Cloud Search Service

Troubleshooting

Issue	01
Date	2025-01-23

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Clusters

1.1 Failed to Open Kibana

Symptom

On the **Clusters** page of the CSS management console, after I click **Access Kibana** in the **Operation** column in the row where cluster **Es-event** resides, the Kibana page fails to be loaded.

Possible Causes

The browser cache is not cleared.

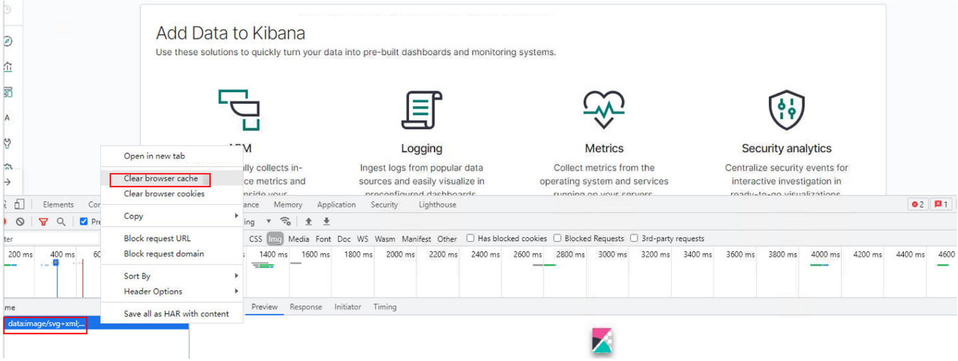
Procedure

1. Log in to the CSS management console.
2. In the navigation pane on the left, choose **Clusters**.
3. On the displayed **Clusters** page, locate the target cluster and click **Access Kibana** in the **Operation** column.

   NOTE

   - If the cluster has the security mode enabled, enter the username and password you for login. The default username is **admin** and the password is the one specified during cluster creation.
   - If you forget the password, set and confirm a new administrator password. In the **Configuration** area, click **Reset** next to **Reset Password**.
4. On the displayed **Kibana** page, press **F12**.
5. Click **Network**, right-click **data:image**, and choose **clear browser cache** from the shortcut menu. In the displayed dialog box, click **OK**. Close the Kibana window.

Figure 1-1 Closing the Kibana page



Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.

6. Switch to the **Clusters** page, locate target cluster and click **Access Kibana** in the **Operation** column.

1.2 How Can I Improve Filebeat Performance?

Symptom

Filebeat is a high-performance file collection tool. By default, one core is allocated to Filebeat, and it writes 1 MB data to Elasticsearch per second. However, in practice, when a large number of service logs are generated, Filebeat cannot promptly collect and write them to Elasticsearch. In this case, you can optimize parameter settings in the **filebeat.yml** file to improve the Filebeat performance.

Possible Causes

For Filebeat, the default configuration of the **filebeat.yml** file cannot deliver optimal performance when handling a large number of logs. In such scenarios, modify parameter settings in the **filebeat.yml** file to meet your demands.

Procedure

1. Optimize the parameters involved in **input** of the **filebeat.yml** configuration file.

   Increase the value of **harvester_buffer_size** based on actual requirements. This parameter defines the buffer size used by every **harvester**.
   harvester_buffer_size: **40,960,000**
   Increase the value of **filebeat.spool_size** based on actual requirements. This parameter defines the number of log records that can be uploaded by the **spooler** at a time.
   filebeat.spool_size: **250,000**
   Adjust the value of **filebeat.idle_timeout** based on actual requirements. This parameter defines how often the **spooler** is flushed. After the **idle_timeout** is reached, the **spooler** is flushed regardless of whether the **spool_size** has been reached.
   filebeat.idle_timeout: **1s**

2. Optimize the parameters involved in **output.elasticsearch** in the **filebeat.yml** configuration file.

   Set the value of **worker** to the number of Elasticsearch clusters based on actual requirements. This parameter indicates the number of Elasticsearch clusters. The default value is **1**.
   worker: **1**
   Increase the value of **bulk_max_size** based on actual requirements. This parameter defines the maximum number of events to bulk in a single Elasticsearch bulk API index request. The default is **50**.
   bulk_max_size: **15,000**
   Adjust the value of **flush_interval** based on the actual requirements. This parameter defines the number of seconds to wait for new events between two bulk API index requests. If **bulk_max_size** is

```
reached before this interval expires, additional bulk index requests are made.
flush_interval: 1s
```

1.3 How Do I Handle the Error "Connection reset by peer" That Occurs When Spring Boot Uses Elasticsearch?

Symptom

When Spring Boot uses Elasticsearch RestHighLevelClient to connect to Elasticsearch, the error "Connection reset by peer" is reported, the TCP connection is interrupted, and service data fails to be written.

Possible Causes

There are many possible causes. For example, the connection was disabled; the firewall, switch, or VPN was faulty; the keepalive settings were incorrect; the connected server node was changed; or the network was unstable.

Procedure

Choose one of the following methods to solve the issue:

- Method 1

  Modify the timeout interval of RestHighLevelClient connection requests. The default value is 1000 ms. You can increase the value to 10,000 ms.

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

  Settings for a single request: **request.timeout(TimeValue.timeValueSeconds(60))**.

- Method 2

  Set the RestHighLevelClient keepalive time to 15 minutes.

- Method 3

  Handle the exception captured in the code and retry the request.

Reference

- TCP connections

  TCP connections are classified into persistent connections and short connections. A short TCP connection is automatically disconnected after data packets are sent. A persistent TCP connection uses the keepalive timer function, and remains open for a certain period of time after data packets are sent.

- TCP keepalive mechanism

  The keepalive mechanism is implemented using a timer. If the timer is activated, the server will send a keepalive probe packet. An ACK message is expected as a response. If the client does not respond, the server will terminate the connection. If the client responds, the keepalive timer will be reset.

The keepalive duration on the server is set to **30m**. In Linux, three parameters can be used to control the keepalive duration: **tcp_keepalive_time** (idle duration for enabling the keepalive function), **tcp_keepalive_intvl** (interval for sending keepalive packets), and **tcp_keepalive_probes** (the number of times the keepalive packets are sent if no response is received).

- http-keepalive

  The http-keepalive mechanism enables a TCP connection to transmit as many packets as possible. The http-keepalive duration is updated each time a packet is transmitted. If the http-keepalive duration expires, it indicates that the client and server did not exchange packets during this period. In this case, the connection is automatically closed and released.

  The tcp-keepalive mechanism retains a TCP connection until the connection is deliberately closed.

1.4 Why Does Cluster Creation Fail?

The possible causes of a cluster creation failure are as follows:

- Insufficient resource quota. You are advised to increase the resource quotas.
- The value of **Port Range/ICMP Type** in **Security Group** does not include port **9200**. Modify the security group information or select another available security group.
- For cluster version 7.6.2 and later versions, the communication port 9300 is enabled on the subnet of user VPC by default. When you create a cluster, check whether the selected security group allows traffic from communication port 9300 in the subnet. If it does not allow traffic, modify the security group or select another security group.
- Insufficient permission. You are advised to obtain required permissions. For details, see **Permissions Management**.

1.5 What Do I Do If "Bulk Reject" Is Displayed in an Elasticsearch Cluster?

Symptom

Sometimes the cluster write rejection rate increases and the "Bulk Reject" message is displayed. When I perform bulk writing operations, an error message similar to the following is reported:

```
[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"
}
```

Issue Analysis

Bulk reject is usually caused by large or unevenly distributed shards. You can use the following methods to locate and analyze the fault:

1. Check whether the data volume in shards is too large.

   The recommended data size in a single shard is 20 GB to 50 GB. You can run the following command on the Kibana console to view the size of each shard of an index:

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

2. Check whether the shards in nodes are unevenly distributed.

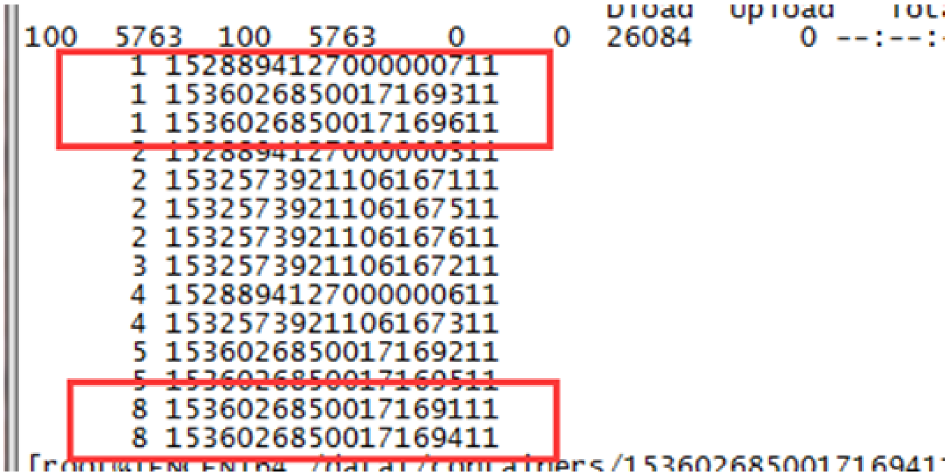
   You can check shard allocation in either of the following ways:

   a. Log in to the CSS management console and choose **Clusters**. Locate the target cluster and click **More** > **View Metric**. For details, see **Viewing Monitoring Metrics**.

   b. On the CURL client, check the number of shards on each node in the cluster.

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

   An example is as follows.



```
                                                  Dload  Upload   Total
100  5763  100  5763    0     0  26084      0 --:--:
      1 152889412700000711
      1 153602685001716931 1
      1 1536026850017169611
      2 1528894127000000511
      2 1532573921106167111
      2 1532573921106167511
      2 1532573921106167611
      3 1532573921106167211
      4 1528894127000000611
      4 1532573921106167311
      5 1536026850017169211
      5 1536026850017169511
      8 1536026850017169111
      8 1536026850017169411
[root@TENCENT64 /data/containers/15360268500171694
```

   NOTE

   The first column indicates the number of shards, and the second column indicates the node ID. As shown in the example, some nodes have only one shard while some have eight. The shard allocation is uneven.

Solution

- If the problem is caused by large shards:

  Set the value of the **number_of_shards** parameter in the **index** template to limit the shard size.

  NOTE

  A newly created template will take effect when a new index is created. Existing indexes cannot be changed.

- If the problem is caused by uneven shard distribution:

  Workarounds

a. You can run the following command to set the **routing.allocation.total_shards_per_node** parameter to dynamically adjust an index:

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

> **NOTE**
>
> When you configure the **total_shards_per_node** parameter, reserve some buffer space to avoid shard allocation failures caused by machine faults. For example, if there are 10 servers and the index has 20 shards, the value of **total_shards_per_node** must be greater than 2.

b. Set the number of shards before creating an index.

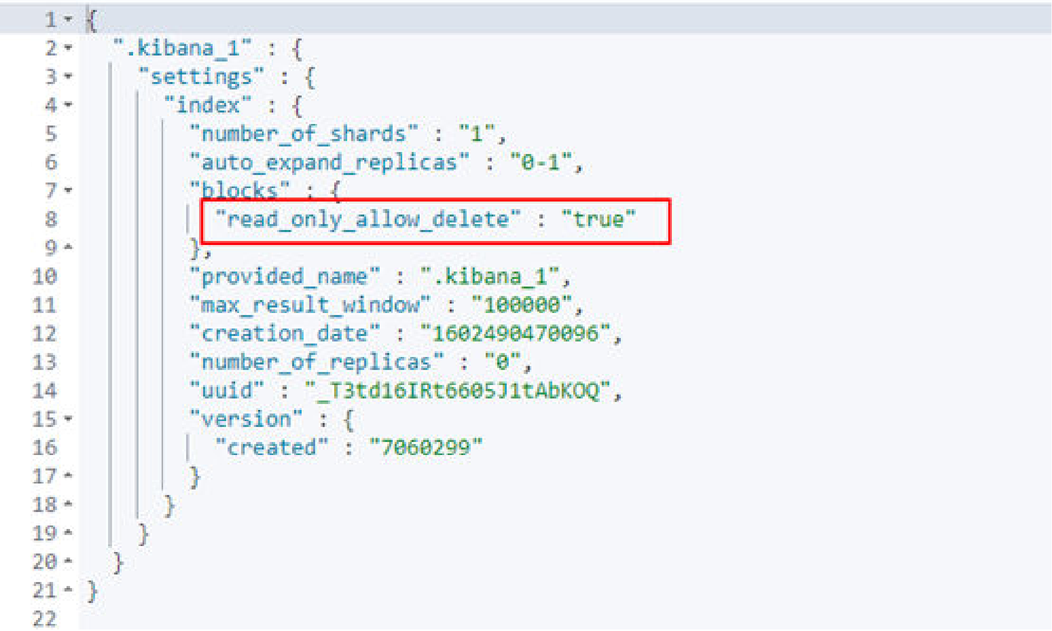
Use the index template to set the number of shards on each node.

```
PUT _template/<template_name>
{
    "order": 0,
    "template": "{index_prefix@}*",  //The index prefix you want to change
    "settings": {
        "index": {
            "number_of_shards": "30", //Total number of shards allocated to nodes. The capacity of
a shard can be assumed as 30 GB.
            "routing.allocation.total_shards_per_node": 3 //Maximum number of shards on a node
        }
    },
    "aliases": {}
}
```

1.6 What Do I Do If I Failed to Create an Index Pattern in an Elasticsearch Cluster?

Symptom

On the **Dev Tools** page of Kibana, run the **GET .kibana/_settings** command to query the Kibana index status. The value of the parameter **read_only_allow_delete** is **true**, which indicates that the cluster disk usage is too high.



```
{
  ".kibana_1" : {
    "settings" : {
      "index" : {
        "number_of_shards" : "1",
        "auto_expand_replicas" : "0-1",
        "blocks" : {
          "read_only_allow_delete" : "true"
        },
        "provided_name" : ".kibana_1",
        "max_result_window" : "100000",
        "creation_date" : "1602490470096",
        "number_of_replicas" : "0",
        "uuid" : "_T3td16IRt6605J1tAbKOQ",
        "version" : {
          "created" : "7060299"
        }
      }
    }
  }
}
```

Possible Cause

The index status is set to read-only.

Solution

1. Change the value of the parameter **read_only_allow_delete** to **false** and then create an index pattern.
2. Run the following command on the **Dev Tools** page of Kibana:

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

1.7 What Do I Do If a Message Indicating that the System Is Busy Is Displayed on the CSS Console?

Symptom

In the navigation pane of the CSS console, choose **Clusters** and the messages "The system is busy. Try again later or call the customer service hotline at 4000-955-988." or "Policy doesn't allow css: cluster:list to be performed." are displayed.

Possible Cause

The account does not have the read permission on CSS. You need to assign the required permissions to the IAM account.

Solution

Grant permissions to the IAM account. For details, see **Permissions Management**.

1.8 An Elasticsearch Cluster Reports An Error Message "unassigned shards all indices"

Symptom

An Elasticsearch cluster reports error message "unassigned shards all indices" and the cluster status is **red**.

Possible Causes

Unallocated shards exist in the current cluster.

Solution

Run the following command on the **Dev Tools** page of Kibana:

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

1.9 A Cross-Domain Error Is Reported When I Connect the es-head Plugin to an Elasticsearch Cluster

Solution

1. Check whether the network is available on the ECS where the es-head plugin is installed.
2. If the network is available, log in to the CSS management console.
3. Choose **Clusters** in the navigation pane. On the **Clusters** page, click the name of a cluster. The **Basic Information** page is displayed.
4. Choose **Parameter Configurations**, click **Edit**, and change the value of **http.cors.allow-origin** to **"*"**.

1.10 An Alarm Is Displayed When I Access Cerebro Through a Single-Node Cluster

Possible Causes

By default, indexes in single-node clusters have copies that cannot deliver requests.

Solution

On the **Dev Tools** page of Kibana, run the following commands to change the number of index copies to **0**:

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

1.11 Why Does My ECS Fail to Connect to a Cluster?

Perform the following steps to troubleshoot this problem:

1. Check whether the ECS instance and cluster are in the same VPC.
   - If they are, go to 2.
   - If they are not, create an ECS instance and ensure that the ECS instance is in the same VPC as the cluster.
2. View the security group rule setting of the cluster to check whether port **9200** (TCP protocol) is allowed or port **9200** is included in the port range allowed in both the outbound and inbound directions.
   - If it is allowed, go to 3.
   - If it is not allowed, switch to the VPC management console and configure the security group rule of the cluster to allow port **9200** in both the outbound and inbound directions.
3. Check whether the ECS instance has been added to a security group.
   - If the instance has been added to a security group, check whether the security group configuration rules are appropriate. You can view the **Security Group** information on the **Basic Information** tab page of the cluster. Then, go to step 4.
   - If the instance has not been added to the security group, go to the VPC page from the ECS instance details page, select a security group, and add the ECS to the group.
4. Check whether the ECS instance can connect to the cluster.

   ssh *<Private network address and port number of a node>*

   NOTE

   If the cluster contains multiple nodes, check whether the ECS can be connected to each node in the cluster.
   - If the connection is normal, the network is running properly.
   - If the connection still fails, contact technical support.

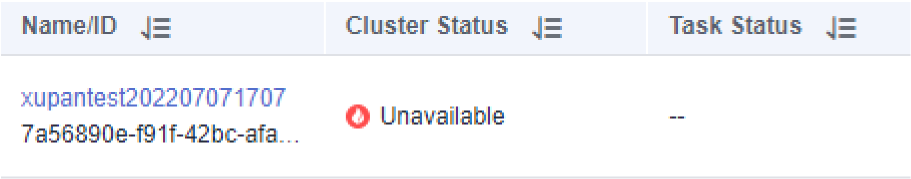
2 Unavailable Clusters

2.1 What Do I Do If My Cluster Status Is Unavailable?

Symptom

A CSS cluster status is **Unavailable**.

Figure 2-1 Unavailable cluster



Name/ID	Cluster Status	Task Status
xupantest202207071707 7a56890e-f91f-42bc-afa...	Unavailable	--

Cause Analysis and Solution

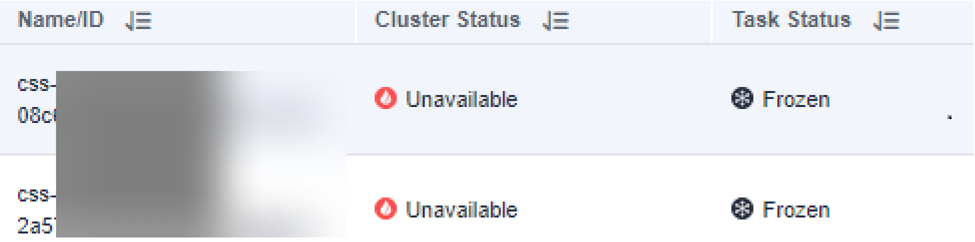
- If the task status is **Frozen**, see **A Cluster Is Frozen and Unavailable**.
- If task status is **Configuration error. Restart failed**, see **What Can I Do If My Custer Is Unavailable Due to an X-pack Parameter Configuration?**.
- If the node log contains the error message "master not discovered or elected yet, an election requires at least 2 nodes with ids [xxx, xxx, xxx, ...], have discovered [xxx...] which is not a quorum", see **A Cluster Is Unavailable Due to Improper Security Group Policy**.
- If the node log contains the error message "fatal error in thread [main], exitingjava.lang. NoClassDefFoundError: xxx/xxx/.../xxxPlugin at ...", see **A Cluster is Unavailable Due to Plugin Incompatibility**.
- If the cluster health status is red and the value of **unassigned shards** is not 0, the cluster has index shards that cannot be allocated. For details, see **A Cluster is Unavailable Due to Improper Shard Allocation**.
- If a cluster is unavailable after being restored or migrated, see **A Cluster is Unavailable Due to Incompatible Data Types**.
- If the node log contains the error message "OutOfMemoryError" and warning information "[gc][xxxxx] overhead spent [x.xs] collecting in the last [x.xs]", see **A Cluster is Unavailable Due to Heavy Load**.

2.2 A Cluster Is Frozen and Unavailable

Symptom

The cluster status is **Unavailable**, and the task status is **Frozen**.

Figure 2-2 Frozen



Name/ID	Cluster Status	Task Status
css- 08c [redacted]	Unavailable	Frozen
css- 2a5 [redacted]	Unavailable	Frozen

Possible Causes

The cluster is frozen because the account is in arrears or the subscription period of a yearly/monthly cluster has expired.

Procedure

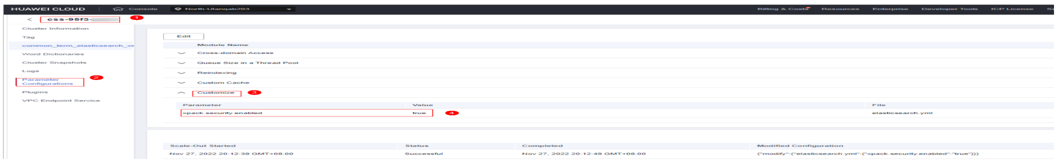
- **Pay-per-use clusters**
  a. On the top of the Huawei Cloud management console, click **Billing & Costs** to go to the billing center.
  b. Choose **Overview** to view the outstanding amount of the account.
     - If you are an IAM user, log in to the account and top up the account.
     - If you are an account user, click **Top Up** to go to the top-up page.
  c. After you top up your account, the cluster state automatically changes to **Available**.
- **Yearly/Monthly clusters**
  a. On the top of the Huawei Cloud management console, click **Billing & Costs** to go to the billing center.
  b. In the navigation pane, choose **Orders** > **Renewals**.
  c. On the **Renewals** page, set **Expires** to **Frozen** and **Service Type** to **Cloud Search Service** to search for the frozen cluster yearly/monthly package.
  d. In the **Operation** column of the frozen cluster yearly/monthly package, click **Renew**. On the displayed page, renew the package as required.
  e. After the status of the yearly/monthly package changes from **Frozen** to **Provisioned**, wait until the status of the cluster becomes **Available**.

2.3 What Can I Do If My Custer Is Unavailable Due to an X-pack Parameter Configuration?

Symptom

The cluster status is **Unavailable**, and the task status of the cluster is **Configuration error. Restart failed**.

Figure 2-3 Incorrect cluster configuration



Parameter	Value	File
xpack.security.enabled	true	elasticsearch.yml

Scale-Out Started	Status	Completed	Modified Configuration
Nov 27, 2022 20:12:39 GMT+08:00	Successful	Nov 27, 2022 20:12:49 GMT+08:00	{"modify":{"elasticsearch.yml":{"xpack.security.enabled":"true"}}}

Possible Causes

You have configured custom X-pack parameters. CSS does not support the X-pack function.

Procedure

1. On the **Clusters** page, click the name of the unavailable cluster. The **Cluster Information** page is displayed.
2. Choose **Parameter Configurations**, expand the **Customize** module, and check whether custom X-pack parameters exist.

   NOTE

   Examples of the custom X-pack parameters:

   - **xpack.security.enabled**: **true**
   - **xpack.security.http.ssl.enabled**: **true**
   - **xpack.security.http.ssl.keystore.path**: **http.p12**

   – If yes, go to the next step.

   – If no, contact technical support to locate the parameter configuration problem.

3. Delete the custom X-pack parameters.

   a. Click **Edit** and click **Delete** on the right of the parameter you want to be delete.

   b. Click **Submit**. In the displayed **Submit Configuration** dialog box, select **I understand that the modification will take effect after the cluster is restarted.** and click **OK**.

      If the **Status** is **Successful** in the parameter modification list, the modification has been saved.

   c. Return to the cluster list and choose **More** > **Restart** in the **Operation** column to restart the cluster and make the change take effect.

d. After the cluster is restarted, the cluster becomes available.

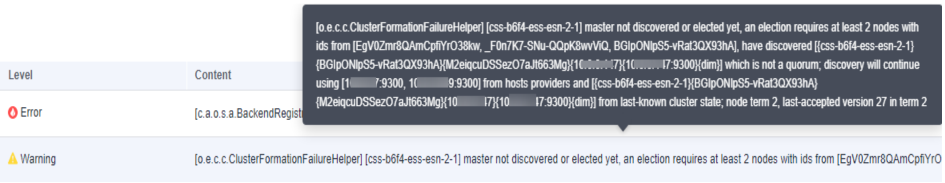
2.4 A Cluster Is Unavailable Due to Improper Security Group Policy

Symptom

The cluster status is **Unavailable**.

Click the cluster name to go to the cluster basic information page, choose **Logs**, and click the **Log Search** tab. The following error message is displayed: "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".

Figure 2-4 Node error logs



```
[o.e.c.c.ClusterFormationFailureHelper] [css-b6f4-ess-esn-2-1] master not discovered or elected yet, an election requires at least 2 nodes with ids from [EgV0Zmr8QAmCpfiYrO38kw, _F0n7K7-SNu-QQpK8wvViQ, BGIpONlpS5-vRat3QX93hA], have discovered [{css-b6f4-ess-esn-2-1}{BGIpONlpS5-vRat3QX93hA}{M2eiqcuDSSezO7aJt663Mg}{1[redacted]7}{1[redacted]7:9300}{dim}] which is not a quorum; discovery will continue using [1[redacted]7:9300, 1[redacted]9:9300] from hosts providers and [{css-b6f4-ess-esn-2-1}{BGIpONlpS5-vRat3QX93hA}{M2eiqcuDSSezO7aJt663Mg}{10[redacted]7}{10[redacted]7:9300}{dim}] from last-known cluster state; node term 2, last-accepted version 27 in term 2
```

Level	Content
Error	[c.a.o.s.a.BackendRegist
Warning	[o.e.c.c.ClusterFormationFailureHelper] [css-b6f4-ess-esn-2-1] master not discovered or elected yet, an election requires at least 2 nodes with ids from [EgV0Zmr8QAmCpfiYrO

Possible Causes

In the preceding error log, nodes in the cluster cannot communicate with each other and the cluster cannot select the active node. A possible cause is that the security group selected for the cluster does not enable the port 9300.

NOTE

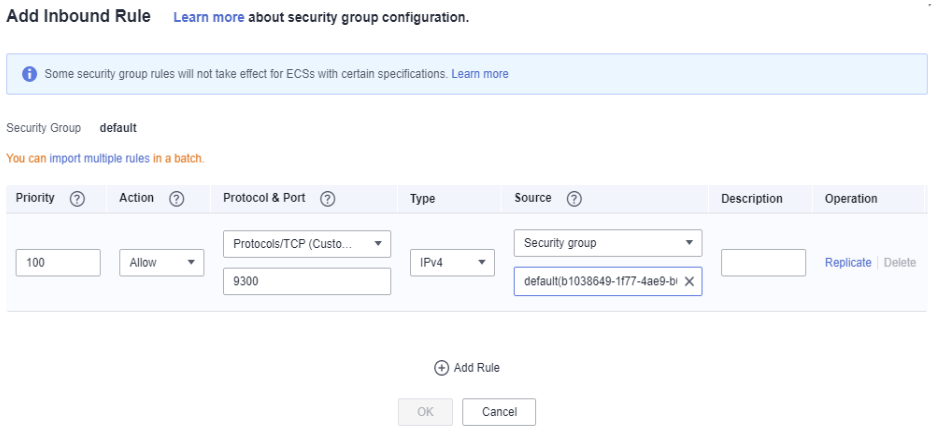
In CSS 7.6.2 or later, port 9300 is enabled on the subnet of the user VPC by default. The security group selected for the cluster must enable the port 9300 in the subnet to ensure communication between nodes.

Procedure

1. On the **Clusters** page, click the name of the unavailable cluster. The **Cluster Information** page is displayed.
2. Choose **Parameter Configurations** and click the security group name. The basic information page of the security group is displayed.
3. On the **Inbound Rules** and **Outbound Rules** tab pages, check whether there is a security group rule whose **Action** is **Allow**, **Protocol & Port** is **TCP:9300**, and **Type** is **IPv4**.
   - If yes, contact technical support to locate the problem.
   - If no, go to the next step.
4. Modify the security group configuration and enable the communication port 9300.
   a. On the basic information page of the security group, click the **Inbound Rules** tab.

b. Click **Add Rule**. In the **Add Inbound Rule** dialog box, set **Priority** to **100**, **Action** to **Allow**, **Protocol & Port** to **Protocols/TCP (Custom)** and **9300**, **Type** to **IPv4**, and **Source** to the name of the current security group.

Figure 2-5 Adding a security group rule



Add Inbound Rule Learn more about security group configuration.

Some security group rules will not take effect for ECSs with certain specifications. Learn more

Security Group default

You can import multiple rules in a batch.

Priority	Action	Protocol & Port	Type	Source	Description	Operation
100	Allow	Protocols/TCP (Custo... 9300	IPv4	Security group default(b1038649-1f77-4ae9-b		Replicate | Delete

c. Click **OK** to enable port 9300.

d. Repeat the preceding steps to enable the port 9300 on the **Outbound Rules** tab page.

5. After the port 9300 is enabled for the security group, wait until the cluster becomes available.

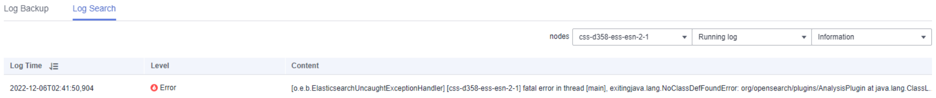
2.5 A Cluster is Unavailable Due to Plugin Incompatibility

Symptom

After a customized plugin is installed and the cluster is restarted, the cluster status changes to **Unavailable**.

Click the cluster name to go to the cluster basic information page, choose **Logs**, and click the **Log Search** tab. The following plug-in error message is displayed: "fatal error in thread [main], exitingjava.lang. NoClassDefFoundError: xxx/xxx/.../xxxPlugin at ...".

Figure 2-6 Node error logs



nodes css-d358-ess-esn-2-1 | Running log | Information

Log Time	Level	Content
2022-12-06T02:41:50,904	Error	[o.e.b.ElasticsearchUncaughtExceptionHandler] [css-d358-ess-esn-2-1] fatal error in thread [main], exitingjava.lang.NoClassDefFoundError: org/opensearch/plugins/AnalysisPlugin at java.lang.ClassL...

NOTE

CSS has disabled the custom plugin function. Clusters of earlier versions that have installed custom plugins may become unavailable.

Possible Causes

An installed custom plugin is incompatible with the current CSS cluster version. As a result, the Elasticsearch process cannot be started properly.

Procedure

1. On the **Clusters** page, click the name of the unavailable cluster. The **Cluster Information** page is displayed.
2. Choose **Plugins** and click the **Custom** tab. Determine whether you still need the customized plugin.
   - If yes, delete the plugin and reinstall it.
     1. In the custom plugin operation list, uninstall and delete the target plugin.
     2. Rectify the plugin fault based on the error information in the node logs. If the fault persists, contact technical support.
     3. After the plugin fault is rectified, upload and install the target plugin in the custom plugin operation list. If the plugin status is **Installed and to be effective upon cluster restart**, the plugin is successfully installed.
   - If no, delete the plugin.

     In the custom plugin operation list, uninstall and delete the target plugin.
3. Return to the cluster list and choose **More** > **Restart** in the **Operation** column of the cluster. After restart, the cluster becomes available.

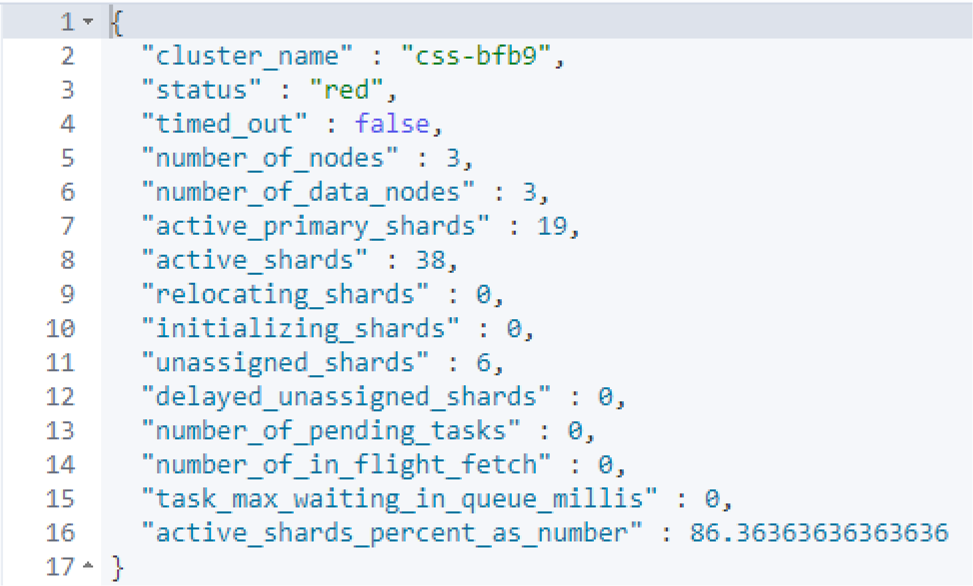
2.6 A Cluster is Unavailable Due to Improper Shard Allocation

Symptom

The cluster status is **Unavailable**.

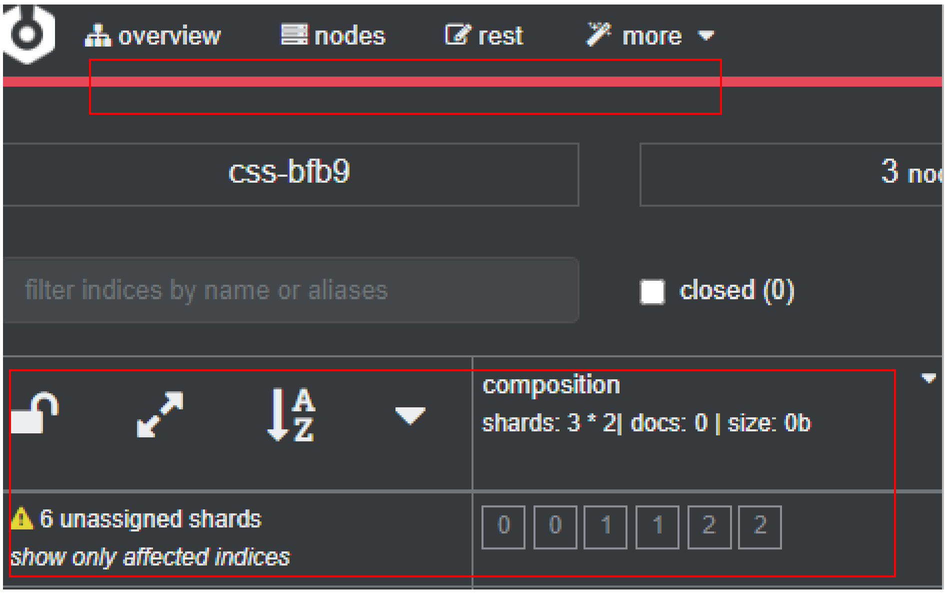
On the **Dev Tools** page of Kibana, run the **GET _cluster/health** command to check the cluster health status. In the output, the value of **status** is **red** and the value of **unassigned_shards** is not **0**. Alternatively, on the **Cerebro** page, click **overview** to view the index shard allocation on each data node. If the cluster status is **red** and the value of **unassigned shards** is not **0**, there are index shards that cannot be allocated in the cluster.

Figure 2-7 Cluster health status



```
{
  "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
}
```

Figure 2-8 Cerebro page



overview
nodes
rest
more
css-bfb9
3 no
filter indices by name or aliases
closed (0)
composition
shards: 3 * 2| docs: 0 | size: 0b
6 unassigned shards
show only affected indices
0
0
1
1
2
2

Possible Causes

Some index shards in the cluster are not properly allocated.

Procedure

Step 1: Determine the reason why the cluster is unavailable.

1. Use Kibana to access the faulty cluster. On the **Dev Tools** page of Kibana, run the **GET /_recovery?active_only=true** command to check whether the cluster is restoring the backup.
   - If **{"index_name":{"shards":[{"id":25,"type":"...** is returned, there are indexes being restored using backup. Wait until the backup restoration is complete. If the cluster status is still **Unavailable**, go to the next step.

– If **{ }** is returned, the cluster is not performing backup restoration. Go to the next step.

2. Run the **GET _cluster/allocation/explain?pretty** command to check why some index shards are not allocated based on the returned information.

Table 2-1 Parameter description

Parameter	Description
index	Index name
shard	Shard ID
current_state	Shard status
allocate_explanation	Shard allocation explanation
explanation	Explanation

Table 2-2 Faults description

Symptom	Causes	Procedure
explanation: **no allocations are allowed due to cluster setting [cluster.routing.allocation.enable=none]**	The cluster allocation policy forbids the allocation of all shards.	For details, see **cluster.routing.allocation.enable** in **Incorrect Shard Allocation Policy Configuration**.
explanation: **too many shards [3] allocated to this node for index [write08]index setting [index.routing.allocation.total_shards_per_node=3]**	The number of shards that can be allocated to each data node from a single index in the cluster is too small, which does not meet the index shard allocation requirements.	For details, see **index.routing.allocation.total_shards_per_node** in **Incorrect Shard Allocation Policy Configuration**.
explanation: **too many shards [31] allocated to this node, cluster setting [cluster.routing.allocation.total_shards_per_node=30]**	The number of shards that can be allocated to each data node in the cluster is too small.	For details, see **cluster.routing.allocation.total_shards_per_node** in **Incorrect Shard Allocation Policy Configuration**.

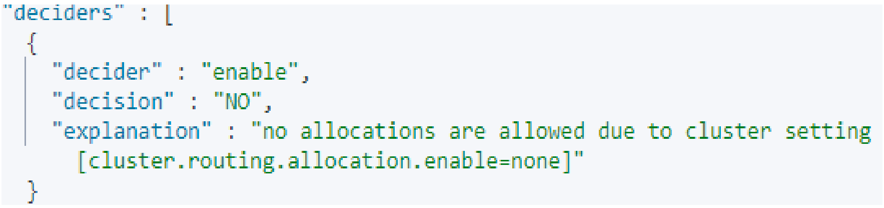
Symptom	Causes	Procedure
explanation: **node does not match index setting [index.routing. allocation. include] filters [box_type:"hot"]**	Index shards can only be allocated to data nodes labeled with **hot**. If a cluster has no nodes labeled with **hot**, shards cannot be allocated.	For details, see **index.routing.allocation.include** in **Incorrect Shard Allocation Policy Configuration**.
explanation: **node does not match index setting [index.routing. allocation. require] filters [box_type:"xl"]**	Index shards can only be allocated to data nodes with specified labels. If a cluster has no such nodes, shards cannot be allocated.	For details, see **index.routing.allocation.require** in **Incorrect Shard Allocation Policy Configuration**.
explanation: **[failed to obtain in-memory shard lock]**	Generally, this problem occurs when a node is removed from a cluster for a short time and then added back to the cluster. In addition, a thread is performing a long-term data writing to a shard, such as bulk or scroll. When the node is added to the cluster again, the master node cannot allocate the shard because the shard lock is not released.	For details, see **shard lock error**.
explanation: **node does not match index setting [index.routing.allocation.include] filters [_tier_preference:"data_hot OR data_warm OR data_cold"]**	The configuration of an index does not match the cluster version.	For details, see **Inconsistent index parameter version**.
explanation: **cannot allocate because all found copies of the shard are either stale or corrupt**	The data on index shards is damaged.	For details, see **Damaged primary shard data**.

Symptom	Causes	Procedure
explanation: **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: [6.9763809974193 24%]**	The node disk usage reaches the upper limit.	For details, see **Excessive disk usage**.

Step 2: Rectify the fault.

- **Incorrect shard allocation policy**
  - **cluster.routing.allocation.enable**
    1. If the value of **explanation** in the output is as follows, the current allocation policy of the cluster forbids the allocation of all shards.

Figure 2-9 Incorrect configuration of **allocation.enable**



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

    2. On the **Dev Tools** page of Kibana, run the following command to set **enable** to **all** to allow all shards to be allocated:

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

NOTE

The index-level configuration overwrites the cluster-level configuration. The parameters are described as follows:

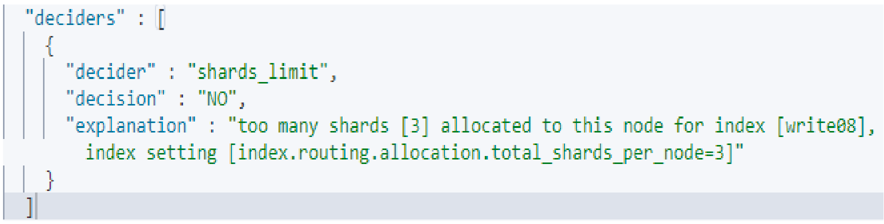
- **all**: Default value. All types of shards can be allocated.
- **primaries**: Only the primary shards can be allocated.
- **new_primaries**: Only the primary shards of the newly created index can be allocated.
- **none**: No shards can be allocated.

    3. Run the **POST _cluster/reroute?retry_failed=true** command to manually allocate shards. Wait until all index shards are allocated and the cluster status changes to **Available**.

- **index.routing.allocation.total_shards_per_node**

  i. If the value of **explanation** in the output is as follows, the value of **index.routing.allocation.total_shards_per_node** is too small and does not meet the index shard allocation requirements.

  Figure 2-10 Incorrect configuration of **index total_shards_per_node**

  ```
  "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. On the **Dev Tools** page of Kibana, run the following command to change the number of index shards that can be allocated to each node:

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

  NOTE

  Value of **index.routing.allocation.total_shards_per_node** = Number of **index_name** index shards/(Number of data nodes - 1)

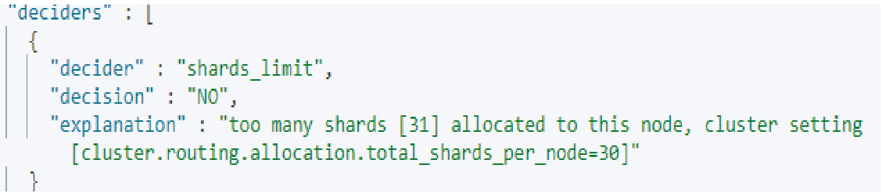
  Set this parameter to a relative large value. Assume that a cluster has 10 nodes, including five data nodes, two client nodes, and three master nodes. The number of shards of an index is 30. If **total_shards_per_node** is set to **4**, the total number of shards that can be allocated is: 4 x 5 = 20. Not all shards cannot be allocated. To allocate all shards in this index, at least six shards should be allocated to each data node (30 shards in total). In case a data node is faulty, at least eight shards should be allocated to each node.

  iii. Run the **POST _cluster/reroute?retry_failed=true** command to manually allocate shards. Wait until the index shards are allocated and the cluster status changes to **Available**.

- **cluster.routing.allocation.total_shards_per_node**

  i. If the value of **explanation** in the output is as follows, the number of shards that can be allocated to each data node in the cluster is too small.

  Figure 2-11 Incorrect configuration of **cluster total_shards_per_node**

  ```
  "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. The value of **cluster.routing.allocation.total_shards_per_node** indicates the maximum number of shards that can be allocated to each data node in a cluster. The default value of this parameter is **1000**. On the **Dev Tools** page of Kibana, run the following command to specify the **cluster.routing.allocation.total_shards_per_node** parameter:

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

iii. In most cases, the problem occurs because **index.routing.allocation.total_shards_per_node** is mistakenly set to **cluster.routing.allocation.total_shards_per_node**. Run the following command to specify the **index.routing.allocation.total_shards_per_node** parameter:

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

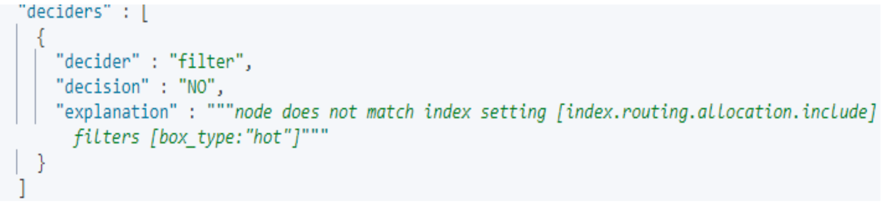
> NOTE
>
> Both of the following parameters are used to limit the maximum number of shards that can be allocated to a single data node:
>
> - **cluster.routing.allocation.total_shards_per_node** is used to limit shard allocation at the cluster level.
> - **index.routing.allocation.total_shards_per_node** is used to limit shard allocation at the index level.

iv. Run the **POST _cluster/reroute?retry_failed=true** command to manually allocate shards. Wait until the index shards are allocated and the cluster status changes to **Available**.

– **index.routing.allocation.include**

i. If the value of **explanation** in the output is as follows, index shards can only be allocated to data nodes with the **hot** label. If no data nodes in the cluster are labeled with **hot**, shards cannot be allocated.

Figure 2-12 Incorrect configuration of **include**



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

ii. On the **Dev Tools** page of Kibana, run the following command to cancel the configuration:

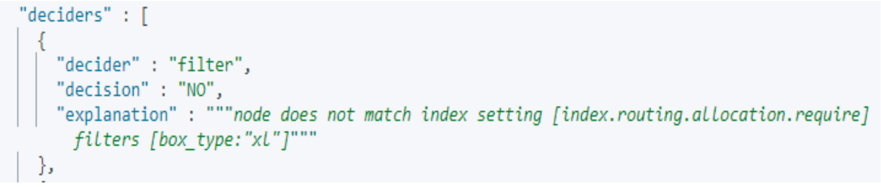
```
PUT index_name/_settings
{
  "index.routing.allocation.include.box_type": null
}
```

iii. Run the **POST _cluster/reroute?retry_failed=true** command to manually allocate shards. Wait until the index shards are allocated and the cluster status changes to **Available**.

- **index.routing.allocation.require**

  i. If the value of **explanation** in the output is as follows, shards can only be allocated to data nodes with specified labels. If no nodes in the cluster have such labels, the shards cannot be allocated.

  Figure 2-13 Incorrect configuration of **require**

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

  ii. On the **Dev Tools** page of Kibana, run the following command to cancel the configuration:

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

  iii. Run the **POST _cluster/reroute?retry_failed=true** command to manually allocate shards. Wait until the index shards are allocated and the cluster status changes to **Available**.

- **Shard lock error**

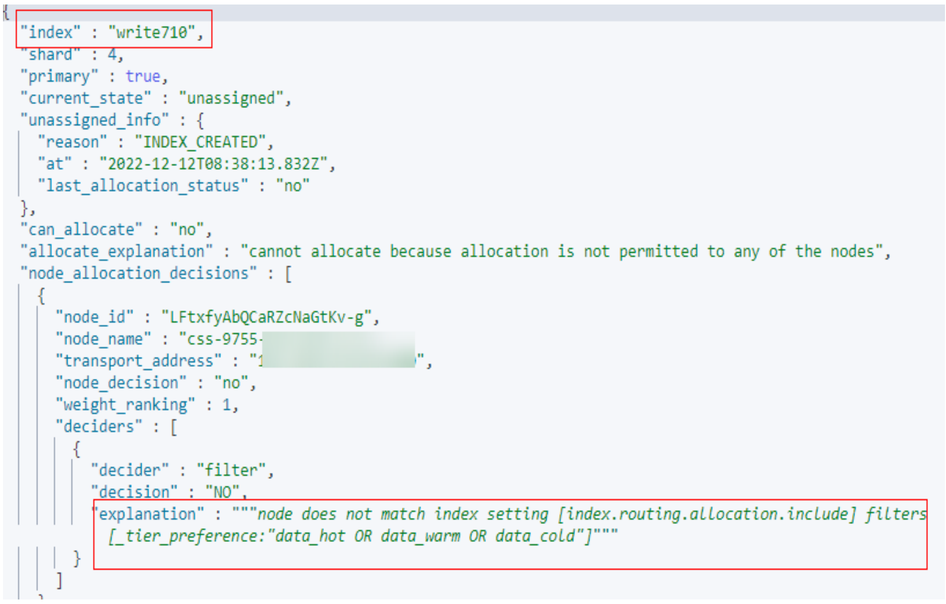
  a. In the output, **explanation** contains **[failed to obtain in-memory shard lock]**. This problem usually occurs when a node is removed from a cluster for a short time and then added back to the cluster, and a thread is performing a long-term data writing to a shard, such as bulk or scroll. When the node is added to the cluster again, the master node cannot allocate the shard because the shard lock is not released.

  b. This problem does not cause fragment data loss. You only need to allocate the shard again. On the **Dev Tools** page of Kibana, run the **POST /_cluster/reroute?retry_failed=true** command to manually allocate the unallocated shard. Wait until the index shards are allocated and the cluster status changes to **Available**.

- **Inconsistent index setting and node version**

  a. The value of **index** and **explanation** in the output are as follows, indicating that the parameter configuration of an index does not match the node version.

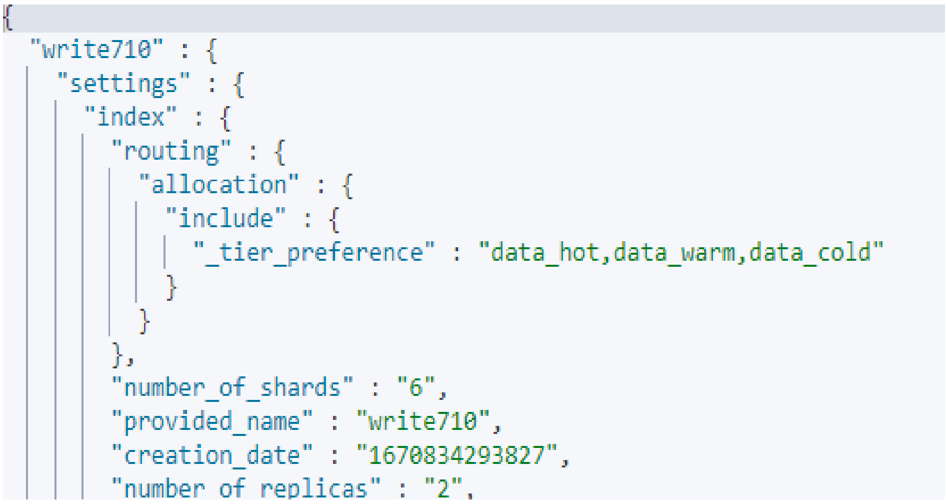
Figure 2-14 Inconsistent index configuration



```
{
  "index" : "write710",
  "shard" : 4,
  "primary" : true,
  "current_state" : "unassigned",
  "unassigned_info" : {
    "reason" : "INDEX_CREATED",
    "at" : "2022-12-12T08:38:13.832Z",
    "last_allocation_status" : "no"
  },
  "can_allocate" : "no",
  "allocate_explanation" : "cannot allocate because allocation is not permitted to any of the nodes",
  "node_allocation_decisions" : [
    {
      "node_id" : "LFtxfyAbQCaRZcNaGtKv-g",
      "node_name" : "css-9755-[redacted]
      "transport_address" : "",
      "node_decision" : "no",
      "weight_ranking" : 1,
      "deciders" : [
        {
          "decider" : "filter",
          "decision" : "NO",
          "explanation" : """node does not match index setting [index.routing.allocation.include] filters
          [_tier_preference:"data_hot OR data_warm OR data_cold"]"""
        }
      ]
```

b. Run the **GET index_name/_settings** command to check the index configuration. In the output, check whether the index features match the node version.

Figure 2-15 Index configuration



```
{
  "write710" : {
    "settings" : {
      "index" : {
        "routing" : {
          "allocation" : {
            "include" : {
              "_tier_preference" : "data_hot,data_warm,data_cold"
            }
          }
        },
        "number_of_shards" : "6",
        "provided_name" : "write710",
        "creation_date" : "1670834293827",
        "number_of_replicas" : "2",
```

For example, assume that a cluster version is **7.9.3**. The index feature **index.routing.allocation.include._tier_preference** is supported by clusters of the version later than **7.10**. If you use this feature in a cluster of the version earlier than **7.10**, index shards cannot be allocated. As a result, the cluster is unavailable.

c. Determine whether the inapplicable feature is mandatory for the cluster.

- If yes, create a cluster of the required version and restore the data of the old cluster to the new cluster using the backup.
- If no, go to the next step.

d. Run the following command to remove the inapplicable index feature:

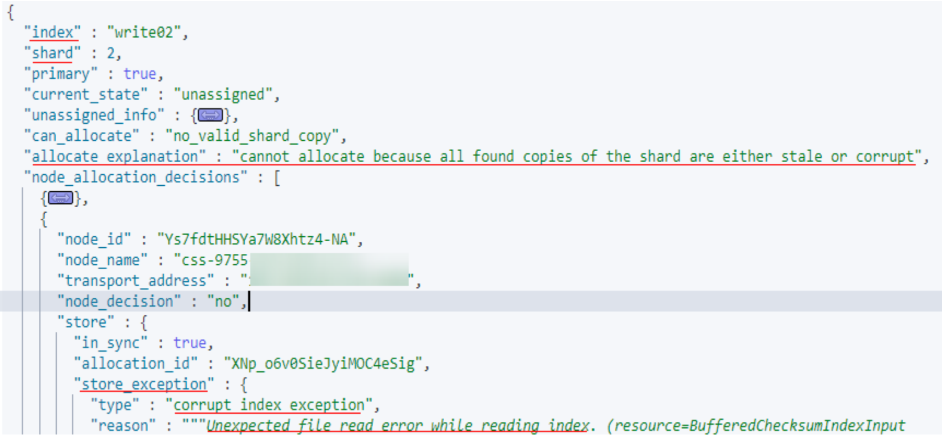
```
PUT /index_name/_settings
{
  "index.routing.allocation.include._tier_preference": null
}
```

e. Run the **POST /_cluster/reroute?retry_failed=true** command to manually allocate the unallocated shards. Wait until the index shards are allocated and the cluster status changes to **Available**.

- **Damaged primary shard data**

a. The values of **index**, **shard**, **allocate_explanation**, and **store_exception** in the output are as follows, indicating that the data of a shard in an index is damaged.

Figure 2-16 Damaged primary shard data



```
{
  "index" : "write02",
  "shard" : 2,
  "primary" : true,
  "current_state" : "unassigned",
  "unassigned_info" : {⟺},
  "can_allocate" : "no_valid_shard_copy",
  "allocate_explanation" : "cannot allocate because all found copies of the shard are either stale or corrupt",
  "node_allocation_decisions" : [
    {⟺},
    {
      "node_id" : "Ys7fdtHHSYa7W8Xhtz4-NA",
      "node_name" : "css-9755[redacted]
      "transport_address" : "",
      "node_decision" : "no",
      "store" : {
        "in_sync" : true,
        "allocation_id" : "XNp_o6v0SieJyiMOC4eSig",
        "store_exception" : {
          "type" : "corrupt_index_exception",
          "reason" : """Unexpected file read error while reading index. (resource=BufferedChecksumIndexInput
```

b. When the index data is damaged or the primary backup of a shard is lost, run the following command to define an empty shard and specify the node to be allocated:

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

NOTICE

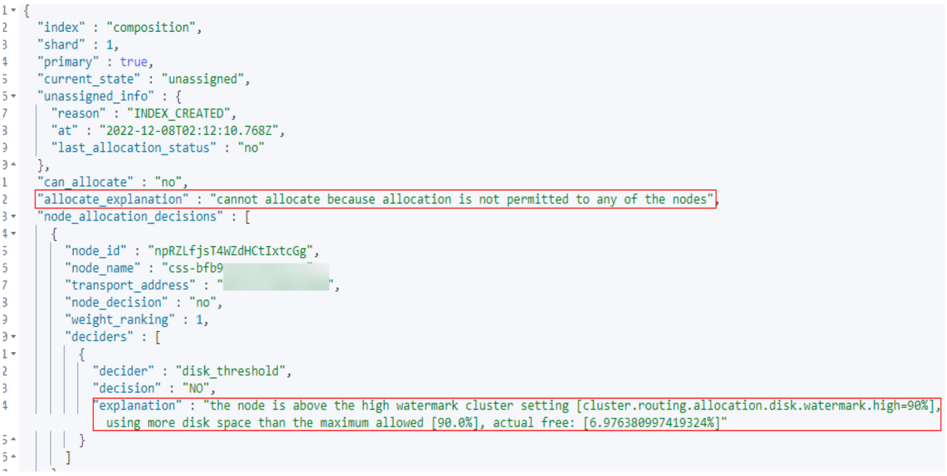
Data in the corresponding shard will be completely cleared. Exercise caution when performing this operation.

c. After index shards are reallocated, the cluster status becomes **Available**.

- **Excessive disk usage**

a. The output is as follows. The value of **allocate_explanation** indicates that shards of an index cannot be allocated to any data node, and the value of **explanation** indicates that node disk usage reaches the upper limit.

Figure 2-17 Query result



```
{
  "index" : "composition",
  "shard" : 1,
  "primary" : true,
  "current_state" : "unassigned",
  "unassigned_info" : {
    "reason" : "INDEX_CREATED",
    "at" : "2022-12-08T02:12:10.768Z",
    "last_allocation_status" : "no"
  },
  "can_allocate" : "no",
  "allocate_explanation" : "cannot allocate because allocation is not permitted to any of the nodes",
  "node_allocation_decisions" : [
    {
      "node_id" : "npRZLfjsT4WZdHCtIxtcGg",
      "node_name" : "css-bfb9[redacted]",
      "transport_address" : "[redacted]",
      "node_decision" : "no",
      "weight_ranking" : 1,
      "deciders" : [
        {
          "decider" : "disk_threshold",
          "decision" : "NO",
          "explanation" : "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%]"
        }
      ]
```

NOTE

- If the disk usage of a node exceeds 85%, new shards will not be allocated to this node.
- If the disk usage of a node exceeds 90%, the cluster attempts to migrate the shards of this node to other data nodes with low disk usage. If data cannot be migrated, the system forcibly sets the **read_only_allow_delete** attribute for each index in the cluster. In this case, data cannot be written to indexes, and the indexes can only be read or deleted.
- A node may be disconnected due to high disk usage. After the node automatically recovers, once the cluster is overloaded, the cluster may fail to respond when you call the Elasticsearch API to query the cluster status. If the cluster status cannot be updated in a timely manner, the cluster becomes **Unavailable**.

b. Increase the available disk capacity of the cluster.

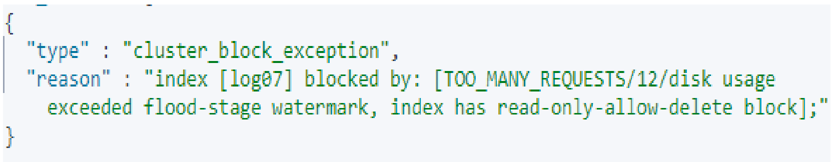
- On the **Dev Tools** page of Kibana, run the **DELETE index_name** command to clear invalid data in the cluster to release disk space.
- Temporarily reduce the number of index copies. After the disk or node capacity is expanded, change the number of index copies back to the original value.

1) On the **Dev Tools** page of Kibana, run the following command to temporarily reduce the number of index copies:

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

The output is as follows.

Figure 2-18 Index status of **read-only-allow-delete**



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

The disk usage exceeds the maximum value allowed by the disk space. The system forcibly sets the **read_only_allow_delete** attribute for all indexes in the cluster. Run the following command to set the attribute value to **null**, and then run the command in **b.1)** to reduce the number of index copies:

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

2) Increase the number of nodes or node storage capacity of the cluster by referring to **Scaling Out a Cluster**.

3) After the scale-out is complete, run the command in step **b.1)** to change the number of index copies back. After all index shards are allocated, the cluster status changes to **Available**.

2.7 A Cluster is Unavailable Due to Incompatible Data Types

Symptom

After a cluster is restored using backup or migrated, the cluster status changes to **Unavailable**.

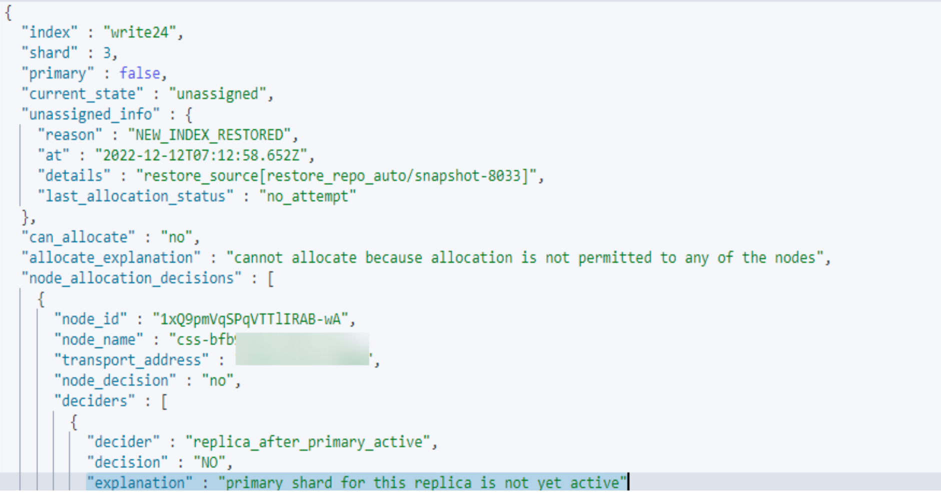
Possible Causes

Some types of the restored data are not supported by the destination cluster. For example, some plugins or settings are installed in the old cluster, which are not supported by the new cluster. As a result, index shards cannot be allocated.

Procedure

1. On the **Dev Tools** page of Kibana, run the **GET _cluster/allocation/explain?pretty** command to check the reason why index shards are not allocated.
2. The value of **index** and **explanation** in the output are as follows, indicating that the shard backup is not activated.

Figure 2-19 Unallocated index shards



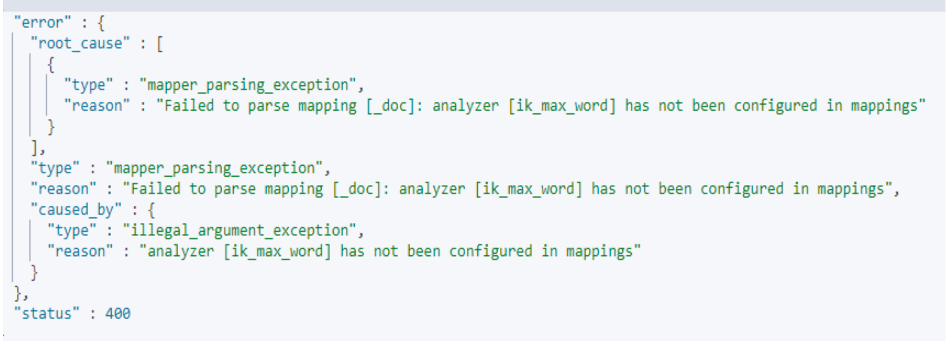
```
{
  "index" : "write24",
  "shard" : 3,
  "primary" : false,
  "current_state" : "unassigned",
  "unassigned_info" : {
    "reason" : "NEW_INDEX_RESTORED",
    "at" : "2022-12-12T07:12:58.652Z",
    "details" : "restore_source[restore_repo_auto/snapshot-8033]",
    "last_allocation_status" : "no_attempt"
  },
  "can_allocate" : "no",
  "allocate_explanation" : "cannot allocate because allocation is not permitted to any of the nodes",
  "node_allocation_decisions" : [
    {
      "node_id" : "1xQ9pmVqSPqVTTlIRAB-wA",
      "node_name" : "css-bfb[redacted]
      "transport_address" : ",
      "node_decision" : "no",
      "deciders" : [
        {
          "decider" : "replica_after_primary_active",
          "decision" : "NO",
          "explanation" : "primary shard for this replica is not yet active"
```

3. Modify the configuration of the index and run the following command to set the number of copies to **0**:

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

The value of **reason** indicates that some types of the data that is being restored are not supported by the CSS cluster.

Figure 2-20 Incompatible data



```
"error" : {
  "root_cause" : [
    {
      "type" : "mapper_parsing_exception",
      "reason" : "Failed to parse mapping [_doc]: analyzer [ik_max_word] has not been configured in mappings"
    }
  ],
  "type" : "mapper_parsing_exception",
  "reason" : "Failed to parse mapping [_doc]: analyzer [ik_max_word] has not been configured in mappings",
  "caused_by" : {
    "type" : "illegal_argument_exception",
    "reason" : "analyzer [ik_max_word] has not been configured in mappings"
  }
},
"status" : 400
```

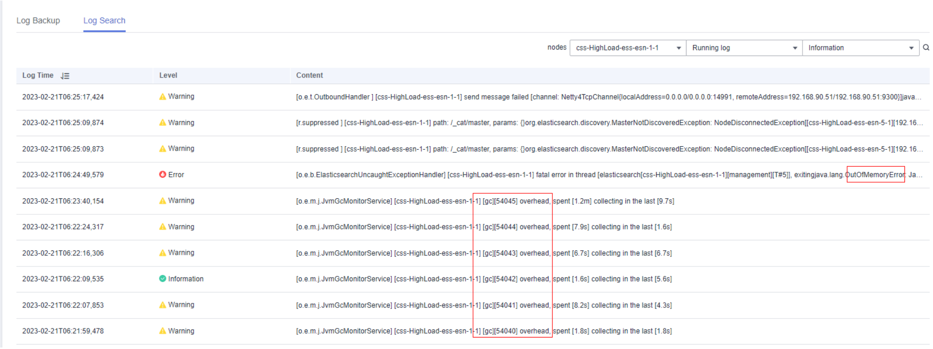
4. According to the root cause, delete the data type that is not supported by the current CSS cluster or select a CSS cluster version that supports the data type, and then restore the backup or migrate the data.

2.8 A Cluster is Unavailable Due to Heavy Load

Symptom

The cluster status is **Unavailable**. Click the cluster name to go to the cluster basic information page, choose **Logs**, and click the **Log Search** tab. The error message "OutOfMemoryError" and alarm "[gc][xxxxx] overhead spent [x.xs] collecting in the last [x.xs]" are displayed.

Figure 2-21 Out-of-memory caused by frequent garbage collection



nodes css-HighLoad-ess-esn-1-1 | Running log | Information

Log Time	Level	Content
2023-02-21T06:25:17,424	Warning	[o.e.t.OutboundHandler] [css-HighLoad-ess-esn-1-1] send message failed [channel: Netty4TcpChannel{localAddress=0.0.0.0/0.0.0.0:14991, remoteAddress=192.168.90.51/192.168.90.51:9300}]java...
2023-02-21T06:25:09,874	Warning	[r.suppressed] [css-HighLoad-ess-esn-1-1] path: /_cat/master, params: {}org.elasticsearch.discovery.MasterNotDiscoveredException: NodeDisconnectedException[[css-HighLoad-ess-esn-5-1][192.16...
2023-02-21T06:25:09,873	Warning	[r.suppressed] [css-HighLoad-ess-esn-1-1] path: /_cat/master, params: {}org.elasticsearch.discovery.MasterNotDiscoveredException: NodeDisconnectedException[[css-HighLoad-ess-esn-5-1][192.16...
2023-02-21T06:24:49,579	Error	[o.e.b.ElasticsearchUncaughtExceptionHandler] [css-HighLoad-ess-esn-1-1] fatal error in thread [elasticsearch[css-HighLoad-ess-esn-1-1][management][T#5]], exitingjava.lang.OutOfMemoryError: Ja...
2023-02-21T06:23:40,154	Warning	[o.e.m.j.JvmGcMonitorService] [css-HighLoad-ess-esn-1-1] [gc][54045] overhead, spent [1.2m] collecting in the last [9.7s]
2023-02-21T06:22:24,317	Warning	[o.e.m.j.JvmGcMonitorService] [css-HighLoad-ess-esn-1-1] [gc][54044] overhead, spent [7.9s] collecting in the last [1.6s]
2023-02-21T06:22:16,306	Warning	[o.e.m.j.JvmGcMonitorService] [css-HighLoad-ess-esn-1-1] [gc][54043] overhead, spent [6.7s] collecting in the last [6.7s]
2023-02-21T06:22:09,535	Information	[o.e.m.j.JvmGcMonitorService] [css-HighLoad-ess-esn-1-1] [gc][54042] overhead, spent [1.6s] collecting in the last [5.6s]
2023-02-21T06:22:07,853	Warning	[o.e.m.j.JvmGcMonitorService] [css-HighLoad-ess-esn-1-1] [gc][54041] overhead, spent [8.2s] collecting in the last [4.3s]
2023-02-21T06:21:59,478	Warning	[o.e.m.j.JvmGcMonitorService] [css-HighLoad-ess-esn-1-1] [gc][54040] overhead, spent [1.8s] collecting in the last [1.8s]

Possible Causes

The cluster is overloaded due to a large number of queries or task stacking. When the heap memory is insufficient, tasks cannot be allocated and garbage collection is frequently triggered. As a result, the Elasticsearch process exits abnormally.

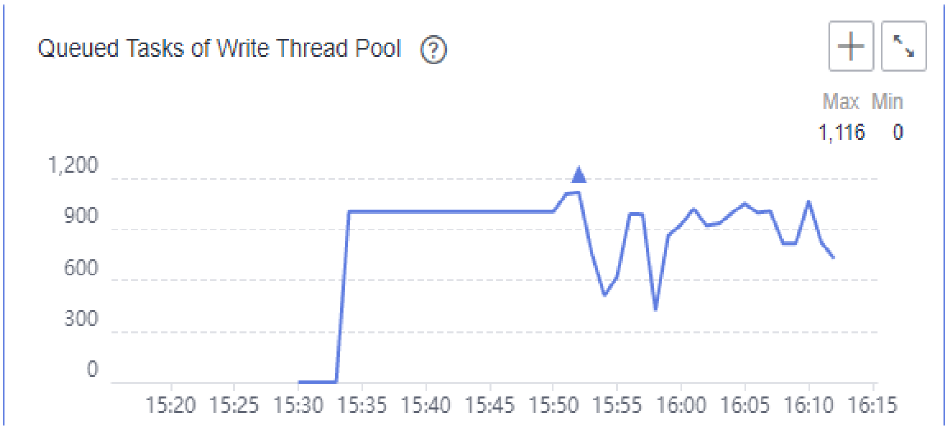
Procedure

NOTE

If a cluster is overloaded for a long time, data write and query may be slow. You are advised to upgrade the node specifications, add new nodes, or scale out the node capacity. For details about how to upgrade the node specifications, increase the number of nodes, or expand the storage capacity of nodes, see **Scaling Out a Cluster**.

1. Check whether tasks are stacked in the cluster.
   - Method 1: On the **Dev Tools** page of Kibana, run the following commands to check whether tasks are being delayed:
     ```
     GET /_cat/thread_pool/write?v
     GET /_cat/thread_pool/search?v
     ```
     If the value of **queue** is not 0, tasks are stacked.
     ```
     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
     ```
   - Method 2: In the cluster management list, click **More** > **View Metric** in the **Operation** column of the cluster. On the displayed page, check the total number of queued tasks in the search thread pool and write thread pool. If the number of queued tasks is not 0, tasks are being delayed.

     Figure 2-22 Queued Tasks of White Thread Pool

   - If a large number of tasks are stacked in the cluster, perform the following steps to optimize the cluster:
     - On the **Logs** page of the cluster, if a large number of slow query logs exist before the node is out of memory, optimize the query statement based on the site requirements.

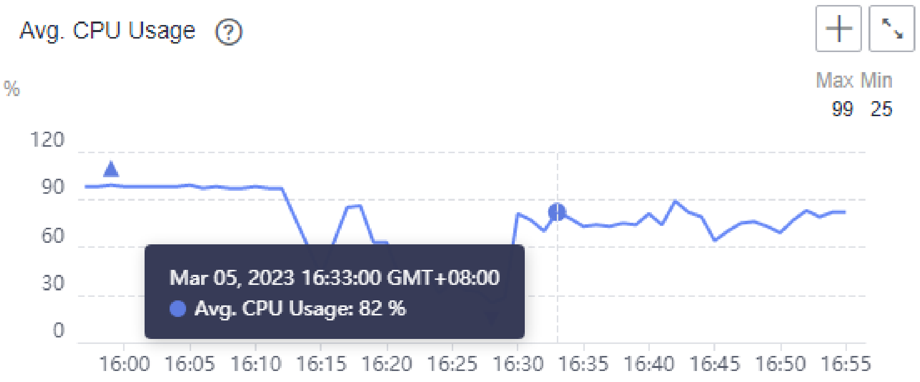
- On the **Logs** page of the cluster, if the error message "Inflight circuit break" or "segment can't keep up" is displayed, the circuit breaker may be triggered due to heavy write pressure. If the amount of written data (write rate) increases sharply recently, you need to properly arrange the write peak time window based on service requirements.

- If no task is stacked in the cluster or the cluster is still unavailable after optimization, go to the next step to check whether the cluster is overloaded.

2. Check whether the cluster is overloaded.

   In the cluster management list, click **More** > **View Metric** in the **Operation** column of the cluster. On the displayed page, view metrics related to the CPU and heap memory, such as average CPU usage and average JVM heap usage. If the average CPU usage exceeds 80% or the average JVM heap usage exceeds 70%, the cluster is under heavy pressure.

   Figure 2-23 Avg. CPU Usage



- If the cluster is overloaded, reduce the client request sending rate or expand the cluster capacity.
- If the cluster pressure is not overloaded or the cluster is still unavailable after the request sending rate is reduced, go to the next step to check whether a large amount of cache exists in the cluster.

3. On the **Dev Tools** page of Kibana, run the following command to check whether the cluster's cache has cached a large number of requests:

```
GET /_cat/nodes?v&h=name,queryCacheMemory,fielddataMemory,requestCacheMemory
```

- If the value of **queryCacheMemory**, **fielddataMemory**, or **requestCacheMemory** in the output exceeds 20% of the heap memory, run the **POST _cache/clear** command to clear the cache. The cached data is generated during data query to speed up the query. After the cached data is cleared, the query latency may increase.

```
name                   queryCacheMemory fielddataMemory requestCacheMemory
css-0323-ess-esn-1-1              200mb           1.6gb              200mb
```

NOTE

You can run the following command to query the maximum heap memory of each node:

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

name indicates the node name and *ip* indicates the IP address of the node.

- If the cluster is still overloaded after the optimization, contact technical support.

3 Data Import and Export

3.1 What Do I Do If Logs Cannot Be Written to CSS Due to High CPU Usage of Elasticsearch?

Symptom

The CPU usage of Elasticsearch is high, an error message "Elasticsearch Unreachable" is displayed on Logstash, and logs cannot be written to Elasticsearch.

Possible Causes

The customer index has only one shard. The node of the shard is overloaded, and the job queue is full. Later jobs are rejected.

Procedure

1. Log in to the CSS management console.
2. Choose **Clusters** in the navigation pane. The cluster list is displayed.
3. Locate the target cluster and choose **More** > **Access Cerebro** in the **Operation** column.

   If the cluster is in security mode, you need to enter the login username (**admin**) and password.
4. In Cerebro, view the number of shards in the cluster and metrics such as the CPU, load, head, and dis of each node.
5. Analyze the possible causes based on metrics and tune your system accordingly.
   a. Increase the number of queues and reduce rejected jobs by changing the value of **write.queue_size**.
      i. Click the name of the target cluster whose parameters you want to modify. The basic information page of the cluster is displayed.
      ii. Click **Parameter Configurations**, search for **write.queue_size**, and change its value.

If this parameter does not exist, add it in the **Customize** area. For details, see **Configuring Parameters**.

b. Rebuild the indexes to ensure that the number of shards is greater than that of nodes in the cluster.

6. If the number of shards and queues are appropriate but the CPU usage and load are still high, you are advised to scale out the cluster.

3.2 Why Is an Error Reported When Data Is Pushed to CSS After Logstash Is Deployed on an ECS?

Symptom

After Logstash is deployed on an ECS, an error is reported when data is pushed to CSS. The error message is as follows:

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

Possible Causes

CSS currently does not integrate the x-pack plugin. When you access CSS after deploying Logstash, the system will check whether x-pack is enabled for CSS.

Procedure

1. Delete the x-pack directory in Logstash.
2. Add the configuration item **ilm_enabled => false** to **elasticsearch** under the **output** tag in the Logstash configuration file.
3. Push data to CSS again.

3.3 "Could not write all entries" Is Reported When I Use ES-Hadoop to Import Data

Issue Analysis

The bulk thread pool of the Elasticsearch background supports a maximum of 200 requests. Excess requests will be rejected.

Solution

1. Set a proper number of the concurrent write requests from the client as required. ES-Hadoop has a retry mechanism for the rejected HTTP requests. You can modify the following parameters:
   - **es.batch.write.retry.count**: The default value for retry times is **3**.
   - **es.batch.write.retry.wait**: The waiting time for each attempt is 10s.
2. If you do not require real-time query, you can adjust the shard refresh time (once per second by default) to improve the write speed.

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

4 Functions

4.1 Why Does Index Backup Fail?

Index backup is implemented by creating cluster snapshots. If index backup fails, perform the following steps to troubleshoot this problem:

Check Whether the Account or IAM User Has the Index Backup Permissions

1. Log in to the IAM management console.
2. Check the user group that the account or the IAM user belongs to.

   For details, see **Viewing and Modifying User Information** in the *Identity and Access Management User Guide*.
3. Check whether the permissions assigned to the user group include the following two permissions: **OBS Administrator** for project **OBS** in region **Global service** and **Elasticsearch Administrator** for the current region.

   For details, see **Viewing and Modifying User Group Information** in the *Identity and Access Management User Guide*.
   - If neither of the preceding permissions has been assigned to the user group, go to **4**.
   - If both the preceding permissions have been assigned to the group, contact Huawei Cloud technical support.
4. Add the following permissions to the user group: **OBS Administrator** for project **OBS** in region **Global service**, and **Elasticsearch Administrator** for the current region.

   For details, see **Viewing and Modifying User Group Information** in the *Identity and Access Management User Guide*.

4.2 Why Is the Word Dictionary Function Not Working?

Perform the following steps to troubleshoot this problem:

Check When the Target Cluster Was Created

Step 1 Log in to the CSS management console.

Step 2 In the left navigation pane, choose **Clusters**.

Step 3 On the **Clusters** page, locate the target cluster where you want to configure a custom word dictionary and view its creation time.

- If the creation time is earlier than March 10, 2018, no further action is required. The function is not working because the custom word dictionary function was unavailable at that time.
- If the creation time is later than March 10, 2018, check whether the account or IAM user used for logging in to the management console has the custom word dictionary permission. For details, see **Check Whether the Account or IAM User Has the Custom Word Dictionary Permission**.

----End

Check Whether the Account or IAM User Has the Custom Word Dictionary Permission

1. Log in to the IAM management console.
2. Check the user group that the account or the IAM user belongs to.

   For details, see **Viewing or Modifying IAM User Information** in *Identity and Access Management User Guide*.
3. Check whether the permissions assigned to the user group include the following two permissions: **Tenant Administrator** for project **OBS** for region **Global service** and **Elasticsearch Administrator** for the current region.

   For details, see **Viewing or Modifying User Group Information** in *Identity and Access Management User Guide*.
   - If neither of the preceding permissions has been assigned to the user group, go to **4**.
   - If both the preceding permissions have been assigned to the group, contact the customer service personnel.
4. Add the following permissions to the user group: **Tenant Administrator** for project **OBS** in region **Global service** and **Elasticsearch Administrator** in the current region.

   For details, see **Viewing or Modifying User Group Information** in *Identity and Access Management User Guide*.

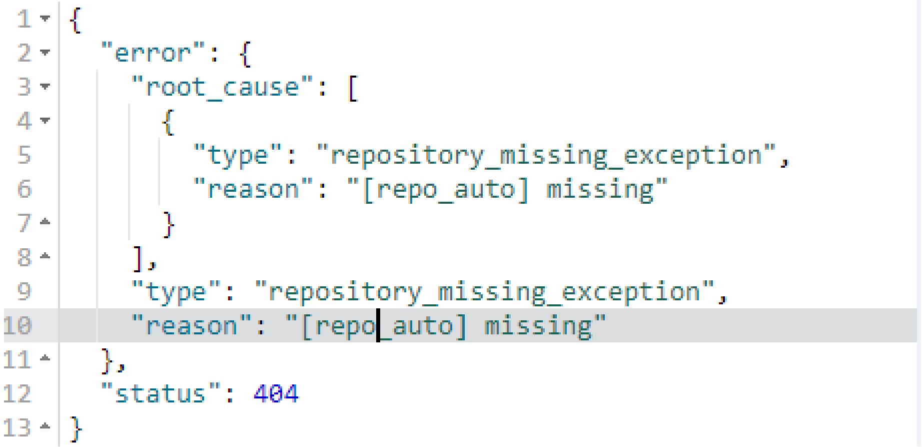
4.3 What Do I Do If the Snapshot Repository Cannot Be Found?

1. On the **Clusters** page of the CSS management console, locate the target cluster and click **Access Kibana** in the **Operation** column.
2. In the navigation pane of Kibana, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.

   Enter the code as required in the left pane, click ▶ to execute the command, and view the result in the right pane.

3. If no information is returned after the **GET _snapshot/_all** is executed, or if the error message shown in **Figure 4-1** is displayed after the **GET _snapshot/repo_auto/_all** command is executed, it indicates that no snapshot is configured. In this case, configure the snapshot again.

Figure 4-1 Returned information



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

4. Click the name of the target cluster. On the displayed cluster details page, click the **Cluster Snapshots** tab.
5. Click ✎ next to **Basic Configuration** to modify the basic configurations.
6. After the modification is complete, click **OK**.

   If the repository still cannot be found after the modification, try modifying and restoring the backup path, and save the settings again.

4.4 What Do I Do If a Cluster Is Always in the Snapshot Creation State?

Possible causes are as follows:

- The cluster is heavily loaded, and snapshot creation takes a long time.

  The default snapshot creation speed of a single node is 40 MB/s. The speed will be lower if the cluster is busy. You can query the status of a snapshot by referring to preceding sections.

  You can run the **GET _snapshot/repo_auto/snapshot-name** command to check the number of shards that are being backed up. You can also terminate snapshot creation via APIs.

  Solution: Wait for the snapshot creation to complete, or terminate the task.

- Failed to update snapshot information.

  Elasticsearch stores ongoing snapshot information in the cluster state. After a snapshot is created, its state needs to be updated, but Elasticsearch may fail to update the snapshot state due to high memory usage. Elasticsearch does not retry failed updates, so the snapshot remains in the Creating state.

  Solution: Call the snapshot deletion API.

- Temporary AKs or SKs expire.

CSS uses an agency to write data in Elasticsearch to OBS. To create a snapshot repository, you need to use the agency to obtain a temporary AK and SK, and configure them in the repository. Temporary AKs and SKs have a validity period (24 hours). Snapshot creation will fail if it does not complete within 24 hours. In this case, the repository cannot be updated, queried, and deleted, and the cluster state information cannot be deleted manually or by a rolling restart. To delete residual snapshot information, perform a normal restart.

Solution: Currently, residual snapshot information can only be deleted in a normal restart. CSS will provide a termination interface to rectify the fault.

4.5 How Do I Back Up Large Amounts of Data Using Snapshots?

Improve snapshot backup configurations to ensure that each snapshot takes less than 24 hours to create. For example:

1. Specify indexes and back up data in batches. The default value is *, indicating that all indexes are backed up.
2. Use a custom snapshot repository.
   a. Create a custom repository.

      CSS provides the **repo_auto** repository by default. You can create one by calling the following API:

```
PUT _snapshot/my_backup
{
    "type" : "obs",
    "settings" : {
      "bucket" : "css-backup-name",    // Bucket name
      "base_path" : "css_backup/711/",  // Backup path
      "chunk_size" : "2g",
      "endpoint" : "obs.xxx.com:443", //OBS domain name address
      "region" : "xxx",        //Region name
      "compress" : "true",
      "access_key": "xxxxx",        //AK
      "secret_key": "xxxxxxxxxxxxxxxxx"    //SK
     "max_restore_bytes_per_sec": "100mb",           // OBS speed. The default value is 40 MB. You
can increase the value if your cluster can achieve higher performance.
      "max_snapshot_bytes_per_sec": "100mb"
    }
}
```

   b. Create a snapshot using a custom repository.

```
PUT _snapshot/my_backup/Snapshot_name
{
  "indices": "*", // Backup index. The asterisk (*) indicates indexes. Multiple indexes are
separated by commas (,).
  "ignore_unavailable": true, // Whether to ignore the availability of a single index. The value
true indicates that the availability is ignored.
"include_global_state": false //: The default value is false, indicating that the cluster state and
some other states are not saved.
}
```

   c. Query the snapshot status.

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

   d. Restore indexes in the custom repository.

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

4.6 How Can I Troubleshoot a Cluster With an Abnormally Heavy Load?

Symptom

A cluster's tasks have been rejected for a long time and a large number of tasks are suspended. The cluster's load value increases sharply.

Possible Causes

Possible causes are as follows:

- Query threads are executed slowly because a large amount of data is obtained.
- Threads are suspended caused by high read pressures.

Troubleshooting Procedure

Method 1: Using Cerebro

1. Log in to the CSS management console.
2. In the navigation pane, choose **Clusters** > **Elasticsearch**.
3. Locate the cluster whose load increases sharply and click **Access Cerebro** in the **Operation** column.
4. Check the CPU and heap metrics. If the values of these two metrics are too high, the cluster is overloaded. In this case, reduce the number of requests sent by the client and wait until the cluster load decreases.
5. Check the number and size of shards. Each shard is recommended to be 20 GB to 40 GB and not exceed 50 GB. On a single node, up to five shards can use the same index.

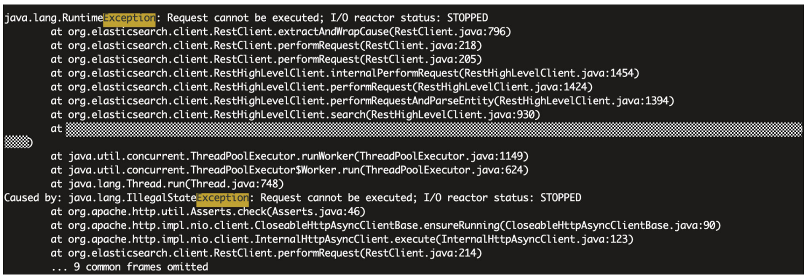
Method 2: Using Kibana

1. Log in to the CSS management console.
2. In the navigation pane, choose **Clusters** > **Elasticsearch**.
3. Locate the cluster whose load increases sharply and click **Access Kibana** in the **Operation** column. Click **Dev Tools**.
4. Run the **GET _cat/thread_pool?** command to view which threads are having tasks piling up and locate the cause of increased cluster workload.
5. Run the **GET /_nodes/hot_threads** command to view which threads occupy a large number of CPU resources and take a long time to execute, and locate the cause of task delaying.

4.7 Why "I/O Reactor STOPPED" Is Reported When I Use the Elasticsearch HLRC?

Symptom

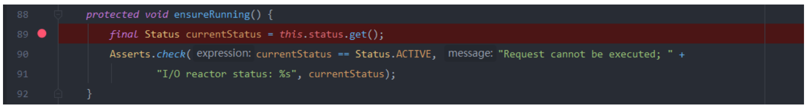
When I use the HLRC (High Level Rest Client) of Elasticsearch, "I/O Reactor STOPPED" is occasionally reported, but no errors are found in Elasticsearch logs.



```
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 [redacted])
	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
```

Checking Why the I/O Reactor Status Changed to STOPPED

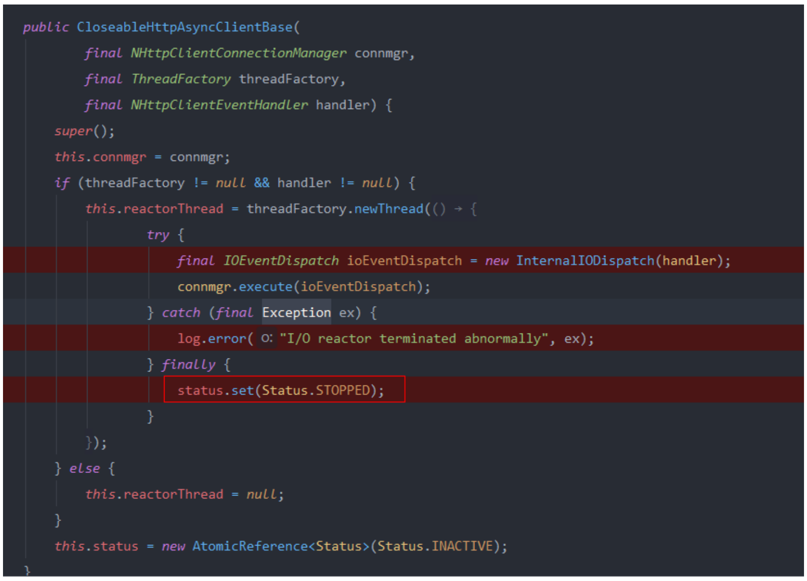
In the call stack, the error is found in line 90 in CloseableHttpAsyncClientBase, as shown in the following figure.



```
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      }
```

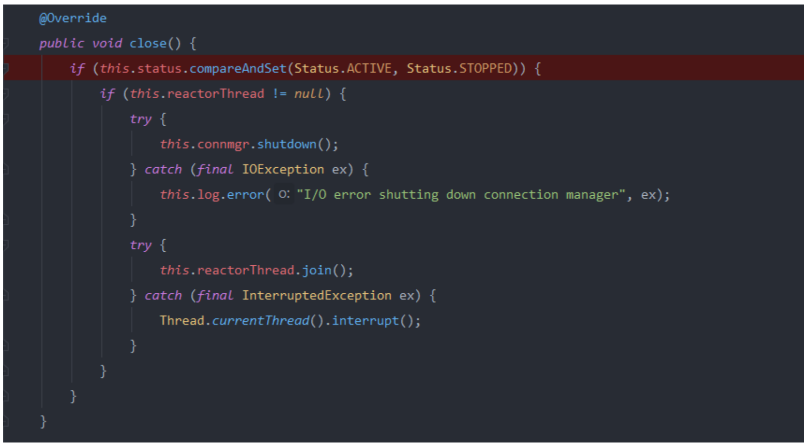
The ensureRunning() method is called at the beginning of request execution to check whether the client is active. If the client is inactive, an error will be reported. Check when the **status** in **CloseableHttpAsyncClientBase** changed to **STOPPED**. Two occurrences are found, as shown in the following figures.

Figure 4-2 First occurrence of **STOPPED**



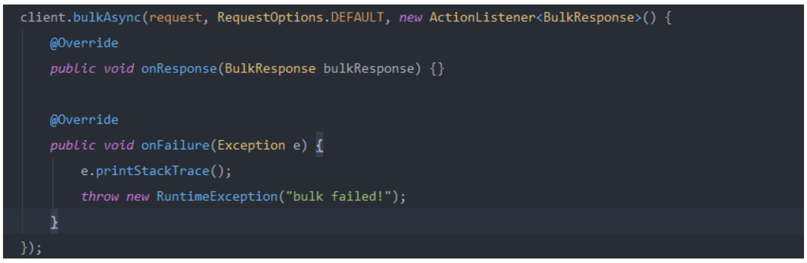
```
public CloseableHttpAsyncClientBase(
        final NHttpClientConnectionManager connmgr,
        final ThreadFactory threadFactory,
        final NHttpClientEventHandler handler) {
    super();
    this.connmgr = connmgr;
    if (threadFactory != null && handler != null) {
        this.reactorThread = threadFactory.newThread(() → {
                try {
                    final IOEventDispatch ioEventDispatch = new InternalIODispatch(handler);
                    connmgr.execute(ioEventDispatch);
                } catch (final Exception ex) {
                    log.error( o: "I/O reactor terminated abnormally", ex);
                } finally {
                    status.set(Status.STOPPED);
                }
        });
    } else {
        this.reactorThread = null;
    }
    this.status = new AtomicReference<Status>(Status.INACTIVE);
}
```

Figure 4-3 Second occurrence of **STOPPED**



```
@Override
public void close() {
    if (this.status.compareAndSet(Status.ACTIVE, Status.STOPPED)) {
        if (this.reactorThread != null) {
            try {
                this.connmgr.shutdown();
            } catch (final IOException ex) {
                this.log.error( o: "I/O error shutting down connection manager", ex);
            }
            try {
                this.reactorThread.join();
            } catch (final InterruptedException ex) {
                Thread.currentThread().interrupt();
            }
        }
    }
}
```

- It can be inferred that only the first occurrence made the status change to **STOPPED**, because generally a user would not close the client before calling an API.
- In the first occurrence, the **reactorThread** thread scheduled schedule I/O events. The status changed to **STOPPED** when an internal exception is thrown. To reproduce this issue, throw an exception and check whether the status will change, as shown in the following figure.



```
client.bulkAsync(request, RequestOptions.DEFAULT, new ActionListener<BulkResponse>() {
    @Override
    public void onResponse(BulkResponse bulkResponse) {}

    @Override
    public void onFailure(Exception e) {
        e.printStackTrace();
        throw new RuntimeException("bulk failed!");
    }
});
```

- When a request fails because of the exception, the status changes to **STOPPED**, the I/O Reactor is disabled, and the HLRC instance is suspended. Then, the HLRC instance fails at any attempt to call a request. Here the exception was manually created to reproduce the issue, but the cause of the I/O Reactor exception in the production environment is still unknown.

Possible Causes

The possible causes of the I/O Reactor exception are as follows:

1. Exception thrown during callback
2. High client concurrency

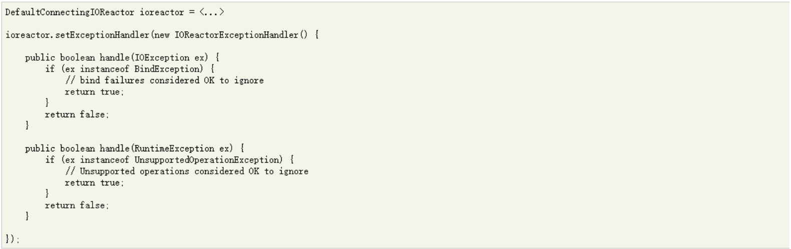
   After the exception was found in the customer's logs, we checked the Elasticsearch cluster monitoring metrics, including the CPU usage and number of network connections.

   The customer's cluster used five i3.4xlarge.8 nodes with 16 vCPUs and 128 GB memory. The customer performed a large number of bulk operations at around 05:00 every morning and wrote 100 GB to 200 GB data. Cluster monitoring metrics showed that the CPU usage and network inbound and outbound rates did not bring heavy workloads to nodes, but the number of network connections on each node was large. The network connections on certain nodes nearly reached 9,000, and even 50,000 on five nodes. The code used multiple threads of a single REST client to concurrently invoke the bulkAsync interface of the HLRC. The 40,000 to 50,000 Elasticsearch connections on a single client node could easily exhaust handles or connections of the node.

3. REST client of Elasticsearch. You are advised to add the exception handler to the REST client of Elasticsearch.

   Both HLRC and LLRC use Apache HTTPComponents Async Client. As mentioned in the documentation of Apache, some I/O exceptions during the interaction with sessions are predictable. Such exceptions may cause the termination of a single session, but do not affect I/O Reactor or other sessions. However, if an error occurs in I/O Reactor, such as the I/O exceptions in some classes of the underlying NIO or runtime exceptions that have not been not handled, I/O Reactor will shut down. Apache recommends that the IOReactorExceptionHandler interface be rewritten.

   Figure 4-4 Rewriting the IOReactorExceptionHandler API

```
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;
    }

});
```

   We rewrote the ExceptionHandler, placed it in the HLRC configurations, and threw an exception in the callback to simulate the problem. However, the exception in the callback was not captured by the I/O Reactor ExceptionHandler, and no exception was thrown during script execution. It was difficult to verify the function of the I/O Reactor ExceptionHandler. According to the verification results from other developers in the Elasticsearch community, no problems have occurred since they added the I/O Reactor ExceptionHandler a long time ago. You are advised to configure the I/O Reactor ExceptionHandler, but do not ignore all exceptions.

   The Elasticsearch REST client should provide an exception handler instead of asking users to configure the I/O Reactor ExceptionHandler, which cannot solve all exceptions.

Solution

1. Adjust the size of the client connection pool based on the site requirements.
2. You are advised to adjust the number of concurrent clients based on the site requirements or configure multiple nodes to share the pressure.
3. You are advised to configure the I/O Reactor ExceptionHandler to handle some exceptions.
4. After the HLRC is called, catch exceptions and check whether they occurred because the I/O Reactor status had changed to STOPPED. Restart the client to restore the connection.

4.8 The Peak Heap Memory of an Elasticsearch Cluster Remains High (Over 90%)

Symptom

The peak heap memory of an Elasticsearch cluster remains high (over 90%). If the heap memory usage of a node does not remain above 90%, the cluster is normal. If the heap memory usage remains above this rate for a long time, the cluster faces a certain risk of unavailability.

Possible Causes

- Check whether there are many tasks waiting in the write and query queues of the cluster.

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

- View the cluster monitoring information. Check the metrics related to the write and query tasks of the cluster.
- If the heap memory usage of the cluster remains high for a long time, check the cluster size and the number of nodes, and scale out the cluster if necessary.

Solution

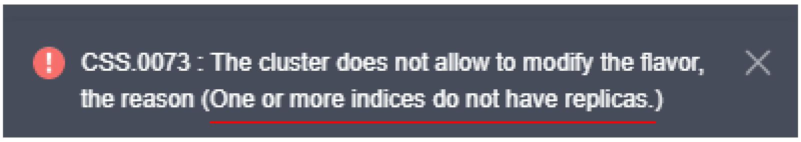
- Optimize the write and query programs on the client based on the task queuing statistics.
- If the cluster is heavily loaded for a long time, it may cause slow write, query, and frequent node disconnections. To avoid these problems, you can increase nodes or redesign the cluster.
- If the heap memory fluctuates around 95% and nodes are sometimes disconnected, use the traffic control function as needed. For more information, see **Configuring Flow Control for an Elasticsearch Cluster**.

4.9 Failed to Modify the Elasticsearch Cluster Specifications

Symptom

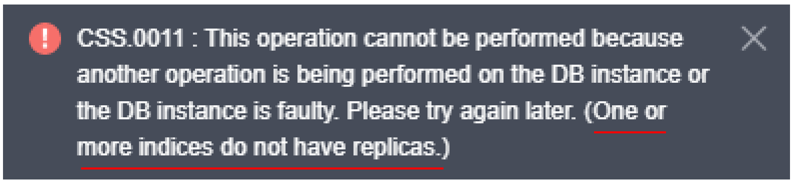
I failed to modify the cluster specifications and an error message is displayed on the console.

Figure 4-5 CSS.0073 error



CSS.0073 : The cluster does not allow to modify the flavor, the reason (One or more indices do not have replicas.)

Figure 4-6 CSS.0011 error



CSS.0011 : This operation cannot be performed because another operation is being performed on the DB instance or the DB instance is faulty. Please try again later. (One or more indices do not have replicas.)

Possible Causes

The number of replicas is not set for the current cluster. The background rejects the request for changing the specifications. You need to set the number of replicas before modifying the specifications. Otherwise, shards may be lost.

Solution

Run the following commands to set backup parameters:

```
PUT /index/_settings
{
   "number_of_replicas" : 1 //Number of replicas
}
```

4.10 An Error Message Is Displayed When I Change the Read-Only Status of an Index

Symptom

If the space of a security cluster is used up, all indexes become read-only. The value of the parameter **read_only_allow_delete** is **true**. Data can only be read from but cannot be written to the indexes. In this case, I run the following command to manually change the value of the **read_only_allow_delete** parameter to **false**:

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

The error information is as follows:

```
{
"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
}
```

Possible Causes

By default, a security cluster has an **. opendistro_security** index, which cannot be written. You need to skip this index when changing the status of indexes.

Solution

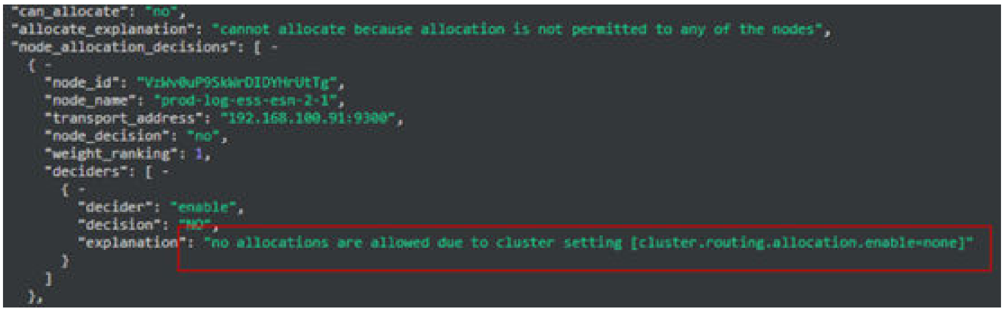
Use wildcards to match specified indexes. (Use a wildcard to replace the **indexname** in the following example.)

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

4.11 A Node in an Elasticsearch Cluster Has No Shards Allocated

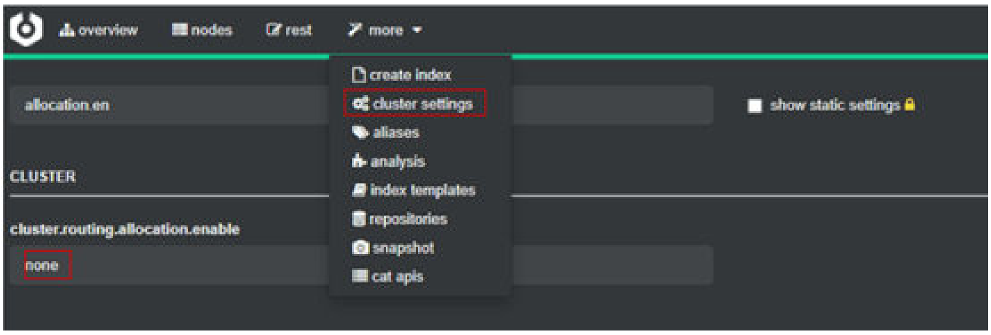
Solution

1. Run the **GET _cluster/allocation/explain?pretty** command to view the unallocated shards.



```
"can_allocate": "no",
"allocate_explanation": "cannot allocate because allocation is not permitted to any of the nodes",
"node_allocation_decisions": [ -
  { -
    "node_id": "VzWv0uP9SkWrDIDYHrUtTg",
    "node_name": "prod-log-ess-esn-2-1",
    "transport_address": "192.168.100.91:9300",
    "node_decision": "no",
    "weight_ranking": 1,
    "deciders": [ -
      { -
        "decider": "enable",
        "decision": "NO",
        "explanation": "no allocations are allowed due to cluster setting [cluster.routing.allocation.enable=none]"
      }
    ]
  },
```

2. On the console, choose **cerebro** > **more** > **cluster settings**, enter **allocation.enable** in the upper left corner, and change **none** to **all**.



allocation.en

CLUSTER

cluster.routing.allocation.enable

none

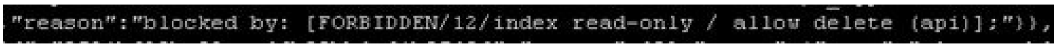
3. If no cerebro is displayed, run the following command:

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

4.12 Failed to Insert Data into a Cluster Index

Symptom

I failed to insert data into a CSS cluster index and the error information is as follows.



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

Issue Analysis

When the disk usage exceeds 95%, Elasticsearch automatically sets the index to the read-only status to prevent the disk space from being used up.

Solution

- For clusters in new versions (later than 7.10.2), the read-only mode is automatically disabled after the cluster disk usage decreases. You only need to free up space or expand the disk capacity.
- For clusters in old versions, you need to manually change the index status by running the following command on Kibana:

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

4.13 Error Message "maximum shards open" Is Displayed When Users Try to Create an Index

Symptom

When users try to create an index, an error message is displayed: "this action would add [2] total shards, but this cluster currently has [1000]/[1000] maximum shards open".

Possible Cause

By default, each node can support a maximum of 1000 shards. An error is reported when this limit is exceeded.

Solution

- Solution 1: Disable or delete unused indexes to reduce the number of shards.
- Solution 2: Change the limit on the maximum number of shards per node. For details, see **max_shards_per_node**.

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

NOTE

Changing the limit on the maximum number of shards per node is just a temporary fix. To solve this issue in the long term, aim for 20 shards or fewer per GB of JVM heap memory (each node should have a heap memory size that is 50% of the node's available memory, up to 31 GB). For more information, see **shard count recommendation**.

4.14 Error Message "403 Forbidden" Is Displayed When I Delete All Indexes

Symptom

When I run the **curl –i –u admin:password –XDELETE https://ip:9200/_all** command (**password** is the admin account password and **ip** is the private network address of the cluster) to delete all indexes, the error message "403 Forbidden" is reported.

Solution

The index **.opendistro_security** cannot be deleted from a security cluster. The command for deleting all indexes is invalid for clusters in the security mode. Use the index name or wildcard to delete specified indexes. Do not delete all indexes at a time.

4.15 Error Message "Forbidden" Is Displayed When I Delete an Index Pattern

Symptom

When I delete an index pattern on Kibana, the error message "Forbidden" is displayed.

Possible Causes

The previously created index pattern cannot be deleted because the Kibana indexes are read-only. If the disk usage exceeds a certain threshold, indexes will be automatically changed to read-only.

Solution

On the **Dev Tools** page of Kibana, run the following command to cancel the read-only state of the indexes:

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

4.16 Error Message "Trying to create too many scroll contexts" Is Displayed When the update-by-query Command Is Executed

Symptom

When I run the **update-by-query** command in an Elasticsearch cluster, the error message "Trying to create too many scroll contexts." is displayed. The specific error information is as follows:

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

Possible Causes

Each time a cluster calls the scroll creation API, a scroll context is created. When the number of scroll contexts reaches the preset threshold, no more scrolls can be created.

Procedure

Run the following command to check the value of **open_contexts** (the number of scroll contexts):

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

When the value reaches the upper limiter, use either of the following methods to solve the problem.

- Method 1: Delete unnecessary scrolls to release the storage space of scroll contexts.
  ```
  DELETE /_search/scroll
  {  "scroll_id" :
  "DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAAD4WYm9laVYtZndUQlNsdDcwakFMNjU1QQ=="}
  ```
- Method 2: Change the value of **search.max_open_scroll_context** to increase the storage space for scroll contexts.
  ```
  PUT /_cluster/settings
  {
      "persistent" : {
          "search.max_open_scroll_context": 2012345678
      },

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

4.17 Failed to Create a Pattern in an Elasticsearch Cluster

Symptom

When I create a pattern, the system does not respond and the pattern cannot be created.

Possible Cause

1. Check whether the disk is full and whether the Kibana index is read-only.
2. Check whether there are multiple Kibana indexes.

Solution

Run the following command to delete unnecessary indexes to release disk space:

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

5 Ports

5.1 Why Can't I Access Port 9200?

Symptom

If a VPN or VPC peering connection is used to access the CSS cluster, no result is returned when the curl command is used to connect to an Elasticsearch cluster.

For example, if you run the following command to connect to the cluster, no result is returned:

```
curl -s 'http://< node private access address >:9200'
```

Possible Causes

If a VPN or VPC peering connection is used to access CSS, that means that the client and CSS are not in the same VPC. Therefore, the subnet of the CSS cluster must be in a different network segment from that of the VPC.

Suppose, for example, there is a CSS cluster in VPC **vpc-8e28** on the network segment **192.168.0.0/16**, the subnet **subnet-4a81** of the VPC is selected, and its network segment is also **192.168.0.0/16**. As the CSS subnet **vpc-8e28** and the subnet it is being accessed from (**subnet-4a81**) are both 192.168.0.0/16, if the VPN or the VPC peering connection tries to access the CSS cluster, the host created on the subnet does not have a gateway corresponding to the VPC. As a result, the default route of the CSS service is affected and access to port 9200 fails.

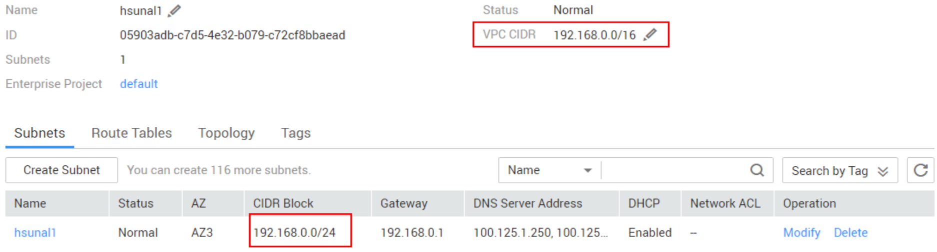
Procedure

If access to port 9200 fails but the CSS cluster is available, do as follows:

1. Go to the CSS management console. In the cluster list, click the cluster name to view the VPC and subnet used by the cluster.
2. Go to the VPC management console. In the VPC list, click the name of the VPC used by the CSS cluster. The VPC details page is displayed. View the VPC and subnet network segment information.

As shown in **Figure 5-1**, the VPC network segment information is the same as the subnet network segment information. When a VPN private line or a VPC peer connection is used, access to port 9200 fails.

Figure 5-1 Viewing network segment information



Name	hsunal1	Status	Normal
ID	05903adb-c7d5-4e32-b079-c72cf8bbaead	VPC CIDR	192.168.0.0/16
Subnets	1		
Enterprise Project	default		

Name	Status	AZ	CIDR Block	Gateway	DNS Server Address	DHCP	Network ACL	Operation
hsunal1	Normal	AZ3	192.168.0.0/24	192.168.0.1	100.125.1.250, 100.125...	Enabled	--	Modify Delete

3. If the preceding error occurs, create another cluster and this time select a subnet that is different from the VPC subnet. If the subnet does not exist, create another subnet on the VPC management console.

   After a new CSS cluster is created, migrate the data of the old cluster to the new cluster, and then use the VPN or VPC peering connection to access the cluster.

   NOTE

   If you require a VPN connection or VPC peering connection to access the CSS cluster, ensure that the VPC and subnet of the newly created CSS are in different network segments.