\\"line_width\\":1,\\"point_size\\":1,\\"fill\\":0.5,\\"stacked\\":\\"none\\",\\"split_color_mode\\":\\"rainbow\\",\\"label\\":\\"Search Rate (/s)\\",\\"type\\":\\"timeseries\\",\\"filter\\":{\\"query\\":\\"\\",\\"language\\":\\"kuery\\",\\"time_field\\":\\"timestamp\\",\\"index_pattern\\":\\"monitoring-eye-css-*\\",\\"interval\\":\\"\\",\\"axis_position\\":\\"left\\",\\"axis_formatter\\":\\"number\\",\\"axis_scale\\":\\"normal\\",\\"show_legend\\":1,\\"show_grid\\":1,\\"default_index_pattern\\":\\"monitoring-eye-css-*\\",\\"default_timefield\\":\\"timestamp\\",\\"isModelInvalid\\":false,\\"legend_position\\":\\"bottom\\",\\"aggs\\":[]}},\\"id\\":\\"811df7a0-a5f8-11ec-9a68-ada9d754c566\\",\\"references\\":[]},\\"type\\":\\"visualization\\",\\"updated_at\\":\\"2022-12-01T12:41:01.165Z\\",\\"version\\":\\"WzlxMCwyXQ==\\"}{\\"attributes\\":{\\"description\\":\\"\\",\\"kibanaSavedObjectMeta\\":{\\"searchSourceJSON\\":\\"\\"},\\"title\\":\\"[monitoring] total docs count\\",\\"uiStateJSON\\":\\"\\"},\\"version\\":1,\\"visState\\":{\\"title\\":\\"[monitoring] total docs count\\",\\"type\\":\\"metrics\\",\\"aggs\\":[],\\"params\\":{\\"id\\":\\"61ca57f0-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"timeseries\\",\\"series\\":{\\"id\\":\\"61ca57f1-469d-11e7-af02-69e470af7417\\",\\"color\\":\\"rgba(218,139,69,1)\\",\\"split_model\\":\\"everything\\",\\"split_color_mode\\":\\"kibana\\",\\"metrics\\":{\\"unit\\":\\"\\",\\"id\\":\\"61ca57f2-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"max\\",\\"field\\":\\"indices_stats_all.total.docs.count\\",\\"separate_axis\\":0,\\"axis_position\\":\\"right\\",\\"formatter\\":\\"number\\",\\"chart_type\\":\\"line\\",\\"line_width\\":1,\\"point_size\\":1,\\"fill\\":0.5,\\"stacked\\":\\"none\\",\\"label\\":\\"total docs count\\",\\"type\\":\\"timeseries\\",\\"time_field\\":\\"timestamp\\",\\"index_pattern\\":\\"monitoring-eye-css-*\\",\\"interval\\":\\"\\",\\"axis_position\\":\\"left\\",\\"axis_formatter\\":\\"number\\",\\"axis_scale\\":\\"normal\\",\\"show_legend\\":1,\\"show_grid\\":1,\\"tooltip_model\\":\\"show_all\\",\\"default_index_pattern\\":\\"monitoring-eye-css-*\\",\\"default_timefield\\":\\"timestamp\\",\\"isModelInvalid\\":false,\\"legend_position\\":\\"bottom\\",\\"aggs\\":[]}},\\"id\\":\\"eea89780-664b-11ed-8cd7-973626cf6f70\\",\\"references\\":[]},\\"type\\":\\"visualization\\",\\"updated_at\\":\\"2022-12-01T12:41:01.165Z\\",\\"version\\":\\"WzlxMSwyXQ==\\"}{\\"attributes\\":{\\"description\\":\\"\\",\\"kibanaSavedObjectMeta\\":{\\"searchSourceJSON\\":\\"\\"},\\"title\\":\\"[monitoring] total docs delete\\",\\"uiStateJSON\\":\\"\\"},\\"version\\":1,\\"visState\\":{\\"title\\":\\"[monitoring] total docs delete\\",\\"type\\":\\"metrics\\",\\"aggs\\":[],\\"params\\":{\\"id\\":\\"61ca57f0-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"timeseries\\",\\"series\\":{\\"id\\":\\"61ca57f1-469d-11e7-af02-69e470af7417\\",\\"color\\":\\"rgba(214,191,87,1)\\",\\"split_model\\":\\"everything\\",\\"split_color_mode\\":\\"kibana\\",\\"metrics\\":{\\"id\\":\\"61ca57f2-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"max\\",\\"field\\":\\"indices_stats_all.total.docs.deleted\\",\\"separate_axis\\":0,\\"axis_position\\":\\"right\\",\\"formatter\\":\\"number\\",\\"chart_type\\":\\"line\\",\\"line_width\\":1,\\"point_size\\":1,\\"fill\\":0.5,\\"stacked\\":\\"none\\",\\"label\\":\\"total docs delete\\",\\"type\\":\\"timeseries\\",\\"hidden\\":false}},\\"time_field\\":\\"timestamp\\",\\"index_pattern\\":\\"monitoring-eye-css-*\\",\\"interval\\":\\"\\",\\"axis_position\\":\\"left\\",\\"axis_formatter\\":\\"number\\",\\"axis_scale\\":\\"normal\\",\\"show_legend\\":1,\\"show_grid\\":1,\\"tooltip_model\\":\\"show_all\\",\\"default_index_pattern\\":\\"monitoring-eye-css-*\\",\\"default_timefield\\":\\"timestamp\\",\\"isModelInvalid\\":false,\\"drop_last_bucket\\":1,\\"legend_position\\":\\"bottom\\",\\"aggs\\":[]}},\\"id\\":\\"cfbb4e20-664c-11ed-8cd7-973626cf6f70\\",\\"references\\":[]},\\"type\\":\\"visualization\\",\\"updated_at\\":\\"2022-12-01T12:41:01.165Z\\",\\"version\\":\\"WzlxMiwYXQ==\\"}{\\"attributes\\":{\\"description\\":\\"\\",\\"kibanaSavedObjectMeta\\":{\\"searchSourceJSON\\":\\"\\"},\\"title\\":\\"[monitoring] total store size in bytes\\",\\"uiStateJSON\\":\\"\\"},\\"version\\":1,\\"visState\\":{\\"title\\":\\"[monitoring] total store size in bytes\\",\\"type\\":\\"metrics\\",\\"aggs\\":[],\\"params\\":{\\"id\\":\\"61ca57f0-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"timeseries\\",\\"series\\":{\\"id\\":\\"61ca57f1-469d-11e7-af02-69e470af7417\\",\\"color\\":\\"#68BC00\\",\\"split_model\\":\\"everything\\",\\"split_color_mode\\":\\"kibana\\",\\"metrics\\":{\\"id\\":\\"61ca57f2-469d-11e7-af02-69e470af7417\\",\\"type\\":\\"max\\",\\"field\\":\\"indices_stats_all.total.store.size_in_bytes\\",\\"separate_axis\\":0,\\"axis_position\\":\\"right\\",\\"formatter\\":\\"bytes\\",\\"chart_type\\":\\"line\\",\\"line_width\\":1,\\"point_size\\":1,\\"fill\\":0.5,\\"stacked\\":\\"none\\",\\"label\\":\\"total store size in bytes\\",\\"type\\":\\"timeseries\\",\\"time_field\\":\\"timestamp\\",\\"index_pattern\\":\\"monitoring-eye-css-*\\",\\"interval\\":\\"\\",\\"axis_position\\":\\"left\\",\\"axis_formatter\\":\\"number\\",\\"axis_scale\\":\\"normal\\",\\"show_legend\\":1,\\"show_grid\\":1,\\"tooltip_model\\":\\"show_all\\",\\"default_index_pattern\\":\\"monitoring-eye-css-*\\",\\"default_timefield\\":\\"timestamp\\",\\"isModelInvalid\\":false,\\"drop_last_bucket\\":1,\\"legend_position\\":\\"bottom\\",\\"aggs\\":[]}},\\"id\\":\\"cfbb4e20-664c-11ed-8cd7-973626cf6f70\\",\\"references\\":[]}
```

```
\"false\",\"legend_position\":\"bottom\", \"background_color_rules\":{ \"id\": \"7712e550-664f-11e7-8b5d-8db37e5b4cc4\"}, \"bar_color_rules\":{ \"id\": \"77680a30-664f-11e7-8b5d-8db37e5b4cc4\"}}}, \"id\": \"c7f72ae0-664e-11e7-8cd7-973626cf6f70\", \"references\": [], \"type\": \"visualization\", \"updated_at\": \"2022-12-01T12:41:01.165Z\", \"version\": \"WzlxMywyXQ==\"}, { \"attributes\": { \"description\": \"\", \"kibanaSavedObjectMeta\": { \"searchSourceJSON\": {}}, \"title\": \"[monitoring] indexing rate of index for top10(/s)\", \"uiStateJSON\": {}}, \"version\": 1, \"visState\": { \"title\": \"[monitoring] indexing rate of index for top10(/s)\", \"type\": \"metrics\", \"aggs\": [], \"params\": { \"id\": \"61ca57f0-469d-11e7-af02-69e470af7417\", \"type\": \"timeseries\", \"series\": { \"id\": \"61ca57f1-469d-11e7-af02-69e470af7417\", \"color\": \"#68BC00\", \"split_mode\": \"terms\", \"metrics\": [{ \"id\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"type\": \"max\", \"field\": \"index_stats.total.indexing.index_total\", \"unit\": \"1s\", \"id\": \"541ed8f0-a5ee-11ec-aa10-992297d21a2e\", \"type\": \"derivative\", \"field\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"unit\": \"\", \"id\": \"67ec1f50-a5ee-11ec-aa10-992297d21a2e\", \"type\": \"positive_only\", \"field\": \"541ed8f0-a5ee-11ec-aa10-992297d21a2e\"}], \"separate_axis\": 0, \"axis_position\": \"right\", \"formatter\": \"number\", \"chart_type\": \"line\", \"line_width\": 1, \"point_size\": 1, \"fill\": 0.5, \"stacked\": \"none\", \"label\": \"indexing_rate\", \"type\": \"timeseries\", \"split_filters\": { \"color\": \"#68BC00\", \"id\": \"81004200-a5ee-11ec-aa10-992297d21a2e\", \"filter\": { \"query\": \"\", \"language\": \"kuery\"}}, \"filter\": { \"query\": \"\", \"language\": \"kuery\"}}, \"terms_field\": \"index_stats.index\", \"terms_order_by\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"terms_size\": \"10\", \"terms_direction\": \"desc\", \"split_color_mode\": \"rainbow\"}}, \"time_field\": \"timestamp\", \"index_pattern\": \"monitoring-eye-css-*\", \"interval\": \"\", \"axis_position\": \"left\", \"axis_formatter\": \"number\", \"axis_scale\": \"normal\", \"show_legend\": 1, \"show_grid\": 1, \"default_index_pattern\": \"monitoring-eye-css-*\", \"default_timefield\": \"timestamp\", \"isModelInvalid\": \"false\", \"tooltip_model\": \"show_all\"}}}, \"id\": \"943b3e00-a5ef-11ec-9a68-ada9d754c566\", \"references\": [], \"type\": \"visualization\", \"updated_at\": \"2022-12-01T12:41:01.165Z\", \"version\": \"WzlxNCwyXQ==\"}, { \"attributes\": { \"description\": \"\", \"kibanaSavedObjectMeta\": { \"searchSourceJSON\": {}}, \"title\": \"[monitoring] search rate of index for top10(/s)\", \"uiStateJSON\": {}}, \"version\": 1, \"visState\": { \"title\": \"[monitoring] search rate of index for top10(/s)\", \"type\": \"metrics\", \"aggs\": [], \"params\": { \"id\": \"61ca57f0-469d-11e7-af02-69e470af7417\", \"type\": \"timeseries\", \"series\": { \"id\": \"61ca57f1-469d-11e7-af02-69e470af7417\", \"color\": \"rgba(99,157,12,1)\", \"split_mode\": \"terms\", \"metrics\": { \"id\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"type\": \"max\", \"field\": \"index_stats.total.search.query_total\", \"unit\": \"1s\", \"id\": \"fdfdfad0-a5ef-11ec-aa10-992297d21a2e\", \"type\": \"derivative\", \"field\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"unit\": \"\", \"id\": \"0aaa26a0-a5f0-11ec-aa10-992297d21a2e\", \"type\": \"positive_only\", \"field\": \"fdfdfad0-a5ef-11ec-aa10-992297d21a2e\"}], \"separate_axis\": 0, \"axis_position\": \"right\", \"formatter\": \"number\", \"chart_type\": \"line\", \"line_width\": 1, \"point_size\": 1, \"fill\": 0.5, \"stacked\": \"none\", \"label\": \"search_rate\", \"type\": \"timeseries\", \"terms_field\": \"index_stats.index\", \"terms_order_by\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"split_color_mode\": \"rainbow\"}}, \"time_field\": \"timestamp\", \"index_pattern\": \"monitoring-eye-css-*\", \"interval\": \"\", \"axis_position\": \"left\", \"axis_formatter\": \"number\", \"axis_scale\": \"normal\", \"show_legend\": 1, \"show_grid\": 1, \"default_index_pattern\": \"monitoring-eye-css-*\", \"default_timefield\": \"timestamp\", \"isModelInvalid\": \"false\", \"tooltip_model\": \"show_all\"}}}, \"id\": \"ab503550-a5ef-11ec-9a68-ada9d754c566\", \"references\": [], \"type\": \"visualization\", \"updated_at\": \"2022-12-01T12:41:01.165Z\", \"version\": \"WzlxNSwyXQ==\"}, { \"attributes\": { \"description\": \"\", \"kibanaSavedObjectMeta\": { \"searchSourceJSON\": {}}, \"title\": \"[monitoring] indices store_size for top10\", \"uiStateJSON\": {}}, \"version\": 1, \"visState\": { \"title\": \"[monitoring] indices store_size for top10\", \"type\": \"metrics\", \"aggs\": [], \"params\": { \"id\": \"61ca57f0-469d-11e7-af02-69e470af7417\", \"type\": \"timeseries\", \"series\": { \"id\": \"38474c50-a5f5-11ec-aa10-992297d21a2e\", \"color\": \"#68BC00\", \"split_mode\": \"terms\", \"metrics\": { \"id\": \"38474c51-a5f5-11ec-aa10-992297d21a2e\", \"type\": \"max\", \"field\": \"index_stats.total.store.size_in_bytes\"}}, \"separate_axis\": 0, \"axis_position\": \"right\", \"formatter\": \"bytes\", \"chart_type\": \"line\", \"line_width\": 1, \"point_size\": 1, \"fill\": 0.5, \"stacked\": \"none\", \"label\": \"store_size for index\", \"type\": \"timeseries\", \"terms_field\": \"index_stats.index\", \"terms_order_by\": \"38474c51-a5f5-11ec-aa10-992297d21a2e\", \"filter\": { \"query\": \"\", \"language\": \"kuery\"}}, \"split_color_mode\": \"rainbow\"}}, \"time_field\": \"timestamp\", \"index_pattern\": \"monitoring-eye-css-*\", \"interval\": \"\", \"axis_position\": \"left\", \"axis_formatter\": \"number\", \"axis_scale\": \"normal\", \"show_legend\": 1, \"show_grid\": 1, \"default_index_pattern\": \"monitoring-eye-css-*\", \"default_timefield\": \"timestamp\", \"isModelInvalid\": \"false\", \"filter\": { \"query\": \"\", \"language\": \"kuery\"}}, \"bar_color_rules\": { \"id\": \"7d9d3cb0-a5f5-11ec-aa10-992297d21a2e\"}}, \"tooltip_model\": \"show_all\"}}}, \"id\": \"c78119a0-a5f5-11ec-9a68-ada9d754c566\", \"references\": [], \"type\": \"visualization\", \"updated_at\": \"2022-12-01T12:41:01.165Z\", \"version\": \"WzlxNiwyXQ==\"}, { \"attributes\": { \"description\": \"\", \"kibanaSavedObjectMeta\": { \"searchSourceJSON\": {}}, \"title\": \"[monitoring] search query time in millis of index for top10(ms)\", \"uiStateJSON\": {}}, \"version\": 1, \"visState\": { \"title\": \"[monitoring] search query time in millis of index for top10(ms)\", \"type\": \"metrics\", \"aggs\": [], \"params\": { \"axis_formatter\": \"number\", \"axis_max\": \"\", \"axis_min\": \"\", \"axis_position\": \"left\", \"axis_scale\": \"normal\", \"default_index_pattern\": \"monitoring-eye-css-*\", \"default_timefield\": \"timestamp\", \"id\": \"61ca57f0-469d-11e7-af02-69e470af7417\", \"index_pattern\": \"monitoring-eye-css-*\", \"interval\": \"\", \"isModelInvalid\": \"false\", \"series\": { \"axis_position\": \"right\", \"chart_type\": \"line\", \"color\": \"#68BC00\", \"fill\": 0.5, \"formatter\": \"number\", \"id\": \"61ca57f1-469d-11e7-af02-69e470af7417\", \"label\": \"index_query_time_in_millis\", \"line_width\": 1, \"metrics\": { \"field\": \"index_stats.total.search.query_time_in_millis\", \"id\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"type\": \"max\", \"unit\": \"1s\", \"id\": \"42c92b10-6645-11ed-925a-6de90846447d\", \"type\": \"derivative\", \"field\"
```

```
\"61ca57f2-469d-11e7-af02-69e470af7417\"},\"point_size\":1,\"separate_axis\":0,\"split_color_mode\n\": \"kibana\", \"split_mode\": \"terms\", \"stacked\": \"none\", \"terms_field\": \"index_stats.index\n\", \"terms_order_by\": \"61ca57f2-469d-11e7-af02-69e470af7417\", \"type\": \"timeseries\"}, \"show_grid\n\": 1, \"show_legend\": 1, \"time_field\": \"timestamp\", \"tooltip_mode\": \"show_all\", \"type\": \"timeseries\n\", \"background_color\": null, \"filter\": {\"query\": \"\", \"language\": \"kuery\"}, \"legend_position\": \"right\n\"}}\", \"id\": \"c8109100-6627-11ed-8cd7-973626cf6f70\", \"references\":\n[], \"type\": \"visualization\", \"updated_at\": \"2022-12-01T12:41:01.165Z\", \"version\": \"WzlxNywyXQ==\"}\n{\"attributes\": {\"description\": \"\", \"hits\": 0, \"kibanaSavedObjectMeta\": {\"searchSourceJSON\": {\"query\":\n\": {\"language\": \"kuery\", \"query\": \"\", \"filter\": []}}, \"optionsJSON\": {\"hidePanelTitles\": false, \"useMargins\n\": true}}, \"panelsJSON\": [{\"gridData\": {\"x\": 0, \"y\": 0, \"w\": 48, \"h\": 5, \"i\": \"971ed6c6-81b9-491b-9f08-\ne3ae9c382abd\"}, \"panelIndex\": \"971ed6c6-81b9-491b-9f08-e3ae9c382abd\", \"embeddableConfig\":\n\": {\"panelRefName\": \"panel_0\"}, \"gridData\": {\"x\": 0, \"y\": 5, \"w\": 24, \"h\": 15, \"i\n\": \"5a6982e7-0c6c-4733-8a2d-e4c57cdf7397\"}, \"panelIndex\": \"5a6982e7-0c6c-4733-8a2d-\ne4c57cdf7397\"}, \"embeddableConfig\": {\"panelRefName\": \"panel_1\"}, \"gridData\": {\"x\": 24, \"y\": 5, \"w\n\": 24, \"h\": 15, \"i\": \"662476f4-739c-4a05-858c-2ee8230cf410\"}, \"panelIndex\n\": \"662476f4-739c-4a05-858c-2ee8230cf410\", \"embeddableConfig\": {\"panelRefName\": \"panel_2\"},\n\": {\"gridData\": {\"x\": 0, \"y\": 20, \"w\": 16, \"h\": 15, \"i\": \"d89c38e2-33f3-4592-\nb503-20460a6a7a57\"}, \"panelIndex\": \"d89c38e2-33f3-4592-b503-20460a6a7a57\", \"embeddableConfig\":\n\": {\"panelRefName\": \"panel_3\"}, \"gridData\": {\"x\": 16, \"y\": 20, \"w\": 16, \"h\": 15, \"i\n\": \"1f693b49-79fa-4807-94e8-0c12f51e54f8\"}, \"panelIndex\n\": \"1f693b49-79fa-4807-94e8-0c12f51e54f8\", \"embeddableConfig\": {\"panelRefName\": \"panel_4\"},\n\": {\"gridData\": {\"x\": 32, \"y\": 20, \"w\": 16, \"h\": 15, \"i\": \"616b143d-74e9-4dac-98ba-5849536f0fba\n\"}, \"panelIndex\": \"616b143d-74e9-4dac-98ba-5849536f0fba\", \"embeddableConfig\": {\"panelRefName\n\": \"panel_5\"}, \"gridData\": {\"x\": 0, \"y\": 35, \"w\": 24, \"h\": 11, \"i\": \"cfa82f27-1b8d-49ba-a7b9-\nd8809d3b258c\"}, \"panelIndex\": \"cfa82f27-1b8d-49ba-a7b9-d8809d3b258c\", \"embeddableConfig\":\n\": {\"panelRefName\": \"panel_6\"}, \"gridData\": {\"x\": 24, \"y\": 35, \"w\": 24, \"h\": 11, \"i\": \"135d13eb-\naab6-43ca-9029-7d26e91d90e3\"}, \"panelIndex\": \"135d13eb-\naab6-43ca-9029-7d26e91d90e3\", \"embeddableConfig\": {\"panelRefName\": \"panel_7\"}, \"gridData\": {\"x\n\": 0, \"y\": 46, \"w\": 24, \"h\": 11, \"i\": \"28a77de1-9110-49e8-b273-724f880b1653\"}, \"panelIndex\n\": \"28a77de1-9110-49e8-b273-724f880b1653\", \"embeddableConfig\": {\"panelRefName\": \"panel_8\"},\n\": {\"gridData\": {\"x\": 24, \"y\": 46, \"w\": 24, \"h\": 11, \"i\": \"80ece867-cf23-4935-bfbc-430afa51bcca\n\"}, \"panelIndex\": \"80ece867-cf23-4935-bfbc-430afa51bcca\", \"embeddableConfig\": {\"panelRefName\n\": \"panel_9\"}, \"gridData\": {\"x\": 0, \"y\": 57, \"w\": 24, \"h\": 11, \"i\": \"2ba970aa-c9c4-491b-bdd3-\nc1b1ee9bc8d3\"}, \"panelIndex\": \"2ba970aa-c9c4-491b-bdd3-c1b1ee9bc8d3\", \"embeddableConfig\":\n\": {\"panelRefName\": \"panel_10\"}, \"gridData\": {\"x\": 24, \"y\": 57, \"w\": 24, \"h\": 11, \"i\": \"f2e1b6ab-\nddf7-492e-aaca-9460f11aa4aa\"}, \"panelIndex\": \"f2e1b6ab-ddf7-492e-aaca-9460f11aa4aa\n\", \"embeddableConfig\": {\"panelRefName\": \"panel_11\"}, \"gridData\": {\"x\": 0, \"y\": 68, \"w\": 24, \"h\n\": 11, \"i\": \"dd14182d-d8b9-47f2-bf36-6cba3b09586c\"}, \"panelIndex\": \"dd14182d-d8b9-47f2-\nbf36-6cba3b09586c\", \"embeddableConfig\": {\"panelRefName\": \"panel_12\"}, \"gridData\": {\"x\": 24, \"y\n\": 68, \"w\": 24, \"h\": 11, \"i\": \"a47f9333-52b7-49b7-8cac-f470cf405131\"}, \"panelIndex\n\": \"a47f9333-52b7-49b7-8cac-f470cf405131\", \"embeddableConfig\": {\"panelRefName\n\": \"panel_13\"}}], \"timeRestore\": false, \"title\": \"[Monitoring] Index monitoring\nDashboard\", \"version\": 1, \"id\": \"524eb000-a5f2-11ec-9a68-ada9d754c566\", \"references\": [{\"id\": \"b2811c70-\na5f1-11ec-9a68-ada9d754c566\", \"name\": \"panel_0\", \"type\": \"visualization\"}, {\"id\": \"de4f8ab0-a5f8-11ec-9a68-\nada9d754c566\", \"name\": \"panel_1\", \"type\": \"visualization\"}, {\"id\": \"811df7a0-a5f8-11ec-9a68-\nada9d754c566\", \"name\": \"panel_2\", \"type\": \"visualization\"},\n\": {\"id\": \"eea89780-664b-11ed-8cd7-973626cf6f70\", \"name\": \"panel_3\", \"type\": \"visualization\"},\n\": {\"id\": \"cfbb4e20-664c-11ed-8cd7-973626cf6f70\", \"name\": \"panel_4\", \"type\": \"visualization\"},\n\": {\"id\": \"c7f72ae0-664e-11ed-8cd7-973626cf6f70\", \"name\": \"panel_5\", \"type\": \"visualization\"}, {\"id\": \"943b3e00-\na5ef-11ec-9a68-ada9d754c566\", \"name\": \"panel_6\", \"type\": \"visualization\"}, {\"id\": \"ab503550-a5ef-11ec-9a68-\nada9d754c566\", \"name\": \"panel_7\", \"type\": \"visualization\"}, {\"id\": \"c78119a0-a5f5-11ec-9a68-\nada9d754c566\", \"name\": \"panel_8\", \"type\": \"visualization\"}, {\"id\": \"225f6020-a5f1-11ec-9a68-\nada9d754c566\", \"name\": \"panel_9\", \"type\": \"visualization\"},\n\": {\"id\": \"17d49220-662a-11ed-8cd7-973626cf6f70\", \"name\": \"panel_10\", \"type\": \"visualization\"},\n\": {\"id\": \"c8109100-6627-11ed-8cd7-973626cf6f70\", \"name\": \"panel_11\", \"type\": \"visualization\"},\n\": {\"id\": \"45d571c0-6626-11ed-8cd7-973626cf6f70\", \"name\": \"panel_12\", \"type\": \"visualization\"},\n\": {\"id\": \"3ae5d820-6628-11ed-8cd7-973626cf6f70\", \"name\": \"panel_13\", \"type\": \"visualization\"}], \"type\": \"dashboar\nd\", \"updated_at\": \"2022-12-01T12:41:01.165Z\", \"version\": \"WzlxOCwyXQ==\"}\n{\"exportedCount\": 16, \"missingRefCount\": 0, \"missingReferences\": []}
```

2.14.3 设置 Elasticsearch 集群 SMN 告警通知

本文主要介绍如何在Kibana中通过开源OpenDistro告警插件设置Elasticsearch集群的SMN告警通知功能。

场景描述

CSS服务默认给7.1.1、7.6.2和7.10.2版本的Elasticsearch集群安装了开源OpenDistro告警插件（opendistro_alerting），该插件具备Dashboard、Monitors、Destinations三个组件，并在Destinations组件中支持通过SMN服务发送告警通知。使用OpenDistro告警插件时，Elasticsearch集群需要通过SMN发送通知，这要求进行服务授权配置，服务授权是指允许CSS服务代理使用其他云资源，例如授权使用SMN服务资源，从而使CSS集群能够发送SMN消息通知。有关OpenDistro告警插件的官方文档，请参见[OpenDistro-Monitors](#)。

约束与限制

仅Elasticsearch 7.1.1、Elasticsearch 7.6.2和Elasticsearch 7.10.2集群默认安装了开源的OpenDistro告警插件，因此只有这几个版本的集群支持设置SMN告警通知。

前提条件

- 已经在SMN服务管理控制台完成了主题创建，具体操作请参见[创建主题](#)。
- 已获取CSS服务的管理员账号，并确认该账号具有CSS服务权限及IAM查看委托列表、创建委托及给委托授权的权限。

授权使用 SMN 服务

1. 通过CSS服务的管理员账号登录云搜索服务管理控制台。
2. 在左侧导航栏选择“服务授权”。
3. 在服务授权页面，单击“创建SMN委托”，在弹窗中确认委托创建成功。
 - 如果已经创建过委托，则右上角会提示“css_smn_agency exist, no need to created.”。
 - 如果无创建权限，则右上角会提示“当前用户没有操作权限，请通过IAM检查账户权限！”，请确认该管理员账号是否配置了IAM使用权限。

给集群设置 SMN 告警通知

1. 登录云搜索服务管理控制台。
2. 在“集群管理 > Elasticsearch”页面选择需要登录的集群，单击操作列中的“Kibana”登录Kibana。
3. 在Kibana页面，选择左侧导航栏的“Open Distro for Elasticsearch > Alerting”。
4. 新建一个SMN目的端用于发送告警消息。
 - a. 在“Alerting”页面选择“Destinations”页签，单击“Add destination”配置目的端信息。

表 2-155 Destinations 参数说明

参数名称	说明
Name	自定义目的端名称。
Type	保持默认值“SMN”。

参数名称	说明
Topic	选择前提条件中创建好的用于发送告警消息的SMN主题。 说明 7.1.1版本的Elasticsearch集群，需要手动输入Topic，注意和SMN服务中的主题名称保持一致。

图 2-60 Add destination

Alerting / Destinations / Create destination

Add destination

Destination

Name

Specify a name of the destination.

Type

SMN

Settings

Topic

test

- b. 单击“Create”返回Destinations列表，在列表可见新建的SMN目的端，表示创建完成。

图 2-61 Destinations 列表

Dashboard Monitors Destinations

Destinations

Search

All type < 1 >

Destination name	Destination type	Actions
test	SMN	Edit Delete

Rows per page: 20

- 5. 新建一个监控，并配置告警的触发条件和监控频率。

- a. 在“Alerting”页面选择“Monitors”页签，单击“Create monitors”配置监控信息。

表 2-156 Monitors 参数说明

参数名称	说明
Monitor name	自定义一个监控名称。
Monitor state	是否关闭这个监控，建议不关闭。
Method of definition	选择一种方式定义监控，推荐使用“Define using extraction query”。 <ul style="list-style-type: none"> Define using visual graph: 可视化的query语句 Define using extraction query: 具体的query语句
Index	选择监控的索引。
Time field	仅当选择“Define using visual graph”方式定义监控时，需要选择一个字段作为时间字段，来定义count等计数参数。
Frequency	选择监控频率，并设置监控频率的间隔信息。 <ul style="list-style-type: none"> By interval: 按时间间隔 Daily: 按每天 Weekly: 按每周 Monthly: 按每月 Custom cron expression: 按cron脚本规划

- b. 单击“Create”跳转到“Create trigger”页面。
c. 在“Create trigger”页面设置指定告警触发条件及告警时应触发的动作。

表 2-157 Trigger 参数说明

参数名称	说明
Trigger name	自定义一个触发器名称。
Severity level	定义触发器的敏感度，选择触发多少条告警之后发送告警消息，“1”为最高敏感度。
Trigger condition	定义触发条件。满足触发条件就会触发告警。
Action name	自定义一个触发器动作名称。
Destination	选择4新建的SMN目的端。
Message subject	仅7.10.2版本的Elasticsearch集群需要设置，定义告警消息的标题。

参数名称	说明
Message	设置SMN主题发送的body，定义消息发布的文本。默认定义了目的端是Email类型时的主题和正文，详细参数说明请参见 消息发布 。
Action throttling	设置消息发送频率用来限制在指定时间段内收到的通知消息数量。 例如设置10分钟，在接下来的10分钟内，即使触发条件满足多次，SMN仅发送1次告警通知。超过10分钟后，如果触发告警条件，SMN才会再次发送告警通知。

图 2-62 设置触发器动作的目的端

Configure actions

SMN notification: test-action

Action name

test-action

Names can only contain letters, numbers, and special characters

Destination

test - (SMN)

- d. 单击“Send test message”，如图2-64所示SMN主题订阅者成功收到邮件消息，则表示触发器设置成功。

图 2-63 发送验证信息

Destination

test - (SMN)

Message subject

test

Message info

```

{
  "subject": "subject",
  "message": "Monitor {{ctx.monitor.name}} just entered alert status. Please investigate the issue. - Trigger: {{ctx.trigger.name}} - Severity: {{ctx.trigger.severity}} - Period start: {{ctx.periodStart}} - Period end: {{ctx.periodEnd}}",
  "time_to_live": "3600"
}

```

Embed variables in your message using Mustache templates. [Learn more about Mustache](#)

Send test message

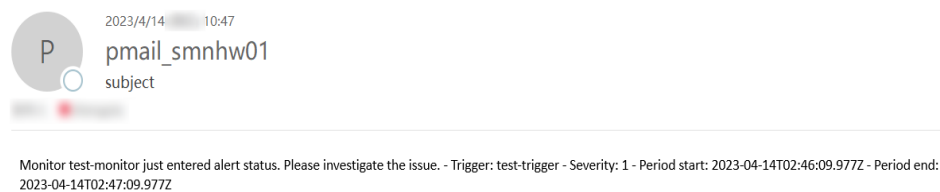
Message preview

```

{
  "subject": "subject",
  "message": "Monitor test-monitor just entered alert status. Please investigate the issue. - Trigger: test-trigger - Severity: 1 - Period start: 2023-04-14T02:16:36Z - Period end: 2023-04-14T02:17:36Z",
  "time_to_live": "3600"
}

```

图 2-64 成功收到邮件通知



e. 单击“Create”返回Monitor详情页。

2.14.4 智能检测 Elasticsearch 集群风险

CSS服务提供智能运维功能，支持检测集群潜在风险，并给出风险处理的建议。

场景描述

集群的智能运维支持以下功能：

- **启动检测任务**：通过启动检测任务，触发系统智能诊断集群的健康状况。
- **查看集群风险项**：检测任务完成后，可以查看集群存在的风险项详情，根据风险建议及时处理集群存在的风险。
- **删除检测任务**：当不需要历史所创建的检测任务时，可删除检测任务。删除检测任务后，系统将删除检测任务所对应的所有诊断信息。

智能运维的检查项包含如下项目。

- 检查当前时刻集群的健康状态，其中red表示有主分片未分配，yellow表示有副分片未分配，green表示所有分片均已分配。
- 检测集群节点数与AZ(可用区)数，判定Elasticsearch分布式集群的高可用性。
- 检测集群索引是否开启副本，未设置副本的索引在节点发生故障后可能导致索引不可用，本地盘集群未设置副本会有数据丢失的风险。
- 检测集群是否存在kibana索引冲突。
- 检测节点磁盘用量百分比，节点磁盘存储过大可能导致节点无法分配新索引分片并影响集群性能。
- 检测集群数据节点/冷数据节点的存储用量是否均衡，不均衡的存储用量分布可能导致集群负载不均，读写延迟增加。
- 连续5分钟检测当前集群是否有节点脱离或不可用。
- 检测节点分片数量是否过多，单节点分片数量过多会消耗大量节点资源，读写请求时延增加，集群元数据更新缓慢等。
- 检测所有分片大小，分片太大可能导致查询性能下降，节点内存消耗增加，影响扩缩容以及节点故障时分片的恢复速率等。
- 检测当前集群是否存在可升级的版本。
- 检测集群7天内是否有快照备份失败，或7天内无快照备份记录。

前提条件

当智能检测任务完成后需要发送SMN告警通知时，需要先在SMN服务创建主题，操作指导请参见[创建主题](#)。同时，需要参考[授权使用SMN服务](#)完成SMN服务授权。

启动检测任务

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击需要进行智能运维的集群名称，跳转至该集群基本信息页面。
3. 左侧菜单栏选择“智能运维”。
4. 在智能运维页面，单击左上角的“启动检测”。
5. 在对话框中填写检测任务的基本信息，单击“确定”。

表 2-158 配置检测任务

参数	说明
名称	检测任务的名称。
描述	检测任务的简短描述。
SMN主题	当勾选“任务完成后发送SMN告警”时，需要选择当前账号已创建的SMN主题。
告警灵敏度	当勾选“任务完成后发送SMN告警”时，需要选择告警灵敏度。 当检测到的风险项存在大于等于此告警敏感度的条目，则发送SMN告警消息，消息内容则包含所有风险项。

6. 当检测任务状态变成“完成”时，表示智能检测结束，可以查看集群风险项。

查看集群风险项

当检测任务完成时，可以查看检测结果。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击集群名称，跳转至该集群基本信息页面。
3. 左侧菜单栏选择“智能运维”。
4. 在智能运维列表页面，选择已启动的检测任务。单击下方的 ^，可查看当前检测任务的“创建时间”、“摘要”、“ID”以及检测任务所发现的“风险项”。
单击风险项左侧 ▾，可查看风险项详情，包括检测项、风险描述、风险建议。
您可以根据风险建议及时处理集群所存在的风险。

图 2-65 风险项



5. 单击智能运维列表操作列的“导出风险项”，支持下载检测结果。

删除检测任务

当检测任务使用完毕，可以删除任务。删除检测任务后，系统将删除检测任务所对应的所有诊断信息。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击集群名称，跳转至该集群基本信息页面。
3. 左侧菜单栏选择“智能运维”。
4. 选择待删除的检测任务，单击操作列的“删除”。
5. 在对话框中输入“DELETE”，单击“确定”，删除检测任务。

2.14.5 查询和管理 Elasticsearch 集群日志

为了方便用户使用日志定位问题，云搜索服务提供了日志备份、日志采集和日志查询功能。


日志备份可以定期将集群的日志存储在OBS桶中，通过OBS可以直接下载需要的日志文件，进行问题分析定位。

日志采集可以实时将集群的日志存放到CSS集群中，方便通过Kibana进行日志搜索分析。

日志备份在OBS桶中需要额外收费，具体收费以[对象存储服务计费说明](#)为准。

查询日志

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要查询日志的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”。
4. 在日志管理页面进行日志查询。

5. 选择需要查询的节点、日志类型和日志级别信息后，单击 ，显示查询结果。

查询日志时，是从最近时刻的1万条日志中进行匹配，查询结果最多显示100条。

备份日志

日志备份可以定期将CSS集群的日志存储到OBS桶中。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要配置日志备份的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”，在“日志备份开关”右侧单击开关，打开集群的日志管理功能。
4. 在“编辑日志备份配置”弹窗中，完成参数配置。

弹窗中默认填写了云搜索服务自动为用户创建的“OBS桶”、“备份路径”和“IAM委托”，用于日志备份。支持用户参考[表2-159](#)修改默认值。

如果集群已经启用了日志备份功能，也可以单击“日志备份配置”右侧的编辑图标，在“编辑日志备份配置”窗口，参考表2-159更新日志备份的配置参数。

表 2-159 日志备份配置

参数	说明	注意事项
“OBS桶”	选择日志存储的OBS桶。单击右侧的“创建桶”支持新建OBS桶。	OBS桶的所在区域必须跟集群的所在区域保持一致。 说明 如果是子账号，需要同时设置GetBucketStoragePolicy、GetBucketLocation、ListBucket、ListAllMyBuckets权限，才能看到OBS桶。
“备份路径”	填写日志在OBS桶中的存放路径。	备份路径配置规则： <ul style="list-style-type: none">• 备份路径不能包括符号“\:*?*<> ”。• 备份路径不能以“/”开头。• 备份路径不能以“.”开头或结尾。• 备份路径的总长度不能超过1023个字符。
“IAM委托”	选择IAM委托，指当前账号授权云搜索服务访问或维护存储在OBS中数据。单击右侧的“创建委托”支持新建委托。	IAM委托需满足如下条件： <ul style="list-style-type: none">• “委托类型”选择“云服务”。• “云服务”选择“Elasticsearch”或者“云搜索服务 CSS”。• 必选策略：“OBS Administrator”

5. 备份日志。

- 自动备份日志。

在“自动备份开关”右侧，单击开关，开启自动备份日志功能。

开启“自动备份开关”后，在“修改日志备份策略”弹窗中设置“备份开始时间”。设置成功后，系统会按照设置的时间进行自动备份。

打开“自动备份开关”后，单击开关右侧的编辑图标，可以修改“备份开始时间”。

- 手动备份日志。

选择“日志备份”页签，单击“开始备份”，在弹出的确认提示框中单击“确定”，开始备份日志。

日志备份列表中的“任务状态”为“Successful”时，表示日志备份成功。

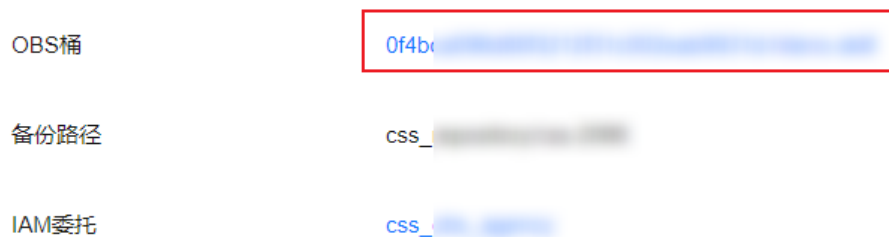
说明

如果日志备份失败，可以在“日志备份”页签的右侧单击查看“失败任务数”，了解失败原因。失败任务数最多显示10条，当日志备份功能关闭或集群删除时，失败记录也会被清空。

6. 查看日志文件。

日志备份成功后，用户可以单击“OBS桶”跳转到桶列表，找到存放备份日志的桶查看日志文件。

图 2-66 进入 OBS



采集日志

日志采集可以实时将CSS集群日志存放到本集群或其他同网络的集群中，方便通过Kibana进行日志搜索分析。

如果要使用日志采集功能，集群必须满足以下条件。当集群不满足要求时，建议先升级集群再启用日志采集。

- 集群版本是Elasticsearch 7.10.2、OpenSearch 1.3.6或OpenSearch 2.11.0。
- 集群的镜像版本号 $\geq 24.2.0$ 。集群的镜像版本号可以在集群列表的“版本”列查看，如图所示。

图 2-67 查看集群子版本号

名称/ID	集群状态	任务状态	版本
css-2e92	可用	-	24.2.0_0625
0ab2278b-7d...			

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要配置日志备份的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”，在“日志采集开关”右侧单击开关，打开集群的日志采集功能。

说明

如果页面上没有“日志采集开关”，则表示该集群不支持采集日志。

4. 在“日志采集配置”弹窗中，完成参数配置。

表 2-160 日志采集配置

参数	说明
索引前缀名称	设置采集的日志中索引名称的前缀，则日志采集的索引名称为“索引前缀名称+日志采集日期”，日志采集日期以天为单位。 索引前缀名称只支持数字、小写字母、下划线和中划线，支持1~128个字符。

参数	说明
保留周期	设置采集的日志的保留周期，单位为天，支持输入1~3650。当采集的日志超过保存周期时，日志将会被删除。
日志保存集群	设置采集的日志保存在哪个集群，支持选择“当前集群”和“其他集群”。 默认选择“当前集群”。当选择“其他集群”时，需要选择日志保存的目标集群，并进行集群网络联通行检查，只有在同一VPC网络下的集群才支持跨集群保存日志，且集群版本必须一致。

如果集群已经启用了日志采集功能，也可以单击右侧的编辑图标修改日志采集配置。

- 单击“确定”，开启集群日志采集开关。
当“状态”变成“运行中”时，表示已启动日志采集。
单击“访问Kibana”，登录集群，可以搜索查看日志。
单击“日志保存集群”处的集群名称，可以跳转到集群详情页面。

图 2-68 日志采集



- 当需要关闭日志采集时，单击“日志采集开关”右侧的开关，在弹窗中单击“确定”，关闭日志采集。
日志采集关闭后，集群中已采集的日志不会被清空，仅当超过采集日志时设置的“保存周期”时才会被删除。

日志文件介绍

Elasticsearch和OpenSearch集群备份的日志文件主要包括废弃操作日志、运行日志、慢索引日志、慢查询日志。

表 2-161 日志文件类型

日志名称	描述
clustername_deprecation.log	弃用操作的日志记录。
clustername_index_indexing_slowlog.log	慢索引日志。
clustername_index_search_slowlog.log	慢索引查询日志。
clustername.log	Elasticsearch运行日志。

日志名称	描述
clustername_access.log	接入日志。

2.15 查看 Elasticsearch 集群审计日志

公有云平台提供了云审计服务。通过云审计服务，您可以记录与云搜索服务相关的操作事件，便于日后的查询、审计和回溯。

前提条件

已开通云审计服务。

支持审计的关键操作列表

表 2-162 支持审计的关键操作列表

操作名称	资源类型	事件名称
创建集群	cluster	createCluster
删除集群	cluster	deleteCluster
扩容集群	cluster	roleExtendCluster
重启集群	cluster	rebootCluster
配置自定义词库	cluster	loadLexicon
删除自定义词库	cluster	deleteLexicon
设置集群快照的基础配置	cluster	updateSnapshotPolicy
设置自动创建快照策略	cluster	updateAutoSnapshotPolicy
集群升级	cluster	upgradeCluster
升级重试	cluster	retryAction
手动创建快照	snapshot	createSnapshot
恢复快照	snapshot	restoreSnapshot
删除快照	snapshot	deleteSnapshot

查看审计事件

在云审计服务管理控制台，用户创建管理类追踪器后，系统会开始记录用户对云服务资源的操作，创建数据类追踪器后，系统会开始记录用户对OBS桶中数据的操作。云审计服务管理控制台会保存最近7天的操作记录。

当需要在云审计服务管理控制台查看或导出最近7天的操作记录时，请参见[查询审计事件](#)。

3 使用 OpenSearch 搜索数据

3.1 OpenSearch 使用流程

表 3-1 OpenSearch 集群的使用流程

任务分类	操作	参考文档	
使用	规划集群	在创建OpenSearch集群前，需要先完成集群规划。规划时，应考虑是否多可用区部署以提高集群的高可用性，合理配置集群的节点类型与节点存储规格，以及根据业务需求选择适当的集群版本和安全模式，同时注意索引分片的优化，以确保集群的稳定性和性能。	OpenSearch集群规划建议
	创建集群	基于集群规划，创建合适的OpenSearch集群。	创建OpenSearch集群
	访问集群	OpenSearch集群提供多样化的访问方式，包括Kibana、Cerebro、开源API、Java、Python和Go客户端，以及通过内网和公网的多种网络配置，用户可根据编程语言偏好和网络环境选择最合适的接入方法。	OpenSearch集群访问方式
	导入数据	OpenSearch集群提供多种数据导入方式，包括Logstash、开源Elasticsearch API、云数据迁移服务CDM和数据复制服务DRS，支持不同来源和格式的数据导入，以及针对关系型数据库的实时同步，用户可根据业务场景和数据特性选择最佳方法。	OpenSearch集群导入数据方式

任务分类	操作	参考文档
	搜索数据	CSS服务推荐使用DSL语言在OpenSearch集群中搜索数据，同时也支持使用SQL语言进行数据搜索。 使用DSL语言在OpenSearch中搜索数据 使用SQL语言在OpenSearch中搜索数据
	增强集群搜索能力	CSS服务的OpenSearch集群基于社区版本提供了一系列增强特性，包括向量检索、存算分离、流量控制、大查询隔离、聚合增强、读写分离、冷热数据切换、索引回收站等，以适应不同场景下的性能和成本优化需求，增强集群稳定性和搜索能力。 OpenSearch集群搜索增强特性介绍
运维	备份与恢复	备份与恢复是通过集群快照备份OpenSearch集群数据，在数据丢失或需要历史数据时快速恢复数据，提升集群的数据可靠性。 创建快照备份OpenSearch集群数据 恢复OpenSearch集群数据
	扩缩容	CSS服务提供灵活的扩容和缩容功能，支持增加节点数量、升级节点规格、增加节点类型，以及随机或指定缩容节点，以动态调整集群资源，适应不同业务需求并优化成本。 扩容OpenSearch集群 缩容OpenSearch集群的节点数量
	管理集群	CSS服务提供全面的集群管理功能，包括集群信息查看、集群用户授权、标签管理、安全模式更改、节点替换、企业项目绑定、可用区切换、自定义词库配置等，旨在帮助用户高效、灵活地管理OpenSearch集群，确保集群的安全性、高可用性和性能优化。 管理OpenSearch集群
	管理集群索引策略	使用OpenSearch的ISM（Index State Management）插件创建和管理索引的生命周期策略，包括策略的创建、索引关联、以及策略的管理与变更，旨在优化集群性能和降低存储成本。 创建及管理OpenSearch索引策略

任务分类	操作	参考文档
	监控与日志管理	CSS服务提供全面的监控和日志管理功能，包括集群和节点监控指标、告警规则配置、日志备份与采集，以及智能运维工具，帮助用户有效监控、分析和维护OpenSearch集群，确保集群的稳定性和性能。 CES中OpenSearch集群支持的监控指标 使用CES监控OpenSearch集群 设置OpenSearch集群SMN告警通知 智能检测OpenSearch集群风险 查询和管理OpenSearch集群日志
	审计日志	通过云审计服务，用户可以记录和查询OpenSearch集群的关键操作事件，且操作记录在管理控制台保存最近7天，便于后续的审计和回溯。 查看OpenSearch集群审计日志

3.2 OpenSearch 集群规划建议

规划集群可用区

为防止数据丢失，并确保在服务中断情况下能降低集群的停机时间，从而增强集群的高可用性，CSS服务支持跨可用区（即多可用区）部署。用户可以在同一个区域内选择两个或三个不同的可用区进行集群部署。

在创建集群时，如果用户选择了两个或三个可用区，CSS服务将自动开启跨AZ的高可用性特性，确保节点在这些可用区内均匀分配。系统均匀分配的节点满足各个AZ之间节点数量的差小于等于1，具体的节点分布情况，包括不同数量的节点如何在各个可用区中分布，可以参考[表3-2](#)。

须知

- 在创建集群时，选择的任意类型的节点数量都要大于等于所选的AZ数量，否则跨可用区部署会失败。
- 部署跨AZ集群时，任意类型的节点都会被均匀的分布在不同的AZ上。

表 3-2 节点数量和 AZ 分布

集群节点个数	单AZ	两AZ		三AZ		
	AZ1	AZ1	AZ2	AZ1	AZ2	AZ3
1个节点	1	不支持		不支持		
2个节点	2	1	1	不支持		

3个节点	3	2	1	1	1	1
4个节点	4	2	2	2	1	1
...

在选择多可用区部署时，建议合理配置副本数量，以更高效地利用跨可用区的高可用性特性。

- 在跨两个可用区的部署中，当其中一个AZ不可用时，剩下的AZ需要继续提供服务，因此索引的副本个数至少为1个。由于Elasticsearch默认副本数为1个，因此如果您对读性能没有特殊要求，可以直接使用默认值。
- 在跨三个可用区部署中，为了保证其中任意一个AZ不可用时，剩余的AZ可以继续提供服务，因此索引的副本数至少要为1个。为了提高集群的查询能力，也可以设置更多的副本。由于Elasticsearch默认的副本数为1个，因此需要用户修改setting配置来实现修改索引副本个数。

可以通过如下命令修改索引的副本个数，如：

```
curl -XPUT http://ip:9200/{index_name}/_settings -d '{"number_of_replicas":2}'
```

也可以通过在模板中指定所有索引的副本个数，如：

```
curl -XPUT http://ip:9200/_template/templatename -d '{"template": "*", "settings": {"number_of_replicas": 2}}'
```

其中，“ip”表示集群内网访问地址，“index_name”表示索引名称，“number_of_replicas”表示修改后的索引副本个数，此处以修改为2个索引副本为例。

当采用多可用区部署时，如果某个可用区发生故障，相关的业务故障行为分析及应对策略请参见表3-3。

表 3-3 AZ 故障的业务故障行为分析

可用区数量	主节点个数	业务中断行为及应对建议
2	0	<ul style="list-style-type: none"> • 如果节点个数为2的倍数：一半的数据节点故障，需要替换故障可用区中的一个节点，才能继续选择主节点。 • 如果节点数为奇数： <ul style="list-style-type: none"> - 故障AZ含多一个节点，需要替换故障可用区中一个节点，才能继续选择主节点。相关替换请联系技术支持。 - 故障AZ含少一个节点，不中断业务，能够继续选择主节点。

可用区数量	主节点个数	业务中断行为及应对建议
2	3	有50%机会的停机时间。当两个专用主节点分配到一个可用区中，一个主节点分配到另一个可用区中时： <ul style="list-style-type: none">• 如果具有一个专用主节点的可用区遇到中断，则剩余可用区具有两个专用主节点，这两个专用主节点可以选择出主节点。• 如果具有两个专用主节点的可用区遇到中断，剩余可用区只有一个专用主节点，无法选择出主节点，业务中断，需要联系技术支持。
3	0	当您选择3个可用区，节点个数为4，三个可用区的节点分布数为2，1，1，如果节点个数为2的可用区故障，那么此时业务中断，建议您选择三个可用区时避免选择4个节点。 一般不会出现业务中断时间。
3	3	无业务中断时间。

📖 说明

当集群创建完成后，支持切换可用区，具体操作请参见[切换OpenSearch集群可用区](#)。

切换可用区包含两大场景：可用区高可用改造和可用区平移切换。

- 可用区高可用改造：适用于单AZ改造成两AZ、单AZ改造成三AZ或两AZ改造成三AZ的场景，目的是为了提升集群的高可用性。
- 可用区平移切换：适用于从一个AZ完全迁移到另一个AZ的场景，目的是为了解决当前可用区资源不足的问题。

规划节点类型

在OpenSearch集群中，合理规划不同节点类型对于优化性能和资源利用率至关重要。在创建集群时，应根据业务需求、查询负载、数据增长模式和性能目标来确定添加哪些类型的节点，以实现合适的集群性能和资源管理。[表3-4](#)是介绍了不同节点类型的适用场景，建议用户根据具体的业务需求和性能预期来选择是否启用该类节点。

📖 说明

- 如果创建集群时未启用Master节点或Client节点，当业务运行一段时间后，发现数据节点压力太大时，支持单独添加Master节点或Client节点，具体操作请参见[添加Master或Client节点](#)。
- 如果创建集群时未启用冷数据节点，则集群创建完成后不支持单独添加冷数据节点，请在创建集群时合理选择是否启用冷数据节点。

表 3-4 各类节点的适用场景

节点类型	节点功能描述	适用场景
数据节点 (ess)	数据节点用于存储数据，当集群没有 Master 节点和 Client 节点时，数据节点会同时兼顾这两类节点的功能。	集群必配节点类型。 <ul style="list-style-type: none">当集群未启用 Master 节点和 Client 节点时，数据节点将同时承担集群管理、存储数据、提供接入集群和分析数据的职责。此时，为保证集群中数据的稳定性，建议设置节点数量大于等于 3 个。当集群启用了 Master 节点但未启用 Client 节点时，数据节点将用于存储数据并提供接入集群和分析数据的功能。当集群未启用 Master 节点但启用了 Client 节点时，数据节点将用于存储数据并提供集群管理功能。当集群同时启用了 Master 节点和 Client 节点时，数据节点将仅用于存储数据。
Master 节点 (ess-master)	Master 节点负责管理集群中所有节点任务，如元数据管理、索引创建与删除、分片分配等。在大规模集群的元数据管理、节点管理、稳定性保障和集群操作控制中发挥着至关重要的作用。	<ul style="list-style-type: none">集群规模较大：当集群的节点数超过 16 个时，为了更有效地管理集群状态和元数据，建议添加专用的 Master 节点。拥有高索引和分片数量：如果索引数量或分片数超过 1 万个，Master 节点可以处理更复杂的集群管理任务，避免对数据节点的性能造成影响。单独管理集群节点：Master 节点负责维护集群的元数据，包括索引映射、设置和别名等，对于复杂的集群结构，专用的 Master 节点可以提供更好的管理。包括节点加入、退出以及故障检测等，Master 节点在集群节点管理中扮演核心角色。提升集群稳定性和可靠性：专用的 master 节点可以提高集群的稳定性和可靠性，因为它减少了对同时承担数据存储和查询任务的节点的依赖。优化数据节点性能：通过将集群管理任务从数据节点分离到 Master 节点，可以优化数据节点的性能，使其专注于数据操作。

节点类型	节点功能描述	适用场景
Client节点 (ess-client)	Client节点负责接收并协调外部请求，如search和write请求，在处理高负载查询、复杂聚合、大量分片管理以及优化集群扩展性方面发挥着重要作用。	<ul style="list-style-type: none">存在高查询QPS：当集群面临高查询每秒查询率（QPS）时，独立的Client节点可以更均匀地分发查询请求，减轻数据节点的负担，提高整体查询性能。存在复杂的聚合查询：对于需要大量计算资源的复杂聚合查询，Client节点可以专门处理聚合结果，从而提升聚合查询的效率和响应速度。集群分片数量多：在分片数量较多的集群中，Client节点可以有效地协调和管理对各个分片的查询请求，提高请求的转发和处理效率。可以减轻数据节点压力：Client节点负责解析search请求，确定索引分片的位置，并协调分片节点执行查询。可以减轻数据节点的负载，使它们更专注于数据的存储和索引。提高集群扩展性：增加client节点可以提供更好的集群扩展性和灵活性，支持更大规模的数据集和更复杂的查询需求。
冷数据节点 (ess-cold)	冷数据节点用于存储对查询时延要求不高，但数据量较大的历史数据，是管理大规模数据集和优化存储成本的有效方式。	<ul style="list-style-type: none">需要存储大量历史数据：当需要存储大量不常访问但对分析有用的历史数据时，使用冷数据节点可以提供成本效益较高的存储解决方案。需要优化热数据性能：通过将旧数据迁移到冷数据节点，可以减少对热数据节点的存储压力，从而优化热数据的查询和写入性能。对查询时延要求不高：对于那些查询频率不高且可以容忍较高查询时延的数据，冷数据节点是合适的存储选择。追求成本效益：冷数据节点通常采用具有较大磁盘容量的规格，这有助于降低存储成本，同时满足大规模数据存储需求。

规划节点存储

- **规划节点机型**

CSS服务支持多种节点机型，每种机型适用于不同的业务场景，建议用户根据具体的业务需求和性能预期来选择最合适的机型，以优化存储性能和成本效益。

表 3-5 节点机型适用场景

节点机型	磁盘类型	规格说明	适用场景
计算密集型	云盘	CPU:内存=1:2	推荐场景 ，用于数据量较少（单节点<100GB）的搜索场景。
通用计算型	云盘	CPU:内存=1:4	通用场景 ，用于单节点数据量在100-1000GB间的搜索与分析场景，例如中等规模的电商搜索、社交搜索、日志搜索等场景。
内存优化型	云盘	CPU:内存=1:8	通用场景 ，用于单节点数据量在100-2000GB间的搜索与分析场景。 适合向量检索场景 ，大内存有利于提升集群的性能与稳定性。
磁盘增强型	本地盘	挂载HDD盘	适合日志场景 ，用于存储冷数据，对冷数据的数据查询性能要求低，并且数据需要更新的场景。
超高I/O型 (CPU架构为鲲鹏计算)	本地盘	挂载SSD盘	适用 大型日志场景 ，用于存储热数据。
超高I/O型 (CPU架构为X86计算)	本地盘	挂载SSD盘	适用 大型搜索与分析场景 ，场景对计算或磁盘I/O均有较高要求，例如舆情分析、专利检索、以及部分数据库加速场景。

- **规划节点规格**

在规划节点规格时，推荐优先考虑高配置但节点数量较少的方案。例如，一个由3个节点组成的集群，每个节点配置为32核CPU和64GB内存，通常比一个由12个节点组成的集群，每个节点配置为8核CPU和16GB内存，在集群的稳定性和扩展性方面更具优势。

优势主要体现在如下方面。

- **集群稳定性**：高配置节点通常能提供更强的处理能力和更大的内存空间，从而提高集群的整体稳定性。
- **扩容便捷性**：当高配置集群遇到性能瓶颈时，可以通过横向扩展轻松解决，即简单地向集群中添加更多具有相同高配置的节点。这种扩展方式简单直接，易于实施。
- **维护简便**：较少的节点数量意味着更少的维护工作和更低的管理复杂性。

相比之下，低配置集群在需要扩容时，往往需要进行纵向扩展，即提升单个节点的配置。这不仅可能涉及更复杂的迁移和升级过程，还可能增加额外的维护成本和技术挑战。

因此，在规划集群时，应综合考虑性能、成本、维护和扩展性，选择最适合业务需求的节点规格。

- **规划存储容量**

在规划CSS集群的存储容量时，应考虑数据量、副本因子、数据膨胀率和磁盘使用率等多个关键因素。以下是一个推荐的计算方法，用以确定所需的集群存储容量。

存储容量=源数据x(1+副本数量)x(1+数据膨胀率)x(1+预留空间比例)

- 源数据：首先确定预期存储的原始数据量。
- 副本数量：设置副本因子，默认建议值为1，以保证数据的高可用性。
- 数据膨胀率：集群在索引过程中可能会产生额外的数据膨胀，通常建议按照25%的膨胀率进行计算。
- 磁盘空间使用率：考虑到操作系统和文件系统本身占用的空间，以及留出一定的空间以优化磁盘性能和冗余，建议将磁盘使用率控制在70%，即预留空间比例为30%。

将具体数值代入公式：存储容量=源数据x2x1.25x1.3

简化计算，如果源数据量已知，最终的存储容量大约是源数据的3.25倍。这个计算方法提供了一个基础的估算，但实际配置时还需要根据具体业务场景和增长预期进行调整。

规划节点数量

创建集群时，集群的节点数量应当基于业务性能需求和预期负载进行规划。[表3-6](#)提供了计算方式用以确定合适的节点数量。通过这个计算方式可以更科学地规划集群的节点数量，以满足业务需求并保证集群的性能和稳定性。

表 3-6 节点数量的计算方式

节点	性能基线	节点数量计算方式	示例
写入节点	<ul style="list-style-type: none">对于挂载云盘的节点，其单核写入性能基线为1MB/s。对于超高IO型的节点，其单核写入性能基线为1.5MB/s。	写入节点数=业务峰值时的流量÷单节点的核数÷单核写入性能基线x副本数	业务峰值写入100MB/s，使用16u64g的节点，预计需要 $100 \div 16 \div 1 \times 2 = 12$ 个节点。
查询节点	相同节点，不同业务场景下的性能差异非常大，单节点的性能基线难以评估。这里以业务 平均查询响应时间 （单位为秒）作为查询的性能基线进行测算。	查询节点数=QPS÷（单节点的核数x3÷2÷平均查询响应时间）x分片数量	查询QPS要求1000，平均查询响应时间100ms，索引规划3个分片，使用16u64g的节点，预计需要 $1000 \div (16 \times 3 \div 2 \div 0.1) \times 3 = 12$ 个节点。
总节点数量	不涉及	总节点数量=写入节点数+查询节点数	总节点数=写入节点数+查询节点数=24个节点数。
说明 这里计算的总节点数量表示数据节点和冷数据节点的数量之和。			

当一个集群包含的节点类型不同时，各节点类型支持的节点数量不同，设置节点数量时可以参考[表3-7](#)。

表 3-7 集群的节点数量取值说明

一个集群包含的节点类型	节点数量的取值范围
ess	ess: 1~32
ess、ess-master	ess: 1~200 ess-master: 3~9的奇数
ess、ess-client	ess: 1~32 ess-client: 1~32
ess、ess-cold	ess: 1~32 ess-cold: 1~32
ess、ess-master、ess-client	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32
ess、ess-master、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-cold: 1~32
ess、ess-client、ess-cold	ess: 1~32 ess-client: 1~32 ess-cold: 1~32
ess、ess-master、ess-client、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32 ess-cold: 1~32
说明 <ul style="list-style-type: none">• ess: 数据节点，即创建集群时必配的节点类型，其他3种节点类型都是基于业务需要可选的类型。• ess-master: Master节点。• ess-client: Client节点。• ess-cold: 冷数据节点。	

规划虚拟私有云和子网

CSS服务支持在非共享VPC和共享VPC内创建集群。

共享VPC和非共享VPC相比，有如下优势：

- 在一个账号内统一创建资源，并将资源共享给其他账号，其他账号无需创建重复资源，可以精简资源数量以及网络架构，提升管理效率并节约成本。
比如不同账号下的VPC网络互通需要建立对等连接，使用共享VPC后，则不同账号用户可以在同一个VPC创建资源，免去了对等连接配置，有效的简化组网结构。

- 在一个账号内统一管理运维资源，便于企业集中配置业务安全策略，并且利于对资源使用情况的监控和审计，支撑业务的安全诉求。

当选择使用共享VPC创建集群时，需要提前完成共享VPC子网的创建，操作流程请参见表3-8。有关共享VPC的使用限制及详细操作指导等信息可以参考[共享VPC概述](#)。

表 3-8 共享子网创建流程说明

方法	说明	操作指导
方法A	<ol style="list-style-type: none"> 通过RAM管理控制台，所有者创建共享，将子网共享给使用者。配置如下： <ol style="list-style-type: none"> 选择共享子网。 为共享子网选择权限，即指定使用者对该共享子网具备的权限。 CSS服务在共享VPC中创建集群需要选择权限：default vpc subnet statement。 指定共享子网的使用者，可以指定多个。 共享创建完成后，通过RAM管理控制台，使用者可以选择接受或者拒绝共享申请。 <ul style="list-style-type: none"> 使用者接受共享申请，子网共享成功。如果后续使用者不再需要使用该共享子网，可以退出该共享。 使用者拒绝共享申请，子网共享失败。 	<ol style="list-style-type: none"> 创建共享 接受/拒绝共享邀请 退出共享
方法B	<ol style="list-style-type: none"> 通过RAM管理控制台，所有者创建共享，将子网共享给使用者。配置如下： <ol style="list-style-type: none"> 选择共享子网。 为共享子网选择权限，即指定使用者对该共享子网具备的权限。 CSS服务在共享VPC中创建集群需要选择权限：default vpc subnet statement。 指定共享子网的使用者，可以指定多个。 通过VPC管理控制台，选择上一步中已创建的共享，将子网加入到该共享内。 共享创建完成后，通过RAM管理控制台，使用者可以选择接受或者拒绝共享申请。 <ul style="list-style-type: none"> 使用者接受共享申请，子网共享成功。如果后续使用者不再需要使用该共享子网，可以退出该共享。 使用者拒绝共享申请，子网共享失败。 	<ol style="list-style-type: none"> 创建共享 将VPC子网共享给其他账号 接受/拒绝共享邀请 退出共享

规划集群安全模式

表 3-9 集群类型介绍

集群类型		集群描述	适用场景
非安全集群	非安全模式的集群	非安全模式的集群无需安全认证即可访问，采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	适合内网业务，用于测试场景。 <ul style="list-style-type: none">• 优点：简单，接入集群容易。• 缺点：安全性差，任何人都可以访问集群。
安全集群	安全模式+HTTP协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	可以实现用户权限隔离，适用于对集群性能敏感的场景。 <ul style="list-style-type: none">• 优点：访问集群需要安全认证，提升了集群安全性。通过HTTP协议访问集群又能保留集群的高性能。• 缺点：无法公网访问集群。
	安全模式+HTTPS协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTPS协议进行通信加密，使数据更安全。	有非常高的安全要求，且需要公网访问集群的场景。 <ul style="list-style-type: none">• 优点：访问集群需要安全认证，提升了集群安全性，且HTTPS协议的通讯加密可以实现集群公网访问功能。• 缺点：通过HTTPS协议访问集群，集群的读取性能相对HTTP协议来说，会下降20%左右。

当访问安全模式的集群时，需要输入用户名和密码通过安全认证才能访问。CSS服务支持以下两类用户的安全认证：

- 集群的管理员：管理员账户名默认为**admin**，密码为创建集群时设置的管理员密码。
- 集群的用户：集群的管理员通过Kibana创建集群的用户和密码。创建方式请参见[使用OpenSearch Dashboard创建用户并授权](#)。

说明

当集群创建完成后，支持切换安全模式，具体操作请参见[更改OpenSearch集群安全模式](#)。

切换安全模式包含三大场景：非安全模式切换为安全模式、安全模式切换为非安全模式、安全模式的协议切换。

规划索引分片数

在使用集群的过程时，特别是在进行数据导入操作之前，建议根据具体的业务需求，提前对集群的数据结构和分布进行规划。这包括合理设计索引和确定分片数量。为了确保集群在性能和可扩展性方面达到最佳状态，以下是一些建议。

- **单个分片大小**：建议将每个分片的大小控制在10GB到50GB之间。这有助于在存储效率和查询性能之间取得平衡。
- **集群总分片数量**：为了管理方便和避免过度扩展，建议将集群的总分片数量控制在3万以内。这有助于保持集群的稳定性和响应速度。
- **内存与分片比例**：在资源分配上，建议每1GB的内存空间放置20到30个分片。这样可以确保每个分片都有足够的内存资源进行索引和查询操作。
- **单节点分片数**：为了避免单点过载，建议每个节点上的分片数量不超过1000个。这有助于避免节点资源竞争，确保节点的稳定运行。
- **索引分片与节点数的关系**：对于单个索引，建议其分片数与集群的节点数保持一致，或者设置为节点数的整数倍。这有助于实现负载均衡，优化查询和索引的性能。

通过以上建议，可以更有效地规划和管理CSS集群的索引分片，从而提升集群的整体性能和可维护性。

3.3 创建 OpenSearch 集群

本文介绍如何创建OpenSearch集群。

场景描述

当创建的集群类型不同时，需要关注如[表3-10](#)所示的关键参数的配置。

表 3-10 集群创建差异

集群类型	“安全模式”	“HTTPS访问”	“公网访问”	“Kibana公网访问”
非安全模式的集群	关闭	不涉及	不支持启用	不支持启用
安全模式+HTTP协议的集群	开启	关闭	不支持启用	支持启用
安全模式+HTTPS协议的集群	开启	开启	支持启用	支持启用

前提条件

已经参考[OpenSearch集群规划建议](#)完成待创建的OpenSearch集群规划。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在总览页面单击右上角的“创建集群”，进入“创建集群”页面。

或者左侧导航栏单击“集群管理 > OpenSearch”，单击右上角的“创建集群”，进入“创建集群”页面。

3. 在“基础配置”页面，完成OpenSearch集群的基本信息和资源配置。

表 3-11 OpenSearch 集群的基础配置

参数	说明
计费模式	集群支持包年/包月和按需计费两种模式。 <ul style="list-style-type: none">● 包年/包月：根据集群购买时长，一次性支付集群费用。最短时长为1个月，最长时长为3年。如果购买时长超过9个月，建议包年购买，价格更优惠。一年计费为购买10个月得12个月。● 按需计费：按实际使用时长计费，计费周期为一小时，不足一小时按一小时计费。
订购周期	选择包年/包月模式后，需要选择购买时长。 您可以根据需求，选择是否需要“自动续费”。
当前区域	选择集群的所在区域。 不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
可用区	选择集群工作区域下关联的可用区。 最多支持配置3个“可用区”，多可用区的使用建议请参见 规划集群可用区 。
集群类型	选择“OpenSearch”。
集群版本	选择所需的集群版本，支持的版本以界面可选项为准。
集群名称	自定义集群名称，可输入的字符范围为4~32个字符，只能包含数字、字母、中划线和下划线，且必须以字母开头。
节点数量	集群中的数据节点个数。可选节点数为1~32，建议节点数为3或3以上，以提升集群可用性。 <ul style="list-style-type: none">● 当集群未启用Master节点和Client节点时，数据节点将同时具备集群管理、存储数据、提供接入集群和分析数据的功能。此时，为保证集群中数据的稳定性，建议设置节点数量大于等于3个。● 当集群启用了Master节点但未启用Client节点时，数据节点将用于存储数据并提供接入集群和分析数据的功能。● 当集群未启用Master节点但启用了Client节点时，数据节点将用于存储数据并提供集群管理功能。● 当集群同时启用了Master节点和Client节点时，数据节点将仅用于存储数据。
CPU架构	支持“X86计算”和“鲲鹏计算”两种类型。具体支持的类型由实际区域环境决定。

参数	说明
节点规格	集群中的节点规格。您可以根据需求，选择对应的规格。每个集群只能选择一个规格，规格的详细说明可参考 弹性云服务器的实例类型与规格 。
节点存储	选择存储类型，支持普通I/O、高I/O、超高I/O、极速型SSD。 说明 如果当前节点的存储类型不支持，则界面不显示。
节点存储容量	设置节点的存储空间大小，其取值范围与“节点规格”关联，不同的规格允许的取值范围不同。 节点存储容量只支持配置为20的倍数。 集群创建完成后，不支持扩容节点存储容量，请基于业务量合理选择容量。
启用Master节点	Master节点负责关键的集群管理任务，如元数据管理、索引创建与删除、分片分配等。在大规模集群的元数据管理、节点管理、稳定性保障和集群操作控制中发挥着至关重要的作用。 启用Master节点后，在下方选择对应的“节点规格”、“节点数量”和“节点存储”。“节点数量”必须是不小于3的奇数，最多设置9个节点。“节点存储”的存储容量为固定值，存储类型可以根据实际情况选择。
启用Client节点	Client节点负责接收并协调外部请求，如search和write请求，在处理高负载查询、复杂聚合、大量分片管理以及优化集群扩展性方面发挥着重要作用。 启用Client节点后，在下方选择对应的“节点规格”、“节点数量”和“节点存储”。“节点数量”可设置为1~32任意数值。“节点存储”的存储容量为固定值，存储类型可以根据实际情况选择。
启用冷数据节点	冷数据节点用于存储对查询时延要求不高，但数据量较大的历史数据，是管理大规模数据集和优化存储成本的有效方式。 启用冷数据节点后，在下方选择对应的“节点规格”、“节点数量”和“节点存储”。“节点数量”可设置为1~32任意数值。“节点存储”的存储类型和存储容量可以根据实际情况选择。 开启冷数据节点之后，支持切换集群的冷热数据，详情请参见 切换OpenSearch集群冷热数据 。
企业项目	如果开通了“企业项目”，在创建集群时可以给集群绑定一个企业项目。 在下拉框中选择企业项目，单击“查看项目管理”跳转到“企业项目管理”管理控制台，查看已有的企业项目。

- 单击“下一步：网络配置”。
- 在“网络配置”页面，完成OpenSearch集群的网络和安全模式配置。

表 3-12 OpenSearch 集群的网络配置

参数	说明
虚拟私有云	<p>指定集群节点使用的虚拟专用网络，实现不同业务的网络隔离。</p> <p>单击“查看虚拟私有云”跳转到虚拟私有云列表，查看已创建或共享至当前账号下的VPC名称和ID。</p> <p>如果没有合适的VPC，建议联系CSS服务管理员新建VPC，具体请参见创建虚拟私有云和子网。</p> <p>说明 此处选择的VPC必须包含网段（CIDR），否则集群将无法创建成功。新建的VPC默认包含网段（CIDR）。</p>
子网	<p>集群使用子网实现与其他网络的隔离，并独享所有网络资源，以提高网络安全。</p> <p>选择当前虚拟私有云下集群需要的子网。支持选择共享VPC下的子网。</p>
安全组	<p>安全组起着虚拟防火墙的作用，为集群提供安全的网络访问控制策略。</p> <p>选择集群需要的安全组，单击“查看安全组”跳转到安全组列表，可以了解安全组详情。</p> <p>说明 请确保安全组的“端口范围/ICMP类型”为“Any”或者包含端口9200的端口范围。</p>
安全模式	<p>选择是否开启集群安全模式。</p> <ul style="list-style-type: none">默认开启，则创建的是安全模式的集群。安全模式的集群会对集群进行通讯加密和安全认证。因此必须配置集群的“管理员账户名”和“管理员密码”。<ul style="list-style-type: none">管理员账户名默认为admin。设置并确认管理员密码。要记住设置的密码，后续访问集群需要输入密码。不开启，则创建的是非安全模式的集群。非安全模式的集群无需安全认证即可访问，并且采用HTTP明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。

参数	说明
HTTPS访问	<p>只有开启集群的安全模式才可以启用HTTPS访问，开启HTTPS访问后，访问集群将进行通讯加密。</p> <p>说明 安全集群使用HTTPS通信，相比非安全集群使用HTTP通信在读取性能上会降低，预期相对HTTP集群在大并发压力下有20%的性能劣化。如果想要读取性能快，又想要使用安全集群所提供的用户权限隔离资源（索引、文档、字段等）的功能，则可以关闭HTTPS访问。关闭HTTPS访问后，会使用HTTP协议与集群通信，无法保证数据安全性，并且无法开启公网访问功能。</p>
公网访问	<p>仅当集群开启了“安全模式”和“HTTPS访问”时，才可以选择是否配置“公网访问”。配置公网访问后，用户可以获得一个公网访问的IP地址，通过这个IP地址可以在公网访问该安全集群，详细配置请参考配置公网访问。</p>

6. 单击“下一步：高级配置”。
7. 在“高级配置”页面，完成OpenSearch集群的快照和其他高级配置。
 - a. 设置集群快照。

系统默认打开集群快照开关，如果您不需要启用自动快照，可以在“集群快照开关”右侧关闭。自动快照会创建委托访问对象存储服务OBS，快照存储在OBS标准存储中需额外计费。

表 3-13 集群快照基础配置

参数	说明
OBS桶	<p>在下拉框中选择存储快照的OBS桶。也可以单击右侧的“创建桶”新建OBS。详细操作步骤请参见创建桶。</p> <p>创建或者已存在的OBS桶需满足如下条件：</p> <ul style="list-style-type: none">• “存储类别”为“标准存储”。• “区域”须与创建的集群所在区域相同。
备份路径	<p>快照在OBS桶中的存放路径。</p> <p>备份路径配置规则：</p> <ul style="list-style-type: none">• 备份路径不能使用符号“\.*?"<> ”。• 备份路径不能以“/”开头。• 备份路径不能以“.”开头或结尾。• 备份路径的总长度不能超过1023个字符。

参数	说明
IAM委托	<p>指当前账号授权云搜索服务访问或维护存储在OBS中数据。如果没有合适的委托可以联系CSS服务管理员新建IAM委托。详细操作步骤请参见如何创建委托。</p> <p>所选的IAM委托需满足如下条件：</p> <ul style="list-style-type: none">“委托类型”选择“云服务”。“云服务”选择“Elasticsearch”或者“云搜索服务CSS”。必选策略：“OBS Administrator”

表 3-14 设置自动创建快照

参数	说明
快照名称前缀	快照名称前缀的长度为1~32个字符，只能包含小写字母、数字、中划线和下划线，且必须以小写字母开头。快照名称由快照名称前缀加上时间戳组成，例如自动生成的快照名称为“snapshot-1566921603720”。
时区	指备份时间对应的时区，不支持修改。基于此时区选择备份开始时间。
备份开始时间	指每天自动开始备份的时间，只能指定整点时间，如00:00、01:00，取值范围为00:00~23:00。请在下拉框中选择时间。
保留快照个数	<p>自定义设置自动快照保留的个数，范围是1~90。系统在半点时刻会自动删除超过保留个数的快照（过期删除策略只针对与当前自动创建快照策略相同执行频次的自动快照）。</p> <p>说明</p> <p>保存快照个数与自动创建快照策略中设置的执行频次和索引有关，当执行频次间隔时间短或索引数量最大，保留自动快照可能会达不到设置的个数，请谨慎选择。</p>

- b. 配置集群高级功能。根据需要选择“默认配置”或“自定义”。
- “默认配置”：默认关闭“终端节点服务”、“Kibana公网访问”和“标签”功能，在集群创建完成后，如果有需要也可以人工启用这些功能。
 - “自定义”：根据需要选择开启“终端节点服务”、“Kibana公网访问”和“标签”功能。

终端节点服务

终端节点服务（VPC Endpoint Service）通过专属网关，可以将VPC中的服务方便的提供给其他VPC中的资源使用，实现跨VPC的访问，而不必暴露服务端相关的网络信息，使您的访问更加安全、可靠。在开启终端节点服务时，系统会默认给用户创建一个终端节点，内网域名由用户自己选择是否创建，通过节点IP或者是内网域名，用户可以跨VPC访问该集群。

📖 说明

- 当集群的网络配置中，“虚拟私有云”选择的是共享VPC，“子网”选择的是共享VPC下的子网，则该集群不支持配置终端节点服务。
- 集群开启终端节点服务之后，终端节点将按需进行收费，终端节点的费用将由用户进行支付，详细的计费方式请参考[终端节点计费说明](#)。

表 3-15 配置终端节点服务

参数	说明
创建内网域名	勾选“创建内网域名”，系统将会自动为用户创建一个内网域名，通过这个域名可以在同一个VPC内访问该集群。
终端节点服务白名单	<p>在“终端节点服务白名单”中添加允许通过节点IP或内网域名访问集群的账号。</p> <ul style="list-style-type: none">• 单击“添加”输入授权账号ID。授权账号ID配置成“*”，则表示允许全部用户访问该集群。• 单击操作列的“删除”，可以删除不允许访问的账号ID。 <p>说明 “授权账号ID”可以在“我的凭证”中进行查看“账号ID”获取。</p>

Kibana公网访问

只有开启“安全模式”的集群，才能配置Kibana公网访问。开启Kibana公网访问后，用户可以获得一个Kibana公网访问地址，通过这个地址可以在公网访问Kibana。

表 3-16 配置 Kibana 公网访问

参数	说明
带宽	设置公网访问的带宽。 取值范围：1-100。 单位：Mbit/s。
访问控制开关	如果关闭访问控制开关，则允许任何IP通过公网IP访问集群Kibana。如果开启访问控制开关，则只允许白名单列表中的IP通过公网IP访问集群Kibana。
白名单	设置允许访问的IP地址或网段，中间用英文逗号隔开。仅当打开“访问控制开关”时才需要配置。 建议开启白名单。 说明 Kibana公网访问配置白名单依赖ELB的白名单能力。更新白名单后，白名单对新建的连接是实时生效的，但对于已存在的长连接，可能会出现去掉的白名单IP地址还能访问Kibana的场景，这是因为要等长连接断开后才生效，预计1分钟左右。

标签

为集群添加标签，方便用户识别和管理拥有的集群资源。此处您可以选择“标签管理服务”中已定义好的“预定义标签”，也可以自己定义标签。

说明

如您的组织已经设定云搜索服务的相关标签策略，则需按照标签策略规则为集群添加标签。标签不符合标签策略的规则，则可能会导致集群创建失败，请联系组织管理员了解标签策略详情。

表 3-17 标签命名规则

参数	说明
标签键	<ul style="list-style-type: none">对于同一个集群，标签键值唯一。长度不超过64个字符。只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @”。不能以空格开头和结尾。不能为空。
标签值	<ul style="list-style-type: none">长度不超过64个字符。只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @/”。不能以空格开头和结尾。不能为空。

- 单击“下一步：确认配置”，确认完成后单击“立即创建”开始创建集群。
- 单击“返回集群列表”，系统将跳转到“集群管理”页面。您创建的集群将展现在集群列表中，且集群状态为“创建中”，创建成功后集群状态会变为“可用”。

如果集群创建失败，请根据界面提示重新创建集群。

后续操作

OpenSearch集群创建完成后，建议参考[优化集群性能](#)进行集群的性能优化，便于提高集群的写入和查询性能，提升使用效率。

3.4 访问 OpenSearch 集群

3.4.1 OpenSearch 集群访问方式

OpenSearch集群支持多种访问方式，方便根据业务使用的编程语言自行选择接入方式。各种访问方式兼容多种网络配置方式，不同网络配置方式获取到的集群访问地址有差异，您可以根据自己的网络环境选择合适的网络配置方式。

访问方式

表 3-18 集群的访问方式

访问方式	适用场景	相关文档
OpenSearch Dashboards (推荐方式)	<ul style="list-style-type: none">图形化操作与展示。监控实例。管理数据。	通过OpenSearch Dashboards登录OpenSearch集群
Cerebro	不限制访问语言。	通过Cerebro登录OpenSearch集群
开源Elasticsearch API	通过Curl命令访问与管理OpenSearch集群。	通过Curl命令行接入OpenSearch集群

网络配置

配置Elasticsearch集群的网络，确保服务器与集群的网络是互通的。同时，根据选择的网络获取集群的访问地址，除了Kibana和Cerebro一键访问集群不需要填写集群访问地址，其他访问方式接入集群时都需要输入集群访问地址。

表 3-19 集群的网络配置

网络环境	访问方式	场景介绍	相关文档
内网	内网IP地址	<p>集群基本信息中获取各个节点的IP地址，通过直连节点IP地址群访问集群。</p> <p>获取方式：</p> <ol style="list-style-type: none">在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。在集群管理列表页面，选择需要访问的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。 <p>当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。</p>	默认方式

网络环境	访问方式	场景介绍	相关文档
内网	终端节点IP地址或内网域名	<p>通过配置终端节点实现内网场景下跨VPC访问集群。适用于对性能要求不高的场景。</p> <p>获取方式：</p> <ol style="list-style-type: none">1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。2. 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。3. 在左侧菜单栏选择“终端节点服务”，获取并记录终端节点服务的“节点IPv4”或“内网域名”。	配置终端节点服务
内网	独享型负载均衡实例的私有IP地址	<p>通过独享型负载均衡实例分配集群的节点数据。适用于对性能要求高的场景。</p> <p>获取方式：</p> <ol style="list-style-type: none">1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。2. 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。3. 在左侧菜单栏选择“负载均衡”，获取并记录负载均衡实例的“私有IP”。	配置独享型负载均衡
公网	公网IP地址	<p>通过配置集群的公网访问白名单实现公网访问集群。仅适用于启用了HTTPS访问的安全集群。</p> <p>获取方式：</p> <ol style="list-style-type: none">1. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。2. 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。获取并记录“公网访问”。 <p>说明 仅安全模式+HTTPS协议的集群才支持启用公网访问。</p>	配置公网访问

网络环境	访问方式	场景介绍	相关文档
公网	独享型负载均衡实例的公网IP地址	通过独享型负载均衡实例分配集群的节点数据。适用于对性能要求高的场景。 <ol style="list-style-type: none">在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。在左侧菜单栏选择“负载均衡”，获取并记录负载均衡实例的“公网IP”。	配置独享型负载均衡

3.4.2 通过 OpenSearch Dashboards 登录 OpenSearch 集群

CSS服务的OpenSearch集群默认提供OpenSearch Dashboards，无需安装部署，即可一键访问，同时CSS服务的OpenSearch集群也兼容了开源OpenSearch Dashboards的可视化展现和统计分析能力。

OpenSearch Dashboards支持多种访问方式，不同访问方式登录OpenSearch集群的操作有差异，具体请参见[表3-20](#)。

表 3-20 通过 OpenSearch Dashboards 登录 OpenSearch 集群的方式

Kibana访问方式	使用约束	参考文档
控制台一键访问 OpenSearch Dashboards	无。	通过控制台访问 OpenSearch Dashboards登录 OpenSearch集群
公网地址访问 OpenSearch Dashboards	<ul style="list-style-type: none">仅安全模式的集群支持通过Kibana公网访问地址访问OpenSearch Dashboards。	通过公网地址访问 OpenSearch Dashboards登录 OpenSearch集群
内网地址访问 OpenSearch Dashboards	同一VPC下的服务器才能通过 OpenSearch集群的内网访问地址访问 OpenSearch Dashboards。	通过内网地址访问 OpenSearch Dashboards登录 OpenSearch集群

OpenSearch Dashboards 使用限制

OpenSearch Dashboards中可以自定义用户名、角色名、租户名等，不能包含中文字符。

通过控制台访问 OpenSearch Dashboards 登录 OpenSearch 集群

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要登录的集群，单击“操作”列中的“Kibana”进入 OpenSearch Dashboards 登录界面。
 - 非安全模式的集群：将直接进入 OpenSearch Dashboards 操作界面。
 - 安全模式的集群：需要在登录页面输入用户名和密码，单击“Log In”进入 OpenSearch Dashboards 操作界面。用户名默认为 admin，密码为创建集群时设置的管理员密码。
3. 登录成功后，可在 OpenSearch Dashboards 界面进行相关操作访问集群。

通过公网地址访问 OpenSearch Dashboards 登录 OpenSearch 集群

📖 说明

- 仅安全模式的集群支持通过 Kibana 公网访问地址访问 OpenSearch Dashboards。
1. 登录云搜索服务管理控制台。
 2. 开启 OpenSearch 集群的 Kibana 公网访问。支持在创建集群的时候就配置 Kibana 公网访问，或者在集群创建完之后再开启 Kibana 公网访问。
 - 在创建集群时配置 Kibana 公网访问：操作指导请参见 [创建 OpenSearch 集群](#)。
 - 集群创建完之后再开启 Kibana 公网访问：
 - i. 在集群管理页面，单击需要配置 Kibana 公网访问的集群名称，进入集群基本信息页面。
 - ii. 左侧菜单栏选择“Kibana 公网访问”，在“Kibana 公网访问”右侧单击开关，打开 Kibana 公网访问功能。
 - iii. 在开启 Kibana 公网访问页面，配置相关参数。如果集群已配置 Kibana 公网访问，此时也支持修改相关配置。

表 3-21 配置 Kibana 公网访问

参数	说明
带宽	设置公网访问的带宽。 取值范围：1-100。 单位：Mbit/s。
访问控制开关	如果关闭访问控制开关，则允许任何 IP 通过公网 IP 访问集群 Kibana。如果开启访问控制开关，则只允许白名单列表中的 IP 通过公网 IP 访问集群 Kibana。

参数	说明
白名单	设置允许访问的IP地址或网段，中间用英文逗号隔开。仅当打开“访问控制开关”时才需要配置。 建议开启白名单。 说明 Kibana公网访问配置的白名单依赖ELB的白名单能力。更新白名单后，白名单对新建的连接是实时生效的，但对于已存在的长连接，可能会出现去掉的白名单IP地址还能访问Kibana的场景，这是因为要等长连接断开后才生效，预计1分钟左右。

- iv. 配置完成后，单击“确定”。
3. 待集群的Kibana公网访问开通以后，在Kibana公网访问页面获取“kibana公网访问地址”。

图 3-1 获取 Kibana 公网访问地址



4. 在浏览器中，输入“kibana公网访问地址”即可进入OpenSearch Dashboards登录界面。
在登录页面输入用户名和密码，单击“Log In”进入OpenSearch Dashboards操作界面。用户名默认为admin，密码为创建集群时设置的管理员密码。
5. 登录成功后，可在OpenSearch Dashboards界面进行相关操作访问OpenSearch集群。

通过内网地址访问 OpenSearch Dashboards 登录 OpenSearch 集群

📖 说明

同一VPC下的服务器才能通过OpenSearch集群的内网访问地址访问OpenSearch Dashboards。

1. 登录云搜索服务管理控制台。
2. 在OpenSearch集群管理页面，单击集群名称，进入集群基本信息页面。
3. 在集群基本信息页面，获取集群的内网访问地址。

图 3-2 获取内网地址

配置信息	
区域	
可用区	
虚拟私有云	vpc-
子网	subnet-
安全组	dws- 更改安全组
安全模式	未开启
企业项目	default
HTTPS访问	关闭
内网访问IPv4地址	

4. 将集群的内网访问地址的端口号从“9200”改为“5601”，即为OpenSearch Dashboards的内网地址。例如，集群的“内网访问IPv4地址”是“192.168.0.***:9200”，则OpenSearch Dashboards的内网地址为“192.168.0.***:5601”。
5. 在服务器中，输入OpenSearch Dashboards的内网地址即可进入OpenSearch Dashboards登录界面。
 - 非安全模式的集群：将直接进入OpenSearch Dashboards操作界面。
 - 安全模式的集群：需要在登录页面输入用户名和密码，单击“Log In”进入OpenSearch Dashboards操作界面。用户名默认为admin，密码为创建集群时设置的管理员密码。
6. 登录成功后，可在OpenSearch Dashboards界面进行相关操作访问OpenSearch集群。

3.4.3 通过 Cerebro 登录 OpenSearch 集群

CSS服务的ELasticsearch和OpenSearch集群默认提供Cerebro，无需安装部署，即可一键访问Cerebro，同时CSS服务的ELasticsearch和OpenSearch集群也完全兼容开源Cerebro。

通过 CSS 控制台的 Cerebro 访问集群

CSS服务创建的ELasticsearch和OpenSearch集群自带Cerebro组件，支持一键打开Cerebro，快速访问ELasticsearch和OpenSearch集群。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要登录的集群，单击“操作”列中的“更多 > Cerebro”进入Cerebro登录页面。
 - 非安全模式的集群：单击Cerebro登录页面的集群名称即可进入Cerebro操作界面。

- 安全模式的集群：单击Cerebro登录页面的集群名称，再输入用户名和密码，单击“Authenticate”进入Cerebro操作界面。用户名默认为admin，密码为创建集群时设置的管理员密码。
3. 登录成功后，可在Cerebro界面进行相关操作访问集群。

通过自建 Cerebro 访问集群

📖 说明

使用自建Cerebro访问集群时，需要确保自建Cerebro与集群的网络是互通的。

1. 获取集群的访问地址。不同网络配置的访问地址获取方式请参见[网络配置](#)。
2. 启动自建Cerebro，填写集群的访问地址对接CSS集群。
 - 安全模式的集群填写：`https://访问地址:9200`
安全模式的集群要输入集群的用户名和密码才能登录。
 - 非安全模式的集群填写：`http://访问地址:9200`

3.4.4 通过 Curl 命令行接入 OpenSearch 集群

开源ELasticsearch提供了一系列RESTful风格的API，通过Curl命令可以在Kibana、Postman等工具中使用这些API。本文介绍如何通过Curl命令接入ELasticsearch和OpenSearch集群。

前提条件

- CSS服务的ELasticsearch和OpenSearch集群处于可用状态。
- 已创建一个ECS服务，且和CSS服务的集群在同一个虚拟私有云和安全组中。
 - 如果ECS和CSS服务的集群不在同一安全组中，请修改ECS安全组或者配置ECS安全组的出入规则，允许集群所有安全组的访问。修改操作请参见[配置安全组规则](#)。
 - ECS的使用请参见[弹性云服务器ECS的操作导航](#)。

操作步骤

1. 获取集群的内网访问地址。访问集群时，需要输入内网访问地址。
本文以内网IP地址方式访问集群为例，当使用其他网络配置访问集群时，仅访问地址有差异，不同访问地址的获取方式请参见[网络配置](#)。
 - a. 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。
 - b. 在集群管理列表页面，选择需要访问的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。
当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。
2. 在ECS中执行如下命令，访问集群。集群的安全模式不同，访问命令也不同。
 - 非安全模式的集群

```
curl "http://<host>:<port>"
```
 - 安全模式+HTTP协议的集群

```
curl -u <user>:<password> "http://<host>:<port>"
```

- 安全模式+HTTPS协议的集群
`curl -u <user>:<password> -k "https://<host>:<port>"`

表 3-22 变量说明

变量名	说明
<host>	集群中各节点的IP地址，当集群包含多个节点时，会存在多个IP地址，可以任选其中一个发送。
<port>	集群节点的访问端口号，一般为9200。
<user>	访问集群的用户名。
<password>	用户名对应的密码。 当密码中存在特殊字符时，请将用户名和密码加上英文双引号，例如“curl -u "user:password!" "http://<host>:<port>"”。

访问示例如下：

```
curl "http://10.62.176.32:9200"
```

返回结果如下：

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8
content-length: 513

{
  "name" : "xxx-1",
  "cluster_name" : "xxx",
  "cluster_uuid" : "xxx_uuid",
  "version" : {
    "number" : "7.10.2",
    "build_flavor" : "oss",
    "build_type" : "tar",
    "build_hash" : "unknown",
    "build_date" : "unknown",
    "build_snapshot" : true,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.7.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

说明

更多命令，请参见[Elasticsearch官方文档](#)。

3.5 导入数据至 OpenSearch 集群

3.5.1 OpenSearch 集群导入数据方式

OpenSearch 集群导入数据方式介绍

OpenSearch 集群支持通过多种方式导入数据，基于业务使用场景可以自行选择导入数据的方式，具体请参见[表3-23](#)。在导入数据前可以根据需要选择是否对Elasticsearch 集群进行导入性能增强，具体操作请参见[增强OpenSearch 集群数据导入性能](#)。

表 3-23 OpenSearch 集群导入数据的方式

导入数据方式	适用场景	支持数据格式	相关文档
数据处理管道 Logstash	Logstash开源服务器端实时数据处理管道，支持多个来源采取数据。适用场景多，比如日志数据、监控数据、metric数据等流数据。	JSON、CSV、文本等多种格式	使用自建Logstash导入数据到OpenSearch
开源Elasticsearch API	使用开源Elasticsearch API导入数据，使用灵活，适用于开发自己写的应用代码场景。	JSON	使用开源OpenSearch API导入数据到OpenSearch
云数据迁移服务 CDM	CDM向导式页面。适用于批数据迁移，比如数据存放在OBS或者Oracle数据库中，推荐使用CDM比较方便。	JSON	使用CDM导入数据到OpenSearch
数据复制服务DRS	DRS用于数据库在线迁移和实时同步数据的云服务。	关系型数据库	使用DRS从数据库导入数据至OpenSearch

使用 DRS 从数据库导入数据至 OpenSearch

数据复制服务（Data Replication Service，简称DRS）是一种易用、稳定、高效、用于数据库在线迁移和数据库实时同步的云服务。实时同步是指在不同的系统之间，将数据通过同步技术从一个数据源复制到其他数据库，并保持一致，实现关键业务的数据实时流动。

使用DRS支持将多种关系型数据库的数据导入至Elasticsearch集群，源数据库和目标集群的版本支持情况请参见[表3-24](#)。

表 3-24 使用 DRS 从数据库导入数据至 OpenSearch

数据导入场景	源数据库	目标OpenSearch集群	参考文档
将RDS for MySQL数据库的数据导入到CSS Elasticsearch集群	RDS for MySQL 5.5、5.6、5.7、8.0 版本	OpenSearch 1.3.6 版本	将MySQL同步到CSS/ES
将GaussDB(for MySQL)数据库的数据导入到CSS Elasticsearch集群	GaussDB(for MySQL)主备实例	OpenSearch 1.3.6 版本	将GaussDB(for MySQL)同步到CSS/ES
将自建MySQL数据库的数据导入到CSS Elasticsearch集群	本地或ECS自建MySQL数据库 5.5、5.6、5.7、8.0 版本	OpenSearch 1.3.6 版本	将MySQL同步到CSS/ES

3.5.2 使用自建 Logstash 导入数据到 OpenSearch

云搜索服务支持使用自建Logstash将其收集的数据迁移到Elasticsearch和OpenSearch中，方便用户通过CSS搜索引擎高效管理和获取数据。数据文件支持JSON、CSV等格式。

Logstash是开源的服务器端数据处理管道，能够同时从多个来源采集数据、转换数据，然后将数据发送到Elasticsearch或OpenSearch中。Logstash的官方文档请参见：<https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html>。

数据导入分为如下2种场景：

- [Logstash部署在外网时导入数据](#)
- [Logstash部署在弹性云服务器上时导入数据](#)

前提条件

- 为方便操作，建议采用Linux操作系统的机器部署Logstash。
- Logstash的下载路径为：<https://www.elastic.co/cn/downloads/logstash-oss>

📖 说明

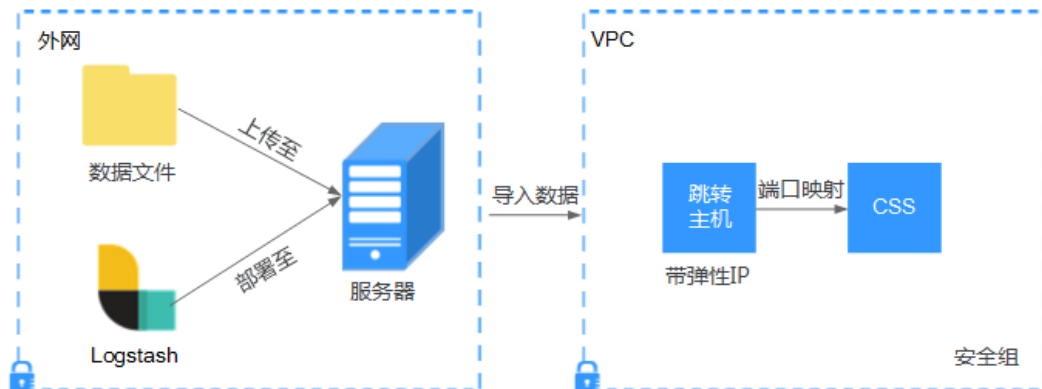
Logstash要求使用OSS版本，选择和CSS一致版本。

- 安装Logstash之前，需要先安装JDK。在Linux操作系统中，您可以执行**yum -y install java-1.8.0**命令直接安装1.8.0版本JDK。在Windows操作系统中，您可以访问[JDK官网](#)，下载符合操作系统版本的JDK，并根据指导安装。
- 安装完Logstash后，再根据如下步骤导入数据。安装Logstash的操作指导，请参见：<https://www.elastic.co/guide/en/logstash/current/installing-logstash.html>
- 在“[Logstash部署在弹性云服务器上时导入数据](#)”场景中，请确保此弹性云服务器与接入的Elasticsearch集群在同一个VPC下。

Logstash 部署在外网时导入数据

当Logstash部署在外网时，导入数据的流程说明如图3-3所示。

图 3-3 Logstash 部署在外网时导入数据示意图



1. 创建一个跳转主机，并按如下要求进行配置。
 - 跳转主机为一台Linux操作系统的弹性云服务器，且已绑定弹性IP。
 - 跳转主机与CSS集群在同一虚拟私有云下。
 - 已开放跳转主机的本地端口，用于SSH转发，能够从本地端口转发至CSS集群某一节点的9200端口。
 - 关于跳转主机的本地端口转发配置，请参见[SSH官方文档](#)。
2. 使用PuTTY，通过弹性IP登录已创建的跳转主机。
3. 执行如下命令进行端口映射，将发往跳转主机对外开放端口的请求转发到待导入数据的集群中。

```
ssh -g -L <跳转主机的本地端口:节点的公网访问地址和端口号> -N -f root@<跳转主机的公网IP地址>
```

说明

- <跳转主机的本地端口>：为步骤1中的端口。
- <节点的公网访问地址和端口号>：为集群中某一节点的公网访问地址和端口号。当该节点出现故障时，将导致命令执行失败。如果集群包含多个节点，可以将<节点的公网访问地址和端口号>替换为集群中另一节点的公网访问地址和端口号；如果集群只包含一个节点，则需要将该节点修复之后再次执行命令进行端口映射。
- <跳转主机的公网IP地址>：打开弹性云服务器管理控制台，从“IP地址”列中获取标有“公网”对应的IP地址。

例如：跳转主机对外开放的端口号为9200，节点的公网访问地址和端口号为192.168.0.81:9200，跳转主机的公网IP地址为192.168.0.227，需要执行如下命令进行端口映射。

```
ssh -g -L 9200:192.168.0.81:9200 -N -f root@192.168.0.227
```

4. 登录部署了Logstash的服务器，将需要进行操作的数据文件存储至此服务器中。例如，需要导入的数据文件“access_20181029_log”，文件存储路径为“/tmp/access_log/”，此数据文件中包含的数据如下所示：

说明

文件存储路径中的access_log文件夹如果不存在，用户可以自建。

All	Heap used for segments	18.6403	MB
All	Heap used for doc values	0.119289	MB

All	Heap used for terms		17.4095	MB
All	Heap used for norms		0.0767822	MB
All	Heap used for points		0.225246	MB
All	Heap used for stored fields		0.809448	MB
All	Segment count		101	
All	Min Throughput	index-append	66232.6	docs/s
All	Median Throughput	index-append	66735.3	docs/s
All	Max Throughput	index-append	67745.6	docs/s
All	50th percentile latency	index-append	510.261	ms

5. 在部署Logstash的服务器中，执行如下命令在Logstash的安装目录下新建配置文件logstash-simple.conf。

```
cd /<Logstash的安装目录>  
vi logstash-simple.conf
```

6. 在配置文件logstash-simple.conf中输入如下内容。

```
input {  
  数据所在的位置  
}  
filter {  
  数据的相关处理  
}  
output {  
  elasticsearch {  
    hosts => "<跳转主机的公网IP地址>:<跳转主机对外开放的端口号>"  
  }  
}
```

- input: 指定数据的来源。实际请根据用户的具体情况来设置。input参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/input-plugins.html>。
- filter: 指定对数据进行处理的方式。例如，对日志进行了提取和处理，将非结构化信息转换为结构化信息。filter参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/filter-plugins.html>。
- output: 指定数据的目的地地址。output参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/output-plugins.html>。<跳转主机的公网IP地址>请从弹性云服务器管理控制台的“IP地址”列中获取标有“弹性公网”对应的IP地址。<跳转主机对外开放的端口号>即为步骤1中的端口，例如：9200。

以步骤4中“/tmp/access_log/”的数据文件为例，输入数据文件从首行开始，且过滤条件保持为空，即不做任何数据处理操作。跳转主机的公网IP和端口号为“192.168.0.227:9200”。导入数据的索引名称为“myindex”。配置文件的示例如下所示，配置文件按实际数据情况修改完成后，输入“:wq”保存。

```
input {  
  file{  
    path => "/tmp/access_log/*"  
    start_position => "beginning"  
  }  
}  
filter {  
}  
output {  
  elasticsearch {  
    hosts => "192.168.0.227:9200"  
    index => "myindex"  
  }  
}
```

📖 说明

如果在使用中出现license相关的报错，可以尝试设置ilm_enabled => false。
如果集群开启了安全模式，则需要先下载证书。

- a. 在云搜索服务管理控制台。
- b. 在“集群管理”页面，单击需要下载证书的集群名称，进入集群基本信息页面。
- c. 在集群基本信息页面下载证书。

图 3-4 下载证书**配置信息**

区域	
可用区	
虚拟私有云	vpc-
子网	subn
安全组	dws 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	-- 绑定
HTTPS访问	开启 下载证书
内网访问IPv4地址	

- d. 将下载的证书存放到部署logstash服务器中。
- e. 修改配置文件logstash-simple.conf。
以步骤4中“/tmp/access_log/”的数据文件为例，输入数据文件从首行开始，且过滤条件保持为空，即不做任何数据处理操作。跳转主机的公网IP和端口号为“192.168.0.227:9200”。导入数据的索引名称为“myindex”，证书存放路径为“/logstash/logstash6.8/config/CloudSearchService.cer”。配置文件的示例如下所示，配置文件按实际数据情况修改完成后，输入“:wq”保存。

```
input{
  file {
    path => "/tmp/access_log/"
    start_position => "beginning"
  }
}
filter {
}
```



```
output{
  elasticsearch{
    hosts => ["https://192.168.0.227:9200"]
    index => "myindex"
    user => "admin"
    password => "*****"
    cacert => "/logstash/logstash6.8/config/CloudSearchService.cer"
    manager_template => false
    ilm_enabled => false
    ssl => true
    ssl_certificate_verification => false
  }
}
```

📖 说明

password：登录安全集群的密码。

7. 执行如下命令将Logstash收集的数据导入到集群中。

```
./bin/logstash -f logstash-simple.conf
```

📖 说明

此命令需要在存放logstash-simple.conf文件的目录下执行。例如，logstash-simple.conf文件存放在/root/logstash-7.1.1/，则需要先进入该路径，再执行此命令。

8. 登录云搜索服务管理控制台。
9. 在左侧导航栏中，对应的集群类型，进入集群管理列表页面。
10. 在集群列表页面中，单击待导入数据的集群“操作”列的“Kibana”。
11. 在Kibana的左侧导航中选择“Dev Tools”，进入Console界面。
12. 在已打开的Kibana的Console界面，通过搜索获取已导入的数据。

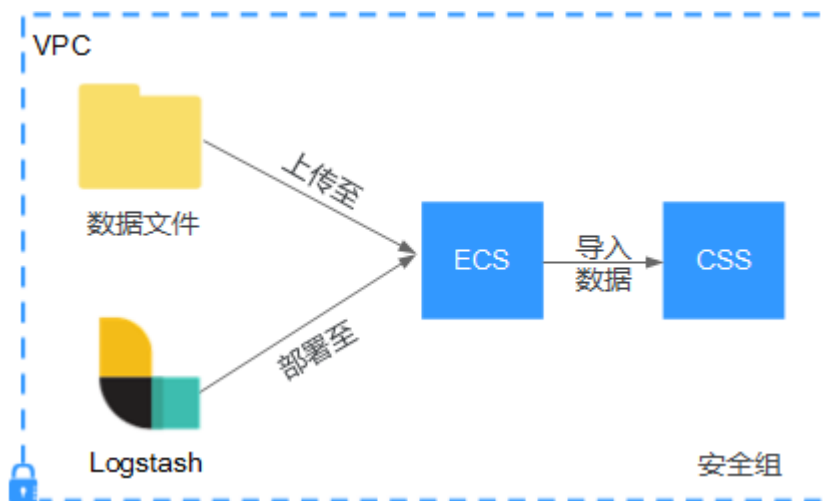
在Kibana控制台，输入如下命令，搜索数据。查看搜索结果，如果数据与导入数据一致，表示数据文件的数据已导入成功。

```
GET myindex/_search
```

Logstash 部署在弹性云服务器上时导入数据

当Logstash部署在同一VPC的弹性云服务时，导入数据的流程说明如图3-5所示。

图 3-5 Logstash 部署在弹性云服务器上时导入数据示意图



1. 确保已部署Logstash的弹性云服务器与待导入数据的集群在同一虚拟私有云下，已开放安全组的9200端口的公网访问权限，且弹性云服务器已绑定弹性IP。

📖 说明

- 如果同一个VPC内有多台服务器，只要其中一台绑定了弹性IP，其他的服务器可以不需要绑定弹性IP。通过绑定弹性IP的节点跳转到部署Logstash的节点即可。
- 如果有专线或者VPN，也不需要绑定弹性IP。

2. 使用PuTTY登录弹性云服务器。

例如此服务器中存储了需要导入的数据文件“access_20181029_log”，文件存储路径为“/tmp/access_log/”，此数据文件中包含的数据如下所示：

```
| All | Heap used for segments | 18.6403 | MB | |
| All | Heap used for doc values | 0.119289 | MB |
| All | Heap used for terms | 17.4095 | MB |
| All | Heap used for norms | 0.0767822 | MB |
| All | Heap used for points | 0.225246 | MB |
| All | Heap used for stored fields | 0.809448 | MB |
| All | Segment count | 101 |
| All | Min Throughput | index-append | 66232.6 | docs/s |
| All | Median Throughput | index-append | 66735.3 | docs/s |
| All | Max Throughput | index-append | 67745.6 | docs/s |
| All | 50th percentile latency | index-append | 510.261 | ms |
```

3. 执行如下命令在Logstash的安装目录下新建配置文件logstash-simple.conf。

```
cd /<Logstash的安装目录>
vi logstash-simple.conf
```

在配置文件logstash-simple.conf中输入如下内容。

```
input {
  <数据所在的位置>
}
filter {
  <数据的相关处理>
}
output {
  elasticsearch{
    hosts => "<节点的内网访问地址和端口号>"
  }
}
```

- input: 指定数据的来源。实际请根据用户的具体情况来设置。input参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/input-plugins.html>。
- filter: 对日志进行了提取和处理，将非结构化信息转换为结构化信息。filter参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/filter-plugins.html>。
- output: 指定数据的地址。output参数的详细解释和使用介绍请参见<https://www.elastic.co/guide/en/logstash/current/output-plugins.html>。<节点的内网访问地址和端口号>为集群中节点的内网访问地址和端口号。

当集群包含多个节点时，为了避免节点故障，建议将上述命令中<节点的内网访问地址和端口号>替换为该集群中多个节点的内网访问地址和端口号，多个节点的内网访问地址和端口号之间用英文逗号隔开，填写格式请参见如下示例。

```
hosts => ["192.168.0.81:9200","192.168.0.24:9200"]
```

当集群只包含一个节点时，填写格式请参见如下示例。

```
hosts => "192.168.0.81:9200"
```

以步骤2中“/tmp/access_log/”的数据文件为例，输入数据文件从首行开始，且过滤条件保持为空，即不做任何数据处理操作。需导入数据的集群，其节点内网访问地址和端口号为“192.168.0.81:9200”。导入数据的索引名称为“myindex”。配置文件的示例如下所示，配置文件按实际数据情况修改完成后，输入“:wq”保存。

```
input {
  file {
    path => "/tmp/access_log/*"
    start_position => "beginning"
  }
}
filter {
}
output {
  elasticsearch {
    hosts => "192.168.0.81:9200"
    index => "myindex"
  }
}
```

如果集群开启了安全模式，则需要先下载证书。

- a. 在集群基本信息页面下载证书。

图 3-6 下载证书

配置信息

区域	
可用区	
虚拟私有云	vpc-
子网	subn
安全组	dws 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	-- 绑定
HTTPS访问	开启 下载证书
内网访问IPv4地址	

- b. 将下载的证书存放部署logstash服务器中。
- c. 修改配置文件logstash-simple.conf。
以步骤2中“/tmp/access_log/”的数据文件为例，输入数据文件从首行开始，且过滤条件保持为空，即不做任何数据处理操作。跳转主机的公网IP和

端口号为“192.168.0.227:9200”。导入数据的索引名称为“myindex”，证书存放路径为“/logstash/logstash6.8/config/CloudSearchService.cer”。配置文件的示例如下所示，配置文件按实际数据情况修改完成后，输入“:wq”保存。

```
input{
  file {
    path => "/tmp/access_log/*"
    start_position => "beginning"
  }
}
filter {
}
output{
  elasticsearch{
    hosts => ["https://192.168.0.227:9200"]
    index => "myindex"
    user => "admin"
    password => "*****"
    cacert => "/logstash/logstash6.8/config/CloudSearchService.cer"
    manager_template => false
    ilm_enabled => false
    ssl => true
    ssl_certificate_verification => false
  }
}
```

📖 说明

password：登录安全集群的密码。

4. 执行如下命令将Logstash收集的弹性云服务器的数据导入到集群中。
`./bin/logstash -f logstash-simple.conf`
5. 登录云搜索服务管理控制台。
6. 在左侧导航栏中，选择对应的集群类型，进入集群管理列表页面。
7. 在集群列表页面中，单击待导入数据的集群“操作”列的“Kibana”。
8. 在Kibana的左侧导航中选择“Dev Tools”，进入Console界面。
9. 在已打开的Kibana的Console界面，通过搜索获取已导入的数据。

在Kibana控制台，输入如下命令，搜索数据。查看搜索结果，如果数据与导入数据一致，表示数据文件的数据已导入成功。

```
GET myindex/_search
```

3.5.3 使用开源 OpenSearch API 导入数据到 OpenSearch

云搜索服务支持在Kibana或者ECS服务器上使用开源Elasticsearch API将数据导入到OpenSearch集群中，数据文件支持JSON等格式。

- **在OpenSearch Dashboards上使用开源Elasticsearch API导入数据**：使用POST命令导入数据。
- **在ECS服务器上使用开源Elasticsearch API导入数据**：使用Curl命令导入数据。

在 OpenSearch Dashboards 上使用开源 Elasticsearch API 导入数据

在OpenSearch Dashboards上支持通过POST命令使用开源Elasticsearch API导入单条数据。

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择“集群管理 > OpenSearch”，进入集群管理列表页面。

3. 选择待导入数据的集群，单击操作列“Kibana”，登录OpenSearch Dashboards。
4. 单击左侧导航栏的“Dev Tools”进入操作页面。
5. 在OpenSearch Dashboards操作页面，执行命令查看集群是否存在索引。
`GET _cat/indices?v`
 - 当待导入数据的集群已存在可用的索引时，则不需要再创建索引，直接执行步骤7。
 - 当待导入数据的集群不存在可用的索引时，则需要执行下一步创建索引。
6. 在OpenSearch Dashboards执行命令，创建待导入数据的索引，并指定自定义映射来定义数据类型。

例如执行如下命令，创建索引“my_store”。

```
PUT /my_store
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "productName": {
        "type": "text"
      },
      "size": {
        "type": "keyword"
      }
    }
  }
}
```

7. 在OpenSearch Dashboards执行命令导入数据，以导入一条数据为例，执行如下命令。

```
POST /my_store/_bulk
{"index":{}}
{"productName":"Latest art shirts for women in 2017 autumn","size":"L"}
```

返回结果如[图3-7](#)所示，当返回结果信息中“errors”字段的值为“false”时，表示导入数据成功。

图 3-7 返回消息

```
1 {
2   "took": 42,
3   "errors": false,
4   "items": [
5     {
6       "index": {
7         "_index": "my_store",
8         "_type": "products",
9         "_id": "AWTGbHt7BwpN-hb3LKau",
10        "_version": 1,
11        "result": "created",
12        "_shards": {
13          "total": 2,
14          "successful": 2,
15          "failed": 0
16        },
17        "created": true,
18        "status": 201
19      }
20    ]
21  }
```

在 ECS 服务器上使用开源 Elasticsearch API 导入数据

在ECS服务器上支持通过Curl命令使用开源Elasticsearch API导入JSON数据文件。

本案例以非安全模式的集群为例介绍Curl命令导入数据，如果是安全模式的集群访问集群的命令请参见[通过Curl命令行接入OpenSearch集群](#)。

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择“集群管理 > OpenSearch”，进入集群管理列表页面。
3. 在集群管理列表页面，选择待导入数据的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>和端口<port>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。

当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。

4. 登录即将接入集群的弹性云服务器ECS。
弹性云服务器器的使用指导请参见[快速购买和使用Linux ECS](#)。
5. 将JSON数据文件上传至ECS。

例如，将如下数据保存为json格式的文件，上传到ECS。

```
{"index": {"_index": "my_store"}}
{"productName": "2019秋装新款文艺衬衫女装", "size": "M"}
{"index": {"_index": "my_store"}}
{"productName": "2019秋装新款文艺衬衫女装", "size": "L"}
```

6. 在ECS中存放JSON数据文件的路径下，执行如下命令将JSON数据文件导入到Elasticsearch集群。

```
curl -X PUT "http://{{Private network address and port number of the node}}/_bulk" -H 'Content-Type: application/json' --data-binary @test.json
```

其中，**{Private network address and port number of the node}**需替换为集群中节点的内网访问地址和端口号。当该节点出现故障时，将导致命令执行失败，如果集群包含多个节点，可以将**{Private network address and port number of the node}**替换为集群中另一节点的内网访问地址和端口号，如果集群只包含一个节点，则需要将该节点修复之后再次执行命令进行导入数据。

“test.json”为导入数据的json文件。

📖 说明

其中，-X参数的参数值为命令，如“-X PUT”，-H参数的参数值为消息头，如“-H 'Content-Type: application/json' --data-binary @test.json”。添加的-k参数时，请勿将-k参数放置在参数与参数值之间。

示例：将“test.json”数据文件中的数据导入至OpenSearch集群，此集群未进行通信加密，其中一个节点内网访问地址为“192.168.0.90”，端口号为“9200”。

- a. 执行如下命令，创建“my_store”索引。

```
curl -X PUT http://192.168.0.90:9200/my_store -H 'Content-Type: application/json' -d '{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "productName": {
        "type": "text"
      },
      "size": {
        "type": "keyword"
      }
    }
  }
}'
```

- b. 执行如下命令，导入“test.json”文件中的数据。

```
curl -X PUT "http://192.168.0.90:9200/_bulk" -H 'Content-Type: application/json' --data-binary @test.json
```

本案例回显如下信息，表示数据导入成功。

```
{"took":204,"errors":false,"items":[{"index":
{"_index":"my_store","_type":"_doc","_id":"DJQkBlwBbJvUd2769Wi","_version":1,"result":"created",
"_shards":{"total":2,"successful":1,"failed":0},"_seq_no":0,"_primary_term":1,"status":201}},
{"index":
{"_index":"my_store","_type":"_doc","_id":"DZQkBlwBbJvUd2769Wi","_version":1,"result":"create
d","_shards":{"total":2,"successful":1,"failed":0},"_seq_no":1,"_primary_term":1,"status":201}}]}
```

3.5.4 使用 CDM 导入数据到 OpenSearch

云搜索服务支持通过云数据迁移服务CDM的向导式界面，将存储在Oracle数据库或对象存储服务OBS中的数据导入到Elasticsearch或OpenSearch集群中，数据文件支持JSON格式。

表 3-25 使用 CDM 导入数据至 CSS 服务

数据导入场景	源数据	目标集群
将Oracle数据库的数据导入到CSS服务	本地或第三方Oracle数据库	ElasticSearch 5.5、6.2、6.5、7.1、7.6、7.9、7.10版本 OpenSearch 1.3.6版本

数据导入场景	源数据	目标集群
将OBS的数据导入到CSS服务	OBS桶中的JSON数据	ElasticSearch 5.5、6.2、6.5、7.1、7.6、7.9、7.10版本 OpenSearch 1.3.6版本

准备工作

1. 确认源数据。
例如，源数据为如下json格式的文件。
Elasticsearch版本<7.x的数据

```
{"index": {"_index": "my_store", "_type": "products"}}  
{"productName": "2019秋装新款文艺衬衫女装", "size": "M"}  
{"index": {"_index": "my_store", "_type": "products"}}  
{"productName": "2019秋装新款文艺衬衫女装", "size": "L"}
```


Elasticsearch版本≥7.x或Opensearch版本的数据

```
{"index": {"_index": "my_store"}}  
{"productName": "2019秋装新款文艺衬衫女装", "size": "M"}  
{"index": {"_index": "my_store"}}  
{"productName": "2019秋装新款文艺衬衫女装", "size": "L"}
```
2. 获取源数据信息。
 - 当源数据是Oracle数据库时，需要获取Oracle数据库的IP、数据库名、用户名和密码。
 - 当源数据是OBS桶中的JSON数据时，需要获取OBS的访问域名、端口，以及AK、SK。
3. 当源数据是Oracle数据库时，需要确保Oracle可通过公网IP访问，或者已经建立好了企业内部数据中心到华为云的VPN通道或专线。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择对应集群类型，进入集群管理列表页面。
3. 在集群管理列表页面，选择待导入数据的集群，在“内网访问地址”列获取并记录集群的内网IP地址<host>和端口<port>，一般是“<host>:<port>”或“<host>:<port>,<host>:<port>”样式。
当集群只有一个节点时，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；当集群有多个节点时，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。
4. 在集群列表，单击待导入数据的集群操作列的“Kibana”，登录Kibana。
5. 在Kibana的左侧导航中选择“Dev Tools”，进入操作页面。
6. 在Kibana执行命令查看集群是否存在索引。

```
GET _cat/indices?v
```

 - 当待导入数据的集群已存在可用的索引时，则不需要再创建索引，直接执行步骤8。
 - 当待导入数据的集群不存在可用的索引时，则需要执行下一步创建索引。
7. 在Kibana执行命令创建待导入数据的索引，并指定自定义映射来定义数据类型。例如：执行如下命令，创建索引“demo”。

Elasticsearch版本<7.x的命令

```
PUT /demo
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "products": {
      "properties": {
        "productName": {
          "type": "text",
          "analyzer": "ik_smart"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
}
```

Elasticsearch版本≥7.x或Opensearch版本的命令

```
PUT /demo
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "properties": {
      "productName": {
        "type": "text",
        "analyzer": "ik_smart"
      },
      "size": {
        "type": "keyword"
      }
    }
  }
}
```

执行成功后显示如下：

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "demo"
}
```

8. 登录云数据迁移CDM管理控制台，通过CDM集群将Oracle或OBS的数据导入Elasticsearch或OpenSearch集群。
 - 当源数据是Oracle数据库时，请参考[Oracle数据迁移到云搜索服务](#)完成数据导入。
 - 当源数据是OBS桶中的JSON数据时，请参考[OBS数据迁移到云搜索服务](#)完成数据导入。
9. 当数据迁移完成后，重新进入Elasticsearch或OpenSearch集群的Kibana操作页面，搜索已导入的数据。

执行如下命令，搜索数据。查看搜索结果，如果数据与源数据一致，表示数据导入成功。

```
GET demo/_search
```

“demo”为创建的索引名称，需根据实际情况填写。

执行成功后显示如下：

```
{
  "took": 18,
  "timed_out": false,
```

```
"_shards": {
  "total": 1,
  "successful": 1,
  "skipped": 0,
  "failed": 0
},
"hits": {
  "total": 2,
  "max_score": 1,
  "hits": [
    {
      "_index": "demo",
      "_type": "products",
      "_id": "g6UepnEBuvdFwWkRmn4V",
      "_score": 1,
      "_source": {
        "size": "size:L",
        "productName": "2019秋装新款文艺衬衫女装"
      }
    },
    {
      "_index": "demo",
      "_type": "products",
      "_id": "hKUepnEBuvdFwWkRmn4V",
      "_score": 1,
      "_source": {
        "size": "size:M",
        "productName": "2019秋装新款文艺衬衫女装"
      }
    }
  ]
}
```

3.5.5 增强 OpenSearch 集群数据导入性能

特性介绍

数据导入性能增强为云搜索服务自主研发的特性，通过优化Bulk路由、文本索引加速、分词加速等方式，有效提升导入性能以及降低写入拒绝的情况。适用于索引分片较多、文本索引量大、导入吞吐量高的场景。

表 3-26 数据导入性能优化方式

优化方式	功能描述	相关文档
Bulk路由优化	<p>根据Elasticsearch默认的路由规则，Bulk请求中的每一条数据会被路由到不同的shard，当索引分片较多时，会产生大量的内部转发请求，在大规模写入场景下容易触发写拒绝。同时，在大规模集群中，长尾效应会导致Bulk请求时延较高。</p> <p>通过指定配置项“index.bulk_routing”可以开启集群的Bulk路由优化，该优化可以减少内部转发的请求数量，在shard数较多的场景下，能够有效提升写入性能以及减少写入拒绝。</p> <p>说明</p> <p>开启Bulk路由优化后（即“index.bulk_routing”设置为“pack”或“ocal_pack”），数据写入不再根据“_id”进行路由，与路由的相关功能使用会受限，例如根据“_id”进行文档GET请求可能失败。</p>	Bulk路由优化

优化方式	功能描述	相关文档
Bulk聚合优化	通过指定配置项“index.aggr_perf_batch_size”可以开启集群的Bulk聚合优化。Bulk聚合优化是通过批量导入将Bulk请求中的doc从单个依次写入变为批量写入，该方案可以有效减少内存申请、锁申请、及其他调用开销，从而提升数据导入性能。	Bulk聚合优化
文本索引加速	通过指定配置项“index.native_speed_up”可以开启文本索引加速。索引加速功能通过优化索引流程以及内存使用等方式实现，对于文本字段（text、keyword）能够极大提升索引构建的性能。当开启文本索引加速时，支持通过指定配置项“index.native_analyzer”同时开启分词加速。对于需要分词的文本字段（text），当无特殊分词需求时可以开启分词加速提升分词性能。 说明 <ul style="list-style-type: none">仅当开启文本索引加速（即“index.native_speed_up”设置为“true”）时，才支持开启分词加速（即“index.native_analyzer”设置为“true”），否则分词加速不生效。包含“nested”字段的索引不支持开启文本索引加速。	文本索引加速
索引merge任务优化	开启以上三种数据导入性能优化后，集群的索引merge任务会增加，通过指定配置项“index.merge.scheduler.max_thread_count”可以降低索引merge任务开销对导入性能的影响。索引merge任务优化可以增加shard的合并线程数，减少合并对数据导入的限流。	索引merge任务优化

约束限制

仅Elasticsearch 7.10.2集群和OpenSearch 1.3.6集群支持数据导入性能增强。

前提条件

待增强数据导入性能的集群处于“可用”状态。

Bulk 路由优化

通过指定配置项“index.bulk_routing”可以开启集群的Bulk路由优化，该优化可以减少内部转发的请求数量，在shard数较多的场景下，能够有效提升写入性能以及减少写入拒绝。

说明

开启Bulk路由优化后（即“index.bulk_routing”设置为“pack”或“ocal_pack”），数据写入不再根据“_id”进行路由，与路由的相关功能使用会受限，例如根据“_id”进行文档GET请求可能失败。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。
2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令开启Bulk路由优化。

```
PUT my_index
{
  "settings": {
    "index.bulk_routing": "local_pack"
  }
}
```

配置项“index.bulk_routing”的取值范围如下所示。

- “default”：缺省值，使用集群默认的路由机制，Bulk请求中的每一条记录会拆分后独立路由。
- “pack”：单个Bulk请求的数据会被随机路由到同一个shard中。
- “ocal_pack”：单个Bulk请求的数据会被路由到接收该Bulk请求的数据节点的本地shard中，如果该节点不包含对应index的shard，则会进行随机路由到其他包含该索引shard节点上。该方案依赖客户端Bulk请求的随机打散和主shard的均衡分布。

Bulk 聚合优化

通过指定配置项“index.aggr_perf_batch_size”可以开启集群的Bulk聚合优化。Bulk聚合优化是通过批量导入将Bulk请求中的doc从单个依次写入变为批量写入，该方案可以有效减少内存申请、锁申请、及其他调用开销，从而提升数据导入性能。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。
2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令开启Bulk聚合优化。

```
PUT my_index
{
  "settings": {
    "index.aggr_perf_batch_size": "128"
  }
}
```

配置项“index.aggr_perf_batch_size”的取值范围为[1, Integer.MAX_VALUE]。缺省值为1，表示关闭Bulk聚合优化。当取值大于1时，表示打开Bulk聚合优化且批量取值为MIN(bulk_doc_size, aggr_perf_batch_size)。

文本索引加速

通过指定配置项“index.native_speed_up”可以开启文本索引加速。索引加速功能通过优化索引流程以及内存使用等方式实现，对于文本字段（text、keyword）能够极大提升索引构建的性能。当开启文本索引加速时，支持通过指定配置项

“index.native_analyzer”同时开启分词加速。对于需要分词的文本字段（text），当无特殊分词需求时可以开启分词加速提升分词性能。

📖 说明

- 仅当开启文本索引加速（即“index.native_speed_up”设置为“true”）时，才支持开启分词加速（即“index.native_analyzer”设置为“true”），否则分词加速不生效。
- 包含“nested”字段的索引不支持开启文本索引加速。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。

2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令开启文本索引加速。

```
PUT my_index
{
  "settings": {
    "index.native_speed_up": true,
    "index.native_analyzer": true
  }
}
```

配置项“index.native_speed_up”和“index.native_analyzer”的取值范围是true和false，缺省值是false。

索引 merge 任务优化

开启以上三种数据导入性能优化后，集群的索引merge任务会增加，通过指定配置项“index.merge.scheduler.max_thread_count”可以降低索引merge任务开销对导入性能的影响。索引merge任务优化可以增加shard的合并线程数，减少合并对数据导入的限流。

1. 在“集群管理”页面，选择可用的集群，单击操作列“Kibana”，登录Kibana界面。
2. 在Kibana左侧导航栏，选择“Dev Tools”。
3. 在“Dev Tools”页面，执行如下命令启动索引merge任务优化。

```
PUT my_index
{
  "settings": {
    "index.merge.scheduler.max_thread_count": 8
  }
}
```

配置项“index.merge.scheduler.max_thread_count”的取值范围是[1, node.processors/2]，缺省值是4，建议设置为8。

3.6 使用 OpenSearch 集群搜索数据

3.6.1 使用 DSL 语言在 OpenSearch 中搜索数据

DSL语言是Elasticsearch和OpenSearch查询域的特定语言，是客户端与Elasticsearch和OpenSearch集群交互的最佳语言。Elasticsearch DSL是基于JSON格式的语言，其他语言如SQL本质上也是先转译为Elasticsearch DSL再与Elasticsearch和OpenSearch集群交互。

DSL 使用示例

在Kibana的DevTools中编写请求内容的JSON并执行搜索请求。

例如，执行如下命令，在“test”索引中匹配所有文档。

```
GET /test/_search
{
  "query": {
    "match_all": {}
  }
}
```

查询结果返回的也是JSON格式的数据。

常用的 DSL 查询语句

下面列举了常用的DSL查询语句，全量的DSL查询语句请参见《[Elasticsearch指南](#)》。

- **设置查询条件过滤**，等同于SQL语言中的where。

如下命令中，查询未在“_search”前面过滤索引，所以是查询所有索引。bool为条件句，filter强制过滤“status”字段为“published”并且“publish_date”字段为“2015-01-01”以后的文档，must规定“title”字段含有“Search”并且“content”字段含有“Search”。

📖 说明

must和filter字段的区别在于filter等同于SQL的where会过滤字段但不参与ES搜索的打分机制，must也属于过滤必须满足的条件，但是会根据查询的匹配程度对搜索到的文档进行打分，在返回的结果中越匹配的文档会越靠前显示。

```
GET /_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "Search"
          }
        },
        {
          "match": {
            "content": "search"
          }
        }
      ],
      "filter": [
        {
          "term": {
            "status": "published"
          }
        },
        {
          "range": {
            "publish_date": {
              "gte": "2015-01-01"
            }
          }
        }
      ]
    }
  }
}
```

- **聚合查询**，近似于SQL语言中的Group by。

该查询是根据genre这个字段进行聚合，会根据test索引里的title字段进行分类统计。如果title为text（含keyword）类型，需要使用“title.keyword”进行聚合，默认情况下Elasticsearch和OpenSearch无法直接对text类型的字段进行聚合。其中titles仅为聚合的命名，可以根据需要命名为titles或者titleaggs等。

```
GET /test/_search
{
  "aggs": {
    "titles": {
      "terms": {
        "field": "title.keyword"
      }
    }
  }
}
```

以上聚合查询的例子包含了test索引的所有文档，即match_all查询条件为隐式条件。可以配合设置查询条件过滤中的查询条件过滤特定的文档进行聚合。

3.6.2 使用 SQL 语言在 OpenSearch 中搜索数据

在OpenSearch版本中提供Open Distro for Elasticsearch SQL插件允许您使用SQL而不是Elasticsearch查询域特定语言（DSL）编写查询。

熟悉SQL语言的用户，可以使用SQL语言在OpenSearch中搜索数据。

SQL 使用示例

- 在Kibana中使用SQL语言搜索数据（推荐）
在Kibana的DevTools中将请求发送到“_plugins/_sqlURI”，可以使用请求参数或请求正文。

例如，执行如下命令，从“my-index”索引中搜索出50条数据。

```
POST _plugins/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

默认情况下，查询结果返回的是JSON格式的数据。当需要返回CSV格式的数据时，则需要在命令中对format参数进行如下设置：

```
POST _plugins/_sql?format=csv
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

查询结果返回CSV格式的数据时，每行对应一个文档，每列对应一个字段。

- 在ECS中使用Curl命令执行SQL搜索数据
例如，执行如下命令，从“kibana_sample_data_flights”索引中搜索出10条数据。

```
curl -XPOST https://localhost:9200/_opendistro/_sql -u username:password -k -d '{"query": "SELECT * FROM kibana_sample_data_flights LIMIT 10"}' -H 'Content-Type: application/json'
```

localhost表示集群的访问地址，username和password分别表示安全模式集群的用户名和密码。

支持的 SQL 操作

支持的SQL操作包括声明、条件、聚合函数、Include和Exclude、常用函数、连接join和展示等操作。

- 声明statements

表 3-27 声明 statements

Statement	Example
Select	SELECT * FROM my-index
Delete	DELETE FROM my-index WHERE _id=1
Where	SELECT * FROM my-index WHERE ['field']='value'

Statement	Example
Order by	SELECT * FROM my-index ORDER BY _id asc
Group by	SELECT * FROM my-index GROUP BY range(age, 20,30,39)
Limit	SELECT * FROM my-index LIMIT 50 (default is 200)
Union	SELECT * FROM my-index1 UNION SELECT * FROM my-index2
Minus	SELECT * FROM my-index1 MINUS SELECT * FROM my-index2

📖 说明

与任何复杂查询一样，大型UNION和MINUS语句可能会使集群资源紧张甚至崩溃。

- 条件Conditions

表 3-28 条件 Conditions

Condition	Example
Like	SELECT * FROM my-index WHERE name LIKE 'j%'
And	SELECT * FROM my-index WHERE name LIKE 'j%' AND age > 21
Or	SELECT * FROM my-index WHERE name LIKE 'j%' OR age > 21
Count distinct	SELECT count(distinct age) FROM my-index
In	SELECT * FROM my-index WHERE name IN ('alejandro', 'carolina')
Not	SELECT * FROM my-index WHERE name NOT IN ('jane')
Between	SELECT * FROM my-index WHERE age BETWEEN 20 AND 30
Aliases	SELECT avg(age) AS Average_Age FROM my-index
Date	SELECT * FROM my-index WHERE birthday='1990-11-15'
Null	SELECT * FROM my-index WHERE name IS NULL

- 聚合函数Aggregation

表 3-29 聚合函数 Aggregation

Aggregation	Example
avg()	SELECT avg(age) FROM my-index
count()	SELECT count(age) FROM my-index
max()	SELECT max(age) AS Highest_Age FROM my-index
min()	SELECT min(age) AS Lowest_Age FROM my-index
sum()	SELECT sum(age) AS Age_Sum FROM my-index

- Include和Exclude字段

表 3-30 Include 和 Exclude

Pattern	Example
include()	SELECT include('a*'), exclude('age') FROM my-index
exclude()	SELECT exclude('*name') FROM my-index

- 函数Functions

表 3-31 函数 Functions

Function	Example
floor	SELECT floor(number) AS Rounded_Down FROM my-index
trim	SELECT trim(name) FROM my-index
log	SELECT log(number) FROM my-index
log10	SELECT log10(number) FROM my-index
substring	SELECT substring(name, 2,5) FROM my-index
round	SELECT round(number) FROM my-index
sqrt	SELECT sqrt(number) FROM my-index
concat_ws	SELECT concat_ws(' ', age, height) AS combined FROM my-index
/	SELECT number / 100 FROM my-index

Function	Example
%	SELECT number % 100 FROM my-index
date_format	SELECT date_format(date, 'Y') FROM my-index

📖 说明

必须在文档映射中启用fielddata才能使大多数字符串函数正常工作。

- 连接操作Joins

表 3-32 连接操作 Joins

Join	Example
Inner join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s JOIN school sc ON sc.name = s.school_name WHERE s.age > 20
Left outer join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s LEFT JOIN school sc ON sc.name = s.school_name
Cross join	SELECT s.firstname, s.lastname, s.gender, sc.name FROM student s CROSS JOIN school sc

相关约束和限制，参考[连接操作Joins](#)。

- 展示Show
展示show操作与索引模式匹配的索引和映射。您可以使用*或%使用通配符。

表 3-33 展示 show

Show	Example
Show tables like	SHOW TABLES LIKE logs-*

连接操作 Joins

Open Distro for Elasticsearch SQL支持inner joins, left outer joins,和cross joins。Join操作有许多约束：

- 您只能加入两个参数。
- 您必须为索引使用别名（例如people p）。
- 在ON子句中，您只能使用AND条件。

- 在WHERE语句中，不要将包含多个索引的树组合在一起。例如，以下语句有效：
`WHERE (a.type1 > 3 OR a.type1 < 0) AND (b.type2 > 4 OR b.type2 < -1)`
以下声明无效：
`WHERE (a.type1 > 3 OR b.type2 < 0) AND (a.type1 > 4 OR b.type2 < -1)`
- 您不能使用GROUP BY或ORDER BY来获得结果。
- LIMIT和OFFSET不支持一起使用（例如LIMIT 25 OFFSET 25）。

JDBC 驱动

Java数据库连接（JDBC）驱动程序允许您将Open Distro for Elasticsearch与您的商业智能（BI）应用程序集成。

有关下载和使用JAR文件的信息，请参阅[GitHub仓库](#)。

3.7 增强 OpenSearch 集群搜索能力

3.7.1 OpenSearch 集群搜索增强特性介绍

云搜索服务的OpenSearch集群在开源OpenSearch基础上增加了许多增强特性，增强特性的功能和支持的集群版本请参见[表3-34](#)。

表 3-34 OpenSearch 集群搜索增强特性列表

增强特性	特性描述	支持的集群版本	相关文档
存算分离	存算分离支持将集群新写入的数据存储在SSD来达到最佳的查询检索性能，将历史数据存储在OBS中降低数据的存储成本。 存算分离和切换冷热数据比，更适用于对搜索性能要求不高的场景，冷数据存储在OBS中，存储成本更低一些。	OpenSearch 1.3.6	配置 OpenSearch 集群存算分离
切换冷热数据	通过切换冷热数据，可以将部分现查要求秒级返回的数据放在高性能机器上面作为热数据，对于历史数据要求分钟级别返回的数据放在大容量低规格节点作为冷数据。冷热数据切换可以减低存储成本，提升搜索效率。 切换冷热数据和存算分离比，更适用于对搜索性能要求高的场景，冷数据存储在集群本地的冷数据节点中，存储的数据量大小依赖冷数据节点数和磁盘容量，存储成本也会比OBS高一些。	只要集群有冷数据节点就支持。	切换 OpenSearch 集群冷热数据

3.7.2 配置 OpenSearch 集群存算分离

存算分离支持将集群新写入的数据存储在SSD来达到最佳的查询检索性能，将历史数据存储在OBS中降低数据的存储成本。

场景描述

存算分离是通过冻结索引将历史数据转储到OBS，以此降低数据的存储成本。

对于有海量数据写入和存储的场景，数据有明显的冷热区分，新写入的数据一般属于热数据，存储在SSD中方便快速被搜索，随着时间的推移，历史数据不再写入，查询QPS也降低，这时候历史数据就算冷数据，通过存算分离将冷数据转储到OBS，热数据依旧存放在SSD中，方便快速搜索。

约束限制

- 仅Elasticsearch 7.6.2、Elasticsearch 7.10.2、OpenSearch 1.3.6集群支持存算分离。
- 在执行冻结索引前，系统会将待冻结的索引配置为只读状态，即便在索引数据转储到OBS后，索引也是只读状态，不支持写入数据。
- 在冻结索引过程中，查询请求不受影响。在冻结完成后，会将索引先Close再Open，在这段时间内，索引不可查询，集群可能短暂出现Red状态，索引Open结束后集群恢复正常。
- 将历史索引数据转储到OBS的过程中会占用网络带宽。
- 冻结索引后，索引会被转储到OBS中，同时本地磁盘中的索引数据会被删除。当查询被冻结的索引时，时延会增加，在聚合时，由于查询复杂、数据读取多，时延变长会更明显。
- 已转储到OBS中的索引不支持解冻，即不可回退为可写的索引。
- 由于存算分离的特性依赖OBS，所以使用过程中要遵守OBS的“带宽”和“每秒请求数（QPS）”的使用限制，详细请参见[OBS使用限制](#)。当超过限制时，集群中涉及到OBS查询的性能都会下降，例如恢复分片的速度变慢、查询数据时变慢等。

访问集群

本文通过Kibana访问集群，介绍如何使用存算分离。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，选择需要进行存算分离的集群，单击操作列“Kibana”，登录Kibana页面。
3. 选择左侧导航栏的“Dev Tools”，进入操作页面。

冻结索引

执行如下命令，将指定索引转储到OBS中。

```
POST ${index_name}/_freeze_low_cost
```

表 3-35 请求参数说明

参数名	说明
index_name	需要冻结的索引名称。

返回结果如下：

```
{
  "freeze_uuid": "pdsRgUtSTymVDWR_HoTGfW"
}
```

表 3-36 返回参数说明

参数名	说明
freeze_uuid	冻结索引任务的ID，在提交冻结请求后会启动一个异步任务，请求返回任务ID，使用该ID可以查询冻结索引任务的进度。

查看冻结索引任务

执行如下命令，查看冻结索引任务的进度。

```
GET _freeze_low_cost_progress/${freeze_uuid}
```

表 3-37 请求参数说明

参数名	说明
freeze_uuid	冻结索引任务的ID，该ID由 冻结索引 时获取。

返回结果如下：

```
{
  "stage": "STARTED",
  "shards_stats": {
    "INIT": 0,
    "FAILURE": 0,
    "DONE": 0,
    "STARTED": 3,
    "ABORTED": 0
  },
  "indices": {
    "data1": [
      {
        "uuid": "7OS-G1-tRke2jHZPlckexg",
        "index": {
          "name": "data1",
          "index_id": "4b5PHXJITLaS6AurlmfQ9A",
          "shard": 2
        },
        "start_ms": 1611972010852,
        "end_ms": -1,
        "total_time": "10.5s",
      }
    ]
  }
}
```

```
"total_time_in_millis" : 10505,
"stage" : "STARTED",
"failure" : null,
"size" : {
  "total_bytes" : 3211446689,
  "finished_bytes" : 222491269,
  "percent" : "6.0%"
},
"file" : {
  "total_files" : 271,
  "finished_files" : 12,
  "percent" : "4.0%"
},
"rate_limit" : {
  "paused_times" : 1,
  "paused_nanos" : 946460970
}
},
{
  "uuid" : "7OS-G1-tRke2jHZPlckexg",
  "index" : {
    "name" : "data1",
    "index_id" : "4b5PHXJITLaS6AurlmfQ9A",
    "shard" : 0
  },
  "start_ms" : 1611972010998,
  "end_ms" : -1,
  "total_time" : "10.3s",
  "total_time_in_millis" : 10359,
  "stage" : "STARTED",
  "failure" : null,
  "size" : {
    "total_bytes" : 3221418186,
    "finished_bytes" : 272347118,
    "percent" : "8.0%"
  },
  "file" : {
    "total_files" : 372,
    "finished_files" : 16,
    "percent" : "4.0%"
  },
  "rate_limit" : {
    "paused_times" : 5,
    "paused_nanos" : 8269016764
  }
},
{
  "uuid" : "7OS-G1-tRke2jHZPlckexg",
  "index" : {
    "name" : "data1",
    "index_id" : "4b5PHXJITLaS6AurlmfQ9A",
    "shard" : 1
  },
  "start_ms" : 1611972011021,
  "end_ms" : -1,
  "total_time" : "10.3s",
  "total_time_in_millis" : 10336,
  "stage" : "STARTED",
  "failure" : null,
  "size" : {
    "total_bytes" : 3220787498,
    "finished_bytes" : 305789614,
    "percent" : "9.0%"
  },
  "file" : {
    "total_files" : 323,
    "finished_files" : 14,
    "percent" : "4.0%"
  },
}
```

```

    "rate_limit" : {
      "paused_times" : 3,
      "paused_nanos" : 6057933087
    }
  }
}
}
}
}
}

```

表 3-38 返回参数说明

参数名	说明
stage	当前所处状态。取值包括： <ul style="list-style-type: none"> • INIT：刚启动或者正在初始化。 • FAILURE：失败。 • DONE：完成。 • STARTED：已启动。 • ABORTED：取消，预留字段。
shards_stats	处在各个状态的shard个数。
indices	每个索引的状态详情。

表 3-39 indices 返回值说明

参数名	说明
uuid	freeze的uuid。
index	索引信息和shard信息。
start_ms	开始时间。
end_ms	结束时间，如果没有结束则显示为-1。
total_time	已花费时间。
total_time_in_millis	已花费时间毫秒数。
stage	当前shard所处的状态。
failure	失败原因，如果没有失败则显示为null。
size.total_bytes	总共需要冻结的文件的字节数。
size.finished_bytes	已经完成冻结的字节数。
size.percent	已经完成冻结的字节数百分比。
file.total_bytes	总共需要冻结的文件个数。
file.finished_bytes	已经完成冻结的文件个数。
file.percent	已经完成冻结的文件个数百分比。

参数名	说明
rate_limit.paused_times	达到限速导致冻结暂停的次数。
rate_limit.paused_nanos	达到限速导致冻结暂停的时间纳秒数。

冻结完成的索引会增加以下settings，可参考[表3-40](#)。

表 3-40 冻结索引 settings

参数	说明
index.frozen_low_cost	标识该索引为冻结索引。取值为true。
index.blocks.write	冻结后的索引禁止写入。取值为true。
index.store.type	标识该索引的存储类型为obs。取值为obs。

基于冻结状态查询索引列表

执行如下命令，基于冻结状态查询索引列表。

```
GET _cat/freeze_indices?stage=${STAGE}
```

表 3-41 请求参数说明

参数名	说明
STAGE	索引的冻结状态，可选值如下： <ul style="list-style-type: none">start：开始冻结但是还未完成的索引列表done：已经完成冻结的索引列表unfreeze：没有冻结的索引列表空或者其他值：正在冻结过程中或者已经冻结完成的所有索引列表

返回结果如下：

```
green open data2 0bNtxWDtRbOSkS4JYaUgMQ 3 0 5 0 7.9kb 7.9kb
green open data3 oYMLvw31QnyasqUNuyP6RA 3 0 51 0 23.5kb 23.5kb
```

说明

此命令的参数和返回值与开源Elasticsearch的_cat/indices一致。

修改存储在 OBS 中冷数据的缓存配置

索引数据转储到OBS后，为了尽可能的减少对OBS的访问请求，并提升集群的查询性能，系统会缓存部分数据。第一次查询冷数据时，集群会直接访问OBS，获取到的数据会被缓存在集群内存中，后续查询时会先检查是否有缓存数据。

集群访问不同的文件，访问的模式是不一样的，缓存系统支持多级缓存，会使用不同大小的block来缓存不同的文件，例如fdx和tip文件会使用大量的小block缓存，对fdt文件会使用较少的大block缓存。缓存配置支持基于业务情况进行修改，配置项请参见表 3-42。

表 3-42 缓存的配置项

配置项	类型	说明
low_cost.obs.blockcache.names	Array	缓存系统支持多级缓存，分别用来缓存不同访问粒度的数据。此配置列出所有缓存的名字，即使不配置，系统也会默认有一个缓存，名字为default。如果自定义配置，请确保有一个名字为default的缓存，其他名字任意。 默认值：default。
low_cost.obs.blockcache.<NAME>.type	ENUM	缓存的类型，支持memory和file。 当使用memory类型的缓存时，会占用一定的内存大小。当使用file类型的缓存时，会使用磁盘作为缓存。建议使用超高IO型的磁盘提升缓存性能。 默认值：memory。
low_cost.obs.blockcache.<NAME>.blockshift	Integer	缓存每个block的大小，为字节左移数，即 2^x 字节。如配置为16，表示block大小为 2^{16} 字节，等于65536字节，即64K。 默认值：13（即8K）。
low_cost.obs.blockcache.<NAME>.bank.count	Integer	缓存分区数。 默认值：1。
low_cost.obs.blockcache.<NAME>.number.blocks.perbank	Integer	每个缓存分区中包含的block数。 默认值：8192。
low_cost.obs.blockcache.<NAME>.exclude.file.types	Array	不缓存的文件后缀名。如果某些后缀既不包含在exclude列表，也不包含在include列表，则会使用default缓存。
low_cost.obs.blockcache.<NAME>.file.types	Array	缓存的文件后缀名。如果某些后缀既不包含在exclude列表，也不包含在include列表，则会使用default缓存。
index.frozen.obs.max_bytes_per_sec	String	冻结过程中往OBS上传文件最大限速。动态配置，修改后立即生效。 默认值：150MB。

配置项	类型	说明
low_cost.obs.index.upload.threshold.use.multipart	String	冻结过程中文件大小超过此配置会使用OBS的分段上传。 默认值：1GB。
index.frozen.reader.cache.expire.duration.seconds	Integer	此参数设置超时时间。 为了减少冻结后的索引占用的堆内存，在索引shard启动后，reader会缓存一段时间，超时后关闭。 默认值：300s。
index.frozen.reader.cache.max.size	Integer	配置缓存最大值。 默认值：100。

下面是一个常见的缓存配置，该配置使用两级缓存，名字分别为“default”和“large”。其中，“default”缓存使用大小为64K的block，一共有30*4096个block，“default”缓存用于缓存除fdt后缀的其他文件，“large”缓存使用大小为2M的block，一共有5*1000个block，“large”缓存用于缓存fdx、dvd和tip后缀的文件。

```
low_cost.obs.blockcache.names: ["default", "large"]
low_cost.obs.blockcache.default.type: file
low_cost.obs.blockcache.default.blockshift: 16
low_cost.obs.blockcache.default.number.blocks.perbank: 4096
low_cost.obs.blockcache.default.bank.count: 30
low_cost.obs.blockcache.default.exclude.file.types: ["fdt"]

low_cost.obs.blockcache.large.type: file
low_cost.obs.blockcache.large.blockshift: 21
low_cost.obs.blockcache.large.number.blocks.perbank: 1000
low_cost.obs.blockcache.large.bank.count: 5
low_cost.obs.blockcache.large.file.types: ["fdx", "dvd", "tip"]
```

查询存储在 OBS 中冷数据的缓存状态

冻结索引之后，当冷数据第一次被查询时，系统会自动进行数据缓存，集群支持查询存储在OBS中冷数据的缓存状态。当需要调试集群性能时，也可以重置缓存状态。

1. 查询节点中存储在OBS中冷数据的缓存状态。
 - 执行如下命令，查询所有节点中冷数据的缓存信息。
GET _frozen_stats
 - 执行如下命令，查询指定节点中冷数据的缓存信息。
GET _frozen_stats/\${node_id}

表 3-43 请求参数说明

参数名	说明
node_id	节点ID。

返回结果如下：

```
{
  "_nodes" : {
    "total" : 3, // 所有节点数量
```

```
"successful" : 3, // 成功的节点数量
"failed" : 0 // 失败的节点数量
},
"cluster_name" : "css-zzz1", // 集群名称
"nodes" : {
  "7uwKO38RRoaON37YsXhCYw" : {
    "name" : "css-zzz1-ess-esn-2-1", // 节点名称
    "transport_address" : "10.0.0.247:9300", // 节点transport连接地址
    "host" : "10.0.0.247", // 节点host
    "ip" : "10.0.0.247", // 节点IP
    "block_cache" : {
      "default" : {
        "type" : "memory", // cache类型, memory表示内存cache
        "block_cache_capacity" : 8192, // cache容量
        "block_cache_blocksize" : 8192, // cache中单个block大小, 单位为bytes, 如此处表示8k
        "block_cache_size" : 12, // cache已被占用的大小
        "block_cache_hit" : 14, // cache命中数
        "block_cache_miss" : 0, // cache未命中数
        "block_cache_eviction" : 0, // cache被逐出次数
        "block_cache_store_fail" : 0 // 如果cache满了, 可能导致新的cache存储失败, 此处表示存储失败
        的次数
      }
    },
    "obs_stats" : {
      "list" : {
        "obs_list_count" : 17, // 调用OBS list接口的次数
        "obs_list_ms" : 265, // 调用OBS list接口的总耗时
        "obs_list_avg_ms" : 15 // 调用OBS list接口的平均耗时
      },
      "get_meta" : {
        "obs_get_meta_count" : 79, // 调用OBS 获取元数据接口的次数
        "obs_get_meta_ms" : 183, // 调用OBS 获取元数据接口的总耗时
        "obs_get_meta_avg_ms" : 2 // 调用OBS 获取元数据接口的平均耗时
      },
      "get_obj" : {
        "obs_get_obj_count" : 12, // 调用OBS getObject接口的次数
        "obs_get_obj_ms" : 123, // 调用OBS getObject接口的总耗时
        "obs_get_obj_avg_ms" : 10 // 调用OBS getObject接口的平均耗时
      },
      "put_obj" : {
        "obs_put_obj_count" : 12, // 调用OBS putObject接口的总次数
        "obs_put_obj_ms" : 2451, // 调用OBS putObject接口的总耗时
        "obs_put_obj_avg_ms" : 204 // 调用OBS putObject接口的平均耗时
      },
      "obs_op_total" : {
        "obs_op_total_ms" : 3022, // 调用OBS接口的总耗时
        "obs_op_total_count" : 120, // 调用OBS接口的总次数
        "obs_op_avg_ms" : 25 // 调用OBS接口的平均耗时
      }
    },
    "reader_cache" : {
      "hit_count" : 0,
      "miss_count" : 1,
      "load_success_count" : 1,
      "load_exception_count" : 0,
      "total_load_time" : 291194714,
      "eviction_count" : 0
    }
  },
  "73EDpEqoQES749umJqxOzQ" : {
    "name" : "css-zzz1-ess-esn-3-1",
    "transport_address" : "10.0.0.201:9300",
    "host" : "10.0.0.201",
    "ip" : "10.0.0.201",
    "block_cache" : {
      "default" : {
        "type" : "memory",
        "block_cache_capacity" : 8192,
        "block_cache_blocksize" : 8192,
```

```
"block_cache_size" : 12,
"block_cache_hit" : 14,
"block_cache_miss" : 0,
"block_cache_eviction" : 0,
"block_cache_store_fail" : 0
}
},
"obs_stats" : {
  "list" : {
    "obs_list_count" : 17,
    "obs_list_ms" : 309,
    "obs_list_avg_ms" : 18
  },
  "get_meta" : {
    "obs_get_meta_count" : 79,
    "obs_get_meta_ms" : 216,
    "obs_get_meta_avg_ms" : 2
  },
  "get_obj" : {
    "obs_get_obj_count" : 12,
    "obs_get_obj_ms" : 140,
    "obs_get_obj_avg_ms" : 11
  },
  "put_obj" : {
    "obs_put_obj_count" : 12,
    "obs_put_obj_ms" : 1081,
    "obs_put_obj_avg_ms" : 90
  },
  "obs_op_total" : {
    "obs_op_total_ms" : 1746,
    "obs_op_total_count" : 120,
    "obs_op_avg_ms" : 14
  }
},
"reader_cache" : {
  "hit_count" : 0,
  "miss_count" : 1,
  "load_success_count" : 1,
  "load_exception_count" : 0,
  "total_load_time" : 367179751,
  "eviction_count" : 0
}
},
"EF8WoLCUQbqJl1Pkqo9-OA" : {
  "name" : "css-zzz1-ess-esn-1-1",
  "transport_address" : "10.0.0.18:9300",
  "host" : "10.0.0.18",
  "ip" : "10.0.0.18",
  "block_cache" : {
    "default" : {
      "type" : "memory",
      "block_cache_capacity" : 8192,
      "block_cache_blocksize" : 8192,
      "block_cache_size" : 12,
      "block_cache_hit" : 14,
      "block_cache_miss" : 0,
      "block_cache_eviction" : 0,
      "block_cache_store_fail" : 0
    }
  }
},
"obs_stats" : {
  "list" : {
    "obs_list_count" : 17,
    "obs_list_ms" : 220,
    "obs_list_avg_ms" : 12
  },
  "get_meta" : {
    "obs_get_meta_count" : 79,
    "obs_get_meta_ms" : 139,
```

```
"obs_get_meta_avg_ms" : 1
},
"get_obj" : {
  "obs_get_obj_count" : 12,
  "obs_get_obj_ms" : 82,
  "obs_get_obj_avg_ms" : 6
},
"put_obj" : {
  "obs_put_obj_count" : 12,
  "obs_put_obj_ms" : 879,
  "obs_put_obj_avg_ms" : 73
},
"obs_op_total" : {
  "obs_op_total_ms" : 1320,
  "obs_op_total_count" : 120,
  "obs_op_avg_ms" : 11
}
}
},
"reader_cache" : {
  "hit_count" : 0,
  "miss_count" : 1,
  "load_success_count" : 1,
  "load_exception_count" : 0,
  "total_load_time" : 235706838,
  "eviction_count" : 0
}
}
}
```

2. 执行如下命令，可以重置缓存状态。

```
POST _frozen_stats/reset
```

说明

此命令主要用于调试性能问题，例如重置缓存状态后再次执行查询可以清晰看到本次查询的缓存命令情况。在业务运行阶段不需要执行此命令。

返回结果如下：

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "Es-0325-007_01",
  "nodes" : {
    "mqTdk2YRSPyOSXfesREFSg" : {
      "result" : "ok"
    }
  }
}
```

提升冷数据的查询性能

须知

仅2023年02月后创建的Elasticsearch 7.6.2、Elasticsearch 7.10.2和OpenSearch 1.3.6集群支持提升冷数据的查询性能。

在Kibana的Discover页面首次查询冷数据时，由于此时无任何缓存，导致所有数据均需要从OBS上获取，当命中的数据较多时，需要耗费大量的时间从OBS上获取对应的时间字段以及文件元数据。如果将这一部分数据直接缓存在本地，即可大幅提升查询性能，解决Discover页面首次查询慢的问题。云搜索服务就是通过冷数据的本地缓

存，实现冷数据的查询性能提升。本地缓存配置是预置的，用户可以基于业务场景修改配置，也可以查询、了解本地缓存信息。

1. 修改冷数据的本地缓存配置。

表 3-44 本地缓存配置项说明

配置项	类型	scope	是否可动态修改	说明
low_cost.local_cache.max.capacity	Integer	node	是	<p>节点上，能够打开的冷数据缓存的最大数量。（每个shard对应一个缓存对象）</p> <p>取值范围：10~5000</p> <p>默认值：500</p> <p>说明</p> <ul style="list-style-type: none"> 当堆内存使用率一直很高时，可以尝试降低该值。 当查询冷数据本地缓存的相关统计指标中load_overflow_count数值一直持续快速增加时，建议调大该值。
index.low_cost.local_cache.threshold	Integer	index	是	<p>启用冷数据本地缓存的阈值。</p> <ul style="list-style-type: none"> 当date类型字段的占比小于此值时，启用冷数据本地缓存date类型字段。否则不使用。 如果当前索引的date类型字段占据当前索引的绝大部分数据量，则不建议使用此功能。 <p>单位：%</p> <p>取值范围：0~100</p> <p>默认值：50</p>
index.low_cost.local_cache.evict_time	字符串	index	是	<p>冷数据的本地缓存的淘汰时间，当缓存时间大于该值，缓存将被删除。根据index.frozen_date（冻结成功的时间）判定，当索引无index.frozen_date时，则根据索引创建时间判定。</p> <p>单位：天</p> <p>取值范围：1d~365d</p> <p>默认值：30d</p> <p>说明</p> <p>建议根据磁盘用量调整淘汰时长，节省磁盘空间。</p>

- 执行如下命令，修改“low_cost.local_cache.max.capacity”。

```
PUT _cluster/settings
{
```

```
"persistent": {
  "low_cost.local_cache.max.capacity": 1000
}
```

- 执行如下命令，修改 “index.low_cost.local_cache.threshold”。

```
PUT es_write_pref2-00000000021/_settings
{
  "index.low_cost.local_cache.threshold": 20
}
```

- 执行如下命令，修改 “index.low_cost.local_cache.evict_time”。

```
PUT es_write_pref2-00000000021/_settings
{
  "index.low_cost.local_cache.evict_time": "7d"
}
```

2. 查询冷数据的本地缓存信息。

- 执行如下命令，查询所有节点中冷数据的本地缓存相关指标。

```
GET /_frozen_stats/local_cache
```

- 执行如下命令，查询指定节点中冷数据的本地缓存相关指标。

```
GET /_frozen_stats/local_cache/{nodeId}
```

“{nodeId}” 为节点ID。

返回结果如下：

```
{
  "_nodes": {
    "total": 1,
    "successful": 1,
    "failed": 0
  },
  "cluster_name": "elasticsearch",
  "nodes": {
    "6by3lPy1R3m55Dcq3lik8Q": {
      "name": "node-1",
      "transport_address": "127.0.0.1:9300",
      "host": "127.0.0.1",
      "ip": "127.0.0.1",
      "local_cache": {
        "get_stats": {
          "get_total_count": 562,           //从冷数据本地缓存查询数据的总次数
          "get_hit_count": 562,           //从冷数据本地缓存查询数据命中的次数
          "get_miss_count": 0,           //从冷数据本地缓存查询数据未命中的次数
          "get_total_ns": 43849200,      //从冷数据本地缓存查询数据的总时长
          "get_avg_ns": 78023,           //从冷数据本地缓存查询数据的平均时长
        },
        "load_stats": {
          "load_count": 2,                //加载冷数据本地缓存的次数
          "load_total_ms": 29,            //加载冷数据本地缓存的总时长
          "load_avg_ms": 14,              //加载冷数据本地缓存的平均时长
          "load_fail_count": 0,           //加载冷数据本地缓存的失败次数
          "load_overflow_count": 0,       //加载冷数据本地缓存时超过缓存池的次数
        },
        "reload_stats": {
          "reload_count": 0,              //重新生成冷数据本地缓存的次数
          "reload_total_ms": 0,           //重新生成冷数据本地缓存的总时长
          "reload_avg_ms": 0,             //重新生成冷数据本地缓存的平均时长
          "reload_fail_count": 0,         //重新生成冷数据本地缓存的失败次数
        },
        "init_stats": {
          "init_count": 0,                //初始化冷数据本地缓存的次数
          "init_total_ms": 0,             //初始化冷数据本地缓存的总时长
          "init_avg_ms": 0,               //初始化冷数据本地缓存的平均时长
          "init_fail_count": 0,           //初始化冷数据本地缓存的失败次数
        }
      }
    }
  }
}
```

查询存储冷数据的 OBS 实时速率

须知

仅2023年02月后创建的Elasticsearch 7.6.2、Elasticsearch 7.10.2和OpenSearch 1.3.6 集群支持提升冷数据的查询性能。

为了更清晰的了解到存算分离的插件在OBS中的使用情况，CSS服务新增了OBS实时速率的统计接口，并且将实时速率记录到系统索引“.freeze_obs_rate-YYYY.mm.dd”中。

OBS实时速率的计算方式：每5秒计算一次，查询前5秒内的平均OBS操作速率。

系统索引“.freeze_obs_rate-YYYY.mm.dd”用于存放OBS操作实时速率和OBS操作数据，便于了解存储冷数据的OBS的操作趋势。该索引的默认保留时间是30天。

1. 查询存储冷数据的OBS实时速率。

- 执行如下命令，查询所有节点中存储冷数据的OBS实时速率。

```
GET _frozen_stats/obs_rate
```

- 执行如下命令，查询指定节点中存储冷数据的OBS实时速率。

```
GET _frozen_stats/obs_rate/{nodeId}
```

“{nodeId}”为节点ID。

响应示例：

```
{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "elasticsearch",
  "nodes" : {
    "dfIDvcSwTJ-fkillT2zE3A" : {
      "name" : "node-1",
      "transport_address" : "127.0.0.1:9300",
      "host" : "127.0.0.1",
      "ip" : "127.0.0.1",
      "update_time" : 1671777600482,           // 当前统计值的更新时间。
      "obs_rate" : {
        "list_op_rate" : 0.0,                 // obs list操作的速率，单位：次/秒。
        "get_meta_op_rate" : 0.0,           // obs get meta操作的速率，单位：次/秒。
        "get_obj_op_rate" : 0.0,           // obs get操作的速率，单位：次/秒。
        "put_op_rate" : 0.0,               // obs put操作的速率，单位：次/秒。
        "obs_total_op_rate" : 0.0,         // obs所有操作的速率，单位：次/秒。
        "obs_upload_rate" : "0.0 MB/s",    // obs上传数据的速率，MB/秒。
        "obs_download_rate" : "0.0 MB/s"   // obs下载数据的速率，MB/秒。
      }
    }
  }
}
```

2. 修改存储OBS实时速率的索引“.freeze_obs_rate-YYYY.mm.dd”的保留时间。索引的默认保留时间是30天。

执行如下命令，将索引保留时间改成7天。

```
PUT _cluster/settings
{
  "persistent": {
    "low_cost.obs_rate_index.evict_time": "7d"
  }
}
```


表 3-45 配置项说明

配置项	类型	scope	是否可动态修改	说明
low_cost.obs_rate_index.evict_time	String	node	是	用于控制索引“.freeze_obs_rate-YYYY.mm.dd”的保留时间。 <ul style="list-style-type: none">取值范围：1d~365d。默认值：30d。单位：天。

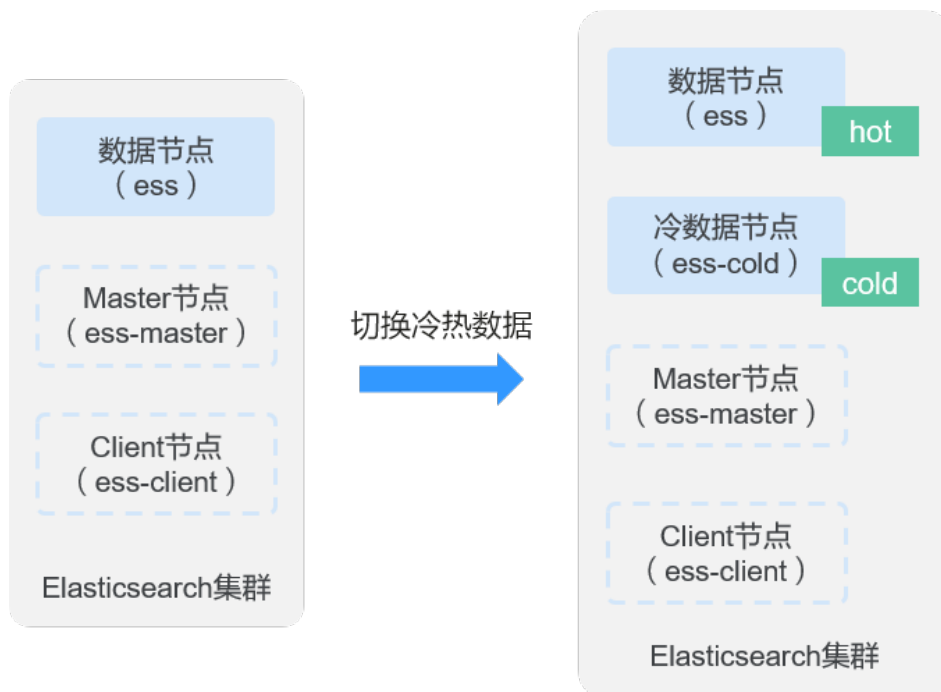
3.7.3 切换 OpenSearch 集群冷热数据

云搜索服务支持给Elasticsearch集群配置冷数据节点，支持对索引数据进行冷热切换。

场景描述

通过切换冷热数据，可以将部分现查要求秒级返回的数据放在高性能机器上面作为热数据，对于历史数据要求分钟级别返回的数据放在大容量低规格节点作为冷数据。冷热数据切换可以减低存储成本，提升搜索效率。

图 3-8 切换冷热数据原理



在创建集群时，启用的冷数据节点会被打上“cold”标签，用来存放冷数据，数据节点会被打上“hot”标签，用于存放热数据。用户可以通过配置指定索引，将数据分配到冷数据节点上。只有集群存在冷数据节点时，数据节点才会被打上热标签。

冷数据节点只有在创建集群时可以选择启用，已建集群不支持添加冷数据节点，如果集群没有冷数据节点又想切换冷热数据，可以使用存算分离功能，具体操作请参见[配置Elasticsearch集群存算分离](#)。

冷数据节点支持节点数量扩缩容和磁盘存储容量的扩容，具体操作请参见[扩缩容Elasticsearch集群](#)。

约束限制

仅启用了冷数据节点的集群才支持进行冷热数据切换。只有在创建集群时可以选择启用冷数据节点，已建集群不支持添加冷数据节点。

操作步骤

1. 登录云搜索服务管理控制台。
2. 确认集群是否已启用冷数据节点。
在“集群管理”页面，选择需要进行存算分离的集群，单击集群名称进入集群详情页，查看“节点信息”中是否有冷数据节点的信息。

图 3-9 查看冷数据节点

节点信息	
节点规格	ess.s 8 GB
节点存储	40 G
节点数量	1
冷数据节点规格	ess.s 8 GB
冷数据节点存储	40 G
冷数据节点数	3

- 是，集群存在冷数据节点，则执行下一步。
 - 否，集群不存在冷数据节点，则不支持切换集群的冷热数据。如果想将历史数据定义为冷数据，减低存储成本，提升搜索效率，可以使用存算分离功能，具体操作请参见[配置Elasticsearch集群存算分离](#)。
3. 单击集群操作列“Kibana”，登录Kibana页面。
 4. 选择左侧导航栏的“Dev Tools”，进入操作页面。
 5. 在Kibana操作页面设置template，通过设置集群模板将相应的索引数据存储到冷热数据节点上。
例如，执行如下命令设置template将以“myindex”开头的索引存储到冷数据节点上。

- Elasticsearch版本<6.x的创建模板命令

```
PUT _template/test
{
  "order": 1,
  "template": "myindex*",
  "settings": {
    "index": {
      "refresh_interval": "30s",
      "number_of_shards": "3",
      "number_of_replicas": "1",
      "routing.allocation.require.box_type": "cold"
    }
  }
}
```

- Elasticsearch版本≥6.x的创建模板命令

```
PUT _template/test
{
  "order": 1,
  "index_patterns": "myindex*",
  "settings": {
    "refresh_interval": "30s",
    "number_of_shards": "3",
    "number_of_replicas": "1",
    "routing.allocation.require.box_type": "cold"
  }
}
```

也支持直接将已有索引存储到冷热数据节点上。

例如，执行如下命令将“myindex”索引存储到冷数据节点上。

```
PUT myindex/_settings
{
  "index.routing.allocation.require.box_type": "cold"
}
```

“myindex”表示索引名称。“cold”可以基于需要改成“hot”。

6. 当需要回退冷热数据时，可以执行如下命令，删除冷热数据配置。这时候索引数据会被当成普通索引，随机均衡分布在冷热数据节点中。

```
PUT myindex/_settings
{
  "index.routing.allocation.require.box_type": null
}
```

“myindex”表示索引名称。

3.8 配置 OpenSearch 集群网络

3.8.1 配置公网访问

当集群开启了“安全模式”和“HTTPS访问”时，可以选择是否配置“公网访问”。配置公网访问后，用户可以获得一个公网访问的IP地址，通过这个IP地址可以在公网访问该安全集群。同时，还支持配置访问控制，设置允许公网访问集群的IP地址或网段。

Elasticsearch和OpenSearch集群的公网访问是通过共享型负载均衡器实现外网访问，共享型负载均衡器资源与其他实例共享资源，如果业务追求更优的负载均衡体验，推荐使用独享型负载均衡器接入集群，配置方案请参见[配置独享型负载均衡](#)。

约束限制

- CSS开启公网访问后，会使用到EIP和带宽资源，涉及相关资源费用。

- 仅Elasticsearch集群版本≥6.5.4和OpenSearch集群才支持开启“安全模式”。
- 公网访问和终端节点服务使用的是同一个负载均衡。当集群开启了公网访问白名单，由于白名单是作用在负载均衡上面，将会同时限制公网访问集群和内网通过VPCEP访问集群的IP地址。此时需要在公网访问白名单中添加一个网络白名单“198.19.128.0/17”，通过该白名单用来放通经过VPCEP的流量。

创建集群时配置公网访问

1. 登录云搜索服务管理控制台。
2. 在创建集群页面，开启“安全模式”。设置管理员密码，并启用HTTPS访问。
3. “公网访问”选择“自动绑定”，配置公网访问相关参数。

图 3-10 创建集群时配置公网访问



表 3-46 公网访问参数说明

参数	说明
带宽	设置公网访问的带宽。
访问控制开关	如果关闭访问控制开关，则允许任何IP通过公网IP访问集群。如果开启访问控制开关，则只允许白名单列表中的IP通过公网IP访问集群。
白名单	设置允许访问的IP地址或网段，中间用英文逗号隔开。仅当打开“访问控制开关”时才需要配置。

已有集群配置公网访问

您可以对已经创建集群的公网访问进行修改，查看，解绑，也可以配置公网访问。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击需要配置公网访问的集群名称，进入集群基本信息页面，管理公网访问相关配置。

图 3-11 修改公网访问相关配置

配置信息	
区域	
可用区	
虚拟私有云	vpc
子网	subnet
安全组	dw: 更改安全组
安全模式	启用
重置密码	重置
企业项目	default
公网访问	解绑
访问控制	未开启 设置
带宽	5 Mbit/s 修改
HTTPS访问	开启 下载证书
内网访问IPv4地址	

- 配置公网访问
如果创建安全集群时，开启了HTTPS访问但未配置公网访问，集群创建成功后，可以在集群基本信息页面配置公网访问。
单击“公网访问”参数右侧的“绑定”，设置访问带宽后，单击“确定”。
如果绑定失败，用户可以等待几分钟后，再次尝试重新绑定公网访问。
- 修改
对已经配置了公网访问的集群，可以通过单击“带宽”参数右侧的“修改”，修改带宽大小，也可以通过单击“访问控制”右侧的“设置”，设置访问控制开关和访问白名单。
- 查看
在“基本信息”页面，可以查看当前集群绑定的公网IP地址。
- 解绑
对于已经绑定的公网IP地址，可以通过单击“公网访问”参数右侧的“解绑”，解绑公网IP地址。

通过公网 IP 地址接入集群

公网访问配置完成后，集群将会获得一个“公网访问”的IP地址，用户可以通过公网IP地址和端口接入集群。

例如，启用HTTPS的安全集群的公网访问地址为“10.62.179.32”，端口为“9200”，则执行Curl命令查看集群中的索引信息。

```
curl -u username:password -k 'https://10.62.179.32:9200/_cat/indices'
```

其中username和password表示安全集群的用户名和密码。

3.8.2 配置终端节点服务

终端节点服务（VPC Endpoint Service）通过专属网关，可以将VPC中的服务方便的提供给其他VPC中的资源使用，实现跨VPC的访问，而不必暴露服务端相关的网络信息，使您的访问更加安全、可靠。在开启终端节点服务时，系统会默认给用户创建一个终端节点，内网域名由用户自己选择是否创建，通过节点IP或者是内网域名，用户可以跨VPC访问该集群。

终端节点服务通过共享型负载均衡器实现内网访问，共享型负载均衡器资源与其他实例共享资源，如果业务追求更优的负载均衡体验，推荐使用独享型负载均衡器接入集群，配置方案请参见[配置独享型负载均衡](#)。

约束限制

- 集群开启终端节点服务之后，终端节点将按需进行收费，终端节点的费用将由用户进行支付，详细的计费方式请参考[终端节点计费说明](#)。
- 当集群的网络配置中，“虚拟私有云”选择的是共享VPC，“子网”选择的是共享VPC下的子网，则该集群不支持配置终端节点服务。
- 创建终端节点需要有相关的权限，请参考[VPC终端节点权限管理](#)。
- 公网访问和终端节点服务使用的是同一个负载均衡。当集群开启了公网访问白名单，由于白名单是作用在负载均衡上面，将会同时限制公网访问集群和内网通过VPC访问集群的IP地址。此时需要在公网访问白名单中添加一个网络白名单“198.19.128.0/17”，通过该白名单用来放通经过VPC的流量。

创建集群时开启终端节点服务

1. 登录云搜索服务管理控制台。
2. 在右上方单击“创建集群”。
3. 在创建集群页面，“高级配置”选择“自定义”后，开启终端节点服务。

图 3-12 开启终端节点服务

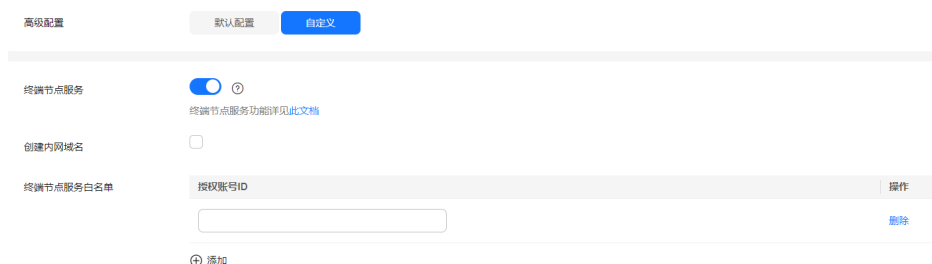


表 3-47 配置终端节点服务

参数	说明
创建内网域名	勾选“创建内网域名”，系统将会自动为用户创建一个内网域名，通过这个域名可以在同一个VPC内访问该集群。
终端节点服务白名单	<p>在“终端节点服务白名单”中添加允许通过节点IP或内网域名访问集群的账号。</p> <ul style="list-style-type: none">单击“添加”输入授权账号ID。授权账号ID配置成“*”，则表示允许全部用户访问该集群。单击操作列的“删除”，可以删除不允许访问的账号ID。 <p>说明 “授权账号ID”可以在“我的凭证”中进行查看“账号ID”获取。</p>

已有集群终端节点服务管理

如果创建集群时未开启终端节点服务，集群创建成功后，可以通过如下步骤进行开启。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击需要开启终端节点服务的集群名称，进入集群基本信息页面。
3. 选择“终端节点服务”，在“终端节点服务”右侧单击开关，打开集群的终端节点服务功能。

表 3-48 配置终端节点服务

参数	说明
创建内网域名	勾选“创建内网域名”，系统将会自动为用户创建一个内网域名，通过这个域名可以在同一个VPC内访问该集群。
终端节点服务白名单	<p>在“终端节点服务白名单”中添加允许通过节点IP或内网域名访问集群的账号。</p> <ul style="list-style-type: none">单击“添加”输入授权账号ID。授权账号ID配置成“*”，则表示允许全部用户访问该集群。单击操作列的“删除”，可以删除不允许访问的账号ID。 <p>说明 “授权账号ID”可以在“我的凭证”中进行查看“账号ID”获取。</p>

4. 管理终端节点。
在终端节点服务页面下，显示所有连接当前终端节点服务的终端节点。

图 3-13 管理终端节点

终端节点ID	状态	最大连接数	拥有者	创建时间	操作
a24dc31c	已接受	3000		2024/03/15 16:14:20 GMT+08:00	接受 拒绝

单击操作列的“接受”或者“拒绝”可以修改节点的“状态”。如果对某个终端节点“拒绝”操作之后，其生成的内网域名将不能再访问到当前集群。

通过节点 IP 或内网域名访问集群

1. 获取集群的内网域名或者节点IP。

登录云搜索服务控制台，进入集群列表，单击集群名称，进入集群“基本信息”页面，选择“终端节点服务”，查看“节点IP”和“内网域名”。

图 3-14 查看节点 IP 和内网域名信息

基本信息	
终端节点服务名称	cn
节点IP	192
内网域名	vpcep- .com
终端节点服务白名单	-- 修改

2. 在弹性云服务器中，直接通过Curl命令调用Elasticsearch API即可使用集群。

弹性云服务器需要满足如下要求：

- 为弹性云服务分配足够的磁盘空间。
- 此弹性云服务器的VPC需要与集群在同一个VPC中，开通终端节点服务后，可以实现跨VPC访问。
- 此弹性云服务器的安全组需要和集群的安全组相同。
如果不同，请修改弹性云服务器安全组或配置弹性云服务器安全组的出入规则允许集群所有安全组的访问。修改操作请参见[配置安全组规则](#)。
- 待接入的CSS集群，其安全组的出方向和入方向需允许TCP协议及9200端口，或者允许端口范围包含9200端口。

例如，执行如下Curl命令，查看集群中的索引信息，集群中的内网访问地址为“vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com”，端口为“9200”。

- 如果接入集群未启用安全模式，接入方式为：

```
curl 'http://vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com:9200/_cat/indices'
```
- 如果接入集群已启用安全模式，则需要使用https方式访问，并附加用户名和密码，在curl命令中添加-u选项。

```
curl -u username:password -k 'https://vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com:9200/_cat/indices'
```


3.8.3 配置独享型负载均衡

云搜索服务集成了共享型负载均衡器的能力，支持绑定公网访问和开启终端节点服务。相比共享型负载均衡器，独享型负载均衡器功能更丰富、性能更强，本章主要介绍如何配置集群的独享型负载均衡。

场景描述

使用独享型负载均衡器接入集群具有如下优点：

- 非安全模式集群也支持集成弹性负载均衡服务的能力。
- 支持用户使用自定义证书进行HTTPS双向认证。
- 支持7层流量监控及告警配置，方便用户随时查看监控情况。

不同安全模式的集群对接独享型负载均衡器共有8种不同的服务形态，不同服务形态对应的ELB能力如表3-49所示，8种组合的配置概览如表3-50所示。

表 3-49 不同集群对应的 ELB 能力

集群安全模式	ELB最终对外提供的服务形态	ELB负载均衡	ELB流量监控	ELB双向认证
非安全	无认证	支持	支持	不支持
	单向认证	支持	支持	支持
	双向认证			
安全+HTTP	密码认证	支持	支持	不支持
	单向认证+密码认证	支持	支持	支持
	双向认证+密码认证			
安全+HTTPS	单向认证+密码认证 双向认证+密码认证	支持	支持	支持

表 3-50 不同集群对接 ELB 的配置概览

集群安全模式	ELB最终对外提供的服务形态	ELB监听器			后端服务器组		
		前端协议	前端端口	SSL解析方式	后端协议	健康检查端口	健康检查路径
非安全	无认证	HTTP	9200	无认证	HTTP	9200	/
	单向认证	HTTPS	9200	单向认证	HTTP	9200	
	双向认证	HTTPS	9200	双向认证	HTTP	9200	

安全+HT TP	密码认证	HTTP	9200	无认证	HTTP	9200	/_opendis tro/ _security /health
	单向认证+密 码认证	HTTPS	9200	单向认 证	HTTP	9200	
	双向认证+密 码认证	HTTPS	9200	双向认 证	HTTP	9200	
安全+HT TPS	单向认证+密 码认证	HTTPS	9200	单向认 证	HTTPS	9200	
	双向认证+密 码认证	HTTPS	9200	双向认 证	HTTPS	9200	

CSS集群对接独享型负载均衡的操作步骤如下：

1. 如果规划ELB监听器的前端协议是HTTPS，则需要准备签名证书并上传至ELB管理控制台：[准备并上传自签名证书](#)
2. 在ELB管理控制台创建独享型负载均衡器：[创建独享型负载均衡器](#)
3. Elasticsearch或OpenSearch集群开启负载均衡：[集群对接负载均衡器](#)
4. 通过独享型负载均衡实例接入集群：[通过Curl命令接入集群](#)

本文还提供了通过独享型负载均衡实例接入集群的Java客户端代码示例：[通过负载均衡器的HTTPS双向认证方式接入集群的代码示例（Java）](#)

约束限制

- 如果ELB绑定了公网，则不推荐接入非安全模式的集群。非安全模式的集群使用HTTP通信且不需要安全认证即可访问，如果对接公网访问存在安全风险。
- 开启了HTTPS访问的安全模式的集群不支持HTTP类型的前端协议认证，如果需要使用HTTP类型的前端协议，需要将集群的“安全模式”变更为“非安全模式”，具体操作请参见[更改Elasticsearch集群安全模式](#)。在变更安全模式前，请先关闭“负载均衡”，安全模式变更完成后再开启“负载均衡”。

准备并上传自签名证书

当规划的ELB监听器的前端协议为HTTPS时，需要参考本节准备自签名证书，作为服务器证书或CA证书上传至ELB。

说明

建议使用云证书管理服务CCM购买的证书，或者其他权威机构颁发的证书。

1. 登录到任意一台安装有OpenSSL工具和JDK的Linux客户端。
2. 执行如下命令制作自签名证书。

```
mkdir ca
mkdir server
mkdir client

#使用OpenSSL制作CA证书
cd ca
#创建CA证书的openssl配置文件ca_cert.conf
cat >ca_cert.conf <<EOF
[ req ]
```

```
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
EOF
#创建CA证书私钥文件ca.key
openssl genrsa -out ca.key 2048
#创建CA证书的csr请求文件ca.csr
openssl req -out ca.csr -key ca.key -new -config ./ca_cert.conf
#创建自签名的CA证书ca.crt
openssl x509 -req -in ca.csr -out ca.crt -sha1 -days 5000 -signkey ca.key
#把ca证书格式转为p12格式
openssl pkcs12 -export -clcerts -in ca.crt -inkey ca.key -out ca.p12
#把ca证书格式转为jks格式
keytool -importkeystore -srckeystore ca.p12 -srcstoretype PKCS12 -deststoretype JKS -destkeystore
ca.jks

#使用CA证书签发服务器证书
cd ../server
#创建服务器证书的openssl配置文件server_cert.conf，CN字段根据需求改为服务器对应的域名、IP地址
cat >server_cert.conf <<EOF
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
CN = 127.0.0.1
EOF
#创建服务器证书私钥文件server.key
openssl genrsa -out server.key 2048
#创建服务器证书的csr请求文件server.csr
openssl req -out server.csr -key server.key -new -config ./server_cert.conf
#使用CA证书签发服务器证书server.crt
openssl x509 -req -in server.csr -out server.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -
CAkey ../ca/ca.key
#把服务端证书格式转为p12格式
openssl pkcs12 -export -clcerts -in server.crt -inkey server.key -out server.p12
#把服务证书格式转为jks格式
keytool -importkeystore -srckeystore server.p12 -srcstoretype PKCS12 -deststoretype JKS -destkeystore
server.jks

#使用CA证书签发客户端证书
cd ../client
#创建客户端证书的openssl配置文件client_cert.conf，CN字段根据需求改为服务器对应的域名、IP地址
cat >client_cert.conf <<EOF
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
O = ELB
CN = 127.0.0.1
EOF
#创建客户端证书私钥文件client.key
openssl genrsa -out client.key 2048
#创建客户端证书的csr请求文件client.csr
openssl req -out client.csr -key client.key -new -config ./client_cert.conf
#使用CA证书签发客户端证书client.crt
openssl x509 -req -in client.csr -out client.crt -sha1 -CAcreateserial -days 5000 -CA ../ca/ca.crt -
CAkey ../ca/ca.key
#把客户端证书格式转为浏览器可识别的p12格式
openssl pkcs12 -export -clcerts -in client.crt -inkey client.key -out client.p12
#把客户端证书格式转为jks格式
keytool -importkeystore -srckeystore client.p12 -srcstoretype PKCS12 -deststoretype JKS -destkeystore
client.jks
```

3. 上传自签名证书，具体操作请参见[创建证书](#)。

创建独享型负载均衡器

1. 登录弹性负载均衡管理控制台。
2. 参考[创建独享型负载均衡器](#)，创建独享型负载均衡器。CSS集群对接独享型负载均衡器所需要关注的参数如表3-51所示，其他参数请根据实际需要填写。

表 3-51 独享型负载均衡器的配置说明

参数	配置说明	取值样例
实例类型	选择“独享型”。	独享型
计费模式	性能独享型负载均衡器的收费类型。	按需计费
区域	选择CSS集群所在的区域。	-
跨VPC后端	开启跨VPC后端才能连接CSS集群。	开启
网络类型	负载均衡器对外提供服务所使用的网络类型。	IPv4私网
所属VPC	所属虚拟私有云。无论选择哪种网络类型，均需配置此项。 需要选择和CSS集群同一VPC。	-
子网	选择创建负载均衡实例的子网。无论选择哪种网络类型，均需配置此项。 需要选择和CSS集群同一子网。	-
规格	建议选择功能和性能更优的应用型规格。	应用型(HTTP/HTTPS) “小型1”

集群对接负载均衡器

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要对接负载均衡器的集群，单击集群名称进入集群“基本信息”页面。
3. 在左侧菜单栏，选择“负载均衡”，打开负载均衡开关，在弹窗中配置负载均衡基础信息。

表 3-52 配置负载均衡

参数	说明
负载均衡器	选择前面创建的独享型负载均衡器。CSS集群属于托管资源，所选的负载均衡器必须开启“跨VPC后端”能力才能正常使用。
委托	授权CSS服务访问和使用ELB资源。所选委托必须包含策略“ELB Administrator”或“ELB FullAccess”。

图 3-15 开启负载均衡

基础配置

负载均衡器 [创建负载均衡器](#) [?](#)

委托 [创建委托](#) [?](#)

[取消](#) [确定](#)


- 单击“确定”，开启负载均衡。
- 在“监听器配置”区域，您可以单击右侧  配置监听器的相关信息。

表 3-53 配置监听器

参数	配置说明
前端协议	客户端与负载均衡监听器建立流量分发连接的协议。支持选择“HTTP”或“HTTPS”。 根据实际业务需要选择协议。
前端端口	客户端与负载均衡监听器建立流量分发连接的端口。 根据实际业务需要自定义。
SSL解析方式	客户端到服务器端认证方式。仅“前端协议”选择“HTTPS”才需要配置。 根据实际业务需要选择解析方式。
服务器证书	服务器证书用于SSL握手协商，需提供证书内容和私钥。仅“前端协议”选择“HTTPS”才需要配置服务器证书。 选择 准备并上传自签名证书 创建的服务器证书。
CA证书	CA证书又称客户端CA公钥证书，用于验证客户端证书的签发者。仅“SSL解析方式”选择“双向认证”时，才需要配置CA证书。 选择 准备并上传自签名证书 创建的CA证书。 在开启HTTPS双向认证功能时，只有当客户端能够出具指定CA签发的证书时，HTTPS连接才能成功。

图 3-16 配置监听器

监听器配置

前端协议	HTTP	HTTPS
前端端口	<input type="text"/>	
SSL解析方式	单向认证	双向认证
服务器证书	server-new	<input type="button" value="Q"/>
CA证书	ca-new	<input type="button" value="Q"/>

- （可选）在监听器配置区域，单击“访问控制”后的“设置”跳转到负载均衡器的监听器列表，单击监听器访问控制列的“设置”，配置允许通过负载均衡实例访问集群的IP地址组，不设置的话默认允许所有的IP地址访问。
- 在健康检查区域，您可以查看各个节点IP的健康检查结果。

表 3-54 健康检查结果状态说明

健康检查结果	说明
正常	节点IP连接正常。
异常	节点IP连接，不可用

通过 Curl 命令接入集群

- 在云搜索服务管理控制台，单击左侧导航栏的“集群管理”。
- 在集群管理列表页面，单击需要访问的集群名称，进入集群基本信息页面。
- 在左侧菜单栏选择“负载均衡”，获取并记录负载均衡实例的“私有IP”或“公网IP”。
- 在ECS服务器中执行如下Curl命令测试能否通过独享型负载均衡实例接入集群。

表 3-55 不同集群的接入命令

集群安全模式	ELB最终对外提供的服务形态	接入集群的Curl命令
非安全	无认证	curl http://IP9200
	单向认证	curl -k --cert ./client.crt --key ./client.key https://IP9200

集群安全模式	ELB最终对外提供的服务形态	接入集群的Curl命令
	双向认证	<code>curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:9200</code>
安全+HTTP	密码认证	<code>curl http://IP:9200 -u user:pwd</code>
	单向认证+密码认证	<code>curl -k --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>
	双向认证+密码认证	<code>curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>
安全+HTTPS	单向认证+密码认证	<code>curl -k --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>
	双向认证+密码认证	<code>curl --cacert ./ca.crt --cert ./client.crt --key ./client.key https://IP:9200 -u user:pwd</code>

表 3-56 变量说明

变量名	说明
IP	负载均衡实例的“私有IP”或“公网IP”。
user	集群的用户名，仅安全集群需要配置。
pwd	用户名对应的密码，仅安全集群需要配置。

当可以正常返回集群信息时，表示连接成功。

通过负载均衡器的 HTTPS 双向认证方式接入集群的代码示例（Java）

在Java客户端，通过负载均衡器的HTTPS双向认证方式接入集群的代码示例。

ESSecuredClientWithCerDemo 代码示例

```
import org.apache.commons.io.IOUtils;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.HttpHost;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.elasticsearch.action.search.SearchRequest;
import org.elasticsearch.action.search.SearchResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.index.query.QueryBuilders;
import org.elasticsearch.search.SearchHit;
import org.elasticsearch.search.SearchHits;
import org.elasticsearch.search.builder.SearchSourceBuilder;
import java.io.FileInputStream;
```

```
import java.io.IOException;
import java.security.KeyStore;
import java.security.SecureRandom;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManagerFactory;
public class ESSecuredClientWithCerDemo {
    private static final String KEY_STORE_PWD = "";
    private static final String TRUST_KEY_STORE_PWD = "";
    private static final String CA_JKS_PATH = "ca.jks";
    private static final String CLIENT_JKS_PATH = "client.jks";
    private static final String ELB_ADDRESS = "127.0.0.1";
    private static final int ELB_PORT = 9200;
    private static final String CSS_USERNAME = "user";
    private static final String CSS_PWD = "";
    public static void main(String[] args) {
        // 建立客户端
        RestHighLevelClient client = initESClient(ELB_ADDRESS, CSS_USERNAME, CSS_PWD);
        try {
            // 查询 match_all, 相当于 {"query": {"match_all": {}}}
            SearchRequest searchRequest = new SearchRequest();
            SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
            searchSourceBuilder.query(QueryBuilders.matchAllQuery());
            searchRequest.source(searchSourceBuilder);
            // query
            SearchResponse searchResponse = client.search(searchRequest, RequestOptions.DEFAULT);
            System.out.println("query result: " + searchResponse.toString());
            SearchHits hits = searchResponse.getHits();
            for (SearchHit hit : hits) {
                System.out.println(hit.getSourceAsString());
            }
            System.out.println("query success");
            Thread.sleep(2000L);
        } catch (InterruptedException | IOException e) {
            e.printStackTrace();
        } finally {
            IOUtils.closeQuietly(client);
        }
    }
    private static RestHighLevelClient initESClient(String clusterAddress, String userName, String password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new UsernamePasswordCredentials(userName,
password));
        SSLContext ctx = null;
        try {
            KeyStore ks = getKeyStore(CLIENT_JKS_PATH, KEY_STORE_PWD, "JKS");
            KeyManagerFactory kmf = KeyManagerFactory.getInstance("SunX509");
            kmf.init(ks, KEY_STORE_PWD.toCharArray());
            KeyStore tks = getKeyStore(CA_JKS_PATH, TRUST_KEY_STORE_PWD, "JKS");
            TrustManagerFactory tmf = TrustManagerFactory.getInstance("SunX509");
            tmf.init(tks);
            ctx = SSLContext.getInstance("SSL", "SunJSSE");
            ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(), new SecureRandom());
        } catch (Exception e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(ctx, new HostnameVerifier() {
            @Override
            public boolean verify(String arg0, SSLSession arg1) {
                return true;
            }
        });
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,
credentialsProvider);
        RestClient.Builder builder = RestClient.builder(new HttpHost(clusterAddress, ELB_PORT, "https"))
.setHttpClientConfigCallback(httpClientConfigCallback);
```



```
RestHighLevelClient client = new RestHighLevelClient(builder);
return client;
}
private static KeyStore getKeyStore(String path, String pwd, String type) {
    KeyStore keyStore = null;
    FileInputStream is = null;
    try {
        is = new FileInputStream(path);
        keyStore = KeyStore.getInstance(type);
        keyStore.load(is, pwd.toCharArray());
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        IOUtils.closeQuietly(is);
    }
    return keyStore;
}
}
```

SecuredHttpClientConfigCallback 代码示例

```
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOStrategy;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;
import java.util.Objects;
class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;
    /**
     * The {@link SSLIOStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOStrategy sslStrategy;
    /**
     * Create a new {@link SecuredHttpClientConfigCallback}.
     *
     * @param credentialsProvider The credential provider, if a username/password have been supplied
     * @param sslStrategy The SSL strategy, if SSL / TLS have been supplied
     * @throws NullPointerException if {@code sslStrategy} is {@code null}
     */
    SecuredHttpClientConfigCallback(final SSLIOStrategy sslStrategy,
        @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }
    /**
     * Get the {@link CredentialsProvider} that will be added to the HTTP client.
     *
     * @return Can be {@code null}.
     */
    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }
    /**
     * Get the {@link SSLIOStrategy} that will be added to the HTTP client.
     *
     * @return Never {@code null}.
     */
    SSLIOStrategy getSSLStrategy() {
        return sslStrategy;
    }
    /**
     * Sets the {@linkplain HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider)
    credential provider},
     *
     * @param httpClientBuilder The client to configure.
     * @return Always {@code httpClientBuilder}.
     */
}
```

```
*/
@Override
public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder httpClientBuilder) {
    // enable SSL / TLS
    httpClientBuilder.setSSLStrategy(sslStrategy);
    // enable user authentication
    if (credentialsProvider != null) {
        httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
    }
    return httpClientBuilder;
}
}
```

pom.xml 代码示例

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>1</groupId>
    <artifactId>ESClient</artifactId>
    <version>1.0-SNAPSHOT</version>
    <name>ESClient</name>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <elasticsearch.version>7.10.2</elasticsearch.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>transport</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch</groupId>
            <artifactId>elasticsearch</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>elasticsearch-rest-high-level-client</artifactId>
            <version>${elasticsearch.version}</version>
        </dependency>
        <dependency>
            <groupId>commons-io</groupId>
            <artifactId>commons-io</artifactId>
            <version>2.11.0</version>
        </dependency>
    </dependencies>
</project>
```

3.9 备份与恢复 OpenSearch 集群数据

3.9.1 创建快照备份 OpenSearch 集群数据

本文介绍CSS集群快照备份的两种方式：自动创建快照和手动创建快照。

约束限制

- 备份与恢复索引功能上线之前（即2018年3月10日之前）创建的集群，无法创建快照。
- 第一次备份时，建议将所有索引数据进行备份。
- 集群快照会导致CPU、磁盘IO上升等影响，建议在业务低峰期进行操作。
- 修改存储快照的OBS桶之后，集群将会从新的OBS桶中获取快照数据。
- 当集群处于“不可用”状态时，快照功能中除了恢复快照功能外，其他快照信息或功能只能查看，无法进行编辑。
- 备份与恢复过程中，支持集群扩容（如果是恢复至本集群，则本集群不支持扩容）、访问Kibana、查看监控、删除其他快照的操作。不支持重启此集群、删除此集群、删除正在创建或恢复的快照、再次创建或恢复快照的操作。当此集群正在进行创建快照或者恢复快照过程中，同时的自动创建快照任务将被取消。
- CSS集群第一次快照是全量，后面再备份快照是在之前的快照基础上增量，CSS是增量快照逻辑，快照之间的文件会相互依赖。

前提条件

- 已创建用于存储快照的OBS桶。快照存储在OBS桶中需要额外收费，具体收费请参见[OBS计费说明](#)。
- 确认账号具有创建或恢复快照功能的权限。可以在IAM服务的“权限管理 > 权限”查看是否有以下权限，如果没有请联系服务管理员添加。
 - “CSS Administrator”角色。
 - “OBS Administrator”策略

设置自动创建快照


1. 登录云搜索服务管理控制台。
2. 开启集群自动创建快照。支持在创建集群的时候就打开集群自动创建快照，操作指导请参见[7](#)，或者在集群创建完成之后再开启集群自动创建快照。
 - 如果在创建集群时已经打开集群自动创建快照，请执行[7](#)。
 - 如果是集群创建完成之后开启集群自动创建快照，请执行下一步。
3. 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
4. 在“集群快照”管理页面，单击“集群快照开关”右侧开关，打开集群快照功能。如果已打开集群快照功能，则跳过该步骤。
打开集群快照功能后，CSS服务会自动为客户创建OBS桶和IAM委托，用于存储快照。自动创建的OBS桶和IAM委托将直接展示在界面中。如果您不希望使用自动创建的OBS桶和IAM委托，您可以在“基础配置”右侧单击进行配置。

表 3-57 集群快照基础配置

参数	说明
OBS桶	在下拉框中选择存储快照的OBS桶。也可以单击右侧的“创建桶”新建OBS。详细操作步骤请参见 创建桶 。 创建或者已存在的OBS桶需满足如下条件： <ul style="list-style-type: none"> “存储类别”为“标准存储”。 “区域”须与创建的集群所在区域相同。
备份路径	快照在OBS桶中的存放路径。 备份路径配置规则： <ul style="list-style-type: none"> 备份路径不能使用符号“\:*?\"<> ”。 备份路径不能以“/”开头。 备份路径不能以“.”开头或结尾。 备份路径的总长度不能超过1023个字符。
IAM委托	指当前账号授权云搜索服务访问或维护存储在OBS中数据。如果没有合适的委托可以联系CSS服务管理员新建IAM委托。详细操作步骤请参见 如何创建委托 。 所选的IAM委托需满足如下条件： <ul style="list-style-type: none"> “委托类型”选择“云服务”。 “云服务”选择“Elasticsearch”或者“云搜索服务 CSS”。 必选策略：“OBS Administrator”


- 在“自动创建快照”右侧，单击开关开启自动创建快照功能，弹出“创建快照策略”页面。如果已启用自动创建快照功能，也可以在开关右侧单击进行快照策略修改。

表 3-58 设置自动创建快照

参数	说明
快照名称前缀	快照名称前缀的长度为1~32个字符，只能包含小写字母、数字、中划线和下划线，且必须以小写字母开头。快照名称由快照名称前缀加上时间戳组成，例如自动生成的快照名称为“snapshot-1566921603720”。
时区	指备份时间对应的时区，不支持修改。基于此时区选择备份开始时间。
备份开始时间	指每天自动开始备份的时间，只能指定整点时间，如00:00、01:00，取值范围为00:00~23:00。请在下拉框中选择时间。

参数	说明
保留快照个数	<p>自定义设置自动快照保留的个数，范围是1~90。系统在半点时刻会自动删除超过保留个数的快照（过期删除策略只针对与当前自动创建快照策略相同执行频次的自动快照）。</p> <p>说明 保存快照个数与自动创建快照策略中设置的执行频次和索引有关，当执行频次间隔时间短或索引数量最大，保留自动快照可能会达不到设置的个数，请谨慎选择。</p>
索引	<p>填写索引名称，支持选择索引进行备份。索引名称不能包含空格和大写字母，且不能包含“\< >/?”特殊字符，多个索引之间使用英文逗号隔开。如果不填写，则默认备份集群中所有索引。支持使用“*”匹配多个索引，例如“index*”，表示备份名称前缀是index的所有索引的数据。</p> <p>说明 在Kibana中使用GET /_cat/indices命令，可以查询集群中所有索引的名称。</p>

图 3-17 自动创建快照

创建快照策略

快照名称前缀

时区 GMT+08:00

备份开始时间

保留快照个数

索引

如果您确定要设置当前选择的自动备份策略，请输入CONFIRM

- 确定要设置当前选择的自动备份策略，请在输入框输入“CONFIRM”，设置完成后，单击“确定”保存快照策略。
- 按照策略自动创建的快照将呈现在快照管理列表中。快照列表同时展示自动创建和手动创建的快照，您可以通过快照类型参数进行区分。在快照列表右上角，您可以输入快照名称或快照ID的关键字进行查找。

📖 说明

如果快照创建失败，可以在快照列表的右侧单击查看“失败任务数”，了解失败原因。失败任务数最多显示20条，当快照功能关闭或集群删除时，失败记录也会被清空。

8. （可选）关闭自动创建快照功能。

关闭自动创建快照功能后，系统将停止继续自动创建快照。如果系统正在根据策略自动创建快照，而快照列表还未呈现正在创建的快照时，无法关闭自动创建快照功能。如果您单击了关闭按钮，系统将提示您无法关闭。建议等快照自动创建成功后，即快照列表已出现最新创建的快照时，再单击关闭按钮，关闭自动创建快照功能。

关闭自动创建快照功能时，您可以在弹出窗口中通过“删除自动创建的快照”选项，选择是否立即删除之前已自动创建的快照，默认不勾选。

- 不勾选：表示不会删除关闭此功能前已自动创建的快照。如果不删除，后续还可以在快照列表中通过删除按钮手动删除，详细操作指导请参见[删除 Elasticsearch 集群快照](#)。如果未手动删除，且之后用户又重新开启了自动创建快照功能，那么此集群中所有“快照类型”为自动创建的快照（包含开启自动创建快照功能前已存在的自动创建的快照）都无法手动删除，只会被系统自动删除。系统会基于重新开启自动创建快照功能时的配置策略进行自动删除，例如此策略中定义的保留快照个数为10个，那么当快照个数超过10个，系统在半点时刻会自动删除超过保留个数的快照。
- 勾选：表示删除此集群快照列表中所有“快照类型”为自动创建的快照。

📖 说明

快照功能关闭时，不会主动清理快照数据。如果需要清理快照数据，可前往OBS控制台存储快照的OBS桶对快照数据进行主动管理。

手动创建快照

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
3. 在“集群快照”管理页面，单击“集群快照开关”右侧开关，打开集群快照功能。如果已打开集群快照功能，则跳过该步骤。


打开集群快照功能后，CSS服务会自动为客户创建OBS桶和IAM委托，用于存储快照。自动创建的OBS桶和IAM委托将直接展示在界面中。如果您不希望使用自动创建的OBS桶和IAM委托，您可以在“基础配置”右侧单击进行配置。

表 3-59 集群快照基础配置

参数	说明
OBS桶	在下拉框中选择存储快照的OBS桶。也可以单击右侧的“创建桶”新建OBS。详细操作步骤请参见 创建桶 。 创建或者已存在的OBS桶需满足如下条件： <ul style="list-style-type: none">• “存储类别”为“标准存储”。• “区域”须与创建的集群所在区域相同。

参数	说明
备份路径	快照在OBS桶中的存放路径。 备份路径配置规则： <ul style="list-style-type: none">• 备份路径不能使用符号“\:*?*<> ”。• 备份路径不能以“/”开头。• 备份路径不能以“.”开头或结尾。• 备份路径的总长度不能超过1023个字符。
IAM委托	指当前账号授权云搜索服务访问或维护存储在OBS中数据。如果没有合适的委托可以联系CSS服务管理员新建IAM委托。详细操作步骤请参见 如何创建委托 。 所选的IAM委托需满足如下条件： <ul style="list-style-type: none">• “委托类型”选择“云服务”。• “云服务”选择“Elasticsearch”或者“云搜索服务 CSS”。• 必选策略：“OBS Administrator”

4. 完成基础配置后，单击下方“创建快照”可手动创建快照。

表 3-60 创建快照配置

参数	说明
快照名称	手动创建的快照名称，4~64个字符，只能包含小写字母、数字、中划线和下划线，且必须以字母开头。与自动创建不同，手动创建的快照名称按照用户设置的名称，不会自动加上时间信息。
索引	填写索引名称，支持选择索引进行备份。索引名称不能包含空格和大写字母，且不能包含“\< >/?”特殊字符，多个索引之间使用英文逗号隔开。如果不填写，则默认备份集群中所有索引。支持使用“*”匹配多个索引，例如“index*”，表示备份名称前缀是index的所有索引的数据。 说明 在Kibana中使用GET /_cat/indices命令，可以查询集群中所有索引的名称。
快照描述	创建的快照描述信息。0~256个字符，不能包含“<>”字符。

图 3-18 手动创建快照

创建快照

快照名称

索引 0/1,024

快照描述 0/256

5. 单击“确定”开始创建快照。

快照创建完成后，将直接呈现在快照管理列表中，快照状态为“可用”表示快照创建成功。快照列表同时展示自动创建和手动创建的快照，您可以通过快照类型参数进行区分。在快照列表右上角，您可以输入快照名称或快照ID的关键字进行查找。

说明

如果快照创建失败，可以在快照列表的右侧单击查看“失败任务数”，了解失败原因。失败任务数最多显示20条，当快照功能关闭或集群删除时，失败记录也会被清空。

3.9.2 恢复 OpenSearch 集群数据

将已有的快照，通过恢复快照功能，将备份的集群快照恢复到本集群或其他满足要求的集群中。

约束限制

- 备份与恢复索引功能上线之前（即2018年3月10日之前）创建的集群，无法创建快照。
- 集群快照会导致CPU、磁盘IO上升等影响，建议在业务低峰期进行操作。
- 当集群处于“不可用”状态时，快照功能中，除了恢复快照功能外，其他快照信息或功能只能查看，无法进行编辑。
- 备份与恢复过程中，支持集群扩容（如果是恢复至本集群，则本集群不支持扩容）、访问Kibana、查看监控、删除其他快照的操作。不支持重启此集群、删除此集群、删除正在创建或恢复的快照、再次创建或恢复快照的操作。当此集群正在进行创建快照或者恢复快照过程中，同时的自动创建快照任务将被取消。
- 快照在恢复过程中不可查询集群数据。
- 当两个集群的shard不一样时，则无法恢复集群数据。
- 用于恢复的目标集群版本不得低于源集群版本，一般保持版本一致即可。

前提条件

- 快照管理列表中有“快照状态”为“可用”的快照。快照存储在OBS桶中需要额外收费，具体收费请参见[OBS计费说明](#)。
- 确认账号具有创建或恢复快照功能的权限。可以在IAM服务的“权限管理 > 权限”查看是否有以下权限，如果没有请联系服务管理员添加。
 - “CSS Administrator”角色。
 - “OBS Administrator”策略

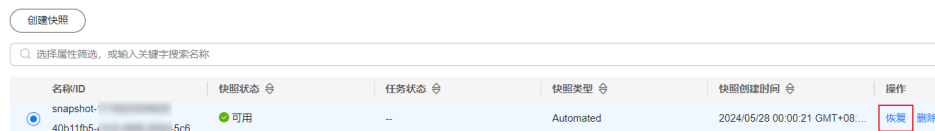
恢复数据

须知

恢复数据将覆盖集群中当前的数据，请谨慎操作。

- 登录云搜索服务管理控制台。
- 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
- 在快照管理列表中，选择需要恢复的快照，单击“操作”列的“恢复”。

图 3-19 选择恢复快照



- 在“恢复”页面配置集群的恢复参数。

表 3-61 恢复配置

参数	说明
索引	指定需要进行恢复的索引名称，默认为空。如保持默认值，即不指定索引名称，则表示恢复所有的索引数据。支持0~1024个字符，不能包含空格和大写字母，且不能包含“\< >/?”特殊字符。支持使用“*”匹配多个索引，比如index*，表示恢复快照中名称前缀是index的所有索引。
索引名称匹配模式	在恢复时，可以根据文本框中定义的过滤条件去恢复符合条件的索引，过滤条件请使用正则表达式。默认值“index_(.)+”表示所有的索引。支持0~1024个字符，不能包含空格和大写字母，且不能包含“\< >/?”特殊字符。 说明 “索引名称匹配模式”和“索引名称替换模式”需要同时设置才会生效。

参数	说明
索引名称替换模式	<p>索引重命名的规则。默认值“restored_index_\$1”表示在所有恢复的索引名称前面加上“restored_”。支持0~1024个字符，不能包含空格和大写字母，且不能包含“\< >/?,”特殊字符。</p> <p>说明 “索引名称匹配模式”和“索引名称替换模式”需要同时设置才会生效。</p>
集群	<p>选择需要进行恢复的集群名称，可选择当前集群或者其他集群。</p> <ul style="list-style-type: none">只能选择处于“可用”状态的集群，如果快照所属的集群处于“不可用”状态，那么也无法将快照恢复到本集群。恢复到其他集群时，目标集群的版本不低于本集群的版本。选择“是否覆盖目标集群同名同shard结构索引”，建议不覆盖，即不勾选。通过快照恢复数据，是以快照文件覆盖的形式进行数据恢复，覆盖目标集群同名的索引后，可能导致目标集群同名索引数据丢失，请谨慎勾选操作。

- 单击“确定”开始恢复。恢复成功，快照列表中“任务状态”将变更为“恢复成功”，索引数据将根据快照信息重新生成。

3.9.3 删除 OpenSearch 集群快照

当快照信息不需要使用时，您可以删除快照释放存储资源。

当自动创建快照功能开启时，自动创建的快照无法手动删除，系统会按照设置的策略在半点时刻自动删除超过“保留时间”的快照。

当自动创建快照功能开启又关闭了，关闭时，之前已自动创建的快照并未同步删除，这时快照列表中自动创建的快照可以手动删除。如果未手动删除，且之后用户又重新开启了自动创建快照功能，那么此集群中所有“快照类型”为自动创建的快照（包含开启自动创建快照功能前已存在的自动创建的快照）都无法手动删除，只会被系统自动删除。

说明

快照信息删除后，数据将无法恢复，请谨慎操作。

手动删除快照

- 登录云搜索服务管理控制台。
- 在“集群管理”页面，单击需要进行备份的集群名称，进入集群基本信息页面。在左侧导航栏选择“集群快照”，进入“集群快照”管理页面。
或者，在“集群管理”页面，单击对应集群“操作”列的“更多>备份与恢复”，进入“集群快照”管理页面。
- 在快照管理列表中，选择需要删除的快照。

- 单击“操作”列的“删除”，在弹窗中确认要删除的快照信息后，单击“确定”删除快照。

3.10 扩缩容 OpenSearch 集群

3.10.1 扩容 OpenSearch 集群

当Elasticsearch或OpenSearch集群的数据面业务变化，需要动态调整集群节点的数量、容量、规格和类型时，可以参考本文扩容集群。

场景描述

CSS服务支持多种扩容场景，如表3-62所示。

表 3-62 扩容场景

场景	描述	相关文档
增加节点数量和容量	<p>只支持基于节点类型扩容集群的“节点数量”和“节点存储容量”。适用于如下场景：</p> <ul style="list-style-type: none">当集群数据节点（ess）的写入与查询压力大、响应时间过长时，可以通过扩容数据节点的“节点存储容量”保证数据的持久性。如果因数据量过大或操作不当导致数据节点状态异常时，可以扩容“节点数量”保证集群的可用性。冷数据节点（ess-cold）主要用于分担数据节点（ess）的压力，当发现冷数据有丢失的风险时，可以扩容冷数据节点的“节点存储容量”保证冷数据的持久性，同时也支持扩容节点个数保证集群的可用性。	扩容节点数量和节点存储容量

场景	描述	相关文档
升级节点规格	<p>只支持基于节点类型变更集群的“节点规格”。一般用于小规格变更为大规格，常用于如下场景：</p> <ul style="list-style-type: none">当新增索引或分片分配的处理时间过长，或管理集群各个节点的协调、调度不足时，可以升级Master节点（ess-master）的“节点规格”保证集群的正常使用。当数据节点任务分发量、结果汇聚量过大时，可以升级Client节点（ess-client）的“节点规格”。当数据的写入与查询突然变得缓慢时，可以升级数据节点（ess）的“节点规格”提高数据节点的查询与写入效率。当存在冷数据查询缓慢时，可以升级冷数据节点（ess-cold）的“节点规格”提高对数据查询的效率。	变更节点规格
增加节点类型	<p>当集群本身没有Master节点（ess-master）和Client节点（ess-client）时，随着数据面业务增长，需要动态调整集群形态时，可以选择增加“Master节点”或“Client节点”。</p>	添加Master或Client节点

约束限制

- 集群的节点存储容量只支持扩容不支持缩容，请基于业务量合理选择节点存储容量。
- 包周期集群不支持同时扩容“节点数量”和“节点存储容量”。
- 在集群数据量比较大的情况下，变更节点规格耗时比较长，建议在业务低峰期更改节点规格，利于更快完成规格更改。
- 变更节点规格也支持将大规格更改为小规格，但是此变更会降低集群的处理性能，进而影响业务能力，请谨慎使用。
- 当集群包含多种节点类型时，一次只支持变更一种类型的节点规格，且变更完成后只生效所选类型的节点规格。
- 变更规格过程中，Kibana不可用。
- 变更规格过程中，会依次对节点进行关机，完成更改后再依次开机，是一个滚动的变更过程。在这过程中，Kibana不可用。
- 当一个集群包含的节点类型不同时，各节点类型支持的节点数量不同，扩容节点数量时请参考[表3-63](#)。

表 3-63 集群的节点数量取值说明

一个集群包含的节点类型	节点数量的取值范围
ess	ess: 1~32
ess、ess-master	ess: 1~200 ess-master: 3~9的奇数
ess、ess-client	ess: 1~32 ess-client: 1~32
ess、ess-cold	ess: 1~32 ess-cold: 1~32
ess、ess-master、ess-client	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32
ess、ess-master、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-cold: 1~32
ess、ess-client、ess-cold	ess: 1~32 ess-client: 1~32 ess-cold: 1~32
ess、ess-master、ess-client、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32 ess-cold: 1~32
说明	<ul style="list-style-type: none">● ess: 数据节点，即创建集群时必配的节点类型，其他3种节点类型都是基于业务需要可选的类型。● ess-master: Master节点。● ess-client: Client节点。● ess-cold: 冷数据节点。

前提条件

- 确认集群处于“可用”状态，且无正在进行的任务。
- 确认CSS服务有足够的配额支持集群扩容。

扩容节点数量和节点存储容量

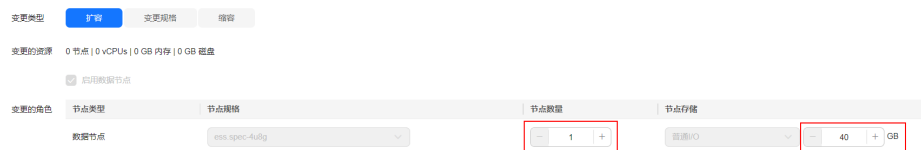
1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。

4. 选择“更改集群规格”页签。
5. 在更改集群规格页面，设置扩容参数。

表 3-64 扩容节点数量和节点存储容量

参数	说明
变更类型	选择“扩容”。
变更的资源	显示本次扩容的资源变化量。
变更的角色	<p>在“节点数量”列和“节点存储”列增加节点数量和节点存储容量。支持多种节点类型同时变更。</p> <ul style="list-style-type: none"> 节点数量的取值范围请参见表3-63。 节点存储容量的取值范围由“节点规格”决定，且只支持配置为20的倍数。 <p>说明 包周期集群不支持同时扩容“节点数量”和“节点存储容量”。</p>

图 3-20 集群扩容



6. 单击“下一步”。
7. 确认变更信息后，单击“提交申请”。
8. 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列显示为“扩容”，表示集群正在扩容。当集群状态变为“可用”，则表示扩容成功。

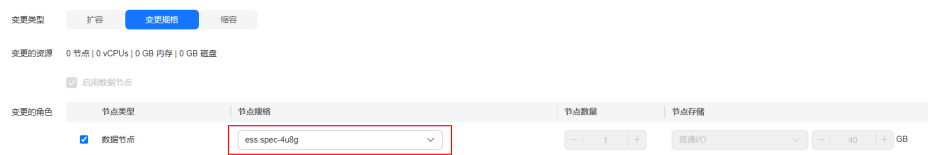
变更节点规格

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 确认业务数据是否都有副本，确保变更规格过程中不会中断业务。
 - a. 在集群列表选择目标集群，单击操作列的Kibana，登录Kibana。
 - b. 选择左侧导航栏的“Dev Tools”，进入操作页面。
 - c. 在Kibana中执行命令`GET _cat/indices?v`。
 - 当返回的“rep”值大于“0”时，则表示有副本，请继续执行下一步。
 - 当返回的“rep”值等于“0”时，则表示没有副本，建议先为集群[手动创建快照](#)再变更规格。
4. 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
5. 选择“更改集群规格”页签。
6. 在更改集群规格页面，设置变更规格的参数。

表 3-65 变更节点规格

参数	说明
变更类型	选择“变更规格”。
变更的资源	显示本次变更规格的资源变化量。
变更的角色	先在“节点类型”列选择本次要变更的节点类型，只能选择一种类型。再在“节点规格”列选择新规格。

图 3-21 集群变更规格



- 单击“下一步”。
- 确认变更信息后，单击“提交申请”。
- 在弹出的窗口确认是否勾选“进行索引副本校验”和“检测集群状态”，单击“确认”启动集群规格变更。
 - 索引副本校验：索引副本校验用于保证索引正常提供服务。如果跳过校验，变更规格操作将不会要求每一个索引都有副本，变更过程中可能会影响业务。
 - 没有Master节点的集群更改节点规格时，如果选择进行索引副本校验，则要求所有索引至少有1个副本，且“节点数量”总和不小于3。
 - 有Master节点的集群更改节点规格时，如果选择进行索引副本校验，则要求所有索引至少有1个副本。
 - 检测集群状态：默认检测集群状态，变更规格为滚动变更，变更过程中，为了保证变更成功率以及数据安全，会确保每个节点进程拉起来后继续后续节点操作。当集群负载过高业务故障，无法正常下发变更请求，依赖更多的资源才能恢复的紧急情况下可忽略检测集群状态，忽略后，变更过程中因为忽略集群状态检测可能会导致集群故障并中断业务，请谨慎跳过。
- 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“规格修改”，表示集群正在更改规格。当集群状态变为“可用”，则表示规格变更成功。

添加 Master 或 Client 节点

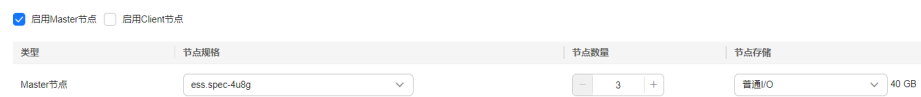
- 登录云搜索服务管理控制台。
- 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
- 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
- 选择“添加Master/Client节点”页签。

当集群同时拥有Master节点和Client节点时，形态变更页面将不存在“添加Master/Client节点”页签。
- 在添加Master/Client节点页面，勾选需要添加的节点，并配置“节点规格”、“节点数量”和“节点存储”。

表 3-66 添加 Master 或 Client 节点

参数	说明
启用节点类型	勾选本次要添加的节点类型。 <ul style="list-style-type: none">只能选择一种节点类型，当需要同时添加Master和Client节点时，就需要执行2次“添加Master/Client节点”任务。当集群本身已经有Master节点或Client节点时，则此处只显示另一种节点类型供选择。
节点规格	根据需求选择节点规格。
节点数量	设置该节点类型的节点数量，取值范围请参见表 3-63。
节点存储	设置该节点类型的存储类型和容量，节点存储容量的取值范围由“节点规格”决定，且只支持配置为20的倍数。

图 3-22 添加 Master/Client 节点



- 单击“下一步”。
- 确认变更信息后，单击“提交申请”。
返回集群列表页面，集群的“任务状态”列中显示为“扩容中”。
 - 如果添加Master节点，当“集群状态”变为“可用”时表示节点添加成功。

须知

如果集群版本小于7.x，当“集群状态”变为“可用”时，还需要重启集群的所有数据节点和冷数据节点，使新增节点的配置生效。如果不重启使新增节点配置生效，使用集群时因为上报机制问题可能显示不可用状态（集群业务并未不可用）。重启操作请参见[重启Elasticsearch集群](#)。

- 如果添加Client节点，当“集群状态”变为“可用”时表示节点添加成功。可选择是否重启数据节点和冷数据节点，下线数据节点上的Cerebro和Kibana的进程。

3.10.2 缩容 OpenSearch 集群的节点数量

当集群有充足的能力处理当前数据时，为节省资源可以参考本文缩容集群的节点数量。

场景描述

CSS服务支持多种缩容场景，如表2-116所示。

表 3-67 缩容场景

场景	描述	相关文档
缩容随机节点	缩容随机节点是基于节点类型随机下线节点减少集群的“节点数量”，减少集群占用的资源。	缩容随机节点
缩容指定节点	缩容指定节点是通过选择下线具体哪个节点减少集群的“节点数量”，减少集群占用的资源。	缩容指定节点

约束限制

- 缩容是通过下线节点减少节点数量，为了降低业务影响，建议在业务低峰期进行缩容操作。
- 缩容过程会涉及数据迁移，将要下线的节点数据迁移到其他节点上，数据迁移的超时阈值为5小时。当超过5小时数据还未迁移完成，那么缩容会失败。建议在集群数据量较大的情况下，分多次进行缩容。
- 要确保缩容之后的磁盘使用量小于80%，且集群每个节点类型中每个AZ的节点数至少为1。
- 关于跨AZ的集群，在不同AZ中同类型节点个数的差值要小于等于1。
- 当集群没有Master节点时，每次缩容的数据节点和冷数据节点个数之和要小于缩容前数据节点和冷数据节点个数之和的一半，缩容后的数据节点和冷数据节点个数之和要大于索引的最大副本个数。
- 当集群有Master节点时，每次缩容的Master节点个数要小于当前Master节点总数的一半，缩容后的Master节点个数必须是奇数且不小于3。
- 当集群没有Master节点时，数据节点数量和冷数据节点数量之和大于等于3时才支持缩容。当集群有Master节点时，数据节点数量大于等于2时才支持缩容。当集群不满足缩容条件却想减少集群节点时，建议新建集群，[备份与恢复OpenSearch 集群数据](#)迁移集群数据。
- 当一个集群包含的节点类型不同时，各节点类型支持的节点数量不同，缩容节点数量时请参考[表3-68](#)。

表 3-68 集群的节点数量取值说明

一个集群包含的节点类型	节点数量的取值范围
ess	ess: 1~32
ess、ess-master	ess: 1~200 ess-master: 3~9的奇数
ess、ess-client	ess: 1~32 ess-client: 1~32
ess、ess-cold	ess: 1~32 ess-cold: 1~32

一个集群包含的节点类型	节点数量的取值范围
ess、ess-master、ess-client	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32
ess、ess-master、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-cold: 1~32
ess、ess-client、ess-cold	ess: 1~32 ess-client: 1~32 ess-cold: 1~32
ess、ess-master、ess-client、ess-cold	ess: 1~200 ess-master: 3~9的奇数 ess-client: 1~32 ess-cold: 1~32
说明 <ul style="list-style-type: none">• ess: 数据节点，即创建集群时必配的节点类型，其他3种节点类型都是基于业务需要可选的类型。• ess-master: Master节点。• ess-client: Client节点。• ess-cold: 冷数据节点。	

前提条件

集群处于“可用”状态，且无正在进行的任务。

缩容随机节点

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
4. 选择“更改集群规格”页签。
5. 在更改集群规格页面，设置缩容参数。

表 3-69 缩容随机节点

参数	说明
变更类型	选择“缩容”。
变更的资源	显示本次缩容的资源变化量。

参数	说明
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平面时才显示该参数。</p> <ul style="list-style-type: none">删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。
变更的角色	<p>在“节点数量”列减少节点数量。支持多种节点类型同时变更。</p> <p>节点数量的取值范围请参见表3-68。</p>

图 3-23 集群缩容



- 单击“下一步”。
- 确认变更信息后，单击“提交申请”。
- 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“缩容中”，表示集群正在缩容。当集群状态变为“可用”，则表示缩容成功。

缩容指定节点

- 登录云搜索服务管理控制台。
- 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
- 选择目标集群，单击操作列的“更多>形态变更”进入形态变更页面。
- 选择“缩容指定节点”页签。
- 在缩容指定节点页面，设置缩容参数。

表 3-70 缩容指定节点

参数	说明
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平面时才显示该参数。</p> <ul style="list-style-type: none">删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。
节点类型	展开本次要变更的节点类型，此处会显示当前节点类型拥有的节点，勾选要下线的节点。

- 单击“下一步”。
- 确认变更信息后，单击“提交申请”，在数据迁移弹窗中确认是否进行数据迁移（建议勾选“进行数据迁移”避免数据丢失），单击“确认”提交申请。

数据迁移过程中，系统会把待下线节点中的所有数据分片移动到剩余节点，待数据迁移完成后再进行节点下线操作。当待下线节点上的数据在剩余节点存在副本时，可跳过数据迁移，缩短集群变更时间，减少数据迁移带来的负载压力。
- 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“缩容中”，表示集群正在缩容。当集群状态变为“可用”，则表示缩容成功。

3.11 管理 OpenSearch 集群

3.11.1 查看 OpenSearch 集群信息

在Elasticsearch/OpenSearch集群的列表页和基本信息页，可以获取集群的状态、版本、节点等信息。

集群列表介绍

集群列表会展示当前账号下同一集群类型的所有集群。当集群数量较多时，会翻页显示，支持查看任何状态下的集群。

集群列表默认按时间顺序排列，时间最近的集群显示在最前端。

在集群列表页单击“导出”可以下载全量的集群列表信息。

表 3-71 集群列表说明


参数	描述
名称/ID	展示集群的名称和ID。单击集群名称可进入集群“基本信息”页面，展现了集群的基本信息。集群ID是系统自动生成的，是集群在服务中的唯一标示。
集群状态	展示集群的运行状态。 <ul style="list-style-type: none">● 可用：正常运行的集群。● 创建中：正在创建中的集群。● 处理中：重启中、扩容中、备份中或恢复中的集群。● 异常：创建失败或不可用的集群。 说明 当集群处于“不可用”状态时，支持删除集群，也支持将集群正常状态时创建的快照恢复至其他集群。但是无法扩容集群、访问Kibana、创建快照或将快照恢复至此集群。建议不要执行导入数据的操作，避免数据丢失。您可以查看监控或重启集群处理异常，当依旧执行失败时，请及时联系技术支持。
任务状态	展示重启集群、扩容集群、缩容集群等任务的状态。
版本	展示集群版本号和镜像版本号。 说明 当集群的创建时间比较早时，数据库可能未记录其镜像信息，导致页面无法显示集群的镜像版本号。
创建时间	展示集群的创建时间。
企业项目	展示集群所归属的企业项目。
内网访问地址	展示集群的内网访问地址和端口号，您可以使用此参数接入集群。集群有多个节点时，此处显示多个节点的内网访问地址和端口号。
计费模式	展示集群的计费模式，是“按需计费”还是“包年/包月”。
操作	展示集群可执行的操作入口，包含监控信息、重启、删除等其他更多操作。当某一操作无法执行时，按钮将置灰。

查看集群基本信息

在集群的基本信息页面，可以获取集群的内网访问地址、公网访问地址、版本、节点等信息。

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群管理页面。
3. 在集群列表，单击集群名称进入集群“基本信息”页面，查看集群的基本信息。

表 3-72 基本信息的参数说明

类别	参数	描述
基本信息	集群名称	集群名称。支持自定义名称。 单击右侧  可以修改集群名称。
	ID	集群的唯一标识，是系统自动生成的。 同一个区域下，集群ID是唯一的。
	集群版本	集群的版本信息。 如果需要升级集群版本，请参见 升级Elasticsearch集群版本 。
	集群状态	集群当前的状态。
	任务状态	集群当前的任务状态，如果没有进行中的任务则显示“--”。
	创建时间	集群创建的时间。
	集群存储容量 (GB)	集群设置的存储容量。
	集群存储使用量 (GB)	集群已使用的存储容量。
配置信息	区域	集群所在区域。
	可用区	集群所在的可用区。
	虚拟私有云	集群所属的虚拟私有云。
	子网	集群所属的子网。
	安全组	集群所属的安全组。 单击右侧的“更改安全组”可以修改集群的安全组信息。 须知 进行安全组切换前，请确保业务接入时需要的9200端口已经放开，错误的安全组配置可能会导致业务无法访问，请谨慎操作。
	安全模式	集群的安全模式。 <ul style="list-style-type: none">• 启用：表示当前集群是安全模式的集群。• 未开启：表示当前集群是非安全集群。 如果需要切换集群的安全模式，请参见 更改OpenSearch集群安全模式 。

类别	参数	描述
	重置密码	<p>仅安全模式的集群显示。</p> <p>单击“重置”可以修改安全集群的管理员账户admin的密码。</p> <p>说明 管理员密码的规则：</p> <ul style="list-style-type: none"> 可输入的字符串长度为8~32个字符。 密码至少包含大写字母、小写字母、数字和特殊字符四类中的三类。其中支持的特殊字符有“~!@#%&*()-_+=+ []{};:<.>/?”。 不能与管理员账户名或倒序的管理员账户名相同。 建议定期修改密码。
	企业项目	<p>集群所属的企业项目。</p> <p>单击项目名称可以跳转到项目管理页面查看企业项目的基本信息。</p>
	公网访问	<p>公网访问信息，仅安全模式的集群显示。</p> <ul style="list-style-type: none"> 启用公网访问的安全集群，此处显示公网访问地址，通过该地址可以在公网访问安全集群。 未启用公网访问的安全集群，此处显示“--”。 <p>使用公网地址访问集群时，建议开启访问控制并配置访问白名单，提高集群安全性。公网访问配置的详细信息，请参见配置公网访问。</p>
	访问控制	<p>启用公网访问的集群是否设置访问控制，仅启用公网访问的集群显示。</p> <ul style="list-style-type: none"> 已开启：启用访问控制开关，只允许白名单列表中的IP地址通过公网访问集群。 未开启：为启用访问控制开关，允许任何IP地址通过公网访问集群。 <p>单击“设置”，可以更新访问控制开关和白名单。</p>
	带宽	<p>公网访问的带宽，仅启用公网访问的集群显示。</p> <p>单击“修改”，可以更新带宽大小。</p>

类别	参数	描述
	HTTPS访问	集群是否启用HTTPS访问协议。 <ul style="list-style-type: none">关闭：表示集群未启用HTTPS访问，集群使用HTTP访问协议。开启：表示集群启用了HTTPS访问协议，仅安全模式的集群支持开启HTTPS访问。启用HTTPS访问的安全集群可以单击“下载证书”获取CER安全证书，用于接入安全模式的集群。安全证书暂不支持在公网环境下使用。 如果安全模式的集群需要切换访问方式，请参见 切换安全模式下的协议 。
	内网访问IPv4地址	集群的内网IP地址和端口号，使用此参数可以接入集群。如果集群只有一个节点，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9200”；如果集群有多个节点，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9200,10.62.179.33:9200”。
节点信息	节点规格	集群中节点的规格信息。
	节点存储	集群中节点的存储容量和存储类型。
	节点数量	集群中节点的个数。

3.11.2 使用 OpenSearch Dashboard 创建用户并授权

CSS服务会对开启安全模式的集群进行访问控制，只有通过身份认证的用户才能访问安全集群。在创建安全集群时，必须配置一个管理员账号。该管理员能够使用OpenSearch Dashboard为集群添加新用户，并授予相应的安全集群使用权限。

背景信息

云搜索服务用opendistro_security安全插件对外提供安全集群能力，opendistro_security安全插件是基于RBAC（Role-Based Access Control）模型构建。RBAC包括三个重要核心概念：用户（User）、权限（Action）、角色（Role）。RBAC简化了用户和权限的关系，降低了权限管理的难度，方便权限扩展易于维护。三者之前的关系如下图所示。

图 3-24 用户、权限和角色



表 3-73 OpenSearch Dashboard 创建用户和授权

参数	描述
Permission	权限：单个动作，例如创建索引（例如indices:admin/create）。
Action group	操作组：表示一组权限。例如，预定义的SEARCH操作组授权角色使用_search和_msearchAPI。
Role	角色：定义为权限或操作组的组合，包括对集群，索引，文档或字段的操作权限。
User	用户：可以向Elasticsearch集群发出操作请求。用户具有凭证（例如，用户名和密码）、零个或多个后端角色以及零个或多个自定义属性。
Role mapping	角色映射：用户在成功进行身份验证后会担任角色，角色映射，就是将角色映射到用户（或后端角色）。例如，kibana_user（角色）到jdoe（用户）的映射意味着John Doe在获得kibana_user身份验证后获得了所有权限。同样，all_access（角色）到admin（后端角色）的映射意味着具有后端角色admin（来自LDAP / Active Directory服务器）的任何用户都获得了all_access身份验证后的所有权限。您可以将每个角色映射到许多用户和/或后端角色。

在OpenSearch Dashboard界面的Security菜单中，您可以控制用户在OpenSearch集群中的权限，实现集群、索引、文档和字段四个级别的细粒度访问权限控制。

支持添加或删除集群的用户，并能够将用户映射到角色，实现用户关联角色权限。

角色映射功能允许配置角色的成员，通过用户名、后端角色和主机名将用户分配给相应的角色。支持为每种角色配置集群访问权限、索引和文档的访问权限，以及OpenSearch Dashboard的使用权限。

有关安全集群的更多安全配置信息以及详细的操作指导，可以参考Elasticsearch的[安全模式官方介绍](#)。

约束限制

OpenSearch Dashboard中可以自定义用户名、角色名、租户名等，但不能包含中文字符。

操作步骤

步骤1 登录OpenSearch Dashboard。

1. 登录云搜索服务控制台。
2. 在左侧导航栏，选择“集群管理 > OpenSearch”，进入集群列表页面。
3. 在集群管理列表，选择对应集群，单击操作列的“Kibana”。
4. 输入管理员账户名和密码登录OpenSearch Dashboard。
 - 账户名：admin（默认管理员账户名）
 - 密码：创建安全模式的集群时，设置的管理员密码。

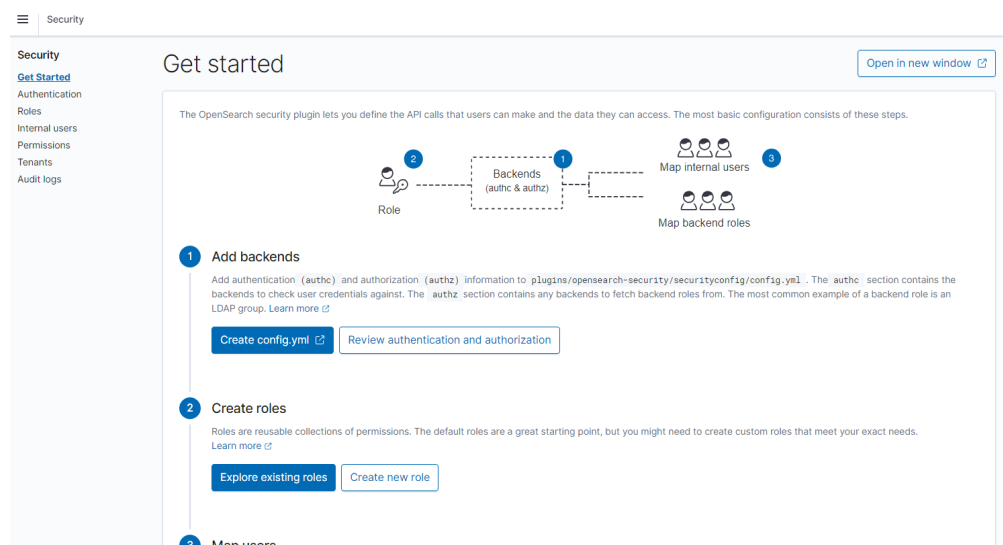
图 3-25 登录 OpenSearch



步骤2 创建用户（User）。

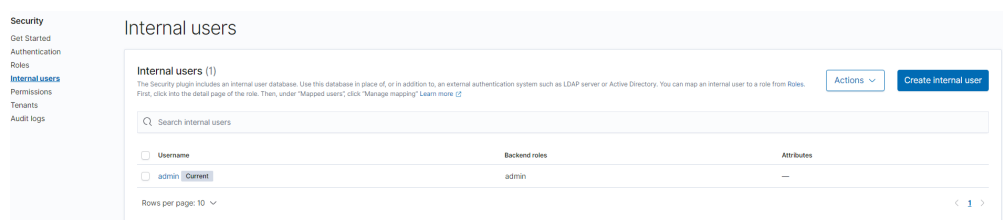
1. 在OpenSearch Dashboard操作界面，选择“Security”，进入Security页面。

图 3-26 进入 Security 界面



2. 在左侧选择“Internal users”，进入用户创建页面。

图 3-27 创建用户



3. 单击“Create internal user”，进入用户信息配置页面。
4. 在“Credentials”区域，输入需要创建的用户名和密码。

图 3-28 输入用户名和用户密码

5. 单击“Create”，用户创建成功后，可以在列表中看到新创建的用户。

图 3-29 用户信息

步骤3 创建角色Role，并授予Role相应权限。

1. 在“Security”中选择“Roles”。
2. 在“Roles”页面，单击“Create role”，进入角色创建页面。
3. 在“Name”区域设置角色名。

图 3-30 设置角色名称

4. 在“Cluster Permissions”页面设置集群权限。根据业务需要选择相应的集群权限，不配置时表示角色不具有任何集群级别的权限。

图 3-31 设置集群权限

Cluster permissions

Specify how users in this role can access the cluster. By default, no cluster permission is granted. [Learn more](#)

Cluster Permissions

Specify permissions using either action groups or single permissions. An action group is a list of single permissions. You can often

✕ ▼ [Create new permission group](#)

5. 在“Index Permissions”区域设置索引权限。

图 3-32 设置索引权限

Index permissions

Index permissions allow you to specify how users in this role can access the specific indices. By default, no index permission is granted. [Learn more](#)

▼ my_store

Index

✕

Specify index pattern using *

Index permissions

You can specify permissions using both action groups or single permissions. A permission group is a list of single permissions. You can often achieve your

✕ ▼ [Create new permission group](#)

6. 在“Tenant Permissions”页面设置角色权限。

图 3-33 角色权限

Tenant permissions

Tenants are useful for safely sharing your work with other OpenSearch Dashboards users. You can control which roles have access to a tenant and

Tenant

✕ ▼ Read and Write ▼ Remove

[Add another tenant permission](#)

设置完成后，即可在“Roles”页面看到创建的角色。

步骤4 角色用户映射，将Role和User绑定。

1. 在“Security”中选择“Roles”。
2. 在“Roles”页面，选择你需要映射的角色名称，进入角色映射页面。

Security / Roles / Role1

Role1 Actions Edit role

Permissions Mapped users

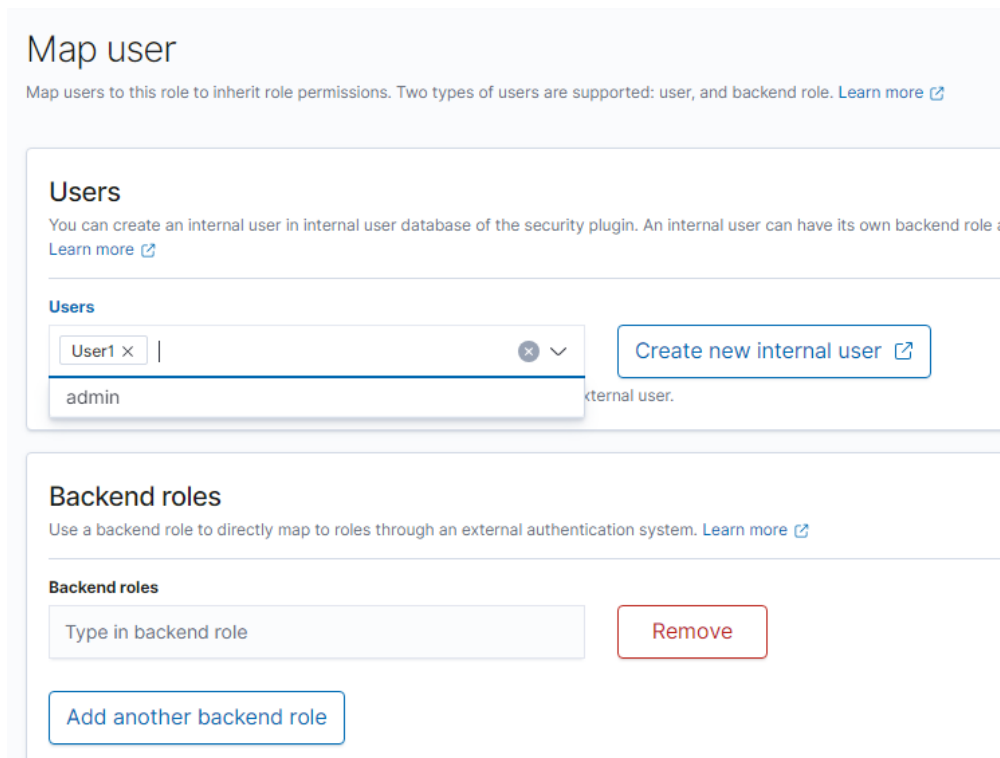
Mapped users (0) Delete mapping Manage mapping

You can map two types of users: users and backend roles. A user can have its own backend role and host for an external authentication and authorization. A backend role directly maps to roles through an external authentication system. [Learn more](#)

User type	User
No user has been mapped to this role	
You can map users or backend roles to this role	

[Create internal user](#) [Map users](#)

3. 在“Mapped users”页签中，单击“Map users”，在users下拉框中选择需要映射的用户。



4. 添加完成后，单击“Map”。
5. 配置完成后，可以在OpenSearch集群中进行验证是否生效。

----结束

3.11.3 设置 OpenSearch 集群标签

标签是集群的标识。为集群添加标签，可以方便用户识别和管理拥有的集群资源。

您可以在创建集群时添加标签，也可以在集群创建完成后，在集群的详情页添加标签。

约束限制

如果您的组织已经设定云搜索服务的相关标签策略，则需按照标签策略规则为集群添加标签。标签如果不符合标签策略的规则，则可能会导致集群创建失败，请联系组织管理员了解标签策略详情。

新建集群的标签管理

1. 登录云搜索服务管理控制台。
2. 单击右上角的“创建集群”，进入创建集群页面。
3. 在创建集群页面，“高级配置”选择“自定义”后，为集群添加标签。
您可以选择预定义标签，并为此标签设置“标签值”。您可以单击“查看预定义标签”，进入“标签管理服务”，了解此用户下已有的标签。
您也可以自定义“标签键”和“标签值”。

图 3-34 创建集群时添加标签



云搜索服务的每个集群最多可以设置20个标签。当设置不正确时, 可单击标签右侧的“删除”按钮, 删除此标签。当不设置标签时, 可保持为空。

表 3-74 标签命名规则

参数	说明
标签键	<ul style="list-style-type: none">对于同一个集群, 标签键值唯一。长度不超过64个字符。只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @”。不能以空格开头和结尾。不能为空。
标签值	<ul style="list-style-type: none">长度不超过64个字符。只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @/”。不能以空格开头和结尾。不能为空。

已有集群的标签管理

您可以对已经创建的集群的标签进行修改, 删除, 也可以添加标签。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面, 单击待管理标签的集群名称。
系统跳转至该集群“基本信息”页面。
3. 左侧菜单栏选择“标签”, 在此可以对集群标签进行添加, 修改, 删除操作。
 - 查看
在“标签”页, 可以查看当前集群的标签详情, 包括标签个数, 以及每个标签的键和值。
 - 添加
单击左上角的“添加标签”, 在弹出的“添加标签”窗口, 输入新添加标签的键和值, 并单击“确定”。
 - 修改
只能修改已有标签的标签值。
单击标签所在行“操作”列下的“编辑”, 在弹出的“编辑标签”窗口, 输入修改后标签值, 并单击“确定”。
 - 删除
单击标签所在行“操作”列下的“删除”, 如果确认删除, 在弹出的“删除标签”窗口, 单击“确定”。

通过标签搜索集群

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击集群列表上方的搜索栏，选择标签键和标签值进行搜索。
标签键和标签值仅支持从下拉列表中选择，当标签键和标签值全匹配时，系统可以自动查询到目标集群。当有多个标签条件时，会取各个标签的交集，进行集群查询。
系统最多支持10个不同标签的组合搜索。

3.11.4 配置 OpenSearch 集群 YML 文件默认参数

云搜索服务支持用户修改elasticsearch.yml文件。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要修改参数配置的集群名称，进入集群基本信息页面。
3. 选择“参数配置”，单击“编辑”，根据需求修改对应模块的参数值。

表 3-75 模块参数信息说明

模块名称	参数名称	说明
跨域访问	http.cors.allow-credentials	跨域访问是否返回头部的Access-Control-Allow-Credentials。 取值范围：true、false。 默认值：false。
	http.cors.allow-origin	允许跨域访问的IP，配置样例如“122.122.122.122:9200”。
	http.cors.max-age	浏览器默认缓存时间。如果超过设置的时间后，缓存将自动清除。 单位：秒。 默认值：1728000。
	http.cors.allow-headers	跨域访问允许的headers，包括X-Requested-With、Content-Type和Content-Length，中间用英文逗号和空格分开。
	http.cors.enabled	是否允许跨域访问。 取值范围：true、false。 默认值：false。
	http.cors.allow-methods	跨域访问允许的方法，包括OPTIONS、HEAD、GET、POST、PUT、DELETE，中间用英文逗号和空格分开。
集群索引重建	reindex.remote.whitelist	配置该参数可以将本集群数据通过reindex接口迁移到配置的集群，配置样例如“122.122.122.122:9200”。

模块名称	参数名称	说明
自定义缓存	indices.queries.cache.size	查询阶段的缓存大小。 取值范围：1-100。 单位：%。 默认值：10%。
线程池队列大小	thread_pool.force_merge.size	用来做forcemerge的队列大小。输入的参数值为整数类型。 默认值：1。
自定义	用户可以根据实际情况，添加相关参数名称。	自定义参数的取值。 说明 <ul style="list-style-type: none">• 如果自定义参数有多个取值，则取值的输入格式为[value1, value1, value1...]• 取值之间用英文逗号和空格隔开。• 自定义参数值中不能包含冒号。

4. 修改完成后，单击上方的“提交”弹出“提交配置”窗口，确认参数无误后勾选“参数修改后需要手动重启才能生效”，单击“确定”。
当下方的参数修改列表显示“作业状态”为“成功”时，表示修改保存成功。系统最多显示20条修改记录。
5. 返回集群列表，单击集群操作列的“更多 > 重启”重启集群，使修改的配置生效。
 - 如果修改了参数配置，未重启集群，则在“集群管理”页面的“任务状态”栏显示为“配置未更新”。
 - 如果修改后重启集群，“任务状态”显示“配置错误”，则表示修改参数配置文件失败。

3.11.5 为 OpenSearch 集群绑定企业项目

企业可以根据组织架构规划企业项目，将企业分布在不同区域的资源按照企业项目进行统一管理，同时可以为每个企业项目设置拥有不同权限的用户组和用户。本章节为您介绍CSS集群如何绑定、修改企业项目。

前提条件

在使用企业项目前，需账号开通权限才可以进行操作。开通权限可以通过提交[工单](#)进行申请。

在绑定企业项目前，您已在“企业项目管理控制台”[创建企业项目](#)。

绑定企业项目

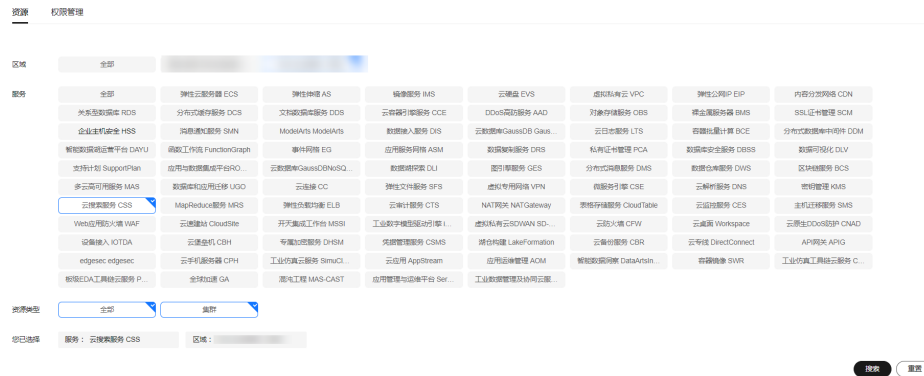
在创建集群时，可以在“企业项目”绑定已创建的企业项目，也可以单击“查看项目管理”，前往企业项目管理控制台，新建企业项目和查看已有的企业项目。

修改企业项目

针对之前已创建的集群，其绑定的企业项目可根据实际情况进行修改。

1. 登录在云搜索服务管理控制台，
2. 在左侧导航栏，选择对应的集群类型，进入集群管理页面。
3. 在集群列表中，单击集群名称进入集群“基本信息”页面。
4. 在集群“基本信息”页面，单击“企业项目”右侧的企业项目名称，进入项目管理页面。
5. 在“资源”页签下，“区域”选项中选择当前集群所在的区域，“服务”选项中选择“云搜索服务 CSS”。此时，资源列表将筛选出对应的CSS集群。

图 3-35 筛选 CSS 集群



6. 勾选需要修改企业项目的集群，然后单击“迁出”。
7. 在“迁出资源”页面，选择“迁出方式”，再选择“请选择要迁入的企业项目”，然后单击“确定”。
8. 迁出完成后，可以在云搜索服务管理控制台集群管理页面，查看修改后的集群企业项目信息。

3.11.6 替换 OpenSearch 集群指定节点

当集群中的节点发生故障时，可以执行“替换指定节点”任务。通过删除故障节点，重建一个一样的新节点，实现节点替换。替换指定节点过程中会将替换节点的数据提前转移，不会造成数据丢失。

前提条件

集群处于“可用”状态，且无正在进行的任务。

约束限制

- 一次只能选择一个节点进行替换。
- 节点替换过程会按照原节点的ID、IP地址、规格、AZ等信息重建节点。
- 节点替换过程中不会保留手动操作。例如被替换的节点手动加过回程路由，那么节点替换完成后，需要重新添加回程路由。
- 当替换的是数据节点或冷数据节点时，需要确认集群或节点是否满足以下条件。
 - a. 替换数据节点或冷数据节点的过程中，会先将替换节点的数据迁移到其他数据节点，因此集群中每个索引的副本数和主分片数之和的最大值要小于集群的数据节点数量和冷数据节点数量之和。替换过程耗时跟数据迁移到其他节点的耗时强相关。
 - b. 版本号在7.6.2以前的集群，不能有close的索引，否则不支持替换数据节点或冷数据节点。

- c. 被替换的数据节点或冷数据节点所在的AZ需要有两个及以上的数据节点或冷数据节点。
- d. 当替换的数据节点或冷数据节点所在集群不存在Master节点时，则集群中可用的数据节点数量和冷数据节点数量之和要大于等于3。
- e. 当替换的是Master节点或Client节点时，则不受以上四条约束。
- f. 当替换的是故障节点时，不管什么节点类型都不受以上四条约束。因为故障节点不包含在“_cat/nodes”中。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
4. 选择“替换指定节点”页签。
5. 在替换指定节点页面，设置参数。

表 3-76 替换指定节点

参数	说明
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平面时才显示该参数。</p> <ul style="list-style-type: none">• 删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。• 当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。• 当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。• 建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。
节点类型	展开本次要变更的节点类型，此处会显示当前节点类型拥有的节点，勾选要替换的节点。

6. 单击“提交申请”。在数据迁移弹窗中确认是否进行数据迁移（建议勾选“进行数据迁移”避免数据丢失），单击“确认”提交申请。

数据迁移过程中，系统会把待替换节点中的所有数据分片移动到剩余节点，待数据迁移完成后再进行节点替换操作。当待替换节点上的数据在剩余节点存在副本时，可跳过数据迁移，缩短集群变更时间，减少数据迁移带来的负载压力。
7. 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“节点替换中”，表示集群正在替换节点。当集群状态变为“可用”，则表示节点替换成功。

3.11.7 更改 OpenSearch 集群安全模式

集群创建完成后，可以参考本文更改集群的安全模式。

场景描述

CSS服务支持创建多种安全模式的集群，不同集群的差异请参见[表3-77](#)。

表 3-77 集群类型介绍

集群类型		集群描述	适用场景
非安全集群	非安全模式的集群	非安全模式的集群无需安全认证即可访问，采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	适合内网业务，用于测试场景。 <ul style="list-style-type: none"> • 优点：简单，接入集群容易。 • 缺点：安全性差，任何人都可以访问集群。
安全集群	安全模式+HTTP协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTP协议明文传输数据。建议确认访问环境的安全性，勿将访问接口暴露到公网环境上。	可以实现用户权限隔离，适用于对集群性能敏感的场景。 <ul style="list-style-type: none"> • 优点：访问集群需要安全认证，提升了集群安全性。通过HTTP协议访问集群又能保留集群的高性能。 • 缺点：无法公网访问集群。
	安全模式+HTTPS协议的集群	安全模式的集群需要通过安全认证才能访问，且支持对集群进行授权、加密等功能。采用HTTPS协议进行通信加密，使数据更安全。	有非常高的安全要求，且需要公网访问集群的场景。 <ul style="list-style-type: none"> • 优点：访问集群需要安全认证，提升了集群安全性，且HTTPS协议的通讯加密可以实现集群公网访问功能。 • 缺点：通过HTTPS协议访问集群，集群的读取性能相对HTTP协议来说，会下降20%左右。

同时，CSS服务支持多种安全模式的更改场景，请参见[表3-78](#)。

表 3-78 安全模式的更改场景

场景	相关文档
“非安全模式的集群”切换为“安全模式+HTTP协议的集群”	非安全模式切换为安全模式
“非安全模式的集群”切换为“安全模式+HTTPS协议的集群”	

场景	相关文档
“安全模式+HTTP协议的集群”切换为“非安全模式的集群”	安全模式切换为非安全模式
“安全模式+HTTPS协议的集群”切换为“非安全模式的集群”	
“安全模式+HTTP协议的集群”切换为“安全模式+HTTPS协议的集群”	切换安全模式下的协议
“安全模式+HTTPS协议的集群”切换为“安全模式+HTTP协议的集群”	

前提条件

- 建议更改集群安全模式前，先完成数据备份。
- 集群必须处于“可用”状态，且无正在进行的任务。
- 确认集群是否启用“负载均衡”，如果启用了，则先关闭集群的“负载均衡”，待安全模式变更完成后再开启“负载均衡”。避免变更过程中造成通过负载均衡实例访问集群异常。

约束限制

- 2022年11月之后创建的集群（且集群版本不小于6.5.4）才支持切换安全模式。
- 在更改集群安全模式的过程中，集群会自动重启。重启会导致业务中断，并且重启后调用集群的认证方式也会发生改变，客户端需要做相应调整。
- 对于已打开过Kibana会话框的集群，在更改集群安全模式后，Kibana会提示Session错误，此时需要清理缓存再打开Kibana才能正常访问。

非安全模式切换为安全模式

介绍集群从“非安全模式”切换为“安全模式+HTTP协议”或“安全模式+HTTPS协议”的操作。当集群从非安全模式变更为安全模式后，访问集群将需要进行安全认证。

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择“集群管理>Elasticsearch”，进入Elasticsearch集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
4. 选择“更改安全模式”页签。
5. 在更改安全模式页面，打开“安全模式”的开关，输入并确认集群的管理员密码。

图 3-36 非安全模式切换为安全模式

安全模式 开启安全模式，访问集群需要安全认证。

管理员账户名 admin

管理员密码

确认密码

HTTPS访问

- 选择集群是否启用“HTTPS访问”。
 - 打开开关：实现“非安全模式”切换为“安全模式+HTTPS协议”。集群通讯使用HTTPS协议，通讯数据将进行加密，且可以启用集群的公网访问功能。
 - 关闭开关：实现“非安全模式”切换为“安全模式+HTTP协议”。集群通讯使用HTTP协议，无法启用集群的公网访问功能。
- 单击“提交申请”，确认后将返回集群列表页面。

集群的“任务状态”列中显示为“安全模式更改中”，当“集群状态”变为“可用”表示变更成功。

安全模式切换为非安全模式

介绍集群从“安全模式+HTTP协议”或“安全模式+HTTPS协议”切换为“非安全模式”的操作。当集群从安全模式变更为非安全模式后，访问集群将不再需要进行安全认证。

须知

- 非安全模式的集群将无需安全认证即可访问，且采用的是HTTP协议传输数据，所以请确保集群访问环境的安全性，勿将访问接口暴露到公网环境上。
- 在安全模式切换为非安全模式的过程中，变更任务会删除原安全模式的集群用到的索引。切换前请做好数据备份，以免数据丢失。
- 如果集群已绑定公网IP，则无法将安全模式切换为非安全模式，需要先解绑公网IP，才能切换。
- 如果集群已启用Kibana公网访问，则无法将安全模式切换为非安全模式，需要先关闭Kibana公网访问功能，才能切换。

- 登录云搜索服务管理控制台。
- 左侧导航栏选择“集群管理”，进入集群列表页面，选择需要更改安全模式的集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
- 选择“更改安全模式”页签。
- 在更改安全模式页面，关闭“安全模式”的开关。

图 3-37 安全模式切换为非安全模式

安全模式 关闭安全模式后，集群将无需安全认证即可访问，并且采用HTTP明文传输数据，所以请确保访问环境的安全性，勿将访问接口暴露到公网环境上。

- 单击“提交申请”，确认后将返回集群列表页面。
集群的“任务状态”列中显示为“安全模式更改中”，当“集群状态”变为“可用”表示变更成功。

切换安全模式下的协议

介绍安全模式的集群切换访问协议的操作，包括“安全模式+HTTP协议”切换为“安全模式+HTTPS协议”、“安全模式+HTTPS协议”切换为“安全模式+HTTP协议”。

须知

如果集群已绑定公网IP，则无法将协议从HTTPS切换到HTTP，需要先解绑公网IP，才能切换安全模式下的协议。

- 登录云搜索服务管理控制台。
- 左侧导航栏选择“集群管理”，进入集群列表页面，选择需要更改安全模式的集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
- 选择“更改安全模式”页签。
- 在更改安全模式页面，修改“HTTPS访问”的开关切换安全模式下的集群访问协议。

图 3-38 切换协议

安全模式	<input checked="" type="checkbox"/> 开启安全模式，访问集群需要安全认证。
管理员账户名	admin
管理员密码	<input type="password"/>
确认密码	<input type="password"/>
HTTPS访问	<input checked="" type="checkbox"/> 开启HTTPS访问，访问集群将进行通讯加密。

- 打开开关：实现“安全模式+HTTP协议”切换为“安全模式+HTTPS协议”。切换为HTTPS协议后，集群通讯将进行加密，且可以启用集群的公网访问功能。
 - 关闭开关：实现“安全模式+HTTPS协议”切换为“安全模式+HTTP协议”。关闭开关会弹出告警提示，单击“确定”后关闭开关。
切换为HTTP协议后，集群通讯将不再加密，且无法启用集群的公网访问功能。
- 单击“提交申请”，确认后将返回集群列表页面。
集群的“任务状态”列中显示为“安全模式更改中”，当“集群状态”变为“可用”表示变更成功。

3.11.8 切换 OpenSearch 集群可用区

CSS服务支持跨可用区的能力，在集群使用过程中遇到可用区资源不足或需要提升集群的高可用性时，可以通过切换可用区实现。

场景描述

切换可用区包含两大场景：可用区高可用改造和可用区平移切换。

- 可用区高可用改造：适用于单AZ改造成两AZ、单AZ改造成三AZ或两AZ改造成三AZ的场景，目的是为了提升集群的高可用性。
- 可用区平移切换：适用于从一个AZ完全迁移到另一个AZ的场景，是为了解决当前可用区资源不足的问题。

前提条件

- 确保存在资源充足的可用区。
- 集群处于“可用”状态，且无正在进行的任务。
- 确认集群不存在非标操作。当集群进行过修改回程路由、系统参数、Kibana配置等这些非标操作时，这些操作无法被记录到系统中，因此在切换过程中将无法继承，切换后可能会影响业务。

约束限制

- 为了保证业务的连续性，集群中数据节点和冷数据节点的个数之和要不小于3。
- 切换过程中会逐一下线节点再新建节点，需要保证单节点下线后，其余节点的磁盘容量能够接纳该节点的所有数据。
- 集群中索引的最大主备分片数能够被分配到剩余数据节点和冷数据节点中，防止切换过程中出现节点下线后副本无法分配的情况。即“索引的主备分片数的最大值+1 ≤ 切换前的数据节点数和冷数据节点之和”。
- 切换前建议做好数据备份，防止升级故障数据丢失。
- 当集群处于切换可用区过程中，部分节点的AZ属性可能已经更改，此时节点的可用区可能会显示出中间状态，等待任务结束后，集群的可用区显示将恢复正常状态。
- **可用区高可用改造**是基于原可用区进行扩展，切换时必须保留原可用区。如果是单AZ改造成两AZ或单AZ改造成三AZ的场景，必须所有节点一起改造；如果是两AZ改造成三AZ的场景，则既支持同时操作集群中所有类型的节点，也支持单独操作集群中某一类型的节点（例如，在已有两AZ的集群中，可以只对Master节点进行切换可用区，将Master节点形变为三AZ，而其他节点依旧是两AZ）。高可用改造会尽可能挪动最小的节点重建集群，改造完成后，因未挪动节点的yml配置更新，需要重启集群才能生效。
- **可用区平移切换**一次只能迁移一个可用区，切换时选择的可用区只能有一个不同。平移切换支持同时操作集群中所有类型的节点，或者单独操作集群中某一类型的节点（例如，在已有两AZ的集群中，可以只对Master节点进行平移切换，将Master节点从当前AZ迁移到另一个AZ，而其他节点依旧在当前AZ）。除了单AZ平移切换场景，其他的多AZ平移切换场景均需要重启集群生效。

操作步骤

1. 登录云搜索服务管理控制台。

2. 在左侧菜单栏，选择对应的集群类型，进入集群管理页面。
3. 选择目标集群，单击操作列的“更多>形态变更”进入更改集群规格页面。
4. 选择“切换可用区”页签。
5. 在切换可用区页面，配置切换参数。

表 3-79 切换可用区的参数说明

参数	描述
切换类型	<ul style="list-style-type: none">• 可用区高可用改造：支持单AZ改造成两AZ、单AZ改造成三AZ或两AZ改造成三AZ。 高可用改造会尽可能挪动最小的节点重建集群，改造完成后，因未挪动节点的yaml配置更新，需要重启集群才能生效。• 可用区平移切换：支持从一个AZ迁移到另一个AZ。 除了单AZ平移切换场景，其他的多AZ平移切换场景均需要重启集群生效。
当前节点类型	选择进行AZ切换的节点类型。每次可选择一种节点类型进行操作，当集群有多种节点类型时，可选择“所有节点”进行全部切换。 说明 如果是单AZ改造成两AZ或单AZ改造成三AZ的场景，只支持所有节点一起改造，因此只能选择“所有节点”。
当前AZ集合	显示当前集群所在的可用区。
目标AZ集合	选择需要切换的目标可用区。 <ul style="list-style-type: none">• 可用区高可用改造：高可用区改造是基于原可用区进行扩展，因此“目标AZ集合”必须包含“当前AZ集合”且至少再选择一个可用区，最多选择3个可用区。• 可用区平移切换：一次切换任务只能迁移一个AZ，因此“目标AZ集合”只能有一个可用区和“当前AZ集合”不同。
配置委托	选择IAM委托，授权当前账号切换可用区的操作权限。 如果没有合适的委托，可以单击“创建委托”跳转到IAM控制台新建委托。 说明 选择的委托必须授权了“Tenant Administrator”或“VPC Administrator”策略。

6. 配置完成后，单击“提交申请”。确认是否进行全量索引快照备份检测，单击“确定”启动切换任务。

图 3-39 全量索引快照备份检测

全量索引快照备份检测

建议您在切换前进行全量索引快照备份检测。
如果您已备份，请检查备份时间是否超过1个月，若超过1个月，建议您备份最新数据。
当前检测只能针对索引名称进行匹配，无法校验具体内容以及备份时间。

进行全量索引快照备份检测

取消

确定

- 在“任务记录”列表，显示当前切换任务。当“任务状态”为“运行中”时，可以展开任务列表，单击“查看进度”查看详细的切换进度。

当“任务状态”为“失败”时，可以重试任务或者直接终止任务。

- 重试切换任务：在任务列表的操作列，单击“重试”，重新切换可用区。
- 终止切换任务：在任务列表的操作列，单击“终止”，结束切换可用区。

当任务终止后，如果原节点还未切换成功，可以通过[替换Elasticsearch集群指定节点](#)尝试恢复节点。

📖 说明

如果存在部分节点已经完成切换，此时集群的AZ形态已经发生变化，此时终止切换任务，可能无法再次按照上一次的切换请求进行命令下发，请谨慎执行终止切换任务。

3.11.9 配置 OpenSearch 集群自定义词库

3.11.9.1 OpenSearch 集群词库介绍

背景信息

云搜索服务的词库用于对文本进行分词，使得一些特殊词语在分词的时候能够被识别出来，便于根据关键词搜索文本数据。例如，根据公司名称来查询，如“华为”；或者根据网络流行词来查询，如“喜大普奔”。也支持基于同义词词库，根据同义词搜索文本数据。

CSS服务使用的分词器包括IK分词器和同义词分词器。IK分词器配备主词词库和停词词库；同义词分词器配备同义词词库。其中，IK分词器包含ik_max_word和ik_smart分词策略。同义词分词器使用的是ik_synonym分词策略。

- ik_max_word：会将文本做最细粒度的拆分，比如会将“昨夜西风吹折干林梢”拆分为“昨夜西风,昨夜,西风,吹折干林梢,吹折,干林梢,干,林,折干林,干林,吹”，会穷尽各种可能的分词组合。
- ik_smart：会做最粗粒度的拆分，比如会将“昨夜西风吹折干林梢”拆分为“昨夜西风,吹折干林梢”。

CSS服务给集群预置了静态主词词库、静态停词词库、Extra主词词库和Extra停词词库这四个词库。

- 当这些预置词库已满足集群业务的分词需求时，则集群无需配置自定义词库即可直接实现关键词搜索。
- 当预置词库不满足集群业务分词需求时，可以给集群添加主词词库、停词词库或同义词词库，亦或者是修改预置的四个词库，使集群能够实现关键词或同义词搜索。

CSS服务的各类词库介绍请参见表3-80。

表 3-80 词库介绍

词库类型		说明	文件要求
自定义	主词词库	主词为用户希望进行分词的特殊词语，例如“智能手机”和“喜大普奔”。主词词库则是用户自定义的特殊词语的集合。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，主词文件最大支持100M。如果涉及单词，必须改成小写字母。
	停词词库	停词为用户不希望进行分词或者关注的词语，例如“的”、“什么”、“怎么”等。停词词库是用户自定义的停词词语的集合。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，停词文件最大支持100M。
	同义词词库	同义词为意义相同的一组词语，例如“开心”和“高兴”。同义词词库是用户自定义的同义词词语的集合。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一组同义词对，同义词对中的同义词间逗号隔开，同义词文件最大支持100MB。
预置	静态主词词库	静态主词词库是CSS服务预置的常用主词集合，当需要查看静态主词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/main.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，静态主词文件最大支持100M。
	静态停词词库	静态停词词库是CSS服务预置的常用停词集合，当需要查看静态停词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/stopword.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，静态停词文件最大支持100M。
	Extra主词词库	Extra主词词库是CSS服务预置的生僻主词集合，当需要查看Extra主词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/extra_main.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，Extra主词文件最大支持100M。

词库类型	说明	文件要求
Extra停词词库	Extra停词词库是CSS服务预置的生僻停词集合，当需要查看Extra停词词库时，可以单击地址 https://github.com/infinilabs/analysis-ik/blob/master/config/extra_stopword.dic 获取词库。	词库文件必须是UTF-8无BOM格式编码的文本文件，一行一个分词，Extra停词文件最大支持100M。

约束限制

- 自定义词库功能上线之前（即2018年3月10日之前）创建的集群，无法使用自定义词库功能。
- 自定义词库一般用于中文分词，如果用于英文分词，会按照除“#&+-.@_”外的特殊符号进行切分。
- 词库文件要满足要求，不同词库的词库文件要求请参见[表3-80](#)。
- 操作账号必须具有配置自定义词库的权限。可以在IAM服务的“权限管理 > 权限”查看是否有以下权限，如果没有请联系服务管理员添加。
 - “CSS Administrator”角色。
 - “OBS Administrator”策略

3.11.9.2 配置和使用 OpenSearch 集群的自定义词库

前提条件

已经准备好集群和自定义词库，且词库文件已上传至OBS桶。

- 集群和词库文件满足[约束限制](#)。
- 上传的OBS桶必须和集群在相同“区域”。上传文件至OBS桶的操作步骤请参见[上传文件](#)。

配置自定义词库

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表页面。
3. 在“集群管理”页面，单击需要配置自定义词库的集群名称，进入集群基本信息页面。
4. 选择“自定义词库”。
5. 在“自定义词库”页面，配置集群的自定义词库或修改预置词库。
 - a. 当需要配置用户自定义的词库时，参考[表3-81](#)完成配置。

表 3-81 配置自定义词库

参数	说明
OBS桶	选择词库文件存储的OBS位置。 单击“创建桶”可以跳转到创建桶页面新建OBS桶，新建OBS桶必须和集群在相同“区域”，且“默认存储类别”只支持“标准存储”或“低频访问存储”。
主词词库	主词词库是用户自定义的词库，初始状态为空。默认选择“不更新”表示不配置该词库。 <ul style="list-style-type: none">当需要添加自定义的主词词库时，单击“更新”，选择txt格式的词库文件。当无需添加自定义的主词词库时，单击“不使用此词库”，删除词库。
停词词库	停词词库是用户自定义的词库，初始状态为空。默认选择“不更新”表示不配置该词库。 <ul style="list-style-type: none">当需要添加自定义的停词词库时，单击“更新”，选择txt格式的词库文件。当无需添加自定义的停词词库时，单击“不使用此词库”，删除词库。
同义词词库	同义词词库是用户自定义的词库，初始状态为空。默认选择“不更新”表示不配置该词库。 <ul style="list-style-type: none">当需要添加自定义的同义词词库时，单击“更新”，选择txt格式的词库文件。当无需添加自定义的同义词词库时，单击“不使用此词库”，删除词库。

- b. 当需要修改预置词库时，单击打开“修改预置词库”右侧的开关，修改对应的预置词库。

说明

如果界面不存在静态词库和Extra词库这4个词库时，表示该集群版本不支持删除或修改这4个预置词库。如果想要使用该功能，建议升级集群版本，或者新建集群并进行数据迁移。

表 3-82 配置预置词库

参数	说明
静态主词词库	静态主词词库是预置的常用词语的主词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的静态主词词库时，单击“更新”，选择txt格式的词库文件。当不使用静态主词库时，单击“不使用此词库”，删除词库。

参数	说明
静态停词词库	静态停词词库是预置的常用词语的停词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的静态停词词库时，单击“更新”，选择txt格式的词库文件。当不使用静态停词库时，单击“不使用此词库”，删除词库。
Extra主词词库	Extra主词词库是预置的生僻词语的主词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的Extra主词词库时，单击“更新”，选择txt格式的词库文件。当不使用Extra主词库时，单击“不使用此词库”，删除词库。
Extra停词词库	Extra停词词库是预置的生僻词语的Extra停词库。默认选择“不更新”表示使用该预置词库。 <ul style="list-style-type: none">当需要修改预置的Extra停词词库时，单击“更新”，选择txt格式的的词库文件。当不使用Extra停词库时，单击“不使用此词库”，删除词库。

- 单击“保存”，在弹窗中单击“确定”。词库信息在下方呈现，此时词库状态为“更新中”。请耐心等待1分钟左右，当词库配置完成后，词库状态变更为“成功”。
- 当删除或更新静态词库和Extra词库这4个词库时，需要重启集群才能使配置的词库生效；其他词库的更新为动态更新，无需重启集群。重启集群的操作指导请参见[重启Elasticsearch集群](#)。

使用示例

通过给集群配置自定义词库，将“智能手机”设置为主词，“是”设置为停词，“开心”和“高兴”设置为同义词。使用配置好的集群，对文本内容“智能手机是很好用”进行关键词搜索，查看关键词查询效果；对文本内容“我今天获奖了我很开心”进行同义词搜索，查看同义词查询效果。

步骤1 配置自定义词库，并查看分词效果。当预置词库即可满足分词需求时，可以跳过此步骤。

- 准备词库文件（UTF-8无BOM格式编码的文本文件），上传到对应OBS路径下。主词词库文件中包含词语“智能手机”；停词词库文件中包含词语“是”；同义词词库文件中包含一组同义词“开心”和“高兴”。

说明

由于系统预置的静态停词词库包含了“是”、“的”等常用词，如果集群未删除或更新预置词库，则此类停用词可以不用上传。

- 参考[配置自定义词库](#)，完成词库配置。
- 待词库配置信息生效后，返回集群列表。单击集群操作列的“Kibana”接入集群。

4. 在Kibana界面，单击左侧导航栏的“Dev Tools”，进入操作页面。
5. 执行如下命令，查看自定义词库的ik_smart分词策略和ik_max_word分词策略的分词效果。

- 使用ik_smart分词策略对文本内容“智能手机是很好用”进行分词。

示例代码：

```
POST /_analyze
{
  "analyzer":"ik_smart",
  "text":"智能手机是很好用"
}
```

运行结束后，查看分词效果：

```
{
  "tokens": [
    {
      "token": "智能手机",
      "start_offset": 0,
      "end_offset": 4,
      "type": "CN_WORD",
      "position": 0
    },
    {
      "token": "很好用",
      "start_offset": 5,
      "end_offset": 8,
      "type": "CN_WORD",
      "position": 1
    }
  ]
}
```

- 使用ik_max_word分词策略对文本内容“智能手机是很好用”进行分词。

示例代码：

```
POST /_analyze
{
  "analyzer":"ik_max_word",
  "text":"智能手机是很好用"
}
```

运行结束后，查看分词效果：

```
{
  "tokens": [
    {
      "token": "智能手机",
      "start_offset": 0,
      "end_offset": 4,
      "type": "CN_WORD",
      "position": 0
    },
    {
      "token": "智能",
      "start_offset": 0,
      "end_offset": 2,
      "type": "CN_WORD",
      "position": 1
    },
    {
      "token": "智",
      "start_offset": 0,
      "end_offset": 1,
      "type": "CN_WORD",
      "position": 2
    },
    {
      "token": "能手",
      "start_offset": 1,

```

```
"end_offset" : 3,
"type" : "CN_WORD",
"position" : 3
},
{
  "token" : "手机",
  "start_offset" : 2,
  "end_offset" : 4,
  "type" : "CN_WORD",
  "position" : 4
},
{
  "token" : "机",
  "start_offset" : 3,
  "end_offset" : 4,
  "type" : "CN_WORD",
  "position" : 5
},
{
  "token" : "很好用",
  "start_offset" : 5,
  "end_offset" : 8,
  "type" : "CN_WORD",
  "position" : 6
},
{
  "token" : "很好",
  "start_offset" : 5,
  "end_offset" : 7,
  "type" : "CN_WORD",
  "position" : 7
},
{
  "token" : "好用",
  "start_offset" : 6,
  "end_offset" : 8,
  "type" : "CN_WORD",
  "position" : 8
},
{
  "token" : "用",
  "start_offset" : 7,
  "end_offset" : 8,
  "type" : "CN_WORD",
  "position" : 9
}
]
}
```

步骤2 创建索引并配置分词策略，导入数据后使用关键词搜索数据。

Elasticsearch 7.x之前的版本和之后的版本，操作命令有差别，所以分开举例。

Elasticsearch版本<7.x的代码示例

1. 创建索引“book”，配置分词策略。

示例中“analyzer”和“search_analyzer”可以根据实际需要“ik_max_word”或“ik_smart”分词策略，此处以“ik_max_word”为例。

```
PUT /book
{
  "settings": {
    "number_of_shards": 2,
    "number_of_replicas": 1
  },
  "mappings": {
    "type1": {
      "properties": {
        "content": {
```

```
        "type": "text",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word"
      }
    }
  }
}
```

2. 导入数据，将文本信息导入“book”索引中。

```
PUT /book/type1/1
{
  "content": "智能手机是很好用"
}
```

3. 使用关键词“智能手机”进行文本搜索，并查看搜索结果。

```
GET /book/type1/_search
{
  "query": {
    "match": {
      "content": "智能手机"
    }
  }
}
```

搜索结果：

```
{
  "took" : 20,
  "timed_out" : false,
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.1507283,
    "hits" : [
      {
        "_index" : "book",
        "_type" : "type1",
        "_id" : "1",
        "_score" : 1.1507283,
        "_source" : {
          "content" : "智能手机是很好用"
        }
      }
    ]
  }
}
```

Elasticsearch版本≥7.x和OpenSearch版本的代码示例

1. 创建索引“book”，配置分词策略。

示例中“analyzer”和“search_analyzer”可以根据实际需要“ik_max_word”或“ik_smart”分词策略，此处以“ik_max_word”为例。

```
PUT /book
{
  "settings": {
    "number_of_shards": 2,
    "number_of_replicas": 1
  },
  "mappings": {
    "properties": {
      "content": {
        "type": "text",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word"
      }
    }
  }
}
```



```
    }  
  }  
}
```

2. 导入数据，将文本信息导入“book”索引中。

```
PUT /book/_doc/1  
{  
  "content": "智能手机是很好用"  
}
```

3. 使用关键词“智能手机”进行文本搜索，并查看搜索结果。

```
GET /book/_doc/_search  
{  
  "query": {  
    "match": {  
      "content": "智能手机"  
    }  
  }  
}
```

搜索结果：

```
{  
  "took" : 16,  
  "timed_out" : false,  
  "_shards" : {  
    "total" : 2,  
    "successful" : 2,  
    "skipped" : 0,  
    "failed" : 0  
  },  
  "hits" : {  
    "total" : {  
      "value" : 1,  
      "relation" : "eq"  
    },  
    "max_score" : 1.7260926,  
    "hits" : [  
      {  
        "_index" : "book",  
        "_type" : "_doc",  
        "_id" : "1",  
        "_score" : 1.7260926,  
        "_source" : {  
          "content" : "智能手机是很好用"  
        }  
      }  
    ]  
  }  
}
```

步骤3 创建索引并配置同义词策略，导入数据后使用同义词搜索数据。

Elasticsearch 7.x之前的版本和之后的版本，操作命令有差别，所以分开举例。

Elasticsearch版本<7.x的代码示例

1. 创建索引“myindex”，配置分词策略。

```
PUT myindex  
{  
  "settings": {  
    "analysis": {  
      "filter": {  
        "my_synonym": {  
          "type": "dynamic_synonym"  
        }  
      },  
      "analyzer": {  
        "ik_synonym": {  
          "filter": [  
            "my_synonym"  
          ]  
        }  
      }  
    }  
  }  
}
```

```
    "my_synonym"
  ],
  "type": "custom",
  "tokenizer": "ik_smart"
}
}
},
"mappings": {
  "mytype": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ik_synonym"
      }
    }
  }
}
```

2. 导入数据，将文本信息导入“myindex”索引中。

```
PUT /myindex/mytype/1
{
  "desc": "我今天获奖了我很开心"
}
```

3. 使用同义词“高兴”进行文本搜索，并查看搜索结果。

```
GET /myindex/_search
{
  "query": {
    "match": {
      "desc": "高兴"
    }
  }
}
```

搜索结果：

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 0.49445358,
    "hits" : [
      {
        "_index" : "myindex",
        "_type" : "mytype",
        "_id" : "1",
        "_score" : 0.49445358,
        "_source" : {
          "desc" : "我今天获奖了我很开心"
        }
      }
    ]
  }
}
```

Elasticsearch版本≥7.x和OpenSearch版本的代码示例

1. 创建索引“myindex”，配置分词策略。

```
PUT myindex
{
  "settings": {
    "analysis": {
```

```
    "filter": {
      "my_synonym": {
        "type": "dynamic_synonym"
      }
    },
    "analyzer": {
      "ik_synonym": {
        "filter": [
          "my_synonym"
        ],
        "type": "custom",
        "tokenizer": "ik_smart"
      }
    }
  },
  "mappings": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ik_synonym"
      }
    }
  }
}
```

2. 导入数据，将文本信息导入“myindex”索引中。

```
PUT /myindex/_doc/1
{
  "desc": "我今天获奖了我很开心"
}
```

3. 使用同义词“高兴”进行文本搜索，并查看搜索结果。

```
GET /myindex/_search
{
  "query": {
    "match": {
      "desc": "高兴"
    }
  }
}
```

搜索结果：

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.1519955,
    "hits" : [
      {
        "_index" : "myindex",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.1519955,
        "_source" : {
          "desc" : "我今天获奖了我很开心"
        }
      }
    ]
  }
}
```

```
}  
}
```

----结束

3.11.10 在 OpenSearch 集群实现搜索数据的简繁体转换

本文介绍如何在Elasticsearch集群实现搜索数据的简繁体转换。

场景介绍

简繁体转换插件是一款可以使中文简体和中文繁体相互转换的插件。通过该插件的转换，用户可以使用中文繁体关键字搜索出包含对应中文简体的索引数据，也可以使用中文简体关键字搜索出包含对应中文繁体的索引数据。

云搜索服务默认安装了简繁体转换插件，用户无需自行安装。

简繁体转换插件通常可以当做analyzer、tokenizer、token-filter或char-filter来使用。

简繁体转换插件的转换类型包含如下两种：

- s2t：将中文简体转换为中文繁体。
- t2s：将中文繁体转换为中文简体。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏中，选择“集群管理”，进入集群列表页面。
3. 在集群列表中，单击需要使用的集群对应“操作”列的“Kibana”。
如果开启了安全模式，需要输入创建集群时设置的管理员账户名和密码。
4. 在Kibana的左侧导航中选择“Dev Tools”，进入Console界面。
5. 在Console界面，执行如下命令，创建索引“stconvert”，并指定自定义映射来定义数据类型。

Elasticsearch低于7.x版本的代码示例

```
PUT /stconvert  
{  
  "settings": {  
    "number_of_shards": 1,  
    "number_of_replicas": 0,  
    "analysis": {  
      "analyzer": {  
        "ts_ik": {  
          "tokenizer": "ik_smart",  
          "char_filter": [  
            "tsconvert",  
            "stconvert"  
          ]  
        }  
      },  
      "char_filter": {  
        "tsconvert": {  
          "type": "stconvert",  
          "convert_type": "t2s"  
        },  
        "stconvert": {  
          "type": "stconvert",  
          "convert_type": "s2t"  
        }  
      }  
    }  
  }  
}
```

```
    }
  },
  "mappings": {
    "type": {
      "properties": {
        "desc": {
          "type": "text",
          "analyzer": "ts_ik"
        }
      }
    }
  }
}
```

Elasticsearch大于或等于7.x版本和OpenSearch集群的代码示例

```
PUT /stconvert
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 0,
    "analysis": {
      "analyzer": {
        "ts_ik": {
          "tokenizer": "ik_smart",
          "char_filter": [
            "tsconvert",
            "stconvert"
          ]
        }
      },
      "char_filter": {
        "tsconvert": {
          "type": "stconvert",
          "convert_type": "t2s"
        },
        "stconvert": {
          "type": "stconvert",
          "convert_type": "s2t"
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ts_ik"
      }
    }
  }
}
```

返回结果如下所示。

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "stconvert"
}
```

6. 在Console界面，执行如下命令，导入数据到“stconvert”索引中。

Elasticsearch低于7.x版本的代码示例

```
POST /stconvert/type/1
{
  "desc": "國際電視臺"
}
```

Elasticsearch大于或等于7.x版本和OpenSearch集群的代码示例

```
POST /stconvert/_doc/1
{
```

```
"desc": "國際電視臺"}
}
```

当返回结果信息中“failed”字段的值为“0”时，表示数据导入成功。

7. 在Console界面，执行如下命令，搜索关键字“国际”，并查看搜索结果。

```
GET /stconvert/_search
{
  "query": {
    "match": {
      "desc": "国际"
    }
  }
}
```

搜索结果如下所示。

```
{
  "took": 15,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.5753642,
    "hits": [
      {
        "_index": "stconvert",
        "_type": "type",
        "_id": "1",
        "_score": 0.5753642,
        "_source": {
          "desc": "國際電視臺"
        }
      }
    ]
  }
}
```

3.11.11 重启 OpenSearch 集群

集群停止工作时，您可通过重启集群恢复运行。

场景描述

重启集群支持快速重启和滚动重启，两种方案的使用说明请参见表3-83。建议在业务空闲时执行重启操作。

表 3-83 重启方案

方案	使用说明	相关文档
快速重启	<ul style="list-style-type: none">所有集群都支持。当选择“节点类型”快速重启时，所选类型的所有节点会一起重启。当选择“节点名称”快速重启时，一次只能重启一个节点。快速重启过程中，集群不可用。	快速重启

方案	使用说明	相关文档
滚动重启	<ul style="list-style-type: none">仅当集群的节点数量（含Master节点、Client节点和冷数据节点）大于等于3时，才支持滚动重启。滚动重启只支持根据“节点类型”进行重启。选择节点类型滚动重启时，所选类型的节点会依次重启。滚动重启过程中，只有正在重启的节点不可用，不在重启过程中的节点可以正常提供服务。当数据量比较大时，滚动重启耗时较长。	滚动重启

前提条件

- 确认集群的“任务状态”没有正在执行中的任务，且集群未被冻结。
- 当集群处于可用状态时，确认集群已停止处理业务数据（如导入数据、搜索数据），否则重启集群时可能导致数据丢失等。

快速重启

- 登录云搜索服务管理控制台。
- 在左侧导航栏，选择对应的集群类型，进入集群管理列表界面。
- 在对应集群的“操作”列中单击“更多>重启”。
- 在“重启集群”页面，选择“快速重启”。
快速重启支持根据“节点类型”或者“节点名称”重启。如果选择“节点类型”，则支持选择多种节点类型同时进行快速重启。如果选择“节点名称”，则一次只能快速重启一个节点。
- 重启集群后，请刷新页面，观察集群状态。重启过程中，集群状态为“处理中”，任务状态为“重启中”。如果集群状态变更为“可用”，表示集群已重启成功。

滚动重启

- 登录云搜索服务管理控制台。
- 在左侧导航栏，选择对应的集群类型，进入集群管理列表界面。
- 在对应集群的“操作”列中单击“更多>重启”。
- 在“重启集群”页面，选择“滚动重启”。
滚动重启支持根据“节点类型”进行重启。如果只需要重启集群中的某些类型的节点时，可以选择需要重启的节点类型。
- 重启集群后，请刷新页面，观察集群状态。重启过程中，集群状态为“处理中”，任务状态为“重启中”。如果集群状态变更为“可用”，表示集群已重启成功。

3.11.12 删除 OpenSearch 集群

当用户已完成数据搜索业务，无需继续使用某一集群时，可删除集群释放资源。

约束限制

- 删除集群时，会清理集群业务数据，请谨慎操作。
- 当集群启用过快照功能，且OBS桶中创建的快照并未被删除，此时删除集群时，并不会释放这部分备份数据。如果有需要，可以通过OBS桶中存储的快照信息恢复集群，具体参见[集群被删除后是否还能恢复？](#)。

删除按需计费的集群

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表界面。
3. 在对应集群的“操作”列中单击“更多>删除”。
4. 在弹出的确认提示框中，输入需要删除的集群名称，单击“确定”完成集群删除。

删除包年/包月的集群

包年包月计费模式的集群需要通过退订包周期删除集群。集群退订后，资源和数据将会被清空，且无法恢复。请务必确认完成数据备份后再提交退订。

1. 在集群管理页面，选择需要退订的集群。
2. 在操作列单击“更多”>“退订/释放”。
3. 输入需要退订/释放的集群名称，单击“确定”。
进入退订资源页面，可以在该页面核对资源信息以及退费金额。
4. 填写退订原因，勾选相关协议后，单击“退订”。
在弹出确认退订提示信息后，再次单击“退订”。

说明

当集群处于生效中状态时，则走退订流程，此时会产生一个订单进行退费，然后删除集群。当集群处于已过期或者已冻结状态时，则走释放流程，直接删除集群。集群退订的使用说明请参考[费用中心](#)相关描述。

3.12 管理 OpenSearch 集群索引策略

3.12.1 创建及管理 OpenSearch 索引策略

本文介绍如何创建和管理OpenSearch集群的生命周期索引策略。

背景信息

OpenSearch的ISM (Index State Management) 是一个索引状态管理插件，支持通过索引使用期限、索引大小或文档数等信息的变化来自动触发周期性的管理操作。通过ISM插件可以自定义索引策略，实现自动处理索引的滚动或删除，以优化集群搜索性能或降低存储成本。ISM的使用流程：

1. **创建生命周期策略**：在Kibana创建索引的生命周期策略Policy。
2. **索引关联生命周期策略**：将索引关联上创建的生命周期策略，生成索引策略。
3. **管理索引策略**：支持对索引策略进行策略修改、重启和变更。

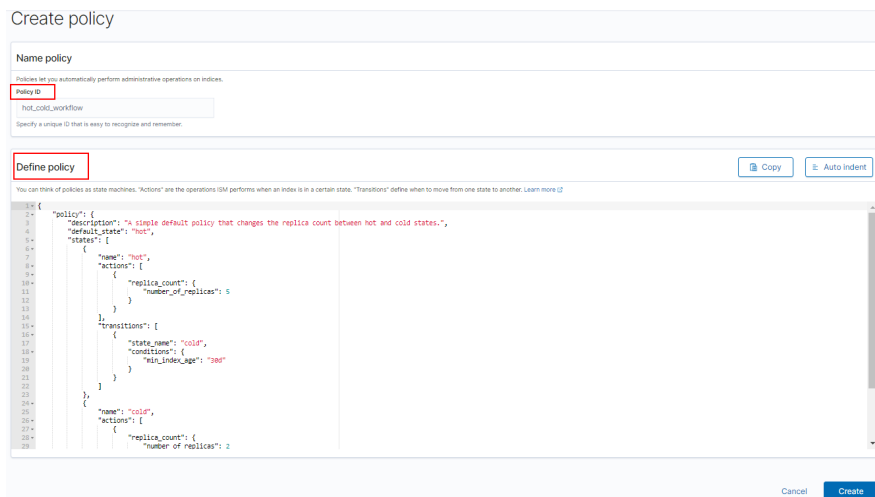
ISM更多功能请参见[索引管理官方介绍](#)。

创建的索引生命周期策略常用于以下场景：[通过索引生命周期实现时间序列数据滚动索引](#)

创建生命周期策略

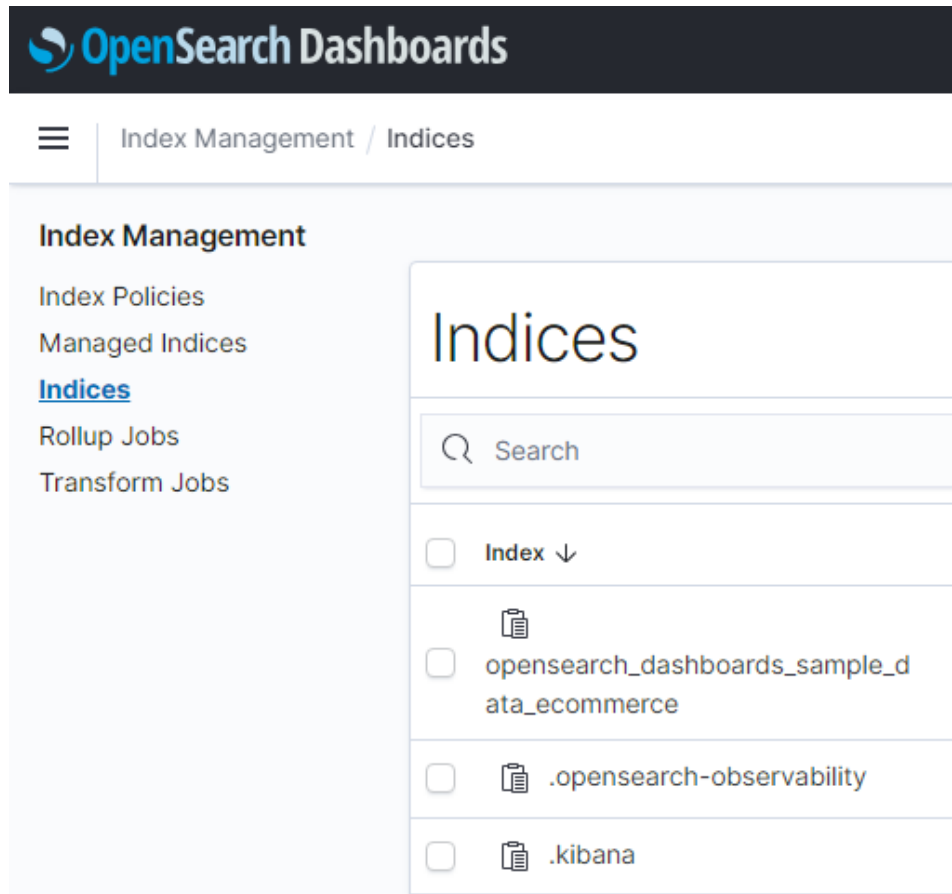
1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面选择需要登录的集群，单击“操作”列中的“Kibana”。
3. 登录OpenSearch Dashboard，在左侧选择“Index Management”，进入索引管理页面。
4. 右侧单击**Create policy**，创建索引策略。在Configuration method对话框中，选择“JSON editor”，单击“Continue”进入创建索引策略页面。
 - “Policy ID”：自定义策略名称。
 - “Define policy”：基于策略示例，配置自定义策略。

图 3-40 配置策略



- a. 在OpenSearch Dashboard的“Index Management”页面，选择Indices。

图 3-41 选择 Indices



- b. 在Indices列表中选择您要附加策略的一个或多个索引。
- c. 单击右上角的“Apply policy”，在弹窗中选择要关联的“Policy ID”。

图 3-42 添加引用策略

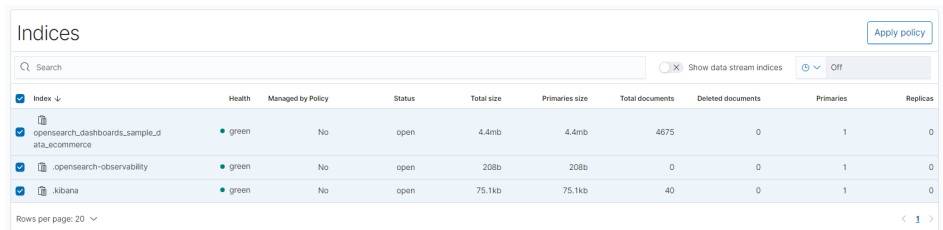
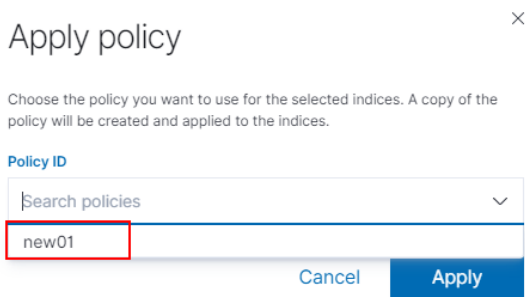


图 3-43 选择索引策略



- d. 单击“Apply”，完成索引关联策略。
当索引关联策略后，ISM会默认创建一个每5分钟运行一次的作业，用于执行策略操作、检查条件并将索引转换为不同的状态。

管理索引策略

1. 在OpenSearch Dashboard的“Index Management”页面，选择“Managed Indices”，进入索引策略管理列表。
此处会展示当前集群已配置的索引策略。
2. 在索引策略管理列表，管理策略。
 - 当索引的策略状态异常时，可以勾选状态异常的索引，单击“Retry policy”，可以重试策略。
 - 当需要更新索引的策略时，可以勾选索引，单击“Change policy”进入变更策略页面，更新策略后，单击“Change”变更策略。
 - 当需要解除索引的策略时，可以勾选索引，单击“Remove policy”，在弹窗确认后单击“Remove”改索引将被解除策略。

具体使用可参考[索引管理官方介绍](#)。

3.12.2 通过索引生命周期实现时间序列数据滚动索引

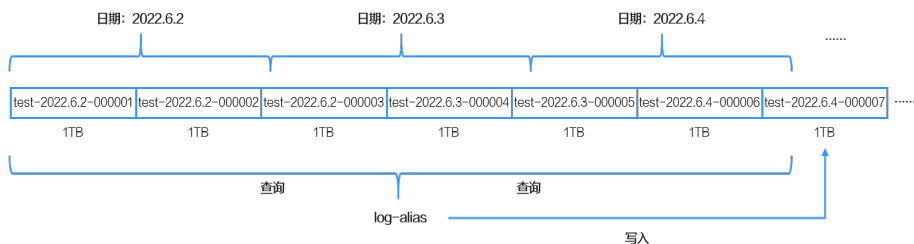
方案概述

对于时间序列数据，随着时间推移数据持续写入，索引会越来越大，通过生命周期管理来定期将数据滚动到新索引，并将历史老索引删除，实现自动滚动索引。

本案例通过配置生命周期策略，当索引的大小达到1TB或索引创建超过1天时，自动滚动生成新索引；当索引创建7天后，关闭数据副本；当索引创建30天后，删除该索引。

假设某个索引，其每天约产生2.4TB的数据，索引别名“log-alias”，其数据在Elasticsearch中的组织形态如下图所示。查询时，指向所有test开头的索引；写入时，指向最新的索引。

图 3-44 log-alias 的组织形态



📖 说明

索引的滚动条件1天是以索引的创建时间来计算的，并不是完整自然日区分的。

前提条件

- CSS集群处于可用状态。
- 使用OpenSearch集群。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表页面。
3. 在集群列表页面中，单击集群操作列的“Kibana”登录OpenSearch Dashboard 页面。
4. 在OpenSearch Dashboard的左侧导航中选择“Dev Tools”，进入命令执行页面。
5. 创建Rollover生命周期策略“rollover_workflow”。

策略定义：当索引的大小达到1TB或索引创建超过1天时，自动进行滚动；当索引创建7天后，关闭数据副本；当索引创建30天后，删除该索引。

```
PUT _plugins/_ism/policies/rollover_workflow
```

```
{
  "policy": {
    "description": "rollover test",
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [
          {
            "rollover": {
              "min_size": "1tb",
              "min_index_age": "1d"
            }
          }
        ],
        "transitions": [
          {
            "state_name": "warm",
            "conditions": {
              "min_index_age": "7d"
            }
          }
        ]
      }
    ],
    "name": "warm",
    "actions": [
      {
        "replica_count": {
          "number_of_replicas": 0
        }
      }
    ],
    "transitions": [
      {
        "state_name": "delete",
        "conditions": {
          "min_index_age": "30d"
        }
      }
    ]
  }
}
```

```

    ]
  },
  {
    "name": "delete",
    "actions": [
      {
        "delete": {}
      }
    ]
  }
],
"ism_template": {
  "index_patterns": [
    "test*"
  ],
  "priority": 100
}
}
}

```

当生命周期策略创建完成后，执行如下命令可以查询策略详情：

```
GET _plugins/_ism/policies/rollover_workflow
```

6. 新建索引模板 “template_test”。

模板定义：新建的所有 “test” 开头的索引自动关联上 Rollover 生命周期策略 “rollover_workflow”，并且 Rollover 时使用 “log_alias” 作为别名。

```

PUT _template/template_test
{
  "index_patterns": "test*",
  "settings": {
    "number_of_replicas": 1,
    "number_of_shards": 1,
    "index.opendistro.index_state_management.rollover_alias": "log_alias"
  },
  "mappings": {
    "properties": {
      "name": {
        "type": "text"
      }
    }
  }
}
}

```

表 3-84 参数说明

参数	说明
number_of_shards	索引分片数
number_of_replicas	索引分片副本数
opendistro.index_state_management.policy_id	生命周期的策略名
index.opendistro.index_state_management.rollover_alias	rollover的索引别名

当索引模板创建完成后，可以通过如下命令查询模板详情：

```
GET _template/template_test
```

7. 新建一个索引，指定 “aliases”，并配置 “is_write_index” 为 “true”。该索引会自动应用索引模板 “template_test”，并通过索引模板的配置与生命周期策略

“rollover_workflow”相关联，实现当索引的大小达到1TB或索引创建超过1天时，自动进行滚动；当索引创建7天后，关闭数据副本；当索引创建30天后，删除该索引。

如下索引是<test-{now/d}-000001>的URL编码，其创建时默认会带上当天的时间，例如当天日期是“2022.6.02”，创建出来的索引名称为

“test-2022.06.02-000001”。

```
PUT %3Ctest-%7Bnow%2Fd%7D-000001%3E
{
  "aliases": {
    "log_alias": {
      "is_write_index": true
    }
  }
}
```

8. 使用别名“log_alias”写入数据，且写入时“log_alias”始终指向最后一个索引。

```
POST log_alias/_bulk
{"index":{}}
{"name":"name1"}
{"index":{}}
{"name":"name2"}
{"index":{}}
{"name":"name3"}
{"index":{}}
{"name":"name4"}
{"index":{}}
{"name":"name5"}
{"index":{}}
{"name":"name6"}
```

9. 查询数据，确认数据是否实现滚动索引。

- 在索引创建一天后查看“test”开头的索引：

```
GET _cat/indices/test?s=i
```

正常情况下会显示至少有两个索引，如下所示：

```
green open test-<日期>-000001 r8ab5NX6T3Ox_hoGUanogQ 1 1 6 0 416b 208b
green open test-<日期>-000002 sfwkVgy8RSSEw7W-xYjM2Q 1 1 0 0 209b 209b
```

其中，“test-<日期>-000001”为7创建的索引，“test-<日期>-000002”为滚动生成的索引。

- 查询别名“log_alias”关联的索引情况：

```
GET _cat/aliases/log_alias?v
```

正常情况下会显示该别名指向多个索引：

alias	index	filter	routing.index	routing.search	is_write_index
log_alias	test-<日期>-000001	-	-	-	false
log_alias	test-<日期>-000002	-	-	-	true

3.13 OpenSearch 集群监控与日志管理

3.13.1 CES 中 OpenSearch 集群支持的监控指标

云监控服务CES支持实时监控云搜索服务集群的核心指标，方便用户掌握集群的指标信息，以便及时处理集群的异常状况。

功能说明

云监控服务CES支持实时监控云搜索服务集群的核心指标，方便用户掌握集群的指标信息，以便及时处理集群的异常状况。本文定义了云搜索服务上报云监控服务的集群和

节点监控指标的命名空间，监控指标列表和维度定义。用户可以通过云监控服务提供的管理控制台或API接口来检索云搜索服务产生的监控指标和告警信息。

命名空间

SYS.ES

集群监控指标列表

- 监控的**指标ID**、**指标名称**、**指标含义**以及**取值范围**参见表3-85。
- 监控的**测量对象**：CSS集群（本文介绍的是Elasticsearch集群和OpenSearch集群的监控指标列表）
- 监控的**监控周期**（原始指标）：1分钟

说明

累计值：从节点启动时开始叠加数值，当节点重启后清零重新累计。

表 3-85 云搜索服务支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
status	集群健康状态	该指标用于统计测量监控对象的状态。	0,1,2,3 <ul style="list-style-type: none">• 0：集群是100%可用的。• 1：数据是完整的，部分副本缺失。高可用性在某种程度上弱化，存在风险，请及时关注集群情况。• 2：数据缺失，集群使用时将出现异常。• 3：没有获取到集群状态。	CSS集群	1分钟
disk_util	磁盘使用率	该指标用于统计测量对象的磁盘使用率。 单位：百分比	0-100%	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_jvm_heap_usage	最大JVM堆使用率	CSS集群中各个节点的JVM堆使用率的最大值。 单位：百分比。	0-100%	CSS集群	1分钟
max_jvm_young_gc_time	最大JVM Young GC耗时	CSS集群中各个节点的JVM Young GC耗时累计值的最大值。 单位：ms	≥ 0 ms	CSS集群	1分钟
max_jvm_young_gc_count	最大JVM Young GC次数	CSS集群中各个节点的JVM Young GC次数累计值的最大值。	≥ 0	CSS集群	1分钟
max_jvm_old_gc_time	最大JVM Old GC耗时	CSS集群中各个节点的JVM Old GC耗时累计值的最大值。 单位：ms	≥ 0 ms	CSS集群	1分钟
max_jvm_old_gc_count	最大JVM Old GC次数	CSS集群中各个节点的JVM Old GC次数累计值的最大值。	≥ 0	CSS集群	1分钟
total_fs_size	文件系统总大小	CSS集群的文件系统总大小。 单位：byte	≥ 0 bytes	CSS集群	1分钟
free_fs_size	文件系统可用大小	CSS集群的文件系统可用大小。 单位：byte	≥ 0 bytes	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_cpu_usage	最大CPU利用率	CSS集群中各个节点的CPU利用率的 最大值。 单位: 百分比	0-100%	CSS集群	1分钟
max_cpu_time_of_jvm_process	最大JVM进程使用的CPU时间	CSS集群中各个节点JVM进程使用CPU的时间 累计值的最大值。 单位: ms	≥ 0 ms	CSS集群	1分钟
max_virtual_memory_size_of_jvm_process	最大JVM进程使用的虚拟内存大小	CSS集群中各个节点JVM进程可使用的 虚拟内存大小的最大值。 单位: byte	≥ 0 bytes	CSS集群	1分钟
max_current_opened_http_count	最大当前打开的HTTP连接数	CSS集群中各个节点打开且尚未关闭的 HTTP连接数的最大值。	≥ 0	CSS集群	1分钟
max_total_opened_http_count	最大全部打开的HTTP连接数	CSS集群中各个节点打开过的HTTP连接 数累计值的最大值。	≥ 0	CSS集群	1分钟
indices_count	索引数量	CSS集群的索引数量。	≥ 0	CSS集群	1分钟
total_shards_count	分片数量	CSS集群的分片数量。	≥ 0	CSS集群	1分钟
primary_shards_count	主分片数量	CSS集群的主分片数量。	≥ 0	CSS集群	1分钟
docs_count	文档数量	CSS集群的文档数量。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
docs_deleted_count	被删除的文档数量	CSS集群的被删除的文档数量。	≥ 0	CSS集群	1分钟
nodes_count	节点数量	CSS集群的节点数量。	≥ 0	CSS集群	1分钟
data_nodes_count	数据节点数量	CSS集群的数据节点数量。	≥ 0	CSS集群	1分钟
coordinating_nodes_count	协调节点数量	CSS集群的协调节点数量。	≥ 0	CSS集群	1分钟
master_nodes_count	Master节点数量	CSS集群的Master节点数量。	≥ 0	CSS集群	1分钟
ingest_nodes_count	Client节点数量	CSS集群的Client节点数量。	≥ 0	CSS集群	1分钟
max_load_average	最大节点Load值	CSS集群中各个节点在操作系统中1分钟平均排队任务数的最大值。	≥ 0	CSS集群	1分钟
avg_cpu_usage	平均CPU使用率	CSS集群中各节点CPU利用率的平均值。 单位：百分比	0-100%	CSS集群	1分钟
avg_load_average	平均节点Load值	CSS集群中各节点在操作系统中1分钟平均排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_jvm_heap_usage	平均JVM堆使用率	CSS集群中各节点JVM堆内存使用率的平均值。 单位：百分比	0-100%	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_open_file_descriptors	已打开的最大文件描述符数	CSS集群中各个节点已打开的文件描述符数的最大值。	≥ 0	CSS集群	1分钟
avg_open_file_descriptors	已打开的平均文件描述符数	CSS集群中各节点已打开的文件描述符数的平均值。	≥ 0	CSS集群	1分钟
sum_max_file_descriptors	最大允许的文件描述符数	CSS集群中各节点最大允许的文件描述符数之和。	≥ 0	CSS集群	1分钟
sum_open_file_descriptors	已打开的文件描述符数	CSS集群中各节点已打开的文件描述符数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_write_queue	Write队列中总排队任务数	写入线程池中的排队任务数。	≥ 0	CSS集群	1分钟
sum_thread_pool_search_queue	Search队列中总排队任务数	CSS集群中各节点在搜索线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_force_merge_queue	ForceMerge队列中总排队任务数	CSS集群中各节点在强制合并线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_write_rejected	Write队列中总的已拒绝任务数	CSS集群中各节点在写入线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_search_rejected	Search队列中总的已拒绝任务数	CSS集群中各节点在搜索线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_force_merge_rejected	ForceMerge队列中总的已拒绝任务数	CSS集群中各节点在强制合并线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_search_queue	Search队列中最大排队任务数	CSS集群中各个节点在搜索线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
max_thread_pool_force_merge_queue	ForceMerge队列中最大排队任务数	CSS集群中各个节点在强制合并线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_write_threads	Write线程池总大小	CSS集群中各节点写入线程池的大小之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_search_threads	Search线程池总大小	CSS集群中各节点搜索线程池的大小之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_force_merge_threads	ForceMerge线程池总大小	CSS集群中各节点强制合并线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_write_queue	Write队列中平均排队任务数	CSS集群中各节点在写入线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_search_queue	Search队列中平均排队任务数	CSS集群中各节点在搜索线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_thread_pool_force_merge_queue	ForceMerge队列中平均排队任务数	CSS集群中各节点在强制合并线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_search_threads	Search线程池平均大小	CSS集群中各节点搜索线程池的大小的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_write_threads	Write线程池平均大小	CSS集群中各节点写入线程池的大小的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_force_merge_threads	ForceMerge线程池平均大小	CSS集群中各节点强制合并线程池的大小的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_write_rejected	Write队列中平均已拒绝任务数	CSS集群中各节点写入线程池中的已拒绝任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_flush_queue	Flush队列中总排队任务数	CSS集群中各个节点在Flush线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_flush_rejected	Flush队列中总的已拒绝任务数	CSS集群中各节点在Flush线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_flush_queue	Flush队列中最大排队任务数	CSS集群中各个节点在Flush线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_flush_threads	Flush线程池总大小	CSS集群中各节点Flush线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_flush_queue	Flush队列中平均排队任务数	CSS集群中各节点在Flush线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_flush_threads	Flush线程池平均大小	CSS集群中各节点在Flush线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_generic_queue	Generic队列中总排队任务数	CSS集群中各个节点在Generic线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_generic_rejecte	Generic队列中总的已拒绝任务数	CSS集群中各节点在Generic线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_generic_queue	Generic队列中最大排队任务数	CSS集群中各个节点在Generic线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_generic_threads	Generic线程池总大小	CSS集群中各节点Generic线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_generic_queue	Generic队列中平均排队任务数	CSS集群中各节点在Generic线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_thread_pool_generic_threads	Generic线程池平均大小	CSS集群中各节点在Generic线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_management_queue	Management队列中总排队任务数	CSS集群中各个节点在Management线程池中的排队任务数只和。	≥ 0	CSS集群	1分钟
sum_thread_pool_management_rejected	Management队列中总的已拒绝任务数	CSS集群中各节点在Management线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_management_queue	Management队列中最大排队任务数	CSS集群中各个节点在Management线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_management_threads	Management线程池总大小	CSS集群中各节点Management线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_management_queue	Management队列中平均排队任务数	CSS集群中各节点在Management线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_management_threads	Management线程池平均大小	CSS集群中各节点在Management线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_refresh_queue	Refresh队列中总排队任务数	CSS集群中各个节点在Refresh线程池中的排队任务数只和。	≥ 0	CSS集群	1分钟
sum_thread_pool_refresh_rejected	Refresh队列中总的已拒绝任务数	CSS集群中各个节点在Refresh线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_refresh_queue	Refresh队列中最大排队任务数	CSS集群中各个节点在Refresh线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_refresh_threads	Refresh线程池总大小	CSS集群中各个节点Refresh线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_refresh_queue	Refresh队列中平均排队任务数	CSS集群中各个节点在Refresh线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_refresh_threads	Refresh线程池平均大小	CSS集群中各个节点在Refresh线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_searcher_queue	OBS Searcher队列中总排队任务数	CSS集群中各个节点在OBS Searcher线程池中的排队任务数只和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_thread_pool_obs_searcher_rejected	OBS Searcher队列中总的已拒绝任务数	CSS集群中各节点在OBS Searcher线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_obs_searcher_queue	OBS Searcher队列中最大排队任务数	CSS集群中各个节点在OBS Searcher线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_searcher_threads	OBS Searcher线程池总大小	CSS集群中各节点OBS Searcher线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_searcher_queue	OBS Searcher队列中平均排队任务数	CSS集群中各节点在OBS Searcher线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_searcher_threads	OBS Searcher线程池平均大小	CSS集群中各节点在OBS Searcher线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_queue	OBS队列中总排队任务数	CSS集群中各个节点在OBS线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_rejected	OBS队列中总的已拒绝任务数	CSS集群中各节点在OBS线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_thread_pool_obs_queue	OBS队列中最大排队任务数	CSS集群中各个节点在OBS线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_threads	OBS线程池总大小	CSS集群中各节点OBS线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_queue	OBS队列中平均排队任务数	CSS集群中各节点在OBS线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_threads	OBS线程池平均大小	CSS集群中各节点在OBS线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_upload_queue	OBS Upload队列中总排队任务数	CSS集群中各个节点在OBS Upload线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_upload_rejected	OBS Upload队列中总的已拒绝任务数	CSS集群中各节点在OBS Upload线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_obs_upload_queue	OBS Upload队列中最大排队任务数	CSS集群中各个节点在OBS Upload线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_upload_threads	OBS Upload线程池总大小	CSS集群中各节点OBS Upload线程池的大小之和。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_thread_pool_obs_upload_queue	OBS Upload队列中平均排队任务数	CSS集群中各节点在OBS Upload线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_uploads_threads	OBS Upload线程池平均大小	CSS集群中各节点在OBS Upload线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_download_queue	OBS Download队列中总排队任务数	CSS集群中各个节点在OBS Download线程池中的排队任务数之和。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_download_rejected	OBS Download队列中总的已拒绝任务数	CSS集群中各节点在OBS Download线程池中的已拒绝任务数之和。	≥ 0	CSS集群	1分钟
max_thread_pool_obs_download_queue	OBS Download队列中最大排队任务数	CSS集群中各个节点在OBS Download线程池中的排队任务数的最大值。	≥ 0	CSS集群	1分钟
sum_thread_pool_obs_download_threads	OBS Download线程池总大小	CSS集群中各节点OBS Download线程池的大小之和。	≥ 0	CSS集群	1分钟
avg_thread_pool_obs_download_queue	OBS Download队列中平均排队任务数	CSS集群中各节点在OBS Download线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_thread_pool_obs_download_threads	OBS Download 线程池平均大小	CSS集群中各节点在OBS Download线程池中的排队任务数的平均值。	≥ 0	CSS集群	1分钟
min_free_fs_size	最小可用存储空间	CSS集群中各个节点可用存储空间的最小值。 单位: byte	≥ 0 bytes	CSS集群	1分钟
avg_jvm_ol_d_gc_count	JVM老年代平均GC次数	CSS集群中各个节点“老年代”垃圾回收的运行次数的累计值的平均值。	≥ 0	CSS集群	1分钟
avg_jvm_ol_d_gc_time	JVM老年代平均GC时间	CSS集群中各个节点执行“老年代”垃圾回收所花费的时间累计值的平均值。 单位: ms	≥ 0 ms	CSS集群	1分钟
avg_jvm_young_gc_count	JVM年轻代平均GC次数	CSS集群中各个节点“年轻代”垃圾回收的运行次数的累计值的平均值。	≥ 0	CSS集群	1分钟
avg_jvm_young_gc_time	JVM年轻代平均GC时间	CSS集群中各个节点执行“年轻代”垃圾回收所花费的时间累计值的平均值。 单位: ms	≥ 0 ms	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
avg_max_file_descriptors	最大允许的文件描述符数-平均值	CSS集群中各节点最大允许的文件描述符数的平均值。	≥ 0	CSS集群	1分钟
avg_mem_free_in_bytes	平均可用内存空间	CSS集群中各节点未使用的内存容量的平均值。 单位: byte	≥ 0 bytes	CSS集群	1分钟
avg_mem_free_percent	平均可用内存比例	CSS集群中各节点未使用的内存比例的平均值。 单位: 百分比	0-100%	CSS集群	1分钟
avg_mem_used_in_bytes	平均已用内存空间	CSS集群中各节点已使用的内存容量的平均值。 单位: byte	≥ 0 bytes	CSS集群	1分钟
avg_mem_used_percent	平均已用内存比例	CSS集群中各节点已使用的内存比例的平均值。 单位: 百分比	0-100%	CSS集群	1分钟
max_mem_free_in_bytes	最大可用内存空间	CSS集群中各个节点未使用的内存容量的最大值。 单位: byte	≥ 0 bytes	CSS集群	1分钟
max_mem_free_percent	最大可用内存比例	CSS集群中各个节点未使用的内存比例的最大值。 单位: 百分比	0-100%	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
max_mem_used_in_bytes	最大已用内存空间	CSS集群中各个节点已使用的内存容量的最大值。 单位: byte	≥ 0 bytes	CSS集群	1分钟
max_mem_used_percent	最大已用内存比例	CSS集群中各个节点已使用的内存比例的最大值。 单位: 百分比	0-100%	CSS集群	1分钟
sum_jvm_oldest_gc_count	JVM老年代总GC次数	CSS集群中各个节点“老年代”垃圾回收的运行次数的累计值之和。	≥ 0	CSS集群	1分钟
sum_jvm_oldest_gc_time	JVM老年代总GC时间	CSS集群中各个节点执行“老年代”垃圾回收所花费的时间累计值之和。 单位: ms	≥ 0ms	CSS集群	1分钟
sum_jvm_young_gc_count	JVM年轻代总GC次数	CSS集群中各个节点“年轻代”垃圾回收的运行次数的累计值之和。	≥ 0	CSS集群	1分钟
sum_jvm_young_gc_time	JVM年轻代总GC时间	CSS集群中各个节点执行“年轻代”垃圾回收所花费的时间累计值之和。 单位: ms	≥ 0 ms	CSS集群	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
sum_current_opened_http_count	当前已打开HTTP连接数	CSS集群中各个节点打开且尚未关闭的HTTP连接数之和。	≥ 0	CSS集群	1分钟
sum_total_opened_http_count	历史已打开HTTP连接数	CSS集群中各个节点打开过的HTTP连接数累计值之和。	≥ 0	CSS集群	1分钟
IndexingLatency	平均索引延迟	分片完成索引操作所需的平均时间。 单位: ms	≥ 0 ms	CSS集群	1分钟
IndexingRate	平均索引速率	入库TPS, 集群每秒平均索引操作数。	≥ 0	CSS集群	1分钟
SearchLatency	平均查询延迟	分片完成搜索操作所需的平均时间。 单位: ms。	≥ 0 ms	CSS集群	1分钟
SearchRate	平均查询速率	查询QPS, 集群每秒平均查询操作数。	≥ 0	CSS集群	1分钟
task_max_running_time	最大Task运行时长	该指标为集群运行的所有task中, 运行时长最长的task的耗时	≥ 0 ms	CSS集群	1分钟
number_of_pending_tasks	Pending Task排队任务数	CSS集群中待Master处理的task的排队任务数。	≥ 0	CSS集群	1分钟

节点监控指标列表

- 监控的**指标ID**、**指标名称**、**指标含义**以及**取值范围**参见表3-86。
- 监控的**测量对象**：CSS集群 - 云服务节点
- 监控的**监控周期**（原始指标）：1分钟

📖 说明

累计值：从节点启动时开始叠加数值，当节点重启后清零重新累计。

表 3-86 云搜索服务节点支持的监控指标

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期（原始指标）
jvm_heap_usage	JVM堆使用率	节点JVM堆内存使用率。 单位：百分比	0-100 %	CSS集群 - 云服务节点	1分钟
cpu_usage	CPU利用率	CPU利用率。 单位：百分比	0-100 %	CSS集群 - 云服务节点	1分钟
load_average	节点Load值	操作系统中1分钟平均排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
open_file_descriptors	已打开的文件描述符数	节点已打开的文件描述符数。	≥ 0	CSS集群 - 云服务节点	1分钟
max_file_descriptors	最大允许的文件描述符数	最大允许的文件描述符数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_write_queue	Write队列中总排队任务数	写入线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_search_queue	Search队列中总排队任务数	搜索线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_force_merge_queue	ForceMerge队列中总排队任务数	强制合并线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
thread_pool_write_rejected	Write队列中总的已拒绝任务数	写入线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_search_rejected	Search队列中总的已拒绝任务数	搜索线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_force_merge_rejected	ForceMerge队列中总的已拒绝任务数	强制合并线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_write_threads	Write线程池总大小	写入线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_search_threads	Search线程池总大小	搜索线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_force_merge_threads	ForceMerge线程池总大小	强制合并线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_flush_queue	Flush队列中总排队任务数	Flush线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_flush_rejected	Flush队列中总的已拒绝任务数	Flush线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_flush_threads	Flush线程池总大小	Flush线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_generic_queue	Generic队列中总排队任务数	Generic线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
thread_pool_generic_rejected	Generic队列中总的已拒绝任务数	Generic线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_generic_threads	Generic线程池总大小	Generic线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_management_queue	Management队列中总排队任务数	Management线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_management_rejected	Management队列中总的已拒绝任务数	Management线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_management_threads	Management线程池总大小	Management线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_refresh_queue	Refresh队列中总排队任务数	Refresh线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_refresh_rejected	Refresh队列中总的已拒绝任务数	Refresh线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_refresh_threads	Refresh线程池总大小	Refresh线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_searcher_queue	OBS Searcher队列中总排队任务数	OBS Searcher线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_searcher_rejected	OBS Searcher队列中总的已拒绝任务数	OBS Searcher线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
thread_pool_obs_searcher_threads	OBS Searcher线程池总大小	OBS Searcher线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_queue	OBS队列中总排队任务数	OBS线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_rejected	OBS队列中总的已拒绝任务数	OBS线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_threads	OBS线程池总大小	OBS线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_upload_queue	OBS Upload队列中总排队任务数	OBS Upload线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_upload_rejected	OBS Upload队列中总的已拒绝任务数	OBS Upload线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_upload_threads	OBS Upload线程池总大小	OBS Upload线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_download_queue	OBS Download队列中总排队任务数	OBS Download线程池中的排队任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_download_rejected	OBS Download队列中总的已拒绝任务数	OBS Download线程池中的已拒绝任务数。	≥ 0	CSS集群 - 云服务节点	1分钟
thread_pool_obs_download_threads	OBS Download线程池总大小	OBS Download线程池的大小。	≥ 0	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
free_fs_size	文件系统可用大小	CSS集群的文件系统可用大小。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟
total_fs_size	文件系统总大小	CSS集群的文件系统总大小。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟
jvm_old_gc_count	JVM老年代总GC次数	“老年代”垃圾回收的运行次数。	≥ 0	CSS集群 - 云服务节点	1分钟
jvm_old_gc_time	JVM老年代总GC时间	执行“老年代”垃圾回收所花费的时间。 单位: ms	≥ 0 ms	CSS集群 - 云服务节点	1分钟
jvm_young_gc_count	JVM年轻代总GC次数	“年轻代”垃圾回收的运行次数。	≥ 0	CSS集群 - 云服务节点	1分钟
jvm_young_gc_time	JVM年轻代GC时间	执行“年轻代”垃圾回收所花费的时间。 单位: ms	≥ 0 ms	CSS集群 - 云服务节点	1分钟
mem_free_in_bytes	可用内存空间	节点未使用的内存容量。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟
mem_free_percent	可用内存比例	节点未使用的内存比例。	≥ 0	CSS集群 - 云服务节点	1分钟
mem_used_in_bytes	已用内存空间	节点已使用的内存容量。 单位: byte	≥ 0 bytes	CSS集群 - 云服务节点	1分钟

指标ID	指标名称	指标含义	取值范围	测量对象	监控周期 (原始指标)
current_opened_http_count	当前已打开HTTP连接数	节点当前打开的HTTP连接数。	≥ 0	CSS集群 - 云服务节点	1分钟
total_opened_http_count	全部打开的HTTP连接数	节点全部打开的HTTP连接数。	≥ 0	CSS集群 - 云服务节点	1分钟

维度

表 3-87 维度说明

Key	Value
cluster_id	CSS集群
cluster_id,instance_id	CSS集群节点

3.13.2 使用 CES 监控 OpenSearch 集群

云搜索服务支持通过云监控服务CES对已创建成功的集群进行日常监控。配置集群监控后，就可以在CES管理控制台直观查看集群的监控指标数据。

配置集群监控的操作流程：

1. **配置告警规则**：根据实际业务需要对监控指标设置自定义告警规则，当监控指标超过设置的阈值时，会以邮箱、HTTP、HTTPS等方式通知您。
2. **配置监控对象**：为集群或集群中某个节点配置监控指标。
3. **查看监控指标**：您可以选择不同的监控时间周期，查看监控指标数据变化情况。

前提条件

- 集群处于“可用”或“处理中”状态。
- 集群正常运行时长大于10分钟。

推荐配置的监控指标

- 监控集群的**cpu、jvm使用情况**，推荐重点配置如下监控指标：平均JVM堆使用率、最大JVM堆使用率、平均CPU使用率、最大CPU利用率。
- 监控集群的**写入、查询延迟和吞吐量情况**，推荐重点配置如下监控指标：平均索引延迟、平均索引速率、平均查询延迟、平均查询速率。

- 监控集群的**写入、查询的排队队列和拒绝情况**，推荐重点配置如下监控指标：Write队列中总排队任务数、Search队列中总排队任务数、Write队列中总的已拒绝任务数、Search队列中总的已拒绝任务数。

配置告警规则

1. 登录云监控服务CES管理控制台。
2. 左侧导航栏选择“告警 > 告警规则”，进入告警规则列表页面。
3. 在“资源类型”列，筛选“云搜索服务”，查看是否有满足要求的告警规则。

图 3-45 查看告警规则



<input type="checkbox"/>	名称/ID	资源类型 	监控对象
<input type="checkbox"/>	alarm- al1654480875465a...	云搜索服务	CSS集群 指定资源

如果没有，请参考云监控服务CES的[创建告警规则和通知](#)，新建CSS服务的告警规则。其中，“资源类型”和“维度”参数的填写说明请参见[表3-88](#)，其他参数可以根据CES服务的参数说明自定义。

表 3-88 告警内容的配置说明

参数	参数解释	配置说明
资源类型	配置告警规则监控的服务名称。	选择 云搜索服务 。
维度	用于指定告警规则对应指标的维度名称。	CSS支持2个维度，根据实际需要选择维度。 <ul style="list-style-type: none">• CSS集群：以集群维度指定告警规则。• CSS集群 - 云服务节点：以集群中的某个节点维度指定告警规则。

配置监控对象

1. 参考云监控服务CES的[创建监控面板](#)，创建一个监控面板。如果已有监控面板，可以跳过该步骤。
2. 参考云监控服务CES的[添加监控视图](#)，添加CSS监控视图。

其中，“资源类型”和“维度”参数的填写说明请参见[表3-89](#)，其他参数可以根据CES服务的参数说明自定义。

表 3-89 监控视图的配置说明

参数	参数解释	配置说明
资源类型	添加监控视图的服务名称。	选择 云搜索服务 。

参数	参数解释	配置说明
维度	指定监控的维度名称。	CSS支持2个维度，根据实际需要选择维度。 <ul style="list-style-type: none">● CSS集群：以集群维度监控。● CSS集群 - 云服务节点：以集群中的某个节点维度监控。

查看监控指标

1. 登录云搜索服务管理控制台。
2. 选择目标集群，单击操作列“监控信息”查看监控指标。

图 3-46 查看监控信息

名称ID	集群状态	任务状态	版本	创建时间	企业项目	内网访问地址	计费模式	操作
	可用	-	7.10.2 elasticsearch	2024/03/20 11:32:14 ...	default		按需计费	Kibana 监控信息 更多

3. 选择待查看的时间段页签。
4. 查看监控指标数据。

3.13.3 设置 OpenSearch 集群 SMN 告警通知

场景描述

云搜索服务默认安装了开源的OpenSearch告警插件（opensearch-alerting），用于提供数据满足特定条件时的通知功能。该插件包含Alerts、Monitors、Destinations三个组件，CSS服务在Destinations组件中适配了消息通知服务SMN，且仅支持通过SMN服务作为目的端（Destination）进行告警消息发送。

本文主要介绍如何在OpenSearch Dashboards中设置OpenSearch集群的SMN告警功能。

说明

开源OpenSearch告警插件OpenSearch Alerting的官方指导文档请参见：[Alerting - OpenSearch Documentation](#)。

约束与限制

当前1.3.6版本的OpenSearch集群默认安装了开源的OpenSearch告警插件（opensearch-alerting）。

授权使用 SMN 服务

1. 通过CSS服务的管理员账号登录云搜索服务管理控制台。
2. 在左侧导航栏选择“服务授权”。
3. 在服务授权页面，单击“创建SMN委托”，在弹窗中确认委托创建成功。
 - 如果已经创建过委托，则右上角会提示“css_smn_agency exist, no need to created.”。
 - 如果无创建权限，则右上角会提示“当前用户没有操作权限，请通过IAM检查账户权限！”，请确认该管理员账号是否配置了IAM使用权限。

前提条件

- 已经完成SMN服务授权，操作指导请参见[授权使用SMN服务](#)。
- 在SMN服务管理控制台完成了主题创建，具体操作请参见[创建主题](#)。

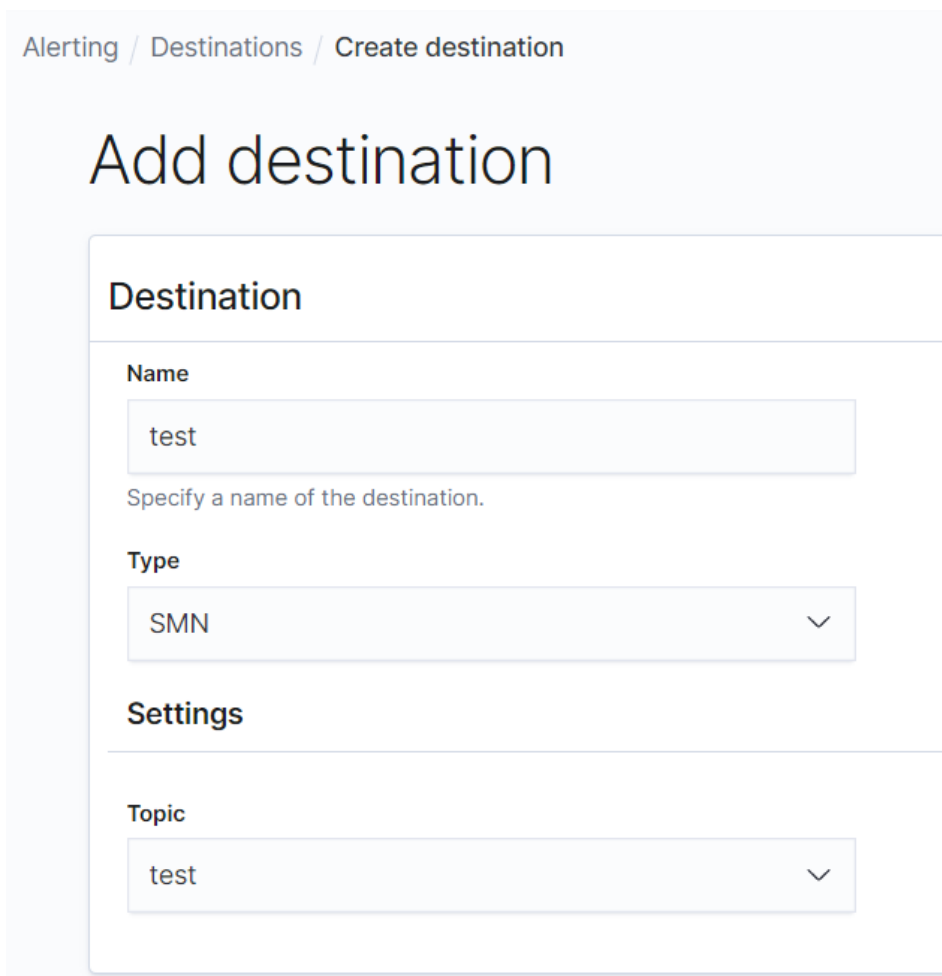
操作步骤

1. 登录云搜索服务管理控制台。
2. 在“集群管理 > OpenSearch”页面选择需要登录的集群，单击操作列中的“Kibana”登录OpenSearch Dashboards。
3. 在Dashboards页面，选择左侧导航栏的“菜单 > Alerting”。
4. 新建一个SMN目的端用于发送告警消息。
 - a. 在“Alerting”页面选择“Destinations”页签，单击“Add destination”配置目的端信息。

表 3-90 Destinations 参数说明

参数名称	说明
Name	自定义目的端名称。
Type	保持默认值“SMN”。
Topic	选择前提条件中创建好的用于发送告警消息的SMN主题，具体操作请参见 创建主题 。

图 3-47 Add destination



- b. 单击“Create”返回Destinations列表，在列表可见新建的SMN目的端，表示创建完成。

图 3-48 Destinations 列表



5. 新建一个监控，并配置告警的触发条件和监控频率。
 - a. 在“Alerting”页面选择“Monitors”页签，单击“Create monitor”配置监控信息。

表 3-91 Monitors 参数说明

参数名称	说明
Monitor details	
Monitor name	自定义一个监控名称。

参数名称	说明
Monitor type	Monitor类型：Per query monitor(普通监控)、Per bucket monitor(聚合桶监控)、Per cluster metrics monitor(集群指标监控)。
Monitor defining method	选择一种方式定义监控，推荐使用“Extraction query editor”。 <ul style="list-style-type: none">• Visual editor：可视化的query语句• Extraction query editor：具体的query语句• Anomaly detector：异常检测插件的监测 选择不同的“Monitor type”，所支持的监控方式不同，请以实际界面为准。
Detector	当“Monitor defining method”选择“Anomaly detector”时，请选择异常检测任务。
Frequency	选择监控频率，并设置监控频率的间隔信息。 <ul style="list-style-type: none">• By interval：按时间间隔• Daily：按每天• Weekly：按每周• Monthly：按每月• Custom cron expression：按cron脚本规划
Data source	
Index	当“Monitor defining method”选择“Visual editor”或“Extraction query editor”方式定义监控时，需要选择监控的索引。
Time field	当“Monitor defining method”选择“Visual editor”方式定义监控时，需要选择一个字段作为时间字段，来定义count等计数参数。
Query	
Metrics	当“Monitor defining method”选择“Visual editor”方式定义监控时，需要设置提取统计数据的指标范围。
Time range for the last	当“Monitor defining method”选择“Visual editor”方式定义监控时，需要设置插件应监控的时间范围。
Data filter	当“Monitor defining method”选择“Visual editor”方式定义监控时，需要设置过滤器进行数据检索。
Group by	当“Monitor defining method”选择“Visual editor”方式定义监控时，需要指定一个字段，使其每一个值都会触发告警。

参数名称	说明
Define extraction query	当“Monitor defining method”选择“Extraction query editor”方式定义监控时，需要输入query语句定义监控。
Request type	当“Monitor type”选择“Per cluster metrics monitor”时，需要指定请求类型来监控集群指标，例如运行状况、CPU使用率等。

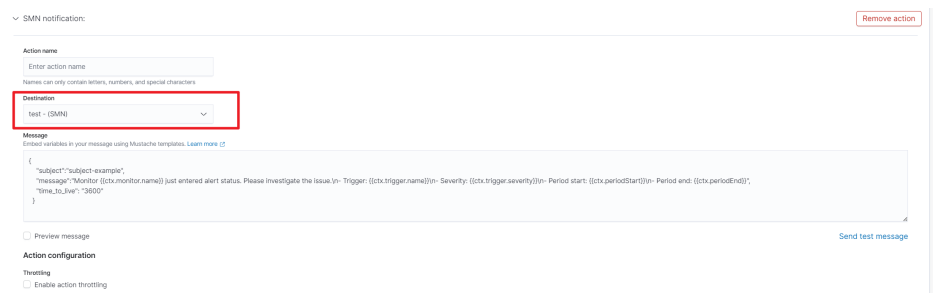
- b. 单击“Add trigger”添加Triggers指定告警触发条件及告警时应触发的动作。
- c. 在“Triggers”设置触发告警的灵敏度和目的端的消息发布。

表 3-92 Trigger 参数说明

参数名称	说明
Trigger name	自定义一个触发器名称。
Severity level	定义触发器的敏感度，选择触发多少条告警之后发送告警消息，“1”为最高敏感度。
Trigger condition	定义触发条件。满足触发条件就会触发告警。
Action name	自定义一个触发器动作名称。
Destination	选择4新建的SMN目的端。
Message	设置SMN主题发送的body，定义消息发布的文本。默认定义了目的端是Email类型时的主题和正文，详细参数说明请参见 消息发布 。
Perform action	当“Monitor type”选择“Per bucket monitor”时，需要设置告警是否组合发送，可选择 <ul style="list-style-type: none"> ● Per execution：多个指标触发条件后组合发送告警。 ● Per alert：多个指标触发条件后各自发送告警。
Actionable alerts	当“Monitor type”选择“Per bucket monitor”时，选择“Per alert”，需要设置触发条件后可执行的告警。 <ul style="list-style-type: none"> ● De-duplicated：已触发的告警，OpenSearch保留现有告警，以防止多次创建相同告警。 ● New：新创建的告警。 ● Completed：不持续触发的告警。

参数名称	说明
Throttling	<p>设置消息发送频率用来限制在指定时间段内收到的通知消息数量。</p> <p>例如设置10分钟，在接下来的10分钟内，即使触发条件满足多次，SMN仅发送1次告警通知。超过10分钟后，如果触发告警条件，SMN才会再次发送告警通知。</p>

图 3-49 设置触发器动作的目的端



- d. 单击“Send test message”，如图3-51所示SMN主题订阅者成功收到邮件消息，则表示触发器设置成功。

图 3-50 发送验证信息

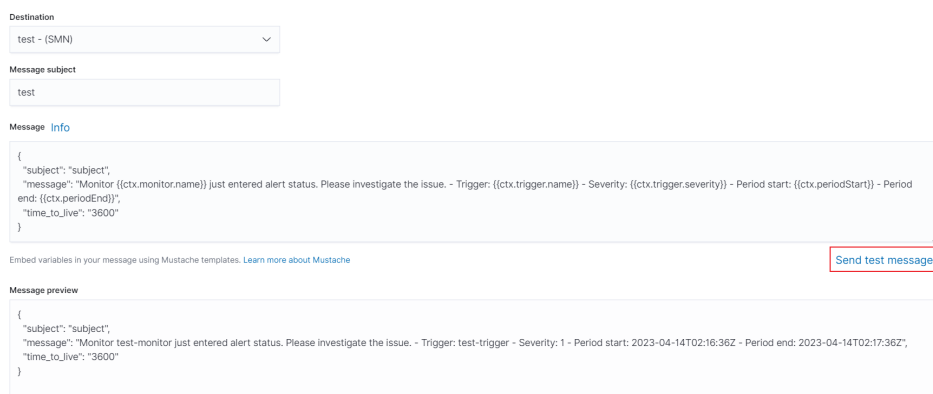
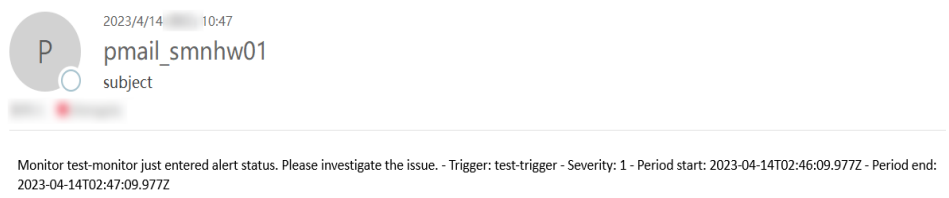


图 3-51 成功收到邮件通知



- e. 单击“Create”返回Monitor详情页，即成功创建检测器。

3.13.4 智能检测 OpenSearch 集群风险

CSS服务提供智能运维功能，支持检测集群潜在风险，并给出风险处理的建议。

场景描述

集群的智能运维支持以下功能：

- **启动检测任务**：通过启动检测任务，触发系统智能诊断集群的健康状况。
- **查看集群风险项**：检测任务完成后，可以查看集群存在的风险项详情，根据风险建议及时处理集群存在的风险。
- **删除检测任务**：当不需要历史所创建的检测任务时，可删除检测任务。删除检测任务后，系统将删除检测任务所对应的所有诊断信息。

智能运维的检查项包含如下项目。

- 检查当前时刻集群的健康状态，其中red表示有主分片未分配，yellow表示有副分片未分配，green表示所有分片均已分配。
- 检测集群节点数与AZ(可用区)数，判定Elasticsearch分布式集群的高可用性。
- 检测集群索引是否开启副本，未设置副本的索引在节点发生故障后可能导致索引不可用，本地盘集群未设置副本会有数据丢失的风险。
- 检测集群是否存在kibana索引冲突。
- 检测节点磁盘用量百分比，节点磁盘存储过大可能导致节点无法分配新索引分片并影响集群性能。
- 检测集群数据节点/冷数据节点的存储用量是否均衡，不均衡的存储用量分布可能导致集群负载不均，读写延迟增加。
- 连续5分钟检测当前集群是否有节点脱离或不可用。
- 检测节点分片数量是否过多，单节点分片数量过多会消耗大量节点资源，读写请求时延增加，集群元数据更新缓慢等。
- 检测所有分片大小，分片太大可能导致查询性能下降，节点内存消耗增加，影响扩缩容以及节点故障时分片的恢复速率等。
- 检测当前集群是否存在可升级的版本。
- 检测集群7天内是否有快照备份失败，或7天内无快照备份记录。

前提条件

当智能检测任务完成后需要发送SMN告警通知时，需要先在SMN服务创建主题，操作指导请参见[创建主题](#)。同时，需要参考[授权使用SMN服务](#)完成SMN服务授权。

启动检测任务

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击需要进行智能运维的集群名称，跳转至该集群基本信息页面。
3. 左侧菜单栏选择“智能运维”。
4. 在智能运维页面，单击左上角的“启动检测”。
5. 在对话框中填写检测任务的基本信息，单击“确定”。

表 3-93 配置检测任务

参数	说明
名称	检测任务的名称。
描述	检测任务的简短描述。
SMN主题	当勾选“任务完成后发送SMN告警”时，需要选择当前账号已创建的SMN主题。
告警灵敏度	当勾选“任务完成后发送SMN告警”时，需要选择告警灵敏度。 当检测到的风险项存在大于等于此告警敏感度的条目，则发送SMN告警消息，消息内容则包含所有风险项。

- 当检测任务状态变成“完成”时，表示智能检测结束，可以查看集群风险项。

查看集群风险项

当检测任务完成时，可以查看检测结果。

- 登录云搜索服务管理控制台。
- 在集群管理页面，单击集群名称，跳转至该集群基本信息页面。
- 左侧菜单栏选择“智能运维”。
- 在智能运维列表页面，选择已启动的检测任务。单击下方的 ^，可查看当前检测任务的“创建时间”、“摘要”、“ID”以及检测任务所发现的“风险项”。
单击风险项左侧 v，可查看风险项详情，包括检测项、风险描述、风险建议。
您可以根据风险建议及时处理集群所存在的风险。

图 3-52 风险项



- 单击智能运维列表操作列的“导出风险项”，支持下载检测结果。

删除检测任务

当检测任务使用完毕，可以删除任务。删除检测任务后，系统将删除检测任务所对应的所有诊断信息。

- 登录云搜索服务管理控制台。

2. 在集群管理页面，单击集群名称，跳转至该集群基本信息页面。
3. 左侧菜单栏选择“智能运维”。
4. 选择待删除的检测任务，单击操作列的“删除”。
5. 在对话框中输入“DELETE”，单击“确定”，删除检测任务。

3.13.5 查询和管理 OpenSearch 集群日志

为了方便用户使用日志定位问题，云搜索服务提供了日志备份、日志采集和日志查询功能。


日志备份可以定期将集群的日志存储在OBS桶中，通过OBS可以直接下载需要的日志文件，进行问题分析定位。

日志采集可以实时将集群的日志存放到CSS集群中，方便通过Kibana进行日志搜索分析。

日志备份在OBS桶中需要额外收费，具体收费以[对象存储服务计费说明](#)为准。

查询日志

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要查询日志的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”。
4. 在日志管理页面进行日志查询。

5. 选择需要查询的节点、日志类型和日志级别信息后，单击 ，显示查询结果。

查询日志时，是从最近时刻的1万条日志中进行匹配，查询结果最多显示100条。

备份日志

日志备份可以定期将CSS集群的日志存储到OBS桶中。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要配置日志备份的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”，在“日志备份开关”右侧单击开关，打开集群的日志管理功能。
4. 在“编辑日志备份配置”弹窗中，完成参数配置。

弹窗中默认填写了云搜索服务自动为用户创建的“OBS桶”、“备份路径”和“IAM委托”，用于日志备份。支持用户参考[表3-94](#)修改默认值。

如果集群已经启用了日志备份功能，也可以单击“日志备份配置”右侧的编辑图标，在“编辑日志备份配置”窗口，参考[表3-94](#)更新日志备份的配置参数。

表 3-94 日志备份配置

参数	说明	注意事项
“OBS桶”	选择日志存储的OBS桶。单击右侧的“创建桶”支持新建OBS桶。	OBS桶的所在区域必须跟集群的所在区域保持一致。 说明 如果是子账号，需要同时设置GetBucketStoragePolicy、GetBucketLocation、ListBucket、ListAllMyBuckets权限，才能看到OBS桶。
“备份路径”	填写日志在OBS桶中的存放路径。	备份路径配置规则： <ul style="list-style-type: none">• 备份路径不能包括符号“\:*?*<> ”。• 备份路径不能以“/”开头。• 备份路径不能以“.”开头或结尾。• 备份路径的总长度不能超过1023个字符。
“IAM委托”	选择IAM委托，指当前账号授权云搜索服务访问或维护存储在OBS中数据。单击右侧的“创建委托”支持新建委托。	IAM委托需满足如下条件： <ul style="list-style-type: none">• “委托类型”选择“云服务”。• “云服务”选择“Elasticsearch”或者“云搜索服务 CSS”。• 必选策略：“OBS Administrator”

5. 备份日志。

- 自动备份日志。

在“自动备份开关”右侧，单击开关，开启自动备份日志功能。

开启“自动备份开关”后，在“修改日志备份策略”弹窗中设置“备份开始时间”。设置成功后，系统会按照设置的时间进行自动备份。

打开“自动备份开关”后，单击开关右侧的编辑图标，可以修改“备份开始时间”。

- 手动备份日志。

选择“日志备份”页签，单击“开始备份”，在弹出的确认提示框中单击“确定”，开始备份日志。

日志备份列表中的“任务状态”为“Successful”时，表示日志备份成功。

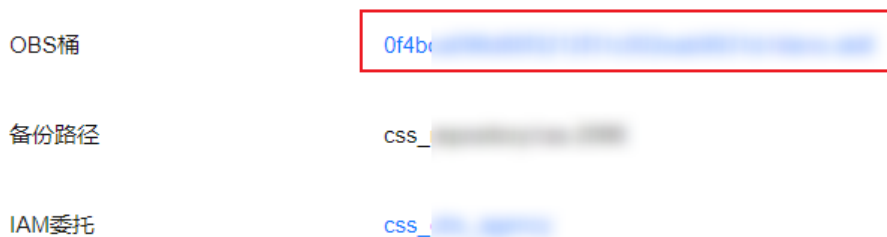
说明

如果日志备份失败，可以在“日志备份”页签的右侧单击查看“失败任务数”，了解失败原因。失败任务数最多显示10条，当日志备份功能关闭或集群删除时，失败记录也会被清空。

6. 查看日志文件。

日志备份成功后，用户可以单击“OBS桶”跳转到桶列表，找到存放备份日志的桶查看日志文件。

图 3-53 进入 OBS



采集日志

日志采集可以实时将CSS集群日志存放到本集群或其他同网络的集群中，方便通过Kibana进行日志搜索分析。

如果要使用日志采集功能，集群必须满足以下条件。当集群不满足要求时，建议先升级集群再启用日志采集。

- 集群版本是Elasticsearch 7.10.2、OpenSearch 1.3.6或OpenSearch 2.11.0。
- 集群的镜像版本号 $\geq 24.2.0$ 。集群的镜像版本号可以在集群列表的“版本”列查看，如图所示。

图 3-54 查看集群子版本号

名称/ID	集群状态	任务状态	版本
css-2e92	可用	-	24.2.0_0625
0ab2278b-7d...			

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要配置日志备份的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”，在“日志采集开关”右侧单击开关，打开集群的日志采集功能。

说明

如果页面上没有“日志采集开关”，则表示该集群不支持采集日志。

4. 在“日志采集配置”弹窗中，完成参数配置。

表 3-95 日志采集配置

参数	说明
索引前缀名称	<p>设置采集的日志中索引名称的前缀，则日志采集的索引名称为“索引前缀名称+日志采集日期”，日志采集日期以天为单位。</p> <p>索引前缀名称只支持数字、小写字母、下划线和中划线，支持1~128个字符。</p>

参数	说明
保留周期	设置采集的日志的保留周期，单位为天，支持输入1~3650。当采集的日志超过保存周期时，日志将会被删除。
日志保存集群	设置采集的日志保存在哪个集群，支持选择“当前集群”和“其他集群”。 默认选择“当前集群”。当选择“其他集群”时，需要选择日志保存的目标集群，并进行集群网络联通行检查，只有在同一VPC网络下的集群才支持跨集群保存日志，且集群版本必须一致。

如果集群已经启用了日志采集功能，也可以单击右侧的编辑图标修改日志采集配置。

- 单击“确定”，开启集群日志采集开关。
当“状态”变成“运行中”时，表示已启动日志采集。
单击“访问Kibana”，登录集群，可以搜索查看日志。
单击“日志保存集群”处的集群名称，可以跳转到集群详情页面。

图 3-55 日志采集



- 当需要关闭日志采集时，单击“日志采集开关”右侧的开关，在弹窗中单击“确定”，关闭日志采集。
日志采集关闭后，集群中已采集的日志不会被清空，仅当超过采集日志时设置的“保存周期”时才会被删除。

日志文件介绍

Elasticsearch和OpenSearch集群备份的日志文件主要包括废弃操作日志、运行日志、慢索引日志、慢查询日志。

表 3-96 日志文件类型

日志名称	描述
clustername_deprecation.log	弃用操作的日志记录。
clustername_index_indexing_slowlog.log	慢索引日志。
clustername_index_search_slowlog.log	慢索引查询日志。
clustername.log	Elasticsearch运行日志。

日志名称	描述
clustername_access.log	接入日志。

3.14 查看 OpenSearch 集群审计日志

公有云平台提供了云审计服务。通过云审计服务，您可以记录与云搜索服务相关的操作事件，便于日后的查询、审计和回溯。

前提条件

已开通云审计服务。

支持审计的关键操作列表

表 3-97 支持审计的关键操作列表

操作名称	资源类型	事件名称
创建集群	cluster	createCluster
删除集群	cluster	deleteCluster
扩容集群	cluster	roleExtendCluster
重启集群	cluster	rebootCluster
配置自定义词库	cluster	loadLexicon
删除自定义词库	cluster	deleteLexicon
设置集群快照的基础配置	cluster	updateSnapshotPolicy
设置自动创建快照策略	cluster	updateAutoSnapshotPolicy
集群升级	cluster	upgradeCluster
升级重试	cluster	retryAction
手动创建快照	snapshot	createSnapshot
恢复快照	snapshot	restoreSnapshot
删除快照	snapshot	deleteSnapshot

查看审计事件

在云审计服务管理控制台，用户创建管理类追踪器后，系统会开始记录用户对云服务资源的操作，创建数据类追踪器后，系统会开始记录用户对OBS桶中数据的操作。云审计服务管理控制台会保存最近7天的操作记录。

当需要在云审计服务管理控制台查看或导出最近7天的操作记录时，请参见[查询审计事件](#)。

4 使用 Logstash 迁移数据

4.1 Logstash 使用流程

表 4-1 Logstash 集群的使用流程

任务分类	操作	参考文档	
使用	规划集群	在创建Logstash集群前，需要先完成集群规划。规划时，应考虑是否多可用区部署以提高集群的高可用性，合理配置集群的节点数量与规格，以确保集群的稳定性和性能。	Logstash集群规划建议
	创建集群	基于集群规划，创建合适的Logstash集群。	创建Logstash集群
	配置路由	当Logstash集群与数据源子网位于同一主网卡时，网络自动互通；如果不同，则需在Logstash集群的每个节点上配置指向源数据服务器IP的回程路由。	配置Logstash集群路由
	配置Logstash迁移任务	通过Logstash集群的配置中心，可以创建并启动Logstash配置文件，以实现数据从不同数据源到目的端的迁移。	创建Logstash配置文件 启动Logstash迁移任务
运维	扩缩容	CSS服务提供灵活的扩容和缩容功能，支持增加节点数量、升级节点规格，以及随机缩容节点，以动态调整集群资源，适应不同业务需求并优化成本。	扩容Logstash集群 缩容Logstash集群

任务分类	操作	参考文档
	管理集群 CSS服务提供全面的集群管理功能，包括集群信息查看、标签管理、企业项目绑定、自定义CA证书上传等，旨在帮助用户高效、灵活地管理Logstash集群。	查看Logstash集群信息 设置Logstash集群标签为Logstash集群绑定企业项目 管理Logstash集群的CA证书 强制重启Logstash集群 删除Logstash集群
	监控与日志管理 CSS服务提供全面的监控和日志管理功能，包括集群、节点和Logstash管道监控指标、告警规则配置、日志备份与查看，帮助用户有效监控和分析Logstash集群，确保集群的稳定性。	CES中Logstash集群支持的监控指标 使用CES监控Logstash集群 查询和管理Logstash集群日志
	审计日志 通过云审计服务，用户可以记录和查询Logstash集群的关键操作事件，且操作记录在管理控制台保存最近7天，便于后续的审计和回溯。	查看Logstash集群审计日志

4.2 Logstash 集群规划建议

规划集群可用区

为防止数据丢失，并确保在服务中断情况下能降低集群的停机时间，从而增强集群的高可用性，CSS服务支持跨可用区（即多可用区）部署。用户可以在同一个区域内选择两个不同的可用区进行集群部署。

在创建集群时，如果用户选择了两个可用区，CSS服务将自动开启跨AZ的高可用性特性，确保节点在这些可用区内均匀分配。系统均匀分配的节点满足各个AZ之间节点数量的差小于等于1，具体的节点分布情况，包括不同数量的节点如何在各个可用区中分布，可以参考[表4-2](#)。

须知

- 在创建集群时，选择的节点数量都要大于等于2，否则跨可用区部署会失败。
- 部署跨AZ集群时，节点会被均匀的分布在不同的AZ上。

表 4-2 节点数量和 AZ 分布

集群节点个数	单AZ	两AZ	
	AZ1	AZ1	AZ2

1个节点	1	不支持	
2个节点	2	1	1
3个节点	3	2	1
4个节点	4	2	2
...

规划节点存储

- **规划节点机型**

由于Logstash集群不涉及存储大规格数据，推荐选用计算密集型机型。

- **规划节点规格**

在规划节点规格时，推荐优先考虑高配置但节点数量较少的方案。例如，一个由3个节点组成的集群，每个节点配置为32核CPU和64GB内存，通常比一个由12个节点组成的集群，每个节点配置为8核CPU和16GB内存，在集群的稳定性和扩展性方面更具优势。

优势主要体现在如下方面。

- **集群稳定性**：高配置节点通常能提供更强的处理能力和更大的内存空间，从而提高集群的整体稳定性。
- **扩容便捷性**：当高配置集群遇到性能瓶颈时，可以通过横向扩展轻松解决，即简单地向集群中添加更多具有相同高配置的节点。这种扩展方式简单直接，易于实施。
- **维护简便**：较少的节点数量意味着更少的维护工作和更低的管理复杂性。

相比之下，低配置集群在需要扩容时，往往需要进行纵向扩展，即提升单个节点的配置。这不仅可能涉及更复杂的迁移和升级过程，还可能增加额外的维护成本和技术挑战。

因此，在规划集群时，应综合考虑性能、成本、维护和扩展性，选择最适合业务需求的节点规格。

- **规划存储容量**

Logstash集群的云盘存储容量默认是40GB，且不支持修改，因此无需规划。

规划虚拟私有云和子网

CSS服务支持在非共享VPC和共享VPC内创建集群。

共享VPC和非共享VPC相比，有如下优势：

- 在一个账号内统一创建资源，并将资源共享给其他账号，其他账号无需创建重复资源，可以精简资源数量以及网络架构，提升管理效率并节约成本。
比如不同账号下的VPC网络互通需要建立对等连接，使用共享VPC后，则不同账号用户可以在同一个VPC创建资源，免去了对等连接配置，有效的简化组网结构。
- 在一个账号内统一管理运维资源，便于企业集中配置业务安全策略，并且利于对资源使用情况的监控和审计，支撑业务的安全诉求。

当选择使用共享VPC创建集群时，需要提前完成共享VPC子网的创建，操作流程请参见[表4-3](#)。有关共享VPC的使用限制及详细操作指导等信息可以参考[共享VPC概述](#)。

表 4-3 共享子网创建流程说明

方法	说明	操作指导
方法A	<ol style="list-style-type: none">通过RAM管理控制台，所有者创建共享，将子网共享给使用者。配置如下：<ol style="list-style-type: none">选择共享子网。为共享子网选择权限，即指定使用者对该共享子网具备的权限。 CSS服务在共享VPC中创建集群需要选择权限：default vpc subnet statement。指定共享子网的使用者，可以指定多个。共享创建完成后，通过RAM管理控制台，使用者可以选择接受或者拒绝共享申请。<ul style="list-style-type: none">使用者接受共享申请，子网共享成功。如果后续使用者不再需要使用该共享子网，可以退出该共享。使用者拒绝共享申请，子网共享失败。	<ol style="list-style-type: none">创建共享接受/拒绝共享邀请 退出共享
方法B	<ol style="list-style-type: none">通过RAM管理控制台，所有者创建共享，将子网共享给使用者。配置如下：<ol style="list-style-type: none">选择共享子网。为共享子网选择权限，即指定使用者对该共享子网具备的权限。 CSS服务在共享VPC中创建集群需要选择权限：default vpc subnet statement。指定共享子网的使用者，可以指定多个。通过VPC管理控制台，选择上一步中已创建的共享，将子网加入到该共享内。共享创建完成后，通过RAM管理控制台，使用者可以选择接受或者拒绝共享申请。<ul style="list-style-type: none">使用者接受共享申请，子网共享成功。如果后续使用者不再需要使用该共享子网，可以退出该共享。使用者拒绝共享申请，子网共享失败。	<ol style="list-style-type: none">创建共享将VPC子网共享给其他账号接受/拒绝共享邀请 退出共享

4.3 创建 Logstash 集群

前提条件

已经参考[Logstash集群规划建议](#)完成待创建的Logstash集群规划。

操作步骤

1. 登录云搜索服务[管理控制台](#)。
2. 在“总览”页面单击右上角的“创建集群”，进入“创建集群”页面。
或者左侧导航栏单击“集群管理>Logstash”，单击右上角的“创建集群”，进入“创建集群”页面。
3. 在“基础配置”页面，完成Logstash集群的基本信息和资源配置。

表 4-4 Logstash 集群的基础配置

参数	说明
计费模式	集群支持包年/包月和按需计费两种模式。 <ul style="list-style-type: none">● 包年/包月：根据集群购买时长，一次性支付集群费用。最短时长为1个月，最长时长为3年。如果购买时长超过9个月，建议包年购买，价格更优惠。一年计费为购买10个月得12个月。● 按需计费：按实际使用时长计费，计费周期为一小时，不足一小时按一小时计费。
订购周期	选择包年/包月模式后，需要选择购买时长。 您可以根据需求，选择是否需要“自动续费”。
当前区域	选择集群的所在区域。 不同区域的云服务产品之间内网互不相通。请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
可用区	选择集群工作区域下关联的可用区。 最多支持配置2个“可用区”，多可用区的使用建议请参见 规划集群可用区 。
集群类型	选择“Logstash”。
集群版本	选择所需的集群版本，支持的版本以界面可选项为准。
集群名称	自定义集群名称，可输入的字符范围为4~32个字符，只能包含数字、字母、中划线和下划线，且必须以字母开头。
节点数量	集群中的节点个数。可选节点数为1~100。
CPU架构	仅支持“X86计算”。
节点规格	集群中的节点规格。您可以根据需求，选择对应的规格。每个集群只能选择一个规格，规格的详细说明可参考 弹性云服务器的实例类型与规格 。
节点存储	选择存储类型，支持普通I/O、高I/O、超高I/O、极速型SSD。 说明 如果当前节点的存储类型不支持，则界面不显示。

参数	说明
节点存储容量	设置节点的存储空间大小，其取值范围与“节点规格”关联，不同的规格允许的取值范围不同。 Logstash集群的节点存储容量只支持配置为40GB。
企业项目	如果开通了“企业项目”，在创建集群时可以给集群绑定一个企业项目。 在下拉框中选择企业项目，单击“查看项目管理”跳转到“企业项目管理”管理控制台，查看已有的企业项目。

- 单击“下一步，网络配置”。
- 在“网络配置”页面，完成Logstash集群的网络和安全组配置。

表 4-5 参数说明

参数	说明
虚拟私有云	指定集群节点使用的虚拟专用网络，实现不同业务的网络隔离。 单击“查看虚拟私有云”跳转到虚拟私有云列表，查看已创建或共享至当前账号下的VPC名称和ID。 如果没有合适的VPC，建议联系CSS服务管理员新建VPC，具体请参见 创建虚拟私有云和子网 。 说明 此处选择的VPC必须包含网段（CIDR），否则集群将无法创建成功。新建的VPC默认包含网段（CIDR）。
子网	集群使用子网实现与其他网络的隔离，并独享所有网络资源，以提高网络安全。 选择当前虚拟私有云下集群需要的子网。支持选择共享VPC下的子网。
安全组	安全组起着虚拟防火墙的作用，为集群提供安全的网络访问控制策略。 选择集群需要的安全组，单击“查看安全组”跳转到安全组列表，可以了解安全组详情。 说明 请确保安全组的“端口范围/ICMP类型”为“Any”或者包含端口9200的端口范围。

图 4-1 设置网络规格

虚拟私有云	vpc-	Q 查看虚拟私有云
	集群所在的虚拟专用网络，可以对不同业务进行网络隔离。	
子网	subnet-	Q
	通过子网提供与其他网络隔离的、可以独享的网络资源，以提高网络安全。	
安全组	dws-	Q 查看安全组
	安全组起着虚拟防火墙的作用，为集群提供安全的网络访问控制策略。	

6. 单击“下一步：高级配置”。
7. 在“高级配置”页面，完成Logstash的高级配置。

如果选择自定义，可设置标签。

- “默认配置”：默认关闭“标签”功能。
- “自定义”：根据需要选择开启“标签”功能。

标签

为集群添加标签，方便用户识别和管理拥有的集群资源。此处您可以选择“标签管理服务”中已定义好的“预定义标签”，也可以自己定义标签。

📖 说明

如果您的组织已经设定云搜索服务的相关标签策略，则需按照标签策略规则为集群添加标签。标签如果不符合标签策略的规则，则可能会导致集群创建失败，请联系组织管理员了解标签策略详情。

表 4-6 标签命名规则

参数	说明
标签键	<ul style="list-style-type: none"> • 对于同一个集群，标签键值唯一。 • 长度不超过64个字符。 • 只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @”。不能以空格开头和结尾。 • 不能为空。
标签值	<ul style="list-style-type: none"> • 长度不超过64个字符。 • 只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @/”。不能以空格开头和结尾。 • 不能为空。

8. 单击“下一步：确认配置”，确认完成后单击“立即创建”开始创建集群。
9. 单击“返回集群列表”，系统将跳转到“集群管理”页面。您创建的集群将展现在集群列表中，且集群状态为“创建中”，创建成功后集群状态会变为“可用”。

如果集群创建失败，请根据界面提示重新创建集群。

4.4 配置 Logstash 集群路由

- 当Logstash集群和数据源的子网在同一个主网卡时，两者的网络互通，无需配置Logstash集群路由。
- 当Logstash集群和数据源的子网不在同一个主网卡时，需要配置Logstash集群路由，给Logstash集群的每个节点新增一个指向目标IP地址（即源数据所在的服务器IP地址）的回程路由，以此打通Logstash集群和数据源之间的网络。

操作步骤

1. 登录云搜索服务管理控制台。
2. 左侧导航栏选择“集群管理 > Logstash”，进入集群列表页面。
3. 在集群列表，单击集群名称，进入集群基本信息页面。
4. 单击“集群路由”后面的“修改”。

图 4-2 配置集群路由

配置信息

区域	
可用区	
虚拟私有云	vpc-
子网	subnet-
安全组	default 更改安全组
集群路由	修改 查看
企业项目	default
内网访问IPv4地址	192.

5. 在“修改集群路由”弹窗中，配置路由信息。

表 4-7 配置集群路由

参数	说明
ip地址	填写源数据所在的服务器IP地址，取前16位或者24位，例如源IP为“192.168.1.1”，可以填“192.168.0.0”。
子网掩码	填写IP地址的子网掩码。 <ul style="list-style-type: none">当IP地址取的是16位，则子网掩码填“255.255.0.0”。当IP地址取的是24位，则子网掩码填“255.255.255.0”。 说明 子网掩码必须要覆盖IP网段，即子网掩码和IP地址转换为二进制后，IP地址最后的0个数一定要比子网掩码的最后为0的个数多。
更改类型	选择集群路由的更改类型。 <ul style="list-style-type: none">“添加”：新增路由“删除”：删除路由

图 4-3 配置集群路由

修改集群路由

ip地址

子网掩码

更改类型

- 单击“确定”完成集群路由配置。
- 单击“集群路由”后面的“查看”，在“集群路由”弹窗中可以了解集群路由更新信息。

4.5 配置 Logstash 迁移任务

4.5.1 创建 Logstash 配置文件

Logstash集群通过配置文件定义迁移方案，从不同的数据源（input）迁移数据到不同的目的端（output）。

约束限制

- 创建的配置文件大小不能超过100k。
- Logstash集群允许最多配置50个配置文件。

前提条件

创建配置文件前，必须先获取数据源和目的端的服务器或集群信息，例如IP地址、用户名、密码等。

创建配置文件

1. 登录云搜索服务管理控制台。
2. 左侧导航栏选择“集群管理 > Logstash”，进入集群列表页面。
3. 在集群列表，单击集群操作列的“配置中心”，进入配置中心页面。
或者，在集群列表，单击集群名称，进入集群基本信息页面，在左侧导航栏选择“配置中心”，进入配置中心页面。
4. 单击右上角“创建”，进入创建配置文件页面，编辑配置文件。

表 4-8 创建配置文件

参数	说明
名称	自定义配置文件名称。 只能包含字母、数字、中划线或下划线，且必须以字母开头。必须大于等于4个字符。
配置文件内容	支持使用配置文件模板进行配置项修改，或是直接开发配置文件内容。 使用配置文件模板时，需要展开“系统模板”或“自定义模板”，根据模板描述选择，单击操作列的“应用”即可直接修改配置项。 <ul style="list-style-type: none">• 系统模板是CSS服务预置的配置文件模板，具体请参见Logstash配置文件模板介绍。• 自定义模板是用户在使用过程中添加的配置文件模板。 说明 配置文件内容大小不能超过100k。
隐藏内容列表	配置隐藏字符串列表后，在返回的配置内容中，会将所有在列表中的字符串隐藏为“***”。 输入需要隐藏的敏感字符串列表，按“Enter”创建。 列表最大支持20条，单个字符串最大长度512字节。

5. 编辑完成后，单击“下一页”配置Logstash管道参数。

表 4-9 管道参数说明

参数	说明
pipeline.workers	并行执行管道的Filters和Outputs阶段的工作线程数。 取值范围：1~20 默认值：CPU核数
pipeline.batch.size	单个工作线程在尝试执行其Filters和Outputs之前将从inputs收集的最大事件数，该值较大通常更有效，但会增加内存开销。 默认值：125
pipeline.batch.delay	创建管道事件批时，在将过小的批调度到管道工作线程之前，等待每个事件的时间。 默认值：50 单位：毫秒
queue.type	用于事件缓冲的内部队列模型。 <ul style="list-style-type: none">“memory”为基于内存的传统队列。“persisted”为基于磁盘的ACKed持久化队列。 默认值：memory
queue.checkpoint.writes	表示强制执行检查点之前写入的最大事件数。当“queue.type”配置为“persisted”时，才需要配置该参数。 默认值：1024
queue.max_bytes	表示持久化队列的总容量，确保磁盘的容量大于该值。当“queue.type”配置为“persisted”时，才需要配置该参数。 默认值：1024 单位：MB

6. 配置完成后，单击“创建”。


在配置中心页面可以看到创建的配置文件，状态为“可用”，表示创建成功。

管理配置文件

在配置文件列表，支持对创建的配置文件进行编辑、添加到自定义模板、删除等操作。

表 4-10 管理配置文件

功能	使用限制	操作指导
修改配置文件	当配置文件有执行中的管道任务时，不支持修改。	在配置文件列表，单击操作列的“编辑”进入编辑配置文件页面，可以修改“配置文件内容”、“隐藏内容列表”及运行参数。

功能	使用限制	操作指导
将配置文件设置为模板	无	在配置文件列表，单击操作列的“添加到自定义模板”进入添加到自定义模板页面，设置模板名称、描述和配置文件内容，单击“确定”添加到自定义模板列表中，方便下次创建配置文件时直接使用。
导出配置文件	无	在配置文件列表，单击右上角的  ，可以批量导出配置文件到本地，便于本地备份。
删除配置文件	当配置文件有执行中的管道任务时，不支持删除。	在配置文件列表，单击操作列的“删除”，在弹窗中单击“确定”删除配置文件。

4.5.2 启动 Logstash 迁移任务

场景描述

使用Logstash集群启动Logstash迁移任务的操作流程：

1. 测试数据源和目的端与Logstash集群之间的网络连通性，确保数据传输畅通：[测试连通性](#)
2. 通过启动配置文件启动Logstash迁移任务。
 - 当管道列表没有“工作中”的任务时，则[启动配置文件](#)。
 - 当管道列表有“工作中”的任务时，则[热启动配置文件](#)。
3. 通过指标监控、操作记录和运行日志[监控迁移任务](#)。

约束限制

- Logstash集群的管道列表中允许最多有50个配置文件在“工作中”。
- 使用Logstash stdin插件的配置文件禁止使用热启动功能。
- 使用热启动功能时，如果配置文件热启动失败且导致Logstash进程异常退出了，集群会自动进行恢复机制重新启动原Logstash进程，请谨慎选择热启动。

测试连通性

在使用Logstash集群迁移数据前，需要先测试下数据源、目的端和Logstash集群的网络是否连通。

1. 登录云搜索服务管理控制台。
2. 左侧导航栏选择“集群管理 > Logstash”，进入集群列表页面。
3. 在集群列表，单击集群操作列的“配置中心”，进入配置中心页面。
或者，在集群列表，单击集群名称，进入集群基本信息页面，在左侧导航栏选择“配置中心”，进入配置中心页面。

4. 在配置中心页面，单击“连通性测试”。
5. 在连通性测试弹窗中，输入数据源、目的端的IP地址和端口号，单击“测试”。连通性测试最多可一次性测试10个IP地址。您可以单击“继续添加”，添加多个IP地址，然后单击“批量测试”，进行一次性测试多个IP地址的连通性。

图 4-4 连通性测试



当显示“可用”时，表示集群间网络连通。

启动配置文件

当管道列表没有“工作中”的任务时，通过启动配置文件启动Logstash迁移任务。可以同时选择多个配置文件进行启动操作，一次不能超过50个。

1. 在配置文件列表，选择需要启动的配置文件，单击左上角的“启动”。
2. 在“启动Logstash服务”对话框中，根据业务需要选择“是否保持常驻”。开启“保持常驻”以后，将会在每个节点上面配置一个守护进程，当logstash服务出现故障的时候，会主动拉起并修复。“保持常驻”适用于需要长期运行的业，不适用于短期运行的业务，短期业务如果源端无数据，开启保持常驻会导致任务失败。
3. 单击“确定”，开始启动配置文件启动Logstash迁移任务。
可以在管道列表看到启动的配置文件。

热启动配置文件

当管道列表有“工作中”的任务时，通过热启动配置文件启动Logstash迁移任务。一次只能选择一个配置文件进行热启动，即一次只能新增一个管道。

1. 在配置文件列表，选择一个需要热启动的配置文件，单击左上角的“热启动”。
2. 在“热启动Logstash服务”对话框中，“是否保持常驻”的选项默认与管道列表中其他任务的值保持一致，且不支持修改。
3. 单击“确定”，开始热启动配置文件启动Logstash迁移任务。
可以在管道列表看到热启动的配置文件。

监控迁移任务

当启动配置文件后，在管道列表会生成一个任务，CSS服务支持查看管道任务的数据监控。

- 在管道列表，单击操作列的“指标监控”，跳转到云监控服务页面，可以查看监控信息。
当管道的“事件”数据动态变化时，监控数据也会同步变化；当管道任务在启动中、停止中或事件数据稳定时，监控数据则无变化。

支持的监控指标请参考[Logstash管道监控指标列表](#)。当需要配置监控告警时，可以参考[使用CES监控Logstash集群](#)。

- 在管道列表，单击左上方“操作记录”或“运行日志”，可以查看配置文件的启停记录和迁移任务的运行日志信息。

4.5.3 停止 Logstash 迁移任务

场景描述

在Logstash集群停止Logstash迁移任务有两种方式。

- 全部停止：在管道列表中，同时停止所有任务。停止成功后，管道“运行状态”变成“已停止”。
- 热停止：在管道列表中，一次只能停止一个管道任务。停止成功后，该管道任务会直接从管道列表中移除。

📖 说明

停止管道会造成Logstash数据迁移中断，请谨慎操作。

前提条件

Logstash集群的配置中心有“工作中”的管道任务。

停止全部迁移任务

- 当需要停止管道列表中全部配置文件的数据迁移任务时，单击管道列表上方的“全部停止”。
- 根据对话框提示，单击“确定”，停止所有管道的数据迁移任务。
全部停止成功则可以看到管道列表所有任务均停止，且全部管道数据迁移中断。

热停止某个迁移任务

- 当需要停止管道列表中某一个配置文件的数据迁移任务时，勾选管道，单击管道列表上方的“热停止”。
- 根据对话框提示，单击“确定”，停止所选管道的数据迁移任务。
热停止成功则可以看到管道列表中目标配置已被移除且该管道数据迁移中断。

4.5.4 Logstash 配置文件模板介绍

系统模板介绍

Logstash集群预置了常见的几种Logstash配置文件模板，请参见[表4-11](#)。

表 4-11 Logstash 配置文件的系统模板

名称	描述	具体介绍
redis	从redis数据库导入数据到Elasticsearch类型集群。	redis模板说明

名称	描述	具体介绍
elasticsearch	从Elasticsearch类型集群导入数据到Elasticsearch类型集群。	elasticsearch模板说明
jdbc	从jdbc导入数据到Elasticsearch类型集群。	jdbc模板说明
kafka	从kafka导入数据到Elasticsearch类型集群。	kafka模板说明
beats	从beats导入数据到Elasticsearch类型集群。	dis模板说明
dis	从dis导入数据到Elasticsearch类型集群。	beats模板说明

redis 模板说明

从redis数据库导入数据到Elasticsearch类型集群。

```
input {
  redis {
    data_type => "pattern_channel" # data_type, one of ["list", "channel", "pattern_channel"]
    key => "lgs-*" # The name of a Redis list or channel.
    host => "xxx.xxx.xxx.xxx"
    port => 6379
  }
}

filter {
  # Delete some fields added by Logstash.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}

output {
  elasticsearch {
    hosts => ["http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200"] #
  host
    # user => "xxx" # user name, just for security cluster
    # password => "xxx" # password, just for security cluster
    index => "xxxxxx" # destination _index
  }
}
```

表 4-12 配置项说明

配置项	是否必填	说明
data_type	是	数据源类型。包括list, channel, pattern_channel。 <ul style="list-style-type: none">• 如果data_type为list, 则使用BLPOP密钥。• 如果data_type为channel, 使用SUBSCRIBE密钥。• 如果data_type为pattern_channel, 则使用PSUBSCRIBE密钥。
key	是	Redis的list或者channel名称。
host	是	Redis服务器的IP。
port	否	需要连接的端口号。 默认值: 6379
hosts	是	Elasticsearch集群的访问地址。
user	否	访问Elasticsearch集群的用户名, 一般为admin。 仅安全集群需要输入此参数。
password	否	访问Elasticsearch集群的密码。此密码为创建集群时设置的密码。 仅安全集群需要输入此参数。
index	是	配置为需要迁移数据的索引, 即用户需要迁移数据到哪个索引, 就配置哪个索引。 不支持配置多条索引。

更多信息请参见Logstash官方文档[Redis input plugin](#)。

elasticsearch 模板说明

从Elasticsearch类型集群导入数据到Elasticsearch类型集群。

```
input {
  elasticsearch {
    # Source ES cluster IP address. When SSL is enabled, please use 7.10.0 and you donot need to add a
    # protocol, otherwise an error is reported.
    hosts => ["http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200"]
    # user => "xxxx" # user name, just for security cluster
    # password => "xxxx" # password, just for security cluster
    index => "xxx,xxx,xxx" # List of indexes to be migrated, separate with commas (,)
    docinfo => true
    # source ES certificate, the cluster on the cloud this value remains the same. Enter the corresponding
    # path when using a custom certificate. Self-built Logstash cluster, You can download the certs file from the
    # ES cluster details page, enter the corresponding path here.
    # ca_file => "/rds/datastore/logstash/v7.10.0/package/logstash-7.10.0/extend/certs" # for 7.10.0
    # ssl => true # Set to true when SSL is enabled.
  }
}
```

```

filter {
  # Delete some fields added by Logstash.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}

output {
  elasticsearch {
    # Destination ES cluster IP address. When SSL is enabled, please use 7.10.0 and you donot need to
    add a protocol.
    hosts => ["http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200"]
    # user => "xxx" # user name, just for security cluster
    # password => "xxx" # password, just for security cluster
    index => "%{[@metadata][_index]}" # destination _index, this config ensure that the index is
    consistent with that on the source. You can also specify the index name.
    # document_type => "%{[@metadata][_type]}" # destination _type, this config ensure that the
    document_type is consistent with that on the source.
    # document_id => "%{[@metadata][_id]}" # destination _id, if you do not need to retain the original
    _id, delete this line to improve performance.
    # source ES certificate, the cluster on the cloud this value remains the same. Enter the corresponding
    path when using a custom certificate. Self-built Logstash cluster, You can download the certs file from the
    ES cluster details page, enter the corresponding path here.
    # cacert => "/rds/datastore/logstash/v7.10.0/package/logstash-7.10.0/extend/certs" # for 7.10.0
    # ssl => true # Set to true when SSL is enabled.
    # ssl_certificate_verification => false # Set to false to ignore the validation server certificate when SSL
    is enabled.
  }
}

```

表 4-13 配置项说明

配置项	是否必填	说明
hosts	是	导入数据的Elasticsearch集群的访问地址。
user	否	访问Elasticsearch集群的用户名，一般为admin。 仅安全集群需要输入此参数。
password	否	访问Elasticsearch集群的密码。此密码为创建集群 时设置的密码。 仅安全集群需要输入此参数。
index	是	导入数据的索引，即用户需要从哪个索引迁移出数 据。
docinfo	否	文档信息。 取值范围：true、false。 如果设置，请在事件中包括Elasticsearch文档信 息，例如索引、类型和ID。
ca_file	否	默认值 “/rds/datastore/logstash/v7.10.0/ package/logstash-7.10.0/extend/certs” 。 云上logstash集群保持默认值不变或使用自定义证 书时输入相应的自定义证书路径。 自建Logstash集群，可以在开启SSL的ES集群详情 页面下载证书文件，在此处输入相应的路径。
ssl	否	源端Elasticsearch集群启用SSL时，设置为true。

配置项	是否必填	说明
hosts	是	输出数据的Elasticsearch集群的访问地址。
user	否	访问Elasticsearch集群的用户名，一般为admin。 仅安全集群需要输入此参数。
password	否	访问Elasticsearch集群的密码。此密码为创建集群时设置的密码。 仅安全集群需要输入此参数。
index	是	配置为需要迁移数据的索引，即用户需要迁移数据到哪个索引，就配置哪个索引。 不支持配置多条索引。
document_type	否	<ul style="list-style-type: none">当docinfo参数配置为true时，此参数生效。当docinfo参数配置为false时，需要从配置文件中删除此参数。
document_id	否	<ul style="list-style-type: none">当docinfo参数配置为true时，此参数生效。当docinfo参数配置为false时，需要从配置文件中删除此参数。
cacert	否	默认值“/rds/datastore/logstash/v7.10.0/package/logstash-7.10.0/extend/certs”。 <ul style="list-style-type: none">云上logstash集群保持默认值不变或使用自定义证书时输入相应的自定义证书路径。自建Logstash集群，可以在开启SSL的ES集群详情页面下载证书文件，在此处输入相应的路径。
ssl	否	目的端Elasticsearch集群启用SSL时，设置为true。
ssl_certificate_verification	否	设置为false时，启用SSL时将忽略验证服务器证书。

更多信息请参见Logstash官方文档[Elasticsearch input plugin](#)。

jdbc 模板说明

从jdbc导入数据到Elasticsearch类型集群。

```
input {
  jdbc {
    # for 7.10.0, jdbc_driver_library => "/rds/datastore/logstash/v7.10.0/package/logstash-7.10.0/extend/
jars/mariadb-java-client-2.7.0.jar"
    jdbc_driver_library => "xxxxxxxxxx"
    jdbc_driver_class => "org.mariadb.jdbc.Driver"
    jdbc_connection_string => "jdbc:mariadb://xxx.xxx.xxx.xxx:xxx/data_base_name"
    jdbc_user => "xxxx"
    jdbc_password => "xxxx"
    statement => "SELECT * from table_name" # This SQL statement determines the data to be input
  }
}
filter {
```

```
# Delete some fields added by Logstash.
mutate {
  remove_field => ["@timestamp", "@version"]
}
}

output {
  elasticsearch {
    hosts => ["http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200"] #
  host
  # user => "xxx" # user name, just for security cluster
  # password => "xxx" # password, just for security cluster
  index => "xxxxxx" # destination_index
}
}
```

表 4-14 配置项说明

配置项	是否必填	说明
jdbc_driver_library	是	JDBC驱动程序库路径。 <ul style="list-style-type: none"> 当Logstash集群版本是7.10.0时，该字段设置为“/rds/datastore/logstash/v7.10.0/package/logstash-7.10.0/extend/jars/mariadb-java-client-2.7.0.jar”。 目前仅支持上述已有驱动，暂不支持用户自定义上传。
jdbc_driver_class	是	要加载的JDBC驱动程序类，例如“org.mariadb.jdbc.Driver”。
jdbc_connection_string	是	JDBC连接字符串。
jdbc_user	是	JDBC用户名。
jdbc_password	是	JDBC密码。
statement	是	输入数据的SQL语句。
hosts	是	输出数据的Elasticsearch集群的访问地址。
user	否	访问Elasticsearch集群的用户名，一般为admin。 仅安全集群需要输入此参数。
password	否	访问Elasticsearch集群的密码。此密码为创建集群时设置的密码。 仅安全集群需要输入此参数。
index	是	配置为需要迁移数据的索引，即用户需要迁移数据到哪个索引，就配置哪个索引。 不支持配置多条索引。

更多信息请参见Logstash官方文档[Jdbc input plugin](#)。

kafka 模板说明

从kafka导入数据到Elasticsearch类型集群。

```
input {
  kafka{
    bootstrap_servers=>"xxx.xxx.xxx.xxx:xxxx"
    topics=>["xxxxxxx"]
    group_id=>"kafka_es_test"
    auto_offset_reset=>"earliest"
  }
}

filter {
  # Delete some fields added by Logstash.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}

output {
  elasticsearch {
    hosts => ["http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200"] #
  host
  # user => "xxxx" # user name, just for security cluster
  # password => "xxxx" # password, just for security cluster
  index => "xxxxxx" # destination_index
}
}
```

表 4-15 配置项说明

配置项	是否必填	说明
bootstrap_servers	是	Kafka实例的IP和端口号。
topics	是	要订阅的主题列表。
group_id	是	该消费者所属的组的标识符。
auto_offset_reset	是	Kafka中初始偏移量。 <ul style="list-style-type: none">• earliest: 将偏移量自动重置为最早的偏移量• latest: 自动将偏移量重置为最新偏移量• none: 如果未找到消费者组的先前偏移量, 则向消费者抛出异常• anything else: 向消费者抛出异常。
hosts	是	输出数据的Elasticsearch集群的访问地址。
user	否	访问Elasticsearch集群的用户名, 一般为admin。 仅安全集群需要输入此参数。
password	否	访问Elasticsearch集群的密码。此密码为创建集群时设置的密码。 仅安全集群需要输入此参数。

配置项	是否必填	说明
index	是	配置为需要迁移数据的索引，即用户需要迁移数据到哪个索引，就配置哪个索引。 不支持配置多条索引。

更多信息请参见Logstash官方文档[Kafka input plugin](#)。

dis 模板说明

从dis导入数据到Elasticsearch类型集群。

```
input {
  dis {
    # For details, see https://support.huaweicloud.com/usermanual-dis/dis_01_0099.html.
    streams => ["YOUR_DIS_STREAM_NAME"]
    endpoint => "https://dis.xxxx.myhuaweicloud.com" # xxxx should be replaced by your region name
    ak => "YOUR_ACCESS_KEY_ID"
    sk => "YOUR_SECRET_KEY_ID"
    region => "YOUR_Region"
    project_id => "YOUR_PROJECT_ID"
    group_id => "YOUR_APP_ID"
    client_id => "YOUR_CLIENT_ID"
    auto_offset_reset => "earliest"
  }
}

filter {
  # Delete some fields added by Logstash.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}

output {
  elasticsearch {
    hosts => ["http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200"] #
  }
  host
  # user => "xxx" # user name, just for security cluster
  # password => "xxx" # password, just for security cluster
  index => "xxxxxx" # destination_index
}
```

表 4-16 配置项说明

配置项	是否必填	说明
streams	是	指定在DIS服务上创建的通道名称。 与DIS控制台“购买接入通道”时配置的“通道名称”取值一致。
endpoint	是	DIS对应Region的数据接口地址。
ak	是	用户的Access Key。 获取方式请参见 检查认证信息 。
sk	是	用户的Secret Key。 获取方式请参见 检查认证信息 。

配置项	是否必填	说明
region	是	将数据上传到指定Region的DIS服务。
project_id	是	用户所属区域的项目ID。 获取方式请参见 检查认证信息 。
group_id	是	DIS App名称，用于标识一个消费组，值可以为任意字符串。
client_id	否	客户端ID，用于标识消费组内的消费者。 起多个pipeline或者多个Logstash实例消费时，需要配置不同的值。比如实例1的值为client1，实例2的值为client2。
auto_offset_reset	否	指定数据从通道中开始消费的位置，支持： <ul style="list-style-type: none">earliest：从通道中最早的数据开始消费。latest：从通道中最新的数据开始消费。
hosts	是	输出数据的Elasticsearch集群的访问地址。
user	否	访问Elasticsearch集群的用户名，一般为admin。 仅安全集群需要输入此参数。
password	否	访问Elasticsearch集群的密码。此密码为创建集群时设置的密码。 仅安全集群需要输入此参数。
index	是	配置为需要迁移数据的索引，即用户需要迁移数据到哪个索引，就配置哪个索引。 不支持配置多条索引。

beats 模板说明

从beats导入数据到Elasticsearch类型集群。

```
input {
  beats {
    port => 5044 # port
  }
}

filter {
  # Delete some fields added by Logstash.
  mutate {
    remove_field => ["@timestamp", "@version"]
  }
}

output {
  elasticsearch {
    hosts => ["http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200", "http://xxx.xxx.xxx.xxx:9200"] #
  host
  # user => "xxx" # user name, just for security cluster
  # password => "xxx" # password, just for security cluster
  index => "xxxxxx" # destination_index
}
```

```
}  
}
```

表 4-17 配置项说明

配置项	是否必填	说明
port	是	使用Logstash通过Beats连接并索引到Elasticsearch的端口号为5044。
hosts	是	输出数据的Elasticsearch集群的访问地址。
user	否	访问Elasticsearch集群的用户名，一般为admin。 仅安全集群需要输入此参数。
password	否	访问Elasticsearch集群的密码。此密码为创建集群时设置的密码。 仅安全集群需要输入此参数。
index	是	配置为需要迁移数据的索引，即用户需要迁移数据到哪个索引，就配置哪个索引。 不支持配置多条索引。

更多信息请参见Logstash官方文档[Beats input plugin](#)。

4.6 扩缩容 Logstash 集群

4.6.1 扩容 Logstash 集群

当Logstash集群的数据面业务变化，需要动态调整集群节点的数量和规格时，可以执行扩容Logstash集群。

场景描述

表 4-18 Logstash 集群的扩容场景

场景	描述	相关文档
增加节点数量	<p>只支持扩容集群的“节点数量”。适用于如下场景：</p> <ul style="list-style-type: none">当需要确保数据的持续可用性时，扩容Logstash集群的节点数量可以增加系统的容错能力。当现有Logstash节点处理能力达到极限，无法满足日益增长的数据量时，扩容可以提供更多的处理能力，保证数据流的顺畅处理。当Logstash集群需要执行更复杂的数据处理逻辑，如数据丰富、转换等，扩容可以提供更多的计算资源来处理这些任务。	扩容节点数量
升级节点规格	<p>只支持变更集群的“节点规格”。变更规格过程中，会依次对节点进行关机，完成更改后再依次开机，是一个滚动的变更过程。一般用于小规格变更为大规格，常用于如下场景：</p> <ul style="list-style-type: none">当现有的Logstash节点处理能力达到瓶颈，无法满足日益增长的数据量时，升级节点规格可以提供更多的计算资源和存储空间，从而提高整体的处理性能。在流量高峰期，为了及时处理增加的数据，可以通过升级Logstash节点规格来提高处理能力。	变更节点规格

前提条件

- 集群处于“可用”状态，且无正在进行的任务。
- CSS服务有足够的配额支持集群扩容。

约束限制

- 变更节点规格也支持将大规格更改为小规格，但是此变更会降低集群的处理性能，进而影响业务能力，请谨慎使用。
- 集群的“节点数量”大于等于2的集群才支持变更规格。

扩容节点数量

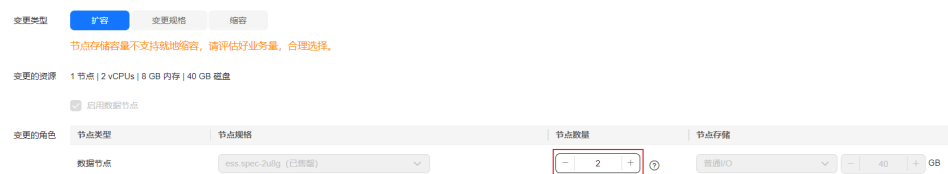
- 登录云搜索服务管理控制台。
- 在左侧菜单栏，选择“集群管理 > Logstash”，进入集群管理页面。

3. 在集群列表选择目标集群，单击操作列的“更多 > 形态变更”进入形态变更页面。
4. 在更改集群规格页面，设置扩容参数。

表 4-19 扩容节点数量

参数	说明
变更类型	选择“扩容”。
变更的资源	显示本次扩容的资源变化量。
变更的角色	在“节点数量”列增加节点数量。

图 4-5 扩容节点数量



5. 单击“下一步”。
6. 确认变更信息后，单击“提交申请”。
7. 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列显示为“扩容”，表示集群正在扩容。当集群状态变为“可用”，则表示扩容成功。

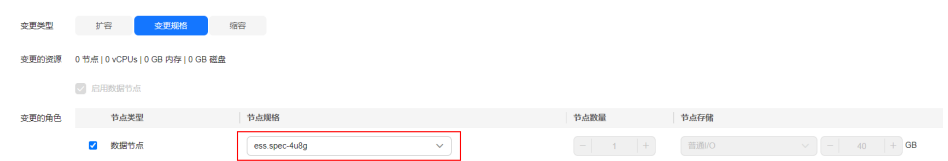
变更节点规格

1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择“集群管理 > Logstash”，进入集群管理页面。
3. 在集群列表选择目标集群，单击操作列的“更多 > 形态变更”进入形态变更页面。
4. 在更改集群规格页面，设置变更规格的参数。

表 4-20 变更节点规格

参数	说明
变更类型	选择“变更规格”。
变更的资源	显示本次变更规格的资源变化量。
变更的角色	先在“节点类型”列勾选“数据节点”，Logstash集群只有数据节点。再在“节点规格”列选择新规格。

图 4-6 集群变更规格



5. 单击“下一步”。
6. 确认变更信息后，单击“提交申请”。
7. 在弹出的窗口确认是否勾选“检测集群状态”，单击“确认”启动集群规格变更。

检测集群状态：默认检测集群状态，变更规格为滚动变更，变更过程中，为了保证变更成功率以及数据安全，会确保每个节点进程拉起来后继续后续节点操作。当集群负载过高业务故障，无法正常下发变更请求，依赖更多的资源才能恢复的紧急情况下可忽略检测集群状态，忽略后，变更过程中因为忽略集群状态检测可能会导致集群故障并中断业务，请谨慎跳过。

8. 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“规格修改”，表示集群正在更改规格。当集群状态变为“可用”，则表示规格变更成功。

4.6.2 缩容 Logstash 集群

当集群有充足的能力处理当前数据时，为节省资源可以缩容Logstash集群。

缩容Logstash集群是通过随机下线节点减少集群的“节点数量”，减少集群占用的资源。

前提条件

集群处于“可用”状态，且无正在进行的任务。

约束限制

- 缩容是通过下线节点减少节点数量，为了降低业务影响，建议在业务低峰期进行缩容操作。
- 关于跨AZ的集群，在不同AZ中同类型节点个数的差值要小于等于1。

操作步骤

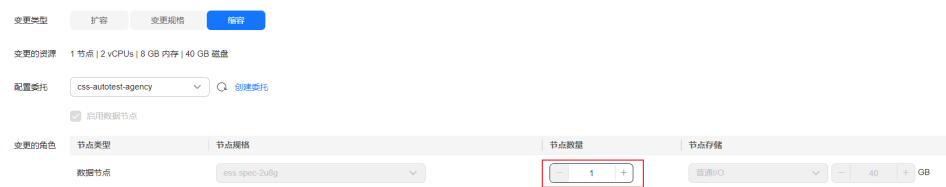
1. 登录云搜索服务管理控制台。
2. 在左侧菜单栏，选择“集群管理 > Logstash”，进入集群管理页面。
3. 在集群列表选择目标集群，单击操作列的“更多 > 形态变更”进入形态变更页面。
4. 在更改集群规格页面，设置缩容参数。

表 4-21 缩容随机节点

参数	说明
变更类型	选择“缩容”。
变更的资源	显示本次缩容的资源变化量。

参数	说明
配置委托	<p>选择IAM委托，授权当前账号访问和使用VPC的权限。仅对接了IAM新平面时才显示该参数。</p> <ul style="list-style-type: none">删除节点会释放网卡，就需要VPC的操作权限，因此选择的委托必须授权“Tenant Administrator”或“VPC Administrator”策略。当首次配置委托时，可以单击“自动创建委托”新建委托“css-auto-agency”直接使用。当没有合适的委托时，可以单击“创建委托”跳转到IAM控制台新建委托。建议联系服务管理员新建委托，创建委托的操作指导请参见创建委托（委托方操作）。
变更的角色	在“节点数量”列减少节点数量。

图 4-7 缩容节点数量



- 单击“下一步”。
- 确认变更信息后，单击“提交申请”。
- 单击“返回集群列表”跳转到集群管理页面。集群的“任务状态”列中显示为“缩容中”，表示集群正在缩容。当集群状态变为“可用”，则表示缩容成功。

4.7 管理 Logstash 集群

4.7.1 查看 Logstash 集群信息

在Logstash集群的列表页和基本信息页，可以获取集群的状态、版本、节点等信息。

集群列表介绍

集群列表会展示当前账号下同一集群类型的所有集群。当集群数量较多时，会翻页显示，支持查看任何状态下的集群。

集群列表默认按时间顺序排列，时间最近的集群显示在最前端。

在集群列表页单击“导出”可以下载全量的集群列表信息。

表 4-22 集群列表说明


参数	描述
名称/ID	展示集群的名称和ID。单击集群名称可进入集群“基本信息”页面，展现了集群的基本信息。集群ID是系统自动生成的，是集群在服务中的唯一标示。
集群状态	展示集群的运行状态。 <ul style="list-style-type: none">● 可用：正常运行的集群。● 创建中：正在创建中的集群。● 处理中：重启中、扩容中、变更规格、缩容中的集群。● 异常：创建失败或不可用的集群。 说明 当集群处于“不可用”状态时，支持删除集群，但是无法扩容集群。您可以查看监控或重启集群处理异常，当依旧执行失败时，请及时联系技术支持。
任务状态	展示重启集群、扩容集群、缩容集群、工作中（管道列表存在运行中的管道）等任务的状态。
版本	展示集群版本号和镜像版本号。 说明 当集群的创建时间比较早时，数据库可能未记录其镜像信息，导致页面无法显示集群的镜像版本号。
创建时间	展示集群的创建时间。
企业项目	展示集群所归属的企业项目。
内网访问地址	展示集群的内网访问地址和端口号，您可以使用此参数接入集群。集群有多个节点时，此处显示多个节点的内网访问地址和端口号。
计费模式	展示集群的计费模式，是“按需计费”还是“包年/包月”。
操作	展示集群可执行的操作入口，包含监控信息、重启、删除等其他更多操作。当某一操作无法执行时，按钮将置灰。

查看集群基本信息

在Logstash集群的基本信息页面，可以获取集群的版本、存储容量、节点等信息。

1. 登录云搜索服务管理控制台。
2. 选择“集群管理 > Logstash”，进入集群列表页面。
3. 单击集群名称进入集群“基本信息”页面，查看集群的基本信息。

表 4-23 基本信息的参数说明

类别	参数	描述
基本信息	集群名称	集群名称。支持自定义名称。 单击右侧  可以修改集群名称。
	ID	集群的唯一标识，是系统自动生成的。 同一个区域下，集群ID是唯一的。
	集群版本	集群的版本信息。
	集群状态	集群当前的状态。
	任务状态	集群当前的任务状态，如果没有进行中的任务则显示“--”。
	创建时间	集群创建的时间。
	集群存储容量 (GB)	集群设置的存储容量。
	集群存储使用量 (GB)	集群已使用的存储容量。
配置信息	区域	集群所在区域。
	可用区	集群所在的可用区。
	虚拟私有云	集群所属的虚拟私有云。
	子网	集群所属的子网。
	安全组	集群所属的安全组。 单击右侧的“更改安全组”可以修改集群的安全组信息。 须知 进行安全组切换前，请确保业务接入时需要的9200端口已经放开，错误的安全组配置可能会导致业务无法访问，请谨慎操作。
	集群路由	单击右侧的“修改”，可修改集群路由信息，详情请参见 配置Logstash集群路由 。
企业项目	集群所属的企业项目。 单击项目名称可以跳转到项目管理页面查看企业项目的基本信息。	

类别	参数	描述
	内网访问地址	集群的内网IP地址和端口号，使用此参数可以接入集群。如果集群只有一个节点，此处仅显示1个节点的IP地址和端口号，例如“10.62.179.32:9600”；如果集群有多个节点，此处显示所有节点的IP地址和端口号，例如“10.62.179.32:9600,10.62.179.33:9600”。
节点信息	节点规格	集群中节点的规格信息。
	节点存储	集群中节点的存储容量和存储类型。
	节点数量	集群中节点的个数。

4.7.2 设置 Logstash 集群标签

标签是集群的标识。为集群添加标签，可以方便用户识别和管理拥有的集群资源。

您可以在创建集群时添加标签，也可以在集群创建完成后，在集群的详情页添加标签。

约束限制

如您的组织已经设定云搜索服务的相关标签策略，则需按照标签策略规则为集群添加标签。标签不符合标签策略的规则，则可能会导致集群创建失败，请联系组织管理员了解标签策略详情。

新建集群的标签管理

1. 登录云搜索服务管理控制台。
2. 单击右上角的“创建集群”，进入创建集群页面。
3. 在创建集群页面，“高级配置”选择“自定义”后，为集群添加标签。

您可以选择预定义标签，并为此标签设置“标签值”。您可以单击“查看预定义标签”，进入“标签管理服务”，了解此用户下已有的标签。

您也可以自定义“标签键”和“标签值”。

图 4-8 创建集群时添加标签



云搜索服务的每个集群最多可以设置20个标签。当设置不正确时，可单击标签右侧的“删除”按钮，删除此标签。当不设置标签时，可保持为空。

表 4-24 标签命名规则

参数	说明
标签键	<ul style="list-style-type: none">• 对于同一个集群，标签键值唯一。• 长度不超过64个字符。• 只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @”。不能以空格开头和结尾。• 不能为空。
标签值	<ul style="list-style-type: none">• 长度不超过64个字符。• 只能包含数字、英文字母、下划线、中划线、中文以及特殊字符“_ . : = + - @/”。不能以空格开头和结尾。• 不能为空。

已有集群的标签管理

您可以对已经创建的集群的标签进行修改，删除，也可以添加标签。

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击待管理标签的集群名称。
系统跳转至该集群“基本信息”页面。
3. 左侧菜单栏选择“标签”，在此可以对集群标签进行添加，修改，删除操作。
 - 查看
在“标签”页，可以查看当前集群的标签详情，包括标签个数，以及每个标签的键和值。
 - 添加
单击左上角的“添加标签”，在弹出的“添加标签”窗口，输入新添加标签的键和值，并单击“确定”。
 - 修改
只能修改已有标签的标签值。
单击标签所在行“操作”列下的“编辑”，在弹出的“编辑标签”窗口，输入修改后标签值，并单击“确定”。
 - 删除
单击标签所在行“操作”列下的“删除”，如果确认删除，在弹出的“删除标签”窗口，单击“确定”。

通过标签搜索集群

1. 登录云搜索服务管理控制台。
2. 在集群管理页面，单击集群列表上方的搜索栏，选择标签键和标签值进行搜索。
标签键和标签值仅支持从下拉列表中选择，当标签键和标签值全匹配时，系统可以自动查询到目标集群。当有多个标签条件时，会取各个标签的交集，进行集群查询。

系统最多支持10个不同标签的组合搜索。

4.7.3 为 Logstash 集群绑定企业项目

企业可以根据组织架构规划企业项目，将企业分布在不同区域的资源按照企业项目进行统一管理，同时可以为每个企业项目设置拥有不同权限的用户组和用户。本章节为您介绍CSS集群如何绑定、修改企业项目。

前提条件

在使用企业项目前，需账号开通权限才可以进行操作。开通权限可以通过提交[工单](#)进行申请。

在绑定企业项目前，您已在“企业项目管理控制台”[创建企业项目](#)。

绑定企业项目

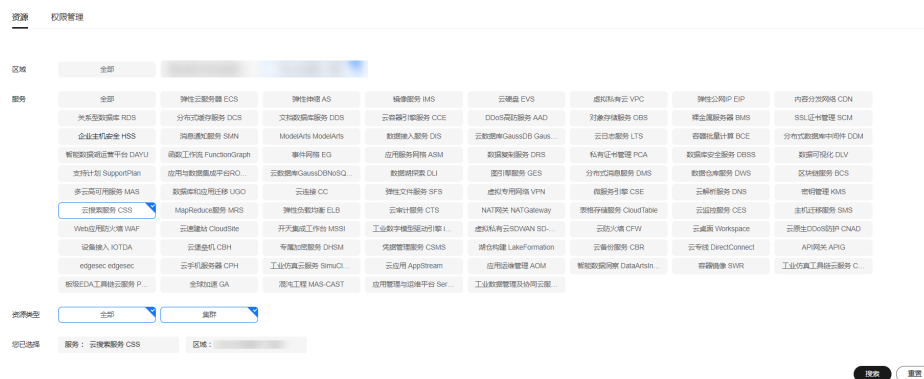
在创建集群时，可以在“企业项目”绑定已创建的企业项目，也可以单击“查看项目管理”，前往企业项目管理控制台，新建企业项目和查看已有的企业项目。

修改企业项目

针对之前已创建的集群，其绑定的企业项目可根据实际情况进行修改。

1. 登录在云搜索服务管理控制台，
2. 在左侧导航栏，选择对应的集群类型，进入集群管理页面。
3. 在集群列表中，单击集群名称进入集群“基本信息”页面。
4. 在集群“基本信息”页面，单击“企业项目”右侧的企业项目名称，进入项目管理页面。
5. 在“资源”页签下，“区域”选项中选择当前集群所在的区域，“服务”选项中选择“云搜索服务 CSS”。此时，资源列表将筛选出对应的CSS集群。

图 4-9 筛选 CSS 集群



6. 勾选需要修改企业项目的集群，然后单击“迁出”。
7. 在“迁出资源”页面，选择“迁出方式”，再选择“请选择要迁入的企业项目”，然后单击“确定”。
8. 迁出完成后，可以在云搜索服务管理控制台集群管理页面，查看修改后的集群企业项目信息。

4.7.4 管理 Logstash 集群的 CA 证书

当Logstash集群连接CSS服务的Elasticsearch集群时，可以跳过本文直接使用默认CA证书，当连接其他需要安全认证的数据源时，则需要参考文本配置自定义证书。

场景描述

- 当Logstash集群连接安全模式+HTTPS协议的Elasticsearch集群时，CSS预置的默认CA证书可以用于信任该服务端。Logstash集群支持[查看默认证书](#)。
- 当Logstash集群连接其他数据源，且需要进行安全认证时，则需要用户上传自定义证书信任服务端，具体操作请参见[上传自定义证书](#)。

查看默认证书

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏选择“集群管理 > Logstash”，进入集群列表页面。
3. 在集群列表，单击集群名称，进入集群基本信息页面。
4. 在左侧导航栏选择“证书管理”。
5. 在“默认证书”页签下查看默认CA证书的信息。

上传自定义证书

1. 准备好自定义证书。
 - 证书名称在4位到32位之间，必须以字母开头，以“.cer|.crt|.rsa|.jks|.pem|.p10|.pfx|.p12|.csr|.der|.keystore”结尾，可以包含字母、数字、中划线、下划线或者小数点，不能包含其他的特殊字符。
 - 证书上传数量不超过50个。
 - 证书文件大小不能超过1M。
2. 登录对象存储服务管理控制台，将自定义证书上传至OBS桶，具体操作请见[上传文件](#)
3. 登录云搜索服务管理控制台。
4. 在左侧导航栏选择“集群管理 > Logstash”，进入集群列表页面。
5. 在集群列表，单击集群名称，进入集群基本信息页面。
6. 在左侧导航栏选择“证书管理”。
7. 选择“自定义证书”页签，单击“上传证书”，在上传证书对话框中选择“OBS桶”和“证书对象”。
 - “OBS桶”：选择存储自定义证书的OBS桶。
 - “证书对象”：单击“选择”，在“选择证书对象”对话框中勾选提前上传至OBS桶的自定义证书，单击“确定”。
8. 单击“确定”，完成自定义证书上传。

自定义证书上传完成后会显示在证书列表中。
9. 当不需要自定义证书时支持删除证书，在自定义证书列表的操作列，单击“删除”，在确认框中单击“确定”，删除证书。

4.7.5 强制重启 Logstash 集群

当Logstash集群由于长时间运行或者其他未知原因导致Logstash故障不可用时，可以通过强制重启VMs集群恢复集群运行。

约束限制

- 重启过程中集群不可用，请谨慎操作。
- 工作中状态的集群，重启过程会主动停止Logstash进程。
 - 当管道列表“是否保持常驻”为“否”时，重启过程中会将所有“运行中”的管道状态置为“已停止”。
 - 当管道列表“是否保持常驻”为“是”时，重启过程中会触发Logstash进程恢复机制，将工作中的管道状态置为“恢复中”，如果十分钟内重新拉起Logstash进程，管道状态恢复为“工作中”，否则置为“失败”状态。

操作步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏选择“集群管理 > Logstash”，进入集群列表页面。
3. 在集群列表，单击集群操作列“更多 > 强制重启”。
4. 在“强制重启集群VMs”对话框中，确认重启信息，单击“确定”。
集群重启过程中，集群状态为“处理中”，任务状态为“重启中”。如果集群状态变更为“可用”，表示集群已重启成功。

4.7.6 删除 Logstash 集群

当用户已完成Logstash数据迁移业务，无需继续使用某一集群时，可删除集群释放资源。

约束限制

删除集群时，会清理集群业务数据，请谨慎操作。

删除按需计费的集群

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，选择对应的集群类型，进入集群列表界面。
3. 在对应集群的“操作”列中单击“更多>删除”。
4. 在弹出的确认提示框中，输入需要删除的集群名称，单击“确定”完成集群删除。

删除包年/包月的集群

包年包月计费模式的集群需要通过退订包周期删除集群。集群退订后，资源和数据将会被清空，且无法恢复。请务必确认完成数据备份后再提交退订。

1. 在集群管理页面，选择需要退订的集群。
2. 在操作列单击“更多” > “退订/释放”。
3. 输入需要退订/释放的集群名称，单击“确定”。
进入退订资源页面，可以在该页面核对资源信息以及退费金额。
4. 填写退订原因，勾选相关协议后，单击“退订”。
在弹出确认退订提示信息后，再次单击“退订”。

📖 说明

当集群处于生效中状态时，则走退订流程，此时会产生一个订单进行退费，然后删除集群。当集群处于已过期或者已冻结状态时，则走释放流程，直接删除集群。集群退订的使用说明请参考[费用中心](#)相关描述。

4.8 Logstash 集群监控与日志管理

4.8.1 CES 中 Logstash 集群支持的监控指标

云监控服务CES支持实时监控云搜索服务集群的核心指标，方便用户掌握集群的指标信息，以便及时处理集群的异常状况。

功能说明

本节定义了云搜索服务上报云监控服务的监控指标的命名空间，监控指标列表和维度定义。用户可以通过云监控服务提供管理控制台或[API接口](#)来检索云搜索服务产生的监控指标和告警信息。

📖 说明

- 当Logstash集群的配置中心无管道列表的操作记录，则Logstash集群的监控记录将为空。
- 当管道的“事件”数据动态变化时，监控数据也会同步变化；当管道任务在启动中、停止中或事件数据稳定时，监控数据则无变化。

命名空间

SYS.ES

CSS.CUSTOM

集群监控指标列表

- 监控的**指标ID**、**指标名称**、**指标含义**以及**取值范围**参见[表4-25](#)。
- 监控的**测量对象**：CSS集群。本文介绍的是Logstash集群的监控指标列表。
- 监控的**监控周期**（原始指标）：1分钟

📖 说明

累计值：从节点启动时开始叠加数值，当节点重启后清零重新累计。

表 4-25 云搜索服务支持的监控指标

指标ID	指标名称	指标含义	取值范围
max_jvm_heap_usage	最大JVM堆使用率	CSS集群中各个节点的JVM堆使用率的最大值。 单位：百分比	0-100%

指标ID	指标名称	指标含义	取值范围
max_jvm_young_gc_time	最大JVM Young GC耗时	CSS集群中各个节点的JVM Young GC耗时累计值的最大值。 单位：ms	≥ 0 ms
max_jvm_young_gc_count	最大JVM Young GC次数	CSS集群中各个节点的JVM Young GC次数累计值的最大值。	≥ 0
max_jvm_old_gc_time	最大JVM Old GC耗时	CSS集群中各个节点的JVM Old GC耗时累计值的最大值。 单位：ms	≥ 0 ms
max_jvm_old_gc_count	最大JVM Old GC次数	CSS集群中各个节点的JVM Old GC次数累计值的最大值。	≥ 0
max_cpu_usage	最大CPU利用率	CSS集群中各个节点的CPU利用率的最大值。 单位：百分比	0-100%
max_load_average	最大节点Load值	CSS集群中各个节点在操作系统中1分钟平均排队任务数的最大值。	≥ 0
avg_cpu_usage	平均CPU使用率	CSS集群中各节点CPU利用率的平均值。 单位：百分比	0-100%
avg_load_average	平均节点Load值	CSS集群中各节点在操作系统中1分钟平均排队任务数的平均值。	≥ 0
avg_jvm_heap_usage	平均JVM堆使用率	CSS集群中各节点JVM堆内存使用率的平均值。 单位：百分比	0-100%
avg_jvm_old_gc_count	JVM老年代平均GC次数	CSS集群中各个节点“老年代”垃圾回收的运行次数的累计值的平均值。	≥ 0
avg_jvm_old_gc_time	JVM老年代平均GC时间	CSS集群中各个节点执行“老年代”垃圾回收所花费的时间累计值的平均值。 单位：ms	≥ 0 ms

指标ID	指标名称	指标含义	取值范围
avg_jvm_young_gc_count	JVM年轻代平均GC次数	CSS集群中各个节点“年轻代”垃圾回收的运行次数的累计值的平均值。	≥ 0
avg_jvm_young_gc_time	JVM年轻代平均GC时间	CSS集群中各个节点执行“年轻代”垃圾回收所花费的时间累计值的平均值。 单位：ms	≥ 0 ms
sum_events_in	集群下所有节点经过input插件的数据总数	该指标用于统计所有节点经过input插件的数据总数。	≥ 0
sum_events_filtered	集群下所有节点经过filter插件的数据总数	该指标用于统计所有节点经过过滤处理的数据总数。	≥ 0
sum_events_out	集群下所有节点经过out插件的数据总数	该指标用于统计所有节点经过out插件的数据总数。	≥ 0

节点监控指标列表

- 监控的指标ID、指标名称、指标含义以及取值范围参见表4-26。
- 监控的测量对象：CSS集群 - 云服务节点
- 监控的监控周期（原始指标）：1分钟

表 4-26 云搜索服务节点支持的监控指标

指标ID	指标名称	指标含义	取值范围
jvm_heap_usage	JVM堆使用率	节点JVM堆内存使用率。 单位：百分比	0-100%
cpu_usage	CPU利用率	CPU利用率。 单位：百分比	0-100%
load_average	节点Load值	操作系统中1分钟平均排队任务数。	≥ 0
jvm_old_gc_count	JVM老年代总GC次数	“老年代”垃圾回收的运行次数。	≥ 0
jvm_old_gc_time	JVM老年代总GC时间	执行“老年代”垃圾回收所花费的时间。 单位：ms	≥ 0 ms

指标ID	指标名称	指标含义	取值范围
jvm_young_gc_count	JVM年轻代总GC次数	“年轻代”垃圾回收的运行次数。	≥ 0
jvm_young_gc_time	JVM年轻代GC时间	执行“年轻代”垃圾回收所花费的时间。 单位：ms	≥ 0 ms
events_in	当前节点经过input插件的数据数	该指标用于统计当前节点经过input插件的数据数。	≥ 0
events_filtered	当前节点经过filtered插件的数据数	该指标用于统计当前节点过滤处理的数据数。	≥ 0
events_out	当前节点经过out插件的数据数	该指标用于统计当前节点经过out插件的数据数。	≥ 0

Logstash 管道监控指标列表

- 监控的**指标ID**、**指标名称**、**指标含义**以及**取值范围**参见[表4-27](#)。
- 监控的**测量对象**：CSS集群 - Logstash管道
- 监控的**监控周期**（原始指标）：1分钟

表 4-27 Logstash 管道的监控指标

指标ID	指标名称	指标含义	取值范围
logstash_pipeline_events_in	当前管道监控周期内经过input插件的数据数。	该指标用于统计当前管道监控周期内经过input插件的数据数。	≥ 0
logstash_pipeline_events_filtered	当前管道监控周期内经过filtered插件的数据数。	该指标用于统计当前管道监控周期内过滤处理的数据数。	≥ 0
logstash_pipeline_events_out	当前管道监控周期内经过out插件的数据数。	该指标用于统计当前管道监控周期内经过out插件的数据数。	≥ 0

维度

表 4-28 维度说明

Key	Value
cluster_id	CSS集群
cluster_id,instance_id	CSS集群节点
cluster_id,instance_id,pipe line_name	CSS集群节点Logstash管道名称
cluster_id,pipe line_name	CSS集群Logstash管道名称

4.8.2 使用 CES 监控 Logstash 集群

云搜索服务支持通过云监控服务CES对已创建成功的集群进行日常监控。配置集群监控后，就可以在CES管理控制台直观查看集群的监控指标数据。

配置集群监控的操作流程：

1. **配置告警规则**：根据实际业务需要对监控指标设置自定义告警规则，当监控指标超过设置的阈值时，会以邮箱、HTTP、HTTPS等方式通知您。
2. **配置监控对象**：为集群或集群中某个节点配置监控指标。
3. **查看监控指标**：您可以选择不同的监控时间周期，查看监控指标数据变化情况。

前提条件

- 集群处于“可用”或“处理中”状态。
- 集群正常运行时长大于10分钟。

推荐配置的监控指标

监控集群的**cpu、jvm使用情况**，推荐重点配置如下监控指标：平均JVM堆使用率、最大JVM堆使用率、平均CPU使用率、最大CPU利用率。

配置告警规则

1. 登录云监控服务CES管理控制台。
2. 左侧导航栏选择“告警 > 告警规则”，进入告警规则列表页面。
3. 在“资源类型”列，筛选“云搜索服务”，查看是否有满足要求的告警规则。

图 4-10 查看告警规则



<input type="checkbox"/>	名称/ID	资源类型	监控对象
<input type="checkbox"/>	alarm- al1654480875465a...	云搜索服务	CSS集群 指定资源

如果没有，请参考云监控服务CES的[创建告警规则](#)和[通知](#)，新建CSS服务的告警规则。其中，“资源类型”和“维度”参数的填写说明请参见[表4-29](#)，其他参数可以根据CES服务的参数说明自定义。

表 4-29 告警内容的配置说明

参数	参数解释	配置说明
资源类型	配置告警规则监控的服务名称。	选择云搜索服务。
维度	用于指定告警规则对应指标的维度名称。	CSS支持2个维度，根据实际需要选择维度。 <ul style="list-style-type: none"> ● CSS集群：以集群维度指定告警规则。 ● CSS集群 - 云服务节点：以集群中的某个节点维度指定告警规则。

配置监控对象

1. 参考云监控服务CES的[创建监控面板](#)，创建一个监控面板。如果已有监控面板，可以跳过该步骤。
2. 参考云监控服务CES的[添加监控视图](#)，添加CSS监控视图。

其中，“资源类型”和“维度”参数的填写说明请参见[表4-30](#)，其他参数可以根据CES服务的参数说明自定义。

表 4-30 监控视图的配置说明

参数	参数解释	配置说明
资源类型	添加监控视图的服务名称。	选择云搜索服务。
维度	指定监控的维度名称。	CSS支持2个维度，根据实际需要选择维度。 <ul style="list-style-type: none"> ● CSS集群：以集群维度监控。 ● CSS集群 - 云服务节点：以集群中的某个节点维度监控。

查看监控指标

1. 登录云搜索服务管理控制台。
2. 选择目标集群，单击操作列“监控信息”查看监控指标。

图 4-11 查看监控信息

名称ID	集群状态	任务状态	版本	创建时间	企业项目	内网访问地址	计费模式	操作
	可用	-	7.10.2 elasticsearch	2024/03/20 11:32:14...	default		按需计费	Kibana 监控信息 更多

3. 选择待查看的时间段页签。
4. 查看监控指标数据。


4.8.3 查询和管理 Logstash 集群日志

为了方便用户使用日志定位问题，云搜索服务提供了日志备份和日志查询功能。日志备份可以定期将集群的日志存储在OBS桶中，通过OBS可以直接下载需要的日志文件，进行问题分析定位。

日志备份在OBS桶中需要额外收费，具体收费以[对象存储服务计费说明](#)为准。

查询日志

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要查询日志的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”。
4. 在日志管理页面进行日志查询。

5. 选择需要查询的节点、日志类型和日志级别信息后，单击 ，显示查询结果。

查询日志时，是从最近时刻的1万条日志中进行匹配，查询结果最多显示100条。

备份日志

日志备份可以定期将CSS集群的日志存储到OBS桶中。

1. 登录云搜索服务管理控制台。
2. 在“集群管理”页面，单击需要配置日志备份的集群名称，进入集群基本信息页面。
3. 左侧导航栏，选择“日志管理”，在“日志备份开关”右侧单击开关，打开集群的日志管理功能。
4. 在“编辑日志备份配置”弹窗中，完成参数配置。

弹窗中默认填写了云搜索服务自动为用户创建的“OBS桶”、“备份路径”和“IAM委托”，用于日志备份。支持用户参考[表4-31](#)修改默认值。

如果集群已经启用了日志备份功能，也可以单击“日志备份配置”右侧的编辑图标，在“编辑日志备份配置”窗口，参考[表4-31](#)更新日志备份的配置参数。

表 4-31 日志备份配置

参数	说明	注意事项
“OBS桶”	选择日志存储的OBS桶。单击右侧的“创建桶”支持新建OBS桶。	OBS桶的所在区域必须跟集群的所在区域保持一致。 说明 如果是子账号，需要同时设置GetBucketStoragePolicy、GetBucketLocation、ListBucket、ListAllMyBuckets权限，才能看到OBS桶。

参数	说明	注意事项
“备份路径”	填写日志在OBS桶中的存放路径。	备份路径配置规则： <ul style="list-style-type: none"> • 备份路径不能包括符号“\:*?"<> ”。 • 备份路径不能以“/”开头。 • 备份路径不能以“.”开头或结尾。 • 备份路径的总长度不能超过1023个字符。
“IAM委托”	选择IAM委托，指当前账号授权云搜索服务访问或维护存储在OBS中数据。单击右侧的“创建委托”支持新建委托。	IAM委托需满足如下条件： <ul style="list-style-type: none"> • “委托类型”选择“云服务”。 • “云服务”选择“Elasticsearch”或者“云搜索服务 CSS”。 • 必选策略：“OBS Administrator”

5. 备份日志。

- 自动备份日志。

在“自动备份开关”右侧，单击开关，开启自动备份日志功能。

开启“自动备份开关”后，在“修改日志备份策略”弹窗中设置“备份开始时间”。设置成功后，系统会按照设置的时间进行自动备份。

打开“自动备份开关”后，单击开关右侧的编辑图标，可以修改“备份开始时间”。

- 手动备份日志。

选择“日志备份”页签，单击“开始备份”，在弹出的确认提示框中单击“确定”，开始备份日志。

日志备份列表中的“任务状态”为“Successful”时，表示日志备份成功。

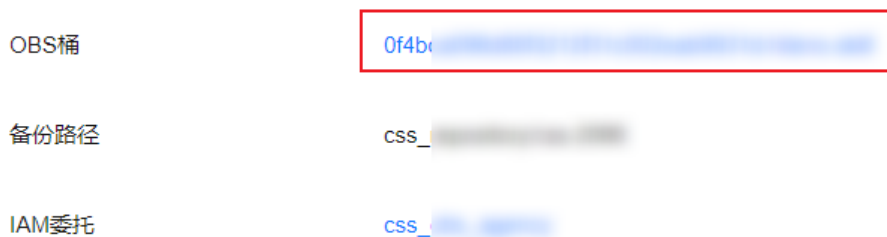
📖 说明

如果日志备份失败，可以在“日志备份”页签的右侧单击查看“失败任务数”，了解失败原因。失败任务数最多显示10条，当日志备份功能关闭或集群删除时，失败记录也会被清空。

6. 查看日志文件。

日志备份成功后，用户可以单击“OBS桶”跳转到桶列表，找到存放备份日志的桶查看日志文件。

图 4-12 进入 OBS



日志文件介绍

Logstash集群备份的日志文件主要包括废弃操作日志、运行日志。

表 4-32 日志类型信息

日志名称	描述
logstash-deprecation.log	弃用操作的日志记录。
logstash-plain.log	Logstash运行日志。

4.9 查看 Logstash 集群审计日志

通过云审计服务，您可以记录与云搜索服务相关的操作事件，便于日后的查询、审计和回溯。

前提条件

已开通云审计服务。

支持审计的关键操作列表

表 4-33 支持审计的关键操作列表

操作名称	资源类型	事件名称
创建集群	cluster	createCluster
删除集群	cluster	deleteCluster
扩容集群	cluster	roleExtendCluster
重启集群	cluster	rebootCluster

查看审计事件

在云审计服务管理控制台，用户创建管理类追踪器后，系统会开始记录用户对云服务资源的操作，创建数据类追踪器后，系统会开始记录用户对OBS桶中数据的操作。云审计服务管理控制台会保存最近7天的操作记录。

当需要在云审计服务管理控制台查看或导出最近7天的操作记录时，请参见[查询审计事件](#)。