

Cloud Search Service

FAQs

Issue 01
Date 2022-08-02



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2022. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

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.

Contents

1 General Consulting	1
1.1 What Are Regions and AZs?	1
1.2 How Does CSS Ensure Data and Service Security?	2
1.3 Which CSS Metrics Should I Focus On?	2
1.4 What Storage Options Does CSS Provide?	3
1.5 What Is the Maximum Storage Capacity of CSS?	3
1.6 How Can I Manage CSS?	3
1.7 Which Elasticsearch Versions Does CSS Support?	3
1.8 How Can I Access CSS?	3
1.9 What Can the Disk Space of a CSS Cluster Be Used For?	4
1.10 How Do I Check the Numbers of Shards and Replicas in a Cluster on the CSS Console?	4
1.11 Can I Modify the TLS Algorithm of an Elasticsearch Cluster?	5
1.12 What VM Specifications Does CSS Support?	5
1.13 Can I Use the CSS Cluster Data as a Data Source?	5
1.14 What Data Compression Algorithms Does CSS Use?	5
2 Functions	7
2.1 How Do I Use Elasticsearch Custom Scoring to Query Data?	7
2.2 Can I Migrate CSS Data to Other Tenants?	11
2.3 How Do I Configure the Threshold for CSS Slow Query Logs?	12
2.4 How Do I Update the CSS Lifecycle Policy?	12
2.5 How Do I Set the Numbers of Index Copies to 0 in Batches?	14
2.6 How Do I Specify Multiple Indexes When Creating Snapshots?	15
2.7 How Can I Find All Snapshots in an OBS Bucket When I Create a Snapshot for a CSS Cluster?	15
2.8 Does CSS Create Full or Incremental Snapshots for Clusters?	15
2.9 If a CSS Cluster Snapshot Is Restored to Another Cluster, Will the Indexes of the Destination Cluster Be Overwritten?	15
2.10 How Do I Query Snapshot Information?	15
2.11 Can I Restore a Deleted Cluster?	17
3 Clusters in Security Mode	19
3.1 What Is the Relationship Between the Filebeat Version and Cluster Version?	19
3.2 Can I Enable Security Mode Later to Add Authentication and Use SSL Communication If I Did Not Enable It When I Initially Purchased a CSS Cluster?	19
3.3 Can I Change the HTTPS Access Mode for a CSS Cluster?	19

3.4 How Do I Obtain a Cluster Security Certificate?.....	19
4 Resource Usage and Change.....	21
4.1 How Do I Clear Expired Data to Release Storage Space?.....	21
4.2 How Do I Configure a Two-Replica CSS Cluster?.....	21
4.3 How Do I Delete Index Resources?.....	22
4.4 Can I Change the Number of Shards to Four with Two Replicas When There Is One Shard Set in the JSON File?.....	22
5 Components.....	23
5.1 Can I Install Search Guard on CSS?.....	23
6 Kibana.....	24
6.1 What Are the Username and Password for Logging in to Kibana and Cerebro?.....	24
6.2 Can I Limit the Time Range When Checking Elasticsearch Data in Kibana?.....	24
6.3 Can I Export Data from Kibana?.....	24
7 Clusters.....	26
7.1 Why Does My ECS Fail to Connect to a Cluster?.....	26
7.2 Which Search Functions Does CSS Support?.....	27
7.3 Can a New Cluster Use the IP Address of the Old Cluster?.....	27
7.4 Can I Use x-pack-sql-jdbc to Access CSS Clusters and Query Data?.....	27
8 Ports.....	28
8.1 Do Ports 9200 and 9300 Both Open?.....	28
9 Change History.....	29

1 General Consulting

1.1 What Are Regions and AZs?

Regions and AZs

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- Regions are divided based on geographical location and network latency. Services, such as ECS, EVS, OBS, VPC, Elastic IP (EIP), and Image Management Service (IMS), can be shared within the same region. There are two types of regions: universal and dedicated. A universal region provides cloud services for all tenants. A dedicated region only provides a certain type of services or only provides services for specific tenants.
- An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-control, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters.

Region Selection

You are advised to select a region close to you or your target users. This reduces network latency and improves the access success rate.

AZ Selection

When determining whether to deploy resources in the same AZ, consider your application's requirements for disaster recovery (DR) and network latency.

- To prioritize DR capabilities, deploy resources in different AZs in the same region.
- To prioritize network latency, deploy resources in the same AZ.

Regions and Endpoints

Before using an API to call resources, you will need to specify the resource region and endpoint. For details, see "Endpoints" in *Cloud Search Service API Reference*.

1.2 How Does CSS Ensure Data and Service 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
Multiple replicas, cross-AZ deployment of clusters, and third-party (OBS) backup of index data ensure the security of user data.

1.3 Which CSS Metrics Should I Focus On?

Disk usage and cluster health status are two key metrics that you can focus on. You can log in to Cloud Eye and configure alarm rules for these metrics. If alarms are reported, handle them by taking appropriate measures.

Configuration examples:

- Alarms are reported if the disk usage is higher than or equal to a specified value (for example, 85%) and has reached this value multiple times (for example, 5 times) within a specified time period (for example, 5 minutes).
- Alarms are reported if the value of the cluster health status metric exceeds 0 for multiple times (for example, 5 times) within a specified time period (for example, 5 minutes).

Measures:

- If disk usage alarms are reported, view available disk space, check whether data can be deleted from cluster nodes or archived to other systems to free up space, or check if you can expand the disk capacity.

- If cluster health status alarms are reported, check whether shard allocation is normal, whether shards have been lost, and check whether the process has been restarted on Cerebro.

1.4 What Storage Options Does CSS Provide?

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).

- Supported 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 selected when you create a cluster.

1.5 What Is the Maximum Storage Capacity of CSS?

You can configure up to 200 nodes for a cluster (each node corresponds to 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 CSS based on the sizes of EVS disks attached to different ECSs. The EVS disk size is determined by the node specifications selected when you create the cluster.

1.6 How Can I Manage CSS?

You can use any of the following three methods to manage CSS or to use search engine APIs. You can initiate requests based on constructed request messages.

- curl
curl is a command-line tool used to transfer data to or from a given URL. It serves as an HTTP client that can send HTTP requests to the HTTP server and receive response messages. You can also use curl to debug APIs. For more information about curl, visit <https://curl.haxx.se/>.
- Code
You can call APIs through code to assemble, send, and process request messages.
- REST client
Both Mozilla Firefox and Google Chrome provide a graphical browser plugin, the REST client, which you can use to send and process requests.
 - For Mozilla Firefox, see [Firefox REST Client](#).
 - For Google Chrome, see [Postman](#).

1.7 Which Elasticsearch Versions Does CSS Support?

In CSS, Elasticsearch 7.6.2, and 7.9.3 and Kibana 7.6.2, and 7.9.3 are supported.

1.8 How Can I Access CSS?

- RESTful API

- Transport Client
- Rest Client

If you use Transport Client to access the cloud service, ensure that versions of both the client and server are the same.

Transport Client is not recommended. You are advised to use High Level Rest Client.

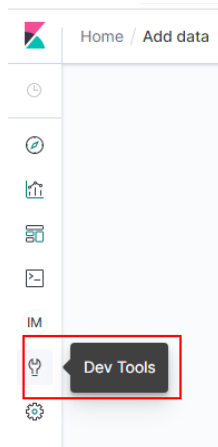
1.9 What Can the Disk Space of a CSS Cluster Be Used For?

You can store the following logs and files:

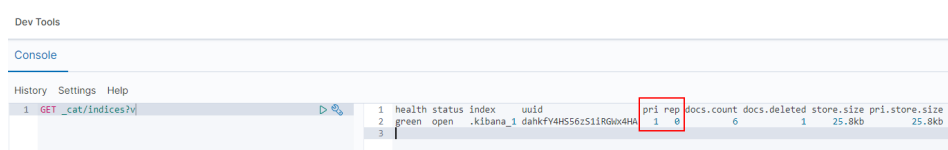
- Log files: Elasticsearch logs
- Data files: Elasticsearch index files
- Other files: cluster configuration files
- OS: 5% storage space reserved for the OS by default

1.10 How Do I Check the Numbers of Shards and Replicas in a Cluster on the CSS Console?

1. Log in to the 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.



1.11 Can I Modify the TLS Algorithm of an Elasticsearch Cluster?

You can modify TLS algorithms in CSS 7.6.2 and later versions.

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. Click the **Parameter Configurations** tab, click to expand the **Customize** area. 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']`.

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. click **Add** in the **Operation** column.
6. After the modification 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 modification list, the modification has been saved.
7. Return to the cluster list and choose **More > Restart** in the **Operation** column to restart the cluster and make the modification take effect.

1.12 What VM Specifications Does CSS Support?

Supported VM specifications vary depending on regions and versions. You can log in to the CSS console and check VM specifications on the **Create Cluster** page.

1.13 Can I Use the CSS Cluster Data as a Data Source?

Yes, CSS is fully compatible with all open-source Elasticsearch APIs.

1.14 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 of 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.

2 Functions

2.1 How Do I Use Elasticsearch Custom Scoring to Query Data?

Perform the following procedure to use the custom scoring function of Elasticsearch to query data.

Procedure

1. Log in to the CSS management console.
2. In the left navigation pane, click **Clusters** to switch to the **Clusters** page.
3. In the cluster list, locate the row that contains the target cluster and click **Access Kibana** in the **Operation** column.
4. In the left navigation pane of Kibana, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.
5. Run commands, as shown in the examples below, to create an index and specify a user-defined mapping to define the data type:

For example, the **tv.json** data file is available and contains the following data:

```
{
  "tv": [
    { "name": "tv1", "description": "USB, DisplayPort", "vote": 0.98 }
    { "name": "tv2", "description": "USB, HDMI", "vote": 0.99 }
    { "name": "tv3", "description": "USB", "vote": 0.5 }
    { "name": "tv4", "description": "USB, HDMI, DisplayPort", "vote": 0.7 }
  ]
}
```

Run the following command to create the **mall** index and specify the user-defined mapping to define the data type:

```
PUT /mall?pretty
{
  "mappings": {
    "tv": {
      "properties": {
        "description": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword"
            }
          }
        }
      }
    }
  }
}
```

```
}
}
},
"name": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword"
    }
  }
},
"vote": {
  "type": "float"
}
}
}
}
```

6. Run the following command to import data in the **tv.json** file to the **mall** index:

```
POST /mall/tv/_bulk?pretty
{ "index": {"_id": "1"}}
{ "name": "tv1", "description": "USB, DisplayPort", "vote": 0.98 }
{ "index": {"_id": "2"}}
{ "name": "tv2", "description": "USB, HDMI", "vote": 0.99 }
{ "index": {"_id": "3"}}
{ "name": "tv3", "description": "USB", "vote": 0.5 }
{ "index": {"_id": "4"}}
{ "name": "tv4", "description": "USB, HDMI, DisplayPort", "vote": 0.7 }
```

7. Query data by using custom scoring.

- The total score of each TV is computed based on the absolute praise rate and the queried products are then sorted based on their total score, from the highest to lowest.

The following sample code illustrates how to query TVs with USB, HDMI, or DisplayPort port. In this example, they get a score 0 for a TV if it has none of the three ports, 1 if it has one, 2 if it has two, and 3 if it has all three. Then, the obtained scores are multiplied by the TV's absolute praise rate to obtain their total scores. Finally, the TVs are sorted in an order of their total scores from the highest to lowest.

```
GET /mall/tv/_search?pretty
{
  "query":{
    "function_score":{
      "query":{
        "bool":{
          "should":[
            {"constant_score":{
              "query":{"match":{"description":"USB"}}
            }},
            {"constant_score":{
              "query":{"match":{"description":"HDMI"}}
            }},
            {"constant_score":{
              "query":{"match":{"description":"DisplayPort"}}
            }
          ]
        }
      }
    },
    "field_value_factor":{
      "field":"vote",
      "factor":1
    },
    "boost_mode":"multiply",
    "max_boost":10
  }
```

```
}
}
```

In the preceding example, the total score is calculated as follows:
 $new_score = query_score \times (factor \times vote)$

The command output is similar to the following:

```
{
  "took": 13,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 4,
    "max_score": 2.1,
    "hits": [
      {
        "_index": "mall",
        "_type": "tv",
        "_id": "4",
        "_score": 2.1,
        "_source": {
          "name": "tv4",
          "description": "USB, HDMI, DisplayPort",
          "vote": 0.7
        }
      },
      {
        "_index": "mall",
        "_type": "tv",
        "_id": "2",
        "_score": 1.98,
        "_source": {
          "name": "tv2",
          "description": "USB, HDMI",
          "vote": 0.99
        }
      },
      {
        "_index": "mall",
        "_type": "tv",
        "_id": "1",
        "_score": 1.96,
        "_source": {
          "name": "tv1",
          "description": "USB, DisplayPort",
          "vote": 0.98
        }
      },
      {
        "_index": "mall",
        "_type": "tv",
        "_id": "3",
        "_score": 0.5,
        "_source": {
          "name": "tv3",
          "description": "USB",
          "vote": 0.5
        }
      }
    ]
  }
}
```

The preceding command output shows that Elasticsearch computes the total score of each TV based on the absolute praise rate and then sorts the TVs in an order of their total score, from the highest to lowest.

- The total score of each TV is computed based on the relative praise rate and the queried products are then sorted based on their total score, from the highest to lowest.

The following sample code illustrates how to query TVs with USB, HDMI, or DisplayPort port. In this example, they get a score 0 for a TV if it has none of the three ports, 1 if it has one, 2 if it has two, and 3 if it has all three. Then, the obtained scores are multiplied by the TV's relative praise rate to obtain their total scores. The relative praise rate is calculated as follows: Specify a praise rate threshold like 0.8. If a TV's absolute praise rate is higher than 0.8, consider the praise rate as 1 in the calculation formula for the TV. Otherwise, consider the praise rate as 0.5. Finally, the TVs are sorted in an order of their total scores, from the highest to lowest.

```
GET /mall/tv/_search?pretty
{
  "query":{
    "function_score":{
      "query":{
        "bool":{
          "should":[
            {"constant_score":{
              "query":{"match":{"description":"USB"}}
            }},
            {"constant_score":{
              "query":{"match":{"description":"HDMI"}}
            }},
            {"constant_score":{
              "query":{"match":{"description":"DisplayPort"}}
            }}
          ]
        }
      },
      "script_score":{
        "script":{
          "params":{"threshold":0.8},
          "inline":"if (doc[\"vote\"].value > params.threshold) {return 1;} return 0.5;"
        }
      },
      "boost_mode":"multiply",
      "max_boost":10
    }
  }
}
```

In the preceding example, the total score is calculated as follows:
 $\text{new_score} = \text{query_score} \times \text{vote}$ (If the value of **vote** is greater than 0.8, round it off to 1 in the formula. If the value of **vote** is 0.8 or less, consider its value as 0.5 in the formula.)

The command output is similar to the following:

```
{
  "took": 634,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 4,
```

```
"max_score": 2,
"hits": [
  {
    "_index": "mall",
    "_type": "tv",
    "_id": "2",
    "_score": 2,
    "_source": {
      "name": "tv2",
      "description": "USB, HDMI",
      "vote": 0.99
    }
  },
  {
    "_index": "mall",
    "_type": "tv",
    "_id": "1",
    "_score": 2,
    "_source": {
      "name": "tv1",
      "description": "USB, DisplayPort",
      "vote": 0.98
    }
  },
  {
    "_index": "mall",
    "_type": "tv",
    "_id": "4",
    "_score": 1.5,
    "_source": {
      "name": "tv4",
      "description": "USB, HDMI, DisplayPort",
      "vote": 0.7
    }
  },
  {
    "_index": "mall",
    "_type": "tv",
    "_id": "3",
    "_score": 0.5,
    "_source": {
      "name": "tv3",
      "description": "USB",
      "vote": 0.5
    }
  }
]
```

The preceding command output shows that Elasticsearch computes the total score of each TV based on the relative praise rate and then sorts the TVs based on their total score, from the highest to lowest.

2.2 Can I Migrate CSS Data to Other Tenants?

No. Data can not be migrated between tenants.

2.3 How Do I Configure the Threshold for CSS Slow Query Logs?

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>

2.4 How Do I Update the CSS Lifecycle Policy?

The CSS lifecycle 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 2-1 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 2-2 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 2-3 Initialized data

```

{
  "_index": ".opendistro-ism-config",
  "_type": "_doc",
  "_id": "FABkSF5G5TCmR0QkH41HWw",
  "_score": 1.0,
  "_source": {
    "managed_index": {
      "name": "data1",
      "enabled": true,
      "index": "data1",
      "index_uuid": "FABkSF5G5TCmR0QkH41HWw",
      "schedule": {
        "interval": {
          "start_time": 1641432652693,
          "period": 1,
          "unit": "Minutes"
        }
      },
      "last_updated_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": 1641432158329,
        "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"
}

```

2.5 How Do I Set the Numbers of Index Copies to 0 in Batches?

1. Log in to the Kibana page of the cluster. In the navigation pane, choose **Dev Tools**.
2. Modify and run the **PUT /*/_settings{"number_of_replicas":0}** command.

NOTE

Do not directly run the preceding command, because the asterisk (*) may match security indexes. You are advised to specify the index required for the batch operation. Example: **PUT /test/*/_settings{"number_of_replicas":0}**

2.6 How Do I Specify Multiple Indexes When Creating Snapshots?

Index length is limited for snapshot creation. To specify multiple indexes, you can create multiple snapshots.

2.7 How Can I Find All Snapshots in an OBS Bucket When I Create a Snapshot for a CSS Cluster?

Each time a new snapshot is created for a CSS cluster, new backup data is added incrementally based on the original snapshot. **index-*x*** records the information about each snapshot. **snap-*xxxx*.dat** and **meta-*xxxx*.dat** record the metadata of the snapshot. **indices** stores the index data.

2.8 Does CSS Create Full or Incremental Snapshots for Clusters?

The first snapshot of a cluster is a full snapshot, and subsequent snapshots are incremental snapshots. CSS snapshot files depend on each other.

2.9 If a CSS Cluster Snapshot Is Restored to Another Cluster, Will the Indexes of the Destination Cluster Be Overwritten?

If the snapshot and the destination cluster have indexes using the same name, such indexes in the destination cluster will be overwritten by those in the snapshot.

NOTE

If the snapshot and the destination cluster use different shards, the indexes with the same name will not be overwritten.

2.10 How Do I Query Snapshot Information?


Prerequisites

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

Querying a Snapshot

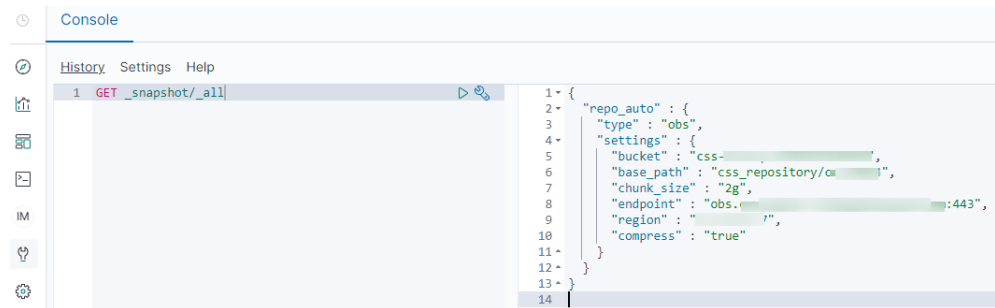
1. Log in to the CSS management console, and click **Clusters** in the navigation pane. On the displayed **Clusters** page, locate the target cluster and click **Access Kibana** in the **Operation** column.

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.

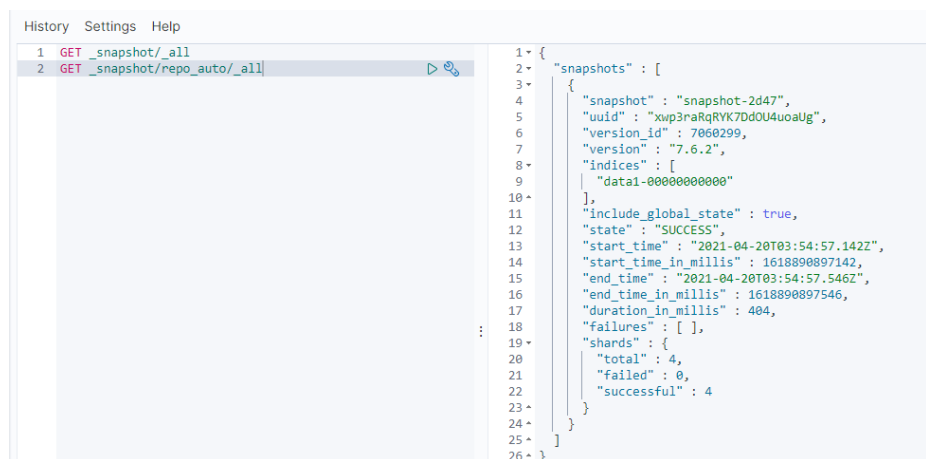
3. Run the **GET _snapshot/_all** command to query information about all repositories.

Figure 2-4 Querying information about all repositories



- **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 2-5 Snapshot information



- **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.

2.11 Can I Restore a Deleted Cluster?

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, perform the following steps:

1. Log in to the CSS management console.
2. In the navigation pane on the left, choose **Clusters**. On the displayed **Clusters** page, click **Create Cluster** in the upper right corner to create a cluster and enable the snapshot function. Set the OBS bucket and backup path to those of the cluster to be restored.

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

In the snapshot management list, you can view the snapshot information of the deleted cluster. If no information is displayed, wait for several minutes and refresh the page.
5. Locate the target snapshot and click **Restore** in the **Operation** column. The **Restore** page is displayed.
6. On the **Restore** page, set restoration parameters.

Index: Enter the name of the index you want to restore. If you do not specify any index name, data of all indices will be restored. The value can contain 0 to 1,024 characters. Uppercase letters, spaces, and certain special characters (including "`<>/?`") are not allowed.

Rename Pattern: Enter a regular expression. Indices that match the regular expression are restored. The default value **index_(.+)** indicates restoring data of all indices. The value contains 0 to 1,024 characters. Uppercase letters, spaces, and certain special characters (including "`<>/?`") are not allowed.

Rename Replacement: Enter the index renaming rule. The default value **restored_index_\$1** indicates that **restored_** is added in front of the names of all restored indices. The value can contain 0 to 1,024 characters. Uppercase

letters, spaces, and certain special characters (including "<|>/?,") are not allowed. You can set **Rename Replacement** only if you have specified **Rename Pattern**.

Cluster: Select the cluster that you want to restore. You can select the current cluster or others. However, you can only restore the snapshot to clusters whose status is **Available**. If the status of the current cluster is **Unavailable**, you cannot restore the snapshot to the current cluster. If you choose to restore the snapshot to another cluster, ensure that the target cluster runs an Elasticsearch version that is not earlier than that of the current cluster. If you select another cluster and two or more indices in the cluster have the same name, data of all indices with the same name as the name you specify will be overwritten. Therefore, exercise caution when you set the parameters.

7. Click **OK**. If restoration succeeds, **Task Status** of the snapshot in the snapshot list will change to **Restoration succeeded**, and the index data is generated again according to the snapshot information.

3 Clusters in Security Mode

3.1 What Is the Relationship Between the Filebeat Version and Cluster Version?

- Non-security mode: no restrictions.
- Cluster 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](#).

3.2 Can I Enable Security Mode Later to Add Authentication and Use SSL Communication If I Did Not Enable It When I Initially Purchased a CSS Cluster?

Security mode cannot be enabled after a cluster is created. To enable security mode, you need to create a cluster in security mode. For details.

3.3 Can I Change the HTTPS Access Mode for a CSS Cluster?

After a cluster is created, its HTTPS access mode cannot be changed. You can create a cluster using the desired access mode, and migrate data to the cluster. For details.

3.4 How Do I Obtain a Cluster Security Certificate?

You can download security certificates on the cluster details page.

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. On the **Basic Information** tab, click **Download Certificate** next to **Security Mode**, as shown in the following figure.

4 Resource Usage and Change

4.1 How Do I Clear Expired Data to Release Storage Space?

- Run the following command to delete a single index data record.
`curl -XDELETE http://IP:9200/index_name`

 **NOTE**

IP: the IP address of any node in the cluster

- Run the following command to delete all Logstash data of a day. For example, delete all data from June 19, 2017:

For a cluster in non-security mode: `curl -XDELETE 'http://IP:9200/logstash-2017.06.19*'`

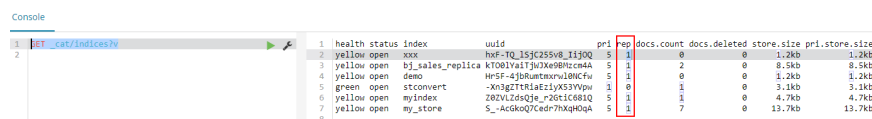
For a cluster in security mode: `curl -XDELETE -u username:password 'https://IP:9200/logstash-2017.06.19' -k`

 **NOTE**

- **username:** username of the administrator. The default value is **admin**.
- **password:** the password set during cluster creation
- **IP:** the IP address of any node in the cluster

4.2 How Do I Configure a Two-Replica CSS Cluster?

1. Run `GET _cat/indices?v` in Kibana to check the number of cluster replicas. If the value of **rep** is **1**, the cluster has two replicas.



health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
1	yellow	open xxx	hxF-TQ_15jC285v0_11j0Q	5	1	0	0	1.2kb	1.2kb
3	yellow	open bj_sales_replica	KT003vAt1j0u2Ae90tccm4	5	1	2	0	0.5kb	0.5kb
4	yellow	open demo	HrSF-dj0uuntxuru10McFu	5	1	0	0	1.2kb	1.2kb
5	green	open stconvert	-Xn3gTTRiazi1yK53Vpu	1	0	1	0	3.1kb	3.1kb
6	yellow	open myIndex	Z8VtL6oQe_r20t1c081Q	5	1	1	0	4.7kb	4.7kb
7	yellow	open mv_store	S_-acckoQ7Cedr7hXg0Qa	5	1	7	0	13.7kb	13.7kb
8									

2. If the value of **rep** is not **1**, run the following command to set the number of replicas:

`PUT /index/_settings`

```
{  
  "number_of_replicas":1 //Number of replicas  
}
```

 **NOTE**

index specifies the index name. Set this parameter based on site requirements.

4.3 How Do I Delete Index Resources?

- Manually: Run the **DELETE /my_index** command in Kibana.
- Automatically: Create scheduled tasks to call the index deletion request and periodically execute the tasks. CSS supports Open Distro Index State Management. For details, see: <https://opendistro.github.io/for-elasticsearch-docs/docs/im/ism/>

4.4 Can I Change the Number of Shards to Four with Two Replicas When There Is One Shard Set in the JSON File?

Once an index is created, the number of primary shards cannot be changed.

You can run the following command in Kibana to change the number of replicas:

```
PUT /indexname/_settings
```

```
{  
  "number_of_replicas":1 //Number of replicas  
}
```

 **NOTE**

index specifies the index name. Set this parameter based on site requirements.

5 Components

5.1 Can I Install Search Guard on CSS?

CSS does not currently support installation of Search Guard.

6 Kibana

6.1 What Are the Username and Password for Logging in to Kibana and Cerebro?

If you create a cluster in security mode, the username for logging in to Kibana and Cerebro is **admin**, and the password is the one set when you create the cluster. If you forget the password, you can reset it on the cluster details page and then log in. For details.

6.2 Can I Limit the Time Range When Checking Elasticsearch Data in Kibana?

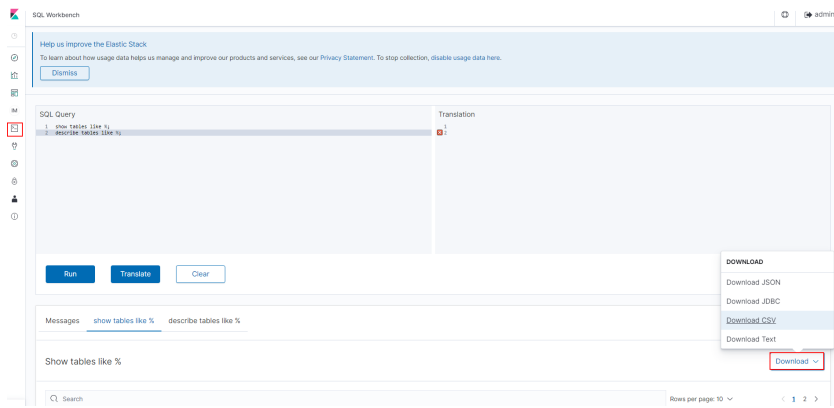
By default, Kibana queries data in the last 15 minutes. The time range can be selected but cannot be limited. You can check the information on the **Discover** page of Kibana.

6.3 Can I Export Data from Kibana?

Exporting data from Kibana requires specified plug-ins. Currently, you can only export data from Kibana 7.6.2 or later.

You can use the SQL Workbench in Kibana to export data.

Figure 6-1 SQL Workbench



7 Clusters

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

Perform the following steps to troubleshoot this problem:

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

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

NOTE

If the cluster contains multiple nodes, check whether the ECS can be connected to each node in the cluster.

- If the connection is normal, the network is running properly.
- If the connection still fails, contact technical support.

7.2 Which Search Functions Does CSS Support?

CSS supports full-text search, highlighting, facet search, near real-time search, dynamic clustering, processing of documents of various types like word and PDF, and geographic information search.

For details about the search function supported by Elasticsearch, see section "Search in Depth" in the [Elasticsearch Reference](#).

7.3 Can a New Cluster Use the IP Address of the Old Cluster?

No.

7.4 Can I Use x-pack-sql-jdbc to Access CSS Clusters and Query Data?

No. Currently, CSS does not integrate the x-pack component.

8 Ports

8.1 Do Ports 9200 and 9300 Both Open?

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 port 9300 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.

9 Change History

Release Date	Description
2022-08-02	This is the first official release.