Cloud Search Service

FAQs

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 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 indices. 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, and ultra-high I/O.
- The EVS disk size varies depending on the node specifications you selected when creating a cluster.

You can configure up to 200 nodes for a cluster (each node is an ECS). The maximum storage capacity of an ECS is the total capacity of EVS disks attached to the 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 in the Disk Spaces of a CSS Cluster?

You can store the following logs and files:

- Log files: Elasticsearch logs
- Data files: Elasticsearch index files
- Other files: cluster configuration files
- OS: 5% storage space 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 Can I Export Data from Kibana in CSS?

- With Elasticsearch 7.10.2 (image version 7.10.2_24.3.3_* or later), Kibana supports one-click data export in CSV.

NOTICE

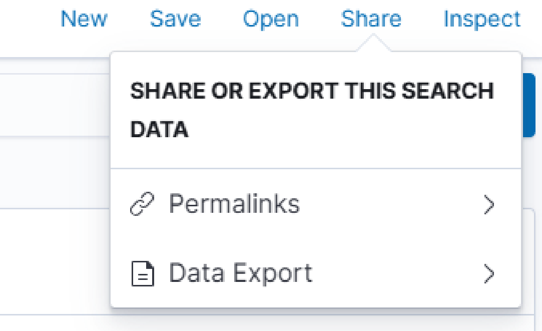
- A maximum of 10 MB of data can be exported. If the data you want to export exceeds 10 MB, only the first 10 MB is exported.
- Any special characters such as **=+-@** in exported CSV files may be identified as part of some formulas, leading to data export failures.
- After a cluster is upgraded to 7.10.2 (cross-version upgrade), by default, users do not have the permission to export logs. To enable this permission, you need to run the following commands on the DevTool page of Kibana:

```
### Create role csv_export.
PUT _opendistro/_security/api/roles/csv_export
{
  "cluster_permissions": [
    "indices_all"
  ],
  "index_permissions": [{
    "index_patterns": [
      ".css-exporting*"
    ],
    "dls": "",
    "fls": [],
    "masked_fields": [],
    "allowed_actions": [
      "indices_all"
    ]
  }]
}

### Assign the csv_export role to user kibanaserver.
PUT _opendistro/_security/api/rolesmapping/csv_export
{
  "backend_roles" : [],
  "hosts" : [],
  "users" : [ "kibanaserver" ]
}
```

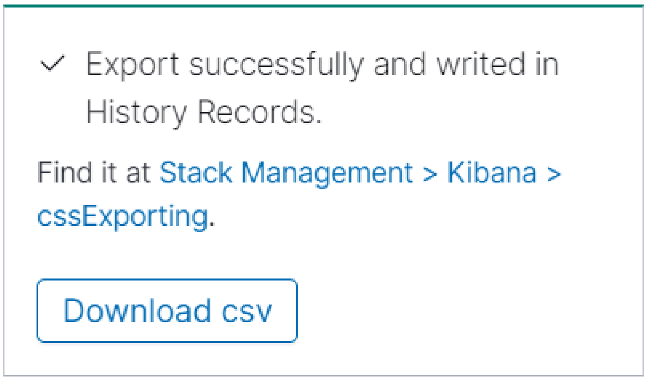
On the **Discover** page of Kibana, choose **Share > Data Export** in the upper right corner, and select **Export CSV**.

Figure 1-1 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 1-2 Downloading data



✓ Export successfully and writed in History Records.

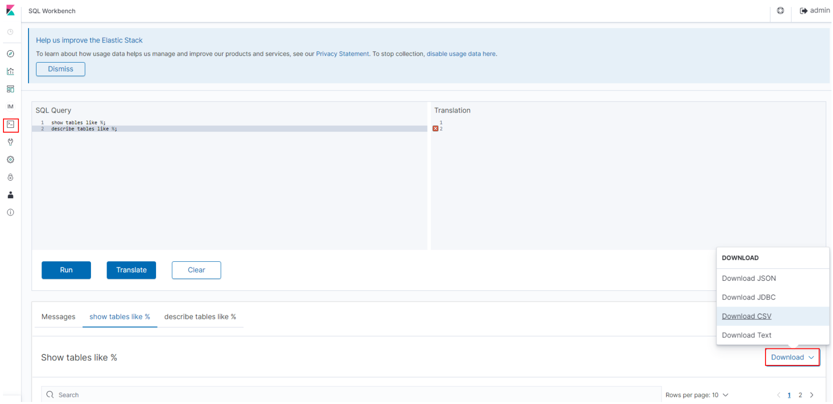
Find it at Stack Management > Kibana > cssExporting.

- 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 Elasticsearch SQL statements, see **Elasticsearch SQL**.

Figure 1-3 SQL Workbench



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.

SQL Query

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

Show tables like %

2 Accessing CSS Clusters

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

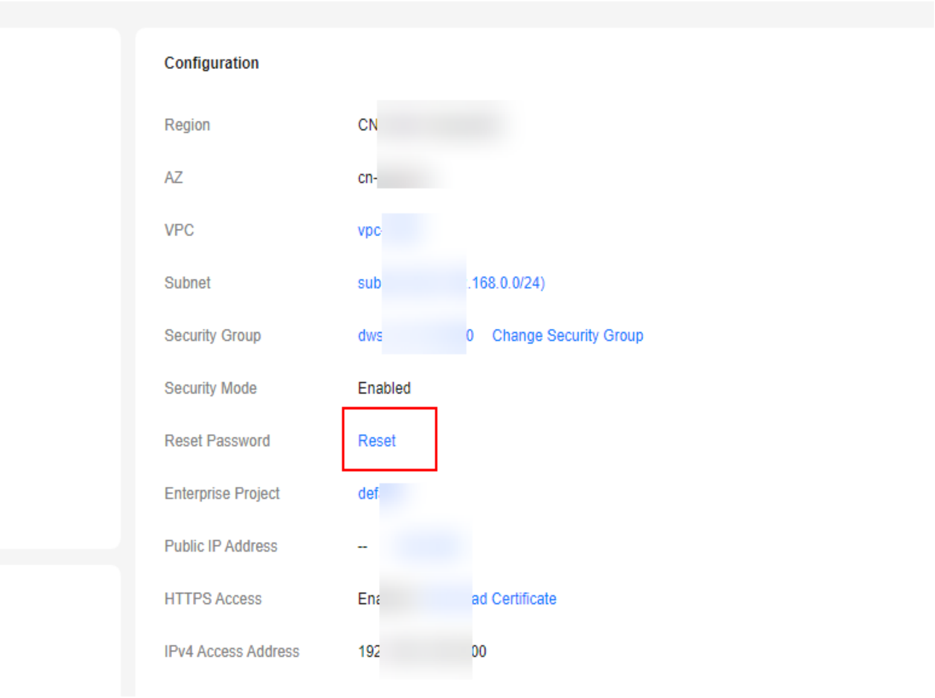
If you want to change the administrator password of a security cluster or you have forgotten the password, reset the password.

1. On the **Clusters** page, locate the target cluster whose password you want to reset and click the cluster name. The **Cluster Information** page is displayed.
2. In the **Configuration** area, click **Reset** next to **Reset Password**. Set and confirm the new administrator password.

   NOTE

   - 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



Configuration

Region	CN[redacted]
AZ	cn-[redacted]
VPC	vpc[redacted]
Subnet	sub[redacted].168.0.0/24)
Security Group	dws[redacted]0 Change Security Group
Security Mode	Enabled
Reset Password	Reset
Enterprise Project	def[redacted]
Public IP Address	-- [redacted]
HTTPS Access	Ena[redacted]ad Certificate
IPv4 Access Address	192[redacted]00

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 Access CSS from the Internet?

Perform the following operations:

1.**Obtaining CSS Information**

2.**Configuring a NAT Gateway**

3.**Modifying Security Group Rules**

4.**Accessing CSS from the Internet**

⚠ CAUTION

If your CSS clusters do not have the security mode enabled, do not access CSS through the NAT gateway. Otherwise, the cluster data will be exposed to the Internet.

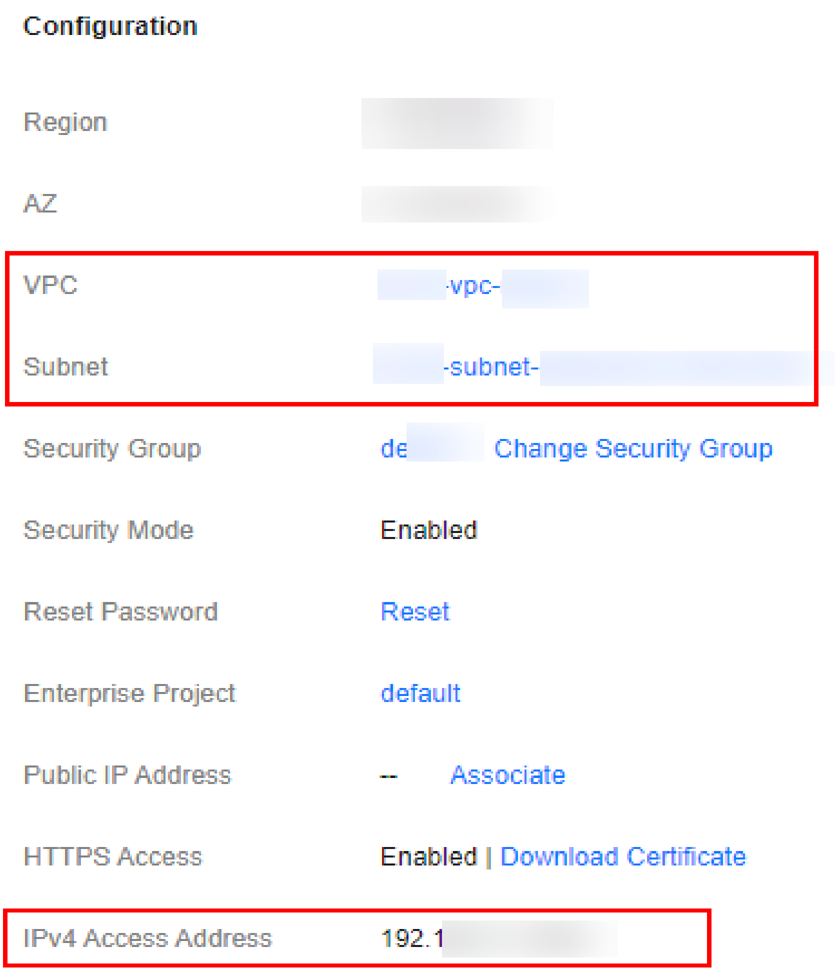
Obtaining CSS Information

Step 1 Log in to the CSS management console.

Step 2 On the **Clusters** page, click the name of a cluster. The **Basic Information** page is displayed by default.

Step 3 In the **Configuration Information** area, view the **Private Network Address**, **VPC**, and **Subnet** information.

Figure 2-2 Required information



Configuration

Region	
AZ	
VPC	vpc-
Subnet	-subnet-
Security Group	de Change Security Group
Security Mode	Enabled
Reset Password	Reset
Enterprise Project	default
Public IP Address	-- Associate
HTTPS Access	Enabled | Download Certificate
IPv4 Access Address	192.1

----End

Configuring a NAT Gateway

Step 1 Create a NAT gateway.

1. Log in to the console and choose **Service List** > **Networking** >**NAT Gateway**. The **Network Console** page is displayed.
2. Click **Buy Public NAT Gateway**. On the displayed page, configure related parameters. For details, see section "Buying a NAT Gateway" in *NAT Gateway User Guide*.

   NOTE

   Set **VPC** and **Subnet** to the values you obtained in **Obtaining CSS Information**.
3. Click **Next**, confirm the configurations, and click **Pay Now**.

Step 2 Add DNAT rules.

1. On the **Public NAT Gateways** page, click the name of the NAT gateway you purchased. The details page is displayed.
2. Choose **DNAT Rules** > **Add DNAT Rule**. For details, see section "Adding a DNAT Rule" in the *NAT Gateway User Guide*. When configuring DNAT rules, use the following settings:

   NOTE

   - **EIP**: Create an EIP on the **EIPs** page based on your service requirements.
   - **Outside Port**: Custom.
   - **Private IP Address**: private network IP address of CSS, which is the **Private Network Address** you obtained in **Obtaining CSS Information**.
   - **Inside Port**: 9200.
   - If your cluster contains multiple private IP addresses, add one DNAT rule for each address.
3. Click **OK**.

----End

Modifying Security Group Rules

Step 1 Log in to the CSS management console. In the navigation pane, click **Clusters**. On the displayed **Clusters** page, click the name of the target cluster to go to the **Basic Information** page

Step 2 On the **Basic Information** page, click **Security Group**.

Step 3 On the **Basic Information** page of the security group, click the **Inbound Rules** tab.

Step 4 Click **Add Rule** to add an inbound rule for port 9200.

Step 5 Click **OK**.

----End

Accessing CSS from the Internet

Enter **https://***IP*:*port* or **http://***IP*:*port* in the address box of the browser.

- ***IP*** and ***port*** are an 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 that you set for security mode on the displayed page.
- If you have not enabled **Security Mode** for the cluster, just 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 **Elastic Cloud Server (ECS) User Guide**.
2. Obtain the address for accessing the Elasticsearch cluster of CSS.
   a. In the navigation pane on the left, choose **Clusters**.
   b. In the cluster list, obtain the IP address of the cluster you want to access from the **Private Network 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
```

NOTE

- In security mode, the opendistro_security_kibana plug-in must be installed. For details, see **security-kibana-plugin**.
- The version of the installed plug-in 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 access the EIP bound to 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 **Elastic Cloud Server (ECS) User Guide**.
2. Obtain the address for accessing the OpenSearch cluster of CSS.
   a. In the navigation pane on the left, choose **Clusters**.
   b. In the cluster list, obtain the IP address of the cluster you want to access from the **Private Network 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

- In security mode, the opensearch_security_dashboards plug-in must be installed. For details, see **security-dashboards-plugin**.
- The version of the installed plug-in 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 access the EIP bound to 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.

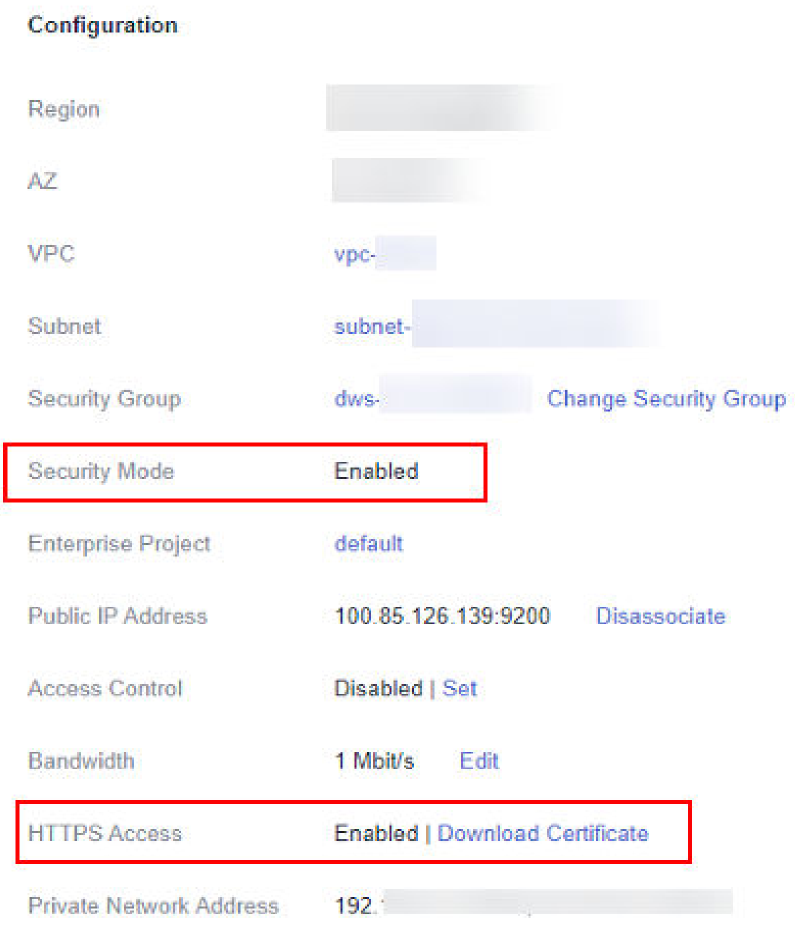
3.3 Examples of Logstash Configuration Files for Migrating Elasticsearch Clusters Using CSS Logstash

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** to go to the Elasticsearch cluster list.
3. Select the source or destination Elasticsearch cluster, and click the cluster name to go to the cluster details page.
4. Check whether the security mode and HTTPS are enabled for the cluster. **Figure 3-1** shows an Elasticsearch cluster with the security mode and HTTPS protocol both enabled.

Figure 3-1 Checking cluster settings



Configuration

Region	
AZ	
VPC	vpc-
Subnet	subnet-
Security Group	dws- Change Security Group
Security Mode	Enabled
Enterprise Project	default
Public IP Address	100.85.126.139:9200 Disassociate
Access Control	Disabled | Set
Bandwidth	1 Mbit/s Edit
HTTPS Access	Enabled | Download Certificate
Private Network Address	192.

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**
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
    }
}
```

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. **Number of replica shards = Number of primary shards x Number of replicas.**

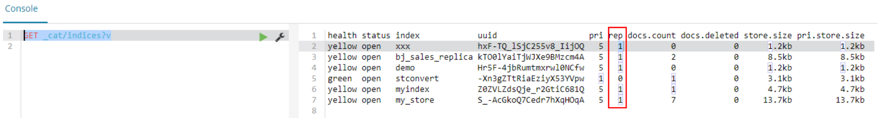
You can change the number of index replicas of an Elasticsearch cluster in Kibana.

1. Log in to the CSS management console.

2. In the navigation pane on the left, choose **Clusters** > **Elasticsearch**.
3. Locate the target cluster, and click **Access Kibana** in the **Operation** column to log in to Kibana.
4. Click **Dev Tools** in the navigation tree on the left.
5. 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_1SjC255v8_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
```

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

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

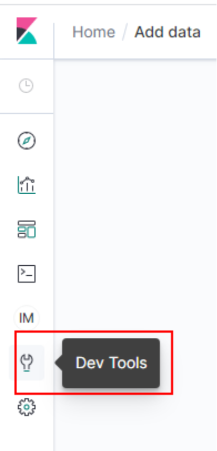
indexname indicates the name of the index to be modified, and **number_of_replicas** indicates the number of replicas to be set.

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

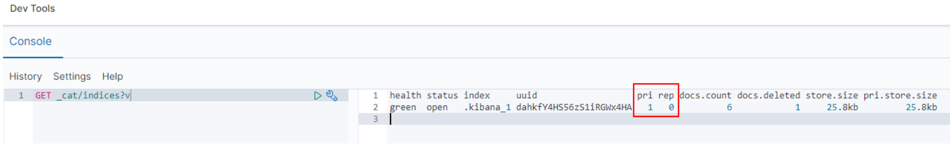
1. A large number of shards in a cluster slows down shard creation.
2. If automatic index creation is enabled, slow index creation may cause a large number of write requests to be stacked in the memory or result in a cluster breakdown.
3. If there are too many shards and you cannot properly monitor workloads,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

1. Log in to the CSS management console.
2. On the **Clusters** page, click **Access Kibana** in the **Operation** column of a cluster.
3. Log in to Kibana and choose **Dev Tools**.



4. On the **Console** page, run the **GET _cat/indices?v** command query the number of shards and replicas in a cluster. In the following figure, the **pri** column indicates the number of index shards, and the **rep** column indicates the number of replicas. After an index is created, its **pri** value cannot be modified. Its **rep** value can be modified.



```
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 for Searches to an Index in an Elasticsearch Cluster of CSS?

Solution

- Method 1

  Open Kibana and run the following commands on the **DevTools** 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

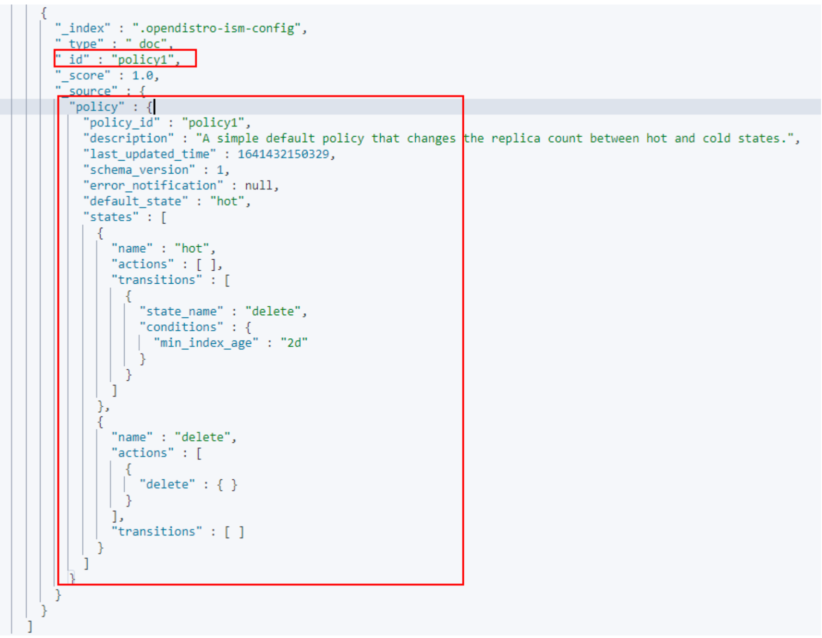
This configuration consumes memory and CPU resources. Exercise caution when setting this parameter.

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.

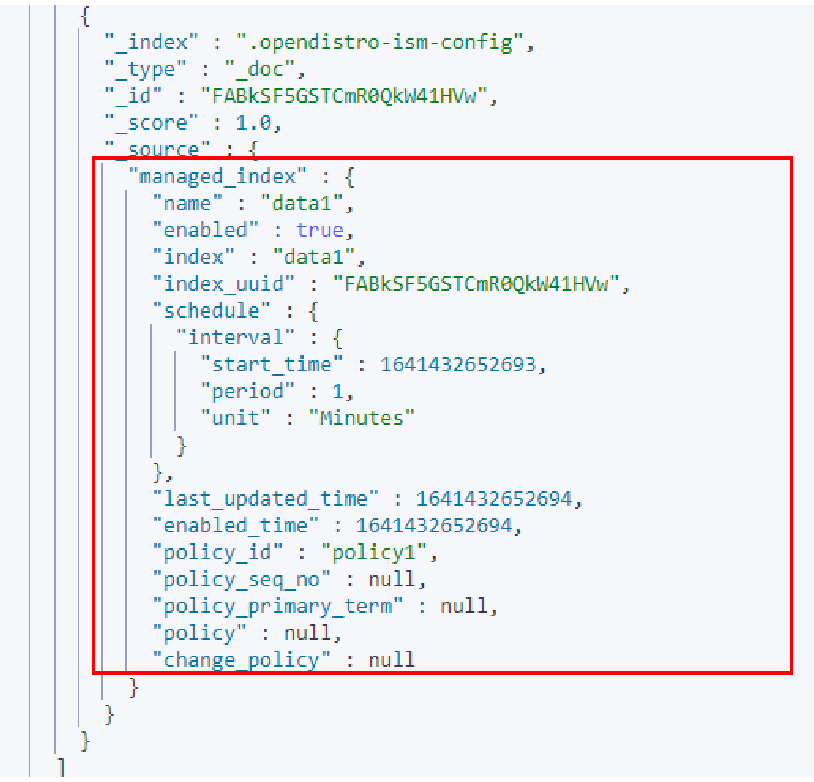
Figure 4-2 Writing a data record



```
{
  "_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 4-3 Initial data status



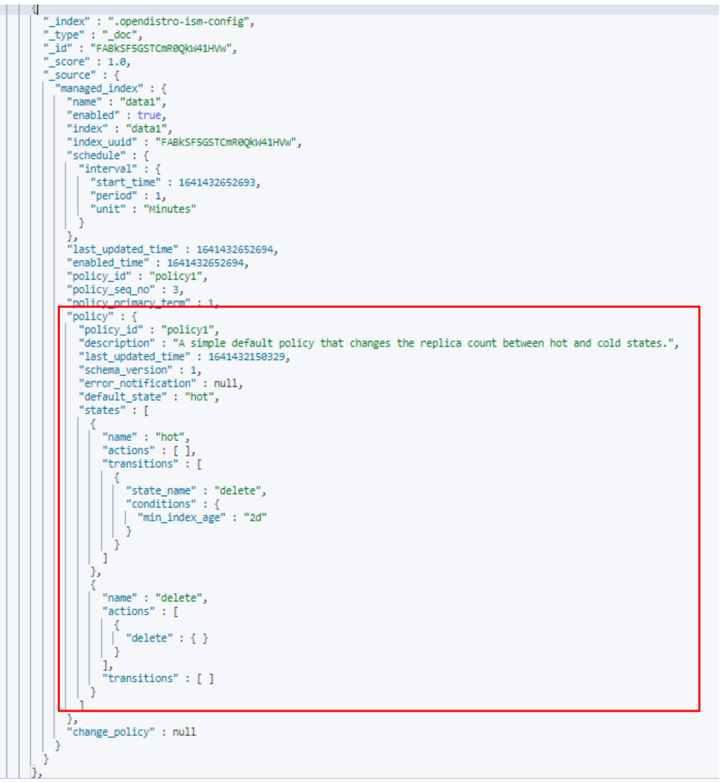
```
{
  "_index" : ".opendistro-ism-config",
  "_type" : "_doc",
  "_id" : "FABkSF5GSTCmR0QkW41HVw",
  "_score" : 1.0,
  "_source" : {
    "managed_index" : {
      "name" : "data1",
      "enabled" : true,
      "index" : "data1",
      "index_uuid" : "FABkSF5GSTCmR0QkW41HVw",
      "schedule" : {
        "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 4-4 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?

NOTE

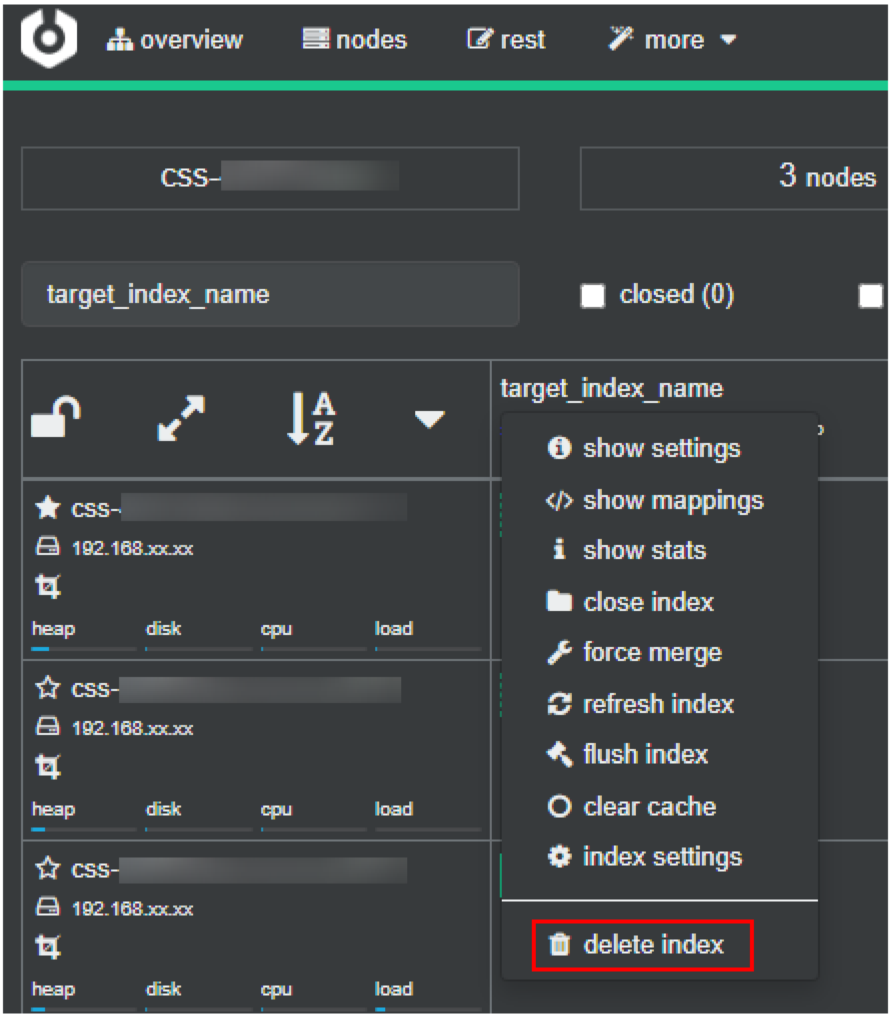
Before deleting index data, carefully evaluate any potential impact on services.

- 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 and run the **DELETE / Index name** command in Dev Tools.
  - 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 4-5 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**.

NOTE

Only Elasticsearch 7.6.2, Elasticsearch 7.10.2, and OpenSearch 1.3.6 clusters support the vector search engine of CSS.

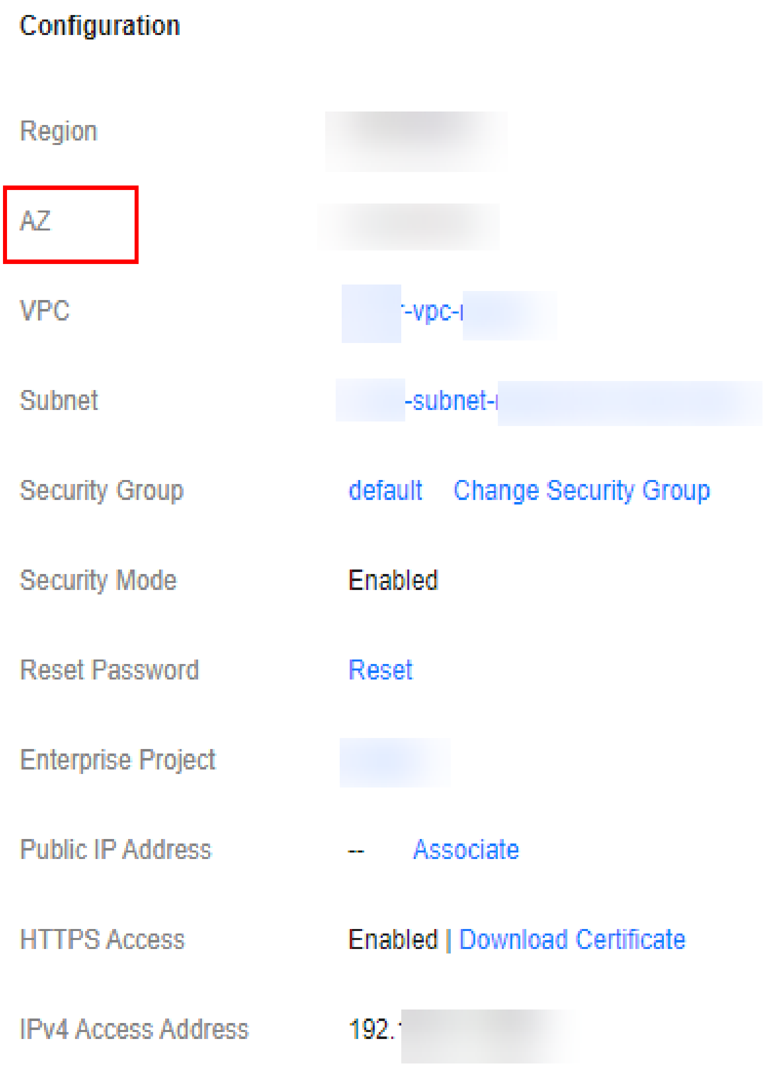
5 Managing CSS Clusters

5.1 How Do I Check the AZ Where a CSS Cluster Is Located?

You view the AZ where a cluster is located on the **Cluster Information** page.

1. Log in to the CSS management console.
2. Choose **Clusters** > **Elasticsearch**. The cluster list is displayed.
3. Click the cluster name to go to the **Cluster Information** page. In the **Configuration** area, view the **AZ** where the cluster is located.

Figure 5-1 Cluster configuration



Configuration

Region	[redacted]
AZ	[redacted]
VPC	[redacted]-vpc-[redacted]
Subnet	[redacted]-subnet-[redacted]
Security Group	default Change Security Group
Security Mode	Enabled
Reset Password	Reset
Enterprise Project	[redacted]
Public IP Address	-- Associate
HTTPS Access	Enabled | Download Certificate
IPv4 Access Address	192.[redacted]

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

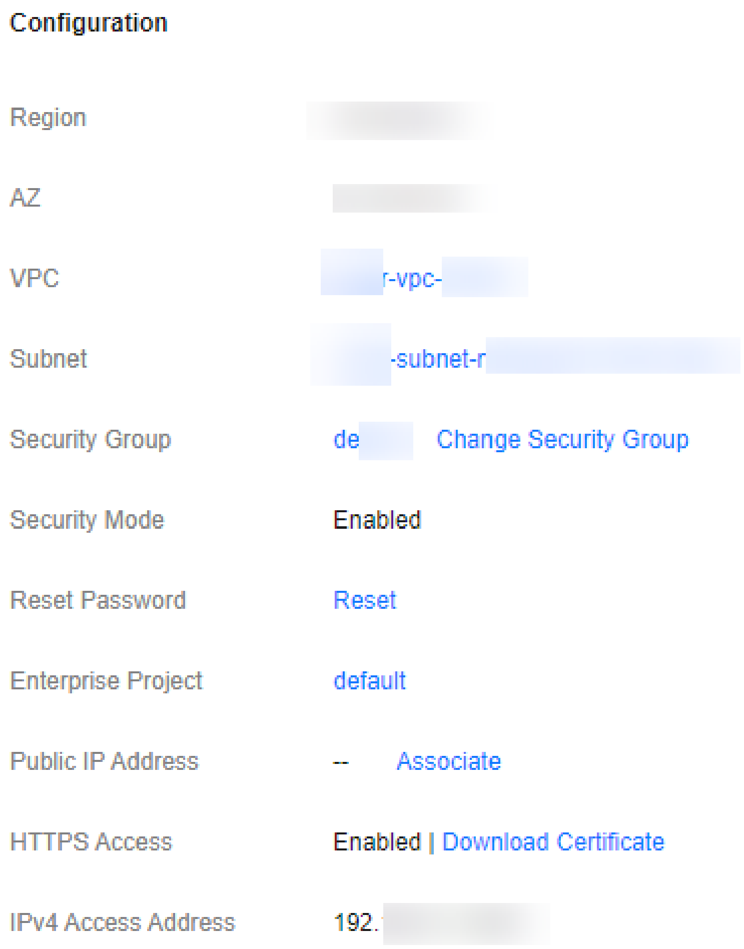
- Clusters in non-security mode: no restrictions.
- Clusters in security mode: The Filebeat OSS version must match the cluster version. For details on how to download the Filebeat OSS version, see **Past Releases of Elastic Stack Software**.

5.3 How Do I Obtain the Security Certificate of CSS?

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

1. Log in to the CSS management console.
2. In the navigation pane, choose **Clusters**. The cluster list is displayed.
3. Click the name of a cluster to go to the cluster details page.
4. On the **Configuration** page, click **Download Certificate** next to **HTTPS Access**.

Figure 5-2 Downloading a certificate



Configuration

Region	[redacted]
AZ	[redacted]
VPC	[redacted]r-vpc-[redacted]
Subnet	[redacted]-subnet-r[redacted]
Security Group	de[redacted] Change Security Group
Security Mode	Enabled
Reset Password	Reset
Enterprise Project	default
Public IP Address	-- Associate
HTTPS Access	Enabled | Download Certificate
IPv4 Access Address	192.[redacted]

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 Modify the Security Group for Elasticsearch and OpenSearch Clusters in CSS?

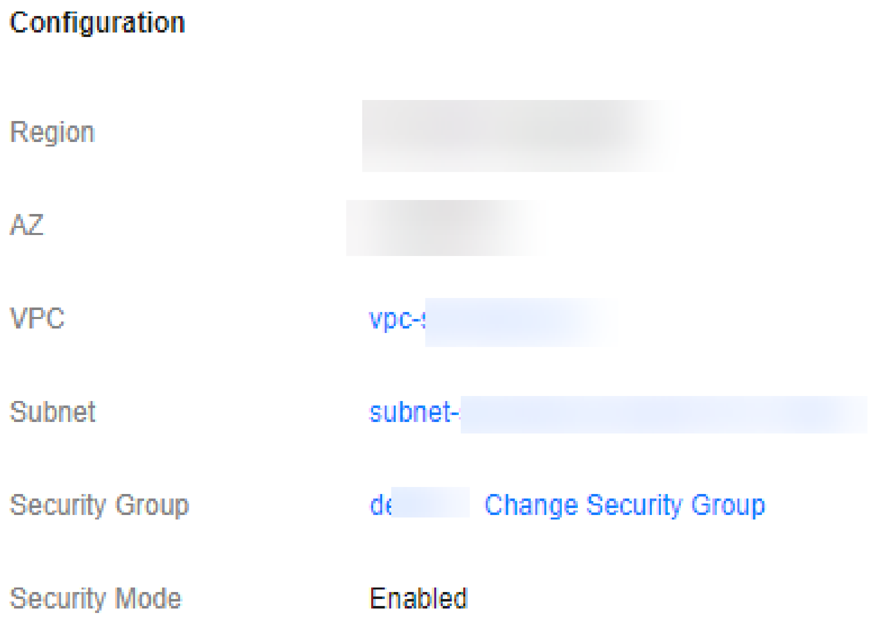
After a cluster is created, you can modify its security group.

NOTICE

- Before changing the security group, ensure that the port 9200 required for service access has been enabled. Incorrect security group configuration may cause service access failures. Exercise caution when performing this operation.
- You are advised to perform this operation during off-peak hours.
- The security group of a cluster created before February 2023 cannot be modified. You are advised to modify the security group of the cluster after **Migrating Data Through Backup and Restoration (from CSS Elasticsearch)** to a new cluster.

1. Log in to the CSS management console.
2. In the navigation pane, choose **Clusters**. The cluster list is displayed.
3. Click the name of a cluster to go to the cluster details page.
4. On the right of **Security Group**, click **Change Security Group**.

Figure 5-3 Changing a security group



Configuration

Region

AZ

VPC vpc-

Subnet subnet-

Security Group de Change Security Group

Security Mode Enabled

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

5.6 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
    }
}
```

5.7 Can I Modify the TLS Algorithm of an Elasticsearch or OpenSearch Cluster in CSS?

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, choose **Clusters**. The cluster list is displayed.
3. Click the name of the target cluster to go to the cluster details page.
4. Select **Parameter Configurations**, click **Edit**, expand the **Customize** parameter, 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']**.

   NOTE

   If the parameter value contains multiple algorithm protocols, enclose the value with a pair of square brackets ([]). If the parameter value is a single algorithm protocol, enclose the value with a pair of single quotation marks(' ').

5. 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.

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

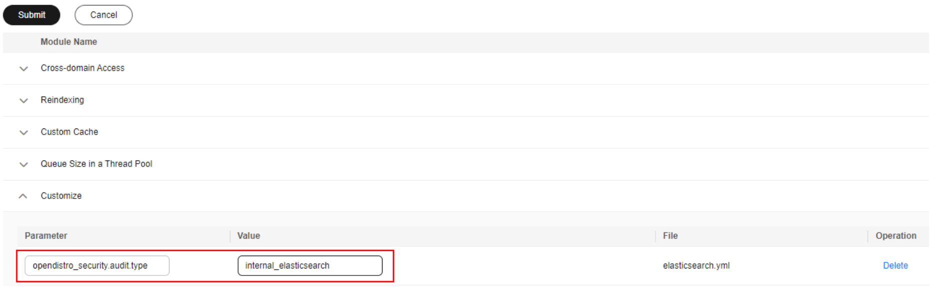
5.8 How Do I Enable Audit Logs for an Elasticsearch or OpenSearch Cluster of CSS?

Audit logs are disabled for Elasticsearch clusters by default.

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

1. Log in to the CSS management console.
2. In the navigation pane, choose **Clusters**. The cluster list is displayed.
3. Click the name of the target cluster to go to the cluster details page.
4. In the navigation pane on the left, choose **Parameter Configurations**. Click **Edit**, expand the **Customize** parameter, 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**.

Figure 5-4 Configuring a custom parameter



Parameter	Value	File	Operation
opendistro_security.audit.type	internal_elasticsearch	elasticsearch.yml	Delete

5. 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.

6. Return to the cluster list and choose **More** > **Restart** in the **Operation** column to restart the cluster and make the change take effect.
7. After the cluster is restarted, click **Access Kibana** in the **Operation** column. On the displayed page, enter the username and password. The **Dev Tools** page is displayed.
8. In the **Console** page, run the **GET _cat/indices?v** command. If there are indexes related to **.*audit***, the audit log function is enabled.

5.9 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
  }
  ```

5.10 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**

  Fees may be incurred when you store indexes in OBS. For details, see the price of standard single-AZ storage in **OBS Price Calculator**.
- **Why can frozen indexes stored in OBS still be queried using 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.

5.11 How Do I Check the List of Default Plugins for Elasticsearch and OpenSearch Clusters?

Default plugins are available for the Elasticsearch and OpenSearch clusters in CSS. You can check the default plugins on the CSS web console or on Kibana or OpenSearch Dashboards.

Checking Plugins on the CSS Console

1. Log in to the CSS management console.
2. In the navigation pane, choose **Clusters**. Click the target cluster name and go to the **Cluster Information** page.
3. Click the **Plugins** tab.
4. On the **Default** page, check the default plugins supported by the current version.

Checking Plugins on Kibana or OpenSearch Dashboards

1. Log in to the CSS management console.
2. In the navigation pane, choose **Clusters**. Locate the target cluster and click **Access Kibana** in the **Operation** column to log in to Kibana (for Elasticsearch) or OpenSearch Dashboards (for OpenSearch).
   - For a non-security mode cluster: The Kibana or OpenSearch Dashboards console is displayed.
   - For a security-mode cluster: Enter the username and password on the login page and click **Log In** to go to the Kibana or OpenSearch Dashboards console. The default username is **admin** and the password is the administrator password you specified during cluster creation.
3. Go to **Dev Tools** and run the following command to view the cluster plugin information:

   ```
   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        7.6.2-xxxx-ei-css-v1.0.1
   css-test-ess-esn-1-1 analysis-icu                    7.6.2-xxxx-ei-css-v1.1.6
   css-test-ess-esn-1-1 analysis-ik                     7.6.2-xxxx-ei-css-v1.0.1
   ......
   ```

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

6 CSS Cluster Backup and Restoration

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

Prerequisites

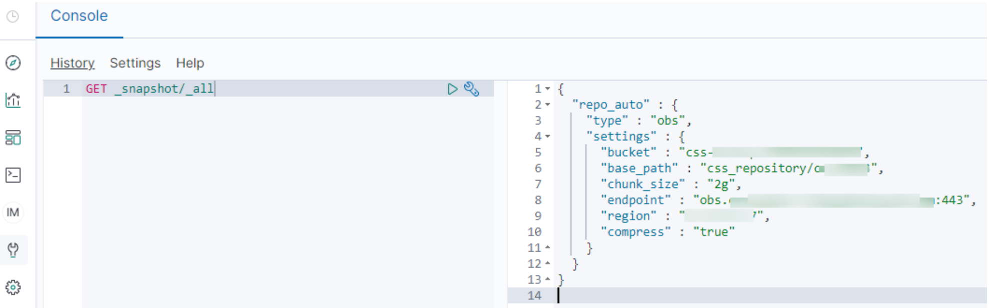
The snapshot function has been enabled for the cluster and snapshot information has been configured.

Querying a Snapshot

1. Log in to the CSS management console, and click **Clusters** in the navigation pane. On the displayed **Clusters** page, locate the target cluster and click **Access Kibana** in the **Operation** column.
2. In the left navigation pane of the Kibana page, 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. Run the **GET _snapshot/_all** command to query information about all repositories.

   Figure 6-1 Querying information about all repositories

   ```
   GET _snapshot/_all
   ```

   ```
   {
     "repo_auto" : {
       "type" : "obs",
       "settings" : {
         "bucket" : "css-[redacted]",
         "base_path" : "css_repository/c[redacted]",
         "chunk_size" : "2g",
         "endpoint" : "obs.[redacted]:443",
         "region" : "[redacted]",
         "compress" : "true"
       }
     }
   }
   ```

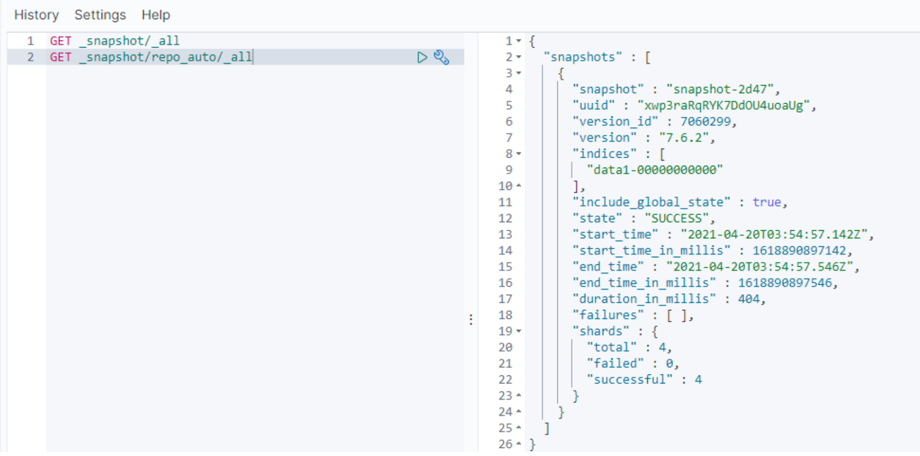
   - **bucket**: OBS bucket name

   - **base_path**: Path. It consists of a fixed prefix and a cluster name.
   - **endpoint**: OBS domain name
   - **region**: your region

4. Query snapshot information.

   a. Run the **GET _snapshot/repo_auto/_all** command to query the list of all the snapshots in the current repository.

      Figure 6-2 Snapshot information



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

```
{
  "snapshots" : [
    {
      "snapshot" : "snapshot-2d47",
      "uuid" : "xwp3raRqRYK7DdOU4uoaUg",
      "version_id" : 7060299,
      "version" : "7.6.2",
      "indices" : [
        "data1-00000000000"
      ],
      "include_global_state" : true,
      "state" : "SUCCESS",
      "start_time" : "2021-04-20T03:54:57.142Z",
      "start_time_in_millis" : 1618890897142,
      "end_time" : "2021-04-20T03:54:57.546Z",
      "end_time_in_millis" : 1618890897546,
      "duration_in_millis" : 404,
      "failures" : [ ],
      "shards" : {
        "total" : 4,
        "failed" : 0,
        "successful" : 4
      }
    }
  ]
}
```

      - **snapshot**: snapshot name
      - **state**: snapshot status
      - **start_time**, **start_time_in_millis**, **end_time**, and **end_time_in_millis**: snapshot time
      - **shards**: the number of shards. **total** indicates the total number of shards. **failed** indicates the number of failures. **successful** indicates the number of successes.

   b. Run the **GET _snapshot/repo_auto/$snapshot-xxx** command to query information about a specified snapshot.

      - Replace **$snapshot-xxx** with the actual snapshot name.
      - **repo_auto** is followed by a snapshot name or wildcard characters.

5. (Optional) Delete information about a specified snapshot.

   To delete a specific snapshot, run the **DELETE _snapshot/ repo_auto/ $snapshot-xxx** command.

   Replace **$snapshot-xxx** with the actual snapshot name.

6.2 Can a Deleted CSS Cluster Be Restored?

Yes. You can use a snapshot stored in OBS to restore a cluster. A deleted cluster that has no snapshots in OBS cannot be restored. Exercise caution when deleting 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. Click **Create Cluster** in the upper right corner to create a cluster. During the cluster creation, disable the cluster snapshot function. After the cluster is created, enable the cluster snapshot function.

   NOTICE

   To restore a deleted cluster to a new cluster, ensure they are in the same region. The new cluster version must be the same as or later than that of the deleted cluster. The quantity of nodes in the new cluster must be greater than half of that in the deleted cluster. Otherwise, the cluster may fail to be restored.

3. If the status of the new cluster changes to **Available**, click the cluster name to go to the **Cluster Information** page.
4. In the navigation pane on the left, choose **Cluster Snapshots**. Enable the cluster snapshot function. Set the OBS bucket and backup path to those set for the cluster that needs to be restored.

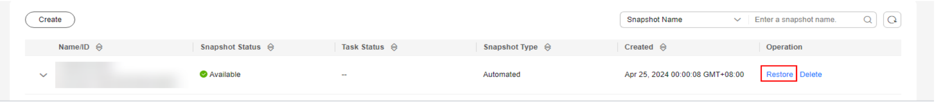
   After the configuration is saved, you can view the snapshot information of the deleted cluster in the snapshot management list **a few minutes later**. **If the snapshot is not displayed, edit the basic snapshot configuration again, change the backup path to another one and then to the correct one, save the modification, and try again.**

   NOTE

   To restore the data of a deleted cluster to an existing cluster, set the OBS bucket and backup path to those of the deleted cluster.

5. Locate the target snapshot and click **Restore** in the **Operation** column. The **Restore** page is displayed.

   Figure 6-3 Selecting a snapshot

Name/ID	Snapshot Status	Task Status	Snapshot Type	Created	Operation
[redacted]	Available	--	Automated	Apr 25, 2024 00:00:08 GMT+08:00	Restore Delete

6. On the **Restore** page, set restoration parameters.

Table 6-1 Restoration settings

Parameter	Description
Index	Enter the name of the index you want to restore. If you do not specify any index name, the data of all indexes will be restored. This parameter is left blank by default. The value can contain 0 to 1024 characters, and cannot contain spaces, uppercase letters, or the following special characters: **"\<|>/?**. You can use the asterisk (*) to match multiple indexes. For example, **index*** indicates that all indexes with the prefix **index** will be restored.
Rename Pattern	Enter a regular expression. Indexes that match the regular expression will be restored. The default value **index_(.+)** indicates all indexes. The value can contain 0 to 1024 characters, and cannot contain spaces, uppercase letters, or the following special characters: **"\<|>/?,**. **NOTE** The **Rename Pattern** and **Rename Replacement** take effect only when they are both configured at the same time.
Rename Replacement	Rule for index renaming. The default value **restored_index_$1** indicates that **restored_** will be added to the beginning of the names of all restored indexes. The value can contain 0 to 1024 characters, and cannot contain spaces, uppercase letters, or the following special characters: **"\<|>/?,**. **NOTE** The **Rename Pattern** and **Rename Replacement** take effect only when they are both configured at the same time.

Parameter	Description

Cluster	Select the cluster where you want to restore the data to. You can select the current cluster or another.
	● You must specify a cluster whose status is **Available**. If the status of the current cluster is **Unavailable**, you cannot restore the snapshot to the current cluster.
	● When restoring data to another cluster, make sure the version of the destination cluster is not earlier than the current cluster. After a frozen index is restored using a snapshot, the index is stored on the local node and is read-only by default. To make it writable, run the following command to configure the index:
	`PUT index_name/_settings`
	`{`
	`  "index.blocks.write": null`
	`}`
	● Select or deselect **Overwrite same-name indexes in the destination cluster**. Data restoration using snapshots works by overwriting existing snapshot files. When there are same-name indexes in the destination cluster, you need to select this option in order to restore same-name, same-shard structure indexes. Indexes with a different shard structure cannot be restored. Exercise caution when performing this operation.

7. Click **OK** to get started. If restoration succeeds, **Task Status** of the snapshot in the snapshot list will change to **Restoration succeeded**, and the index data is re-generated based on the snapshot.

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

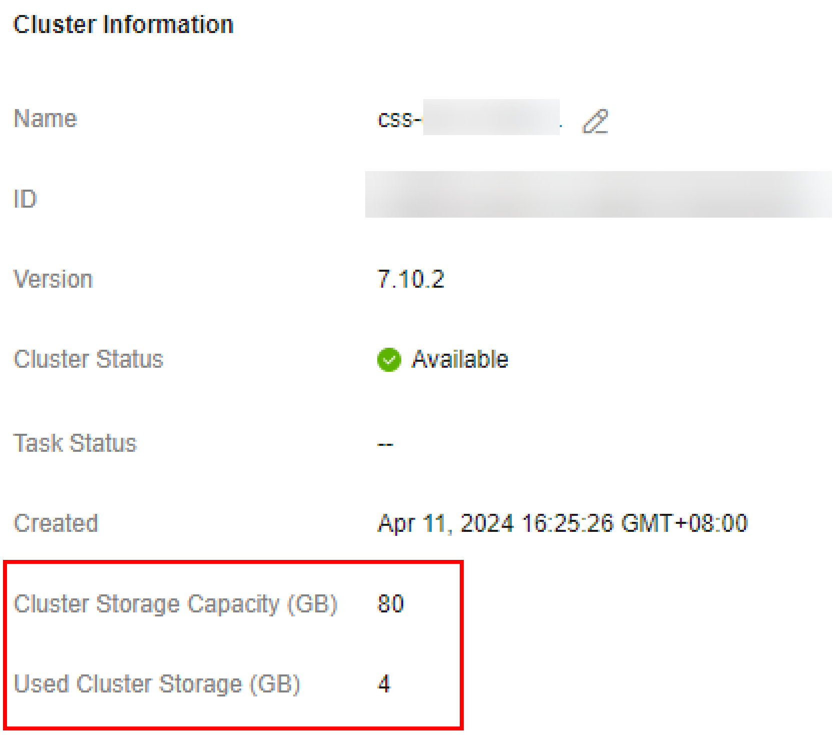
You can monitor the cluster memory usage by checking the maximum JVM heap usage and average JVM heap usage.

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

You can view the disk usage of a cluster on the **Cluster Information** page.

1. Log in to the CSS management console.
2. Choose **Clusters** > **Elasticsearch**. The cluster list is displayed.
3. Click the cluster name to go to the **Cluster Information** page. In the **Cluster Information** area, the value of **Cluster Storage Capacity (GB)**/**Used Cluster Storage (GB)** indicates the cluster disk usage.

Figure 7-1 Cluster information



Cluster Information

Name	css-[redacted]
ID	[redacted]
Version	7.10.2
Cluster Status	✓ Available
Task Status	--
Created	Apr 11, 2024 16:25:26 GMT+08:00
Cluster Storage Capacity (GB)	80
Used Cluster Storage (GB)	4

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.