Cloud Search Service

User Guide

Issue 12

Date 2020-08-25

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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 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 Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Contents

1 Getting Started

1.1 Getting Started with Elasticsearch

For details about the concept, advantages, functions, and application scenarios of Cloud Search Service (CSS), see the ***CSS Service Overview***.

This section provides a simple example. For details, see **Scenario Description**. You can use the Elasticsearch search engine of CSS to search for data based on the scenario example. The basic operation process is as follows:

- **Step 1: Create a Cluster**
- **Step 2: Import Data**
- **Step 3: Search for Data**
- **Step 4: Delete the Cluster**

Scenario Description

A women's clothing brand builds an e-commerce website. It uses traditional databases to provide a commodity search function for users. However, with an increase in users and business, it suffers from the slow response and low accuracy of the traditional databases. To improve user experience and avoid user loss, the e-commerce website adopts Elasticsearch to provide the commodity search function for users. This solves the issues caused by traditional databases and increases user quantity.

This section describes how to use Elasticsearch to provide the search function for users.

Assume that the e-commerce website provides the following data:

```
{
"products":[
{"productName":"Latest art shirts for women in autumn 2017","size":"L"}
{"productName":"Latest art shirts for women in autumn 2017","size":"M"}
{"productName":"Latest art shirts for women in autumn 2017","size":"S"}
{"productName":"Latest jeans for women in spring 2018","size":"M"}
{"productName":"Latest jeans for women in spring 2018","size":"S"}
{"productName":"Latest jeans for women in spring 2017","size":"L"}
{"productName":"Latest casual pants for women in spring 2017","size":"S"}
```



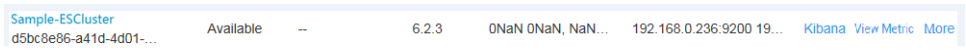
```
]
}
```

Step 1: Create a Cluster

Before searching for data, create a cluster using Elasticsearch. In this example, suppose that you create a cluster named **Sample-ESCluster**. This cluster is used only for getting started with Elasticsearch. For this cluster, you are advised to select **ess.spec-2u16g** for **Node Specifications**, **High I/O** for **Node Storage Type**, and **40 GB** for **Node Storage Capacity**. For details, see **Creating a Cluster**.

After a cluster is created, switch to the cluster list to view the created cluster. If the **Status** of the cluster is **Available**, the cluster is created successfully. See **Figure 1-1**.

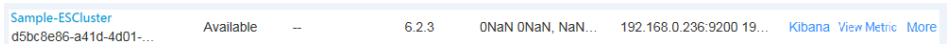
Figure 1-1 Creating a cluster



Step 2: Import Data

CSS supports importing data to Elasticsearch using Cloud Data Migration (CDM), Data Ingestion Service (DIS), Logstash, Kibana, or APIs. Kibana lets you visualize your Elasticsearch data. The following procedure illustrates how to import data to Elasticsearch using Kibana.

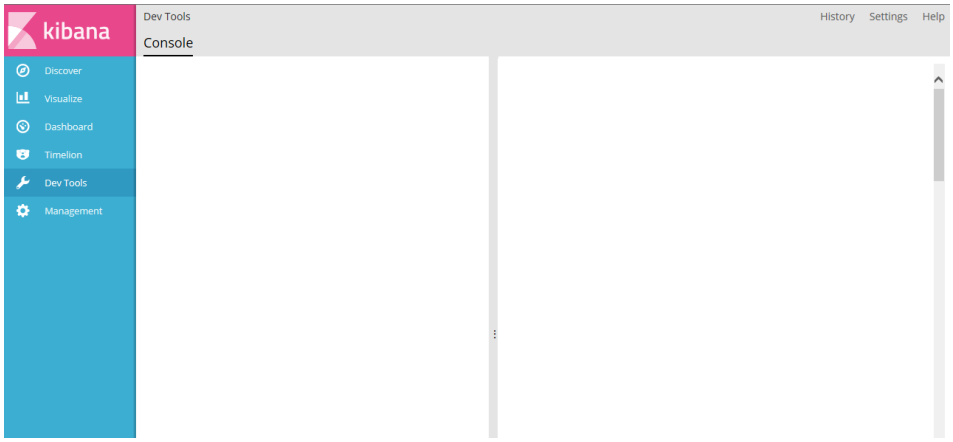
1. On the **Clusters** page of the CSS management console, locate the row where the target cluster resides and click **Kibana** in the **Operation** column.

2. In the left navigation pane of Kibana, click **Dev Tools**. Click **Get to work** to switch to the **Console** page. See **Figure 1-2**.

   Enter the code as required in the left pane and view the result in the right pane.

   Figure 1-2 Console page

3. On the **Console** page, run the following command to create index named **my_store**:

   (Versions earlier than 7.*x*)

```
PUT /my_store
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "products": {
      "properties": {
        "productName": {
          "type": "text",
          "analyzer": "ik_smart"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
}
```

(Versions later than 7.*x*)

```
PUT /my_store
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
          "properties": {
        "productName": {
          "type": "text",
          "analyzer": "ik_smart"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
```

The command output is similar to the following:

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "my_store"
}
```

4. On the **Console** page, run the following command to import data to index named **my_store**:

   (Versions earlier than 7.*x*)

```
POST /my_store/products/_bulk
{"index":{}}
{"productName":"Latest art shirts for women in autumn 2017","size":"L"}
{"index":{}}
{"productName":"Latest art shirts for women in autumn 2017","size":"M"}
{"index":{}}
{"productName":"Latest art shirts for women in autumn 2017","size":"S"}
{"index":{}}
{"productName":"Latest jeans for women in spring 2018","size":"M"}
{"index":{}}
{"productName":"Latest jeans for women in spring 2018","size":"S"}
{"index":{}}
{"productName":"Latest casual pants for women in spring 2017","size":"L"}
{"index":{}}
{"productName":"Latest casual pants for women in spring 2017","size":"S"}
```

   (Versions later than 7.*x*)

```
POST /my_store/_doc/_bulk
{"index":{}}
```



```
{"productName":"Latest art shirts for women in autumn 2017","size":"L"}
{"index":{}}
{"productName":"Latest art shirts for women in autumn 2017","size":"M"}
{"index":{}}
{"productName":"Latest art shirts for women in autumn 2017","size":"S"}
{"index":{}}
{"productName":"Latest jeans for women in spring 2018","size":"M"}
{"index":{}}
{"productName":"Latest jeans for women in spring 2018","size":"S"}
{"index":{}}
{"productName":"Latest casual pants for women in spring 2017","size":"L"}
{"index":{}}
{"productName":"Latest casual pants for women in spring 2017","size":"S"}
```

If the value of the **errors** field in the command output is **false**, the data is imported successfully.

Step 3: Search for Data

- **Full-text search**

  If you access the e-commerce website and want to search for commodities whose names include "spring jeans", enter "spring jeans" to begin your search. The following text provides the command to be executed on Kibana and the command output.

  Command to be executed on Kibana:

  (Versions earlier than 7.*x*)

  ```
  GET /my_store/products/_search
  {
    "query": {"match": {
      "productName": "spring jeans"
    }}
  }
  ```

  (Versions later than 7.*x*)

  ```
  GET /my_store/_search
  {
    "query": {"match": {
      "productName": "spring jeans"
    }}
  }
  ```

  The command output is similar to the following:

  ```
  {
    "took" : 3,
    "timed_out" : false,
    "_shards" : {
      "total" : 1,
      "successful" : 1,
      "skipped" : 0,
      "failed" : 0
    },
    "hits" : {
      "total" : {
        "value" : 4,
        "relation" : "eq"
      },
      "max_score" : 1.7965372,
      "hits" : [
        {
          "_index" : "my_store",
          "_type" : "_doc",
          "_id" : "9xf6VHIBfClt6SDjw7H5",
          "_score" : 1.7965372,
          "_source" : {
            "productName": "Latest jeans for women in spring 2018",
  ```



```
        "size" : "M"
      }
    },
    {
      "_index" : "my_store",
      "_type" : "_doc",
      "_id" : "-Bf6VHIBfClt6SDjw7H5",
      "_score" : 1.7965372,
      "_source" : {
        "productName": "Latest jeans for women in spring 2018",
        "size" : "S"
      }
    },
    {
      "_index" : "my_store",
      "_type" : "_doc",
      "_id" : "-Rf6VHIBfClt6SDjw7H5",
      "_score" : 0.5945667,
      "_source" : {
        "productName": "Latest casual pants for women in spring 2017",
        "size" : "L"
      }
    },
    {
      "_index" : "my_store",
      "_type" : "_doc",
      "_id" : "-hf6VHIBfClt6SDjw7H5",
      "_score" : 0.5945667,
      "_source" : {
        "productName": "Latest casual pants for women in spring 2017",
        "size" : "S"
      }
    }
  ]
 }
}
```

  - Elasticsearch supports word segmentation. The preceding command segments "spring jeans" into "spring" and "jeans".
  - Elasticsearch supports full-text search. The preceding command searches for the information about all commodities whose names include "spring" or "jeans".
  - Unlike traditional databases, Elasticsearch can return results in milliseconds by using inverted indices.
  - Elasticsearch supports sorting by score. In the command output, information about the first two commodities contains both "spring" and "jeans", while that about the last two commodities contains only "spring". Therefore, the first two commodities rank prior to the last two due to high keyword match.

- **Aggregation result display**

  The e-commerce website provides the function of displaying aggregation results. For example, it classifies commodities corresponding to "spring" based on the size so that you can collect the number of commodities of different sizes. The following provides the command to be executed on Kibana and the command output.

  Command to be executed on Kibana:

  (Versions earlier than 7.*x*)

```
GET /my_store/products/_search
{
"query": {
"match": { "productName": "spring" }
```



```
},
"size": 0,
"aggs": {
"sizes": {
"terms": { "field": "size" }
}
}
}
```

(Versions later than 7.*x*)

```
GET /my_store/_search
{
"query": {
"match": { "productName": "spring" }
},
"size": 0,
"aggs": {
"sizes": {
"terms": { "field": "size" }
}
}
}
```

The command output is similar to the following:

(Versions earlier than 7.*x*)

```
{
  "took" : 31,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 4,
    "max_score" : 0.0,
    "hits" : [ ]
  },
  "aggregations" : {
    "sizes" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 0,
      "buckets" : [
        {
          "key" : "S",
          "doc_count" : 2
        },
        {
          "key" : "L",
          "doc_count" : 1
        },
        {
          "key" : "M",
          "doc_count" : 1
        }
      ]
    }
  }
}
```

(Versions later than 7.*x*)

```
{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
```



```
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "sizes" : {
      "doc_count_error_upper_bound" : 0,
      "sum_other_doc_count" : 0,
      "buckets" : [
        {
          "key" : "S",
          "doc_count" : 2
        },
        {
          "key" : "L",
          "doc_count" : 1
        },
        {
          "key" : "M",
          "doc_count" : 1
        }
      ]
    }
  }
}
```


Step 4: Delete the Cluster

Once you understand the process and method of using Elasticsearch, you can perform the following steps to delete the sample cluster and sample data to avoid resource waste.

After a cluster is deleted, its data cannot be recovered. Exercise caution when deleting a cluster.

1. Log in to the CSS management console. In the left navigation pane, click **Clusters** to switch to the **Clusters** page.
2. Locate the row where the **Sample-ESCluster** cluster resides and click **More > Delete** in the **Operation** column.
3. In the displayed dialog box, click **Yes**.

2 Permissions Management

2.1 Creating a User and Granting Permissions

This section describes how to use a group to grant permissions to a user. **Figure 2-1** shows the process for granting permissions.

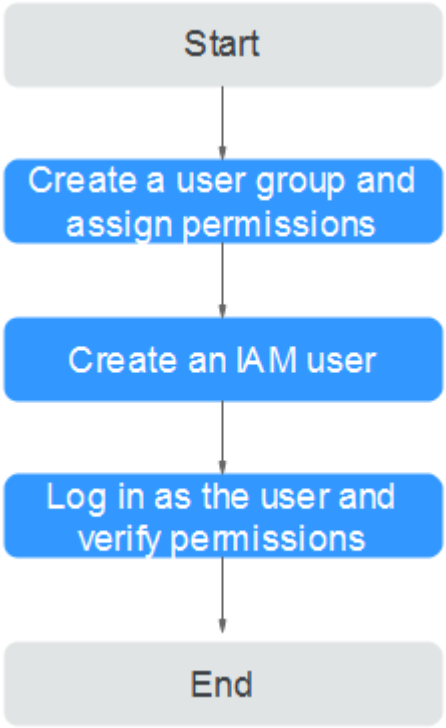
CSS has two types of user permissions: CSS administrator permission and read-only permission. Plan the two types of user groups during permission planning.

Prerequisites

Before assigning permissions to user groups, you should learn about the system policies listed in **Permissions Management**. For the system policies of other services, see **System Permissions**.

Process Flow

Figure 2-1 Process for granting CSS permissions



1. **Create a user group and assign permissions.**

   Create a user group on the IAM console, and assign the CSS permission to the group.
2. **Create an IAM user**.

   Create a user on the IAM console and add the user to the group created in **1. Create a user group and grant permissions to it**.
3. **Log in** and verify permissions.

   Log in to the CSS console as the created user, and verify that it only has read permissions for CSS.

2.2 CSS Custom Policies

Custom policies can be created to supplement the system-defined policies of CSS. For the actions supported for custom policies, see **Permissions Policies and Supported Actions**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following section contains examples of common CSS custom policies.

Example System Policies

Example 1: Granting users the CSS **FullAccess** permission, that is, configuring all CSS permissions for users

Enabling the CSS **FullAccess** permission depends on the OBS and IAM permissions. In addition to configuring the CSS **FullAccess** permission, you also need to add IAM **AgencyFullAccess** permission and all permissions of OBS. To view cluster monitoring information, a user must have the read-only permission of Cloud Eye.

1. Grant the CSS **FullAccess** permission to a user.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Action": [
            "css:*:*",
            "vpc:securityGroups:get",
            "vpc:securityGroups:create",
            "vpc:securityGroups:delete",
            "vpc:securityGroupRules:get",
            "vpc:securityGroupRules:create",
            "vpc:securityGroupRules:delete",
            "vpc:vpcs:list",
            "vpc:privateIps:list",
            "vpc:ports:get",
            "vpc:ports:create",
            "vpc:ports:update",
            "vpc:ports:delete",
            "vpc:quotas:list",
            "vpc:subnets:get",
            "ecs:cloudServerFlavors:get",
            "ecs:serverInterfaces:use",
            "ecs:cloudServers:addNics",
            "ecs:quotas:get",
            "evs:types:get",
            "evs:quotas:get"
         ],
         "Effect": "Allow"
      }
   ]
}
```

2. Grant the **IAM Agency** custom policy to a user.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Action": [
            "iam:agencies:createAgency",
            "iam:agencies:updateAgency",
            "iam:agencies:listAgencies",
            "iam:agencies:getAgency",
            "iam:agencies:deleteAgency"
         ],
         "Effect": "Allow"
      }
   ]
}
```

3. Grant all permissions of OBS to a user.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Action": [
            "OBS:*:*"
         ],
```



```
            "Effect": "Allow"
        }
    ]
}
```

4. (Optional) Grant a user the permission to view cluster monitoring information.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "ces:*:get*",
                "ces:*:list*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

NOTE

If a user account has enabled the enterprise project function:

- When the CSS FullAccess permission is granted to the account, all enterprise projects have the CSS FullAccess permission even if only an enterprise project is configured with the CSS ReadOnlyAccess permission.
- If the CSS FullAccess permission is granted to an enterprise project, all users in the enterprise project can have this permission. For example, if the CSS FullAccess permission is granted to an enterprise project by default, all users in this enterprise project can read and write clusters in this enterprise project.

Example 2: Granting users the CSS **ReadOnlyAccess** permission, that is, allowing users to only read CSS resources To view cluster monitoring information, a user must have the read-only permission of Cloud Eye.

1. Grant the CSS **ReadOnlyAccess** permission to a user.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "css:*:get*",
                "css:*:list*",
                "vpc:securityGroups:get",
                "vpc:securityGroupRules:get",
                "vpc:vpcs:list",
                "vpc:privateIps:list",
                "vpc:ports:get",
                "vpc:quotas:list",
                "vpc:subnets:get",
                "ecs:cloudServerFlavors:get",
                "ecs:quotas:get",
                "evs:types:get",
                "evs:quotas:get"
            ],
            "Effect": "Allow"
        }
    ]
}
```

2. (Optional) Grant a user the permission to view cluster monitoring information.

```
{
    "Version": "1.1",
    "Statement": [
```



```
    {
      "Action": [
        "ces:*:get*",
        "ces:*:list*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

NOTE

If a user account has enabled the enterprise project function:

- If the CSS ReadOnlyAccess permission is granted to the account in IAM and the CSS FullAccess permission is granted to an enterprise project, users in this enterprise project can read clusters in all enterprise projects but can write only clusters in the enterprise project with the CSS FullAccess permission. For example, if the CSS FullAccess permission is granted to an enterprise project by default, users in this enterprise project can read clusters in all enterprise projects, but can write only clusters in the enterprise project with the CSS FullAccess permission.
- If the CSS ReadOnlyAccess permission is granted to the account in IAM but no authorization is configured for any enterprise project, users can only read clusters in this enterprise project. For example, if the CSS ReadOnlyAccess permission is granted to an enterprise project by default, users in this enterprise project can only read clusters in the enterprise project with the CSS ReadOnlyAccess permission.

Example Custom Policies

Example 1: Allowing users to create a CSS cluster

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "css:cluster:create",
        "vpc:securityGroups:get",
        "vpc:securityGroups:create",
        "vpc:securityGroups:delete",
        "vpc:securityGroupRules:get",
        "vpc:securityGroupRules:create",
        "vpc:securityGroupRules:delete",
        "vpc:vpcs:list",
        "vpc:privateIps:list",
        "vpc:ports:get",
        "vpc:ports:create",
        "vpc:ports:update",
        "vpc:ports:delete",
        "vpc:quotas:list",
        "vpc:subnets:get",
        "ecs:cloudServerFlavors:get",
        "ecs:serverInterfaces:use",
        "ecs:cloudServers:addNics",
        "ecs:quotas:get",
        "evs:types:get",
        "evs:quotas:get"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example 2: Denying cluster deletion

A policy with only **Deny** permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both **Allow** and **Deny**, the **Deny** permissions take precedence over the **Allow** permissions.

The following method can be used if you need to assign permissions of the **CSS Admin** policy to a user but you want to prevent the user from deleting clusters. Create a custom policy for denying cluster deletion, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on CSS except deleting clusters. The following is an example of a deny policy:

```
{
    "Version": "1.1",
    "Statement": [
        {
       "Effect": "Deny",
            "Action": [
                "css:cluster:delete"
            ]
        }
    ]
}
```

Example 3: Defining permissions for multiple services in a policy

A custom policy can contain the actions of multiple services that are of the global or project-level type. The following is an example policy containing actions of multiple services:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "ecs:cloudServers:resize",
        "ecs:cloudServers:delete",
        "ecs:cloudServers:delete",
        "css:cluster:restart",
        "css:*:get*",
        "css:*:list*"
      ],
      "Effect": "Allow"
    }
  ]
}
```


3 Creating and Accessing a Cluster

3.1 Creating a Cluster

To use CSS, create a cluster first.

Context

- If you choose the yearly/monthly discount package billing mode, purchase a discount package and then create a cluster with the same node specifications and node storage in the region as those of the discount package. For details about how to purchase a discount package, see **Discount Package**.
- If you choose the pay-per-use billing mode, you can directly create a cluster.

Procedure

1. Log in to the CSS **management console**.
2. On the **Dashboard** or **Clusters** page, click **Create Cluster** to switch to the **Create Cluster** page.
3. Specify **Region** and **AZ**.

   Region: Select the region for the cluster from the drop-down menu to the right of **Region**.

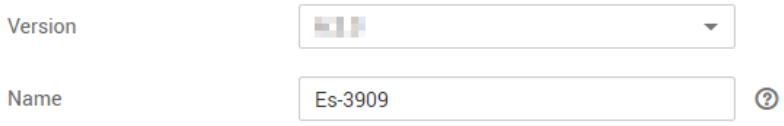
   AZ: Select an AZ associated with the cluster region. For details, see **What Are Regions and AZs?**.

   You can select one or more AZs. For details, see **Multi-AZ HA**.
4. Set basic information about the cluster. Specifically, specify **Version** and **Name**.
   - **Version**: Currently, versions 5.5.1, 6.2.3, 6.5.4, 7.1.1, and 7.6.2 are supported.
   - **Name**: Enter a cluster name containing 4 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The value must start with a letter.

NOTE

After a cluster is created, you can modify the cluster name as required. Click the name of a cluster to be modified. On the displayed **Basic Information** page, click next to the cluster name. After the modification is completed, click to save the modification. If you want to cancel the modification, click .

Figure 3-1 Configuring basic information



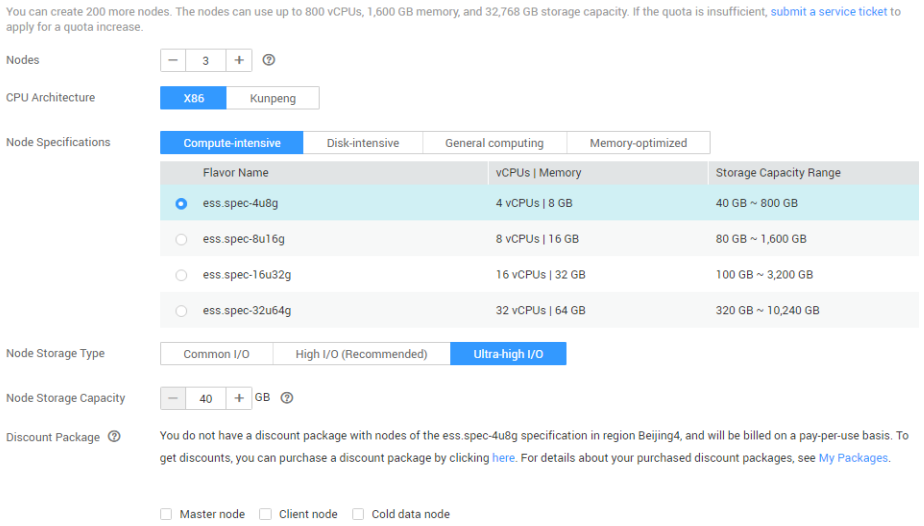
5. Set host specifications of the cluster.

Table 3-1 Parameter description

Parameter	Description
Nodes	Number of nodes in a cluster. • If neither a master node nor client node is enabled, the nodes specified by this parameter are used to serve as both the master node and client node. In addition, the nodes provide the functions of cluster management, data storage, cluster access, and data analysis. To ensure data stability in the cluster, it is recommended that you set this parameter to a value not less than 3. • If only a master node is enabled, the nodes specified by this parameter are used to store data and provide functions of the client node. • If both master and client nodes are enabled, the nodes specified by this parameter are only used for storing data. • If only the client node is enabled, the nodes specified by this parameter are used to store data and provide functions of the master node.
CPU Architecture	Currently, **x86** and **Kunpeng** are supported. The exact type is determined by the region selected during cluster creation.
Node Specifications	Flavor of nodes in a cluster. You can select a type and select a node flavor from the type. Only one node flavor can be selected for each cluster. For details about node flavors, see **ECS Types**. You cannot select the CPU and memory resources that have been sold out.

Parameter	Description
Node Storage Type	In the current version, the following options are available: **Common I/O**, **High I/O**, and **Ultra-high I/O**.
Node Storage Capacity	Storage space. Its value is related to node specifications and varies with node specifications.
Master node	The master node manages all nodes in the cluster. If 20 or more nodes are required to store and analyze the large amount of data, you are advised to enable the master node to ensure cluster stability. Otherwise, you are advised to set only the **Nodes** parameter and purchase nodes as both master and client nodes. After enabling the master node, specify **Node Specifications**, **Nodes**, and **Node Storage Capacity**. The value of **Nodes** must be an odd number greater than 3. You can set a maximum of nine nodes. The value of **Node Storage** is fixed. You can select a storage type based on your needs.
Client node	The client node allows clients to access clusters and analyze data. If 20 or more nodes are required to store and analyze the large amount of data, you are advised to enable the client node to ensure cluster stability. Otherwise, you are advised to set only the **Nodes** parameter and purchase nodes as both master and client nodes. After enabling the client node, specify **Node Specifications**, **Nodes** and **Node Storage Capacity**. The value of **Nodes** ranges from 1 to 32. The value of **Node Storage** is fixed. However, you can select a storage type based on your needs.
Cold data node	The cold data node is used to store historical data, for which query response can be returned in minutes. If you do not demand for timely query response, store historical data on cold data nodes, reducing costs. This parameter is optional. A maximum of 32 cold data nodes are supported. After enabling the cold data node, CSS automatically adds cold or hot tags to related nodes. For details about the parameters, see **https://www.elastic.co/guide/en/elasticsearch/reference/master/allocation-awareness.html**.

Figure 3-2 Configuring host specifications



6. Set network specifications of the cluster. Specifically, specify **VPC**, **Subnet**, and **Security Group**.
   - **VPC**: A VPC is a secure, isolated, logical network environment.

     Select the target VPC. Click **View VPC** to enter the VPC management console to view the created VPC names and IDs. If no VPC is available, create a VPC.

     NOTE

     The selected VPC must contain CIDRs. Otherwise, cluster creation will fail. By default, a created VPC contains CIDRs.
   - **Subnet**: A subnet provides dedicated network resources that are isolated from other networks for higher network security.

     Select the target subnet. You can access the VPC management console to view the names and IDs of the existing subnets in the target VPC.
   - **Security Group**: A security group is a collection of access control rules for ECSs that have the same security protection requirements and are mutually trusted in a VPC. To view more details about the security group, click **View Security Group**.

     NOTE

     Ensure that **Port Range/ICMP Type** is **Any** or a port range includes port **9200** for the selected security group.
   - **Security Mode**: It is supported in version 6.5.4 and later versions. After enabling the security mode, communication is encrypted and authentication is performed on the cluster. The default administrator username is **admin**, and the password needs to be set and confirmed. For details about the security mode, see **Clusters in Security Mode** in *Service Overview*.
   - **HTTPS Access**: When security mode is enabled for a cluster, HTTPS access is enabled by default. A security cluster uses HTTPS for communication. Compared with a non-security cluster that uses HTTP for communication, the read performance of a security cluster is much slower. If you need fast read performance and user permission isolation for a security cluster

to isolate resources (such as indices, documents, and fields), you can disable HTTPS access. After HTTPS access is disabled, HTTP is used to communicate with the cluster. In this case, data security cannot be ensured. In addition, public IP address access cannot be enabled.

- **Public IP Address**: You can configure this parameter only when the cluster has the security mode enabled. After enabling this function, you can obtain an IP address for accessing the cluster on the Internet. For details, see **Public IP Address Access**.
- **Enterprise Project**: When creating a CSS cluster, you can bind an enterprise project to the cluster if you enable the enterprise project. You can select an enterprise project created by the current user from the drop-down list on the right or click **View Project Management** to go to the Enterprise Project Management console to create a new one and view existing projects.

Figure 3-3 Configuring network specifications

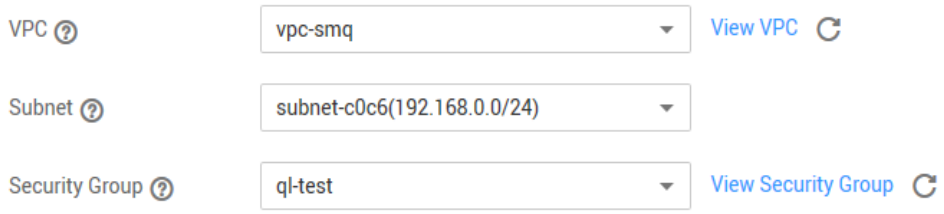
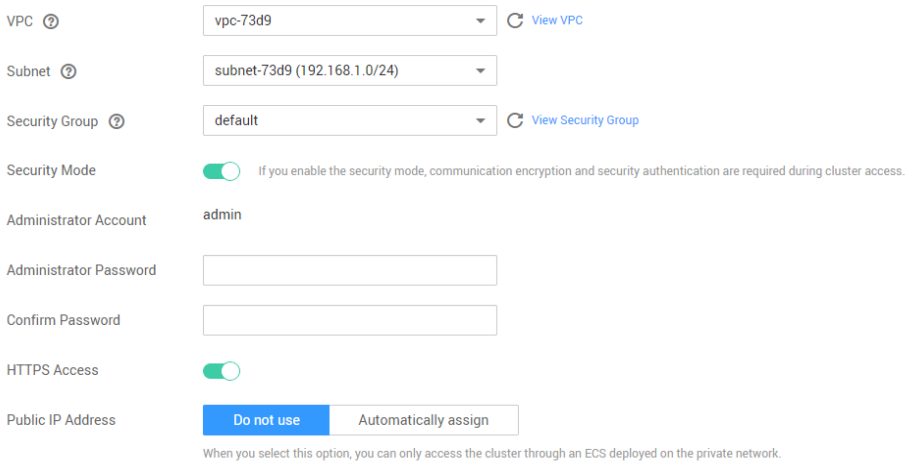


Figure 3-4 Configuring network specifications (security mode enabled)



7. **Advanced Settings**: If you select **Default**, **VPC Endpoint Service** and **Tag** functions are disabled by default, and the **Automatic Snapshot Creation** function is enabled. If you need to configure **VPC Endpoint Service**, **Tag**, and **Automatic Snapshot Creation** functions, select **Custom**.
8. **VPC Endpoint Service**: After enabling this function, you can obtain a private domain name for accessing the cluster in the same VPC. For details, see **VPC Endpoint Service**.
9. **Kibana Public Access**: You can configure this parameter only when security mode is enabled for a cluster. After enabling this function, you can obtain a public IP address for accessing Kibana. For details, see **Kibana Public Access**.

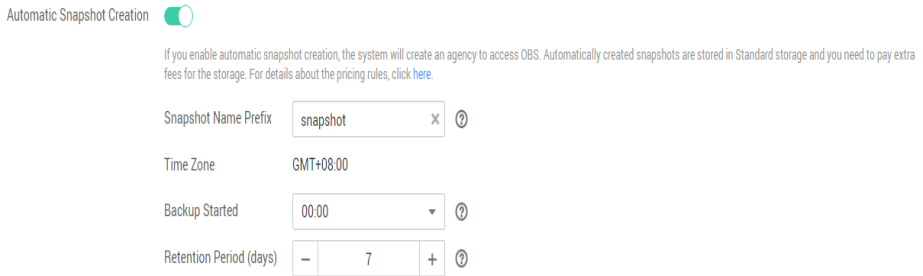
10. **Tag**: Adding tags to clusters can help you identify and manage your cluster resources. You can customize tags or use tags predefined by Tag Management Service (TMS). For details, see **Managing Tags**.
11. Set parameters related to the cluster snapshot.

    By default, the automatic snapshot creation function is enabled. You can set **Snapshot Name Prefix**, **Backup Started**, and **Retention Period (days)** as required. If you do not need to enable the automatic snapshot creation function, click the icon next to **Automatic Snapshot Creation** to disable the automatic snapshot creation function.

    - **Snapshot Name Prefix**: The snapshot name consists of the snapshot name prefix (indicated by this parameter) and time, such as **snapshot-1566921603720**, which is an automatically generated snapshot name. The snapshot name prefix contains 1 to 32 characters and must start with a lowercase letter. Only lowercase letters, digits, hyphens (-), and underscores (_) are allowed.
    - **Backup Started**: indicates the time when the backup starts automatically every day. You can only specify this parameter to an hour's time, for example, **00:00** or **01:00**. The value ranges from **00:00** to **23:00**. Select the backup time from the drop-down list box.
    - **Retention Period (days)**: indicates the duration when snapshots are retained in the OBS bucket, in days. The value ranges from 1 to 90. You can specify this parameter as required. The system automatically deletes snapshots that have been retained for the allowed maximum duration on the half hour.

    For example, if you set the automatic snapshot creation policy as shown in **Figure 3-5**, the system, at 00:30 35 days later, will automatically delete the automated snapshots that were created at 00:00 35 days ago.

    Figure 3-5 Setting parameters related to automatic snapshot creation

12. Click **Next** to switch to the **Confirm** page.
13. After the specifications are confirmed, click **Submit**.
14. Click **Back to Cluster List** to switch to the **Clusters** page. The cluster you created is listed on the displayed page and its status is **Creating**. If the cluster is successfully created, its status will change to **Available**.

    If the cluster fails to be created, create the cluster again as prompted.

3.2 Discount Package

CSS also supports the yearly/monthly discount package. You can make a one-off payment for your purchased nodes and storage capacity according to the service

purchase duration. The service duration ranges from one month to one year. This mode is recommended for long-term users. The yearly/monthly discount package billing mode has the following advantages:

- Preferential price

  The yearly/monthly discount package billing mode saves you 40% on your monthly fees compared with the pay-per-use billing mode. In addition, if you purchase a 10-month discount package, you can enjoy a 12-month available service.

- No binding relationship with clusters

  Within the valid duration of a discount package, you can delete existing clusters at any time and create clusters with the same node specifications, region and node storage as those specified in the discount package.

- Easy scaling of cluster capacity

  For the new nodes added to a cluster, you can purchase a discount package where the required duration is the same as the remaining available duration in the current discount package of existing nodes.

  Yearly/monthly discount package: applies to node instances and provides a larger discount than pay-per-use billing mode. This mode is recommended for long-term users.

  You need to purchase certain quotas (in units of hour) of CSS node instances in advance. When you use the resources, the quotas are preferentially deducted. You only need to pay for the resources that exceed the purchased quotas.

  There is no binding relationship between the yearly/monthly discount package, node instances and node storage. For example:

  a. If your existing node instances or node storage uses the pay-per-use billing mode, the system automatically charges you according to the discount package you purchase, and you do not need to perform any association operations.
  b. After you purchase a discount package, the system does not automatically create a CSS cluster. Instead, you need to log in to the CSS management console and create a cluster with specific node specifications and node storage in the corresponding region.
  c. If you add new node instances to the existing cluster, the node specifications and storage consumed by new node instances will be preferentially deducted from your purchased discount packages by default. To avoid impact on the available quota for the existing node instances, you are advised to purchase a new discount package, which accommodates the same node quantity, node specifications and node storage as those of newly added node instances.
  d. If you create a cluster that has the same node specifications and storage as the deleted cluster, the discount package still applies to the new cluster.

NOTICE

- The discount package is charged as a one-off fee and takes effect immediately after purchase. Currently, you cannot specify the date when the resource package takes effect.
- Within the validity period of your purchased discount package, the system deducts the quota from the discount package first. You are billed for the amount exceeding the upper limit.
- Discount packages are bound to specific regions and node specifications. Your purchased discount packages can be used only in the corresponding regions and on nodes of specific specifications.
- Discount packages are bound to specific regions and node storage. Your purchased discount packages can be used only in the corresponding regions and on nodes of specific storage capacity.

Procedure

Step 1 Log in to the CSS management console.

Step 2 On the **Dashboard** page, click **Buy Discount Package**.

Step 3 On the displayed **Buy Discount Package** page, specify **Region**.

Table 3-2 Parameter description

Parameter	Description
Region	Select the region where nodes in the cluster run.

Step 4 Specify **Package Type**.

Table 3-3 Parameter description

Parameter	Description
Package Type	Currently, only **Node** and **Storage** are supported. • **Node**: Indicates the vCPU quantity and memory size for each node in the cluster. Currently, node specifications include **General computing-plus** and **Memory-optimized**. You can select a type and select a node flavor from the type. Only one node flavor can be selected for each cluster. For details about node flavors, see **ECS Types**. You cannot select the CPU and memory resources that have been sold out. • **Storage**: Indicates the node storage capacity. In the current version, the following three storage types are supported: **Common I/O**, **High I/O**, and **Ultra-high I/O**.

Step 5 Specify **Usage Duration**.

Step 6 Specify **Purchase Quantity**.

NOTE

- After you specify **Usage Duration** and **Purchase Quantity**, the configuration free is automatically displayed in the lower part.
- You can click **Price Details** to view price details of CSS on the displayed **Product Pricing Details** page.

Step 7 Click **Next**.

Step 8 After confirming the order, click **Pay Now**.

Step 9 Choose either **Balance** or **Online Payment** to pay for your order.

Step 10 Pay for your order.

NOTE

Self-service unsubscription of purchased discount packages is not supported. To unsubscribe from a discount package, submit a work order to customer service. The refund amount depends on your usage of the discount package.

- If you choose **Balance** and your available balance is greater than the configuration price, click **Pay**.
- If you choose **Online Payment**, click **Next**, select a specific online payment method, and **PAY NOW**.

After you purchase a discount package, the system does not automatically create a cluster for you to use the package. Instead, you need to go to the CSS management console to create a cluster with the same node specifications and region as those in the discount package. For details about how to create a cluster, see **Creating a Cluster** in the *Cloud Search Service User Guide*.

----End

3.3 Accessing a Cluster

After a cluster is created, you can access the cluster to use Elasticsearch to perform operations, for example, defining index data, importing data, searching for data, and much more. For more information about Elasticsearch, see the **Elasticsearch Reference**. You can use any of the following methods to access a cluster:

- **Accessing a Cluster Using Kibana on the Management Console**
- **Accessing a Cluster by Calling Elasticsearch APIs on the ECS That Is Located in the Same VPC as the Cluster**
- **Accessing a Cluster Using Java API in Non-security Mode**
- **Accessing a Cluster Using the Java API in Security Mode with Elasticsearch**

Accessing a Cluster Using Kibana on the Management Console

1. Log in to the CSS management console.

2. In the left navigation pane, click **Clusters**.
3. On the displayed **Clusters** page, locate the row where the target cluster resides and click **Kibana** in the **Operation** column.

   NOTE

   Normally when you click **Kibana**, a new window will be displayed. If no new window is displayed when you click **Kibana**, the pop-up has been blocked. In this case, manage pop-up blocking to allow access to the blocked pop-up (the Kibana access address).
4. On the Kibana page that is displayed, you can create indices, query indices and documents, and analyze document fields. For details about how to import data to Elasticsearch, see the following sections:
   - **Using CDM to Import Data from OBS to Elasticsearch**
   - **Using DIS to Import Local Data to Elasticsearch**
   - **Using Logstash to Import Data to Elasticsearch**
   - **Using Kibana or APIs to Import Data to Elasticsearch**

Accessing a Cluster by Calling Elasticsearch APIs on the ECS That Is Located in the Same VPC as the Cluster

The ECS that you use to access the cluster by calling Elasticsearch APIs, must meet the following requirements. For details about how to purchase and log in to an ECS, see **Logging In to an ECS** or **Logging In to an ECS**.

- Robust disk space is allocated for the ECS.
- The ECS and the cluster must be in the same VPC.
- The security group of the ECS must be the same as that of the cluster.

  If this requirement is not met, modify the ECS security group or configure the inbound and outbound rules of the ECS security group to allow the ECS security group to be accessed by all security groups of the cluster. For details, see **Configuring Security Group Rules**.
- For security group rule settings of the target CSS cluster, set **Protocol** to **TCP** and **Port Range** to **9200** or a port range including port **9200** for both the outbound and inbound directions.

To access a cluster by calling Elasticsearch APIs on the ECS that is located in the same VPC as the cluster, perform the following steps:

1. Purchase and then log in to an ECS that meets the requirements.
2. To access a cluster, use the private network address and port number of one node in the cluster. You can obtain the private network addresses of nodes from the **Private Network Address** column in the cluster list. If there is only one node in the cluster, the private network address and port number of the only node are displayed. If there are multiple nodes in the cluster, private network addresses and port numbers of all nodes are displayed.

   Assume that there are two nodes in a cluster. Value **10.62.179.32:9200 10.62.179.33:9200** indicates that the private network addresses of the two nodes are **10.62.179.32** and **10.62.179.33** respectively, and port **9200** is used to access both nodes.
3. Run the cURL command to execute the API or call the API by using a program and then execute the program to use the cluster. For details about Elasticsearch operations and APIs, see the **Elasticsearch Reference**.

For example, run the following cURL command to view the index information in the cluster. In this example, the private network address of one node in the cluster is **10.62.179.32** and port **9200** is used to access the cluster.

- If the cluster you access does not have the security mode enabled, run the following command:
  ```
  curl 'http://10.62.179.32:9200/_cat/indices'
  ```
- If the cluster you access has the security mode enabled, access the cluster using HTTPS and add the username, password and **-u** to the cURL command.
  ```
  curl -u username.password -k 'https://10.62.179.32:9200/_cat/indices'
  ```

NOTE

In the preceding command, the private network address and port number of only one node in the cluster are used. If the node fails, the command will fail to be executed. If the cluster contains multiple nodes, you can replace the private network address and port number of the faulty node with those of any available node in the cluster. If the cluster contains only one node, restore the node and execute the command again.

If communication encryption is not enabled in the cluster, the command output is similar to that shown in the following figure.

Figure 3-6 Command output

```
SZX1000355659:/home/elasticsearch-5.5.2/bin # curl 'http://10.62.179.32:9200
/_cat/indices'
green open new_twitter BSVY8wt0SIWSXGzZ5U5mzw 5 1  3 0 21.3kb 10.6kb
green open .kibana     ks7lz4ggTUCy2UDWkXqEgw 1 1  2 0 22.8kb 11.4kb
green open tweets_1    FXOn8ykvQrmvBISyKbFRHA 5 1  0 0  1.5kb   810b
green open my_store    AWybpSpLQPK_2T4cWDN-TQ 5 1 20 0 41.2kb 20.6kb
green open my_index    QF5ARy2VQ6G0t8BzZEI86g 5 1  1 0    9kb  4.5kb
green open tweets_2    uLdSGZ0BS7uaml0fxDlEsQ 5 1  0 0  1.5kb   810b
green open twitter     lzPIdrmRQpeBglI76SAYGA 5 2  3 0 40.5kb 13.5kb
green open my_index2   oLjbtIBPSNqeVIXgHOXQHg 5 1  0 0  1.8kb   955b
```

Accessing a Cluster Using Java API in Non-security Mode

The non-security mode indicates the status of cluster 6.5.4 and later versions without the security mode enabled and the status of clusters of other versions. You can use either of the following methods to access a cluster: the TransportClient or RestHighLevelClient class. For cluster 6.2.3 and 5.5.1, you are advised to use the TransportClient class. For cluster 6.5.4 and later versions, you are advised to use the RestHighLevelClient class.

- Create a client using the default method of the TransportClient class.
  ```
  Settings settings = ImmutableSettings.settingsBuilder().put("client.transport.sniff",false).build();
  TransportClient client = new TransportClient(settings) .addTransportAddress(new
  InetSocketTransportAddress("host1", 9300));
  ```
- Create a client using the default method of the RestHighLevelClient class.
  ```
  RestHighLevelClient client = new RestHighLevelClient(
        RestClient.builder(
              new HttpHost("localhost", 9200, "http")));
  ```

Accessing a Cluster Using the Java API in Security Mode with Elasticsearch

After enabling the security mode function for Elasticsearch 6.5.4 and later versions, accessing a cluster requires the use of HTTPS and username and password for authentication.

You need to use clusters of version 6.5.4 and later as well as related APIs if using the Java API to access a cluster, because the TransportClient class in the earlier version cannot access a cluster using the username and password.

Two accessing modes are available: Create a client using either the TransportClient or RestHighLevelClient class. RestHighLevelClient is recommended.

- **Create a client using the TransportClient class.**

  Run the following commands on the client to generate the keystore and truststore files. The certificate (**CloudSearchService.cer**) downloaded from the cluster management page is used.

  ```
  keytool -genkeypair -alias certificatekey -keyalg RSA -keystore transport-keystore.jks
  keytool -import -alias certificatekey -file CloudSearchService.cer  -keystore truststore.jks
  ```

  Use the keystore and truststore files to access a cluster, create the TransportClient class using the PreBuiltTransportClient method, and add the settings in the client thread.

  The key code is as follows:

  ```
  String userPw = "username:password";
  String path =
  Paths.get(SecurityTransportClientDemo.class.getClassLoader().getResource(".").toURI()).toString();

  Settings settings = Settings.builder()
                  .put("opendistro_security.ssl.transport.enforce_hostname_verification", false)
                  .put("opendistro_security.ssl.transport.keystore_filepath", path + "/transport-keystore.jks")
                  .put("opendistro_security.ssl.transport.keystore_password", "tscpass")
                  .put("opendistro_security.ssl.transport.truststore_filepath", path + "/truststore.jks")
                  .put("client.transport.ignore_cluster_name", "true")
                  .put("client.transport.sniff", false).build();

  TransportClient client = (new PreBuiltTransportClient(settings, new Class[]
  {OpenDistroSecurityPlugin.class})).addTransportAddress(new
                      TransportAddress(InetAddress.getByName(ip), 9300));

  String base64UserPw = Base64.getEncoder().encodeToString(userPw.getBytes("utf-8"));
              client.threadPool().getThreadContext().putHeader("Authorization", "Basic " +
  base64UserPw);
  ```

- **Create a client using the RestHighLevelClient class.**

  The HttpHost class is used to process HTTP requests. In the HttpHost class, the CredentialsProvider and SSLIOSessionStrategy configuration parameter classes are encapsulated in the customized SecuredHttpClientConfigCallback class to configure request connection parameters.

  SecuredHttpClientConfigCallback: encapsulates all user-defined connection parameters.

  CredentialsProvider: Elasticsearch API, which is used to encapsulate the username and password using the method provided by Elasticsearch.

  SSLIOSessionStrategy: Configure SSL parameters, including the SSL domain name authentication mode and certificate processing mode. The SSLContext class is used to encapsulate related settings.

  You can access a cluster in either of the following modes: ignore certificates and use certificates.

  – Ignore all certificates and skip certificate authentication.

    Construct the TrustManager. Use the default X509TrustManager. Do not rewrite any method. That is, ignore all related operations.

    Construct the SSLContext. Use TrustManager in the preceding step as the parameter and construct the SSLContext with the default method.


```
static TrustManager[] trustAllCerts = new TrustManager[] { new X509TrustManager() {
        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {

        }
        @Override
        public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {

        }
        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    }};
 final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY,
                new UsernamePasswordCredentials(userName, password));
        SSLContext sc = null;
        try{
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        }catch(KeyManagementException e){
                e.printStackTrace();
        }catch(NoSuchAlgorithmException e){
                e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NullHostNameVerifier());

        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,credentialsProvider);

        RestClientBuilder builder = RestClient.builder(new HttpHost(clusterAddress, 9200,
                        "https")).setHttpClientConfigCallback(httpClientConfigCallback);

        RestHighLevelClient client = new RestHighLevelClient(builder);
```

- Load the downloaded certificate (**CloudSearchService.cer**) for accessing a cluster.

  Upload the certificate to the client and use the keytool to convert the certificate into a format that can be read by Java. (The default password of the keytool is **changeit**).

  ```
  keytool -import -alias custom name -keystore path for exporting the certificate and its new name -file path for uploading the certificate
  ```

  Customize the TrustManager class, which is inherited from the X509TrustManager. Read the certificate generated in the preceding step, add it to the trust certificate, and rewrite the three methods of the X509TrustManager interface.

  Construct the SSLContext. Use TrustManager in the preceding step as the parameter and construct the SSLContext with the default method.

  ```
  public static class MyX509TrustManager implements X509TrustManager {

        X509TrustManager sunJSSEX509TrustManager;
        MyX509TrustManager() throws Exception {
            File file = new File("certification file path");
            if (file.isFile() == false) {
                throw new Exception("Wrong Certification Path");
            }
            System.out.println("Loading KeyStore " + file + "...");
            InputStream in = new FileInputStream(file);
            KeyStore ks = KeyStore.getInstance("JKS");
            ks.load(in, "changeit".toCharArray());
            TrustManagerFactory tmf =
                    TrustManagerFactory.getInstance("SunX509", "SunJSSE");
  ```



```
            tmf.init(ks);
            TrustManager tms [] = tmf.getTrustManagers();
            for (int i = 0; i < tms.length; i++) {
                if (tms[i] instanceof X509TrustManager) {
                    sunJSSEX509TrustManager = (X509TrustManager) tms[i];
                    return;
                }
            }
            throw new Exception("Couldn't initialize");
        }

final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY,
                new UsernamePasswordCredentials(userName, password));

        SSLContext sc = null;
        try{
            TrustManager[] tm = {new MyX509TrustManager()};
            sc = SSLContext.getInstance("SSL", "SunJSSE");
            sc.init(null, tm, new SecureRandom());
        }catch (Exception e) {
            e.printStackTrace();
        }

        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NullHostNameVerifier());

        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy,credentialsProvider);
        RestClientBuilder builder = RestClient.builder(new HttpHost(clusterAddress, 9200, "https"))
                .setHttpClientConfigCallback(httpClientConfigCallback);
        RestHighLevelClient client = new RestHighLevelClient(builder);
```

- Sample code

  When the code is running, transfer three parameters, including the access address, cluster login username, and password. The request is **GET /_search{"query": {"match_all": {}}}**.

  NOTE

  The access address of a cluster with security mode enabled usually starts with **https**.

  ESSecuredClient class (Ignore certificates)

```
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.HttpHost;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.action.search.SearchRequest;
import org.elasticsearch.action.search.SearchResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.index.query.QueryBuilders;
import org.elasticsearch.search.SearchHit;
import org.elasticsearch.search.SearchHits;
import org.elasticsearch.search.builder.SearchSourceBuilder;
import javax.net.ssl.*;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
public class ESSecuredClient {
    public static void main(String[] args) throws Exception {
```



```
        String clusterAddress = args[0];
        String userName = args[1];
        String password = args[2];
        RestHighLevelClient client = initESClient(clusterAddress, userName, password);
        //Specific operations based on demand
        try {
            SearchResponse searchResponse = client.search(searchRequest,
RequestOptions.DEFAULT);
            SearchHits hits = searchResponse.getHits();
            for (SearchHit hit : hits) {
                System.out.println(hit.getSourceAsString());
            }
            System.out.println("connected");
            Thread.sleep(2000L);
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                client.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    private static RestHighLevelClient initESClient(String clusterAddress, String userName, String
password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(userName, password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException e) {
            e.printStackTrace();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NullHostNameVerifier());
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy, credentialsProvider);
        RestClientBuilder builder = RestClient.builder(new HttpHost(clusterAddress, 9200,
"https")).setHttpClientConfigCallback(httpClientConfigCallback);
        RestHighLevelClient client = new RestHighLevelClient(builder);
        return client;
    }
    static TrustManager[] trustAllCerts = new TrustManager[]{new X509TrustManager() {
        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }
        @Override
        public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }
        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    }};
    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
```



```
    }
}
```

ESSecuredClient class (Uses certificates)

```
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.client.BasicCredentialsProvider;
import org.apache.http.HttpHost;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.action.search.SearchRequest;
import org.elasticsearch.action.search.SearchResponse;
import org.elasticsearch.client.RequestOptions;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.client.RestHighLevelClient;
import org.elasticsearch.index.query.QueryBuilders;
import org.elasticsearch.search.SearchHit;
import org.elasticsearch.search.SearchHits;
import org.elasticsearch.search.builder.SearchSourceBuilder;
import javax.net.ssl.*;
import java.security.KeyManagementException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
public class ESSecuredClient {
    public static void main(String[] args) throws Exception {
        String clusterAddress = args[0];
        String userName = args[1];
        String password = args[2];
        RestHighLevelClient client = initESClient(clusterAddress, userName, password);
        //Specific operations based on demand
        try {
            SearchResponse searchResponse = client.search(searchRequest,
RequestOptions.DEFAULT);
            SearchHits hits = searchResponse.getHits();
            for (SearchHit hit : hits) {
                System.out.println(hit.getSourceAsString());
            }
            System.out.println("connected");
            Thread.sleep(2000L);
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                client.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    private static RestHighLevelClient initESClient(String clusterAddress, String userName, String
password) {
        final CredentialsProvider credentialsProvider = new BasicCredentialsProvider();
        credentialsProvider.setCredentials(AuthScope.ANY, new
UsernamePasswordCredentials(userName, password));
        SSLContext sc = null;
        try {
            sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new SecureRandom());
        } catch (KeyManagementException e) {
            e.printStackTrace();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        SSLIOSessionStrategy sessionStrategy = new SSLIOSessionStrategy(sc, new
NullHostNameVerifier());
```



```
        SecuredHttpClientConfigCallback httpClientConfigCallback = new
SecuredHttpClientConfigCallback(sessionStrategy, credentialsProvider);
        RestClientBuilder builder = RestClient.builder(new HttpHost(clusterAddress, 9200,
"https")).setHttpClientConfigCallback(httpClientConfigCallback);
        RestHighLevelClient client = new RestHighLevelClient(builder);
        return client;
    }
    static TrustManager[] trustAllCerts = new TrustManager[]{new X509TrustManager() {
        @Override
        public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }
        @Override
        public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
        }
        @Override
        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    }};
    public static class NullHostNameVerifier implements HostnameVerifier {
        @Override
        public boolean verify(String arg0, SSLSession arg1) {
            return true;
        }
    }
}
```

SecuredHttpClientConfigCallback class

```
import org.apache.http.client.CredentialsProvider;
import org.apache.http.impl.nio.client.HttpAsyncClientBuilder;
import org.apache.http.nio.conn.ssl.SSLIOSessionStrategy;
import org.elasticsearch.client.RestClientBuilder;
import org.elasticsearch.common.Nullable;
import java.util.Objects;
class SecuredHttpClientConfigCallback implements RestClientBuilder.HttpClientConfigCallback {
    @Nullable
    private final CredentialsProvider credentialsProvider;
    /**
     * The {@link SSLIOSessionStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOSessionStrategy sslStrategy;
    /**
     * Create a new {@link SecuredHttpClientConfigCallback}.
     *
     * @param credentialsProvider The credential provider, if a username/password have been
supplied
     * @param sslStrategy         The SSL strategy, if SSL / TLS have been supplied
     * @throws NullPointerException if {@code sslStrategy} is {@code null}
     */
    SecuredHttpClientConfigCallback(final SSLIOSessionStrategy sslStrategy,
                                    @Nullable final CredentialsProvider credentialsProvider) {
        this.sslStrategy = Objects.requireNonNull(sslStrategy);
        this.credentialsProvider = credentialsProvider;
    }
    /**
     * Get the {@link CredentialsProvider} that will be added to the HTTP client.
     *
     * @return Can be {@code null}.
     */
    @Nullable
    CredentialsProvider getCredentialsProvider() {
        return credentialsProvider;
    }
    /**
     * Get the {@link SSLIOSessionStrategy} that will be added to the HTTP client.
     *
     * @return Never {@code null}.
     */
```



```
    SSLIOSessionStrategy getSSLStrategy() {
        return sslStrategy;
    }
    /**
     * Sets the {@linkplain
HttpAsyncClientBuilder#setDefaultCredentialsProvider(CredentialsProvider) credential provider},
     *
     * @param httpClientBuilder The client to configure.
     * @return Always {@code httpClientBuilder}.
     */
    @Override
    public HttpAsyncClientBuilder customizeHttpClient(final HttpAsyncClientBuilder
httpClientBuilder) {
        // enable SSL / TLS
        httpClientBuilder.setSSLStrategy(sslStrategy);
        // enable user authentication
        if (credentialsProvider != null) {
            httpClientBuilder.setDefaultCredentialsProvider(credentialsProvider);
        }
        return httpClientBuilder;
    }
}
```

pom.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>1</groupId>
    <artifactId>ESClient</artifactId>
    <version>1.0-SNAPSHOT</version>
    <name>ESClient</name>
    <!-- FIXME change it to the project's website -->
    <url>http://www.example.com</url>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.7</maven.compiler.source>
        <maven.compiler.target>1.7</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>transport</artifactId>
            <version>6.5.4</version>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch</groupId>
            <artifactId>elasticsearch</artifactId>
            <version>6.5.4</version>
        </dependency>
        <dependency>
            <groupId>org.elasticsearch.client</groupId>
            <artifactId>elasticsearch-rest-high-level-client</artifactId>
            <version>6.5.4</version>
        </dependency>
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-api</artifactId>
            <version>2.7</version>
        </dependency>
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
```



```
            <artifactId>log4j-core</artifactId>
            <version>2.7</version>
        </dependency>
    </dependencies>
    <build>
        <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may
be moved to parent pom) -->
            <plugins>
                <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/
lifecycles.html#clean_Lifecycle -->
                <plugin>
                    <artifactId>maven-clean-plugin</artifactId>
                    <version>3.1.0</version>
                </plugin>
                <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-
core/default-bindings.html#Plugin_bindings_for_jar_packaging -->
                <plugin>
                    <artifactId>maven-resources-plugin</artifactId>
                    <version>3.0.2</version>
                </plugin>
                <plugin>
                    <artifactId>maven-compiler-plugin</artifactId>
                    <version>3.8.0</version>
                </plugin>
                <plugin>
                    <artifactId>maven-surefire-plugin</artifactId>
                    <version>2.22.1</version>
                </plugin>
                <plugin>
                    <artifactId>maven-jar-plugin</artifactId>
                    <version>3.0.2</version>
                </plugin>
                <plugin>
                    <artifactId>maven-install-plugin</artifactId>
                    <version>2.5.2</version>
                </plugin>
                <plugin>
                    <artifactId>maven-deploy-plugin</artifactId>
                    <version>2.8.2</version>
                </plugin>
                <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/
lifecycles.html#site_Lifecycle -->
                <plugin>
                    <artifactId>maven-site-plugin</artifactId>
                    <version>3.7.1</version>
                </plugin>
                <plugin>
                    <artifactId>maven-project-info-reports-plugin</artifactId>
                    <version>3.0.0</version>
                </plugin>
            </plugins>
        </pluginManagement>
    </build>
</project>
```

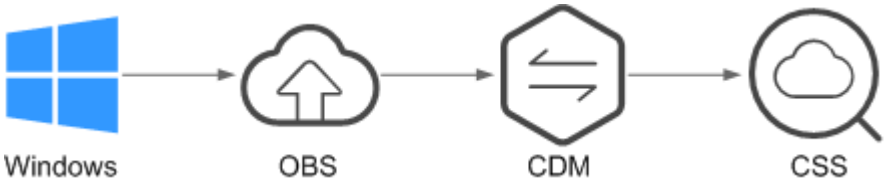

4 Importing Data to Elasticsearch

4.1 Using CDM to Import Data from OBS to Elasticsearch

You can use the CDM-provided wizard to import data stored in OBS to Elasticsearch in CSS. Data files can be in the JSON or CSV format.

Figure 4-1 shows the data transmission process.

Figure 4-1 Process of using CDM to import OBS data to Elasticsearch



Procedure

1. Log in to the OBS management console.
2. Create an OBS bucket for storing data.

   For details, see **Creating a Bucket** in the *Object Storage Service Console Operation Guide*.
3. Upload the data file to the OBS bucket.

   For details, see **Uploading a File** in the *Object Storage Service Console Operation Guide*.

   For example, save the following data as a JSON file and upload the file to the OBS bucket.

```
{"productName":"Latest art shirts for women in autumn 2017","size":"L"}
{"productName":"Latest art shirts for women in autumn 2017","size":"M"}
{"productName":"Latest art shirts for women in autumn 2017","size":"S"}
{"productName":"Latest jeans for women in spring 2018","size":"M"}
{"productName":"Latest jeans for women in spring 2018","size":"S"}
{"productName":"Latest casual pants for women in spring 2017","size":"L"}
{"productName":"Latest casual pants for women in spring 2017","size":"S"}
```


4. Log in to the CSS management console.
5. In the left navigation pane, click **Clusters** to switch to the **Clusters** page.
6. From the cluster list, locate the row where the cluster to which you want to import data resides, and click **Kibana** in the **Operation** column.
7. In the left navigation pane of Kibana, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.
8. On the **Console** page, run the related command to create an index for the data to be stored and specify a custom mapping to define the data type:

   If there is an available index in the cluster where you want to import data, this step is not required. If there is no available index, create an index by referring to the following sample code.

   For example, on the **Console** page, run the following command to create index **demo** and specify a user-defined mapping to define the data type:

   ```
   PUT /demo
   {
     "settings": {
       "number_of_shards": 1
     },
     "mappings": {
       "products": {
         "properties": {
           "productName": {
             "type": "text",
             "analyzer": "ik_smart"
           },
           "size": {
             "type": "keyword"
           }
         }
       }
     }
   }
   ```

   The command is successfully executed if the following information is displayed.

   ```
   {
     "acknowledged" : true,
     "shards_acknowledged" : true,
     "index" : "demo"
   }
   ```
9. Log in to the CDM management console.
10. Purchase a CDM cluster.

    For details, see **Creating a Cluster** in the *Cloud Data Migration User Guide*.
11. Create a link between CDM and CSS.

    For details, see **Creating a Link** in the *Cloud Data Migration User Guide*.
12. Create a link between CDM and OBS.

    For details, see **Creating a Link** in the *Cloud Data Migration User Guide*.
13. Create a job on the purchased CDM cluster and migrate the data in the OBS bucket to the target cluster in CSS.

    For details, see **Table/File Migration** in the *Cloud Data Migration User Guide*.
14. On the **Console** page of Kibana, search for the imported data.

    On the **Console** page of Kibana, run the following command to search for data. View the search results. If the searched data is consistent with the imported data, then the data has been imported successfully.


```
GET demo/_search
```

The command is successfully executed if the following information is displayed.


```
{
  "took": 18,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 7,
    "max_score": 1,
    "hits": [
      {
        "_index": "demo",
        "_type": "products",
        "_id": "g6UepnEBuvdFwWkRmn4V",
        "_score": 1,
        "_source": {
          "size": """"size":"L"}""",
          "productName": """{"productName":"Latest art shirts for women in autumn 2017""""
        }
      },
      {
        "_index": "demo",
        "_type": "products",
        "_id": "hKUepnEBuvdFwWkRmn4V",
        "_score": 1,
        "_source": {
          "size": """"size":"M"}""",
          "productName": """{"productName":"Latest art shirts for women in autumn 2017""""
        }
      },
      {
        "_index": "demo",
        "_type": "products",
        "_id": "haUepnEBuvdFwWkRmn4V",
        "_score": 1,
        "_source": {
          "size": """"size":"S"}""",
          "productName": """{"productName":"Latest art shirts for women in autumn 2017""""
        }
      },
      {
        "_index": "demo",
        "_type": "products",
        "_id": "hqUepnEBuvdFwWkRmn4V",
        "_score": 1,
        "_source": {
          "size": """"size":"M"}""",
          "productName": """{"productName":"Latest jeans for women in autumn 2018""""
        }
      },
      {
        "_index": "demo",
        "_type": "products",
        "_id": "h6UepnEBuvdFwWkRmn4V",
        "_score": 1,
        "_source": {
          "size": """"size":"S"}""",
          "productName": """{"productName":"Latest jeans for women in autumn 2018""""
        }
      },
      {
        "_index": "demo",
```



```
      "_type": "products",
      "_id": "iKUepnEBuvdFwWkRmn4V",
      "_score": 1,
      "_source": {
        "size": """"size":"L"}""",
        "productName": """{"productName":"Latest casual pants for women in autumn 2017""""
      }
    },
    {
      "_index": "demo",
      "_type": "products",
      "_id": "iaUepnEBuvdFwWkRmn4V",
      "_score": 1,
      "_source": {
        "size": """"size":"S"}""",
        "productName": """{"productName":"Latest casual pants for women in autumn 2017""""
      }
    }
  ]
 }
}
```

NOTE

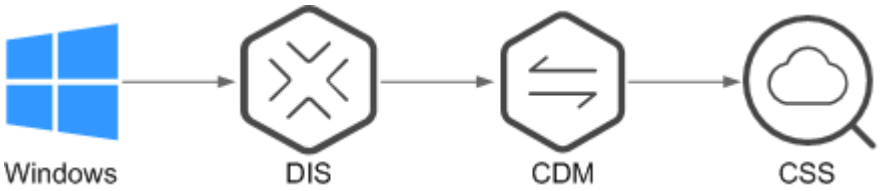
demo specifies the created index name. Set this parameter based on site requirements.

4.2 Using DIS to Import Local Data to Elasticsearch

You can use DIS to upload log data stored on the local Windows PC to the DIS queue and use CDM to migrate the data to Elasticsearch in CSS. In this way, you can efficiently manage and obtain logs through Elasticsearch. Data files can be in the JSON or CSV format.

Figure 4-2 shows the data transmission process.

Figure 4-2 Process of using DIS to import local data to Elasticsearch



Procedure

1. Log in to the DIS management console.
2. Purchase a DIS stream.

   For details, see **Creating a DIS Stream** in the *Data Ingestion Service User Guide*.
3. Install and configure DIS Agent.

   For details, see **Installing DIS Agent** and **Configuring DIS Agent** in the *Data Ingestion Service User Guide*.
4. Start DIS Agent and upload the collected local data to the DIS queue.

   For details, see **Starting DIS Agent** in the *Data Ingestion Service User Guide*.

   For example, upload the following data to a DIS queue using the DIS Agent:


```
{"logName":"aaa","date":"bbb"}
{"logName":"ccc","date":"ddd"}
{"logName":"eee","date":"fff"}
{"logName":"ggg","date":"hhh"}
{"logName":"mmm","date":"nnn"}
```

5. Log in to the CSS management console.
6. In the left navigation pane, click **Clusters** to switch to the **Clusters** page.
7. From the cluster list, locate the row where the cluster to which you want to import data resides, and click **Kibana** in the **Operation** column.
8. In the left navigation pane of Kibana, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.
9. On the **Console** page, run the related command to create an index for the data to be stored and specify a custom mapping to define the data type:

   If there is an available index in the cluster where you want to import data, this step is not required. If there is no available index, create an index by referring to the following sample code.

   For example, on the **Console** page, run the following command to create index **apache** and specify a custom mapping to define the data type:

   ```
   PUT /apache
   {
      "settings": {
         "number_of_shards": 1
      },
      "mappings": {
         "logs": {
            "properties": {
               "logName": {
                  "type": "text",
                  "analyzer": "ik_smart"
               },
               "date": {
                  "type": "keyword"
               }
            }
         }
      }
   }
   ```

   The command is successfully executed if the following information is displayed.

   ```
   {
     "acknowledged" : true,
     "shards_acknowledged" : true,
     "index" : "apache"
   }
   ```

10. Log in to the CDM management console.
11. Purchase a CDM cluster.

    For details, see **Creating a Cluster** in the *Cloud Data Migration User Guide*.
12. Create a link between CDM and CSS.

    For details, see **Creating a Link** in the *Cloud Data Migration User Guide*.
13. Create a link between CDM and DIS.

    For details, see **Creating a Link** in the *Cloud Data Migration User Guide*.
14. Create a job on the purchased CDM cluster and migrate the data in the DIS queue to the target cluster in CSS.

    For details, see **Table/File Migration** in the *Cloud Data Migration User Guide*.

15. On the **Console** page of Kibana, search for the imported data.

    On the **Console** page of Kibana, enter the following command to search for data. View the search results. If the searched data is consistent with the imported data, then the data has been imported successfully.

    ```
    GET apache/_search
    ```

    The command is successfully executed if the following information is displayed.

    ```
    {
      "took": 81,
      "timed_out": false,
      "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
      },
      "hits": {
        "total": 5,
        "max_score": 1,
        "hits": [
          {
            "_index": "apache",
            "_type": "logs",
            "_id": "txfbqnEBPuwwWJWL-qvP",
            "_score": 1,
            "_source": {
              "date": """{"logName":"aaa""""",
              "logName": """"date":"bbb"}"""
            }
          },
          {
            "_index": "apache",
            "_type": "logs",
            "_id": "uBfbqnEBPuwwWJWL-qvP",
            "_score": 1,
            "_source": {
              "date": """{"logName":"ccc""""",
              "logName": """"date":"ddd"}"""
            }
          },
          {
            "_index": "apache",
            "_type": "logs",
            "_id": "uRfbqnEBPuwwWJWL-qvP",
            "_score": 1,
            "_source": {
              "date": """{"logName":"eee""""",
              "logName": """"date":"fff"}"""
            }
          },
          {
            "_index": "apache",
            "_type": "logs",
            "_id": "uhfbqnEBPuwwWJWL-qvP",
            "_score": 1,
            "_source": {
              "date": """{"logName":"ggg""""",
              "logName": """"date":"hhh"}"""
            }
          },
          {
            "_index": "apache",
            "_type": "logs",
            "_id": "uxfbqnEBPuwwWJWL-qvP",
            "_score": 1,
            "_source": {
    ```



```
        "date": """{"logName":"mmm"""",
        "logName": """date":"nnn"}"""
      }
    }
  ]
 }
}
```

NOTE

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

4.3 Using Logstash to Import Data to Elasticsearch

You can use Logstash to collect data and migrate collected data to Elasticsearch in CSS. This method helps you effectively manage and obtain data through Elasticsearch. Data files can be in the JSON or CSV format.

Logstash is an open-source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to Elasticsearch. For details about Logstash, visit the following website: **https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html**

Data importing involves the following two scenarios depending on the Logstash deployment:

- **Importing Data When Logstash Is Deployed on the External Network**
- **Importing Data When Logstash Is Deployed on an ECS**

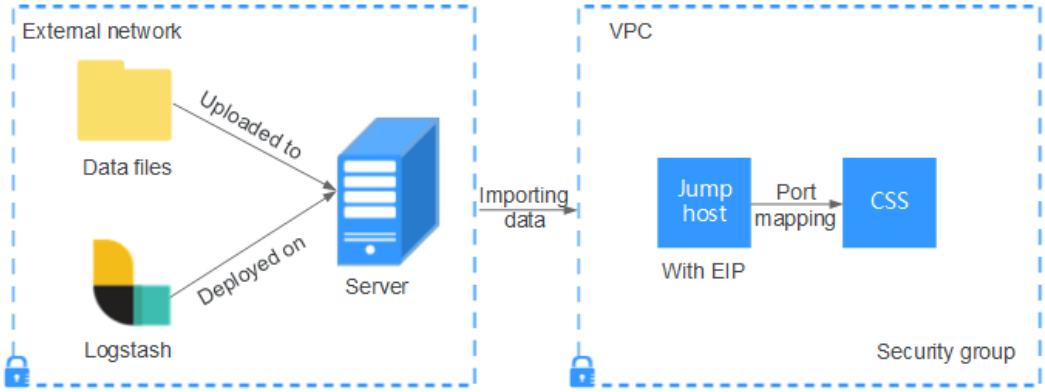
Prerequisites

- To facilitate operations, you are advised to deploy Logstash on a host that runs the Linux operating system (OS).
- To download Logstash, visit the following website: **https://www.elastic.co/downloads/logstash**
- After installing Logstash, perform the following steps to import data. For details about how to install Logstash, visit the following website: **https://www.elastic.co/guide/en/logstash/current/installing-logstash.html**
- The JDK must be installed before the installation of Logstash. In the Linux OS, you can run the **yum -y install java-1.8.0** command to install JDK 1.8.0. In the Windows OS, you can download the required JDK version from the **official website of JDK**, and install it by following the installation guide.
- In the scenario of **Importing Data When Logstash Is Deployed on an ECS**, ensure that the ECS and the Elasticsearch cluster to which data is imported reside in the same VPC.

Importing Data When Logstash Is Deployed on the External Network

Figure 4-3 illustrates how data is imported when Logstash is deployed on the external network.

Figure 4-3 Importing data when Logstash is deployed on the external network



1. Create a jump host and configure it as follows:
   - The jump host is an ECS running the Linux OS and has been bound with an EIP.
   - The jump host resides in the same VPC as the CSS cluster.
   - SSH local port forwarding is configured for the jump host to forward requests from a chosen local port to port **9200** on one node of the CSS cluster.
   - Refer to **SSH documentation** for the local port forwarding configuration.
2. Use PuTTY to log in to the created jump host with the EIP.
3. Run the following command to perform port mapping to transfer the request sent to the port on the jump host to the target cluster:

   ```
   ssh -g -L <Local port of the jump host:Private network address and port number of a node> -N -f root@<Private IP address of the jump host>
   ```

   NOTE

   - In the preceding command, *<Local port of the jump host>* refers to the port obtained in 1.
   - In the preceding command, *<Private network address and port number of a node>* refers to the private network address and port number of a node in the cluster. If the node fails to work, the command will fail to be executed. If the cluster contains multiple nodes, you can replace the value of *<private network address and port number of a node>* with the private network address and port number of any available node in the cluster. If the cluster contains only one node, restore the node and execute the command again.
   - Replace <Private IP address of the *jump host*> in the preceding command with the IP address (with **Private IP**) of the created jump host in the **IP Address** column in the ECS list on the ECS management console.

   For example, port **9200** on the jump host is assigned external network access permissions, the private network address and port number of the node are **192.168.0.81** and **9200**, respectively, and the private IP address of the jump host is **192.168.0.227**. You need to run the following command to perform port mapping:

   ```
   ssh -g -L 9200:192.168.0.81:9200 -N -f root@192.168.0.227
   ```

4. Log in to the server where Logstash is deployed and store the data files to be imported on the server.

For example, data file **access_20181029_log** needs to be imported, the file storage path is **/tmp/access_log/**, and the data file includes the following data:

```
|   All |                 Heap used for segments |                       |    18.6403 |      MB |
|   All |              Heap used for doc values |                       |   0.119289 |      MB |
|   All |                   Heap used for terms |                       |    17.4095 |      MB |
|   All |                   Heap used for norms |                       |  0.0767822 |      MB |
|   All |                  Heap used for points |                       |   0.225246 |      MB |
|   All |           Heap used for stored fields |                       |   0.809448 |      MB |
|   All |                         Segment count |                       |        101 |         |
|   All |                    Min Throughput |  index-append |    66232.6 |  docs/s |
|   All |                 Median Throughput |  index-append |    66735.3 |  docs/s |
|   All |                    Max Throughput |  index-append |    67745.6 |  docs/s |
|   All |          50th percentile latency |  index-append |    510.261 |      ms |
```

5. In the server where Logstash is deployed, run the following command to create configuration file **logstash-simple.conf** in the Logstash installation directory:

```
cd /<Logstash installation directory>/
vi logstash-simple.conf
```

6. Input the following content in **logstash-simple.conf**:

```
input {
Location of data
}
filter {
Related data processing
}
output {
    elasticsearch {
        hosts => "<Public IP address of the jump host>:<Number of the port assigned external network access permissions on the jump host>"
    }
}
```

   - The **input** parameter indicates the data source. Set this parameter based on the actual conditions. For details about the **input** parameter and parameter usage, visit the following website: **https://www.elastic.co/guide/en/logstash/current/input-plugins.html**
   - The **filter** parameter specifies the mode in which data is processed. For example, extract and process logs to convert unstructured information into structured information. For details about the **filter** parameter and parameter usage, visit the following website: **https://www.elastic.co/guide/en/logstash/current/filter-plugins.html**
   - The **output** parameter indicates the destination address of the data. For details about the **output** parameter and parameter usage, visit **https://www.elastic.co/guide/en/logstash/current/output-plugins.html**. Replace <*Public IP address of the jump host*> with the IP address (with **EIP**) of the created jump host in the **IP Address** column in the ECS list on the ECS management console. *<Number of the port assigned external network access permissions on the jump host>* is the number of the port obtained in **1**, for example, **9200**.

   Take the data files in the **/tmp/access_log/** path mentioned in **4** as an example. Assume that data importing starts from data in the first row of the data file, the filtering condition is left unspecified (indicating no data processing operations are performed), the public IP address and port number of the jump host are **192.168.0.227** and **9200**, respectively, and the name of the target index is **myindex**. Edit the configuration file as follows, and enter **:wq** to save the configuration file and exit.

```
input {
    file{
```

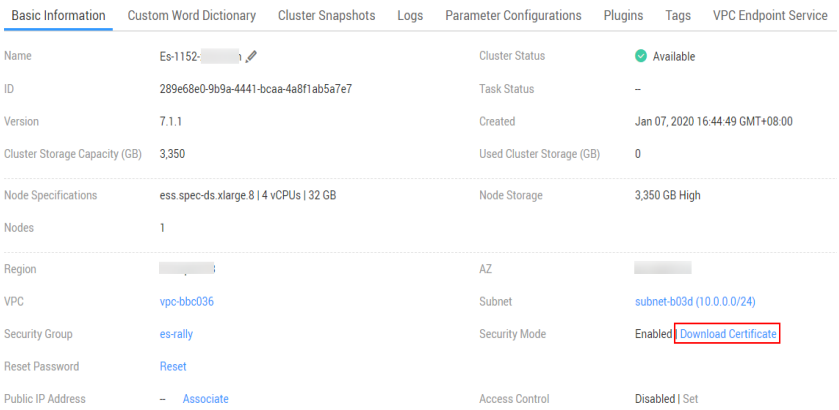


```
    path => "/tmp/access_log/*"
    start_position => "beginning"
  }
}
filter {
}
output {
  elasticsearch {
    hosts => "192.168.0.227:9200"
    index => myindex
    document_type => mytype
  }
}
```

If a cluster has the security mode enabled, you need to download a certificate first.

a. Download a certificate on the **Basic Information** page of the cluster.

Figure 4-4 Downloading a certificate



b. Store the certificate to the server where Logstash is deployed.

c. Modify the **logstash-simple.conf** configuration file.

Take the data files in the **/tmp/access_log/** path mentioned in **4** as an example. Assume that data importing starts from data in the first row of the data file, the filtering condition is left unspecified (indicating no data processing operations are performed), the public IP address and port number of the jump host are **192.168.0.227** and **9200**, respectively. The name of the index for importing data is **myindex**, and the certificate is stored in **/logstash/logstash6.8/config/CloudSearchService.cer**. Edit the configuration file as follows, and enter **:wq** to save the configuration file and exit.

```
input{
   file {
      path => "/tmp/access_log/*"
      start_position => "beginning"
   }
}
filter {
   }
output{
   elasticsearch{
      hosts => ["https://192.168.0.227:9200"]
      index => "myindex"
      user => "admin"
      password => "******"
      cacert => "/logstash/logstash6.8/config/CloudSearchService.cer"
```



```
  }
}
```

NOTE

password: password for logging in to the cluster

7. Run the following command to import the data collected by Logstash to the cluster:
   ```
   ./bin/logstash -f logstash-simple.conf
   ```
8. Log in to the CSS management console.
9. In the left navigation pane, click **Clusters** to switch to the **Clusters** page.
10. From the cluster list, locate the row where the cluster to which you want to import data resides and click **Kibana** in the **Operation** column.
11. In the left navigation pane of the displayed Kibana window, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.
12. On the **Console** page of Kibana, search for the imported data.

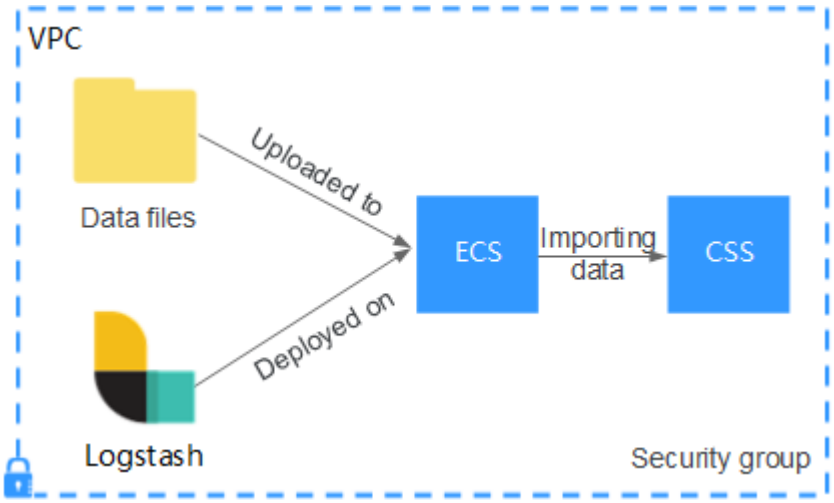
    On the **Console** page of Kibana, enter the following command to search for data. View the search results. If the searched data is consistent with the imported data, then the data has been imported successfully.

    ```
    GET myindex/_search
    ```

Importing Data When Logstash Is Deployed on an ECS

Figure 4-5 illustrates how data is imported when Logstash is deployed on an ECS that resides in the same VPC as the cluster to which data is to be imported.

Figure 4-5 Importing data when Logstash is deployed on an ECS



1. Ensure that the ECS where Logstash is deployed and the cluster to which data is to be imported reside in the same VPC, port **9200** of the ECS security group has been assigned external network access permissions, and an EIP has been bound to the ECS.

NOTE

- If there are multiple servers in a VPC, you do not need to associate EIPs to other servers as long as one server is associated with an EIP. Switch to the node where Logstash is deployed from the node with which the EIP is associated.
- If a private line or VPN is available, you do not need to associate an EIP.

2. Use PuTTY to log in to the ECS.

   For example, data file **access_20181029_log** is stored in the **/tmp/access_log/** path of the ECS, and the data file includes the following data:

```
|   All |             Heap used for segments |                  |    18.6403 |      MB |
|   All |           Heap used for doc values |                  |   0.119289 |      MB |
|   All |                Heap used for terms |                  |    17.4095 |      MB |
|   All |                Heap used for norms |                  |  0.0767822 |      MB |
|   All |               Heap used for points |                  |   0.225246 |      MB |
|   All |        Heap used for stored fields |                  |   0.809448 |      MB |
|   All |                      Segment count |                  |        101 |         |
|   All |                   Min Throughput | index-append |    66232.6 |  docs/s |
|   All |                Median Throughput | index-append |    66735.3 |  docs/s |
|   All |                   Max Throughput | index-append |    67745.6 |  docs/s |
|   All |             50th percentile latency | index-append |    510.261 |      ms |
```

3. Run the following command to create configuration file **logstash-simple.conf** in the Logstash installation directory:

```
cd /<Logstash installation directory>/
vi logstash-simple.conf
```

   Input the following content in **logstash-simple.conf**:

```
input {
Location of data
}
filter {
Related data processing
}
output {
    elasticsearch{
        hosts => "<Private network address and port number of the node>"}
}
```

   - The **input** parameter indicates the data source. Set this parameter based on the actual conditions. For details about the **input** parameter and parameter usage, visit the following website: **https://www.elastic.co/guide/en/logstash/current/input-plugins.html**
   - The **filter** parameter indicates to extract and process logs to convert unstructured information into structured information. For details about the **filter** parameter and parameter usage, visit the following website: **https://www.elastic.co/guide/en/logstash/current/filter-plugins.html**
   - The **output** parameter indicates the destination address of the data. For details about the **output** parameter and parameter usage, visit **https://www.elastic.co/guide/en/logstash/current/output-plugins.html**. *<private network address and port number of a node>* refers to the private network address and port number of a node in the cluster.

     If the cluster contains multiple nodes, you are advised to replace the value of *<Private network address and port number of a node>* with the private network addresses and port numbers of all nodes in the cluster to prevent node faults. Use commas (,) to separate the nodes' private network addresses and port numbers. The following is an example:

```
hosts => ["192.168.0.81:9200","192.168.0.24:9200"]
```

     If the cluster contains only one node, the format is as follows:

```
hosts => "192.168.0.81:9200"
```

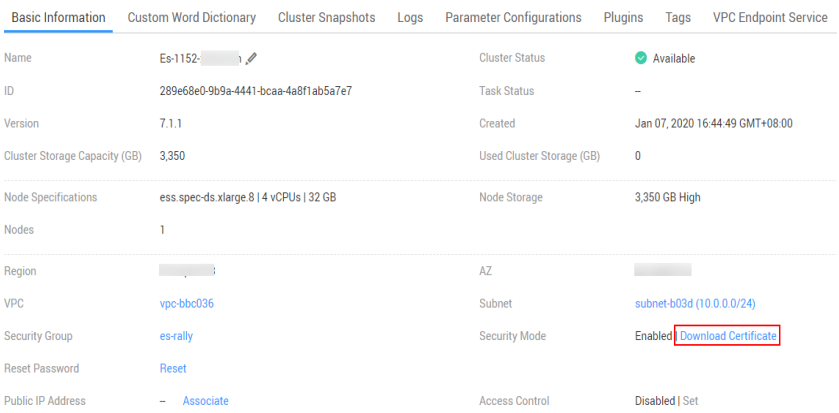

Take the data files in the **/tmp/access_log/** path mentioned in 2 as an example. Assume that data importing starts from data in the first row of the data file, the filtering condition is left unspecified (indicating no data processing operations are performed), the private network address and port number of the node in the cluster where data is to be imported are **192.168.0.81** and **9200**, respectively, and the name of the target index is **myindex**. Edit the configuration file as follows, and enter **:wq** to save the configuration file and exit.

```
input {
    file{
      path => "/tmp/access_log/*"
      start_position => "beginning"
    }
}
filter {
}
output {
    elasticsearch {
      hosts => "192.168.0.81:9200"
      index => myindex
      document_type => mytype
    }
}
```

If a cluster has the security mode enabled, you need to download a certificate first.

a. Download a certificate on the **Basic Information** page of the cluster.

Figure 4-6 Downloading a certificate



b. Store the certificate to the server where Logstash is deployed.

c. Modify the **logstash-simple.conf** configuration file.

Take the data files in the **/tmp/access_log/** path mentioned in 2 as an example. Assume that data importing starts from data in the first row of the data file, the filtering condition is left unspecified (indicating no data processing operations are performed), the public IP address and port number of the jump host are **192.168.0.227** and **9200**, respectively. The name of the index for importing data is **myindex**, and the certificate is stored in **/logstash/logstash6.8/config/CloudSearchService.cer**. Edit the configuration file as follows, and enter **:wq** to save the configuration file and exit.

```
input{
    file {
        path => "/tmp/access_log/*"
```



```
        start_position => "beginning"
    }
}
filter {
    }
output{
    elasticsearch{
        hosts => ["https://192.168.0.227:9200"]
        index => "myindex"
        user => "admin"
        password => "******"
        cacert => "/logstash/logstash6.8/config/CloudSearchService.cer"
    }
}
```

NOTE

password: password for logging in to the cluster

4. Run the following command to import the ECS data collected by Logstash to the cluster:

```
./bin/logstash -f logstash-simple.conf
```

5. Log in to the CSS management console.
6. In the left navigation pane, click **Clusters** to switch to the **Clusters** page.
7. From the cluster list, locate the row where the cluster to which you want to import data resides and click **Kibana** in the **Operation** column.
8. In the left navigation pane of the displayed Kibana window, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.
9. On the **Console** page of Kibana, search for the imported data.

   On the **Console** page of Kibana, enter the following command to search for data. View the search results. If the searched data is consistent with the imported data, then the data has been imported successfully.

```
GET myindex/_search
```

4.4 Using Kibana or APIs to Import Data to Elasticsearch

You can import data in various formats, such as JSON and CSV, to Elasticsearch in CSS by using Kibana or APIs.

Importing Data Using Kibana

Before importing data, ensure that you can use Kibana to access the cluster. The following procedure illustrates how to use the **POST** command to import data.

1. Log in to the **Console** page of Kibana. For details, see **Accessing a Cluster Using Kibana on the Management Console**.

   If it is your first time visiting the **Console** page of Kibana, choose **Dev Tools** from the left navigation pane. Click **Get to work** to switch to the **Console** page of Kibana. If it is not your first time, click **Dev Tools** to directly access the **Console** page of Kibana.
2. (Optional) On the **Console** page, run the related command to create an index for the data to be stored and specify a user-defined mapping to define the data type:

If there is an available index in the cluster where you want to import data, skip this step. If there is no available index, create an index by referring to the following sample code.

For example, on the **Console** page of Kibana, run the following command to create an index named **my_store** and specify a user-defined mapping to define the data type:

```
PUT /my_store
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "products": {
      "properties": {
        "productName": {
          "type": "text"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
}
```

3. In the text box on the right of the **Console** page, enter the following **POST** command (In this example, a data record is imported.):

```
POST /my_store/products/_bulk
{"index":{}}
{"productName":"Latest art shirts for women in 2017 autumn","size":"L"}
```

The command output looks like that in **Figure 4-7**. If the value of the **errors** field in the result is **false**, the data is successfully imported.

Figure 4-7 Response message

Importing Data Using APIs

The following procedure illustrates how to import a JSON data file using bulk APIs through the cURL command.

NOTE

The size of the imported file cannot exceed 50 MB.

1. Log in to the ECS that you use to access the cluster.

   For details about how to access a cluster, see **Accessing a Cluster by Calling Elasticsearch APIs on the ECS That Is Located in the Same VPC as the Cluster**.

2. Run the following command to import JSON data:

   In the command, replace the value of {*Private network address and port number of the node*} with the private network address and port number of a node in the cluster. If the node fails to work, the command will fail to be executed. If the cluster contains multiple nodes, you can replace the value of {*Private network address and port number of the node*} with the private network address and port number of any available node in the cluster. If the cluster contains only one node, restore the node and execute the command again. **test.json** indicates the JSON file whose data is to be imported.

   ```
   curl -X PUT "http://{Private network address and port number of the node} /_bulk" -H 'Content-Type: application/json' --data-binary @test.json
   ```

   NOTE

   The value of the **-X** parameter is a command and that of the **-H** parameter is a message header. In the preceding command, **PUT** is the value of the **-X** parameter and **'Content-Type: application/json' --data-binary @test.json** is the value of the **-H** parameter. Do not add **-k** between a parameter and its value.

   Example: In this example, assume that you need to import data in the **testdata.json** file to an Elasticsearch cluster, where communication encryption is disabled and the private network address and port number of one node are **192.168.0.90** and **9200** respectively. The data in the **testdata.json** file is as follows:

   ```
   {"index": {"_index":"my_store","_type":"products"}}
   {"productName":"Latest art shirts for women in autumn 2019","size":"M"}
   {"index": {"_index":"my_store","_type":"products"}}
   {"productName":"Latest art shirts for women in autumn 2019","size":"L"}
   ```

   Perform the following steps to import the data:

   a. Run the following command to create an index named **my_store**:

   ```
   curl -X PUT http://192.168.0.90:9200/my_store -H 'Content-Type: application/json' -d '
   {
     "settings": {
       "number_of_shards": 1
     },
     "mappings": {
       "products": {
         "properties": {
           "productName": {
             "type": "text"
             },
           "size": {
             "type": "keyword"
           }
         }
       }
   ```



```
  }
}'
```

b. Run the following command to import the data in the **testdata.json** file:

```
curl -X PUT "http://192.168.0.90:9200/_bulk" -H 'Content-Type: application/json' --data-binary @testdata.json
```


5 Suggestions on Using Elasticsearch

Elasticsearch is an open-source search engine. This section provides some experience and suggestions on using Elasticsearch for you to better use CSS.

Improving Indexing Efficiency

- Sending data to Elasticsearch through multiple processes or threads

  A single thread that sends bulk requests is unlikely to max out the indexing capability of a cluster. To maximize utilization of cluster resources, send data through multiple threads or processes, which helps improve data processing efficiency.

  By testing, you can figure out the optimal number of threads for the bulk requests of the same size. The number of threads can be progressively increased until either the load or CPU is saturated in the cluster. You are advised to use the **nodes stats** API to view the CPU and load status of a node. You can learn details by viewing the **os.cpu.percent**, **os.cpu.load_average.1m**, **os.cpu.load_average.5m**, and **os.cpu.load_average.15m** parameter settings. For details about how to use the **nodes stats** API and parameter descriptions, see **https://www.elastic.co/guide/en/elasticsearch/reference/6.2/cluster-nodes-stats.html#os-stats**.

  For example, check whether the load or CPU is saturated if two threads are used during execution of bulk requests. If not saturated, increase threads. If the load or CPU is saturated when the number of threads reaches *N*, you are advised to use *N* threads (the optimal number according to your testing) to execute bulk requests to improve indexing efficiency.

- Increasing the refresh interval

  By default, each shard is automatically refreshed once per second. However, the refresh frequency is not applicable to all scenarios. If you use Elasticsearch to index a great number of log files and want to increase the indexing speed instead of obtaining near-real-time search performance, you can reduce the refresh frequency of each index.

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

- Disabling refresh and replicas for initial loads

If you need to import a large amount of data at a time, it is recommended that you disable refresh and replicas by setting **refresh_interval** to **-1** and **number_of_replicas** to **0**. After data is imported, set **refresh_interval** and **number_of_replicas** to the original values.

Selecting an Appropriate Number of Shards and Replicas

During index data creation, you are advised to specify the number of shards and replicas. Otherwise, default settings (five shards and one replica) will be used.

The shard quantity is strongly relevant to the indexing speed. Too many or too few shards will lead to slow indexing. If too many shards are specified, numerous files will be opened during retrieval, slowing down the communication between servers. If only a few shards are specified, the index size of a single shard may be too large, also slowing down the indexing speed.

Specify the shard quantity based on the node quantity, disk quantity, and index size. It is recommended that the size of a single shard not exceed 30 GB. The shard size is calculated using the following formula: Size of a shard = Total amount of data/Shard quantity

```
PUT /my_index
{
  "settings": {
    "number_of_shards":   1,
    "number_of_replicas":  0
  }
}
```

Storing Data in Different Indices

Elasticsearch relies on Lucene to index and store data and it suits dense data, which means that all documents have the same field.

- Avoiding putting unrelated data in the same index

  Do not put documents that have different data structures into the same index. You can consider creating some smaller indices and use fewer shards to store the documents into the indices.

- Avoiding putting different types in the same index

  It is a good practice to put different types into an individual index. However, be aware that Elasticsearch does not store documents based on type. Therefore, putting different types into one index will slow down indexing. If your documents do not have similar mappings, use different indices.

- Avoiding field conflicts between different types in an index

  Elasticsearch does not allow two different types that have fields of the same name but different mappings.

Creating Indices by Time Range

You are advised to create indices to store time-related data, such as log data, by time range, instead of storing all data in a super large index.

For example, you can store data in an index named by year such as logs_2014 or by month such as logs_2014-10. When the volume of data becomes very large, store data in an index named by day such as logs_2014-10-24.

Creating indices by time range has the following advantages:

- Specifying a suitable number of shards and replicas based on the current volume of data

  You can flexibly set the number of shards and replicas for each index created by time range so that there is no need to set a large shard at the beginning. After the cluster capacity is expanded, the time range can be set to adapt to the cluster scale.

- Deleting old data by deleting old indices

```
DELETE /logs_2014-09
```

- Switching between indices using the alias mechanism

  The following example illustrates how to delete index **logs_2014-09** from alias mechanism **logs_current** and add index **logs_2014-10**.

```
POST /_aliases
{
  "actions": [
    { "add":    { "alias": "logs_current",  "index": "logs_2014-10" }},
    { "remove": { "alias": "logs_current",  "index": "logs_2014-09" }}
  ]
}
```

- Optimizing the indices that stop being updated, such as indices generated last week or month, to increase query efficiency

  Combine multiple segments in the **logs_2014-09-30** index into a shard, improving the query efficiency.

```
PUT /logs_2014-09-30/_settings
{ "number_of_replicas": 0 }

POST /logs_2014-09-30/_forcemerge?max_num_segments=1

PUT /logs_2014-09-30/_settings
{ "number_of_replicas": 1 }
```

Optimizing Index Configurations

- Distinguishing between texts and keywords

  In Elasticsearch, the **string** field is divided into two new data types: text used for full-text search and keyword used for keyword search.

  You are advised to configure exact-value fields without sub-words, such as tags or enumerated values, as the keyword type.

```
PUT my_index1
{
  "mappings": {
    "my_type": {
      "properties": {
        "tags": {
          "type":  "keyword"
        },
        "full_name": {
          "type":  "text"
        }
      }
    }
  }
}
```

- Aggregated statistics based on the text field

  Aggregated statistics based on the text field is not a common requirement. In Elasticsearch, aggregated statistics based on the text field needs to use

fielddata (disabled by default). Enabling fielddata will consume significant memory.

You are advised to conduct multifield mapping on the sub-word string to a text field for full-text search and a keyword field for aggregated statistics.

```
PUT my_index2
{
  "mappings": {
    "my_type": {
      "properties": {
        "full_name": {
          "type": "text",
          "fields": {
            "raw": {
              "type":  "keyword"
            }
          }
        }
      }
    }
  }
}
```

Using Index Templates

Elasticsearch allows you to use index templates to control settings and mappings of certain created indices, for example, controlling the shard quantity to 1 and disabling the _all field. The index template can be used to control which settings can be applied to the created indices.

- In the index template, you can use the template field to specify a wildcard.
- In the event of multiple index templates, you can use order to specify the overwriting sequence. The greater the value, the higher the priority.

In the following example, the index matching **logstash-*** uses the **my_logs** template, and the priority value of the **my_logs** template is 1.

```
PUT /_template/my_logs
{
  "template": "logstash-*",
  "order":    1,
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "_default_": {
      "_all": {
        "enabled": false
      }
    }
  },
  "aliases": {
    "last_3_months": {}
  }
}
```

Data Backup and Restoration

Elasticsearch replicas provide high availability during runtime, which ensures service continuity even when sporadic data loss occurs.

However, replicas do not provide protection against failures. In the case of a failure, you need a real backup for your cluster so that you have a complete copy to restore data.

To back up cluster data, you can create snapshots to save cluster data to OBS buckets. This backup process is "smart". You are advised to use your first snapshot to store a copy of your data. All subsequent snapshots can save the differences between the existing snapshots and the new data. As the number of snapshots increases, backups are added or deleted accordingly. This means that subsequent backups will be very fast since only a small volume of data needs to be transferred.

Improving Query Efficiency by Filtering

Filters are important because they are fast. They do not calculate relevance (avoiding the entire scoring phase) and are easily cached.

Usually, when looking for an exact value, we do not want to score the query. We just want to include/exclude documents, so we will use a constant_score query to execute the term query in a non-scoring mode and apply a uniform score of one.

```
GET /my_store/products/_search
{
   "query" : {
      "constant_score" : {
         "filter" : {
            "term" : {
               "city" : "London"
            }
         }
      }
   }
}
```

Retrieving a Large Amount of Data Through the Scroll API

In the scenario where a large amount of data is returned, the query-then-fetch process supports pagination with the **from** and **size** parameters, but within limits. Results are sorted on each shard before being returned. However, with big-enough from values, the sorting process can become very heavy, using vast amounts of CPU, memory, and bandwidth. For this reason, we strongly advise against deep paging.

To avoid deep pagination, use the scroll query to retrieve a large amount of data.

A scroll query is used to retrieve large numbers of documents from Elasticsearch efficiently, without the hindrance in system performance as with deep pagination. Scrolling allows you to do an initial search and to keep pulling batches of results from Elasticsearch until there are no more results left.

Differences Between Query and Filter

Performance difference: In general, a filter will outperform a scoring query. And it will do so consistently.

When used in filtering context, the query is said to be a **non-scoring** or **filtering** query. That is, the query simply asks the question: Does this document match? The answer is always a simple, binary yes|no.

Typical filtering cases are listed as follows:

- Is the created time in the range from 2013 to 2014?
- Does the **status** field contain the term "published"?
- Is the **lat_lon** field within 10 km of a specified point?

When used in a querying context, the query becomes a "**scoring**" query. Similar to its non-scoring sibling, this determines if a document matches and how well the document matches. A typical use for a query is to find documents:

- Matching the words "full text search"
- Containing the word "run", but maybe also matching "runs", "running", "jog", or "sprint"
- Containing the words "quick", "brown", and "fox" – the closer together they are, the more relevant the document
- Tagged with lucene, search, or java – the more tags, the more relevant the document

Checking Whether a Query Is Valid

Queries can become quite complex and, especially when combined with different analyzers and field mappings, can become a bit difficult to follow. The **validate-query** API can be used to check whether a query is valid.

For example, on the Kibana Console page, run the following command to check whether the query is valid. In this example, the validate request tells you that the query is invalid.

```
GET /gb/tweet/_validate/query
{
 "query": {
 "tweet" : {
   "match" : "really powerful"
  }
}
}
```

The response to the preceding validate request tells us that the query is invalid. To find out why it is invalid, add the explain parameter to the query string and execute the following command.

```
GET /gb/tweet/_validate/query?explain
{
"query": {
   "tweet" : {
  "match" : "really powerful"
  }
  }
 }
```

According to the command output shown in the following, the type of query (match) is mixed up with the name of the field (tweet).

```
{
  "valid": false,
  "error": "org.elasticsearch.common.ParsingException: no [query] registered for [tweet]"
}
```

The following lists a valid query. You can use the validate request together with the **explain** parameter to check its validity.

```
GET /gb/tweet/_validate/query?explain
{
```



```
"query": {
   "match" : {
  "tweet" : "really powerful"
  }
  }
 }
```

Using the explain parameter has the added advantage of returning a human-readable description of the (valid) query, which can be useful for understanding exactly how your query has been interpreted by CSS.

6 Customizing Word Dictionaries

6.1 Configuring a Custom Word Dictionary

When using search engines, certain special Chinese terms, can be recognized during word segmentation.

CSS provides the custom word dictionary function to complete word segmentation in the preceding scenarios. Hot updates of your custom word dictionary are supported. Specifically, the custom word dictionary can take effect without having to restart he cluster.

NOTE

You cannot use the custom word dictionary function for clusters that were created before March 10, 2018 (the launch time of the function).

Basic Concepts

- Main word dictionary: Main words are the words on which users want to perform word segmentation. The main word dictionary is a collection of the main words. The main word dictionary file must be a text file encoded using UTF-8 without BOM, with one subword per line. The maximum size of a main word dictionary file is 100 MB.
- Stop word dictionary: Stop words are the words which users can ignore. A stop word dictionary is a collection of stop words. The stop word dictionary file must be a text file encoded using UTF-8 without BOM, with one subword per line. The maximum size of a stop word dictionary file is 20 MB.
- Synonym dictionary: Synonyms are words with the same meaning. A synonym dictionary is a collection of synonyms. The synonym dictionary file must be a text file encoded using UTF-8 without BOM, with a pair of comma-separated synonyms per line. The maximum size of a synonym dictionary file is 20 MB.

Prerequisites

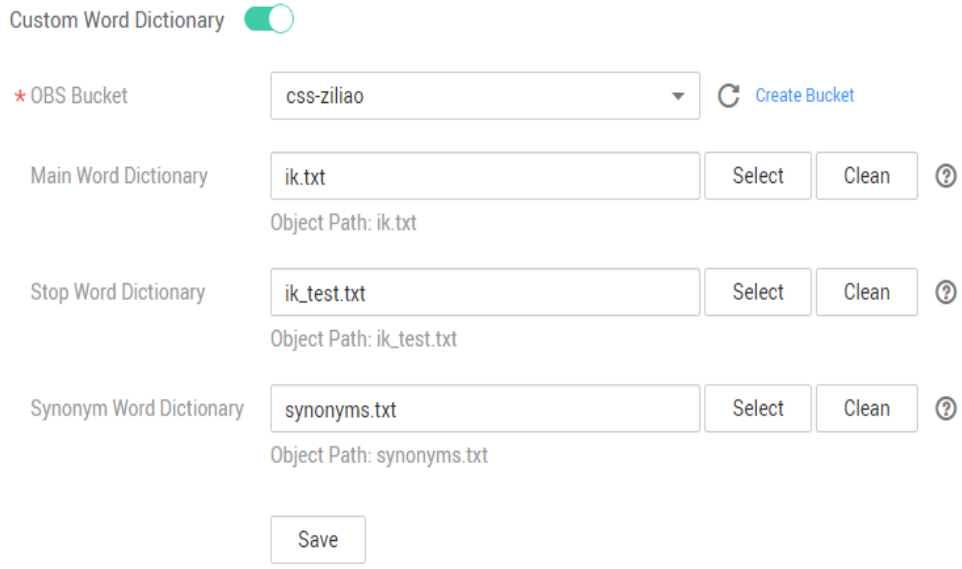
To use the custom word dictionary, the account or IAM user used for logging in to the CSS management console must have both of the following permissions:

- **Tenant Administrator** for project **OBS** in region **Global service**
- **Elasticsearch Administrator** in the current region

Configuring a Custom Word Dictionary

1. In the left navigation pane of the CSS management console, click **Clusters**.
2. On the **Clusters** page that is displayed, click the name of the target cluster.
3. On the displayed page, click **Custom Word Dictionary**.
4. On the displayed **Custom Word Dictionary** page, set the switch to enable or disable the custom word library function.
   - **OBS Bucket**: indicates the OBS bucket where the main word dictionary file, stop word dictionary file, and synonym dictionary file are stored. If no OBS bucket is available, click **Create Bucket** to create one. For details, see **Creating a Bucket**. The OBS bucket to be created must be in the same region as the cluster.
   - **Main Word Dictionary**: indicates the main word dictionary file. Currently, only text files encoded using UTF-8 without BOM are supported. The main word dictionary file must be stored in the corresponding OBS path.
   - **Stop Word Dictionary**: indicates the stop word dictionary file. Currently, only text files encoded using UTF-8 without BOM are supported. The stop word dictionary file must be stored in the corresponding OBS path.
   - **Synonym Word Dictionary**: indicates the synonym dictionary file. Currently, only text files encoded using UTF-8 without BOM are supported. The synonym dictionary file must be stored in the corresponding OBS path.

   Figure 6-1 Configuring a custom word dictionary

5. Click **Save**. In the displayed **Confirm** dialog box, click **OK**. The word dictionary information is displayed in the lower part of the page. In this case, the word dictionary status is **Updating**. Wait about 1 minute. After the word dictionary configuration is complete, the word dictionary status changes to **Succeeded**. In this case, the configured word dictionary has taken effect in the cluster.

Figure 6-2 Word dictionary information

Modifying the Custom Word Dictionary

You can modify the parameters of your configured custom word dictionary as required. You need to upload the target word dictionary files to the corresponding OBS bucket in advance.

On the **Custom Word Dictionary** page, modify **OBS Bucket**, **Main Word Dictionary**, **Stop Word Dictionary**, or **Synonym Word Dictionary**, and click **Save**. Click **OK** in the dialog box that is displayed. After the custom word dictionary is modified, its status changes to **Succeeded**.

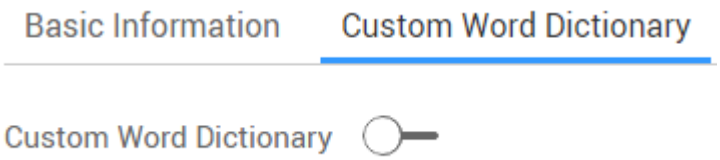
Figure 6-3 Word dictionary information displayed after the modification

Deleting a Custom Word Dictionary

You can delete your custom word dictionary as required to release resources.

On the **Custom Word Dictionary** page, click  . In the displayed dialog box, click **OK**. The following figure shows the **Custom Word Dictionary** page displayed after your configured custom word dictionary is deleted.

Figure 6-4 Page displayed after the custom word dictionary is deleted



6.2 Example

Analyzers

Elasticsearch provides the following two analyzers for using the word dictionary:

- ik_max_word: segments the text at a fine-grained level.

- ik_smart: segments the text at a coarse-grained level.

Example

1. Log in to the CSS management console. Switch to the **Clusters** page. Click the name of the target cluster to switch to the **Basic Information** page.
2. Prepare the main word dictionary file, stop word dictionary file, and synonym dictionary file. Upload the files encoded using UTF-8 without BOM to the corresponding OBS bucket, for example, **obs-b8ed**.

   NOTE

   The default word dictionary contains common stop words. Therefore, you do not need to upload the stop words mentioned in the preceding example.
3. Select the corresponding OBS path by referring to **Configuring a Custom Word Dictionary** and select corresponding main word dictionary file, stop word dictionary file, and synonym dictionary file. Click **Save**.
4. After the word dictionary status changes to **Succeeded**, switch to the **Clusters** page. In the cluster list, locate the row where the target cluster resides and click **Kibana** in the **Operation** column.
5. On the displayed page, click **Dev Tools**. On the displayed page, enter the following code and click  . You can view the word segmentation result on the right pane.
   - Use the ik_smart analyzer to perform word segmentation on *Text used for word segmentation*.

     Example code:

     ```
     POST /_analyze
     {
       "analyzer":"ik_smart",
     "text": "Text used for word segmentation"
     }
     ```

     After the operation is completed, view the word segmentation result.

     ```
     {
       "tokens": [
         {
           "token": "The word segmentation result",
           "start_offset": 0,
           "end_offset": 4,
           "type": "CN_WORD",
           "position": 0
         },
         {
     "token": "The word segmentation result",
           "start_offset": 5,
           "end_offset": 8,
           "type": "CN_WORD",
           "position": 1
         }
       ]
     }
     ```
   - Use the ik_max_word analyzer to perform word segmentation on *Text used for word segmentation*.

     Example code:

     ```
     POST /_analyze
     {
       "analyzer":"ik_max_word",
     ```



```
  "text":"Text used for word segmentation"
}
```

After the operation is completed, view the word segmentation result.

```
{
  "tokens" : [
    {
Smartphones
      "start_offset" : 0,
      "end_offset" : 4,
      "type" : "CN_WORD",
      "position" : 0
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 0,
      "end_offset" : 2,
      "type" : "CN_WORD",
      "position" : 1
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 0,
      "end_offset" : 1,
      "type" : "CN_WORD",
      "position" : 2
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 1,
      "end_offset" : 3,
      "type" : "CN_WORD",
      "position" : 3
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 2,
      "end_offset" : 4,
      "type" : "CN_WORD",
      "position" : 4
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 3,
      "end_offset" : 4,
      "type" : "CN_WORD",
      "position" : 5
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 5,
      "end_offset" : 8,
      "type" : "CN_WORD",
      "position" : 6
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 5,
      "end_offset" : 7,
      "type" : "CN_WORD",
      "position" : 7
    },
    {
      "token" : "The word segmentation result",
      "start_offset" : 6,
      "end_offset" : 8,
      "type" : "CN_WORD",
      "position" : 8
    },
    {
```



```
    "token" : "The word segmentation result",
    "start_offset" : 7,
    "end_offset" : 8,
    "type" : "CN_WORD",
    "position" : 9
  }
 ]
}
```

6. Refer to the following procedure to perform related operations, including creating an index, importing data, conducting search based on the keyword, and viewing the search result.

   a. Create an index named **book**. In this example, set both **analyzer** and **search_analyzer** to **ik_max_word**. You can also select ik_smart.

   (Versions earlier than 7.*x*)

```
PUT /book
{
   "settings": {
      "number_of_shards": 2,
      "number_of_replicas": 1
   },
   "mappings": {
      "type1": {
         "properties": {
            "content": {
               "type": "text",
               "analyzer": "ik_max_word",
               "search_analyzer": "ik_max_word"
            }
         }
      }
   }
}
```

   (Version 7.*X* and later versions)

```
PUT /book
{
   "settings": {
      "number_of_shards": 2,
      "number_of_replicas": 1
   },
   "mappings": {
      "properties": {
         "content": {
            "type": "text",
            "analyzer": "ik_max_word",
            "search_analyzer": "ik_max_word"
         }
      }
   }
}
```

   b. Import data. Import the text information to the **book** index.

   (Versions earlier than 7.*x*)

```
PUT /book/type1/1
{
  "content":"Imported text"
}
```

   (Version 7.*X* and later versions)

```
PUT /book/_doc/1
{
  "content":"Imported text"
}
```

   c. Conduct search based on the keywords.

(Versions earlier than 7.*x*)

```
GET /book/type1/_search
{
  "query": {
    "match": {
      "content": "Keyword"
    }
  }
}
```

(Version 7.*X* and later versions)

```
GET /book/_doc/_search
{
  "query": {
    "match": {
      "content": "Keyword"
    }
  }
}
```

Search result

(Versions earlier than 7.*x*)

```
{
  "took" : 12,
  "timed_out" : false,
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 1.7260926,
    "hits" : [
      {
        "_index" : "book",
        "_type" : "type1",
        "_id" : "1",
        "_score" : 1.7260926,
        "_source" : {
          "content" : "Imported text"
        }
      }
    ]
  }
}
```

(Version 7.*X* and later versions)

```
{
  "took" : 16,
  "timed_out" : false,
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 1.7260926,
    "hits" : [
      {
        "_index" : "book",
        "_type" : "_doc",
        "_id" : "1",
```



```
      "_score" : 1.7260926,
      "_source" : {
        "content" : "Imported text"
      }
    }
  ]
 }
}
```

7. Refer to the following procedure to perform related operations, including creating an index, importing data, conducting search based on the synonym, and viewing the search result.

   a. Create an index.

   (Versions earlier than 7.*x*)

```
PUT myindex
{
  "settings": {
    "analysis": {
      "filter": {
        "my_synonym": {
          "type": "dynamic_synonym"
        }
      },
      "analyzer": {
        "ik_synonym": {
          "filter": [
            "my_synonym"
          ],
          "type": "custom",
          "tokenizer": "ik_smart"
        }
      }
    }
  },
  "mappings": {
    "mytype" :{
      "properties": {
        "desc": {
          "type": "text",
          "analyzer": "ik_synonym"
        }
      }
    }
  }
}
```

   (Version 7.*x* and earlier versions)

```
PUT myindex
{
    "settings": {
        "analysis": {
            "filter": {
                "my_synonym": {
                    "type": "dynamic_synonym"
                }
            },
            "analyzer": {
                "ik_synonym": {
                    "filter": [
                        "my_synonym"
                    ],
                    "type": "custom",
                    "tokenizer": "ik_smart"
                }
            }
        }
    },
```



```
  "mappings": {
    "properties": {
      "desc": {
        "type": "text",
        "analyzer": "ik_synonym"
      }
    }
  }
}
```

b. Import data. Import the text information to the **myindex** index.

(Versions earlier than 7.*x*)

```
PUT /myindex/mytype/1
{
  "desc": "Imported text"
}
```

(Version 7.*X* and later versions)

```
PUT /myindex/_doc/1
{
  "desc": "Imported text"
}
```

c. Conduct search based on the synonym **Keyword** and view the search results.

Run the following command to search for **Keyword**:

```
GET /myindex/_search
{
  "query": {
    "match": {
      "desc": "Keyword"
    }
  }
}
```

Search result

(Versions earlier than 7.*x*)

```
{
  "took": 12,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.41048482,
    "hits": [
      {
        "_index": "myindex",
        "_type": "mytype",
        "_id": "1",
        "_score": 0.41048482,
        "_source": {
          "desc": "Imported text"
        }
      }
    ]
  }
}
```

(Version 7.*X* and later versions)

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
```



```
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.1519955,
    "hits" : [
      {
        "_index" : "myindex",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.1519955,
        "_source" : {
          "desc" : "Imported text"
        }
      }
    ]
  }
}
```


7 Simplified-Traditional Chinese Conversion Plugin

By default, a simplified-traditional Chinese conversion plugin is installed in CSS. The plugin implements conversion between simplified and traditional Chinese. With this plugin, you can search index data containing the corresponding simplified Chinese based on the traditional Chinese keyword, and vice versa.

The simplified-traditional Chinese conversion plugin can be used as the analyzer, tokenizer, token-filter, or char-filter.

The simplified-traditional Chinese conversion plugin provides the following two conversion types:

- s2t: converts the simplified Chinese to the traditional Chinese.
- t2s: converts the traditional Chinese to the simplified Chinese.

Examples (Version 6.5.4)

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 where the target cluster resides and click **Kibana** in the **Operation** column.

   If the target cluster has the security mode enabled, enter the username and password you set when creating the cluster.
4. In the left navigation pane of the displayed Kibana window, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.
5. On the **Console** page, run the following command to create index **stconvert** and specify a user-defined mapping to define the data type:

```
PUT /stconvert
{
  "settings": {
    "number_of_shards": 1,
    "number_of_replicas": 0,
    "analysis": {
      "analyzer": {
        "ts_ik": {
          "tokenizer": "ik_smart",
          "char_filter": [
            "tsconvert",
            "stconvert"
```



```
                ]
            }
        },
        "char_filter": {
            "tsconvert": {
                "type": "stconvert",
                "convert_type": "t2s"
            },
            "stconvert": {
                "type": "stconvert",
                "convert_type": "s2t"
            }
        }
    },
    "mappings": {
        "type": {
            "properties": {
                "desc": {
                    "type": "text",
                    "analyzer": "ts_ik"
                }
            }
        }
    }
}
```

The command output is similar to the following:

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "stconvert"
}
```

6. On the **Console** page, run the following command to import data to index **stconvert**:

```
POST /stconvert/type/1
{
  "desc": "Text in traditional Chinese"
}
```

If the value of **failed** in the command output is **0**, the data is imported successfully.

7. On the **Console** page, run the following command to search for the keyword and view the search result:

```
GET /stconvert/_search
{
    "query": {
        "match": {
      "desc": "Keyword"
        }
    }
}
```

The command output is similar to the following:

```
{
  "took" : 15,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 0.5753642,
    "hits" : [
      {
```



```
      "_index" : "stconvert",
      "_type" : "type",
      "_id" : "1",
      "_score" : 0.5753642,
      "_source" : {
        "desc": "Text in traditional Chinese"
      }
    }
  ]
 }
}
```


8 Managing Clusters

8.1 Cluster Status and Storage Capacity Status

On the **Dashboard** page of the CSS management console, you can view information about the status and storage capacity of existing clusters.

Table 8-1 Cluster status description

Status	Description
Available	Indicates that the cluster is running properly and provides services for users.
Abnormal	Indicates that cluster creation failed or the cluster is unavailable. If a cluster is in the **Unavailable** state, the cluster can be deleted or snapshots created when the cluster is available can be restored to other clusters. However, operations such as expanding cluster capacity, accessing Kibana, creating snapshots, and restoring snapshots to the cluster are not allowed. Data importing is not recommended to avoid data loss. You can view the cluster metrics or restart the cluster. However, the operations may fail because of cluster faults. If the operations fail, contact the customer service personnel in a timely manner.
Processing	Indicates that the cluster is in the middle of a restart, expansion, backup, or recovery.
Creating	Indicates that a cluster is being created.

Table 8-2 Cluster storage capacity status description

Status	Description
Idle	Indicates that the storage capacity usage of all nodes in a cluster is less than 50%.
Warning	Indicates that the storage capacity usage of any node in a cluster is from 50% to less than 80%.
Danger	Indicates that the storage capacity usage of any node in a cluster is greater than or equal to 80%. You are advised to increase the storage space of the cluster to achieve normal data search or analysis.
Abnormal	Indicates that the cluster storage capacity usage is unknown. For example, if the status of a cluster is **Abnormal** due to faults, the storage space status of the cluster is **Abnormal**.

8.2 Introduction to the Cluster List

The cluster list displays all CSS clusters. If there are a large number of clusters, these clusters will be displayed on multiple pages. You can view clusters of all statuses from the cluster list.

Clusters are listed in chronological order by default in the cluster list, with the most recent cluster displayed at the top. You can click  next to the related parameter in the table heading to modify cluster sorting. **Table 8-3** describes the parameters involved in the cluster list.

In the upper right corner of the cluster list, you can enter the name or ID of a cluster and click  to search for a cluster. You can also click  in the upper right corner to refresh the cluster list.

Table 8-3 Cluster list parameter description

Parameter	Description
Name/ID	Name and ID of a cluster. You can click a cluster name to switch to the cluster details page, where basic information about the cluster is displayed. The cluster ID is automatically generated by the system and uniquely identifies a cluster in the service.
Cluster Status	Status of a cluster. For details about the cluster status, see **Cluster Status and Storage Capacity Status**.
Task Status	Status of a task, such as cluster restart, cluster capacity expansion, cluster backup, and cluster restoration.
Version	Elasticsearch version of the cluster.

Parameter	Description
Created	Time when the cluster is created.
Private Network Address	Private network address and port number of the cluster. You can use this parameter value to access the cluster. If the cluster has multiple nodes, the private network addresses and port numbers of all nodes are displayed.
Operation	Operations that can be performed on a cluster, including **Kibana**, **View Metric**, **Modify**, **Restart**, **Delete**, **Custom Word Dictionary**, **Migrate**, **Cerebro,** and **Back Up and Restore**. If an operation is not allowed, the button is gray.

8.3 Viewing Package Details

On the **Package Details** page, you can view the information about the nodes, storage, and bandwidth of all clusters in this account, as well as the usage of your packages.

1. Log in to the CSS console and click **Package Details** on the left navigation pane.
2. On the **Package Details** page, you can view the usage of all packages in this account.

Table 8-4 Node package

Parameter	Description
Specifications	Node specifications of all packages in this account
Package Status	• **Normal** indicates the current package suffices the usage of the node specifications. • **Critical** indicates the current package cannot suffice the usage of the node specifications. • **Warning** indicates the current package has been used up. If you do not renew the package, resources will be billed on a pay-per-use basis.
Nodes	Number of nodes in a CSS cluster, clusters in the **Creation failed** status not included
Packages	Total number of nodes in your packages
Suggestion	Suggestion on planning the usage of the current package
Nodes in Cluster	Distribution of nodes in each cluster in this account
Package Project	Order ID and billing information of your package

Table 8-5 Storage package

Parameter	Description
Specifications	Storage type of all packages in this account
Package Status	• **Normal** indicates the current package suffices the usage of the storage capacity. • **Critical** indicates the current package cannot suffice the usage of the storage capacity. • **Warning** indicates the current package has been used up. If you do not renew the package, resources will be billed on a pay-per-use basis.
Storage Capacity (GB)	Storage capacity of clusters in this account, clusters in the **Creation failed** status not included
Total Storage Capacity (GB)	Total storage capacity of your packages
Suggestion	Suggestion on planning the usage of the current package
Storage in Cluster	Distribution of storage capacity in each cluster in this account
Package Project	Order ID and billing information of your package

Table 8-6 Bandwidth package

Parameter	Description
Specifications	Bandwidth type of clusters in this account
Package Status	• **Normal** indicates the current package suffices the usage of bandwidth. • **Critical** indicates the current package cannot suffice the usage of bandwidth. • **Warning** indicates the current package has been used up. If you do not renew the package, resources will be billed on a pay-per-use basis.
Bandwidth (Mbit/s)	Bandwidth of the cluster in this account, clusters in the **Creation failed** status not included
Total Bandwidth (Mbit/s)	Total bandwidth of your packages
Suggestion	Suggestion on planning the usage of the current package
Bandwidth in Cluster	Distribution of bandwidth in each cluster in this account

Parameter	Description
Package Project	Order ID and billing information of your package

8.4 Index Backup and Restoration

You can back up index data in clusters to avoid data loss. If data loss occurs or you want to retrieve data of a specified duration, you can restore the index data to obtain the data quickly. Index backup is implemented by creating cluster snapshots. When backing up for the first time, you are advised to back up data of all indices.

NOTE

Snapshot creation is unavailable for clusters that were created before March 10, 2018, when the backup and index restoration function was deployed.

- **Managing Automatic Snapshot Creation**: Snapshots are automatically created at a specified time each day according to the rules you create. You can enable or disable the automatic snapshot creation function and set the automatic snapshot creation policy.
- **Manually creating a snapshot**: You can manually create a snapshot at any time to back up all data or data of specified indices.
- **Restoring data**: You can use existing snapshots to restore the backup index data to a specified cluster.
- **Deleting a snapshot**: You are advised to delete invalid snapshots to release storage resources.

NOTE

- Before creating a snapshot, you need to perform basic configurations, including configuring the OBS bucket for storing snapshots, the snapshot backup path, and IAM agency used for security authentication.
- If there are available snapshots in the snapshot list when you configure the OBS bucket for storing cluster snapshots for the first time, the bucket cannot be changed for all snapshots that are automatically or manually created later. Therefore, exercise caution when you configure the OBS bucket.
- If you want to change the OBS bucket where there are snapshots, do as follows: Disable the snapshot function, enable it, and specify a new OBS bucket.

  Once the snapshot function is disabled, the previously created snapshots cannot be used to restore the cluster.
- If a cluster is in the **Unavailable** state, you can only use the cluster snapshot function to restore clusters and view existing snapshot information without being able to edit.
- During backup and restoration of a cluster, allowed operations on the cluster include capacity expansion, Kibana access, metric viewing, and deletion of other snapshots of clusters, but the following operations are not allowed: restart or deletion of the cluster, deletion of a snapshot that is in the **Creating** or **Restoring** state, and creating or restoring another snapshot. If a snapshot is being created or restored for a cluster, then the automatic snapshot creation task initiated for the cluster will be canceled.

Prerequisites

To use the function of creating or restoring snapshots, the account or IAM user used for logging in to the CSS management console must have both of the following permissions:

- **Tenant Administrator** for project **OBS** in region **Global service**
- **Elasticsearch Administrator** in the current region

Managing Automatic Snapshot Creation

1. In the left navigation pane of the CSS management console, click **Clusters**.
2. On the **Clusters** page that is displayed, click the name of the target cluster. On the displayed page, click **Cluster Snapshots**.

   Alternatively, on the **Clusters** page, locate the row where the target cluster resides and click **More > Back Up and Restore** in the **Operation** column to switch to the **Cluster Snapshots** page.
3. On the displayed **Cluster Snapshots** page, click the icon to the right of **Cluster Snapshot** to enable the cluster snapshot function.

    indicates that the cluster snapshot function is disabled.  indicates that the cluster snapshot function is enabled.
4. (Optional) After the cluster snapshot function is enabled, CSS automatically creates the OBS bucket, backup path, and IAM agency for you to store snapshots. The automatically created OBS bucket, backup path, and IAM agency are displayed on the page. If you want to change the OBS bucket, backup path, and IAM agency, click  on the right of **Basic Configuration**.

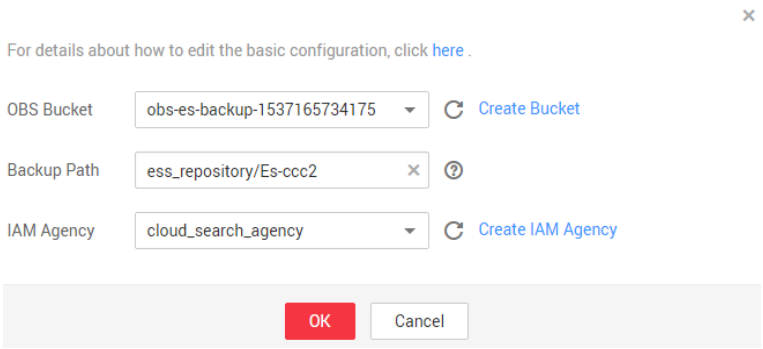
   In the displayed **Edit Basic Configuration** dialog box, you can either select an existing OBS bucket and an IAM agency or create an OBS bucket and an IAM agency. To create an OBS bucket, click **Create Bucket**. To create an IAM agency, click **Create IAM Agency**. For details, see **Creating a Bucket** and **Creating an Agency**.

Table 8-7 Parameter description

Parameter	Description	Precautions
OBS Bucket	Name of the OBS bucket used for storing snapshots.	The following conditions must be met for existing OBS buckets or those to be created: • **Storage Class** is **Standard** or **Infrequent Access**. • **Region** is **CN North-Beijing1**, **CN East-Shanghai2**, or **CN South-Guangzhou**.

Parameter	Description	Precautions
Backup Path	Storage path of the snapshot in the OBS bucket.	The backup path configuration rules are as follows: • The backup path cannot contain the following characters: \:*?"<>| • The backup path cannot start with a slash (/). • The backup path cannot start or end with a period (.). • The total length of the backup path cannot exceed 1,023 characters.
IAM Agency	IAM agency authorized by the current account to CSS to access or maintain data stored in OBS.	The following conditions must be met for existing IAM agencies or those to be created: • **Agency Type** is **Cloud service**. • **Cloud Service** is **Elasticsearch**. • The agency has the **Tenant Administrator** permission for the **OBS** project in **Global service**.

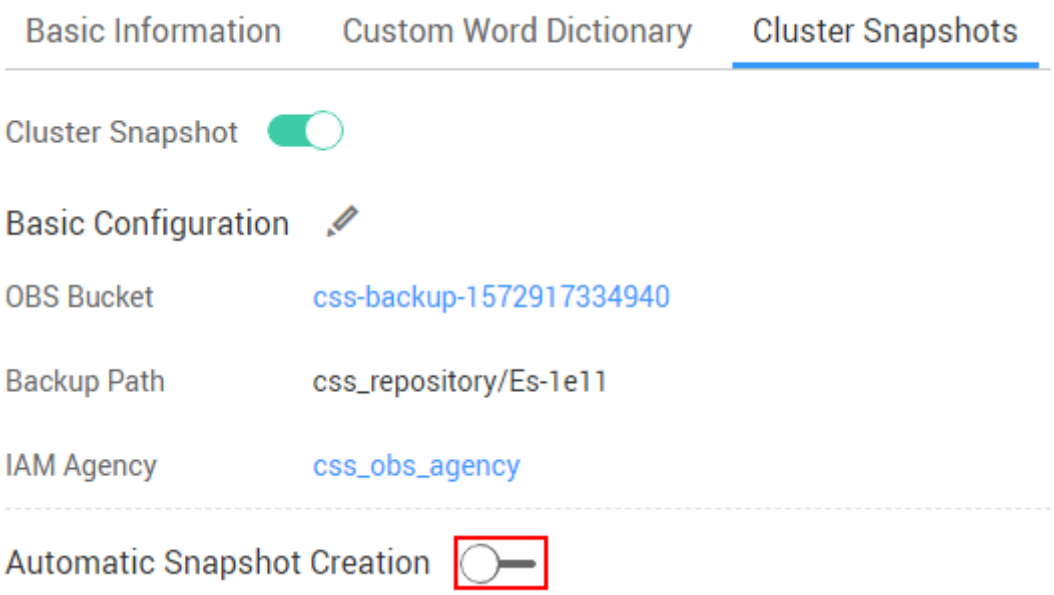
Figure 8-1 Basic configuration



5. Click the icon to the right of **Automatic Snapshot Creation** to enable the automatic snapshot creation function.

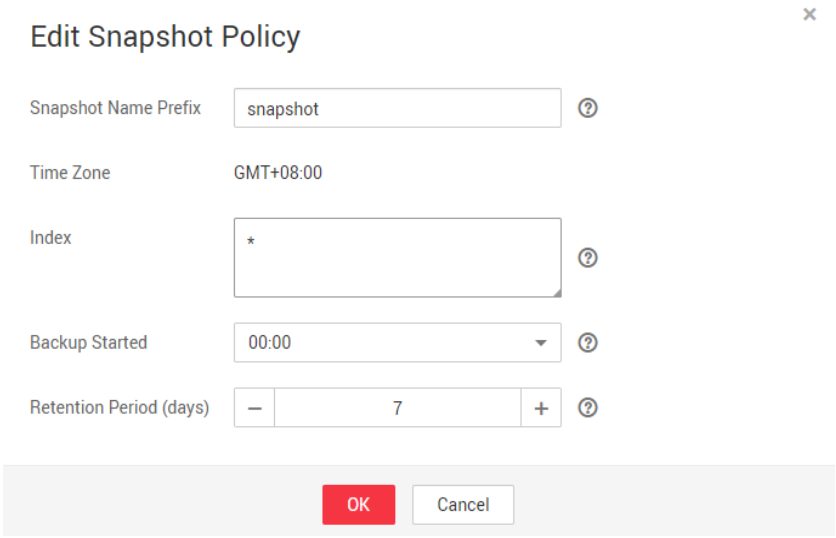
    indicates that the automatic snapshot creation function is enabled, and  indicates that the automatic snapshot creation function is disabled.

Figure 8-2 Enabling or disabling the automatic snapshot creation function



6. In the displayed **Edit Snapshot Policy** dialog box, specify parameters as required.
   - **Snapshot Name Prefix**: The snapshot name consists of the snapshot name prefix (indicated by this parameter) and time. For example, **snapshot-2018022405925**, an automatically generated snapshot name. The snapshot name prefix contains 1 to 31 characters and must start with a lowercase letter. Only lowercase letters, digits, hyphens (-), and underscores (_) are allowed.
   - **Time Zone**: indicates the time zone for the backup time. Specify **Backup Started** based on the time zone.
   - **Backup Started**: indicates the time when the backup starts automatically every day. You can only specify this parameter to an hour's time, for example, **00:00** or **01:00**. The value ranges from **00:00** to **23:00**. Select the backup time from the drop-down list box.
   - **Retention Period (days)**: indicates the duration when snapshots are retained in the OBS bucket, in days. The value ranges from **1** to **90**. You can specify this parameter as required. The system automatically deletes snapshots that are retained over the specified retention period on the half hour. For example, if you set the snapshot policy as shown in **Figure 8-3**, the system will automatically delete in 35 days at 00:30 the automated snapshots that were created 35 days earlier at 00:00.

Figure 8-3 Automatically creating a snapshot



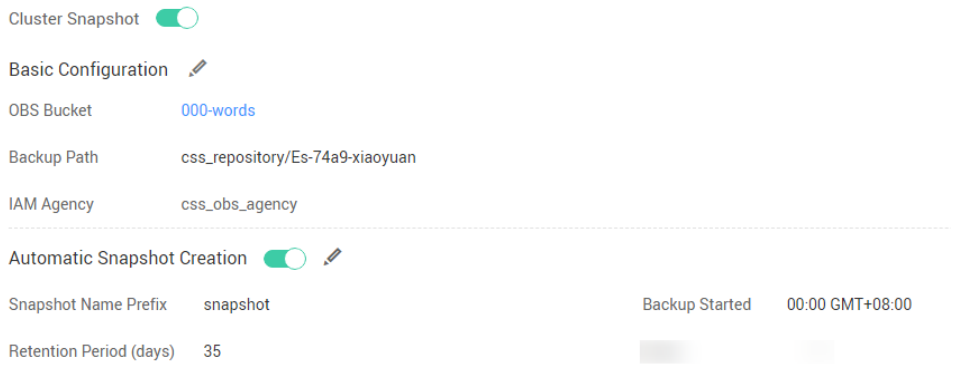
7. Click **OK**.

   After the policy for automatic snapshot creation is created, the policy information will be displayed on the **Cluster Snapshots** page. See **Figure 8-4**.

   If you need to change the policy due to business changes, click .

   Snapshots that are automatically created according to the snapshot policy are displayed in the snapshot list. All automatically and manually created snapshots are displayed in the snapshot list. You can distinguish them by the **Snapshot Type** setting. In the upper right corner of the snapshot list, enter the keyword of the snapshot name or snapshot ID to search for the desired snapshots. You can also click to sort the snapshots based on related parameter settings.

Figure 8-4 Automatic snapshot creation policy



8. (Optional) Disable the automatic snapshot creation function.

   After you disable the automatic snapshot creation function, the system stops automatic creation of snapshots. If the system is creating a snapshot based on the automatic snapshot creation policy and the snapshot has not been displayed in the snapshot list, you cannot disable the automatic snapshot creation function. In this case, if you click the button next to **Automatic Snapshot Creation**, a message is displayed, indicating that you cannot disable the function. You are advised to disable the function after the system

completes automatic creation of the snapshot, specifically, the created snapshot is displayed in the snapshot list.

When disabling the automatic snapshot creation function, you can choose whether to delete the snapshots that have been automatically created by configuring **Delete automated snapshots** in the displayed dialog box. By default, automatically created snapshots are not deleted.

- If you do not select **Delete automated snapshots**, automatically created snapshots are not deleted when you disable the automatic snapshot creation function. In this case, you can manually delete them in the future. For details, see **Deleting a Snapshot**. If you do not manually delete the automatically created snapshots and enable the automatic snapshot creation function again, then all snapshots with **Snapshot Type** set to **Automated** in the snapshot list of the cluster can only be automatically deleted by the system. Specifically, the system automatically deletes snapshots based on the snapshot policy configured when you enable the automatic snapshot creation function again. For example, if you set **Retention Period (days)** to **10**, the system will automatically delete the snapshots that have been retained for more than 10 days.
- If you select **Delete automated snapshots**, all snapshots with **Snapshot Type** set to **Automated** in the snapshot list will be deleted when you disable the automatic snapshot creation function.

Manually Creating a Snapshot

1. In the left navigation pane of the CSS management console, click **Clusters**.
2. On the **Clusters** page that is displayed, click the name of the target cluster. On the displayed page, click **Cluster Snapshots**.

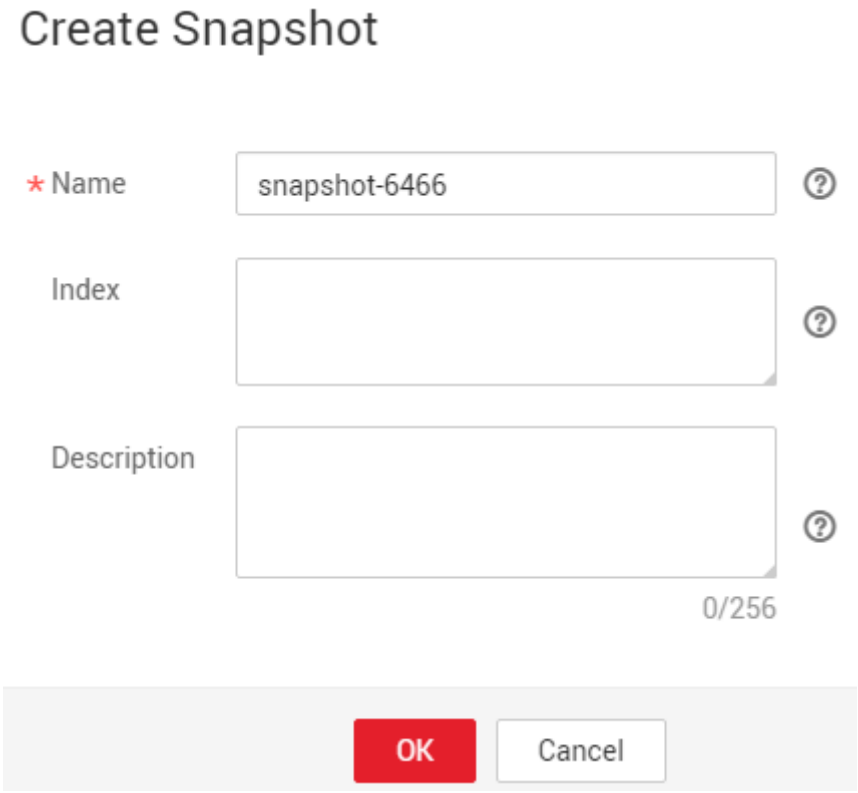
   Alternatively, on the **Clusters** page, locate the row where the target cluster resides and click **More** > **Back Up and Restore** in the **Operation** column to switch to the **Cluster Snapshots** page.
3. On the displayed **Cluster Snapshots** page, click the icon to the right of **Cluster Snapshot** to enable the cluster snapshot function.

    indicates that the cluster snapshot function is disabled.  indicates that the cluster snapshot function is enabled.
4. (Optional) After the cluster snapshot function is enabled, CSS automatically creates the OBS bucket, backup path, and IAM agency for you to store snapshots. The automatically created OBS bucket, backup path, and IAM agency are displayed on the page. If you want to change the OBS bucket, backup path, and IAM agency, click  on the right of **Basic Configuration**. For details about how to configure parameters involved in the basic configuration, see **4**.
5. After basic configurations are completed, click **Create**.
   - **Name**: indicates the name of the manually created snapshot, which contains 4 to 64 characters and must start with a lowercase letter. Only lowercase letters, digits, hyphens (-), and underscores (_) are allowed. Unlike the name of an automatically created snapshot, the name of a manually created snapshot is set as specified and time information is not automatically added to the name.

- **Index**: Enter the name of an index. The manually created snapshot can back up data of certain indices in the cluster. The value contains 0 to 1,024 characters. Uppercase letters, spaces, and certain special characters (including "\<|>/?) are not allowed. Multiple index names are separated by commas (,). If this parameter is left unspecified, data of all indices in the cluster is backed up by default. You can use the asterisk (*) to back up data of certain indices. For example, if you enter **2018-06***, then data of indices with the name prefix of **2018-06** will be backed up.

  You can use the **GET /_cat/indices** command in Kibana to query names of all indices in the cluster. You can then enter the names of the indices you want to back up.
- **Description**: indicates the description of the created snapshot. The value contains 0 to 256 characters, and certain special characters (**<>**) are not allowed.

Figure 8-5 Manually creating a snapshot



6. Click **OK**.

   After the snapshot is created, it will be displayed in the snapshot list. Status **Available** indicates that the snapshot is created successfully. All automatically and manually created snapshots are displayed in the snapshot list. You can distinguish them by the **Snapshot Type** setting. In the upper right corner of the snapshot list, enter the keyword of the snapshot name or snapshot ID to search for the desired snapshots. You can also click to sort the snapshots based on related parameter settings.

Restoring Data

You can use snapshots whose **Snapshot Status** is **Available** to restore cluster data. The stored snapshot data can be restored to other clusters.

Restoring data will overwrite current data in clusters. Therefore, exercise caution when restoring data.

1. In the **Snapshots** area, locate the row where the snapshot you want to restore resides and click **Restore** in the **Operation** column.
2. In the displayed dialog box, specify parameters as required.

   Index: Enter the name of the index you want to restore. By default, this option is left blank, indicating that data of all indices is restored. The value contains 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 contains 0 to 1,024 characters. Uppercase letters, spaces, and certain special characters (including "\<|>/?,) are not allowed. The setting of **Rename Replacement** takes effect only when **Rename Pattern** is specified.

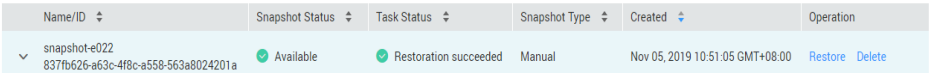
   Cluster: Select the cluster that you want to restore. You can select the current cluster or others. However, you can only restore the snapshot to clusters in the **Available** state. If the current cluster is in the **Unavailable** state, you cannot restore the snapshot to the current cluster. If you choose to restore the snapshot to another cluster, ensure that the target cluster runs an Elasticsearch version 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 one you specify will be overwritten. Therefore, exercise caution when you set the parameters.

Figure 8-6 Restoring a snapshot

3. 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.

   In the snapshot list, the **Task Status** column indicates the latest status of a snapshot and displays **Restoration succeeded** only when the latest restoration of a snapshot succeeds.

   Figure 8-7 Successful restoration



Deleting a Snapshot

If you no longer need a snapshot, delete it to release storage resources. If the automatic snapshot creation function is enabled, snapshots that are automatically created cannot be deleted manually, and the system automatically deletes these snapshots on the half hour after the time specified by **Retention Period (days)**. If you disable the automatic snapshot creation function while retaining the automated snapshots, then you can manually delete them later. If you do not manually delete the automatically created snapshots and enable the automatic snapshot creation function again, then all snapshots with **Snapshot Type** set to **Automated** in the snapshot list of the cluster can only be automatically deleted by the system.

NOTE

After a snapshot is deleted, its data cannot be restored. Therefore, exercise caution when deleting a snapshot.

1. In the **Snapshots** area, locate the row where the target snapshot resides and click **Delete** in the **Operation** column.
2. In the **Delete Snapshot** dialog box that is displayed, click **Yes**.

8.5 Modifying Specifications

If the specifications of a cluster cannot meet business requirements, modify its specifications to improve storage and usage efficiency. You can reduce cluster nodes based on service requirements to optimize the cluster storage at lower O&M costs.

Scaling out Clusters

1. Log in to the CSS management console.
2. Click **Clusters**. Locate the row where the target cluster resides and click **Modify** in the **Operation** column.
3. On the displayed **Modify Configuration** page, specify **New Nodes** and **Node Storage Capacity**.

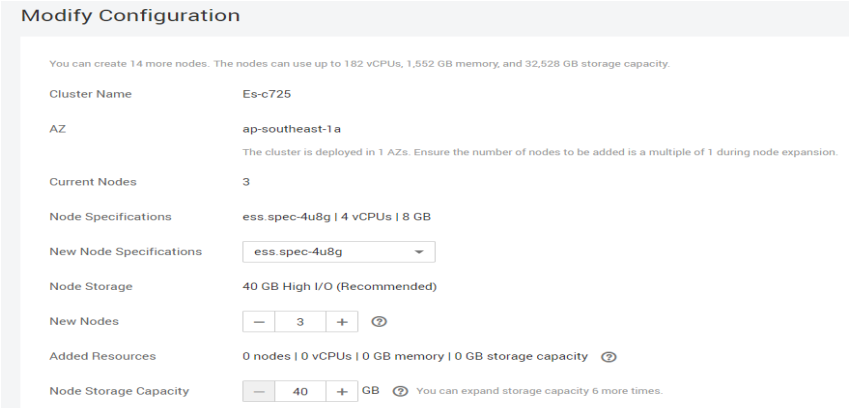
   If a cluster does not have the master node or client node enabled, you can modify the number of nodes or the node storage capacity. Add at least one node and a maximum of 32 nodes are supported.

If a cluster has the master node or client node enabled, you can modify the number of master nodes or client nodes, or the node storage capacity. Add at least one node. A maximum of 200 nodes are supported. A maximum of 9 master nodes and 32 client nodes are supported.

NOTE

- If you only expand the node quantity, the **Node Specifications** and **Node Storage Capacity** settings of newly added nodes are the same as those specified during cluster creation.
- If you expand both the node quantity and the storage capacity, the **Node Specifications** settings of newly added nodes are the same as those specified during cluster creation, while the **Node Storage Capacity** settings of all nodes are changed to the new storage capacity.
- If you only expand the node storage capacity, the **Node Storage Capacity** setting of all nodes is changed to the new storage capacity.
- You can expand the storage capacity six times at most.
- Services are not interrupted during the cluster scale-out.

Figure 8-8 Modifying the cluster specifications



4. Click **Next**.
5. On the displayed **Details** page, confirm the specifications and click **Submit**.
6. Click **Back to Cluster List** to switch to the **Clusters** page. If **Scaling out** is displayed in the **Task Status** column, the cluster specifications are being modified. If **Available** is displayed in the **Cluster Status** column, the modification succeeds.

Scaling in Clusters

1. Log in to the CSS management console.
2. Click **Clusters**. Locate the row where the target cluster resides and click **Modify** in the **Operation** column.
3. On the displayed **Modify Configuration** page, specify **New Nodes**.

NOTE

- The number of nodes to be scaled in should be less than half of the number of nodes in the target cluster.
- The number of nodes after scale-in should be greater than the number of replicas.
- The disk usage after scale-in should be less than 80%.
- Services are not interrupted during the cluster scale-in.

4. Click **Next**.
5. On the displayed **Details** page, confirm the specifications and click **Submit**.
6. Click **Back to Cluster List** to switch to the **Clusters** page. If **Scaling in** is displayed in the **Task Status** column, the cluster specifications are being modified. If **Available** is displayed in the **Cluster Status** column, the modification succeeds.

Modifying the Node Specifications

NOTE

- Only a cluster with three or more nodes can have the node specifications modified.
- The node specifications can only be scaled up to a higher specification.
- The cluster created before this function is brought online cannot have node specifications modified.
- Kibana is unavailable when modifying node specifications.
- You cannot modify node specifications, node quantity, and node storage capacity at the same time.
- If the data volume is large, modifying the node specifications may take more time.

1. Log in to the CSS management console.
2. Click **Clusters**. Locate the row where the target cluster resides and click **Modify** in the **Operation** column.
3. On the displayed page, specify **New Node Specifications**.
4. Click **Next**.
5. On the displayed **Details** page, confirm the specifications and click **Submit**.
6. Click **Back to Cluster List** to switch to the **Clusters** page. If **Configuration modified** is displayed in the **Task Status** column, the node specifications are being modified. If **Available** is displayed in the **Cluster Status** column, the modification succeeds.

8.6 Binding an Enterprise Project

Each cluster must be configured with an enterprise project. If this parameter is not required, you can bind the cluster to the **default** project. For a cluster that is created before the binding of the enterprise project feature, the enterprise project of the cluster is bound to the **default** project. You can modify the bound enterprise project based on the site requirements.

NOTE

To use the enterprise project function, you need to assign permissions to the corresponding account. You can **submit a service ticket** to apply for the permissions.

Binding an Enterprise Project

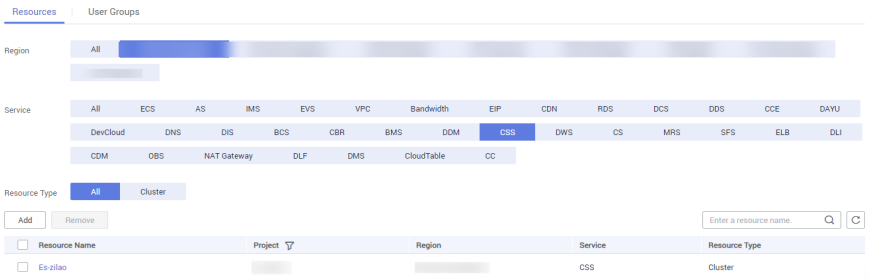
When creating a cluster, you can bind it to an enterprise project by specifying the **Enterprise Project** parameter. For details, see **Creating a Cluster**.

Modifying an Enterprise Project

For a cluster that has been created, you can modify the bound enterprise project based on the site requirements.

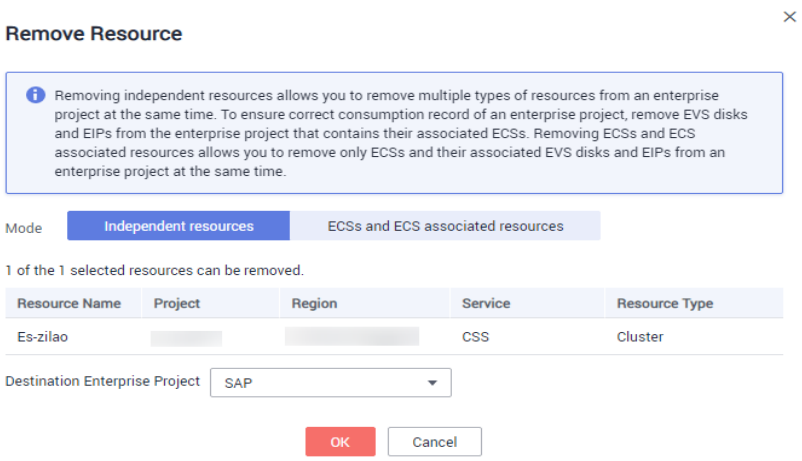
1. On the CSS management console, click **Clusters**.
2. In the cluster list on the displayed page, click the target cluster name to switch to the **Basic Information** page.
3. On the cluster details page, click the parameter value to the right of the **Enterprise Project** parameter. The **Enterprise Project Management** page of the **Enterprise Management Service** is displayed.
4. On the **Resource** tab page, select the corresponding **Region** and select **CSS** from the **Service** drop-down list box. In this case, the corresponding CSS cluster is displayed in the resource list.

Figure 8-9 Filtering CSS clusters



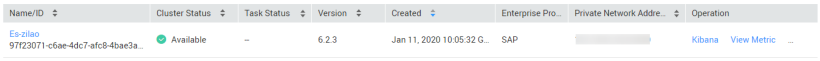
5. Select the cluster whose enterprise project needs to be modified and click **Remove**.
6. On the **Remove Resource** page, specify **Mode** and select **Destination Enterprise Project**, and click **OK**.

Figure 8-10 Removing resources



7. After the cluster resources are removed, its information cannot be obtained from the original enterprise project resource page. You can view the enterprise project bound to the cluster in either of the following ways:
   - Switch to CSS cluster list, where the value of **Enterprise Project** for the cluster is changed to the new enterprise project.

     Figure 8-11 Viewing the enterprise project corresponding to the cluster

   - On the Enterprise Management Service console, choose **Project Management** in the navigation pane on the left. On the **Enterprise Project Management** page, click **View Migration Event** to obtain the cluster information.

     Figure 8-12 Viewing resources

8.7 Restarting a Cluster

If a cluster stops working, you can restart it to restore normal running. Only clusters in the **Available** or **Abnormal** status can be restarted.

Quick Restart

- The cluster is in the **Available** or **Abnormal** status.
- There are no running tasks, such as importing data, searching for data on the cluster.

NOTICE

- The cluster is unavailable during quick restart. If quick restart fails, data may be lost or the cluster may become unavailable. Therefore, exercise caution when performing this operation.
- If the cluster you want to restart is available, you are advised to stop all data processing tasks on the cluster before restarting it. If there are running tasks, for example, importing data, searching for data, the transmitted data may be lost upon the restart. Therefore you are advised to stop all cluster tasks before the quick restart.

1. Log in to the CSS management console.
2. Click **Clusters** to switch to the **Clusters** page. In the row where the cluster you want to restart is located, click **More > Restart** in the **Operation** column. The **Restart Cluster** page is displayed. Select **Quick Restart**.

   The cluster is unavailable during quick restart.
3. Refresh the page and check the cluster status. During the restart, the cluster status is **Processing**, and the task status is **Restarting**. If the cluster status changes to **Available**, the cluster has been restarted successfully. If the cluster

status changes to **Abnormal**, you are advised to contact the customer service for troubleshooting.

Rolling Restart

NOTICE

- Data may be lost during rolling restart. Exercise caution when performing this operation. Perform this operation in off-peak hours.
- Rolling restart is supported only when the number of nodes in a cluster is greater than or equal to three.
- When the data volume is large, rolling restart takes a long time.

1. Log in to the CSS management console.
2. Click **Clusters** to switch to the **Clusters** page. In the row where the cluster you want to restart is located, click **More > Restart** in the **Operation** column. The **Restart Cluster** page is displayed. Select **Rolling Restart**.

   During the rolling restart, the cluster is able to provide services, and only the node that is being restarted is unavailable. When a client fails to access one node, try other nodes.
3. Refresh the page and check the cluster status. During the restart, the cluster status is **Processing**, and the task status is **Restarting**. If the cluster status changes to **Available**, the cluster has been restarted successfully. If the cluster status changes to **Abnormal**, you are advised to contact the customer service for troubleshooting.

8.8 Migrating a Cluster

Cluster migration migrates data from a cluster to another one. In certain scenarios, for example, if demands cannot be met by directly changing specifications of the current cluster due to the growing business data, you can create a cluster of higher specifications and migrate all data of the current cluster to the new one. Alternatively, you can merge indices in two clusters to one cluster to satisfy your business needs. CSS enables cluster migration by using the index backup and restoration function, specifically, by restoring the snapshot of a cluster to the target cluster.

Prerequisites

- The source and target clusters are in the same region.
- The version of the target cluster is the same as or later than that of the source cluster.
- The number of nodes in the target cluster must be greater than half of the number of nodes in the source cluster.

Suggestions

- The number of nodes in the target cluster is no less than the number of replicas in the source cluster.

- The CPU, memory, and disk configurations of the target cluster are greater than or equal to those of the source cluster, minimizing service loss after migration.

In this section, assume that data of cluster **Es-1** is migrated to cluster **Es-2**. Cluster **Es-2** runs a version later than that of cluster **Es-1** and the number of nodes in cluster **Es-2** is greater than half of that in cluster **Es-1**.

Procedure

1. On the **Clusters** page, click **Es-1**. On the displayed page, click **Cluster Snapshots**.
2. Click **Create Snapshot** to manually create a snapshot. In the displayed dialog box, enter the snapshot name and click **OK**.

   If you use the index backup and restoration function for the first time, you need to perform basic configurations first. For details, see **Manually Creating a Snapshot**.

   Figure 8-13 Creating a snapshot

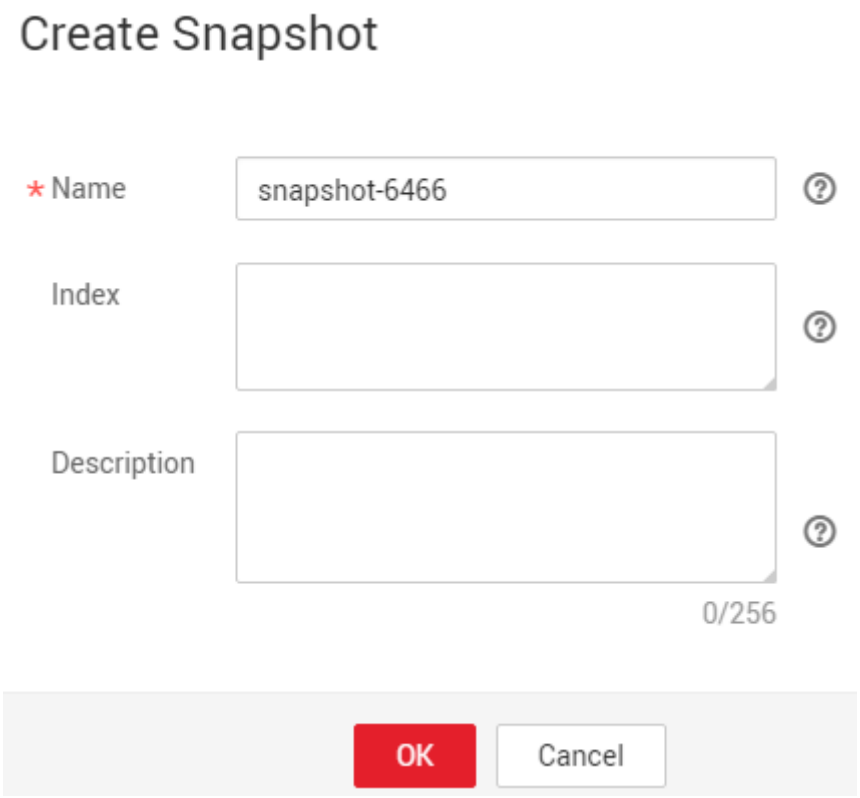
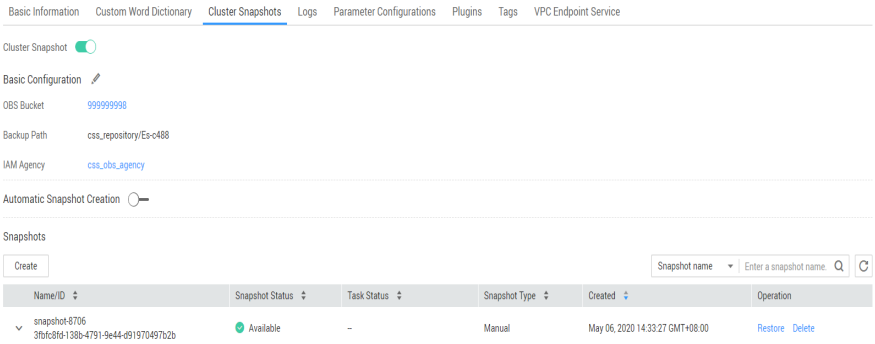


   Figure 8-14 shows the page displayed when the snapshot is created.

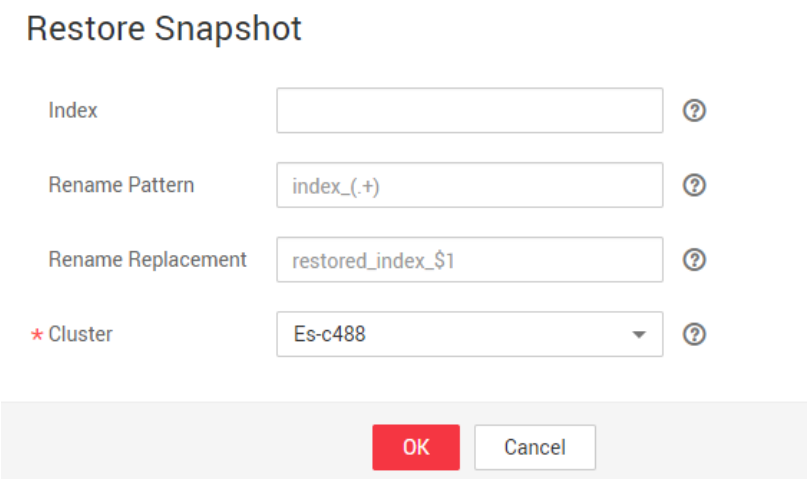
Figure 8-14 Successful snapshot creation



3. In the snapshot list, locate the row where the target snapshot resides and click **Restore** in the **Operation** column to restore data to cluster **Es-2**.
   - In the text box next to **Index**, enter *, indicating to restore data of all indices in cluster **Es-1**.
   - From the **Cluster** drop-down list, select **Es-2**.

   Click **OK**. You can also rename the restored index. For details, see **Index Backup and Restoration**.

Figure 8-15 Restoring data



4. After restoration is complete, data in cluster **Es-1** is migrated to cluster **Es-2**.

8.9 Deleting a Cluster

If you have completed data search and do not need a cluster, you can delete it. If you delete a cluster, snapshots created for the cluster are not deleted, but saved in the OBS bucket.

NOTE

After a cluster is deleted, its data cannot be recovered. Therefore, exercise caution when performing this operation.

Procedure

1. Log in to the CSS management console.
2. Click **Clusters** to switch to the **Clusters** page. In the row where the cluster you want to restart is located, click **More > Delete** in the **Operation** column.
3. In the dialog box that is displayed, click **Yes**.

8.10 Managing Tags

Tags are cluster identifiers. Adding tags to clusters can help you identify and manage your cluster resources.

You can add tags to a cluster when creating the cluster or add them on the details page of the created cluster.

Adding Tags to a Cluster

1. Log in to the CSS management console.
2. On the **Create Cluster** page, set **Advanced Settings** to **Custom**. Add tags for a cluster.

   You can select a predefined tag and set **Tag value** for the tag. You can click **View predefined tags** to switch to the **TMS** management console and view existing tags of the current user.

   You can also create new tags by specifying **Tag key** and **Tag value**.

   You can add a maximum of 10 tags for a CSS cluster. If the entered tag is incorrect, you can click **Delete** to the right of the tag to delete the tag. If you do not want to add tags, leave this parameter blank.

   Table 8-8 Naming rules for a tag key and value

Parameter	Description
Tag key	Cannot be left blank. Must be unique in a cluster. Contains a maximum of 36 characters. Can only consist of digits, letters, hyphens (-), and underscores (_).
Tag value	Can contain a maximum of 43 characters. Can only consist of digits, letters, hyphens (-), and underscores (_). Cannot be left blank.

Searching for Clusters by Tag

1. Log in to the CSS management console.
2. On the **Clusters** page, click **Search by Tag** in the upper right corner of the cluster list.

3. Enter the target tag key and value.

   You can select a tag key or tag value from their drop-down lists. When the tag key or tag value is exactly matched, the system can automatically locate the target cluster. If you enter multiple tags, their intersections are used to search for the cluster.

   You can add a maximum of 10 tags at one time.

4. Click **Search**.

   The system searches for the target cluster by tag key and value.

Tags Management

You can modify, delete, or add tags for a cluster.

1. Log in to the CSS management console.
2. On the **Clusters** page, click the name of a cluster for which you want to manage tags.

   The **Basic Information** page is displayed.

3. Select the **Tags** tab, then you can add, modify, or delete tags on the displayed page.
   - View

     On the **Tags** page, you can view details about tags of the cluster, including the number of tags and the key and value of each tag.
   - Add

     Click **Add Tag** in the upper left corner. In the displayed **Add Tag** dialog box, enter the key and value of the tag to be added, and click **OK**.
   - Modify

     You can only change the value of an existing tag.

     In the **Operation** column of a tag, click **Edit**. In the displayed **Edit Tag** page, enter a new tag value and click **OK**.
   - Delete

     In the **Operation** column of the tag, click **Delete**. After confirmation, click **Yes** on the displayed **Delete Tag** page.

8.11 Public IP Address Access

For a cluster that has the security mode enabled, you can access the cluster through the public IP address provided by the system.

Configuring the Public IP Address

1. Log in to the CSS management console.
2. On the **Create Cluster** page, enable **Security Mode**.

   You can enable **Security Mode** for clusters in Version 6.5.4 and later versions.

3. Select **Automatically assign** for **Public IP Address** and set related parameters.

Table 8-9 Public IP address access parameters

Parameter	Description
Bandwidth	Bandwidth of the public IP address access
Access Control	If you disable the access control, all IP addresses can access the cluster through the public IP address. If you enable the access control, only IP addresses in the whitelist can access the cluster through the public IP address.
Whitelist	IP address or IP address range allowed to access a cluster. Use commas (,) to separate multiple addresses.

Managing Public IP Addresses

You can modify, view the public IP address of, or disassociate the public IP address from a cluster, or configure the public IP address.

1. Log in to the CSS management console.
2. On the **Clusters** page that is displayed, click the name of the target cluster.
   - Configure the public IP address

     If you do not configure the public IP address during cluster creation, you can configure it on the **Basic Information** page after configuring a cluster.

     Click **Associate** next to **Public IP Address**, set the access bandwidth, and click **OK**.

     If the association fails, wait for several minutes and try again.
   - Modify

     For a cluster for which you have configured the public IP address, you can click **Edit** next to **Bandwidth** to modify the bandwidth, or you can click **Set** next to **Access Control** to set the access control function and the whitelist for access.
   - View

     On the **Basic Information** page, you can view the public IP address associated with the current cluster.
   - Disassociate

     To disassociate the public IP address, click **Disassociate** next to **Public IP Address**.

Accessing a Cluster Through the Public IP Address

After configuring the public IP address, you can use it to access the cluster. The access address is **https://*public IP address*:9200/*interface URL***.

8.12 Managing Logs

CSS provides log backup and query functions to help you locate faults. You can back up cluster logs to OBS buckets and download required log files to analyze and locate faults.

Enabling Log Management

1. Log in to the CSS management console.
2. On the **Clusters** page, click the name of the cluster for which you want to back up the logs. The **Basic Information** page is displayed.
3. Click the **Logs** tab, and enable the **Log Management** function.

    indicates that the log management function is disabled.  indicates that the log management function is enabled.
4. (Optional) After the log management function is enabled, CSS automatically creates the OBS bucket, backup path, and IAM agency for you to back up logs. The automatically created OBS bucket, backup path, and IAM agency are displayed on the page. If you want to change the OBS bucket, backup path, and IAM agency, click  on the right of **Log Backup Configuration**.

   In the displayed **Edit Log Backup Configuration** dialog box, you can either select an existing OBS bucket and an IAM agency or create an OBS bucket and an IAM agency. To create an OBS bucket, click **Create Bucket**. To create an IAM agency, click **Create IAM Agency**. For details, see **Creating a Bucket** and **Creating an Agency**.

   Table 8-10 Parameter description

Parameter	Description	Remarks
OBS Bucket	Name of the OBS bucket used for storing logs	The OBS bucket must be in the same region as that of the cluster.
Backup Path	Storage path of logs in the OBS bucket	The backup path configuration rules are as follows: • The backup path cannot contain the following characters: \:*?"<>| • The backup path cannot start with a slash (/). • The backup path cannot start or end with a period (.). • The total length of the backup path cannot exceed 1,023 characters.

Parameter	Description	Remarks
IAM Agency	IAM agency authorized by the current account to CSS to access or maintain data stored in OBS.	The following conditions must be met for existing IAM agencies or those to be created: • Select **Cloud Service** for the **Agency Type**. • Select **Elasticsearch** for **Cloud Service**. • The agency has the **Tenant Administrator** permission for the **OBS** project in **Global service**.

5. Back up logs.

   a. Automatically backing up logs

      Click the icon on the right of **Auto Backup** to enable the auto backup function.

       indicates that the auto backup function is enabled, and  indicates that the auto backup function is disabled.

      After enabling the auto backup function, set the backup start time on the **Edit Auto Backup Policy** page. When the setting is successful, the system automatically backs up logs at the scheduled time.

   b. Manually backing up logs

      On the **Log Backup** tab page, click **Start Backup**. On the displayed page, click **Yes** to start backup.

      If **Task Status** in the log backup list is **Successful**, the backup is successful.

      NOTE

      All logs in the cluster are copied to a specified OBS path. You can view or download log files in the path of the OBS bucket.

6. Query logs.

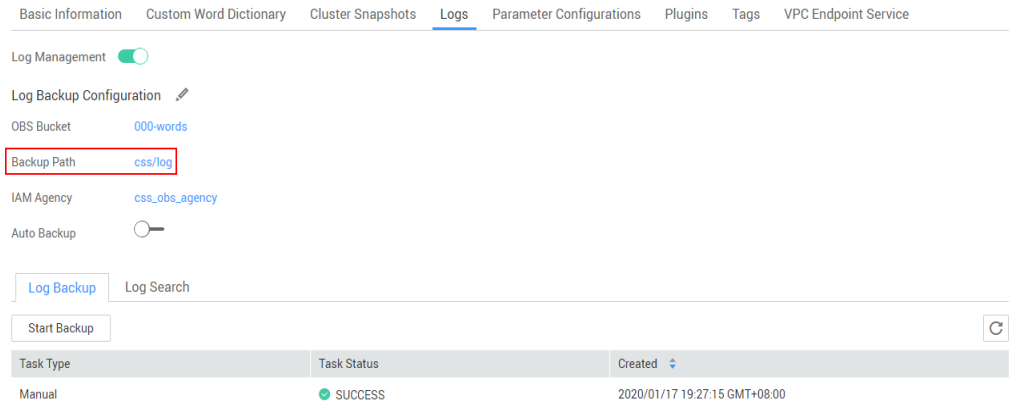
   You can query logs of each node in a cluster based on the node, log type, and log level. The following log types are supported: running logs, index slow logs, search slow logs, and deprecation logs. When you query logs, the latest 10,000 logs are matched. A maximum of 100 logs are displayed.

   On the **Log Search** page, select the desired node, log type, and log level, and click  . The search results are displayed.

Viewing Logs

After logs are backed up, you can click **Backup Path** to go to the OBS console and view the logs.

Figure 8-16 Viewing logs



Backed up logs mainly include deprecation logs, run logs, index slow logs, and search slow logs. **Table 8-11** lists the storage types of the OBS bucket.

Table 8-11 Log list

Log File	Description
clustername_deprecation.log	Deprecation logs
clustername_index_indexing_slowlog.log	Search slow logs
clustername_index_search_slowlog.log	Index slow logs
clustername.log	Elasticsearch run logs
clustername_access.log	Access logs
clustername_audit.log	Audit logs

8.13 Managing Plugins

CSS allows you to view default and custom plugins. If the plugin provided by CSS cannot meet your requirements, you can install, uninstall, or delete the plugin based on your needs.

1. Log in to the CSS management console.
2. On the **Clusters** page, click the name of a cluster for which you want to install a plugin.

   The **Basic Information** page is displayed.
3. Click the **Plugins** tab.
4. View the information about default plugins.

   On the **Default Plugins** page, view default plugins supported by the current version.

Table 8-12 Cluster versions supported by default plugins

Plugin	Supported Cluster Version
analysis-dynamic-synonym	5.5.1, 6.2.3, 6.5.4, and 7.1.1
analysis-icu	6.2.3, 6.5.4, and 7.1.1
analysis-ik	5.5.1, 6.2.3, 6.5.4, and 7.1.1
analysis-pinyin	5.5.1, 6.2.3, 6.5.4, and 7.1.1
analysis-poisson	6.2.3
analysis-stconvert	5.5.1, 6.2.3, 6.5.4, and 7.1.1
lasthit	5.5.1 and 6.2.3
repository-obs	5.5.1, 6.2.3, 6.5.4, and 7.1.1
vector-search	6.2.3
opendistro_alerting	6.5.4 and 7.1.1
opendistro_security	6.5.4 and 7.1.1
opendistro_sql	6.5.4 and 7.1.1
analysis-kuromoji	6.5.4 and 7.1.1
analysis-nori	6.5.4 and 7.1.1
ingest-attachment	6.5.4 and 7.1.1

5. Install custom plugins.

   NOTE

   If you want to use custom plugins, submit a service ticket to apply for permissions. Keep custom plugins available and secure because they may affect cluster stability.

   a. On the **Custom Plugins** page, click **Upload** to upload the desired plugin from the OBS bucket to the cluster.

      - **OBS Bucket**: OBS bucket where the plugin to be installed is stored. If no OBS bucket is available, click **Create Bucket** to create one. For details, see **Creating a Bucket**. The OBS bucket to be created must be in the same region as the cluster.
      - **Plugin**: Plugin to be uploaded. Select a .zip file.

   b. Click **OK**.

      After the upload completes, view the plugin information on the **Plugins** page. You can also view plugin operation records in the upper right corner of the page.

Figure 8-17 Viewing operation records

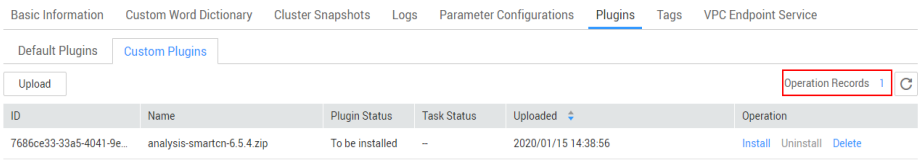


Table 8-13 Viewing the plugin information

Parameter	Description
ID	Plugin ID, which is automatically generated
Name	Name of the plugin to be uploaded The plugin name must be unique.
Plugin Status	Plugin status. The options are as follows: • **To be installed**: indicates the plugin has been uploaded but not installed. • **Installed and to be effective upon cluster restart**: If this status is displayed, restart the cluster for the plugin to take effect. • **Installed**: If this status is displayed after restarting the cluster, the plugin can work properly. • **Uninstalled and to be effective upon cluster restart**: If this status is displayed, restart the cluster for the plugin to take effect. • **Uninstalled**: If this status is displayed after restarting the cluster, the plugin is not available.
Task Status	Running status of the plugin. The options are as follows: • **Uploading** • **Installing** • **Uninstalling** • **Deleting**
Uploaded	Time when the plugin is uploaded
Operation	• **Install**: After uploading the plugin, click **Install**. **Task Status** changes to **Installing**. • **Uninstall**: If you want to uninstall the plugin, click **Uninstall**. **Task Status** changes to **Uninstalling**. The uninstalled plugin is still displayed in the list. If you want to use it again, install it directly. • **Delete**: If you want to delete the plugin, click **Delete**. **Task Status** changes to **Deleting**. If you delete the plugin, you need to upload it before installing it again.

NOTE

- After you install plugins such as **readonlyrest** and **siren-federate**, manually modify the configurations. You cannot install the plugins by yourself. If you need to install plugins, contact the technical support.
- Custom plugins can be automatically uploaded and installed after the cluster scale-out.
- If a plugin fails to be installed, view the failure cause on the **Operation Records** page. If you still cannot locate the fault, contact technical support.

Table 8-14 Parameters on the Operation Records page

Parameter	Description
Name/ID	Name of the plugin you perform operations on
Operation Type	Operations performed on the plugin, including **Upload**, **Install**, **Uninstall**, and **Delete**.
Task Status	Status of a task, including **Successful**, **Failed**, and **Running**.
Created	Time when a task is created
Completed	Time when a task is completed
Error Message	Message reported when a task fails Common error messages are as follows: • If the error message is **Instance: xx.xx.xx.xx. ERROR: `elasticsearch` directory is missing in the plugin zip.**, Check whether the plugin file is a standard .zip file. • If the error message is **Instance: xx.xx.xx.xx.. Exception in thread "main" java.lang.IllegalArgumentException: plugin [analysis-pinyin] is incompatible with version [5.5.1]; was designed for version [6.2.3] at**, the plugin version does not match the cluster version. • If you still cannot locate the fault, contact technical support.

8.14 Hot and Cold Data Storage

CSS provides you with cold data nodes. You can store data that requires query response in seconds on high-performance nodes and store data that requires query response in minutes on cold data nodes with large capacity and low specifications.

NOTE

- When creating a cluster, you need to configure nodes as data nodes. Only after you select the cold data node, data nodes become hot nodes.
- You can select the cold data node, master node, and client node at the same time.
- You can increase nodes and expand storage capacity of cold data nodes. The maximum storage capacity is determined by the node specifications. Local disks do not support storage capacity expansion.

Hot and Cold Data Node Switchover

After you enable the cold data node function, the cold data node is labeled with **cold**. In addition, data nodes are labeled with **hot** and become hot nodes. You can specify index to distribute data to the cold or hot nodes.

You can configure a template to store index to the specified cold or hot node.

The following figure shows this process. Log in to the **Kibana Console** page of the cluster, modify the template by configuring the index starting with **myindex**, and store the indice on the cold node. In this case, the **myindex*** date is stored on the cold data node by modifying the template.

```
PUT _template/test
{
   "order": 1,
   "template": "myindex*",
   "settings": {
      "index": {
         "refresh_interval": "30s",
         "number_of_shards": "3",
         "number_of_replicas": "1",
         "routing.allocation.require.box_type": "cold"
      }
   }
}
```

You can also perform operations on the created index.

```
PUT myindex/_settings
 {
      "index.routing.allocation.require.box_type": "cold"
   }
```

You can also cancel the configurations of hot and cold data nodes.

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

8.15 Configuring Parameters

CSS allows you to modify configurations in the **elasticsearch.yml** file on the CSS console. You need to restart the cluster for the modifications to take effect.

Modifying Parameter Configurations

1. Log in to the CSS management console.
2. On the **Clusters** page, click the name of the cluster for which you want to modify parameter configurations. The **Basic Information** page is displayed.

3. Click **Parameter Configurations** and modify parameters of the corresponding module based on your needs.

Table 8-15 Module parameters

Module Name	Parameter	Description
Cross-domain Access	http.cors.allow-credentials	Whether to return the Access-Control-Allow-Credentials of the header during cross-domain access Value: **true** and **false** Default value: **false**
	http.cors.allow-origin	Origin IP address allowed for cross-domain access, for example, 122.122.122.122:9200
	http.cors.max-age	Cache duration of the browser. The cache is automatically cleared after the time range you specify. Unit: s Default value: **1,728,000**
	http.cors.allow-headers	Headers allowed for cross-domain access, including X-Requested-With, Content-Type, and Content-Length. Separated headers with commas (,) and spaces.
	http.cors.enabled	Whether to allow cross-domain access Value: **true** and **false** Default value: **false**
	http.cors.allow-methods	Methods allowed for cross-domain access, including OPTIONS, HEAD, GET, POST, PUT, and DELETE. Separated methods with commas (,) and spaces.
Reindexing	reindex.remote.whitelist	Configured for migrating data from the current cluster to the target cluster through the reindex API. The example value is 122.122.122.122:9200.
Custom Cache	indices.queries.cache.size	Cache size in the query phase Value range: 1 to 100 Unit: % Default value: **10%**
Queue Size in a Thread Pool	thread_pool.bulk.queue_size	Queue size in the bulk thread pool. The value is an integer. Default value: **200** This parameter is displayed when the cluster version is earlier than 7.*x*.

Module Name	Parameter	Description
	thread_pool.write.queue_size	Queue size in the write thread pool. The value is an integer. Default value: **200** This parameter is displayed when the cluster version is later than 7.*x*.
	thread_pool.force_merge.size	Queue size in the force merge thread pool. The value is an integer. Default value: **1**
Customize	You can add parameters based on your needs.	Customized parameters **NOTE** Enter multiple values in the format as **[value1, value2, value3...]**. Separate values by commas (,) and spaces.

4. Click **Confirm**.

   In the displayed **Confirm Modification** dialog box, select the box indicating "The modification will take effect only when you restart the cluster." and click **Yes**.

   You can view the modification records on the displayed page. The system displays a maximum of 20 records.

   NOTE

   If you do not restart the cluster after modifying the parameter configurations, **Configuration unupdated** is displayed in the **Task Status** column on the **Clusters** page.

   If you restart the cluster after the modification, **Task Status** displays **Configuration error**, the parameter configuration file fails to be modified.

8.16 VPC Endpoint Service

The VPC endpoint service allows you to access the cluster through the private domain name. When the VPC endpoint service is enabled, the system creates a VPC endpoint for you by default. The VPC endpoint service is now available for the open beta test (OBT). Therefore, you need to apply for the OBT before using it.

Enabling the VPC Endpoint Service

1. Log in to the CSS management console.
2. On the **Create Cluster** page, set **Advanced Settings** to **Custom**. Enable the VPC endpoint service.
   - **Create Private Domain Name**: If this function is enabled, the system automatically creates a private domain name for you, which you can use to access the cluster.

- **Whitelist**: Add an authorized account ID to the VPC endpoint service whitelist, then you can access the cluster using the domain name or the node IP address.

  For multiple accounts, click  to add them. You can also click **Delete** in the **Operation** column to delete the accounts that are not allowed for access.

NOTE

- If the authorization account ID is set to *, all users are allowed to access the cluster.
- You can view authorized account IDs on the **My Credentials** page.

Managing VPC Endpoint Service

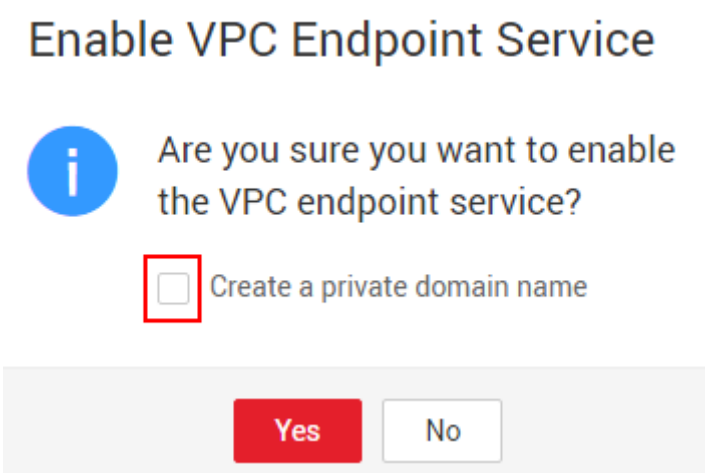
In addition to enable VPC Endpoint Service while creating a cluster, you can enable it by performing the following steps after cluster creation.

1. Log in to the CSS management console.
2. On the **Clusters** page, click the name of the target cluster.
3. Click the **VPC Endpoint Service** tab, and turn on the button next to **VPC Endpoint Service**.

    indicates disabling the VPC endpoint service and  indicates enabling the VPC endpoint service.

   In the displayed dialog box, you can determine whether to enable the private domain name as required. After a private domain name is created, you can access the cluster using the private domain name.

Figure 8-18 Enabling the VPC endpoint service



NOTE

- After you enable the VPC endpoint service, you can use the private domain name or node IP address generated by the endpoint to access the cluster. For details, see **Accessing the Cluster Using the Private Domain Name or Node IP Address**.
- If you disable the VPC endpoint service, all users cannot access the cluster using the private domain name.

4. Click **Yes** to enable the VPC endpoint service.
5. (Optional) Click **Update** next to **Whitelist** to update the existing whitelist.
6. Manage connections of the VPC endpoint.

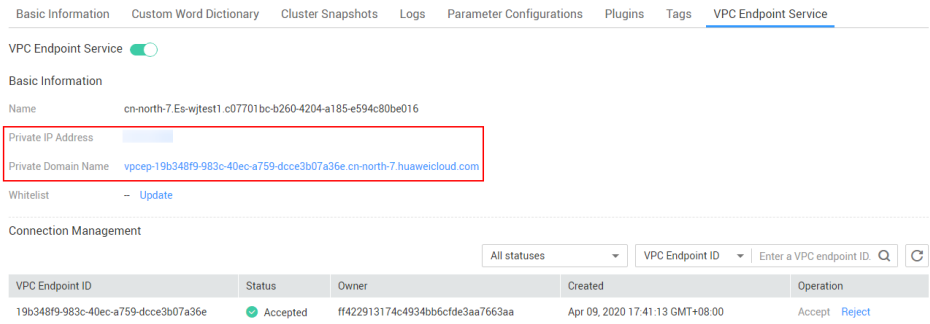
   The **VPC Endpoint Service** page displays all VPC endpoints connected to the current VPC endpoint service. You can accept or reject the connection with these endpoints. If you reject the connection with a VPC endpoint, you cannot access the cluster through the private domain name generated by the VPC endpoint.

Accessing the Cluster Using the Private Domain Name or Node IP Address

1. Obtain the private domain name or node IP address.
   - Current user

     Log in to the CSS console, click the target cluster name and go to the cluster details page. Click the **VPC Endpoint Service** tab and view the private domain name, as shown in **Figure 8-19**.

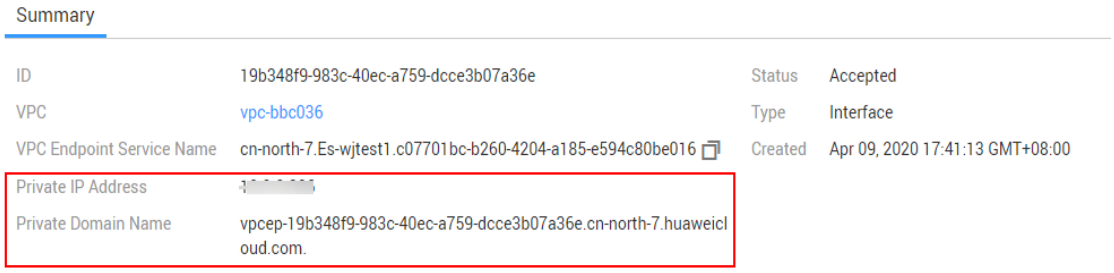
     Figure 8-19 Viewing the private domain name and node IP address (1)

   - Other user

     If you have applied for the VPC endpoint service, log in to the **VPC Endpoint console** and click the target ID to go to the **Summary** page and view the private domain name. See **Figure 8-20**.

Figure 8-20 Viewing the private domain name and node IP address (2)



2. Run the cURL command to execute the API or call the API by using a program before accessing the cluster. For details about Elasticsearch operations and APIs, see the **Elasticsearch Reference**.

   The ECS must meet the following requirements:

   - Robust disk space is allocated for the ECS.
   - The ECS and the cluster must be in the same VPC. After enabling the VPC endpoint service, you can access the cluster accross the VPC.
   - The security group of the ECS must be the same as that of the cluster.

     If this requirement is not met, modify the ECS security group or configure the inbound and outbound rules of the ECS security group to allow the ECS security group to be accessed by all security groups of the cluster. For details, see **Configuring Security Group Rules**.
   - For security group rule settings of the target CSS cluster, set **Protocol** to **TCP** and **Port Range** to **9200** or a port range including port **9200** for both the outbound and inbound directions.

   For example, run the following cURL command to view the index information in the cluster. In this example, the private network address is **vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com** and port **9200** is used to access the cluster.

   - If the cluster you access does not have the security mode enabled, run the following command:

     ```
     curl 'http://vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com:9200/_cat/indices'
     ```
   - If the cluster you access has the security mode enabled, access the cluster using HTTPS and add the username, password and **-u** to the cURL command.

     ```
     curl -u username:password -k 'https://vpcep-7439f7f6-2c66-47d4-b5f3-790db4204b8d.region01.huaweicloud.com:9200/_cat/indices'
     ```

8.17 Kibana Public Access

For CSS clusters with security mode enable, you can access Kibana through the Internet after configuring Kibana public access. For a cluster with security mode enabled, CSS allows you to enable Kibana public access and configure access bandwidth. After the configuration is complete, the cluster obtains a Kibana public IP address, with which you can access Kibana of this cluster.

You can configure Kibana public access when creating a cluster or configure this function after enabling security mode for a cluster.

NOTE

Clusters purchased before rollout of this feature do not support this function.

Configuring Kibana Public Access When Creating a Cluster

1. Log in to the CSS management console.
2. On the **Create Cluster** page, enable **Security Mode**.

   You can enable **Security Mode** for clusters in Version 6.5.4 and later versions.

3. Set **Advanced Settings** to **Custom**, enable **Kibana Public Access**, and set related parameters.

   Table 8-16 Kibana public access parameters

Parameter	Description
Bandwidth	Bandwidth for accessing Kibana with the public IP address Value range: 1 to 100 Unit: Mbit/s
Access Control	If you disable this function, any IP addresses can access the cluster through the public IP address. If you enable this function, only IP addresses or IP address ranges in the whitelist can access the cluster through the public IP address.
Whitelist	IP address or IP address range allowed to access a cluster. Use commas (,) to separate multiple addresses. You are advised to enable this function.

   After the cluster is created, click the cluster name to go to the **Basic Information** page. On the **Kibana Public Access** page, you can view the Kibana public IP address.

Configuring Kibana Public Access for an Existing Cluster

You can enable, disable, modify, and view Kibana public access for an existing cluster with security mode enabled.

1. Log in to the CSS management console.
2. On the **Clusters** page, click the name of the target cluster.
3. Click the **Kibana Public Access** tab. Turn on the **Kibana Public Access** switch to enable the Kibana public access function.

    indicates that Kibana public access is disabled.  indicates that Kibana public access is enabled.
4. On the displayed page, set related parameters.

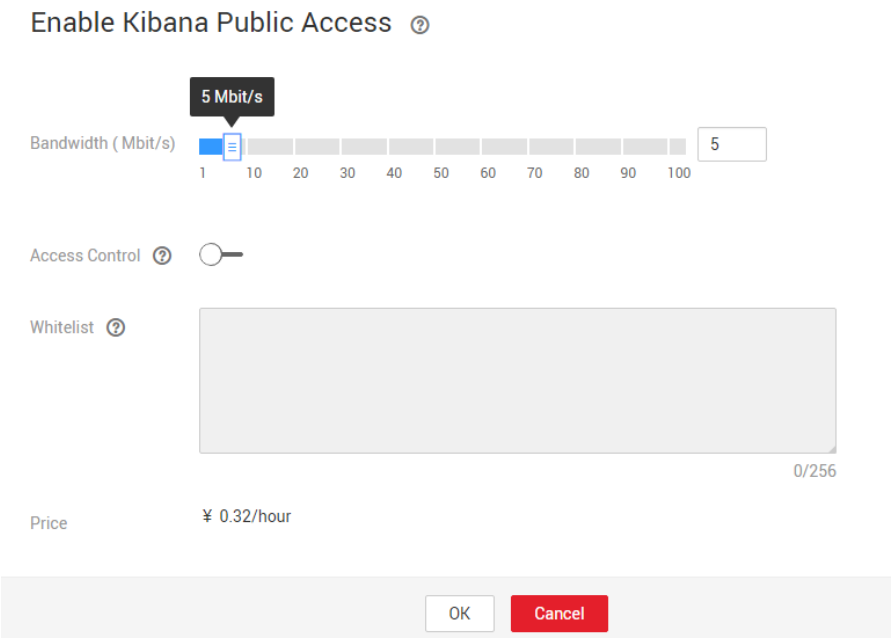


Table 8-17 Kibana public access parameters

Parameter	Description
Bandwidth	Bandwidth for accessing Kibana with the public IP address Value range: 1 to 100 Unit: Mbit/s
Access Control	If you disable this function, any IP addresses can access the cluster through the public IP address. If you enable this function, only IP addresses or IP address ranges in the whitelist can access the cluster through the public IP address.
Whitelist	IP address or IP address range allowed to access a cluster. Use commas (,) to separate multiple addresses. You are advised to enable this function.

5. After the configuration is completed, click **OK**.

Modifying Kibana Public Access

For clusters configured with Kibana public access, CSS allows you to modify bandwidth and access control.

- Modifying bandwidth

  Click **Modify** on the right of **Bandwidth**. On the **Modify Bandwidth** page, modify the bandwidth and click **OK**.
- Modifying access control

  Click **Modify** on the right of **Access Control**. On the **Modify Access Control** page, set **Access Control** and **Whitelist**, and click **OK**.

Accessing Kibana with the Public IP Address

After configuring Kibana public access, you will obtain a public IP address, with which you can access Kibana of this cluster.

1. Log in to the CSS management console.
2. On the **Clusters** page, click the name of the target cluster.
3. Click the **Kibana Public Access** tab to obtain the Kibana public IP address.
4. Use this IP address to access Kibana of this cluster through the Internet.

9 Monitoring a Cluster

9.1 Supported Metrics

Function

This section describes monitoring metrics reported by CSS to Cloud Eye as well as their namespaces and dimensions. You can use the management console or **APIs** provided by Cloud Eye to view the monitoring metrics and alarms generated for CSS.

Namespace

SYS.ES

Monitoring Metrics

Table 9-1 Monitoring metrics

Metric ID	Metric	Description	Value Range	Measurement Object & Dimension	Monitoring Period (Raw Data)
status	Cluster Health Status	Health status of the monitored object	0,1,2 **0**: All primary and replica shards are allocated. Your cluster is 100% operational. **1**: All primary shards are allocated, but at least one replica is missing. No data is missing, so search results will still be complete. However, your high availability is compromised to some degree. If more shards disappear, you might lose data. Think of this state as a warning that should prompt investigation. **2**: Data missing occurs and the cluster fails to work.	CSS cluster	1 minute

Metric ID	Metric	Description	Value Range	Measurement Object & Dimension	Monitoring Period (Raw Data)
disk_util	Disk Usage	Disk usage of the monitored object Unit: Percent	0-100%	CSS cluster	1 minute
max_jvm_heap_usage	Max. JVM Heap Usage	Maximum JVM heap usage of nodes in a CSS cluster Unit: Percent	0-100%	CSS cluster	1 minute
max_jvm_young_gc_time	Max. JVM Young GC Duration	Maximum JVM Young GC duration of nodes in a CSS cluster Unit: ms	≥ 0 ms	CSS cluster	1 minute
max_jvm_young_gc_count	Max. JVM Young GC Count	Maximum JVM Young GC count of nodes in a CSS cluster	≥ 0	CSS cluster	1 minute
max_jvm_old_gc_time	Max. JVM Old GC Duration	Maximum JVM Old GC duration of nodes in a CSS cluster Unit: ms	≥ 0 ms	CSS cluster	1 minute
max_jvm_old_gc_coun	Max. JVM Old GC Count	Maximum JVM Old GC count of nodes in a CSS cluster	≥ 0	CSS cluster	1 minute
total_fs_size	Total Size of File Systems	Total size of file systems in a CSS cluster Unit: byte	≥ 0 bytes	CSS cluster	1 minute

Metric ID	Metric	Description	Value Range	Measurement Object & Dimension	Monitoring Period (Raw Data)
free_fs_size	Available Size of File Systems	Available size of file systems in a CSS cluster Unit: byte	≥ 0 bytes	CSS cluster	1 minute
max_cpu_usage	Max. CPU Usage	Maximum CPU usage of nodes in a CSS cluster Unit: Percent	0-100%	CSS cluster	1 minute
max_cpu_time_of_jvm_process	Max. CPU Time of JVM Process	Maximum CPU time of node JVM processes in a CSS cluster Unit: ms	≥ 0 ms	CSS cluster	1 minute
max_virtual_memory_size_of_jvm_process	Max. Virtual Memory Size of JVM Process	Maximum virtual memory size of node JVM processes in a CSS cluster Unit: byte	≥ 0 bytes	CSS cluster	1 minute
max_current_opened_http_count	Current Max. Opened HTTP Connections	Maximum number of HTTP connections that are currently opened for nodes in a CSS cluster	≥ 0	CSS cluster	1 minute
max_total_opened_http_count	Total Max. Opened HTTP Connections	Maximum number of HTTP connections that have been opened for nodes in a CSS cluster	≥ 0	CSS cluster	1 minute

Metric ID	Metric	Description	Value Range	Measurement Object & Dimension	Monitoring Period (Raw Data)
indices_count	Indices	Number of indices in a CSS cluster	≥ 0	CSS cluster	1 minute
total_shards_count	Shards	Number of shards in a CSS cluster	≥ 0	CSS cluster	1 minute
primary_shards_count	Primary Shards	Number of primary shards in a CSS cluster	≥ 0	CSS cluster	1 minute
docs_count	Documents	Number of documents in a CSS cluster	≥ 0	CSS cluster	1 minute
docs_deleted_count	Deleted Documents	Number of documents deleted in a CSS cluster	≥ 0	CSS cluster	1 minute
nodes_count	Nodes	Number of nodes in a CSS cluster	≥ 0	CSS cluster	1 minute
data_nodes_count	Data Nodes	Number of data nodes in a CSS cluster	≥ 0	CSS cluster	1 minute
coordinating_nodes_count	Coordinating Nodes	Number of coordinating nodes in a CSS cluster	≥ 0	CSS cluster	1 minute
master_nodes_count	Master Nodes	Number of master nodes in a CSS cluster	≥ 0	CSS cluster	1 minute
ingest_nodes_count	Client Nodes	Number of client nodes in a CSS cluster	≥ 0	CSS cluster	1 minute

Dimensions

Table 9-2 Dimension description

Key	Value
cluster_id	CSS cluster

9.2 Creating Alarm Rules

You can create the alarm rules for cluster metrics on the Cloud Eye management console. If the monitored metrics meet the specified alarm rule, alarms are reported. In this case, you can learn about cluster exceptions in time and take proper measures to prevent business loss.

Procedure

1. Log in to the management console.
2. Choose **Service List** > **Management & Deployment** > **Cloud Eye**.
3. In the left navigation pane, choose **Alarm Management > Alarm Rules**.
4. On the displayed **Alarm Rules** page, click **Create Alarm Rule**.
5. In the displayed **Create Alarm Rule** dialog box, set parameters as prompted.

   You can create an alarm rule for a specific metric or use the alarm template to create alarm rules in batches for multiple cloud service instances. In this example, assume that you use the alarm template to create the alarm rule for the CSS cluster.

   a. Configure the name and description of an alarm rule.

   Figure 9-1 Configuring the alarm rule name and description

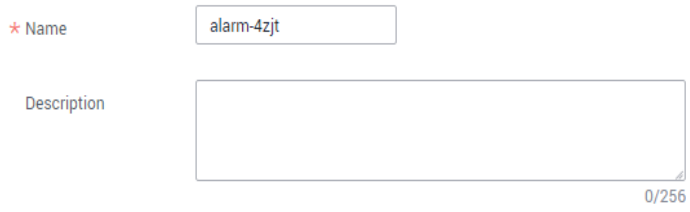
   

   Table 9-3 Parameter description

Parameter	Description	Example Value
Name	Name of the alarm rule. The system generates a name randomly but you can change it.	alarm-p8v9

Parameter	Description	Example Value
Description	Alarm rule description. This parameter is optional.	-

b. Select a monitored object and set alarm content parameters.

Figure 9-2 Configuring alarm content

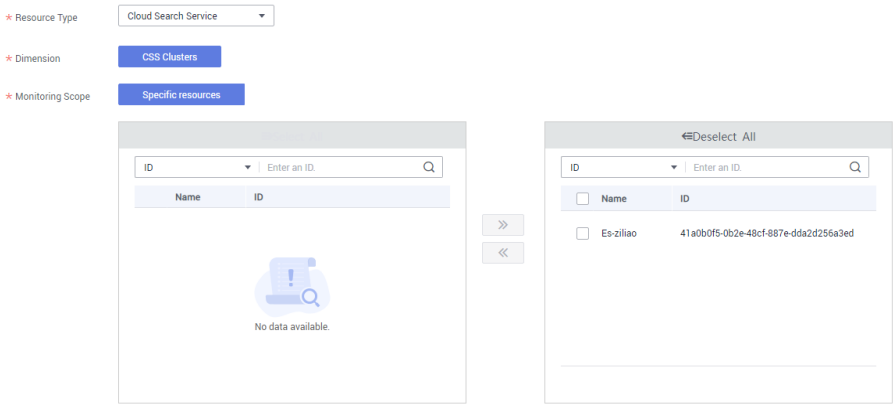


Table 9-4 Parameters for configuring the alarm content

Parameter	Description	Example Value
Resource Type	Name of the service for which the alarm rule is configured	Cloud Search Service
Dimension	Metric dimension of the alarm rule	CSS Clusters
Monitoring Scope	Name of the cluster to be monitored. Select one or more monitored objects and click  to synchronize the monitored objects to the dialog box on the right.	Es-ziliao

c. Specify **Method**, **Template**, and **Alarm Notification**.

Table 9-5 Parameter description

Parameter	Description	Example Value
Method	Select **Use template** or **Create manually** as required. If you set **Monitoring Scope** to **Specific resources**, you can set **Method** to **Use template**. For details about manually creating the monitoring scope, see **Creating an Alarm Rule**.	Create manually
Template	Select the template to be imported.	-
Alarm Notification	If this function is enabled, specify **Validity Period**, **Notification Object**, and **Trigger Condition**.	-

d. Click **Create**.

After an alarm rule is successfully created, it will be displayed in the alarm rule list

9.3 Viewing Metrics

Cloud Eye provides daily monitoring on core cluster metrics of CSS. You can log in to the Cloud Eye management console to view cluster metrics.

Cloud Eye only monitors clusters that have been successfully created in real time.

Prerequisites

- The cluster status is **Available** or **Processing**.

  NOTE

  You cannot view the metrics of deleted clusters or those whose **Status** is **Abnormal** or **Creating** on the Cloud Eye management console. If the status of a cluster changes from **Abnormal** or **Creating** to **Available**, you can view its metrics in real time after approximately 10 minutes.

- The cluster has been running for about 10 minutes.
- Alarm rules have been created.

Procedure

1. Log in to the management console.
2. Under **Management & Deployment**, click **Cloud Eye**.
3. In the left navigation pane, choose **Cloud Service Monitoring > Cloud Search Service**.
4. Locate the row where the target cluster resides, click **View Metric** in the **Operation** column.

5. Click the tab for the time range to be viewed.
6. View the monitoring data.

10 Elasticsearch SQL

For Elasticsearch 6.5.4 and later versions, Open Distro for Elasticsearch SQL lets you write queries in SQL rather than the Elasticsearch query domain-specific language (DSL).

If you are already familiar with SQL and do not want to learn the query DSL, this feature is a great option.

Basic Operations

To use this function, send requests to the **_opendistro/_sql** URI. You can use a request parameter or the request body (recommended).

```
GET https://<host>:<port>/_opendistro/_sql?sql=select * from my-index limit 50
POST https://<host>:<port>/_opendistro/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

You can use the **curl** command:

```
curl -XPOST https://localhost:9200/_opendistro/_sql -u username:password -k -d '{"query": "SELECT * FROM kibana_sample_data_flights LIMIT 10"}' -H 'Content-Type: application/json'
```

By default, JSON is returned for query. You can also return data in CSV format. You need to set the format parameter.

```
POST _opendistro/_sql?format=csv
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

When data is returned in CSV format, each row corresponds to a document and each column corresponds to a field.

Supported Operations

Open Distro for Elasticsearch supports the following SQL operations: statements, conditions, aggregations, include and exclude fields, common functions, joins, and show.

- Statements

Table 10-1 Statements

Statement	Example
Select	SELECT * FROM my-index
Delete	DELETE FROM my-index WHERE _id=1
Where	SELECT * FROM my-index WHERE ['field']='value'
Order by	SELECT * FROM my-index ORDER BY _id asc
Group by	SELECT * FROM my-index GROUP BY range(age, 20,30,39)
Limit	SELECT * FROM my-index LIMIT 50 (default is 200)
Union	SELECT * FROM my-index1 UNION SELECT * FROM my-index2
Minus	SELECT * FROM my-index1 MINUS SELECT * FROM my-index2

NOTE

As with any complex query, large UNION and MINUS statements can strain or even crash your cluster.

- Conditions

Table 10-2 Conditions

Condition	Example
Like	SELECT * FROM my-index WHERE name LIKE 'j%'
And	SELECT * FROM my-index WHERE name LIKE 'j%' AND age > 21
Or	SELECT * FROM my-index WHERE name LIKE 'j%' OR age > 21
Count distinct	SELECT count(distinct age) FROM my-index
In	SELECT * FROM my-index WHERE name IN ('alejandro', 'carolina')
Not	SELECT * FROM my-index WHERE name NOT IN ('jane')
Between	SELECT * FROM my-index WHERE age BETWEEN 20 AND 30
Aliases	SELECT avg(age) AS Average_Age FROM my-index
Date	SELECT * FROM my-index WHERE birthday='1990-11-15'
Null	SELECT * FROM my-index WHERE name IS NULL

- Aggregations

  Table 10-3 Aggregations

Aggregation	Example
avg()	SELECT avg(age) FROM my-index
count()	SELECT count(age) FROM my-index
max()	SELECT max(age) AS Highest_Age FROM my-index
min()	SELECT min(age) AS Lowest_Age FROM my-index
sum()	SELECT sum(age) AS Age_Sum FROM my-index

- Include and exclude fields

  Table 10-4 Include and exclude fields

Pattern	Example
include()	SELECT include('a*'), exclude('age') FROM my-index
exclude()	SELECT exclude('*name') FROM my-index

- Functions

  Table 10-5 Functions

Function	Example
floor	SELECT floor(number) AS Rounded_Down FROM my-index
trim	SELECT trim(name) FROM my-index
log	SELECT log(number) FROM my-index
log10	SELECT log10(number) FROM my-index
substring	SELECT substring(name, 2,5) FROM my-index
round	SELECT round(number) FROM my-index
sqrt	SELECT sqrt(number) FROM my-index
concat_ws	SELECT concat_ws(' ', age, height) AS combined FROM my-index

Function	Example
/	SELECT number / 100 FROM my-index
%	SELECT number % 100 FROM my-index
date_format	SELECT date_format(date, 'Y') FROM my-index

NOTE

You must enable fielddata in the document mapping for most string functions to work properly.

- Joins

Table 10-6 Joins

Join	Example
Inner join	SELECT p.firstname, p.lastname, p.gender, dogs.name FROM people p JOIN dogs d ON d.holdersName = p.firstname WHERE p.age > 12 AND d.age > 1
Left outer join	SELECT p.firstname, p.lastname, p.gender, dogs.name FROM people p LEFT JOIN dogs d ON d.holdersName = p.firstname
Cross join	SELECT p.firstname, p.lastname, p.gender, dogs.name FROM people p CROSS JOIN dogs d

For details about the constraints and limitations, see **Joins**.

- Show

  Show commands show you indices and mappings that match an index pattern. You can use ***** or **%** for wildcards.

Table 10-7 Show

Show	Example
Show tables like	SHOW TABLES LIKE logs-*

Joins

Open Distro for Elasticsearch SQL supports inner joins, left outer joins and cross joins. Joins have the following constraints:

- You can only join two indices.
- You must use an alias for an index (for example, people p).
- In an ON clause, you can only use the AND conditions.
- In a WHERE statement, do not combine trees that contain multiple indices. For example, the following statement works:
  ```
  WHERE (a.type1 > 3 OR a.type1 < 0) AND (b.type2 > 4 OR b.type2 < -1)
  ```
  The following statement does not:
  ```
  WHERE (a.type1 > 3 OR b.type2 < 0) AND (a.type1 > 4 OR b.type2 < -1)
  ```
- You cannot use GROUP BY or ORDER BY to obtain results.
- LIMIT with OFFSET (for example, LIMIT 25 OFFSET 25) is not supported.

JDBC Driver

The Java Database Connectivity (JDBC) driver allows you to integrate Open Distro for Elasticsearch with your business intelligence (BI) applications.

For details about how to download and use JAR files, see **GitHub Repositories**.

11 Querying Cluster Logs

11.1 Key Operations Recorded by CTS

Cloud Trace Service (CTS) is available on the public cloud platform. With CTS, you can record operations associated with CSS for later query, audit, and backtrack operations.

Prerequisites

CTS has been enabled.

Key Operations Recorded by CTS

Table 11-1 Key operations recorded by CTS

Operation	Resource Type	Event Name
Creating a cluster	cluster	createCluster
Deleting a cluster	cluster	deleteCluster
Expanding the cluster capacity	cluster	growCluster
Restarting a cluster	cluster	rebootCluster
Configuring a custom word dictionary	cluster	loadLexicon
Deleting a custom word dictionary	cluster	deleteLexicon
Performing basic configurations for a cluster snapshot	cluster	updateSnapshotPolicy
Setting the automatic snapshot creation policy	cluster	updateAutoSnapshotPolicy

Operation	Resource Type	Event Name
Manually creating a snapshot	snapshot	createSnapshot
Restoring a snapshot	snapshot	restoreSnapshot
Deleting a snapshot	snapshot	deleteSnapshot

11.2 Viewing Audit Logs

After CTS is enabled, CTS starts recording operations related to CSS. The CTS management console stores the last seven days of operation records. This section describes how to query the last seven days of operation records on the CTS management console.

Procedure

1. Log in to the CTS management console.
2. Click  in the upper left corner and select a region.
3. In the left navigation pane, click **Trace List**.
4. You can use filters to query traces. The following four filter criteria are available:
   - **Trace Source**, **Resource Type**, and **Search By**

     Select a filter criterion from the drop-down list.

     When you select **Trace name** for **Search By**, select a specific trace name.

     When you select **Resource ID** for **Search By**, enter a specific resource ID.

     When you select **Resource name** for **Search By**, select or enter a specific resource name.
   - **Operator**: Select a specific operator (at user level rather than tenant level).
   - **Trace Status**: Available options include **All trace statuses**, **normal**, **warning**, and **incident**. You can only select one of them.
   - **Time Range**: You can query traces generated during any time range of the last seven days.
5. Click ˅ on the left of a trace to expand its details.
6. Click **View Trace** in the **Operation** column. In the displayed **View Trace** dialog box, the trace structure details are displayed.

   For details about the key fields in the CTS trace structure, see the **Cloud Trace Service User Guide**.

