云搜索服务

故障排除

文档版本	01
发布日期	2025-01-23

华为云计算技术有限公司

商标声明


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 访问集群类

1.1 无法正常打开 Kibana

问题现象

Es-event集群单击进入kibana后，会出现一直卡在加载页面中，不能进入Kibana控制台。

原因分析

浏览器缓存导致，清理缓存。

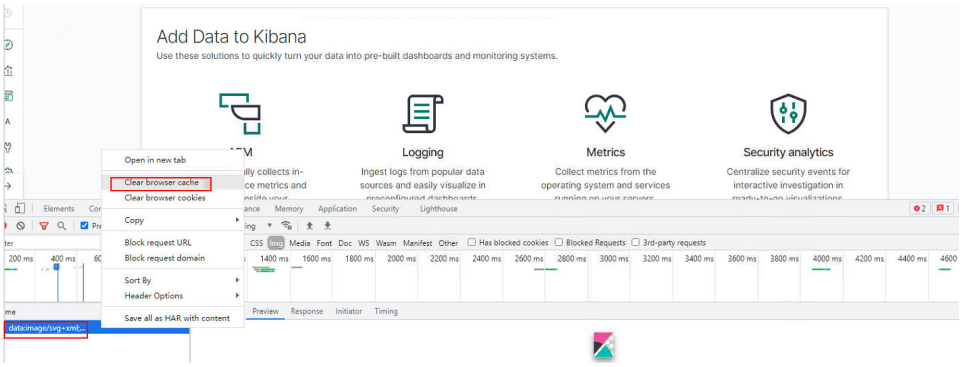
处理步骤

1. 登录云搜索服务管理控制台。
2. 在左侧导航栏，单击“集群管理”。
3. 在集群对应的“操作”列，单击“Kibana”，打开Kibana界面。

   说明

   - 如果您开启了安全模式，登录时候需要输入用户名和密码。用户名默认为admin，密码为创建集群时设置的密码。
   - 如果忘记密码，可以在集群详情页面的“配置信息”区域，单击“重置密码”后的“重置”，设置并确认新的管理员密码。
4. 在kibana页面按F12。
5. 单击“Network”，选中“data:image”，右键选择“clear browser cache”，弹出对话弹框，单击确定，关闭Kibana界面。

图 1-1 关闭 Kibana 界面



6. 在集群对应的“操作”列，单击“Kibana”即可访问。

1.2 Elasticsearch 针对 filebeat 配置调优

问题现象

filebeat是性能非常出色的文件采集工具，绝大多数的业务日志可以很容易的在1秒内收集至elasticsearch内，但是个别日志量大的业务日志无法及时收集，按照官方的默认配置通常1核CPU分配给filebeat时，写ES的速率低于1M/S，这里可以针对filebeat.yml配置文件做优化，提高写入ES的性能。

原因分析

filebeat.yml的默认配置比较保守，在日志量很大的业务场景，需要修改filebeat.yml参数进行调优。

处理步骤

1. 针对filebeat.yml配置文件做参数优化，调整input端配置：

```
#根据实际情况调大harvester_buffer_size参数（该参数是指每个harvester监控文件时，使用的buffer大小）。
harvester_buffer_size:40960000
#根据实际情况调大filebeat.spool_size参数（该参数是指spooler的大小，一次Publish，上传多少行的日志量）。
filebeat.spool_size:250000
#根据实际情况调整 filebeat.idle_timeout参数（该参数是指spooler的超时时间，如果到了超时时间，不论是否到达容量阈值，spooler会清空发送出去）。
filebeat.idle_timeout:1s
```

2. 针对filebeat.yml配置文件做参数优化，调整output.elasticsearch端配置：

```
#根据实际情况将worker参数调整为跟ES个数一致（该参数是指对应ES的个数，默认1）。
worker:1
#根据实际情况调大bulk_max_size参数（该参数是指单个elasticsearch批量API索引请求的最大事件数，默认是50）。
bulk_max_size:15000
#根据实际情况调整flush_interval参数（该参数是指新事件两个批量API索引请求之间需要等待的秒数，如果bulk_max_size在该值之前到达，额外的批量索引请求生效）。
flush_interval:1s
```

1.3 Spring Boot 使用 Elasticsearch 出现 Connection reset by peer 问题

问题现象

Spring Boot服务使用Elasticsearch RestHighLevelClient链接Elasticsearch运行一段时间就会出现Connection reset by peer，TCP连接中断，业务数据写入失败。

原因分析

连接关闭有很多原因，是Elasticsearch服务器端不能完全控制的。例如，有可能关闭了连接，有可能有防火墙，交换机，VPN等，也有可能是keepalive设置问题，更换了连接服务器节点，网络不稳定等。

处理步骤

如果出现这种情况可以选择如下多种方式解决问题。

- 方法一

  修改RestHighLevelClient连接请求的超时时间 ，默认1000ms可以尝试增加到10000ms。

  ```
  RestClientBuilder builder = RestClient.builder(new HttpHost(endpoint, port))
        .setHttpClientConfigCallback(httpClientBuilder->
  httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider))
        .setRequestConfigCallback(requestConfigBuilder ->
  requestConfigBuilder.setConnectTimeout(10000).setSocketTimeout(60000));
     return new RestHighLevelClient(builder);
  ```

  单个请求修改：request.timeout(TimeValue.timeValueSeconds(60))。

- 方法二

  设置RestHighLevelClient keepalive时间，建议设置到15分钟。

- 方法三

  针对代码层面捕获异常情况，建议重试请求。

参考

- TCP长连接和短连接

  TCP协议中有长连接和短连接之分。短连接在数据包发送完成后会自己断开，长连接在发包完成后， 会在一定的时间内保持连接，即通常所说的Keepalive（存活定时器）功能。

- TCP保活机制

  保活机制是由一个保活计时器实现的。当计时器被激发，连接一端将发送一个保活探测报文， 另一端接收报文的同时会发送一个ACK作为响应。 如果客户端无响应，服务器将中断连接，否则会重置保活计时器。

  服务器端Keepalive设置时间30m，Linux有三个参数可以控制保活时间：tcp_keepalive_time（开启Keepalive的闲置时长）、 tcp_keepalive_intvl（Keepalive探测包的发送间隔）、tcp_keepalive_probes （如果对方不予应答，探测包的发送次数）。

- http-keepalive

http-keepalive是保证一个TCP连接尽可能传递多的报文，每次交互一个报文后就会更新http-keepalive时间。如果http-keepalive时间超时，意味这个这段时间client和server没有报文交互，本端会主动关闭释放连接。

tcp-keepalive是一种探测TCP连接状态的保活机制，TCP连接建立后如果不主动关闭，client和server没有发生异常，这个连接理论上是一直存在的，http-keepalive是保证一个TCP连接上尽可能传递更多的报文，如果http-keepalive时间内没有报文交互则会主动关闭连接。

1.4 为什么集群创建失败

集群创建失败原因有如下4种：

- 资源配额不足，无法创建集群。建议申请足够的资源配额。
- 如果集群配置信息中，“安全组”的“端口范围/ICMP类型”不包含“9200”端口，导致集群创建失败。请修改安全组信息或选择其他可用安全组。
- 7.6.2以及7.6.2之后的版本，集群内通信端口9300默认开放在用户VPC的子网上面。创建集群时需要确认所选安全组是否放通子网内的9300通信端口，如果未放通，请修改安全组信息或选择其他可用安全组。
- 权限不足导致集群创建失败。建议参考**权限管理**获取权限，然后创建集群。

1.5 Elasticsearch 集群出现写入拒绝“Bulk Reject”，如何解决?

问题现象

集群在某些情况下会出现写入拒绝率增大“bulk reject”的现象，具体表现为bulk写入时，会有类似以下报错：

```
[2019-03-01 10:09:58][ERROR]rspItemError: {
    "reason": "rejected execution of org.elasticsearch.transport.TransportService$7@5436e129 on EsThreadPoolExecutor[bulk, queue capacity = 1024, org.elasticsearch.common.util.concurrent.EsThreadPoolExecutor@6bd77359[Running, pool size = 12, active threads = 12, queued tasks = 2390, completed tasks = 20018208656]]",
    "type": "es_rejected_execution_exception"
}
```

问题分析

引起bulk reject的大多原因是shard容量过大或shard分配不均，具体可通过以下方法进行定位分析。

1. 检查分片（shard）数据量是否过大。

   单个分片数据量过大，可能引起Bulk Reject，建议单个分片大小控制在20GB - 50GB左右。可在kibana控制台，通过如下命令查看索引各个分片的大小。

   ```
   GET _cat/shards?index=index_name&v
   ```

2. 检查分片数是否分布不均匀。

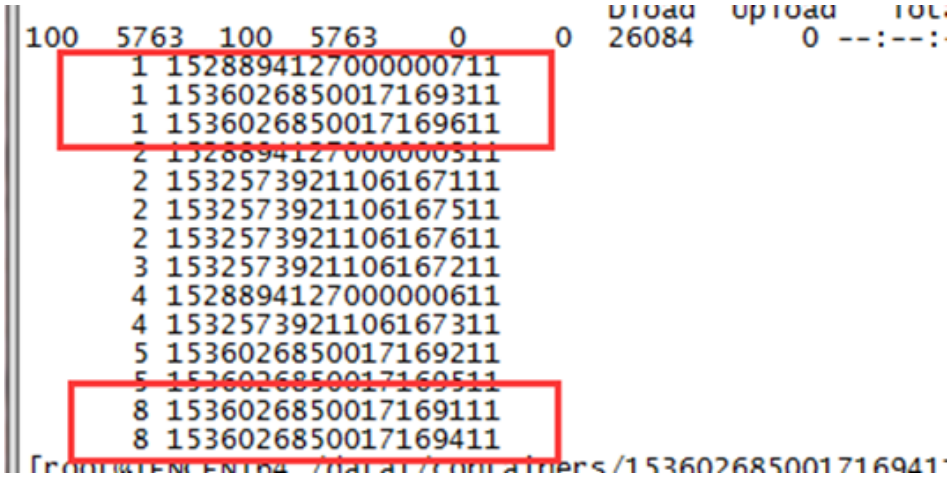
   提供如下两种方式查看：

   a. 通过CSS控制台集群详情页的“集群监控”中的“节点状态”查看，具体操作可参见**查看监控指标**。

b. 通过CURL客户端，查看集群各个节点的分片个数。

```
curl "$p:$port/_cat/shards?index={index_name}&s=node,store:desc" | awk '{print $8}' | sort | uniq -c | sort
```

结果如下图所示：



说明

第一列为分片个数，第二列为节点ID，有的节点分片为1，有的为8，分布极不均匀。

解决方案

- 如果问题是由分片数据量过大导致。
  分片大小可以通过index模板下的“number_of_shards”参数进行配置。

  说明

  模板创建完成后，再次新创建索引时生效，旧的索引不能调整。

- 如果问题是由分片数分布不均匀导致。
  临时解决方案：

  a. 可以通过如下命令设置“routing.allocation.total_shards_per_node”参数，动态调整某个index解决。

```
PUT <index_name>/_settings
{
   "settings": {
      "index": {
         "routing": {
            "allocation": {
               "total_shards_per_node": "3"
            }
         }
      }
   }
}
```

  说明

  “total_shards_per_node”要留有一定的buffer，防止机器故障导致分片无法分配（例如10台机器，索引有20个分片，则total_shards_per_node设置要大于2，可以取3）。

  b. 索引产生前设置。

  通过索引模板，设置其在每个节点上的分片个数。

```
PUT _template/<template_name>
{
```

```
  "order": 0,
  "template": " { index_prefix@ } *",  //要调整的index前缀
  "settings": {
     "index": {
        "number_of_shards": "30", //指定index分配的shard数，可以根据一个shard 30GB左右的空间来分配
        "routing.allocation.total_shards_per_node": 3 //指定一个节点最多容纳的shards数
     }
  },
  "aliases": {}
}
```

1.6 Elasticsearch 集群创建 index pattern 卡住，如何解决?

问题现象

在Kibana的“Dev Tools”页面，执行**GET .kibana/_settings**命令查询kibana索引的状态，如果状态为“true”，说明集群磁盘过高。

问题原因

索引置为只读状态导致。

解决方案

1. 将kibana索引只读状态改为“false”后再执行创建**index pattern**。
2. 在kibana的“Dev Tools”页面，执行如下命令：

```
PUT .kibana/_settings
{
  "index": {
     "blocks": {
        "read_only_allow_delete": false
     }
```



```
  }
}
```

1.7 云搜索控制台页面提示系统繁忙

问题描述

登录云搜索控制台，单击集群列表，显示“系统繁忙，请稍后重试或拨打客服电话4000-955-988”和“当前策略不允许css:cluster:list”。

问题原因

该问题是由于此账号没有云搜索服务的读权限导致的，需要主账号赋予此账号需要使用功能的IAM权限。

解决方案

为IAM账号进行权限授权，更多信息请参见**权限管理**。

1.8 Elasticsearch 集群报错：unassigned shards all indices

问题描述

Elasticsearch集群报错unassigned shards all indices，集群状态为red。

原因分析

当前集群存在未分配的shard。

解决方案

在Kibana的“Dev Tools”页面，执行如下命令：

```
POST /_cluster/reroute?retry_failed=true
```

1.9 es-head 插件连接 Elasticsearch 集群报跨域错误

解决方案

1. 在安装es-head的云主机上测试网络是否连通。
2. 网络连通后，登录云搜索服务管理控制台。
3. 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。
4. 选择“参数配置”，单击“编辑”，将“http.cors.allow-origin” 参数更改为“"*"”。

1.10 单节点集群打开 Cerebro 界面显示告警

原因分析

单节点集群索引默认有副本，但是副本无法下发请求，所以显示告警。

解决方案

在Kibana的“Dev Tools”页面，执行以下命令将索引副本数量修改为“0”。

```
PUT _all/_settings
{
"index" :
{
"number_of_replicas" : 0
}
}
```

1.11 ECS 无法连接到集群

遇到该问题，请按照如下操作步骤排查解决。

1. 先确认ECS实例和集群是否在同一个VPC。
   - 如果在，执行步骤2。
   - 如果不在，需要重新创建ECS实例，使之和集群在同一个VPC下。
2. 查看集群的安全组的出方向和入方向是否已允许9200端口（TCP协议），或者允许的端口范围已包含9200端口（TCP协议）。
   - 如果是，执行步骤3。
   - 如果不是，请前往VPC页面，设置“安全组”的出方向和入方向已允许9200端口或允许的端口范围已包含9200端口。
3. 查看ECS实例是否添加安全组。
   - 如果有，检查安全组的配置规则是否满足要求，在集群“基本信息”页面，可以查看“安全组信息”。然后执行步骤4。
   - 如果没有，从ECS的实例详情页面，进入VPC页面，选择“安全组”，添加安全组。
4. 在ECS实例上，测试是否可以正常连接到集群。

   ***ssh** <节点的内网访问地址和端口号>*

   说明

   当集群包含多个节点时，需要逐个节点测试是否可以正常连接到该集群中的每个节点。
   - 如果可以通信，说明网络是正常的。
   - 如果端口不通，请联系技术支持协助排查。

2 集群不可用

2.1 集群不可用排查指导

问题现象

云搜索服务的集群列表中，“集群状态”出现“不可用”。

图 2-1 集群不可用

原因分析及处理方法

- 如果集群列表的任务状态显示“冻结”，可能是**集群冻结状态导致集群不可用**。
- 如果集群列表的任务状态显示“配置错误，重启失败”，可能是**X-pack参数配置导致集群不可用**。
- 如果集群节点的日志内容存在警告“master not discovered or elected yet, an election requires at least 2 nodes with ids [xxx, xxx, xxx, ...], have discovered [xxx...] which is not a quorum”，可能是**安全组策略设置不合理导致集群不可用**。
- 如果集群节点的日志内容存在明显的关于插件的报错“fatal error in thread [main], exitingjava.lang. NoClassDefFoundError: xxx/xxx/.../xxxPlugin at ...”，可能是**插件不兼容导致集群不可用**。
- 如果集群的健康状态为红色和且“unassigned shards”不为0，表示集群存在无法分配的索引分片，是**分片未正常分配导致集群不可用**。
- 如果集群进行备份恢复或集群迁移操作后，出现的不可用现象，可能是**数据类型不兼容导致集群不可用**。
- 如果集群节点的日志内容存在报错“OutOfMemoryError”和警告“[gc][xxxxx] overhead spent [x.xs] collecting in the last [x.xs]”，可能是**集群负载过高导致集群不可用**。

2.2 集群冻结状态导致集群不可用

问题现象

“集群状态”为“不可用”，集群的“任务状态”为“冻结”。

图 2-2 集群冻结状态

原因分析

集群出现冻结状态的原因是账户欠费或包年包月集群的订购周期已到期。

处理步骤

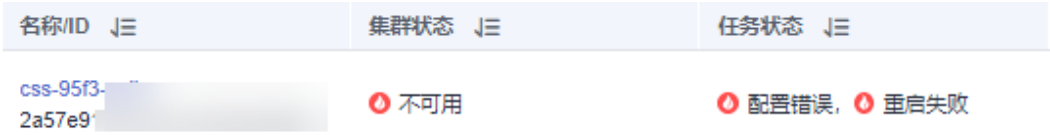
- **按需计费集群**
  a. 在华为云控制台上方单击“费用与成本”进入费用中心。
  b. 在“总览”页面查看账户的欠费情况。
     - 如果是IAM用户，请登录IAM账号对应的账号进行费用充值。
     - 如果是账号用户，请直接单击“充值”前往费用充值页面。
  c. 确认账户有现金预算后，等待集群自动恢复可用状态。
- **包年包月集群**
  a. 在华为云控制台上方单击“费用与成本”进入费用中心。
  b. 导航栏选择“订单管理 > 续费管理”。
  c. 在续费管理页面，“到期时间”选择“已冻结”，“产品类型”选择“云搜索服务”，找到已冻结的集群周期包。
  d. 在已冻结集群周期包对应的操作列，单击“续费”进入续费页面，根据需要续费周期。
  e. 确认集群周期包的“状态”由“已冻结”变更为“使用中”后，等待集群自动恢复可用状态。

2.3 X-pack 参数配置导致集群不可用

问题现象

“集群状态”为“不可用”，集群的“任务状态”为“配置错误，重启失败”。

图 2-3 集群配置错误



原因分析

集群可能配置了X-pack相关的自定义参数导致集群不可用。CSS服务不支持X-pack功能。

处理步骤

1. 在集群管理页面，单击不可用的集群名称，进入集群基本信息页面。
2. 选择“参数配置”，展开“自定义”模块，查看是否存在X-pack相关的自定义参数。

   说明

   X-pack相关的自定义参数示例：
   - “xpack.security.enabled”：“true”
   - “xpack.security.http.ssl.enabled”：“true”
   - “xpack.security.http.ssl.keystore.path”：“http.p12”

   - 是，则执行下一步。
   - 否，联系技术支持定位参数配置问题。

3. 删除X-pack相关的自定义参数。
   a. 单击“编辑”，在需要删除的参数右侧单击“删除”。
   b. 单击“提交”，在“提交配置”弹窗中，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。
   c. 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。
   d. 集群重启成功后，集群恢复可用。

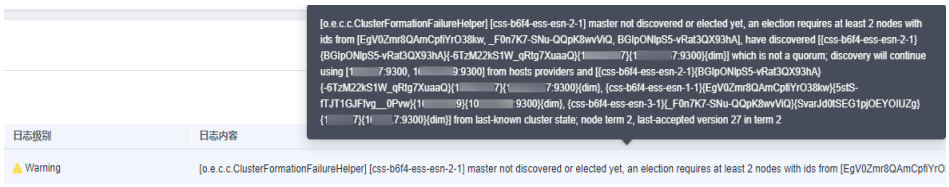
2.4 安全组策略设置不合理导致集群不可用

问题现象

“集群状态”为“不可用”。

单击集群名称进入集群基本信息页面，选择“日志管理”，单击“日志查询”页签，可见日志内容存在警告“master not discovered or elected yet, an election requires at least 2 nodes with ids [xxx, xxx, xxx, ...], have discovered [xxx...] which is not a quorum”。

图 2-4 节点报错日志示例



原因分析

出现以上报错日志表示集群各节点之间无法通信，导致集群无法进行选主，可能原因是集群当前所选安全组未放通9300端口。

说明

云搜索服务在7.6.2及以上的版本，集群内通信端口9300默认开放在用户VPC的子网上。集群所选安全组需要放通子网内的9300通信端口才能保证节点之间通信。

处理步骤

1. 在集群管理页面，单击不可用的集群名称，进入集群基本信息页面。
2. 单击“配置信息”中的安全组名称，进入当前集群所选安全组的基本信息页面。
3. 分别查看“入方向规则”和“出方向规则”页签下，是否存在“策略”为“允许”，“协议端口”为“TCP：9300”，“类型”为“IPv4”的安全组规则。
   - 是，联系技术支持定位集群不可用问题。
   - 否，执行下一步。
4. 修改集群当前所选安全组信息，放通9300通信端口。
   a. 在当前集群所选安全组基本信息界面，选择“入方向规则”页签。
   b. 单击“添加规则”，在添加入方向规则对话框设置“优先级”为“100”，“策略”选择“允许”，“协议端口”选择“基本协议/自定义TCP”，端口填写“9300”，“类型”选择“IPv4”，“源地址”选择“安全组”下的集群当前安全组名称，即同安全组内放通。

   图 2-5 添加安全组规则

   c. 单击“确定”即可完成放通9300端口的设置。

d. 同样的步骤，在“出方向规则”页签添加放通9300端口的设置。

5. 安全组放通9300端口后，等待集群自动恢复可用状态。

2.5 插件不兼容导致集群不可用

问题现象

安装自定义插件后重启集群，“集群状态”变为“不可用”。

单击集群名称进入集群基本信息页面，选择“日志管理”，单击“日志查询”页签，可见日志内容存在明显的关于插件的报错“fatal error in thread [main], exitingjava.lang. NoClassDefFoundError: xxx/xxx/.../xxxPlugin at ...”。

图 2-6 节点报错日志示例

说明

CSS服务已下线自定义插件功能，但历史版本的集群可能还装有自定义插件，只有这类集群可能出现该故障。

原因分析

可能是安装的自定义插件与CSS集群版本不兼容，导致Elasticsearch进程无法正常启动。

处理步骤

1. 在集群管理页面，单击不可用的集群名称，进入集群基本信息页面。
2. 选择“插件管理”，单击“自定义插件列表”页签。确认是否继续使用该自定义插件。
   - 是，删除插件后重新安装。
     i. 在自定义插件操作列表，卸载并删除插件。
     ii. 根据节点日志的报错信息解决插件存在的问题，无法自行解决时可联系技术支持。
     iii. 插件问题解决后，在自定义插件操作列表，上传并安装插件。当“插件状态”为“已安装，待重启集群后生效”时，表示插件安装成功。
   - 否，删除插件。
     在自定义插件操作列表，卸载并删除插件。
3. 返回集群列表，单击集群操作列的“更多 > 重启”，集群重启成功后，集群恢复可用。

2.6 分片未正常分配导致集群不可用

问题现象

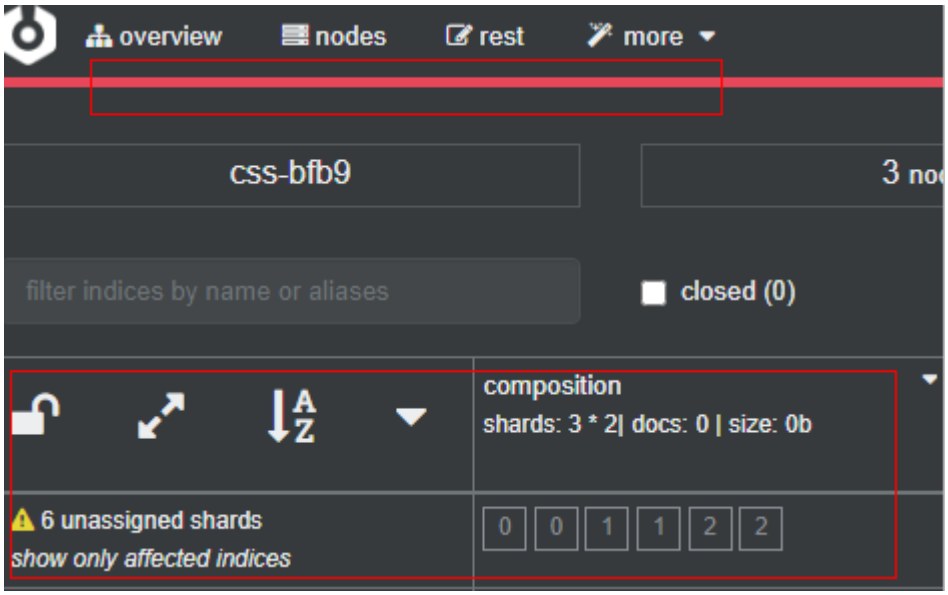
“集群状态”为“不可用”。

在Kibana的“Dev Tools”页面，执行命令**GET _cluster/health**查看集群健康状态，结果中“status”为“red”，“unassigned_shards”不为0。或者在“Cerebro”可视化页面，单击“overview”查看索引分片在各数据节点的分配情况，可见集群状态为红色和“unassigned shards”不为0，表示集群存在无法分配的索引分片。

图 2-7 集群健康状态


```
{
  "cluster_name" : "css-bfb9",
  "status" : "red",
  "timed_out" : false,
  "number_of_nodes" : 3,
  "number_of_data_nodes" : 3,
  "active_primary_shards" : 19,
  "active_shards" : 38,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 6,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 86.36363636363636
}
```


图 2-8 cerebro 可视化界面



原因分析

集群出现不可用状态的原因是集群有索引分片未正常分配。

处理步骤

步骤一：确认集群不可用原因

1. 通过Kibana接入故障集群，在Kibana的“Dev Tools”页面，执行命令**GET /_recovery?active_only=true**查看集群是否在进行副本恢复：
   - 返回“{"index_name":{"shards":[{"id":25,"type":"...”，代表集群存在正在进行副本恢复的索引。等待副本恢复完毕，如果集群状态仍为“不可用”，则执行下一步。
   - 返回“{ }”：代表集群未进行副本恢复，则执行下一步。
2. 执行命令**GET _cluster/allocation/explain?pretty**查看索引分片未分配的原因，根据返回信息进行筛选。

表 2-1 参数说明

参数	描述
index	索引名称
shard	分片标号
current_state	分片当前状态
allocate_explanation	分片分配解释
explanation	解释说明

表 2-2 不同故障说明

现象	原因	处理步骤
“explanation”中存在“no allocations are allowed due to cluster setting [cluster.routing.allocation.enable=none]”	集群当前设置的allocation策略禁止所有分片分配。	参考“shard allocation策略配置错误”之**▪cluster.routing.allocat...**
“explanation”中存在“too many shards [3] allocated to this node for index [write08]index setting [index.routing.allocation.total_shards_per_node=3]”	集群当前设置的单个索引的分片允许分配给每个数据节点的分片数值过小，不满足索引分片的分配要求。	参考“shard allocation策略配置错误”之**▪index.routing.allocatio...**

现象	原因	处理步骤
“explanation”中存在“too many shards [31] allocated to this node, cluster setting [cluster.routing.allocation.total_shards_per_node=30]”	集群当前设置的集群所有索引分片允许分配给每个数据节点的分片数值太小。	参考“shard allocation策略配置错误”之**▪cluster.routing.allocat...**
“explanation”中存在“node does not match index setting [index.routing. allocation. include] filters [box_type:"hot"]”	索引分片需要下发到标记为“hot”的数据节点，而集群中所有数据节点都没有打这个标记时，分片无法下发。	参考“shard allocation策略配置错误”之**▪index.routing.allocatio...**
“explanation”中存在“node does not match index setting [index.routing. allocation. require] filters [box_type:"xl"]”	索引分片需要下发到特定标记的数据节点，而集群中所有节点都没有打这个标记时，分片便无法下发。	参考“shard allocation策略配置错误”之**▪index.routing.allocatio...**
“explanation”中存在“[failed to obtain in-memory shard lock]”	这种情况一般出现在有节点短暂离开集群，然后马上重新加入，并且有线程正在对某个shard做bulk或者scroll等长时间的写入操作，等节点重新加入集群的时候，由于shard lock没有释放，master无法allocate这个shard。	参考**•shardlock错误**
“explanation”中存在“node does not match index setting [index.routing.allocation.include] filters [_tier_preference:"data_hot OR data_warm OR data_cold"]”	集群的某个索引设置的参数与版本不匹配。	参考**•索引参数版本不匹配**
“explanation”中存在“cannot allocate because all found copies of the shard are either stale or corrupt”	集群的索引分片数据被损坏。	参考“主分片数据损坏”**•主分片数据损坏**

现象	原因	处理步骤
“explanation”中存在“the node is above the high watermark cluster setting [cluster.routing.allocation.disk.watermark.high=90%], using more disk space than the maximum allowed [90.0%], actual free: [6.976380997419324%]”	节点的磁盘使用率已超过磁盘空间允许的最大值。	参考“磁盘使用率过高”•**磁盘使用率过高**

步骤二：根据不同的问题现象处理故障

- **shard allocation策略配置错误**
  - cluster.routing.allocation.enable参数
    i. 返回结果中“explanation”如下，表示集群当前设置的allocation策略禁止所有分片分配导致。

    图 2-9 allocation.enable 参数配置错误

    ```
    "deciders" : [
      {
        "decider" : "enable",
        "decision" : "NO",
        "explanation" : "no allocations are allowed due to cluster setting
          [cluster.routing.allocation.enable=none]"
      }
    ```

    ii. 在Kibana的“Dev Tools”页面，执行命令将“enable”设置为“all”，允许所有分片进行分配。

    ```
    PUT _cluster/settings
    {
      "persistent": {
        "cluster": {
          "routing": {
            "allocation.enable": "all"
          }
        }
      }
    }
    ```

    说明

    index级别会覆盖cluster级别配置，参数设置含义如下：

    - **all** - (默认) 所有类型均允许allocation。
    - **primaries** - 只允许allocation主分片。
    - **new_primaries** - 只允许allocation新创建index的主分片。
    - **none** - 所有的分片都不允许allocation。

    iii. 再执行命令**POST _cluster/reroute?retry_failed=true**手动进行分片分配，等待索引分片分配完成，集群状态变为可用。
  - index.routing.allocation.total_shards_per_node参数。

i. 返回结果中“explanation”如下，表示设置的“index.routing.allocation.total_shards_per_node”值过小，不满足索引的分片分配要求。

图 2-10 index total_shards_per_node 设置错误

```
"deciders" : [
  {
    "decider" : "shards_limit",
    "decision" : "NO",
    "explanation" : "too many shards [3] allocated to this node for index [write08],
      index setting [index.routing.allocation.total_shards_per_node=3]"
  }
]
```

ii. 在Kibana的“Dev Tools”页面，执行命令修改索引在每个节点允许分配的分片数。

```
PUT index_name/_settings
{
 "index": {
  "routing": {
   "allocation.total_shards_per_node": 3
  }
 }
}
```

说明

“index.routing.allocation.total_shards_per_node”的值 = index_name索引分片数 / (数据节点个数 - 1)

参数值应设置稍微大一些，假设集群有10个节点，其中5个数据节点，2个client节点，3个master节点，有个索引的分片数为30，如果将total_shards_per_node值设为4，能分配的shard总数只有4*5=20，分片无法完全分配。5个数据节点，需要分配30个分片，每个节点应最少分配6个分片，防止某数据节点故障脱离，那最少应设置每个节点允许分配8个分片。

iii. 再执行命令**POST _cluster/reroute?retry_failed=true**手动进行分片分配，等待索引分片分配完成，集群状态变为可用。

- cluster.routing.allocation.total_shards_per_node参数。

i. 返回结果中“explanation”如下，表示集群允许分配给每个数据节点的分片数设置太小。

图 2-11 cluster total_shards_per_node 设置错误

```
"deciders" : [
  {
    "decider" : "shards_limit",
    "decision" : "NO",
    "explanation" : "too many shards [31] allocated to this node, cluster setting
      [cluster.routing.allocation.total_shards_per_node=30]"
  }
```

ii. “cluster.routing.allocation.total_shards_per_node”参数为限制集群每个数据节点可分配的分片数量，此参数默认设置为“1000”，在Kibana的“Dev Tools”页面执行如下命令设置“cluster.routing.allocation.total_shards_per_node”参数。

```
PUT _cluster/settings
{
 "persistent": {
  "cluster": {
   "routing": {
    "allocation.total_shards_per_node": 1000
   }
```

```
  }
 }
}
```

iii. 出现此场景大多数是参数使用错误，误将“index.routing.allocation.total_shards_per_node”参数设置为“cluster.routing.allocation.total_shards_per_node”参数。执行如下命令可以设置“index.routing.allocation.total_shards_per_node”参数：

```
PUT index_name/_settings
{
 "index": {
  "routing": {
   "allocation.total_shards_per_node": 30
  }
 }
}
```

说明

两个参数都是限制单个数据节点所能分配的最大分片数。

- “cluster.routing.allocation.total_shards_per_node”是**集群**级别的分片限制。
- “index.routing.allocation.total_shards_per_node”是**索引**级别的分片限制。

iv. 再执行命令**POST _cluster/reroute?retry_failed=true**手动进行分片分配，等待索引分片分配完成，集群状态变为可用。

- index.routing.allocation.include参数。

i. 返回结果中“explanation”如下，表示是将索引分片下发到标记为“hot”的数据节点，而集群中所有数据节点都没有打这个标记时，分片无法下发。

图 2-12 include 参数配置错误

```
"deciders" : [
  {
    "decider" : "filter",
    "decision" : "NO",
    "explanation" : """node does not match index setting [index.routing.allocation.include]
      filters [box_type:"hot"]"""
  }
]
```

ii. 在Kibana的“Dev Tools”页面执行命令取消该配置：

```
PUT index_name/_settings
{
 "index.routing.allocation.include.box_type": null
}
```

iii. 再执行命令**POST _cluster/reroute?retry_failed=true**手动进行分片分配，等待索引分片分配完成，集群状态变为可用。

- index.routing.allocation.require参数。

i. 返回结果中“explanation”如下，表示是将分片下发到特定标记的数据节点，而集群中所有节点都没有打这个标记时，分片便无法下发。

图 2-13 require 参数配置错误

```
"deciders" : [
  {
    "decider" : "filter",
    "decision" : "NO",
    "explanation" : """node does not match index setting [index.routing.allocation.require]
      filters [box_type:"xl"]"""
  },
```

ii. 在Kibana的“Dev Tools”页面执行命令取消该配置：

```
PUT index_name/_settings
{
  "index.routing.allocation.require.box_type": null
}
```

iii. 再执行命令**POST _cluster/reroute?retry_failed=true**手动进行分片分配，等待索引分片分配完成，集群状态变为可用。

- **shard lock错误**

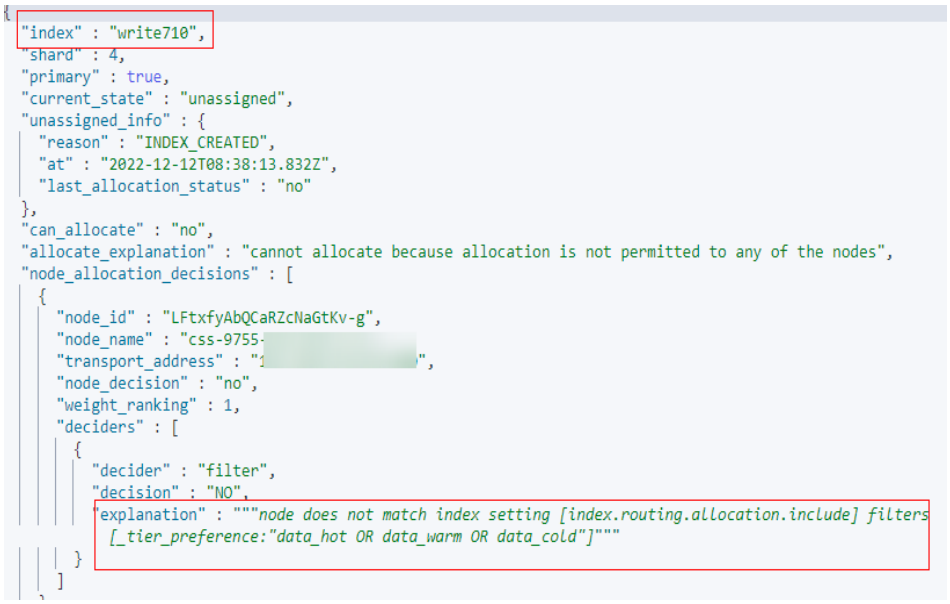
a. 返回结果中“explanation”存在“[failed to obtain in-memory shard lock]”，这种情况一般出现在有节点短暂离开集群，然后马上重新加入，并且有线程正在对某个shard做bulk或者scroll等长时间的写入操作，等节点重新加入集群的时候，由于shard lock没有释放，master无法allocate这个shard。

b. 此现象不会造成分片数据丢失，只需要重新触发一下分配即可。在Kibana的“Dev Tools”页面执行命令**POST /_cluster/reroute?retry_failed=true**手动对未分配分片进行分配，等待索引分片分配完成，集群状态变为可用。

- **索引参数版本不匹配**

a. 返回信息中的索引名称“index”和索引未分配解释“explanation ”：“node does not match index setting [index.routing.allocation.include] filters [_tier_preference:"data_hot OR data_warm OR data_cold"]”，表示集群的某个索引设置的参数与节点不匹配。

图 2-14 索引参数不匹配



b. 执行命令 **GET index_name/_settings**查看索引配置，返回结果中是否存在不符合自身版本的索引特性。

图 2-15 索引设置

以index.routing.allocation.include._tier_preference特性为例，当前集群是7.9.3版本，这个索引特性是在7.10版本之后才支持的，低版本集群使用该特性将无法分配索引的分片，导致集群不可用。

c. 确定集群是否必须使用该不匹配的特性。

- 是，创建与所需索引特性相匹配的版本集群，然后将老集群的数据通过备份恢复至新集群。
- 否，执行下一步。

d. 执行命令去除索引中不符合集群版本的特性。

```
PUT /index_name/_settings
{
 "index.routing.allocation.include._tier_preference": null
}
```

e. 执行命令**POST /_cluster/reroute?retry_failed=true**手动对未分配分片进行分配，等待索引分片分配完成，集群状态变为可用。

- **主分片数据损坏**

a. 返回信息中的索引名称"index"、分片标号"shard"、分配解释"allocate_explanation" 和"store_exception":"type":"corrupt index exception"，表示集群的某个索引的某个分片数据被损坏。

图 2-16 索引数据损坏

b. 当索引数据被损坏或者某个分片的主副本都丢失时，为了能使集群恢复green状态，解决方法是划分一个空shard，执行以下命令划分空分片并指定分配的节点。

```
POST /_cluster/reroute
{
   "commands" : [
         {
         "allocate_empty_primary" : {
            "index" : "index_name",
            "shard" : 2,
            "node" : "node_name",
            "accept_data_loss":true
         }
      }
   ]
}
```

须知

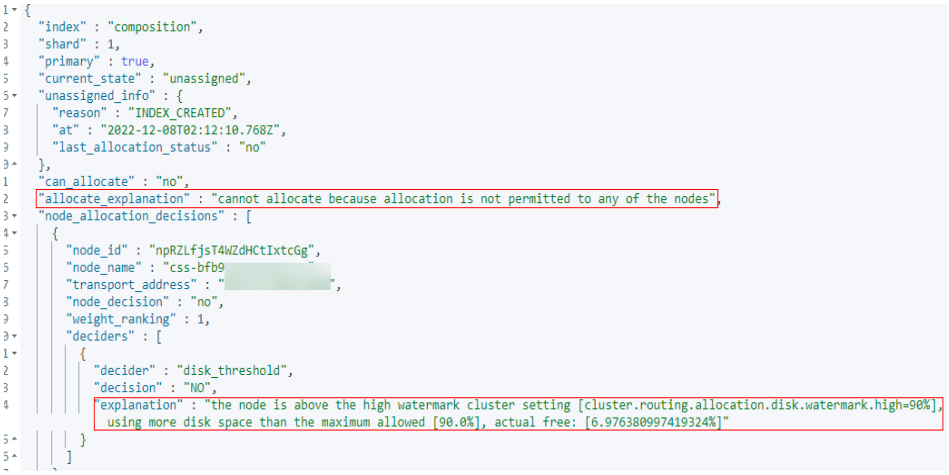
一定要谨慎该操作，会导致对应分片的数据完全清空。

c. 索引分片重新分配后，集群状态恢复可用。

- 磁盘使用率过高

a. 返回结果如下，其中“allocate_explanation”表示该索引的分片无法分配给任何数据节点，“explanation”表示节点磁盘使用率已超过磁盘空间允许的最大值。

图 2-17 explain 查询结果



说明

- 磁盘使用率超过85%：会导致新的分片无法分配。
- 磁盘使用率超过90%：集群会尝试将对应节点中的分片迁移到其他磁盘使用率比较低的数据节点中。无法迁移时系统会对集群每个索引强制设置“read_only_allow_delete”属性，此时索引将无法写入数据，只能读取和删除对应索引。
- 磁盘使用率过高时可能会发生节点脱离，后续节点自动恢复后也可能会因为集群压力过大，监控调ES接口查询集群状态时无响应，无法及时更新集群状态导致集群状态为不可用。

b. 增加集群可用磁盘容量。

- 在Kibana的“Dev Tools”页面，执行命令**DELETE index_name**清理集群的无效数据释放磁盘空间。
- 临时降低索引副本数，待扩容磁盘容量或扩容节点完成后改回索引副本数。

1) 在Kibana的“Dev Tools”页面，执行命令临时降低索引副本数。

```
PUT index_name/_settings
{
  "number_of_replicas": 1
}
```

如果返回结果如下：

图 2-18 索引 read-only-allow-delete 状态

```
{
  "type" : "cluster_block_exception",
  "reason" : "index [log07] blocked by: [TOO_MANY_REQUESTS/12/disk usage
    exceeded flood-stage watermark, index has read-only-allow-delete block];"
}
```

则是因为磁盘使用率已超过磁盘空间允许的最大值，集群所有索引被强制设置“read_only_allow_delete”属性，先执行命令将该属性值置为“null”，再执行**b.1)**的命令降低索引副本数。

```
PUT /_settings
{
  "index.blocks.read_only_allow_delete": null
}
```

2) 参考**扩容**对集群进行节点数量或节点存储容量进行扩容。

3) 待扩容完成后再执行**b.1)**改回索引副本数，待索引分片完全分配后，集群状态变为可用。

2.7 数据类型不兼容导致集群不可用

问题现象

集群进行备份恢复或集群迁移操作后，“集群状态”变为“不可用”。

原因分析

集群出现此场景的原因可能是目标集群不支持被恢复的数据中某些数据类型，比如旧集群有安装一些插件或者定义settings，新集群没有，导致的索引分片无法分配。

处理步骤

1. 在Kibana的“Dev Tools”页面，执行命令**GET _cluster/allocation/explain?pretty**查看索引分片未分配的原因。
2. 返回结果中显示索引名称“index” 和未分配解释“explanation” : “primary shard for this replica is not yet active”，表示分片副本未激活。

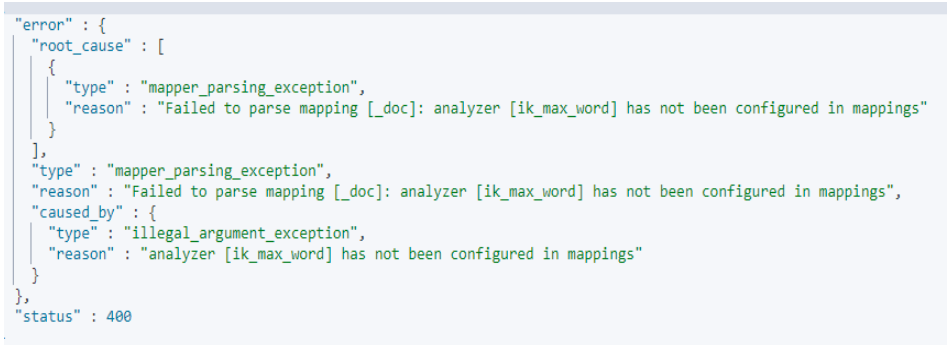
图 2-19 索引分片未分配

3. 尝试修改该索引的配置，执行命令将其副本数置为0。

```
PUT /index_name/_settings
{
 "number_of_replicas": 0
}
```

返回信息“reason”中表示在恢复的数据中存在CSS集群不支持的数据类型。

图 2-20 数据不兼容



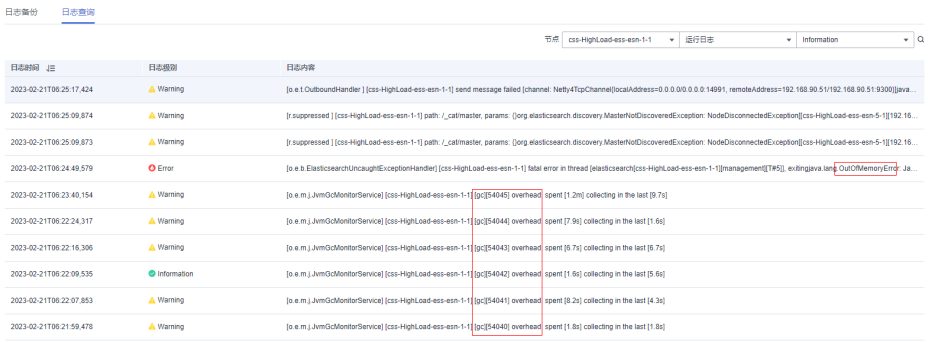
4. 根据问题根因，将数据中CSS集群不支持的数据类型删除或选择支持该数据类型的CSS集群版本，再进行备份恢复或数据迁移。

2.8 集群负载过高导致集群不可用

问题现象

“集群状态”为“不可用”，单击集群名称进入集群基本信息页面，选择“日志管理”，单击“日志查询”页签，可见日志内容存在报错“OutOfMemoryError”和警告“[gc][xxxxx] overhead spent [x.xs] collecting in the last [x.xs]”。

图 2-21 频繁 GC 导致 OOM



原因分析

集群负载过高，可能是有大量查询或写入任务堆积。当堆内存不足时，任务无法分配，将频繁触发GC，导致Elasticsearch进程异常退出。

处理步骤

说明

如果集群长期处于高负载状态，则集群会存在写入、查询缓慢等情形，建议根据业务需要升级节点规格或者对集群节点的数量和存储容量进行扩容，使集群更好的满足业务需求。升级节点规格、扩容节点数量和节点存储容量的指导请参见**扩容Elasticsearch集群**。

1. 查询集群是否存在任务堆积。
   - 方式一：在Kibana的“Dev Tools”页面，分别执行以下命令查询是否存在任务堆积。

     ```
     GET /_cat/thread_pool/write?v
     GET /_cat/thread_pool/search?v
     ```

     如下所示“queue”的值为非0，表示存在任务堆积。

     ```
     node_name              name   active queue rejected
     css-0323-ess-esn-2-1     write     2   200     7662
     css-0323-ess-esn-1-1     write     2   188     7660
     css-0323-ess-esn-5-1     write     2   200     7350
     css-0323-ess-esn-3-1     write     2   196     8000
     css-0323-ess-esn-4-1     write     2   189     7753
     ```

   - 方式二：在集群管理列表，单击集群操作列的“监控信息”查看监控指标，在集群监控信息页面查看集群的“Search队列中总排队任务数”和“Write队列中总排队任务数”，如果排队任务数值非0表示存在任务堆积。

图 2-22 Write 队列中总排队任务数示例

- 如果集群存在大量的任务堆积，则参考如下步骤优化集群。
  - 在集群的“日志管理”页面查看节点日志，查看节点在OOM前是否存在大量慢查询日志记录，分析查询是否会对节点造成压力导致节点内存不足，如果存在则根据业务实际情况优化查询语句。
  - 在集群的“日志管理”页面查看节点日志，查看节点日志是否有“Inflight circuit break”或“segment can't keep up”的报错信息，如果存在则可能是写入压力过大，对集群造成较大的压力导致熔断。需要查看监控信息，排查近期数据写入量（写入速率）是否存在激增，如果存在则根据业务实际情况合理安排写入高峰时间窗。
- 如果集群不存在任务堆积或者集群优化完依旧不可用，则执行下一步，查看集群是否压力过大。

2. 查看集群是否压力过大。

   在集群管理列表，单击集群操作列的“监控信息”查看监控指标，在监控信息页面查看CPU和堆内存相关指标，如“平均CPU使用率”和“平均JVM堆使用率”。如“平均CPU使用率”超过80%或“平均JVM堆使用率”高于70%，则说明集群当前压力较大。

图 2-23 “平均 CPU 使用率”示例

- 如果集群压力过大，请降低客户端的请求发送速率或扩容集群。
- 如果集群压力正常或降低发送请求速率后集群依旧不可用，则执行下一步，查看集群是否存在大量缓存。

3. 在Kibana的“Dev Tools”页面，执行以下命令查询集群是否存在大量缓存。
   ```
   GET /_cat/nodes?v&h=name,queryCacheMemory,fielddataMemory,requestCacheMemory
   ```
   - 如果返回结果中queryCacheMemory、fielddataMemory或requestCacheMemory的数值超过堆内存的20%，则表示缓存过大，可执行命令**POST _cache/clear**清除缓存。这些缓存数据是在数据查询时生成的，目的是为了加快查询速度，当缓存清除则可能使查询时延增加。
     ```
     name                 queryCacheMemory fielddataMemory requestCacheMemory
     css-0323-ess-esn-1-1              200mb          1.6gb              200mb
     ```

     说明

     每个节点的最大堆内存可以执行如下命令查询：

     ```
     GET _cat/nodes?v&h=name,ip,heapMax
     ```

     其中，*name*为节点名称，*ip*为节点的IP地址。
   - 如果排查优化后，集群依旧负载过高，则联系技术支持。

3 数据导入导出类

3.1 Elasticsearch 显示 CPU 使用率高，导致日志无法写入

问题现象

Elasticsearch在某一时间段CPU比较高，logstash在该时刻报错Elasticsearch Unreachable，导致日志无法写入到Elasticsearch里。

原因分析

客户index是单shard，压力承载于单个节点，负载过高，造成队列满后，作业被拒绝。

处理步骤

1. 登录云搜索服务控制台。
2. 选择“集群管理”进入集群管理列表。
3. 选择对应集群操作列“更多”>“Cerebro”。
   如果是安全模式集群，需要输入登录账号（admin）和密码。
4. 在Cerebro中查看集群的分片数、各节点的cpu、load、head、dis等数据指标。
5. 根据指标分析可能出现的原因，针对性优化。
   a. 增加队列数，减少拒绝作业，修改参数write.queue_size取值。
      i. 单击需要修改参数的集群名称，进入集群基本信息页面。
      ii. 选择“参数配置”，查找write.queue_size并修改取值。
      如果没有此参数，可以在自定义参数列进行添加。详细请参考**参数配置**章节。
   b. 重建索引，使分片数大于集群节点数。
6. 如果分片数和队列大小都满足条件，但是cpu和负载依然比较高，建议扩容节点。

3.2 ECS 服务器部署 Logstash 推送数据到 CSS 服务报错

问题现象

ECS服务器部署logstash，然后推送数据到云搜索服务CSS，出现错误信息如下：

LogStash::Outputs::ElasticSearch::HttpClient::Pool::BadResponseCodeError: Got response code '500' contacting Elasticsearch at URL 'https://192.168.xx.xx:9200/_xpack'。

原因分析

目前云搜索服务没有集成x-pack插件，自行搭建logstash连接css服务的时候，会检查es是否启用了x-pack。

处理步骤

1. 删除logstash中的x-pack目录。
2. 修改logstash配置文件output标签下elasticsearch内添加配置项ilm_enabled => false。
3. 重新尝试推送数据到es。

3.3 ES-Hadoop 导数据时报"Could not write all entries"异常

问题分析

Elasticsearch后台的bulk的线程池最大只支持接受200请求数队列，超过的请求会被rejected。

解决方案

1. 建议根据实际情况调整客户端的并发写入请求数（调整到一个合适的阈值），另外被rejected的http请求ES-Hadoop是有重试机制的，可修改以下参数:
   - “es.batch.write.retry.count”：默认重试3次。
   - “es.batch.write.retry.wait”：每次重试等待时间10s。
2. 如果对查询的实时性级别要求不高的话，可以调整下分片刷新的时间（默认是每秒刷新一次），提高写入速度。

```
PUT /my_logs
{
"settings": {
"refresh_interval": "30s"
}
}
```


4 功能使用类

4.1 无法备份索引

索引的备份是通过创建集群快照实现的。遇到无法备份索引问题，请按照如下操作步骤排查解决。

排查是否有权限

1. 登录统一身份认证服务管理控制台。
2. 查看当前登录所用的账号或IAM用户所属的用户组。

   具体操作请参见《统一身份认证服务用户指南》中的**查看或修改用户信息**章节。
3. 查看用户组的权限中是否包含：“全局服务”中“对象存储服务”项目的“OBS Administrator”权限、当前所属区域的“Elasticsearch Administrator”权限。

   具体操作请参见《统一身份认证服务用户指南》中的**查看或修改用户组**章节。
   - 如果用户组的权限中不包含以上两个权限，请执行步骤**4**。
   - 如果用户组的权限中包含以上两个权限，请联系华为云技术支持协助解决。
4. 为用户组添加：“全局服务”中“对象存储服务”项目的“OBS Administrator”权限、当前所属区域的“Elasticsearch Administrator”权限。

   具体操作请参见《统一身份认证服务用户指南》中的**查看或修改用户组**章节。

4.2 无法使用自定义词库功能

遇到该问题，请按照如下操作步骤排查解决。

排查集群的创建时间

步骤1 登录云搜索服务管理控制台。

步骤2 在左侧导航栏，选择“集群管理”。

步骤3 在集群管理列表页，查看待配置自定义词库的集群的“创建时间”。

- 如果创建时间早于2018年3月10日，则创建该集群时自定义词库功能尚未上线，当前无法为该集群配置自定义词库。

- 如果创建时间晚于2018年3月10日，则需要排查当前登录所用的账号或IAM用户是否具有使用自定义词库功能的权限，具体操作请参见**排查是否有权限**。

----结束

排查是否有权限

1. 登录统一身份认证服务管理控制台。
2. 查看当前登录所用的账号或IAM用户所属的用户组。

   具体操作请参见《统一身份认证服务用户指南》中的**查看或修改IAM用户信息**章节。
3. 查看用户组的权限中是否包含：“全局服务”中“对象存储服务”项目的“Tenant Administrator”权限、当前所属区域的“Elasticsearch Administrator”权限。

   具体操作请参见《统一身份认证服务用户指南》中的**查看或修改用户组**章节。
   - 如果用户组的权限中不包含以上两个权限，请执行**4**。
   - 如果用户组的权限中包含以上两个权限，请联系人工客服协助解决。
4. 为用户组添加：“全局服务”中“对象存储服务”项目的“Tenant Administrator”权限、当前所属区域的“Elasticsearch Administrator”权限。

   具体操作请参见《统一身份认证服务用户指南》中的**查看或修改用户组**章节。

4.3 快照仓库找不到

1. 在云搜索服务的“集群管理”页面上，单击集群“操作”列的“Kibana”访问集群。
2. 在Kibana的左侧导航中选择“Dev Tools”，单击“Get to work”，进入Console界面。

   Console左侧区域为输入框，右侧为结果输出区域， 为执行命令按钮。
3. 执行命令**GET _snapshot/_all**返回为空，或者执行命令**GET _snapshot/repo_auto/_all**返回如**图4-1**提示错误，这个表示没有设置快照。需要重新修改快照配置信息。

   图 4-1 返回信息

```
{
  "error": {
    "root_cause": [
      {
        "type": "repository_missing_exception",
        "reason": "[repo_auto] missing"
      }
    ],
    "type": "repository_missing_exception",
    "reason": "[repo_auto] missing"
  },
  "status": 404
}
```

4. 单击集群名称，进入集群基本信息页面，选择“集群快照”，进入集群快照页面。

5. 单击“基础配置”后面的 ，修改基础配置。
6. 修改完成后，单击“确定”。

   如果保存后还不能生效，可以尝试修改下备份路径，换一个不一样的值保存然后再修改回来。

4.4 集群一直处于快照中

集群一直处于快照中，有三个比较常见的原因：

- 集群数据量大或者集群压力大，备份快照耗时长。

  单个节点的快照速度默认是40MB/s，同时，快照的性能还受集群情况影响，如果此时集群负载较高，耗时将会更久。可以通过上述章节的查询单个快照信息查询正在执行的快照情况。

  执行**GET _snapshot/repo_auto/snapshot-name**，可以看到剩余还需要完成的shard个数，也可以通过删除快照接口提前终止。

  解决方法：等待或者提前终止。

- 快照信息更新失败。

  Elasticsearch将进行中的快照信息保存在cluster state中，快照完成后需要更新快照状态，由于Elasticsearch更新快照状态的接口没有加入重试或者容错机制，比如由于当时集群内存压力大，更新快照动作被熔断，那么这个快照将会一直处于快照中。

  解决方法：调用快照删除接口。

- 临时AK、SK过期。

  CSS通过委托将Elasticsearch中的数据写入到用户的OBS中，快照仓库创建的时候，需要去使用委托获取临时的AK 、SK设置到仓库中。由于临时的AK、SK是有时效性的（24小时过期），如果一个快照超过24小时还未完成，那么这个快照将会失败。这种情况会有一个比较大的风险，因为此时仓库的AK、SK过期，无法对这个仓库进行更新、查询、删除操作，这种情况下cluster state信息将会无法清除，只能通过普通重启（滚动重启无法生效）集群来清除cluster state里面残留的快照信息。

  解决方法：暂时只能通过普通重启集群来消除，后期CSS会提供终止接口，可以来解决这种无法消除状态的现象。

4.5 数据量很大，如何进行快照备份?

如果快照数据量极大，快照备份要超过一天时，可参考如下方法进行优化。

1. 快照备份的时候指定索引，比如先分批，默认是*，将会备份所有的索引。
2. 使用自定义快照仓库。
   a. 创建自定义仓库。

      除了使用云搜索服务提供的repo_auto之外，客户也可以自己创建一个仓库，接口见如下：

```
PUT _snapshot/my_backup
{
  "type" : "obs",
  "settings" : {
    "bucket" : "css-backup-name",    //桶名
    "base_path" : "css_backup/711/",  //备份路径
```



```
    "chunk_size" : "2g",
    "endpoint" : "obs.xxx.com:443", //OBS域名地址
    "region" : "xxx",        //Region名称
    "compress" : "true",
    "access_key": "xxxxx",        //AK
    "secret_key": "xxxxxxxxxxxxxxxxx"     //SK
   "max_restore_bytes_per_sec": "100mb",           //OBS速度，默认是40MB,可以根据实际性能调大
    "max_snapshot_bytes_per_sec": "100mb"
  }
}
```

b. 使用自定义仓库创建快照。

```
PUT _snapshot/my_backup/snapshot_name（快照名称）
{
  "indices": "*", //备份的索引，*表示索引，逗号分隔
  "ignore_unavailable": true,  //是否忽略单个index是否可用,true表示忽略
  "include_global_state": false //默认false表示cluster state和其他的一些state不会保存下来
}
```

c. 查询快照状态。

```
GET _snapshot/my_backup/snapshot_name/_status
```

d. 恢复自定义仓库中的索引。

```
POST /_snapshot/my_backup/snapshot_name/_restore
{
  "indices": "test-00000000000",
  "ignore_unavailable": true,
  "include_global_state": false,
  "rename_pattern": "(.+)",
  "rename_replacement": "$1"
}
```

4.6 集群突现 load 高的故障排查

问题现象

集群任务被长时间拒绝，且大量任务出现卡死的情况，在Cerebro界面可以看到集群的load数值突然飙升。

原因分析

集群出现load升高的可能原因如下：

- 查询请求命中的数据较多导致查询线程执行缓慢。
- 写入压力过大导致很多线程出现卡死现象。

排查步骤

方法1：Cerebro工具

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 > Elasticsearch”，进入集群列表页面。
3. 找到load飙升的集群，单击集群操作列的“Cerebro”进入可视化页面。
4. 查看cpu和heap指标，如果这两个指标过高则说明集群当前压力较大，客户端可以适当减少大请求发送，等待集群压力下降。
5. 查看shards是否合理，官方建议单个shard大小为20-40GB，建议不要超过50GB；单个节点上的同一索引shard数不要超过5个。

方法2：Kibana工具

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 > Elasticsearch”，进入集群列表页面。
3. 找到load飙升的集群，单击集群操作列的“Kibana”进入页面，单击可以执行命令的Dev Tools工具。
4. 执行**GET _cat/thread_pool?v**查看哪些线程任务堆积，定位是什么原因导致的集群压力倍增。
5. 执行**GET /_nodes/hot_threads**可以查看当前占用大量CPU且执行时间很长的线程，定位何处导致任务积压。

4.7 使用 ElasticSearch 的 HLRC（High Level Rest Client）时，报出 I/O Reactor STOPPED

问题现象

使用ElasticSearch的HLRC（High Level Rest Client）时，偶现报出I/O Reactor STOPPED。排查ElasticSearch日志，未有报错。

```
java.lang.RuntimeException: Request cannot be executed; I/O reactor status: STOPPED
        at org.elasticsearch.client.RestClient.extractAndWrapCause(RestClient.java:796)
        at org.elasticsearch.client.RestClient.performRequest(RestClient.java:218)
        at org.elasticsearch.client.RestClient.performRequest(RestClient.java:205)
        at org.elasticsearch.client.RestHighLevelClient.internalPerformRequest(RestHighLevelClient.java:1454)
        at org.elasticsearch.client.RestHighLevelClient.performRequest(RestHighLevelClient.java:1424)
        at org.elasticsearch.client.RestHighLevelClient.performRequestAndParseEntity(RestHighLevelClient.java:1394)
        at org.elasticsearch.client.RestHighLevelClient.search(RestHighLevelClient.java:930)
        at
)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.IllegalStateException: Request cannot be executed; I/O reactor status: STOPPED
        at org.apache.http.util.Asserts.check(Asserts.java:46)
        at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase.ensureRunning(CloseableHttpAsyncClientBase.java:90)
        at org.apache.http.impl.nio.client.InternalHttpAsyncClient.execute(InternalHttpAsyncClient.java:123)
        at org.elasticsearch.client.RestClient.performRequest(RestClient.java:214)
        ... 9 common frames omitted
```

I/O Reactor STOPPED 是什么问题?

首先根据调用栈可以定位到报错来自CloseableHttpAsyncClientBase中的90行，如下图所示：

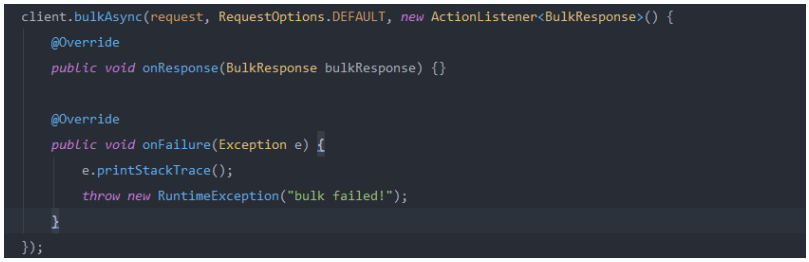
```
88      protected void ensureRunning() {
89          final Status currentStatus = this.status.get();
90          Asserts.check( expression: currentStatus == Status.ACTIVE,  message: "Request cannot be executed; " +
91                  "I/O reactor status: %s", currentStatus);
92      }
```

ensureRunning()方法是在每次请求执行开始的时候调用，用来确认client状态是否为ACTIVE，如果不是ACTIVE，则会报错。然后观察CloseableHttpAsyncClientBase中的status何时会变成STOPPED，发现有且仅有两处会set为STOPPED，如下图所示：

图 4-2 第一处 STOPPED

图 4-3 第二处 STOPPED

- 由于客户不会手动关闭客户端之后再调用接口，所以有且仅有第一处会导致status切换为STOPPED状态。
- 对于第一处STOPPED，reactorThread线程是用来在io events发生时调度io events，当内部抛出异常时，最终会将status改为STOPPED状态。然后在bulkAsync请求的回调中抛出异常验证status的状态切换，如下图所示：



- 当请求失败，status会切换为STOPPED且I/O Reactor将关闭，并使HLRC实例卡住。后续使用该HLRC实例调用任何请求都会失败。此处手动抛出异常是为了复现问题，生产环境中很难分辨是什么原因导致I/O Reactor关闭。

原因分析

导致出现I/O Reactor STOPPED的原因，大致可以分为以下3类：

1. 回调中抛出异常导致。
2. 客户端并发太高导致。

   在日志中发现异常后，查看ElasticSearch集群监控指标，如CPU使用率、网络连接数等。

   当用户集群配置为5台16U128G的i3.4xlarge.8节点时，且每天上午5点左右会做大量bulk操作，写入大概100G-200G的数据，根据集群监控指标的CPU使用率、网络流入流出速率来看对ElasticSearch节点造成不了压力，网络连接数较高，其它节点情况也相同。但是，有的节点网络连接数高达近9000，5个节点瞬间有将近5万连接数，用户的代码大致是用同一个Rest Client多个线程并发且调用HLRC的bulkAsync接口。客户端一个节点，单是ES的连接就消耗了4-5万个连接数，这种情况很容易造成客户端节点句柄数耗尽，或者连接数耗尽。
3. ElasticSearch的Rest client导致。建议ElasticSearch完善Rest client，添加exception handler。

   Apache（HLRC和LLRC都使用了Apache HTTPComponents Async Client）手册中提到，在与会话通道交互过程中有些I/O异常是可以预料的，这些异常可能会导致单个session终止，但不会影响I/O Reactor和其他session。但某些情况下，当I/O Reactor本身遇到内部问题时，例如底层NIO的一些类中的I/O异常或者没被handle的一些Runtime Exception。这些异常是致命的，会使I/O Reactor关闭。Apache官方建议重写IOReactorExceptionHandler接口。

   图 4-4 重写 IOReactorExceptionHandler 接口

```
DefaultConnectingIOReactor ioreactor = <...>

ioreactor.setExceptionHandler(new IOReactorExceptionHandler() {

    public boolean handle(IOException ex) {
        if (ex instanceof BindException) {
            // bind failures considered OK to ignore
            return true;
        }
        return false;
    }

    public boolean handle(RuntimeException ex) {
        if (ex instanceof UnsupportedOperationException) {
            // Unsupported operations considered OK to ignore
            return true;
        }
        return false;
    }

});
```

   但是尝试了重写ExceptionHandler并放到HLRC的配置中，并通过在回调中抛出异常来模拟异常场景，最后发现这种回调的异常并不会被IOReactorExceptionHandler捕获，运行脚本后也没有异常抛出，所以此处IOReactorExceptionHandler的实现并不好验证。通过ElasticSearch社区issue中其他开发者的验证，添加了IOReactorExceptionHandler后跑很久也不会有问题。所以建议添加IOReactorExceptionHandler，但是注意不要忽略所有异常。

   Elasticsearch Rest Client需要有一个exception handler，而不是让用户通过设置IOReactorExceptionHandler来处理异常，且这种方式也不会解决所有的异常。

处理建议

1. 客户端连接池大小建议根据业务实际情况调整。
2. 客户端并发数建议根据业务实际情况调整，或者配置多个节点、分摊压力。
3. 建议配置IOReactorExceptionHandler，可以用来处理一些异常。
4. 调用HLRC时catch exception，并判断cause是否为I/O Reactor STOPPED，然后通过重启客户端恢复连接。

4.8 Elasticsearch 集群最大堆内存持续过高（超过 90%）

问题描述

关于Elasticsearch集群的最大堆内存持续超过90%的问题。其中如果节点在90%堆内存上下波动，有增有减，则无异常；持续高内存时，集群存在一定的风险。

原因分析

- 排查集群的写入和查询队列，查看是否有大量任务堆积。
  ```
  GET /_cat/thread_pool/write?v
  GET /_cat/thread_pool/search?v
  ```
- 查看集群监控，排查集群的写入和查询任务相关指标。
- 如果集群长期处于高堆内存占用状态，查看集群节点个数、节点规模，确认是否需要扩容。

解决方案

- 根据任务堆积现象优化客户端写入或查询程序。
- 根据业务情况，如果集群长期处于高负载状态，则集群会存在写入、查询缓慢，节点频繁掉线等情况。需要根据需要扩容节点规模，或分业务重新规划集群规模。
- 如果日常使用有95%堆内存波动和节点掉线情况，可根据需要使用流量控制功能。更多信息，请参见**配置ELasticsearch集群读写流量控制策略**。

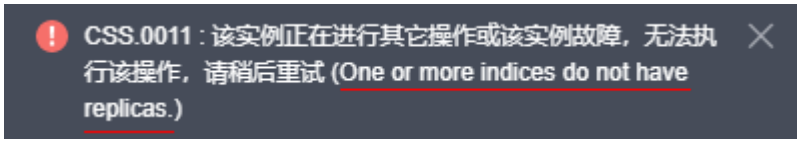
4.9 Elasticsearch 集群更改规格失败

问题描述

执行集群更改规格操作失败，console界面报错详情如下图所示。

图 4-5 CSS.0073 错误

图 4-6 CSS.0011 错误



原因分析

当前集群未设置副本数，后台拦截了更改规格的请求。需要设置好副本数，再进行更改规格操作，否则会有分片丢失的风险。

解决方案

设置副本参数：

```
PUT /index/_settings
{
    "number_of_replicas" : 1 //表示需要设置的副本数
}
```

4.10 安全集群索引只读状态修改报错

问题描述

安全集群空间存满之后，索引会全部变为只读模式**"read_only_allow_delete": "true"**，导致无法再写入，需手动修改只读模式为“false”，执行如下命令：

```
PUT _settings
{
"index": {
"blocks": {
"read_only_allow_delete": "false"
}
}
}
```

报错为：

```
{
"error": {
"root_cause": [
{
"type": "security_exception",
"reason": "no permissions for [] and User [name=admin, roles=[admin], requestedTenant=null]"
}
],
"type": "security_exception",
"reason": "no permissions for [] and User [name=admin, roles=[admin], requestedTenant=null]"
},
"status": 403
}
```

原因分析

安全集群，默认有一个“. opendistro_security”索引，不可执行写操作，修改索引读写模式时要忽略掉这个索引。

解决方案

使用通配符进行匹配，将indexname用通配符代替。

```
PUT indexname/_settings
{
"index": {
"blocks": {
"read_only_allow_delete": "false"
}
```

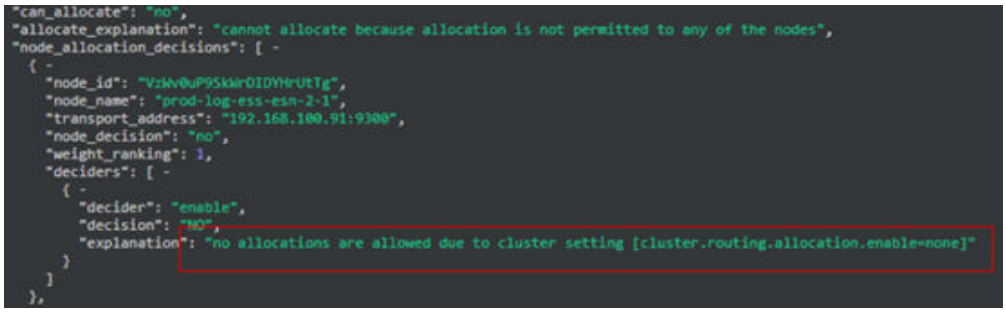


```
}
}
```

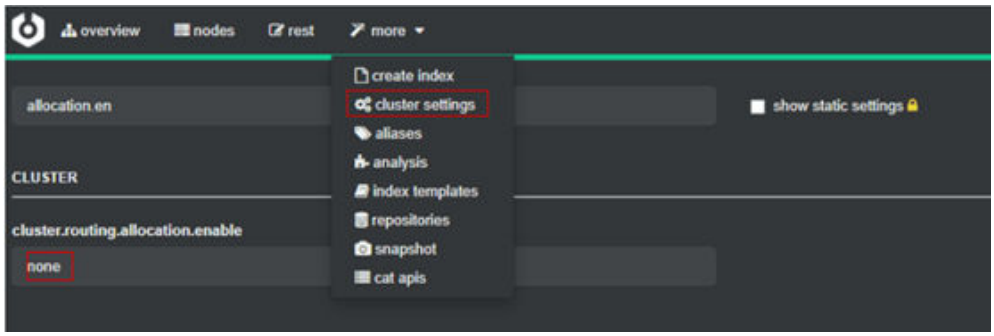
4.11 Elasticsearch 集群某一节点分配不到 shard

解决方案

1. 使用**GET _cluster/allocation/explain?pretty** 查看未分配shard。



2. 在console上，选择“cerebro > more > cluster settings ”-，在左上角输入“allocation.enable”，将“none”改为“all”。



3. 如果没有cerebro，请执行如下命令：

```
curl -X PUT "http://内网ip:9200/_cluster/settings" -H 'Content-Type: application/json' -d'{"persistent": {"cluster.routing.allocation.enable": "all"}}'
```

4.12 集群索引插入数据失败

问题现象

向CSS集群索引中插入数据失败，报错如下：

```
"reason":"blocked by: [FORBIDDEN/12/index read-only / allow delete (api)];")),
```

问题分析

当磁盘的使用率超过95%时，Elasticsearch为了防止节点耗尽磁盘空间，自动将索引设置为只读模式。

解决方案

- 新版本（7.10.2之后）集群磁盘使用率下降后会自动关闭只读模式，只需清理或扩容磁盘。

- 旧版本需要手动操作，kibana执行如下命令：
  ```
  PUT /_all/_settings { "index.blocks.read_only_allow_delete": null }
  ```

4.13 CSS 创建索引报错“maximum shards open”

问题描述

创建索引时，报错显示“this action would add [2] total shards, but this cluster currently has [1000]/[1000] maximum shards open”。

问题原因

由于节点shard数量超过最大值限制，Elasticsearch默认每个节点的shard数量是1000，如果shard数量超过这个值创建索引会报错。

解决方案

- 方案一：关闭或者删除不用的索引，减少shard数量。
- 方案二：修改节点的shard数量的限制，参数配置请参考**max_shards_per_node**。

```
PUT _cluster/settings
{
  "persistent": {
    "cluster": {
      "max_shards_per_node": 2000
    }
  }
}
```

说明

修改节点的shard数量的限制属于临时规避方案，如果要长期解决，建议每GB的JVM堆内存小于等于20个shards（节点堆内存大小为节点内存规格的1/2，最大值为31GB），shard的建议数量详细请参见**shard count recommendation**。

4.14 删除索引报错“403 Forbidden”是什么原因?

问题描述

执行命令**curl –i –u admin:password –XDELETE https://ip:9200/_all**（“password”为admin账号的密码，“ip”为集群的内网访问地址）删除所有索引时，报错“403 Forbidden”。

解决方案

安全模式的集群中索引“.opendistro_security”无法被删除，所以删除所有索引的命令对安全集群无效。建议使用索引名称或者通配符进行删除，不要使用全删除命令。

4.15 Kibana 中删除 index pattern 报错 Forbidden

问题描述

在Kibana界面删除索引模式，单击删除按钮报错Forbidden。

原因分析

之前创建的索引模式无法删除索引模式是因为kibana索引只读导致的，磁盘使用率超过一定阈值会自动转为只读，所以报错没有权限。

解决方案

在Kibana的“Dev Tools”页面，执行以下命令解除Kibana索引只读状态。

```
PUT .kibana*/_settings
{
"index.blocks.read_only_allow_delete":null
}
```

4.16 执行命令 update-by-query 报错“Trying to create too many scroll contexts”

问题现象

云搜索服务的Elasticsearch集群执行命令**update-by-query**，出现报错“Trying to create too many scroll contexts.”，具体报错信息如下：

```
{"error":{"root_cause":[{"type":"exception","reason":"Trying to create too many scroll contexts. Must be less than or equal to: [100000]. This limit can be set by changing the [search.max_open_scroll_context] setting."},{"type":"exception","reason":"Trying to create too many scroll contexts. Must be less than or equal to: [100000]. .....
```

原因分析

当集群每调用一次scroll的创建接口，都会新建一个scroll使用的context，当scroll contexts的数目达到预定值时，将无法继续创建scroll。

处理步骤

执行如下命令，查看“open_contexts”的参数值即为当前scroll contexts的数目。

```
GET /_nodes/stats/indices/search
```

当数目达到最大值时，可以通过以下两种方式解决。

- 方式一：删除无用的scroll，释放scroll contexts的存储空间。

  ```
  DELETE /_search/scroll
  {  "scroll_id" :
  "DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAAD4WYm9laVYtZndUQlNsdDcwakFMNjU1QQ=="}
  ```

- 方式二：调整“search.max_open_scroll_context”的值，增加scroll contexts的存储空间。

  ```
  PUT /_cluster/settings
  {
     "persistent" : {
        "search.max_open_scroll_context": 2012345678
     },

     "transient": {
        "search.max_open_scroll_context": 2012345678
     }
  }
  ```

4.17 Elasticsearch 集群无法创建 pattern

问题描述

单击创建pattern无反应，无法创建pattern。

问题原因

1. 检查磁盘是否太满，导致Kibana索引为只读状态。
2. 检查是否有多个Kibana索引。

解决方案

执行以下命令删除不必要的索引数据，释放磁盘空间。

```
PUT .kibana/_settings
{
"index": {
"blocks": {
"read_only_allow_delete": "null"
}
}
}
```

5 端口访问类

5.1 9200 端口访问失败

问题现象

通过VPN专线或VPC的对等连接访问CSS集群的场景下，使用curl命令接入Elasticsearch集群时，无返回结果。

例如，执行如下命令接入集群，无返回结果。

```
curl -s 'http://<节点内网访问地址>:9200'
```

原因分析

在“使用VPN专线访问CSS集群”或“通过VPC的对等连接访问CSS集群”场景下，其所在的客户端与CSS不在同一VPC下。因此，要求CSS集群的子网与其VPC具有不同的网段。

例如，某一CSS集群，选用的VPC为vpc-8e28，其网络配置为192.168.0.0/16。选用了此VPC下的子网subnet-4a81，subnet-4a81子网的网段与vpc-8e28一致，均为192.168.0.0/16。此时，如果使用VPN专线访问CSS集群或通过VPC的对等连接访问CSS集群，会导致此子网创建的机器内没有该VPC对应的网关，从而影响CSS服务的默认路由的设置，最终导致9200端口访问失败。

处理步骤

当出现9200端口访问失败错误时，且CSS集群状态为可用状态。执行步骤如下所示：

1. 进入CSS服务管理控制台，在集群列表中，单击集群名称进入集群详情页面，查看此集群使用的VPC和子网。
2. 进入VPC服务管理控制台，在虚拟私有云列表中，单击CSS集群使用的VPC名称，进入VPC详情页面。查看VPC和子网的网段信息。

   如图5-1所示，VPC的网段信息，与子网的网段信息一致。在使用VPN专线访问或使用VPC对等连接访问时，会导致9200端口访问失败。

图 5-1 查看网段信息

3. 如果出现上述错误，请重新创建集群，并选择一个网段与VPC不同的子网，如不存在这样的子网，请在VPC管理控制台重新创建一个子网。

   创建新的CSS集群后，将旧集群的数据迁移至新集群中，然后再通过VPN专线访问或使用VPC对等连接访问使用。

   说明

   如果需要VPN专线访问或使用VPC对等连接访问CSS集群时，请务必保证，新创建的CSS集群，其VPC与子网，具备不同的网段信息。