Cloud Search Service

FAQs

Issue 01

Date 2025-09-04

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 General Consulting

1.1 How Does CSS Ensure Data and Workload Security?

CSS uses network isolation in addition to various host and data security measures.

- Network isolation

  The entire network is divided into two planes: service plane and management plane. The two planes are deployed and isolated physically to ensure the security of the service and management networks.

  – Service plane: This is the network plane of the cluster. It provides service channels for users and delivers data definitions, indexing, and search capabilities.

  – Management plane: This is the management console, where you manage CSS.

- Host security

  CSS provides the following security measures:

  – The VPC security group ensures the security of the hosts in a VPC.

  – Network access control lists (ACLs) allow you to control what data can enter or exit your network.

  – The internal security infrastructure (including the network firewall, intrusion detection system, and protection system) monitors all network traffic that enters or exits the VPC through an IPsec VPN.

- Data security

  CSS uses multiple replicas, cross-AZ deployment of clusters, and third-party (OBS) backup of index data to ensure the security of user data.

1.2 What Storage Options Are Available for a CSS Cluster?

CSS uses EVS and local disks to store your indexes. During cluster creation, you can specify the EVS disk type and specifications (the EVS disk size).

- EVS disk types include common I/O, high I/O, ultra-high I/O, and extreme SSD.
- The EVS disk size varies depending on the node specifications you selected during cluster creation.

During cluster creation, a certain number of EVS disks can be attached to each node (corresponding to an ECS). You can calculate the total storage capacity of a CSS cluster based on the sizes of EVS disks attached to different ECSs. The EVS disk size is determined by the node specifications you selected when creating the cluster.

1.3 What Files Are Stored on the Disks of a CSS Cluster?

You can store the following logs and files:

- Log files: cluster logs
- Data files: cluster index files
- Other files: cluster configuration files
- OS: 5% of the storage space is reserved for the OS by default.

1.4 What Data Compression Algorithms Does CSS Use?

CSS supports two data compression algorithms: LZ4 (by default) and best_compression.

- **LZ4 algorithm**

  LZ4 is the default compression algorithm for Elasticsearch. This algorithm can compress and decompress data quickly, but its compression ratio is low.

  LZ4 scans data with a 4-byte window, which slides 1 byte forward at a time. Duplicate data is compressed. This algorithm applies to scenarios where a large amount of data to be read while a small amount of data to be written.

- **best_compression algorithm**

  This algorithm can be used when a large amount of data is written and the index storage cost is high, such as logs and time sequence analysis. This algorithm can greatly reduce the index storage cost.

Run the following command to switch the default compression algorithm (LZ4) to best_compression:

```
PUT index-1
{
    "settings": {
        "index": {
            "codec": "best_compression"
        }
    }
}
```

The LZ4 algorithm can quickly compress and decompress data while the best_compression algorithm has a higher compression and decompression ratio.

1.5 Differences Between Elasticsearch and OpenSearch

CSS provides a fully managed cloud search service based on open-source engines. CSS Elasticsearch and OpenSearch both have the following core capabilities:

- Unified architecture: a distributed, RESTful search engine, supporting near-real-time search and analytics over petabytes of data
- A wide range of use cases: log analytics, enterprise search, big data analytics, vector search, semantic search, RAG, etc.
- Enhanced features: deep optimization based on open-source versions, high performance, high availability, cost effective, and fully managed

A Comparison of Core Functions

Table 1-1 A comparison of core functions between Elasticsearch and OpenSearch

Dimension	CSS Elasticsearch	CSS OpenSearch
Origin	Built on Apache Lucene, Elasticsearch is a mature, widely adopted search engine.	OpenSearch, a fork of Elasticsearch, inherits its core search and analytics capabilities while keeps evolving.
Compatibility	● Compatible with the Elasticsearch ecosystem ● Compatible with later-version Elasticsearch SDKs	● Compatible with the OpenSearch ecosystem ● Compatible with Elasticsearch 7.10.2
Version policy	The mainstream version is 7.10.2, which will be continuously optimized. We recommend upgrading all Elasticsearch clusters to this version.	The version will be continuously updated to keep up with open-source innovations.
Kernel features	CSS will provide continuous kernel enhancement powered by in-house R&D.	CSS will integrate its own capabilities with open-source innovations to ensure continuous kernel enhancement.
Evolution	Emphasizes the stability of the 7.x version and in-house enhancements.	Actively integrates new cloud native features.

Engine Selection Suggestions

Table 1-2 When to choose Elasticsearch or OpenSearch

Scenario	Recommended Engine	Reason
Running Elasticsearch 7.10.2 or earlier for long-term stability	**Elasticsearch**	● Mature and stable, fully compatible with the native Elasticsearch toolchain ● Compatible with later-version Elasticsearch SDKs ● Backed by the unique strengths of CSS (such as vector search)
New features of Elasticsearch 8+ are required	**OpenSearch**	● Inherits Elasticsearch capabilities while keeps evolving ● Backed by the unique strengths of CSS (such as vector search)
Smooth migration of existing Elasticsearch 7.x clusters	Elasticsearch or OpenSearch	Both are compatible with Elasticsearch 7.10.2 APIs, with a similar migration cost.

2 Accessing CSS Clusters

2.1 How Do I Reset the Administrator Password of a Security-mode Cluster in CSS?

If you want to change the administrator password of a security-mode cluster, or if you have forgotten the password, reset the password.

1. Log in to the **CSS management console**.
2. In the navigation pane, choose **Clusters** > **Elasticsearch** or **Clusters** > **OpenSearch**.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab.
5. In the **Configuration** area, click **Reset** next to **Reset Password**. Set and confirm the new administrator password.
   - The password can contain 8 to 32 characters.
   - The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. The following special characters are allowed: **~!@#$%^&*()-_=+\|[{}];:,<.>/?**
   - Do not use the administrator name, or the administrator name spelled backwards, as the password.
   - You are advised to change the password periodically.

Figure 2-1 Reset Password

Security Mode Enabled

Reset Password Reset

2.2 Are Ports 9200 and 9300 Open for Access to Elasticsearch Clusters?

Yes. Port 9200 is used by external systems to access CSS clusters, and port 9300 is used for communication between nodes.

The methods for accessing port 9300 are as follows:

- If your client is in the same VPC and subnet with the CSS cluster, you can access it directly.
- If your client is in the same VPC with but different subnet from the CSS cluster, apply for a route separately.
- If your client is in the different VPCs and subnets from the CSS cluster, create a VPC peering connection to enable communication between the two VPCs, and then apply for routes to connect the two subnets.

2.3 How Do I Use a NAT Gateway to Enable Public Network Access for an Elasticsearch/OpenSearch Cluster?

Perform the following operations:

1. **Obtaining Cluster Information**
2. **Configuring a NAT Gateway**
3. **Modifying Security Group Rules for the Cluster**
4. **Accessing a Cluster over the Public Network**

⚠ CAUTION

If your CSS clusters do not have the security mode enabled, do not allow public network access to them via the NAT gateway. Otherwise, your data will be exposed to the Internet.

Obtaining Cluster Information

Step 1 Log in to the **CSS management console**.

Step 2 In the navigation pane, choose **Clusters** > **Elasticsearch** or **Clusters** > **OpenSearch**.

Step 3 In the cluster list, click the name of the target cluster. The cluster information page is displayed.

Step 4 Click the **Overview** tab.

Step 5 In the **Configuration** area, obtain the cluster's **Region**, **VPC**, **Current Subnet**, and **Private IPv4 Address**.

----End

Configuring a NAT Gateway

Step 1 Create a public NAT gateway to enable public network access for the current cluster.

For details, see **Buying a Public NAT Gateway**. **Table 2-1** describes the key parameters. Set other parameters based on service requirements.

Table 2-1 Configuring a public NAT gateway

Parameter	Description
Region	Use the region of the Elasticsearch/OpenSearch cluster.
VPC	Use the VPC of the Elasticsearch/OpenSearch cluster.
Subnet	Use the subnet of the Elasticsearch/OpenSearch cluster.

Step 2 After a public NAT gateway is created, add DNAT rules to allow the cluster in your VPC to provide services accessible from the Internet.

For details, see **Adding a DNAT Rule**. **Table 2-2** describes the key parameters. Set other parameters based on service requirements.

Table 2-2 Adding a DNAT rule

Parameter	Description
Public IP Address Type	Select **EIP**. Remember the configured IP address, which will be needed for accessing the cluster from the public network.
Public Port	A custom port can be configured. Remember the configured port, which will be needed for accessing the cluster from the public network.
Private IP Address	Enter the cluster's private IPv4 address obtained **Obtaining Cluster Information**.
Private Port	Enter 9200.

⚠ CAUTION

If the cluster has multiple private IPv4 addresses, add multiple DNAT rules.

----End

Modifying Security Group Rules for the Cluster

Step 1 Log in to the **CSS management console**.

Step 2 In the navigation pane, choose **Clusters** > **Elasticsearch** or **Clusters** > **OpenSearch**.

Step 3 In the cluster list, click the name of the target cluster. The cluster information page is displayed.

Step 4 Click the **Overview** tab.

Step 5 In the **Configuration** area, find **Security Group**, and click the security group name to go to the details page.

Step 6 Click the **Inbound Rules** tab.

Step 7 Click **Add Rule** to add an inbound rule to allow port 9200.

Step 8 Click **OK**.

----End

Accessing a Cluster over the Public Network

Enter **https://{IP}:{port}** or **http://{IP}:{port}** in the browser address box to access the Elasticsearch or OpenSearch cluster.

- ***IP*** and ***port*** are the EIP and port you set when you added DNAT rules.
- If you have enabled **Security Mode** for the cluster, enter **https://{IP}:{port}** and then enter the username and password for the cluster.
- If you have not enabled **Security Mode** for the cluster, enter **http://{IP}:{port}**.

2.4 How Do I Connect In-house Developed Kibana to an Elasticsearch Cluster in CSS?

Constraints

Only Kibana images of the OSS version can be connected to Elasticsearch clusters in CSS.

Procedure

1. Create an ECS.
   - The ECS must be within the same VPC as the CSS cluster.
   - Port 5601 must be allowed by the security group associated with the ECS.
   - An EIP must be allocated to the ECS.

   For details, see **Purchasing an ECS**.
2. Obtain the address for accessing the Elasticsearch cluster of CSS.
   a. Log in to the **CSS management console**.

b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.

c. In the cluster list, obtain the target cluster's private IP address from the **Private IP Address** column. Generally, the IP address format is *<host>*:*<port>* or *<host>*:*<port>*,*<host>*:*<port>*.

   If the cluster has only one node, the IP address and port number of this one node are displayed, for example, **10.62.179.32:9200**. If the cluster has multiple nodes, the IP addresses and port numbers of all nodes are displayed, for example, **10.62.179.32:9200,10.62.179.33:9200**.

3. Install Kibana on the ECS and modify the configuration file.

   – The following is an example of the configuration file for a security-mode cluster:

```
elasticsearch.username: "***"   //Username of the security cluster
elasticsearch.password: "***"   //Password of the security cluster
elasticsearch.ssl.verificationMode: none
server.ssl.enabled: false
server.rewriteBasePath: false
server.port: 5601
logging.dest: /home/Ruby/log/kibana.log
pid.file: /home/Ruby/run/kibana.pid
server.host: 192.168.xxx.xxx   //IP address or DNS name of the Kibana server. localhost is recommended.
elasticsearch.hosts: http://10.0.0.xxx:9200   //Address for accessing the Elasticsearch cluster
elasticsearch.requestHeadersWhitelist: ["securitytenant","Authorization"]
opendistro_security.multitenancy.enabled: true
opendistro_security.multitenancy.tenants.enable_global: true
opendistro_security.multitenancy.tenants.enable_private: true
opendistro_security.multitenancy.tenants.preferred: ["Private", "Global"]
opendistro_security.multitenancy.enable_filter: false
```

   ⚠ CAUTION

   To access a security-mode cluster, the opendistro_security_kibana plug-in must be installed. For details, see **security-kibana-plugin**. The plug-in version must be the same as that of the cluster. To check the plug-in version, run the **GET _cat/plugins** command.

   – The following is an example of the configuration file for a non-security mode cluster:

```
server.port: 5601
logging.dest: /home/Ruby/log/kibana.log
pid.file: /home/Ruby/run/kibana.pid
server.host: 192.168.xxx.xxx   //IP address or DNS name of the Kibana server. localhost is recommended.
elasticsearch.hosts: http://10.0.0.xxx:9200   //Address for accessing the Elasticsearch cluster
```

4. Use a browser on your local PC to connect to the EIP associated with the ECS. The URL is **http://EIP:5601**. Log in to Kibana to access the Elasticsearch cluster.

2.5 How Do I Connect In-house Developed OpenSearch Dashboards to an OpenSearch Cluster in CSS?

Constraints

Only OpenSearch Dashboards images of the OSS version can be connected to OpenSearch clusters in CSS.

Procedure

1. Create an ECS.
   - The ECS must be within the same VPC as the CSS cluster.
   - Port 5601 must be allowed by the security group associated with the ECS.
   - An EIP must be allocated to the ECS.

   For details, see **Purchasing an ECS**.

2. Obtain the address for accessing the OpenSearch cluster of CSS.

   a. Log in to the **CSS management console**.

   b. In the navigation pane on the left, choose **Clusters > OpenSearch**.

   c. In the cluster list, obtain the target cluster's private IP address from the **Private IP Address** column. Generally, the IP address format is *<host>*:*<port>* or *<host>*:*<port>*,*<host>*:*<port>*.

      If the cluster has only one node, the IP address and port number of this one node are displayed, for example, **10.62.179.32:9200**. If the cluster has multiple nodes, the IP addresses and port numbers of all nodes are displayed, for example, **10.62.179.32:9200,10.62.179.33:9200**.

3. Install OpenSearch Dashboards on the ECS and modify the configuration file.
   - The following is an example of the configuration file for a security-mode cluster:

```
opensearch.username: "***" //Username of the security cluster
opensearch.password: "***" //Password of the security cluster
opensearch.ssl.verificationMode: none
server.ssl.enabled: false
server.rewriteBasePath: false
server.port: 5601
logging.dest: /home/Ruby/log/kibana.log
pid.file: /home/Ruby/run/kibana.pid
server.host: 192.168.xxx.xxx  //IP address or DNS name of the OpenSearch Dashboards server.
localhost is recommended.
opensearch.hosts: http://10.0.0.xxx:9200   //Address for accessing the OpenSearch cluster
opensearch.requestHeadersWhitelist: ["securitytenant","Authorization"]
opensearch_security.multitenancy.enabled: true
opensearch_security.multitenancy.tenants.enable_global: true
opensearch_security.multitenancy.tenants.enable_private: true
opensearch_security.multitenancy.tenants.preferred: ["Private", "Global"]
opensearch_security.multitenancy.enable_filter: false
```

     NOTE

     To access a security-mode cluster, the opendistro_security_kibana plug-in must be installed. For details, see **security-kibana-plugin**. The plug-in version must be the same as that of the cluster. To check the plug-in version, run the **GET _cat/plugins** command.

   - The following is an example of the configuration file for a non-security mode cluster:

```
server.port: 5601
logging.dest: /home/Ruby/log/opensearch-dashboards.log
pid.file: /home/Ruby/run/opensearch-dashboards.pid
server.host: 192.168.xxx.xxx  //IP address or DNS name of the OpenSearch Dashboards server.
localhost is recommended.
opensearch.hosts: http://10.0.0.xxx:9200   //Address for accessing the OpenSearch cluster
```

4. Use a browser on your local PC to connect to the EIP associated with the ECS. The URL is **http://EIP:5601**. Log in to OpenSearch Dashboards to access the OpenSearch cluster.

3 Migrating CSS Clusters

3.1 Can CSS Elasticsearch Clusters Be Migrated Across VPCs?

Use either of the following methods to migrate Elasticsearch clusters across different VPCs:

Method 1

Use the backup and restoration function to migrate cluster data. For details, see **Index Backup and Restoration**.

Method 2

1. Connect the VPC network and establish a VPC peering connection. For details, see **VPC Peering Connection Overview**.
2. After the network is connected, use Logstash to migrate data.

3.2 Can CSS Clusters Be Migrated Across Different Regions?

In CSS, perform the following to migrate Elasticsearch clusters across different regions:

- If the OBS bucket is in the same region as your CSS cluster, migrate the cluster by following the instructions in **Index Backup and Restoration**.
- If the OBS bucket is not in the same region as your CSS cluster, **configure cross-region replication** to back up the cluster to the bucket, and migrate the cluster by following the instructions in **Index Backup and Restoration**.

NOTE

- Before cross-region replication, ensure the snapshot folder of the destination cluster is empty. Otherwise, the snapshot information cannot be updated to the snapshot list of the destination cluster.
- Before every migration, ensure the folder is empty.

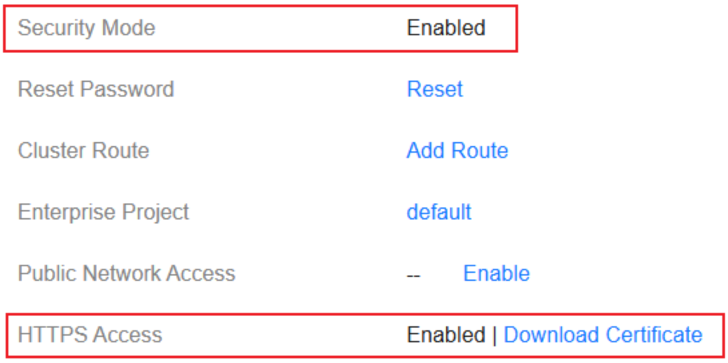
3.3 Examples of Logstash Configuration Files for Migrating Elasticsearch Clusters Using CSS Logstash

In our example, both the source and destination ends are Elasticsearch clusters of the same type (such as security mode and the web protocol used) in CSS. If the source and destination Elasticsearch clusters are of different types, modify the input and output modules provided in our example to obtain the configuration file you need.

Checking the Cluster Type

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. In the cluster list, find the source or destination Elasticsearch cluster, and click the cluster name to go to the cluster information page.
4. Click the **Overview** tab. In the **Configuration** area, check whether **Security Mode** and **HTTPS Access** are enabled. **Figure 3-1** shows an Elasticsearch cluster with the security mode and HTTPS both enabled.

Figure 3-1 Checking cluster settings



Security Mode Enabled
Reset Password Reset
Cluster Route Add Route
Enterprise Project default
Public Network Access -- Enable
HTTPS Access Enabled | Download Certificate

Table 3-1 Examples of Logstash configuration files for different types of clusters

Scenario	Example Logstash Configuration File
Migrating data between non-security mode clusters	**Example of a Logstash Configuration File for Non-Security Mode Clusters**

Scenario	Example Logstash Configuration File
Migrating data between security-mode clusters that use HTTP	**Example of a Logstash Configuration File for Security-Mode Clusters That Use HTTP**
Migrating data between security-mode clusters that use HTTPS	**Example of a Logstash Configuration File for Security-Mode Clusters That Use HTTPS**

Example of a Logstash Configuration File for Non-Security Mode Clusters

The following is an example of a Logstash configuration file when security mode is disabled for both the source and destination Elasticsearch clusters.

```
input {
    elasticsearch {
        # Address of the source Elasticsearch cluster
        hosts => ["xx.xx.xx.xx:9200", "xx.xx.xx.xx:9200"]
        # List of indexes to be migrated, separated by commas (,).
        index => "xxx,xxx,xxx"
        # Retain the default values.
        docinfo => true
    }
}

filter {
    # Delete fields added by Logstash.
    mutate {
        remove_field => ["@timestamp", "@version"]
    }
}

output {
    elasticsearch {
        # Address of the destination Elasticsearch cluster
        hosts => ["xx.xx.xx.xx:9200", "xx.xx.xx.xx:9200"]
        # Index name of the destination cluster. The following configurations must be the same as that of the source cluster.
        index => "%{[@metadata][_index]}"
        # ID of the destination data. If you do not need to retain the original ID, delete the following line for better performance.
        document_id => "%{[@metadata][_id]}"
        # Retain the default values.
        manage_template => false
        ilm_enabled => false
    }
}
```

Example of a Logstash Configuration File for Security-Mode Clusters That Use HTTP

The following is an example of a Logstash configuration file when security mode is enabled for both the source and destination Elasticsearch clusters but HTTPS is disabled for them.

```
input {
    elasticsearch {
        # Username at the source end
        user => "xxx"
        # Password at the source end
        password => "xxx"
        # Address of the source Elasticsearch cluster
        hosts => ["xx.xx.xx.xx:9200", "xx.xx.xx.xx:9200"]
        # List of indexes to be migrated, separated by commas (,).
        index => "xxx,xxx,xxx"
        # Retain the default values.
        docinfo => true
    }
}

filter {
    # Delete fields added by Logstash.
    mutate {
        remove_field => ["@timestamp", "@version"]
    }
}

output {
    elasticsearch {
        # Username at the destination end
        user => "xxx"
        # Password at the destination end
        password => "xxx"
        # Address of the destination Elasticsearch cluster
        hosts => ["xx.xx.xx.xx:9200", "xx.xx.xx.xx:9200"]
        # Index name of the destination cluster. The following configurations must be the same as that of the 
source cluster.
        index => "%{[@metadata][_index]}"
        # ID of the destination data. If you do not need to retain the original ID, delete the following line for 
better performance.
        document_id => "%{[@metadata][_id]}"
        # Retain the default values.
        manage_template => false
        ilm_enabled => false
    }
}
```

Example of a Logstash Configuration File for Security-Mode Clusters That Use HTTPS

The following is an example of a Logstash configuration file when security mode and HTTPS are enabled for both the source and destination Elasticsearch clusters.

```
input {
    elasticsearch {
        # Username at the source end
        user => "xxx"
        # Password at the source end
        password => "xxx"
        # Address of the source Elasticsearch cluster
        hosts => ["xx.xx.xx.xx:9200", "xx.xx.xx.xx:9200"]
        # List of indexes to be migrated, separated by commas (,).
        index => "xxx,xxx,xxx"
        # Certificate of the source Elasticsearch cluster. For clusters on the cloud, retain the following 
information. For user-built Logstash clusters, download the certificate from the cluster details page. Enter 
the certificate path plus certificate name here.
        ca_file => "/rds/datastore/logstash/v7.10.0/package/logstash-7.10.0/extend/certs"
        # Retain the default values.
        docinfo => true
        ssl => true
    }
}

filter {
    # Delete fields added by Logstash.
    mutate {
        remove_field => ["@timestamp", "@version"]
```

```
  }
}

output {
    elasticsearch {
        # Username at the destination end
        user => "xxx"
        # Password at the destination end
        password => "xxx"
        # Address of the destination Elasticsearch cluster
        hosts => ["xx.xx.xx.xx:9200", "xx.xx.xx.xx:9200"]
        # Index name of the destination cluster. The following configurations must be the same as that of the source cluster.
        index => "%{[@metadata][_index]}"
        # ID of the destination data. If you do not need to retain the original ID, delete the following line for better performance.
        document_id => "%{[@metadata][_id]}"
        # Certificate of the destination Elasticsearch cluster. For clusters on the cloud, retain the following information. For user-built Logstash clusters, download the certificate to the node from the cluster details page. Enter the certificate path plus certificate name here.
        cacert => "/rds/datastore/logstash/v7.10.0/package/logstash-7.10.0/extend/certs"
        # Retain the default values.
        manage_template => false
        ilm_enabled => false
        ssl => true
        ssl_certificate_verification => false
    }
}
```

3.4 Can I Export Data from Kibana in CSS?

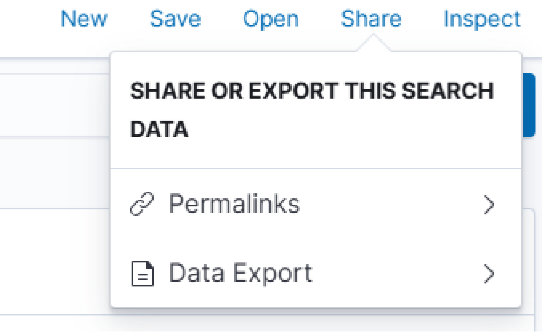
- With Elasticsearch 7.10.2 (with an image version no earlier than 7.10.2_24.3.3_x.x.x), Kibana supports one-click data export in CSV.

NOTICE

- A maximum of 10 MB of data can be exported. If the data you want to export exceeds 10 MB, only the first 10 MB is exported.
- Any special characters such as **=+-@** in exported CSV files may be identified as part of some formulas, leading to data export failures.

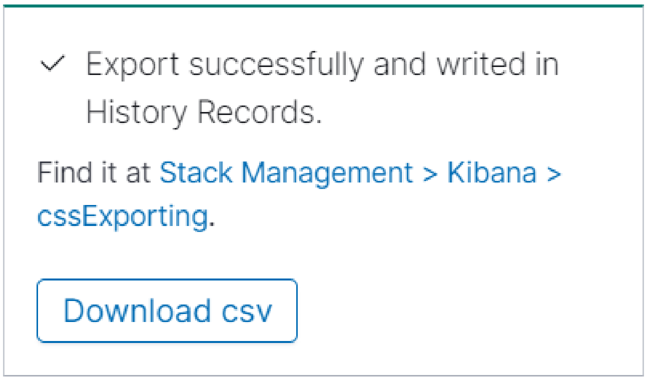
On the **Discover** page of Kibana, choose **Share > Data Export** in the upper right corner, and select **Export CSV**.

Figure 3-2 Exporting data



Wait for a few minutes. Then click **Download csv** in the lower right corner to download the data to the local PC.

Figure 3-3 Downloading data



✓ Export successfully and writed in History Records.

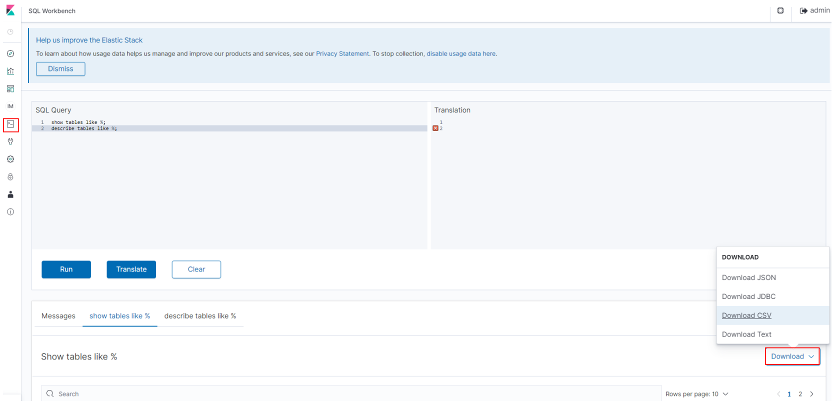
Find it at Stack Management > Kibana > cssExporting.

- With Elasticsearch 7.6.2, 7.9.3, and 7.10.2 (the image version is earlier than 24.3.0), the SQL Workbench plug-in is required for exporting data on Kibana.

  In **SQL Workbench** of Kibana, you can run Elasticsearch SQL statements to query data or click **Download** to export data. You can export 1 to 200 records. By default, 200 records are exported.

  For details about the Elasticsearch SQL statements used, see **Elasticsearch SQL**.

Figure 3-4 SQL Workbench



Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

```
show tables like %;
describe tables like %;
```

4 Using CSS Cluster Search Engines

4.1 Why Are Newly Created Index Shards Allocated to a Single Node in CSS?

Possible Causes

The possible causes are as follows:

- Shards were unevenly distributed in previous index allocations, and the predominate parameter in the latest indexed shard allocation was **balance.shard**. To balance the shard distribution across nodes, the new shards were allocated to the node with only a small number of shards.
- After a new node was added to a cluster and before the automatic cluster rebalancing completes, the predominate parameter was **balance.shard**. The shards of a new index are allocated to the new node, where there are no shards yet.

The following two parameters are used to balance the shard allocation in a cluster:

cluster.routing.allocation.balance.index (default value: **0.45f**)

cluster.routing.allocation.balance.shard (default value: **0.55f**)

NOTE

- **balance.index**: A larger value indicates that all the shards of an index are more evenly distributed across nodes. For example, if an index has six shards and there are three data nodes, two shards will be distributed on each node.
- **balance.shard**: A larger value indicates that all the shards of all the indexes are more evenly distributed across nodes. For example, if index **a** has two shards, index **b** has four, and there are three data nodes, two shards will be distributed on each node.
- You can specify both **balance.index** and **balance.shard** to balance the shard allocation.

Solution

To prevent the all the shards of an index from being allocated to a single node, use either of the following methods:

1. To create an index during cluster scale-out, configure the following parameter:

   ```
   PUT INDEX_NAME/_settings
   {
     "index.routing.allocation.total_shards_per_node": 2
   }
   ```

   That is, allow no more than two shards of an index to be allocated on each node. Determine the maximum number of shards allocated to each node based on the number of data nodes in your cluster and the number of index shards (both primary and secondary).

2. If too many shards are distributed on only a few nodes, you can move some of the shards to other nodes to balance the distribution. Run the **move** command of **POST _cluster/reroute**. The rebalance module will automatically exchange the shard with a shard on the destination node. Determine the values of **balance.index** and **balance.shard** as needed.

4.2 How Do I Create a Type Under an Index in an Elasticsearch 7.*x* Cluster of CSS?

In Elasticsearch 7.*x* and later versions, types cannot be created for indexes.

If you need to use types, add **include_type_name=true** to the command. Only a single type is supported.

```
PUT index?include_type_name=true
{
  "mappings": {
    "my_type": {
      "properties": {
        "@timestamp": {
          "type": "date"
        }
      }
    }
  }
}
```

After a multi-type index is created, run the following command to write data into it:

```
PUT index/my_type/1
{
  "@timestamp":"2019-02-20"
}
```

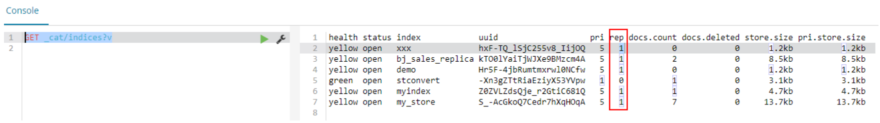
4.3 How Do I Change the Number of Replicas for Elasticsearch Indexes in CSS?

When creating an index for an Elasticsearch cluster, you can specify the number of shards, that is, the number of primary shards. Once an index is created, the number of primary shards cannot be changed, but the number of replicas can be changed. **Number of replica shards = Number of primary shards x Number of replicas.**

1. Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
   c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.
   d. In the left navigation pane, choose **Dev Tools**.
2. On the Kibana console, run the following command to check the number of replicas for each Elasticsearch index:

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

Figure 4-1 Checking the number of replicas



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

```
health status index            uuid                   pri rep docs.count docs.deleted store.size pri.store.size
yellow open   xxx              hxF-TQ_lSjC255v8_IijOQ   5   1          0            0      1.2kb          1.2kb
yellow open   bj_sales_replica kTO0lYaiTjWJXe9BMzcm4A   5   1          2            0      8.5kb          8.5kb
yellow open   demo             Hr5F-4jbRumtmxrwl0NCfw   5   1          0            0      1.2kb          1.2kb
green  open   stconvert        -Xn3gZTtRiaEziyX53YVpw   1   0          1            0      3.1kb          3.1kb
yellow open   myindex          Z0ZVLZdsQje_r2GtiC681Q   5   1          1            0      4.7kb          4.7kb
yellow open   my_store         S_-AcGkoQ7Cedr7hXqHOqA   5   1          7            0     13.7kb         13.7kb
```

3. Run the following command to configure the number of index replicas:

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

indexname indicates the name of the index to be modified, and **number_of_replicas** indicates the number of replicas to be set.

4.4 What Are the Impacts If an Elasticsearch Cluster of CSS Has Too Many Shards?

1. A large number of shards in a cluster slows down shard creation.
2. If automatic index creation is enabled, slow index creation may cause a large number of write requests to be stacked in the memory or result in a cluster breakdown.
3. If there are too many shards and you cannot properly monitor workloads, the number of records in a single shard may exceed the threshold, and write requests may be denied.

4.5 How Do I Check the Number of Index Shards and Replicas in a CSS Elasticsearch Cluster?

1. Log in to Kibana and go to the command execution page. Elasticsearch clusters support multiple access methods. This topic uses Kibana as an example to describe the operation procedures.
   a. Log in to the **CSS management console**.
   b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
   c. In the cluster list, find the target cluster, and click **Kibana** in the **Operation** column to log in to the Kibana console.

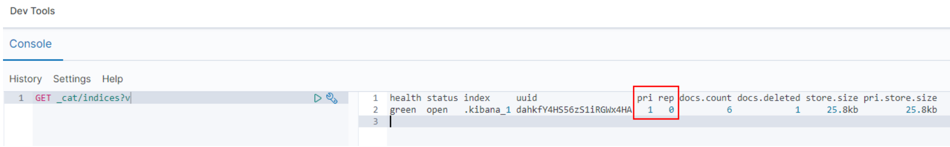
d. In the left navigation pane, choose **Dev Tools**.

2. Run the following command on Kibana to query the number of shards and replicas of each index:

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

In **Figure 4-2**, the **pri** column indicates the number of index shards, and the **rep** column indicates the number of replicas. Once an index is created, its **pri** value cannot be changed, but its **rep** value can be changed.

Figure 4-2 Querying index information



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

```
health status index     uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   .kibana_1 dahkfY4HS56zS1iRGWx4HA   1   0          6            1     25.8kb         25.8kb
```

4.6 What Does the Value i for node.roles Mean for Nodes in an Elasticsearch Cluster of CSS?

Function

If the value of **node.roles** of a client node is **i**, then is this client node an ingest node?

- Are there coordinating only nodes in clusters? Are the client requests distributed to coordinating nodes?
- Are ingest nodes in idle state when there are no ingest requests?

Solution

If the value of **node.roles** of a client node is **i**, the ingest node mode is enabled.

- The coordinating only nodes of Elasticsearch are called client nodes in CSS. If a cluster has no client nodes, client requests will be distributed to all nodes.
- An ingest node functions as a set of ELK for data conversion. If there is no ingest requests, ingest nodes are not in the idle state.

4.7 How Do I Change the Maximum Number of Results Returned in Response to a Single Search Request to a Specific Index in a CSS Elasticsearch Cluster?

Solution

- Method 1

  Open Kibana and run the following commands on the **Dev Tools** page:

```
PUT _all/_settings?preserve_existing=true
{
"index.max_result_window" : "10000000"
}
```

- Method 2

  Run the following command on a server (a non-security mode cluster is used as an example here):

  ```
  curl -k –XPUT 'http://localhost:9200/_all/_setting?preserve_existing=true'-d
  {
  "index.max_result_window":"1000000"
  }
  ```

  localhost indicates the address of the Elasticsearch cluster.

CAUTION

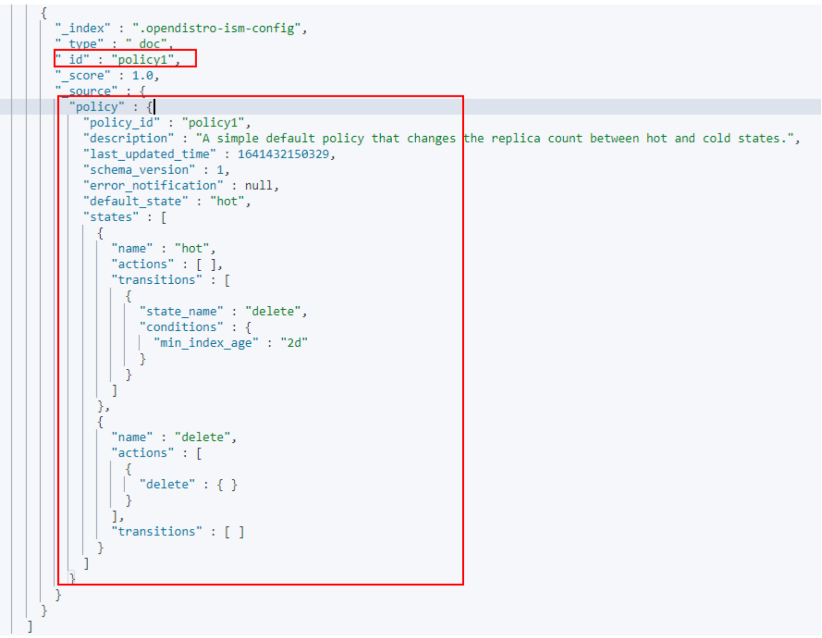
This setting affects the consumption of memory and CPU resources. Exercise caution when setting this parameter.

4.8 How Do I Update Index Lifecycle Policies for an Elasticsearch Cluster of CSS?

The lifecycle of Elasticsearch clusters is implemented using the Index State Management (ISM) of Open Distro. For details about how to configure policies related to the ISM template, see the **Open Distro documentation**.

1. When a policy is created, the system writes a record to the **.opendistro-ism-config** index. In the record, **_id** is the policy name, and the content is the policy definition.

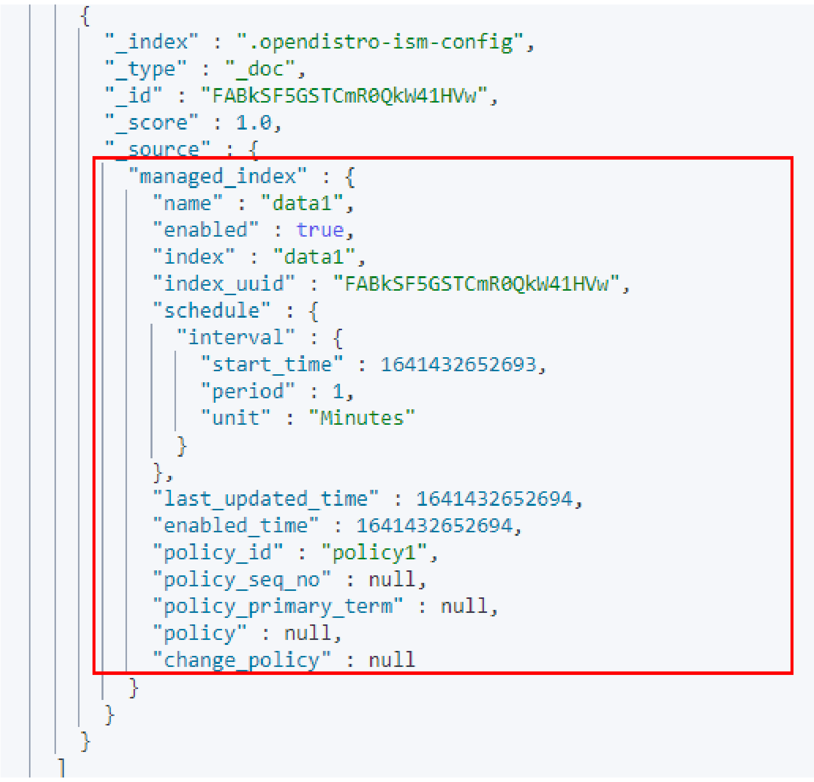
   Figure 4-3 Writing a data record

   ```
   {
     "_index" : ".opendistro-ism-config",
     "_type" : "_doc",
     "_id" : "policy1",
     "_score" : 1.0,
     "_source" : {
       "policy" : {
         "policy_id" : "policy1",
         "description" : "A simple default policy that changes the replica count between hot and cold states.",
         "last_updated_time" : 1641432150329,
         "schema_version" : 1,
         "error_notification" : null,
         "default_state" : "hot",
         "states" : [
           {
             "name" : "hot",
             "actions" : [ ],
             "transitions" : [
               {
                 "state_name" : "delete",
                 "conditions" : {
                   "min_index_age" : "2d"
                 }
               }
             ]
           },
           {
             "name" : "delete",
             "actions" : [
               {
                 "delete" : { }
               }
             ],
             "transitions" : [ ]
           }
         ]
       }
     }
   }
   ]
   ```

2. After a policy is bound to an index, the system writes another record to the **.opendistro-ism-config** index. The following figure shows the initial status of a record.

Figure 4-4 Initial data status



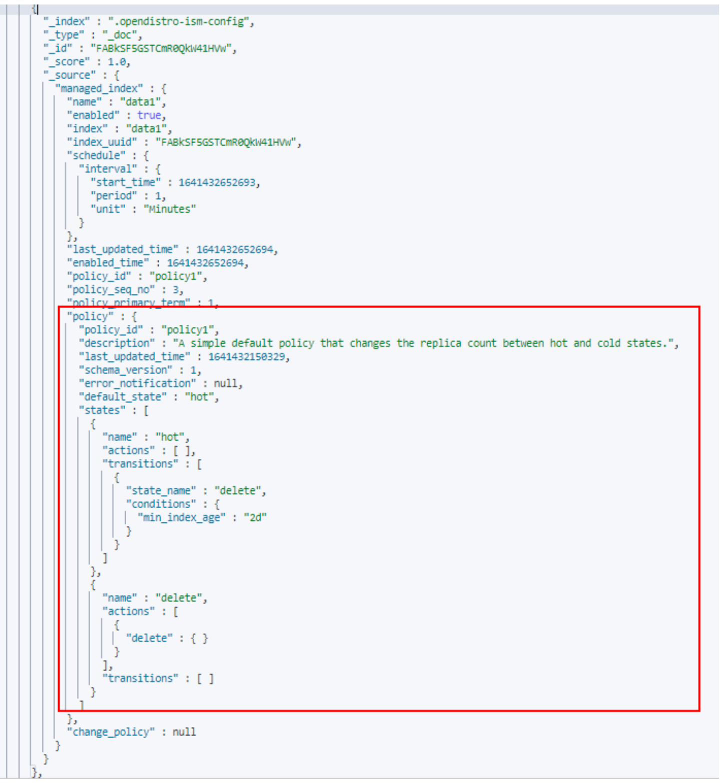
```
{
  "_index" : ".opendistro-ism-config",
  "_type" : "_doc",
  "_id" : "FABkSF5GSTCmR0QkW41HVw",
  "_score" : 1.0,
  "_source" : {
    "managed_index" : {
      "name" : "data1",
      "enabled" : true,
      "index" : "data1",
      "index_uuid" : "FABkSF5GSTCmR0QkW41HVw",
      "schedule" : {
        "interval" : {
          "start_time" : 1641432652693,
          "period" : 1,
          "unit" : "Minutes"
        }
      },
      "last_updated_time" : 1641432652694,
      "enabled_time" : 1641432652694,
      "policy_id" : "policy1",
      "policy_seq_no" : null,
      "policy_primary_term" : null,
      "policy" : null,
      "change_policy" : null
    }
  }
}
]
```

3. Run the **explain** command. Only a policy ID will be returned.

```
GET _opendistro/_ism/explain/data2
{
  "data2" : {
    "index.opendistro.index_state_management.policy_id" : "policy1"
  }
}
```

Open Distro will execute an initialization process to fill the policy content in the record. The following figure shows the initialized data.

Figure 4-5 Initialized data



```
{
  "_index" : ".opendistro-ism-config",
  "_type" : "_doc",
  "_id" : "FABkSF5GSTCmR0QkW41HVw",
  "_score" : 1.0,
  "_source" : {
    "managed_index" : {
      "name" : "data1",
      "enabled" : true,
      "index" : "data1",
      "index_uuid" : "FABkSF5GSTCmR0QkW41HVw",
      "schedule" : {
        "interval" : {
          "start_time" : 1641432652693,
          "period" : 1,
          "unit" : "Minutes"
        }
      },
      "last_updated_time" : 1641432652694,
      "enabled_time" : 1641432652694,
      "policy_id" : "policy1",
      "policy_seq_no" : 3,
      "policy_primary_term" : 1,
      "policy" : {
        "policy_id" : "policy1",
        "description" : "A simple default policy that changes the replica count between hot and cold states.",
        "last_updated_time" : 1641432150329,
        "schema_version" : 1,
        "error_notification" : null,
        "default_state" : "hot",
        "states" : [
          {
            "name" : "hot",
            "actions" : [ ],
            "transitions" : [
              {
                "state_name" : "delete",
                "conditions" : {
                  "min_index_age" : "2d"
                }
              }
            ]
          },
          {
            "name" : "delete",
            "actions" : [
              {
                "delete" : { }
              }
            ],
            "transitions" : [ ]
          }
        ]
      },
      "change_policy" : null
    }
  }
},
```

After the initialization, **min_index_age** in the policy will be copied.

NOTE

The initialized index uses a copy of this policy. The policy update will not take effect on the index.

4. After the policy is modified, call the **change_policy** API to update the policy.

```
POST _opendistro/_ism/change_policy/data1
{
  "policy_id": "policy1"
}
```

4.9 How Do I Set Slow Query Log Thresholds for an Elasticsearch Cluster of CSS?

The slow query log settings of CSS are the same as those of Elasticsearch. You can configure slow query logs via the _settings API. For example, you can run the following command in Kibana to set the index level:

```
PUT /my_index/_settings
{
    "index.search.slowlog.threshold.query.warn": "10s",
    "index.search.slowlog.threshold.fetch.debug": "500ms",
    "index.indexing.slowlog.threshold.index.info": "5s"
}
```

- If a query takes longer than 10 seconds, a WARN log will be generated.
- If retrieval takes longer than 500 milliseconds, a DEBUG log will be generated.
- If an index takes longer than 5 seconds, an INFO log will be generated.

For details, visit the official website: https://www.elastic.co/guide/cn/elasticsearch/guide/current/logging.html

4.10 How Do I Clear Elasticsearch Indexes in CSS?

⚠ CAUTION

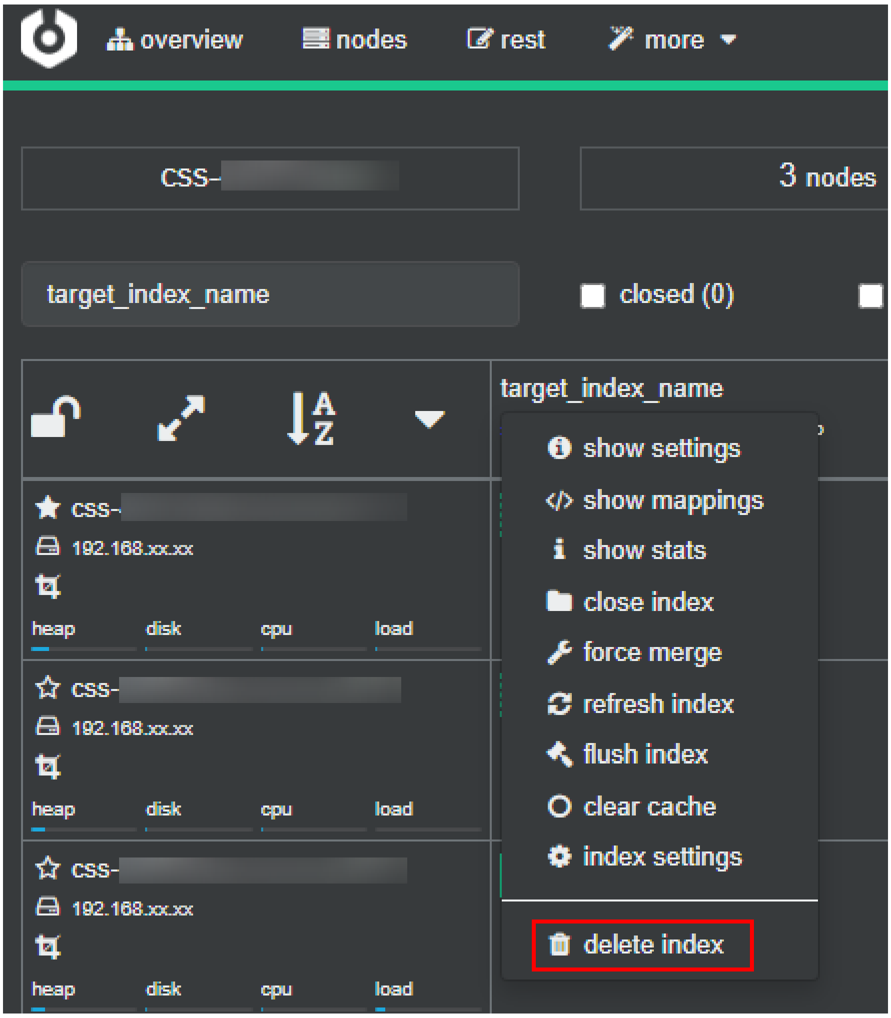
Before deleting index data, carefully evaluate any potential impact on services.

- Have indexes automatically cleared on a regular basis.

  You can create a scheduled task to execute an index deletion request periodically. CSS supports Open Distro Index State Management. For details about how to clear obsolete indexes periodically, see **Decoupling Index Storage and Compute Through Index Lifecycle Management**.

  For details about Open Distro Index State Management, see https://opendistro.github.io/for-elasticsearch-docs/docs/im/ism/.
- Manually clear indexes.
  - Log in to Kibana, go to the **Dev Tools** page, and run the following command to delete a specified index:
    ```
    DELETE /{index_name}
    ```
  - Log in to Cerebro, search for the target index name, click the index name, click **delete index**, and click **Confirm** in the displayed dialog box.

Figure 4-6 Deleting an index from Cerebro



4.11 How Do I Clear Elasticsearch Cache in CSS?

- **Clear the fielddata**

  During aggregation and sorting, data are converted to the fielddata structure, which occupies a large amount of memory.

  a. Run the following command on Kibana to query the fielddata cache status:

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

  b. If the memory usage of **fielddata** is too high, you can run the following command to clear the **fielddata cache** of a specified index or all indexes:

  ```
  POST /test/_cache/clear?fielddata=true
  ```

  In the preceding command, *test* indicates the name of the index whose fielddata occupies a large amount of memory.

  ```
  POST /_cache/clear?fielddata=true
  ```

- **Clear segments**

  The FST structure of each segment is loaded to the memory and will not be cleared. If the number of index segments is too large, the memory usage will be high. You are advised to periodically clear the segments.

a. Run the following command on Kibana to check the number of segments and their memory usage on each node:

```
GET /_cat/nodes?v&h=segments.count,segments.memory&s=segments.memory:desc
```

b. If the memory usage of segments is too high, you can delete or disable unnecessary indexes, or periodically combine indexes that are not updated.

- **Clear the cache**

  Run the following command on Kibana to clear the cache:

  ```
  POST /_cache/clear
  ```

4.12 Why Does the Disk Usage Increase After the delete_by_query Command Was Executed to Delete Data in an Elasticsearch Cluster?

Running the **delete_by_query** command only add a deletion mark to the target data, instead of really deleting it. When you search for data, all data is searched and the data with the deletion mark is filtered out.

The space occupied by an index with the deletion mark will not be released immediately after you call the disk deletion API. The disk space is released only when the segment merge is performed next time.

Querying the data with deletion mark occupies disk space. In this case, the disk usage increases when you run the disk deletion commands.

4.13 Do CSS Elasticsearch Clusters Support script dotProduct?

The Elasticsearch-native vector search function is provided via an X-Pack plugin, which is currently not integrated in CSS. This is why the native **script dotProduct** cannot be executed in Elasticsearch clusters created in CSS.

You are advised to use the vector search service provided by CSS. Based on an in-house developed vector search engine, CSS's vector search service is deeply integrated with the Elasticsearch plug-in architecture, featuring high-performance, high-precision, low-cost, and multi-modality. It offers an efficient, cost-effective solution to meet diversified, high-dimensional vector search needs. For more information, see **Vector Search**.

⚠ CAUTION

Only Elasticsearch 7.6.2 and 7.10.2 clusters in CSS support vector search.

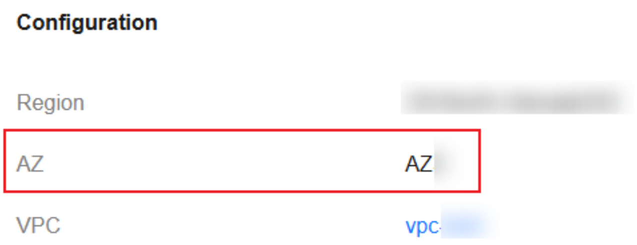
5 Managing CSS Clusters

5.1 How Do I Check the AZ or AZs of a CSS Cluster?

You can check the AZ or AZs (in the case of a multi-AZ deployment) of a cluster on the cluster information page.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab. In the **Configuration** area, find **AZ**, and check the cluster's AZ or AZs.

Figure 5-1 Checking the cluster's AZs



Configuration

Region [redacted]

AZ AZ[redacted]

VPC vpc-[redacted]

5.2 What Is the Relationship Between the Filebeat Version and Cluster Version in CSS?

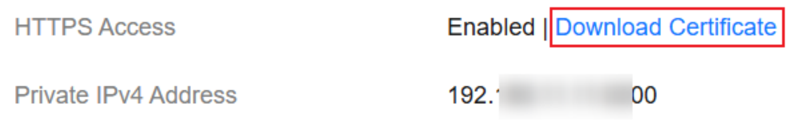
- Clusters with the security mode disabled: no restrictions.
- Clusters with the security mode enabled: The Filebeat OSS version must match the cluster version. For the correct Filebeat OSS version to download, see **https://www.elastic.co/downloads/past-releases#filebeat-oss**.

5.3 How Do I Obtain the Security Certificate of a CSS Security-Mode Cluster?

The security certificate (**CloudSearchService.cer**) can be downloaded only for security-mode clusters that have enabled HTTPS access. The security certificate cannot be used in the public network environment.

- Obtain the security certificate of an Elasticsearch cluster.
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > Elasticsearch**.
  c. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
  d. Click the **Overview** tab. In the **Configuration** area, click **Download Certificate** next to **HTTPS Access**.

Figure 5-2 Downloading a security certificate

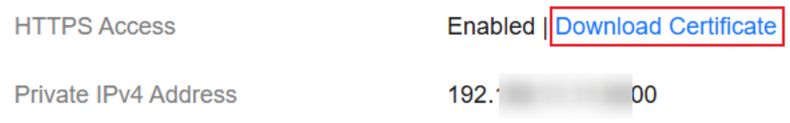


HTTPS Access Enabled | Download Certificate

Private IPv4 Address 192.[redacted]00

- Obtain the security certificate of an OpenSearch cluster.
  a. Log in to the **CSS management console**.
  b. In the navigation pane on the left, choose **Clusters > OpenSearch**.
  c. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
  d. Click the **Overview** tab. In the **Configuration** area, click **Download Certificate** next to **HTTPS Access**.

Figure 5-3 Downloading a security certificate



HTTPS Access Enabled | Download Certificate

Private IPv4 Address 192.[redacted]00

5.4 How Do I Convert the Format of a CER Security Certificate in CSS?

The security certificate (**CloudSearchService.cer**) can be downloaded only for security clusters that have enabled HTTPS access. Most software supports certificates in the **.pem** or **.jks** format. You need to convert the format of the CSS security certificate.

- Run the following command to convert the security certificate from **.cer** to **.pem**:

```
openssl x509 -inform pem -in CloudSearchService.cer –out newname.pem
```

- Run the following command to convert the security certificate from **.cer** to **.jks**:

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

In the preceding commands, *newname* indicates the user-defined certificate name.

After the command is executed, set the certificate password and confirm the password as prompted. Securely store the password. It will be used for accessing the cluster.

5.5 Can I Change the Security Group of a CSS Cluster?

You can change the security group of an existing CSS cluster.

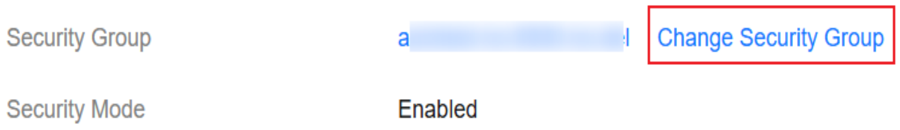
Constraints

- It is advisable to perform this operation during off-peak hours.
- The security group cannot be changed for clusters created before February 2023. To do that, create a new cluster by selecting the desired security group, then perform **Migrating Data Between Huawei Cloud Elasticsearch Clusters Using Backup and Restoration** to migrate its data to that new cluster.
- Make sure the inbound rules of the new security group allow all the ports required for service access. For Elasticsearch and OpenSearch clusters, port 9200 must be allowed. For Logstash clusters, port 9600 must be allowed.

Changing the Security Group

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab. In the **Configuration** area, click **Change Security Group** on the right of **Security Group**.

Figure 5-4 Changing the security group



Security Group a[redacted]l Change Security Group

Security Mode Enabled

5. In the **Change Security Group** dialog box, select a new security group and click **OK**.

5.6 Can I Expand the Subnet for an Elasticsearch or OpenSearch Cluster?

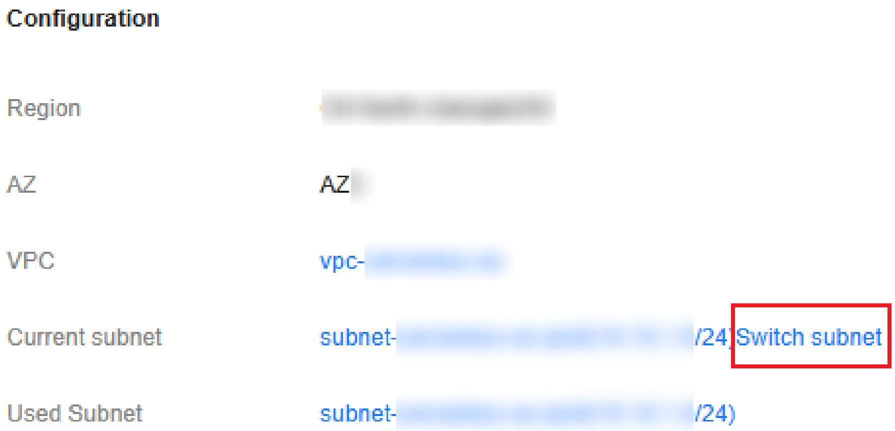
If the current subnet of an existing Elasticsearch or OpenSearch cluster cannot meet your requirements, you can switch to another subnet.

⚠ CAUTION

- After switching the subnet, the new subnet is used for adding nodes, including scaling, adding dedicated master/client nodes, and enabling VPC Endpoint. IP addresses will be allocated to newly added nodes from the new subnet.
- A cluster can be associated with a maximum of two subnets.
- Subnet switching does not affect the network settings of existing nodes.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab.
5. Click **Change Subnet** on the right of **Current Subnet**. In the displayed dialog box, select a new subnet. If there are no subnets that meet your requirements, click **Create Subnet** to go to the networking service console to create a new subnet.

Figure 5-5 Switching to another subnet



Configuration

Region [redacted]

AZ AZ[redacted]

VPC vpc-[redacted]

Current subnet subnet-[redacted]/24) Switch subnet

Used Subnet subnet-[redacted]/24)

6. Click **OK** after you change the subnet.

5.7 How Do I Set search.max_buckets for an Elasticsearch Cluster of CSS?

Function

By default, CSS allows a maximum of 10,000 buckets to be returned during aggregation. If more than 10,000 buckets need to be returned, you can increase the value of **search.max_buckets**. Note that increasing the value of **search.max_buckets** also increases the cluster load and memory usage. Exercise caution when performing this operation.

Solution

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

```
PUT _cluster/settings
{
   "persistent": {
      "search.max_buckets": 20000
   }
}
```

5.8 How Do I Modify the TLS Algorithm for a CSS Cluster?

The TLS algorithm can be modified for Elasticsearch 7.6.2 and later as well as OpenSearch clusters.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings**.
5. Click **Edit**, expand **Custom**, and click **Add**.
   - For an Elasticsearch cluster, add the **opendistro_security.ssl.http.enabled_ciphers** parameter and set it to **['TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256', 'TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384']**.
   - For an OpenSearch cluster, add the **plugins.security.ssl.http.enabled_ciphers** parameter and set it to **['TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256', 'TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384']**.

   If the parameter value contains multiple algorithms, enclose the value with a pair of square brackets ([]). If the parameter value is a single algorithm, enclose the value with a pair of single quotation marks(' ').
6. After the change is complete, click **Submit**.In the displayed **Submit Configuration** dialog box, select the box indicating "I understand that the modification will take effect after the cluster is restarted." and click **Yes**.

If the **Status** is **Succeeded** in the parameter change list, the change has been saved.

7. On the cluster information page, click **Restart** in the upper-right corner to restart the cluster, thus making the change take effect.

5.9 How Do I Enable Audit Logs for a CSS Cluster?

Audit logs can be enabled for security-mode Elasticsearch 7.6.2 clusters as well as security-mode OpenSearch clusters.

Audit logs are disabled for Elasticsearch clusters by default.

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Parameter Settings**.
5. Click **Edit**, expand **Custom**, and click **Add**.
   - For an Elasticsearch cluster, set **Key** to **opendistro_security.audit.type** and **Value** to **internal_elasticsearch**.
   - For an OpenSearch cluster, set **Key** to **plugins.security.audit.type** and **Value** to **internal_opensearch**.
6. After the change is complete, click **Submit**.In the displayed **Submit Configuration** dialog box, select the box indicating "I understand that the modification will take effect after the cluster is restarted." and click **Yes**.

   If the **Status** is **Succeeded** in the parameter change list, the change has been saved.
7. On the cluster information page, click **Restart** in the upper-right corner to restart the cluster, thus making the change take effect.
8. After cluster restart, check whether audit logs have been enabled.
   a. For an Elasticsearch cluster, click **Kibana** in the **Operation** column to log in to Kibana. For an OpenSearch cluster, click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
   b. Expand the menu in the upper-left corner, and choose **Dev Tools**.
   c. Run the following command. If the result contains indexes whose name contain **.*audit***, audit logs have been enabled.

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

5.10 Can I Stop a CSS Cluster?

To stop a CSS cluster, you delete it. If you want to stop a cluster upon completion of a cluster migration, stop all services in the source cluster, and confirm that all data has been migrated to the destination cluster. Then delete the source cluster. You can stop services in the source cluster in the following ways:

- If the cluster version in use supports the flow control function, you can enable **one-click traffic blocking** to block traffic everywhere except the O&M interface, rejecting all requests.

- If your cluster version in use does not support traffic control, you can disable read and write for all service indexes instead. For example, if all service indexes start with **log**, run the following command on the **Dev Tools** page of Kibana:

  ```
  PUT log*/_settings
  {
    "index.blocks.read": true,
    "index.blocks.write": true,
    "index.blocks.metadata": true
  }
  ```

5.11 How Do I Query the Index Size on OBS After the Freezing of Indexes for a CSS Cluster?

The size of indexes remains unchanged after freezing. By querying the size of frozen indexes in OBS, you obtain the size of all indexes stored on OBS.

Run the following command to obtain information about all indexes that are being frozen or have already been frozen:

```
GET _cat/freeze_indices?stage=$
```

The output is as follows (as an example only):

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

The last column of the returned result contains the index size information.

Related Questions

- **Billing for index storage on OBS**

  Storing index data in OBS incurs additional charges. For details, see "Standard cold storage prices" in **Cloud Search Service Pricing Details**.

- **Why are frozen indexes stored in OBS still searchable via commands?**

  Elasticsearch and OpenSearch clusters use local storage by default, and Lucene index files are stored on local disks. Lucene interacts with the underlying storage via the Directory API. Files can be read through the following API:

  ```
  public abstract IndexInput openInput(String name, IOContext context) throws IOException;
  ```

  The storage-compute decoupling feature enables interaction with OBS through the Directory API to read files stored in OBS. This is why information about frozen indexes stored in OBS can be queried using commands.

5.12 How Do I Check the List of Default Plugins for a CSS Cluster?

Default plugins are available for the Elasticsearch and OpenSearch clusters in CSS. You can check them on the CSS management console or query them by running commands on Kibana or OpenSearch Dashboards.

Querying Default Plugins on the CSS Management Console

1. Log in to the **CSS management console**.

2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Choose **Cluster Settings** > **Plugins**.
5. On the **Default Plugins** tab, check the default plugins supported by the current cluster.

Querying Default Plugins by Running Commands

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. For an Elasticsearch cluster, click **Kibana** in the **Operation** column to log in to Kibana. For an OpenSearch cluster, click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. Expand the menu in the upper-left corner, and choose **Dev Tools**.
5. Run the following command to check the cluster's plugins:

   ```
   GET _cat/plugins?v
   ```

   The following is an example of the response body:

   ```
   name                  component                       version
   css-test-ess-esn-1-1 analysis-dynamic-synonym        x.x.x-xxxx-ei-css-v1.0.1
   css-test-ess-esn-1-1 analysis-icu                    x.x.x-xxxx-ei-css-v1.1.6
   css-test-ess-esn-1-1 analysis-ik                     x.x.x-xxxx-ei-css-v1.0.1
   ......
   ```

   name indicates the cluster node name, **component** indicates the plugin name, and **version** indicates the plugin version.

5.13 How Do I Plan the Quantity of Index Shards for a Cluster?

Before importing data to a cluster, carefully consider your service needs and plan the cluster's data structure and distribution in advance. This includes properly designing indexes and deciding on the appropriate number of index shards. To ensure optimal performance and scalability for a cluster, consider following these best practices:

- **The size of a single shard**: Keep the size of each shard between 10 GB and 50 GB. This helps strike a balance between storage efficiency and query performance.
- **Memory-to-shards ratio**: Limit the number of shards per 1 GB of memory to 20 to 30. This ensures that each shard has sufficient memory resources to respond to indexing and query requests.
- **Number of shards per node**: To prevent node overload, keep the number of shards on each node under 1000. This helps to improve node stability.
- **Relationship between the number of index shards and the number of nodes**: For each index, make sure the number of shards is an integral multiple of the total number of data nodes and cold data nodes in the cluster. This helps improve load balancing and optimize query and indexing performance.

- **Total number of shards in a cluster**: To facilitate management and avoid oversized shards, make sure the total number of shards in a cluster is less than 30,000. This helps maintain the stability and responsiveness of the cluster.

Following these suggestions, you can plan and manage index shards for a CSS cluster more effectively, improving the cluster's overall performance and maintainability.

6 CSS Cluster Backup and Restoration

6.1 How Do I Query the Snapshot Information of a CSS Cluster?

You can query snapshot information only if cluster snapshots are enabled and some snapshots have been created.

You can check them on the CSS management console or query them by running commands on Kibana or OpenSearch Dashboards.

Querying Snapshots on the CSS Management Console

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Cluster Snapshots** tab.
5. In the cluster snapshot task list, click the target snapshot name, and check snapshot information in the displayed dialog box.

Querying Snapshots by Running Commands

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. For an Elasticsearch cluster, click **Kibana** in the **Operation** column to log in to Kibana. For an OpenSearch cluster, click **Dashboards** in the **Operation** column to log in to OpenSearch Dashboards.
4. Expand the menu in the upper-left corner, and choose **Dev Tools**.
5. Run the following command to query the snapshot repository information, that is, the basic cluster snapshot settings.

   ```
   GET _snapshot/_all
   ```

   Example response:

```
{
  "repo_auto": {
    "type": "obs",
    "settings": {
      "bucket": "123xxx",      // OBS bucket name
      "chunk_size": "2g",      // Chunk size in snapshots (unit: GB)
      "endpoint": "obs.xxx.example.com:443",      // OBS VPC endpoint
      "max_restore_bytes_per_sec": "0MB",      // Maximum recovery rate (per second)
      "compress": "true",      // Whether to enable data compression
      "base_path": "css_repository/css-xxx",      // Backup path
      "region": "xxx",      // Region
      "max_snapshot_bytes_per_sec": "40MB"      // Maximum backup rate (per second)
    }
  }
}
```

6. Run the following command to query the snapshot list:

```
GET _snapshot/repo_auto/_all
```

Example output (two snapshots are returned):

```
{
  "snapshots": [
    {
      "snapshot": "snapshot-2dc3",       // Snapshot name
      "uuid": "VW5y2NBJS9iPh7YcGLxxxx",  // Snapshot ID
      "version_id": xxxxxxx,           // Internal cluster version
      "version": "x.x.x",           // Cluster version
      "indices": [               // Backed up indexes
        ".opendistro_security"
      ],
      "data_streams": [ ],           // Data streams
      "include_global_state": true,      // Whether to include the global cluster status
      "state": "SUCCESS",               // Snapshot status
      "start_time": "2025-08-30T01:41:57.068Z",  // Snapshot start time
      "start_time_in_millis": 1756518117068,      // Snapshot start time in milliseconds
      "end_time": "2025-08-30T01:41:57.469Z",      // Snapshot end time
      "end_time_in_millis": 1756518117469,      // Snapshot end time in milliseconds
      "duration_in_millis": 401,               // Snapshot creation duration (from start to completion) in milliseconds
      "failures": [ ],                   // Failed shards during snapshot creation
      "shards": {                        // Shard statistics
        "total": 1,                       // Total number of shards
        "failed": 0,                       // Number of failed shards
        "successful": 1                    // Number of successful shards
      }
    },
    {
      "snapshot": "snapshot-dd37",
      "uuid": "FD4VcooLS8yjPY3w0-x-xx",
      "version_id": xxxxxxx,
      "version": "x.x.x",
      "indices": [
        ".kibana",
        ".opendistro_security"
      ],
      "data_streams": [ ],
      "include_global_state": true,
      "state": "SUCCESS",
      "start_time": "2025-08-30T01:54:55.750Z",
      "start_time_in_millis": 1756518895750,
      "end_time": "2025-08-30T01:54:55.950Z",
      "end_time_in_millis": 1756518895950,
      "duration_in_millis": 200,
      "failures": [ ],
      "shards": {
        "total": 2,
        "failed": 0,
        "successful": 2
      }
    }
  ]
}
```

To query information about a specified snapshot, run the following command:

```
GET _snapshot/repo_auto/{snapshot_name}
```

Replace *snapshot_name* with the actual snapshot name. Wildcard matching is supported.

7. (Optional) Run the following command to delete a specified snapshot:

   ```
   DELETE _snapshot/repo_auto/{snapshot_name}
   ```

   Exercise caution when choosing snapshots to delete.

6.2 Can a Deleted CSS Cluster Be Restored?

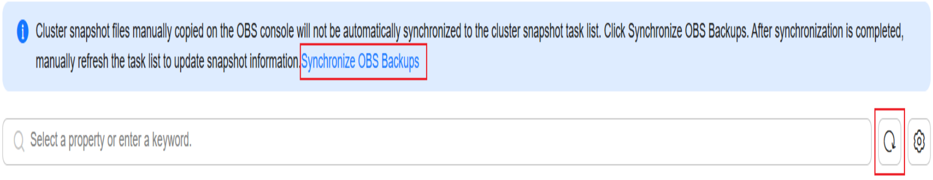
For a deleted Elasticsearch or OpenSearch, if it still has snapshots stored in OBS, it can be restored using these snapshots. Otherwise, it cannot be restored. Therefore, exercise caution when deciding to delete a cluster.

To restore a deleted cluster using one of its snapshots stored in OBS, perform the following steps:

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. Click **Create Cluster** in the upper-right corner to create a cluster that is of the same type as the deleted cluster. Make sure that:
   - The new cluster is in the same region as the deleted cluster.
   - The cluster version is the same as or later than that of the deleted cluster.
   - The number of nodes in the new cluster is greater than half of that in the deleted cluster. Ideally, they should have the same number of nodes. This is to prevent data restoration failures.
   - Deselect **Automatic Snapshot Creation**.
4. When the status of the new cluster changes to **Available**, enable cluster snapshots.
   a. In the cluster list, click the name of the new cluster. The cluster information page is displayed.
   b. Click the **Cluster Snapshots** tab.
   c. Click **Enable Snapshot**. In the displayed dialog box, configure the necessary settings. Keep **OBS Bucket** and **Backup Path** consistent with those of the deleted cluster.
   d. Click **OK**.
5. Above the cluster snapshot task list, click **Synchronize OBS Backups**. After data synchronization, manually refresh the list to show the updated snapshot information.

   The snapshots of the deleted cluster are displayed in the cluster snapshot task list.

Figure 6-1 Synchronizing snapshots stored in OBS



Cluster snapshot files manually copied on the OBS console will not be automatically synchronized to the cluster snapshot task list. Click Synchronize OBS Backups. After synchronization is completed, manually refresh the task list to update snapshot information. Synchronize OBS Backups

6. Restore data from snapshots.
   a. In the cluster snapshot task list, locate the needed snapshot, and click **Restore** in the **Operation** column. In the displayed dialog box, configure index settings and select the current cluster in **Cluster**.
   b. Click **OK**. When the snapshot's **Task Status** changes to Restoration succeeded, data has been restored successfully.

      If there are multiple snapshots, restore all of them in the right order to restore the data of the deleted cluster.

7 CSS Cluster Monitoring and O&M

7.1 What Do I Do If the Average Memory Usage of a CSS Cluster Reaches 98%?

Symptom

The cluster monitoring result shows that the average memory usage of a cluster is 98%. Does it affect cluster performance?

Possible Cause

In an Elasticsearch cluster, 50% of the memory is occupied by Elasticsearch and the other 50% is used by Lucene to cache files. It is normal that the average memory usage reaches 98%.

Solution

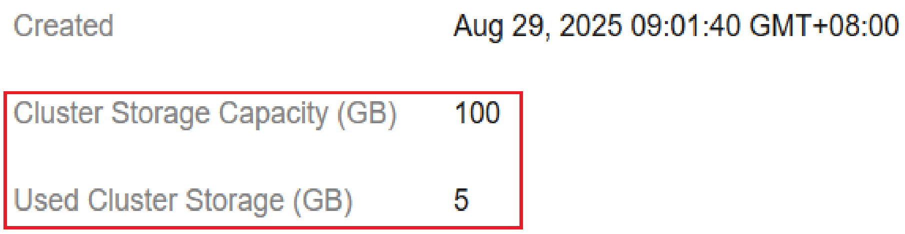
You can monitor the cluster memory usage by checking the maximum JVM heap usage and average JVM heap usage.

7.2 How Do I Check the Total Disk Usage of a CSS Cluster?

1. Log in to the **CSS management console**.
2. In the navigation pane on the left, expand **Clusters**. Select a cluster type based on the target cluster. The cluster list is displayed.
3. In the cluster list, click the name of the target cluster. The cluster information page is displayed.
4. Click the **Overview** tab.
5. In the **Cluster Information** area, obtain **Cluster Storage Capacity (GB)** and **Used Cluster Storage (GB)**.

   The cluster's total disk usage = Used cluster storage/Cluster storage capacity

Figure 7-1 Obtaining a cluster's storage capacity and usage information



Created	Aug 29, 2025 09:01:40 GMT+08:00
Cluster Storage Capacity (GB)	100
Used Cluster Storage (GB)	5

7.3 Will CSS Cluster Services Be Affected If the Disk Usage of a Single Node Gets Too High?

Symptom

According to the cluster monitoring information, the disk usage of an Elasticsearch cluster exceeds 80%. Does it affect cluster performance?

Impact on Services

- When the disk usage of a node exceeds 85%, disk space cannot be allocated to new replicas, but can still be allocated to new primary shards. This ensures service continuity, but it impacts the high availability of the Elasticsearch cluster.
- When the disk usage of a node exceeds 90%, a shard migration mechanism is automatically triggered to reallocate shards on this node to other data nodes with lower disk usage. During this process, disk space cannot be allocated to new shards. Shard migration and reallocation may increase query delay or temporarily interrupt services, impacting service continuity.
- When the disk usage of a node exceeds 95%, the **read_only_allow_delete** attribute will be enabled in its indexes. In this case, indexes on this node can only be read or deleted but data cannot be written in.

If per-node resource usage is too high, you can add more nodes or expand the capacity of existing nodes. For details, see **Scaling Out an Elasticsearch Cluster**. Indexes will not be allocated to new nodes immediately. You can open the Cerebro file to check index allocation to nodes. You can also change the values of **indices.recovery.max_bytes_per_sec** and **cluster.routing.allocation.cluster_concurrent_rebalance** to speed up index allocation.