

**Cloud Search Service**

# **User Guide**

**Date**      2020-11-07

---

# Contents

---

|  |           |
|--|-----------|
| <b>1 Overview.....</b>   | <b>1</b>  |
| 1.1 CSS.....   | 1         |
| 1.2 Application Scenarios.....                                 | 2         |
| 1.3 Basic Concepts.....  | 3         |
| 1.4 Kibana.....  | 4         |
| 1.5 Cerebro.....   | 4         |
| 1.6 Clusters in Security Mode.....                             | 5         |
| 1.7 Multi-AZ HA.....   | 10        |
| 1.8 Related Services.....                                      | 12        |
| 1.9 Permissions Management.....                                | 13        |
| 1.10 Constraints.....  | 17        |
| <b>2 Getting Started.....</b>                                  | <b>19</b> |
| 2.1 Getting Started with Elasticsearch.....                    | 19        |
| <b>3 Permissions Management.....</b>                           | <b>26</b> |
| 3.1 Creating a User and Granting Permissions.....              | 26        |
| 3.2 CSS Custom Policies.....                                   | 27        |
| <b>4 Creating and Accessing a Cluster.....</b>                 | <b>32</b> |
| 4.1 Creating a Cluster.....                                    | 32        |
| 4.2 Accessing a Cluster.....                                   | 36        |
| <b>5 Importing Data to Elasticsearch.....</b>                  | <b>47</b> |
| 5.1 Using Logstash to Import Data to Elasticsearch.....        | 47        |
| 5.2 Using Kibana or APIs to Import Data to Elasticsearch.....  | 54        |
| <b>6 Suggestions on Using Elasticsearch.....</b>               | <b>58</b> |
| <b>7 Customizing Word Dictionaries.....</b>                    | <b>66</b> |
| 7.1 Configuring a Custom Word Dictionary.....                  | 66        |
| 7.2 Example.....   | 68        |
| <b>8 Simplified-Traditional Chinese Conversion Plugin.....</b> | <b>75</b> |
| <b>9 Managing Clusters.....</b>                                | <b>78</b> |
| 9.1 Cluster Status and Storage Capacity Status.....            | 78        |
| 9.2 Introduction to the Cluster List.....                      | 79        |

|   |            |
|---|------------|
| 9.3 Index Backup and Restoration.....                                       | 80         |
| 9.4 Modifying Specifications.....   | 86         |
| 9.5 Binding an Enterprise Project.....                                      | 87         |
| 9.6 Restarting a Cluster.....   | 88         |
| 9.7 Migrating a Cluster.....  | 90         |
| 9.8 Deleting a Cluster.....   | 91         |
| 9.9 Managing Tags.....  | 91         |
| 9.10 Public IP Address Access.....  | 93         |
| 9.11 Managing Logs.....   | 94         |
| 9.12 Managing Plugins.....  | 96         |
| 9.13 Hot and Cold Data Storage.....   | 99         |
| 9.14 Configuring Parameters.....  | 100        |
| <b>10 Monitoring a Cluster.....</b>   | <b>103</b> |
| 10.1 Supported Metrics.....   | 103        |
| 10.2 Creating Alarm Rules.....  | 108        |
| 10.3 Viewing Metrics.....   | 110        |
| <b>11 Elasticsearch SQL.....</b>  | <b>111</b> |
| <b>12 Querying Cluster Logs.....</b>  | <b>116</b> |
| 12.1 Key Operations Recorded by CTS.....                                    | 116        |
| 12.2 Viewing Audit Logs.....  | 117        |
| <b>13 FAQs.....</b>   | <b>118</b> |
| 13.1 What Are Regions and AZs?.....   | 118        |
| 13.2 How Does CSS Ensure Secure Running of Data and Services?.....          | 119        |
| 13.3 Which CSS Metrics Should I Focus On?.....                              | 119        |
| 13.4 Which Storage Options Does CSS Provide?.....                           | 120        |
| 13.5 What Is the Upper Limit for the Storage Capacity of CSS?.....          | 120        |
| 13.6 What Can Be the Disk Space of the Requested Cluster Used For?.....     | 120        |
| 13.7 Which Tools Can I Use to Manage CSS?.....                              | 121        |
| 13.8 Which Elasticsearch Versions Does CSS Support?.....                    | 121        |
| 13.9 Which Methods Can I Use to Access CSS?.....                            | 121        |
| 13.10 Does CSS Support APIs or Functions of Open-Source Elasticsearch?..... | 121        |
| 13.11 Can CSS Interconnect with Logstash?.....                              | 121        |
| 13.12 What Should I Do If an ECS Cannot Connect to a Cluster?.....          | 122        |
| 13.13 Which Search Functions Does CSS Support?.....                         | 122        |
| 13.14 Why Do I Fail to Create a Cluster?.....                               | 122        |
| 13.15 Why Cannot I Perform Index Backup?.....                               | 123        |
| 13.16 Filebeat Configuration Optimization.....                              | 123        |
| 13.17 What Should I Do If the Access to Kibana Fails?.....                  | 124        |
| <b>A Change History.....</b>  | <b>126</b> |

# 1 Overview

---

## 1.1 CSS

Cloud Search Service (CSS) is a fully managed, distributed search service. It is fully compatible with open-source Elasticsearch and provides users with structured and unstructured data search, statistics, and report capabilities. CSS works in a similar way as a database.

CSS can be automatically deployed, allowing you to quickly create clusters. It provides search engine optimization practices and requires no O&M. Additionally, it has a robust monitoring system that provides key metrics, including systems, clusters, and query performance, freeing you to focus on business logic.

For details about Elasticsearch, see the [Elasticsearch Reference](#).

### Advantages

- **Efficiency and ease of use**  
You can get insights from terabyte-scale data in milliseconds. In addition, you can use the visualized platform for data display and analysis.
- **Flexibility and scalability**  
You can request resources as needed and perform capacity expansion online with zero service interruption.
- **Custom word dictionary**  
Custom word dictionaries are supported. You can modify the word dictionary without having to restart instances.
- **Easy O&M**  
CSS is a fully-managed, out-of-the-box service. You can start using it with a few clicks. Professionals are ready to assist you whenever you want.
- **Solid reliability**  
You can choose to trigger snapshots manually or periodically for backup and restore snapshots to the current or other clusters. Snapshots of a cluster can be restored to another cluster to implement cluster data migration.

## Functions

CSS provides the following functions:

- Professional cluster management platform  
The CSS management console provides various function menus, allowing you to securely manage and maintain clusters with ease using a web browser.
- Robust monitoring system  
The CSS management console allows you to view the running status of created clusters via the dashboard and cluster list. You can obtain the current running status of clusters through metric views.
- Support for Elasticsearch  
Elasticsearch is a popular enterprise-class Lucene-powered search server that provides distributed multi-user capabilities. CSS adopts Elasticsearch and delivers multiple functions, including full-text search, structured search, analytics, aggregation, and highlighting. With CSS, you can achieve stable, reliable, and real-time search.

## 1.2 Application Scenarios

CSS applies to diversified scenarios, such as log analysis and site search.

### Log Analysis

In this scenario, you can perform O&M analysis and fault location for IT devices as well as operation analytics based on service metrics.

- Statistical analysis: Over 20 statistical analysis methods and nearly ten analytical dimensions are available.
- Real-time and efficient: You can get insights within seconds once new data is stored in indices.
- Visualized data: CSS provides multiple report display modes, such as table, line chart, heat map, and cloud map.

### Site Search

In this scenario, you can search website content by keyword as well as search for commodities on e-commerce sites with recommendations obtained.

- Real-time search: You can get the content or commodities you want within seconds or minutes.
- Categorized statistics: You can get categorized statistics on the searched commodities that meet conditions.
- Custom highlight style: You can customize the highlight style as you like.

## 1.3 Basic Concepts

### Cluster

CSS provides functions on a per cluster basis. A cluster represents an independent search service that consists of multiple nodes.

### Index

Index, similar to "database" in the relational database (RDB), stores Elasticsearch data. It refers to a logical space that consists of one or more shards.

**Table 1-1** Mapping between Elasticsearch and RDB

|                      |          |       |          |        |         |
|----------------------|----------|-------|----------|--------|---------|
| <b>Elasticsearch</b> | Index    | Type  | Document | Field  | Mapping |
| <b>RDB</b>           | Database | Table | Row      | Column | Schema  |

### Shard

An index can potentially store a large amount of data that can exceed the hardware limits of a single node. To solve this problem, Elasticsearch subdivides your index into multiple pieces called shards. When you create an index, you can simply define the number of shards that you want. Each shard is in itself a fully-functional and independent "index" that can be hosted on any node in the cluster.

You need to specify the number of shards before creating an index and cannot change the number after the index is successfully created.

### Replica

A replica is a copy of the actual storage index in a shard. It can be understood as a backup of the shard. Replicas help prevent single point of failures (SPOFs). You can increase or decrease the number of replicas based on your service requirements.

### Document

An entity for Elasticsearch storage. Equivalent to the row in the RDB, the document is the basic unit that can be indexed.

### Type

Similar to the table in the RDB, the type is used to distinguish between different data. One index can contain multiple document types. A document actually must be indexed to a document type inside an index.

## Mapping

A mapping is used to restrict the type of a field and can be automatically created based on data. It is similar to the schema in the database.

## Field

Minimum unit of a document. A field is similar to a column in a database.

## 1.4 Kibana

Kibana is an open-source data analytics and visualization platform and works with Elasticsearch. You can use Kibana to search, view, and interact with data stored in Elasticsearch indices as well as to visualize your data in a variety of charts, tables, and maps.

To learn more about Kibana, go to the Kibana official website: <https://www.elastic.co/guide/en/kibana/current/index.html>

### Accessing Kibana with a Few Clicks

CSS is integrated with Kibana. You can access Kibana with a few clicks, without having to install Kibana.

Log in to the CSS management console. In the left navigation pane, click **Clusters**. On the displayed **Clusters** page, locate the row where the target cluster resides and click **Kibana** in the **Operation** column.

### Kibana Functions

Kibana delivers the functions of visualization and Elasticsearch statistics and analysis. Specifically, it supports:

- Over 10 data presentation modes
- Nearly 20 data statistics methods
- Classification in terms of various dimensions, such as time and tag

## 1.5 Cerebro

Cerebro is an open-source Elasticsearch web visualized management tool built using Scala, Play Framework, AngularJS, and Bootstrap. You can use Cerebro to perform web visualized management on a cluster, such as executing REST requests, modifying Elasticsearch configurations, monitoring real-time disks, cluster load, and memory usage.

### Accessing Cerebro with a Few Clicks

CSS is integrated with Cerebro. You can access Cerebro with a few clicks, without having to install Cerebro.

Log in to the CSS management console. In the left navigation pane, click **Clusters**. On the displayed **Clusters** page, locate the row where the target cluster resides and click **Cerebro** in the **Operation** column.

On the displayed page, enter one private access address of the cluster.

- If the cluster does not have the security mode enabled, enter **http://IP address:9200**.
- If the cluster has the security mode enabled, enter **https://IP address:9200** and then enter the username and password for security mode login.

## Cerebro Functions

Cerebro is a fully compatible open-source tool and supports the latest version 0.8.4.

- Elasticsearch visualized and real-time load monitoring
- Elasticsearch visualized data management

## 1.6 Clusters in Security Mode

Security mode is supported for Elasticsearch 6.5.4 and later versions. After enabling it, identity verification, authorization, and encryption are required.

The following describes the security mode using Kibana as an example.

### NOTE

Security mode can be enabled only during cluster creation. It cannot be enabled after a cluster is created.

## Key Terms

**Table 1-2** Key terms of security mode

| Term         | Description  |
|--------------|--|
| Permission   | Single action, for example, creating an index (for example, <b>indices:admin/create</b> )  |
| Action group | A group of permissions. For example, the predefined <b>SEARCH</b> action group grants roles permissions to use <b>_search</b> and <b>_msearchAPI</b> .                                   |
| Role         | A role is a combination of permissions or action groups, including operation permissions on clusters, indices, documents, or fields.   |
| Backend role | (Optional) Other external roles from the backend such as LDAP/Active Directory   |
| User         | A user can send operation requests to the Elasticsearch cluster. The user has credentials such as username and password, zero or more backend roles, and zero or more custom attributes. |

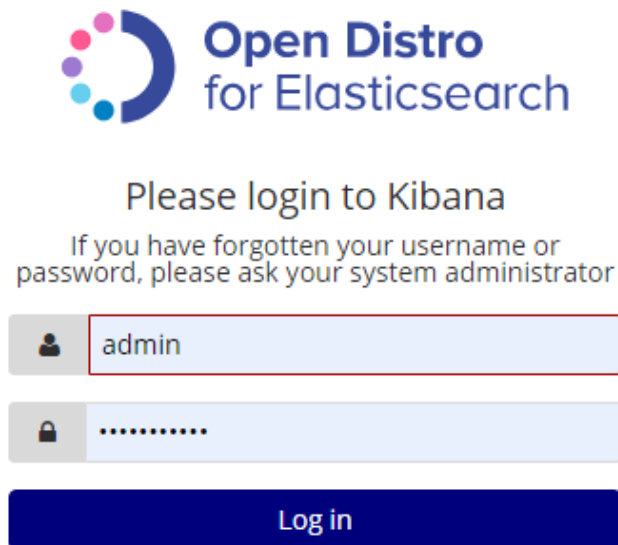


| Term         | Description  |
|--------------|--|
| Role mapping | A user will be assigned a role after successful authentication. Role mapping is to map a role to a user (or a backend role). For example, the mapping from <b>kibana_user</b> (role) to <b>jdoe</b> (user) means that John Doe obtains all permissions of <b>kibana_user</b> after authenticated by <b>kibana_user</b> . Similarly, the mapping from <b>all_access</b> (role) to <b>admin</b> (backend role) means that any user with the backend role <b>admin</b> (from the LDAP/Active Directory server) has all the permissions of role <b>all_access</b> after authenticated. You can map each role to multiple users or backend roles. |

## Identity Verification

After enabling the security mode, you need to log in to the cluster with the username and password that you set when creating the cluster. You can perform other operations after you log in successfully.

**Figure 1-1** Login for identity verification

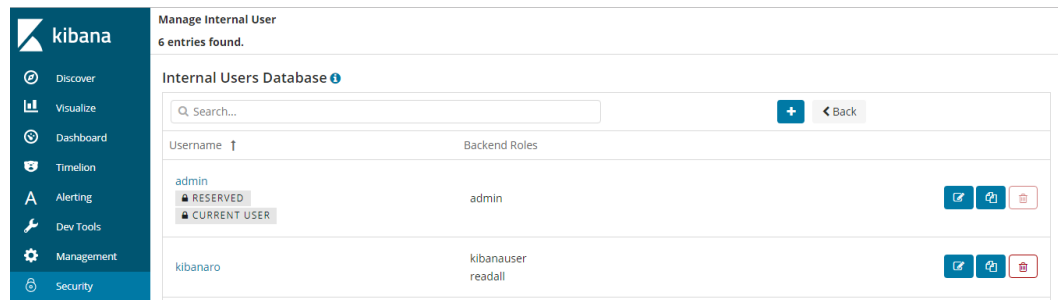


## Authorization

Choose **Kibana > Security** to control user permissions in Elasticsearch clusters. You can configure hierarchical user permissions by the cluster, index, document, and field.

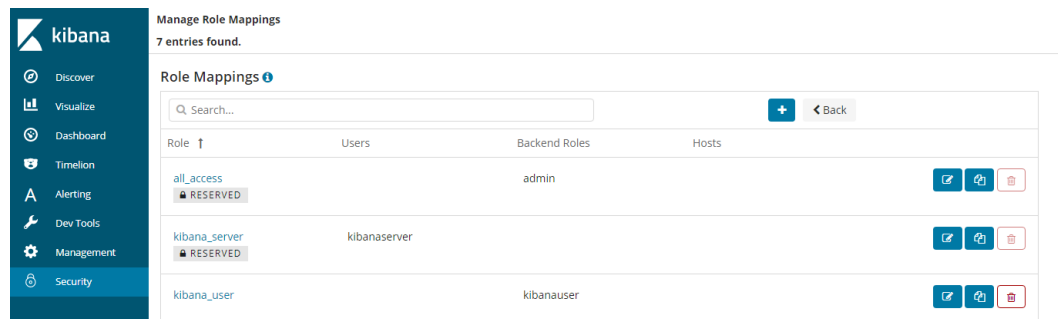
You can add or delete users, and map users to different roles for permissions control.

**Figure 1-2** Configuring users



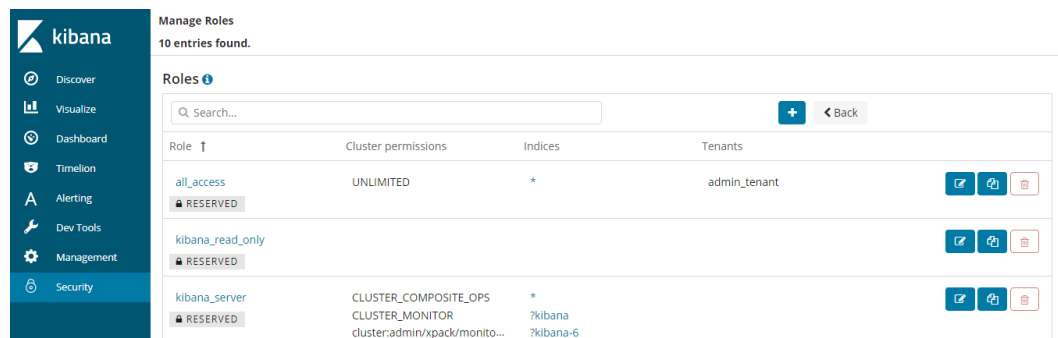
You can use role mapping to configure roles and map a user's username, backend role, and host name to a role.

**Figure 1-3** Role mapping



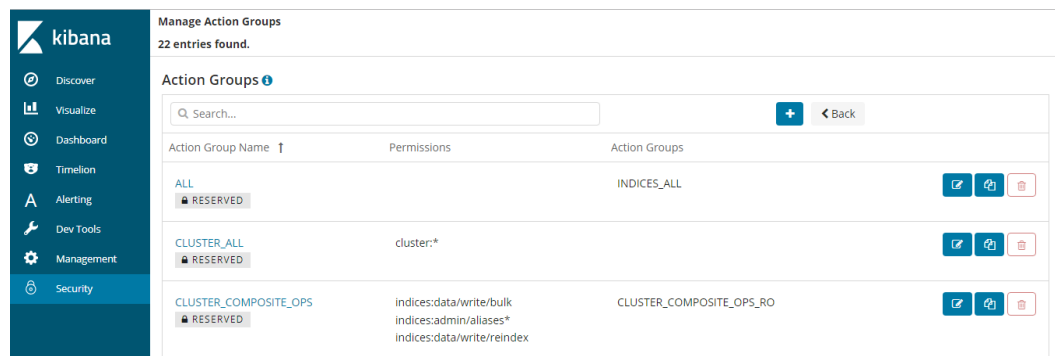
You can set permissions for each role to access clusters, indices and documents and assign Kibana tenants different roles.

**Figure 1-4** Configuring role permissions



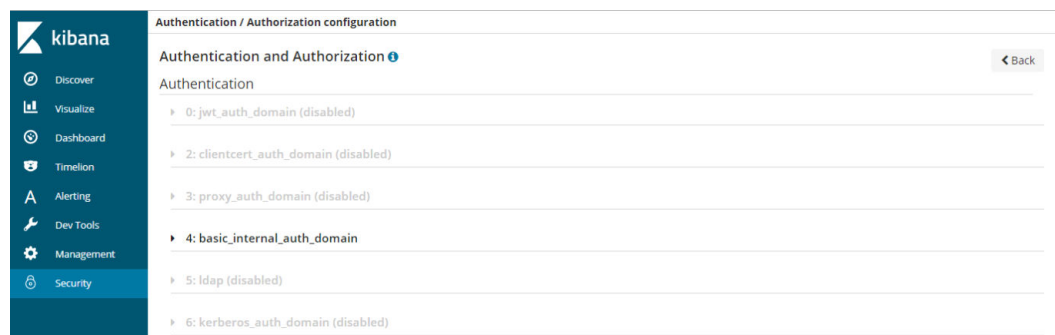
You can set action groups, assign the groups to roles, and configure the roles' permission for accessing indices and documents.

Figure 1-5 Configuring action groups



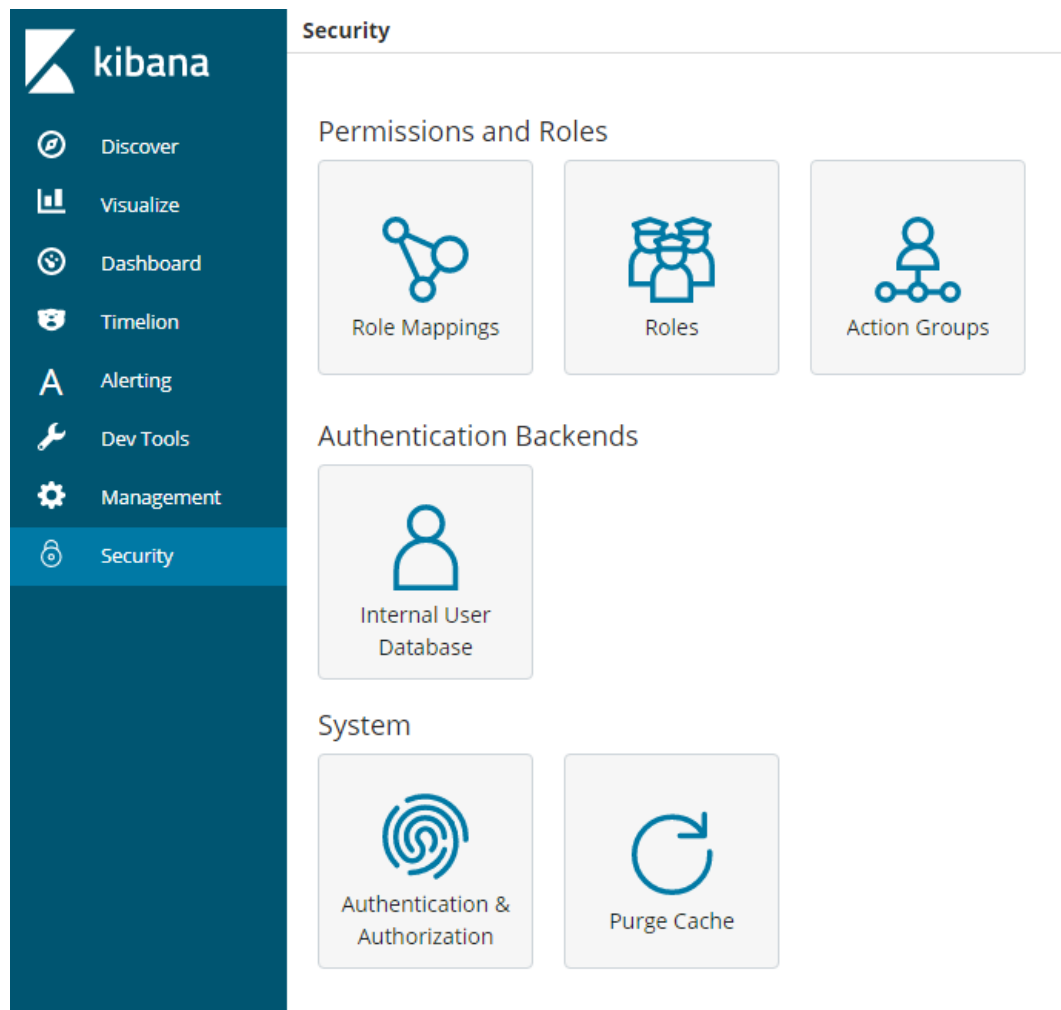
You can view the parameters of authentication and authorization for the current cluster. You can also run the **securityadmin** command to modify the configuration.

Figure 1-6 Viewing cluster parameters



What's more, you are allowed to clear all security caches.

Figure 1-7 Clearing the security cache



## Encryption

When key data is transferred between nodes or over HTTP, SSL/TLS encryption can be used to protect data security.

You can perform the preceding functions on Kibana, using **.yaml** files (not recommended), or by calling RESTful APIs. For more information about the security mode, see [Security](#).

## Resetting Passwords

When you want to change the login password of a cluster with the security mode enabled or just forgot the password, reset the cluster password.

1. On the **Clusters** page, locate the target cluster whose password you want to reset and click the cluster name. The **Basic Information** page is displayed.
2. On the **Basic Information** page, click **Reset** next to **Reset Password** to reset the password.

 **NOTE**

- The password can contain 8 to 32 characters.
- It must include letters, digits, and special characters. No spaces are allowed.
- It cannot be the username or the username spelled backwards.
- You are advised to change the password periodically.

## 1.7 Multi-AZ HA

To prevent data loss and minimize the cluster downtime upon service interruption, select two or three AZs in the same region when creating a cluster, and then the system properly allocates nodes to the AZs.

### Allocating Nodes

If you select two or three AZs when creating a cluster, CSS automatically enables the cross-AZ HA function and properly allocates nodes to different AZs.

The following table lists the way nodes are allocated.

| Nodes | One AZ | Two AZs       |     | Three AZs     |     |     |
|-------|--------|---------------|-----|---------------|-----|-----|
|       | AZ1    | AZ1           | AZ2 | AZ1           | AZ2 | AZ3 |
| 1     | 1      | Not supported |     | Not supported |     |     |
| 2     | 2      | 1             | 1   | Not supported |     |     |
| 3     | 3      | 2             | 1   | 1             | 1   | 1   |
| 4     | 4      | 2             | 2   | 2             | 1   | 1   |
| ...   | ...    | ...           | ... | ...           | ... | ... |

 **NOTE**

- CSS does not require that the number of nodes be a multiple of that of AZs.
- When creating a cluster, ensure that the number of nodes you configure is no less than the number of AZs.
- The node quantity gap between any two AZs must be no more than one.

### Configuring Replicas

HA can be ensured when you properly configure the number of replicas.

- In a two-AZ deployment mode, when one AZ is unavailable, the other one is required to provide services. Therefore, at least one replica is required. The default number of Elasticsearch replicas is one. In this case, you can retain the default value if you do not expect too much for the read performance.
- In a three-AZ deployment mode, when one or two of the AZs are unavailable, the remaining AZs are required to provide services. Therefore, at least two replicas are needed. In this case, you need to modify the replica

configurations to change the number of replicas because the default number of Elasticsearch replicas is one.

You can run the following command to modify the number of index replicas:

```
curl -XPUT http://ip:9200/{index_name}/_settings -d
'{"number_of_replicas":2}'
```

Alternatively, specify the number of replicas in the template:

```
curl -XPUT http://ip:9200/_template/templatename -d '{"template": "*",
"settings": {"number_of_replicas": 2}}'
```

 NOTE

- **ip**: private network address
- **number\_of\_replicas**: number of replicas after modification. The value in the preceding command indicates that two replicas are required.

## Selecting Master Nodes

If you select the master node function when creating a cluster, master nodes are properly allocated in different AZs when you select multiple AZs.

## Service Interruption

**Table 1-3** shows the service fault analysis if you select two or three AZs when creating a cluster and one AZ is faulty.

**Table 1-3** Service fault analysis when an AZ is faulty

| AZs | Master Nodes | Service Interruption Analysis   |
|-----|--------------|---|
| 2   | 0            | <ul style="list-style-type: none"> <li>• When the number of nodes is a multiple of 2, <ul style="list-style-type: none"> <li>– If half of data nodes are faulty, replace one node in the faulty AZ before selecting the master node.</li> </ul> </li> <li>• When the number of nodes is an odd number, <ul style="list-style-type: none"> <li>– If the faulty AZ contains one more node than the normal AZ does, you need to replace one node in the faulty AZ before selecting the master node. For details about how to replace nodes, contact technical support.</li> <li>– If the faulty AZ contains one less node than the normal AZ does, services are not interrupted and you can select the master node.</li> </ul> </li> </ul> |

| AZs | Master Nodes | Service Interruption Analysis   |
|-----|--------------|---|
| 2   | 3            | <p>You may have 50% of possibilities for service interruption. When two dedicated master nodes are allocated to one AZ and another master node is allocated to the other AZ,</p> <ul style="list-style-type: none"> <li>• If service interruption happens in the AZs with one master node, you can select master nodes from the AZs that have two dedicated master nodes.</li> <li>• If service interruption happens in the AZs with two dedicated master nodes, you cannot select two master nodes from the remaining AZ because it has only one dedicated master node. In this case, services are interrupted and you need to contact technical support.</li> </ul> |
| 3   | 0            | <p>If you configure four nodes in three AZs, each AZ is allocated with two, one, and one node respectively. Services will be interrupted if the AZ with two nodes is faulty. Therefore, you are advised not to configure four nodes when selecting three AZs.</p> <p>Generally, service interruption does not occur.</p>  |
| 3   | 3            | Service interruption does not occur.  |

## 1.8 Related Services

This section describes the relationship between CSS and other services.

- **Virtual Private Cloud (VPC)**  
CSS clusters are created in the subnets of a VPC. VPCs provide a secure, isolated, and logical network environment for your clusters.
- **Elastic Cloud Server (ECS)**  
In a CSS cluster, each node represents an ECS. When you create a cluster, ECSs are automatically created to serve as nodes.
- **Elastic Volume Service (EVS)**  
CSS uses EVS to store index data. When you create a cluster, EVSs are automatically created for cluster data storage.
- **Object Storage Service (OBS)**  
Snapshots of CSS clusters are stored in OBS buckets.
- **Cloud Eye**  
CSS uses Cloud Eye to monitor cluster metrics in real time to ensure normal service running. The supported CSS metrics include the disk usage and cluster health status. You can learn about the disk usage of the cluster in a timely manner based on the disk usage metric. You can learn about the health status of a cluster based on the cluster health status metric.

## 1.9 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your CSS resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you securely access your resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use CSS resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using CSS resources.

If the account has met your requirements, you do not need to create an independent IAM user for permission management. Then you can skip this section. This will not affect other functions of CSS.

IAM can be used free of charge. You pay only for the resources in your account. For more information about IAM, see [IAM Service Overview](#).

### Permissions Management

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

CSS is a project-level service deployed in specific physical regions. Therefore, CSS permissions are assigned to users in specific regions and only take effect for these regions. If you want the permissions to take effect for all regions, you need to assign the permissions to users in each region. When accessing CSS, the users need to switch to a region where they have been authorized to use cloud services.

You can grant users permissions by using roles and policies.

- Roles are a type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies are a type of fine-grained authorization mechanism that defines the permissions for performing operations on specific cloud resources under certain conditions. This mechanism allows for flexible policy-based authorization and meets requirements for secure access control. For example, you can grant CSS users only the permissions for managing a certain type of CSS. For the API actions supported by CSS, see [Permissions Policies and Supported Actions](#).

**Table 1-4** lists all the system roles supported by CSS. For example, some CSS roles are dependent on the roles of other services. When assigning CSS roles to users, you need to also assign dependent roles for the CSS permissions to take effect.



**Table 1-4** System-defined roles and policies supported by CSS

| Role Name                   | Description       | Dependency   |
|-----------------------------|-------------------|--|
| Elasticsearch Administrator | CSS administrator | <p>Dependent on the <b>Tenant Guest</b> and <b>Server Administrator</b> roles.</p> <ul style="list-style-type: none"> <li>• <b>Tenant Guest:</b> A global role, which must be assigned in the global project.</li> <li>• <b>Server Administrator:</b> A project-level role, which must be assigned in the same project.</li> </ul> |

**Table 1-5** Relationship between user permissions and roles

| Permission Type | Description  | Type                | Required Role  |
|-----------------|--|---------------------|--|
| Permission 1    | <p>Permissions:</p> <ul style="list-style-type: none"> <li>• Creating, deleting, and expanding CSS clusters</li> <li>• Manually and automatically backing up CSS cluster data</li> <li>• Restoring CSS cluster data</li> <li>• Creating an IAM agency</li> <li>• Creating an OBS bucket</li> <li>• Creating a VPC and security group</li> <li>• Kibana</li> <li>• Customizing a word dictionary</li> </ul> | System-defined role | <ul style="list-style-type: none"> <li>• Elasticsearch Administrator</li> <li>• Server Administrator</li> <li>• Tenant Guest</li> <li>• VPC Administrator</li> <li>• Security Administrator</li> <li>• Tenant Administrator</li> </ul> |
| Permission 2    | <p>Permissions:</p> <ul style="list-style-type: none"> <li>• Creating, deleting, and expanding CSS clusters</li> <li>• Manually backing up CSS cluster data</li> <li>• Restoring CSS cluster data</li> <li>• Kibana</li> <li>• Customizing a word dictionary</li> </ul>  | System-defined role | <ul style="list-style-type: none"> <li>• Elasticsearch Administrator</li> <li>• Server Administrator</li> <li>• Tenant Guest</li> </ul>  |

| Permission Type | Description  | Type                | Required Role   |
|-----------------|--|---------------------|---|
| Permission 3    | Permissions: <ul style="list-style-type: none"> <li>Viewing the cluster list</li> <li>Viewing the Overview page</li> <li>Kibana</li> </ul> | System-defined role | This permission is dependent on the <b>Tenant Guest</b> role, which must be assigned in the same project as <b>Permission 3</b> . |

**Table 1-6** lists the common operations supported by each system permission of CSS. Please choose proper system policies according to this table.

**Table 1-6** Common operations supported by each system-defined policy

| Operation  | CSS FullAccess | CSS ReadOnlyAccess | Elasticsearch Administrator | Remarks |
|--|----------------|--------------------|-----------------------------|---------|
| Creating a cluster                                       | √              | x                  | √                           | -       |
| Querying a cluster list                                  | √              | √                  | √                           | -       |
| Querying cluster details                                 | √              | √                  | √                           | -       |
| Deleting a cluster                                       | √              | x                  | √                           | -       |
| Restarting a cluster                                     | √              | x                  | √                           | -       |
| Expanding cluster capacity                               | √              | x                  | √                           | -       |
| Adding instances and expanding instance storage capacity | √              | x                  | √                           | -       |
| Querying tags of a specified cluster                     | √              | √                  | √                           | -       |

| Operation  | CSS FullAccess | CSS ReadOnlyAccess | Elasticsearch Administrator | Remarks                              |
|--|----------------|--------------------|-----------------------------|--------------------------------------|
| Querying all tags  | √              | √                  | √                           | -                                    |
| Creating a Poisson word dictionary                               | √              | x                  | √                           | Depending on OBS and IAM permissions |
| Querying the Poisson word dictionary status                      | √              | √                  | √                           | -                                    |
| Deleting a Poisson word dictionary                               | √              | x                  | √                           | -                                    |
| Loading a custom word dictionary                                 | √              | x                  | √                           | Depending on OBS and IAM permissions |
| Querying the status of a custom word dictionary                  | √              | √                  | √                           | -                                    |
| Deleting a custom word dictionary                                | √              | x                  | √                           | -                                    |
| Automatically setting basic configurations of a cluster snapshot | √              | x                  | √                           | Depending on OBS and IAM permissions |
| Modifying basic configurations of a cluster snapshot             | √              | x                  | √                           | Depending on OBS and IAM permissions |

| Operation                                       | CSS FullAccess | CSS ReadOnlyAccess | Elasticsearch Administrator | Remarks |
|---|----------------|--------------------|-----------------------------|---------|
| Setting the automatic snapshot creation policy  | √              | x                  | √                           | -       |
| Querying the automatic snapshot creation policy | √              | √                  | √                           | -       |
| Manually creating a snapshot                    | √              | x                  | √                           | -       |
| Querying the snapshot list                      | √              | √                  | √                           | -       |
| Restoring a snapshot                            | √              | x                  | √                           | -       |
| Deleting a snapshot                             | √              | x                  | √                           | -       |
| Disabling the snapshot function                 | √              | x                  | √                           | -       |

## Helpful Links

- [IAM Service Overview](#)
- [Creating a User and Granting Permissions](#)
- [Permissions Policies and Supported Actions](#)

## 1.10 Constraints

### Restrictions on Clusters and Nodes

The following table describes restrictions on clusters and nodes in CSS.

**Table 1-7** Restrictions on clusters and nodes

| Cluster and Node                     | Restriction  |
|--------------------------------------|--|
| Maximum number of nodes in a cluster | Default: 32. Maximum: 200. To change the default value, contact technical support. |
| Minimum number of nodes in a cluster | 1  |

## Restrictions on Browsers

- You are advised to use the following browsers to access the CSS management console:
  - Google Chrome 36.0 or later
  - Mozilla Firefox 35.0 or later
  - Internet Explorer 9.0 or later

If you use Internet Explorer 9.0, you may fail to log in to the CSS management console because user **Administrator** is disabled by default in some Windows systems, such as Windows 7 Ultimate. The Internet Explorer automatically selects a system user for installation. As a result, the Internet Explorer cannot access the management console. Reinstall Internet Explorer 9.0 or later (recommended) or run Internet Explorer 9.0 as user **Administrator**.
- You are advised to use the following browsers to access Kibana integrated in CSS:
  - Google Chrome 36.0 or later
  - Mozilla Firefox 35.0 or later
  - Internet Explorer 11.0 or later (Internet Explorer 9 is not supported.)

# 2 Getting Started

---

## 2.1 Getting Started with Elasticsearch

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

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-4u8g** 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.

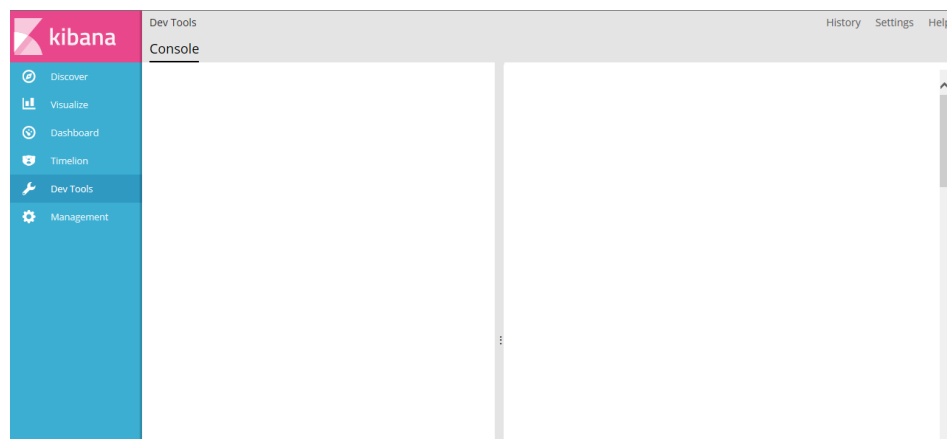
## Step 2: Import Data

CSS supports importing data to Elasticsearch using 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 2-1](#).

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

**Figure 2-1** 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": "-Rf6VHIBfCl6SDjw7H5",
    "_score": 0.5945667,
    "_source": {
      "productName": "Latest casual pants for women in spring 2017",
      "size": "L"
    }
  },
  {
    "_index": "my_store",
    "_type": "_doc",
    "_id": "-hf6VHIBfCl6SDjw7H5",
    "_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**.

# 3 Permissions Management

---

## 3.1 Creating a User and Granting Permissions

This section describes how to use a group to grant permissions to a user. [Figure 3-1](#) shows the process for granting permissions.

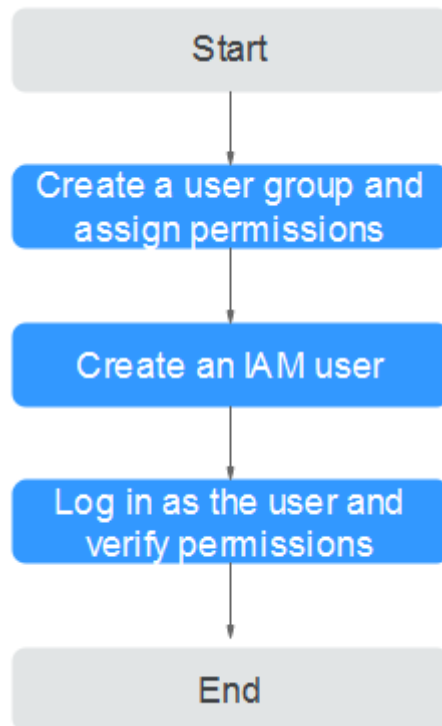
CSS has two types of user permissions: CSS administrator permission and read-only permission. Plan the two types of user groups during permission planning.

### Prerequisites

Before assigning permissions to user groups, you should learn about the system policies listed in [Permissions Management](#).

## Process Flow

**Figure 3-1** Process for granting CSS permissions



1. **Create and authorize a user group.**  
Create a user group on the IAM console, and assign the CSS permission to the group.
2. **Create a user and add it to a user group.**  
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.

## 3.2 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 about how to create custom policies, 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:privatelps: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:privatelps: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"
    }
  ]
}
```

# 4 Creating and Accessing a Cluster


---




## 4.1 Creating a Cluster

To use CSS, create a cluster first.

### 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 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 .
5. Set host specifications of the cluster.

**Table 4-1** Parameter description

| Parameter             | Description  |
|-----------------------|--|
| Nodes                 | <p>Number of nodes in a cluster.</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• If both master and client nodes are enabled, the nodes specified by this parameter are only used for storing data.</li> <li>• 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.</li> </ul> |
| Node Specifications   | <p>Flavor of nodes in a cluster. You can select a specified specification based on your needs. You can only select one node specification for a cluster and cannot select the CPU and memory resources that have been sold out.</p>  |
| Node Storage Type     | <p>In the current version, the following options are available: <b>Common I/O</b>, <b>High I/O</b>, and <b>Ultra-high I/O</b>.</p>   |
| Node Storage Capacity | <p>Storage space. Its value is related to node specifications and varies with node specifications.</p>   |
| Master node           | <p>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 <b>Nodes</b> parameter and use the nodes as both master and client nodes.</p> <p>After enabling the master node, specify <b>Node Specifications</b>, <b>Nodes</b>, and <b>Node Storage Capacity</b>. The value of <b>Nodes</b> must be an odd number greater than 3. You can set a maximum of nine nodes. The value of <b>Node Storage</b> is fixed. You can select a storage type based on your needs.</p>   |

| Parameter      | Description   |
|----------------|---|
| Client node    | <p>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 <b>Nodes</b> parameter and use the nodes as both master and client nodes.</p> <p>After enabling the client node, specify <b>Node Specifications</b>, <b>Nodes</b> and <b>Node Storage Capacity</b>. The value of <b>Nodes</b> ranges from 1 to 32. The value of <b>Node Storage</b> is fixed. However, you can select a storage type based on your needs.</p>                  |
| Cold data node | <p>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.</p> <p>This parameter is optional. A maximum of 32 cold data nodes are supported.</p> <p>After enabling the cold data node, CSS automatically adds cold or hot tags to related nodes. For details about the parameters, see <a href="https://www.elastic.co/guide/en/elasticsearch/reference/master/allocation-awareness.html">https://www.elastic.co/guide/en/elasticsearch/reference/master/allocation-awareness.html</a>.</p> |

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](#).
  - **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.
7. **Advanced Settings:** If you select **Default**, the **Tag** function is disabled by default and the **Automatic Snapshot Creation** function is enabled. If you need to configure **Tag** and **Automatic Snapshot Creation** functions, select **Custom**.
8. **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](#).
9. 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.

10. Click **Next** to switch to the **Confirm** page.
11. After the specifications are confirmed, click **Submit**.
12. 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.

## 4.2 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 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 create and log in to an ECS, see Logging In to a Linux ECS in the *Elastic Cloud Server User Guide* or Logging In to a Windows 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 the *Virtual Private Cloud User Guide*.
- 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. Create 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 4-1 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 ks71z4ggTUCy2UDWkXqEgw 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 uLdSGZ0BS7uam1QfxD1EsQ 5 1 0 0 1.5kb 810b
green open twitter lzPIdrMRQpeBg1I76SAYGA 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
InetSocketAddress("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.getByAddress(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 `SSLIOStrategy` 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.

`SSLIOStrategy`: 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();
}
```

```

SSLIOStrategy sessionStrategy = new SSLIOStrategy(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();
}

```

```

SSLIOStrategy sessionStrategy = new SSLIOStrategy(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.SSLIOStrategy;
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.SSLIOStrategy;
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 SSLIOStrategy} for all requests to enable SSL / TLS encryption.
     */
    private final SSLIOStrategy 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 SSLIOStrategy 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 SSLIOStrategy} that will be added to the HTTP client.
     *
     * @return Never {@code null}.
     */
    SSLIOStrategy 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>
```

# 5 Importing Data to Elasticsearch

---

## 5.1 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](#)

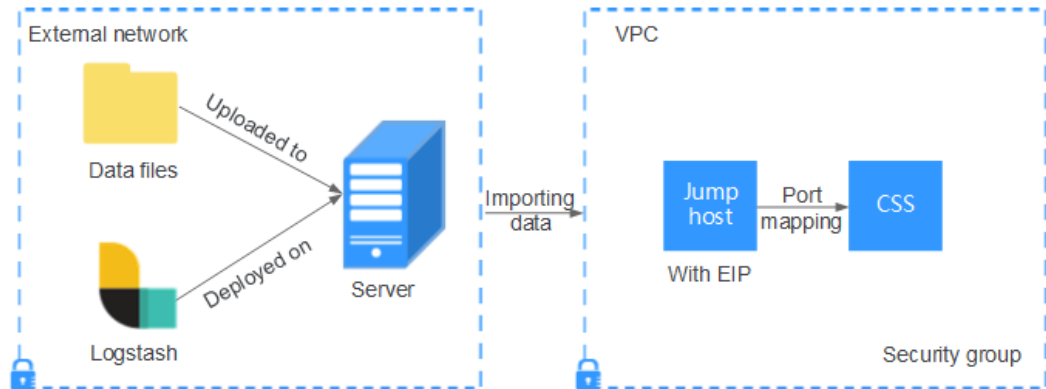
### Prerequisites

- 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 5-1** illustrates how data is imported when Logstash is deployed on the external network.

**Figure 5-1** 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
```

- 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     |

- 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
```

- 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.
- 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.

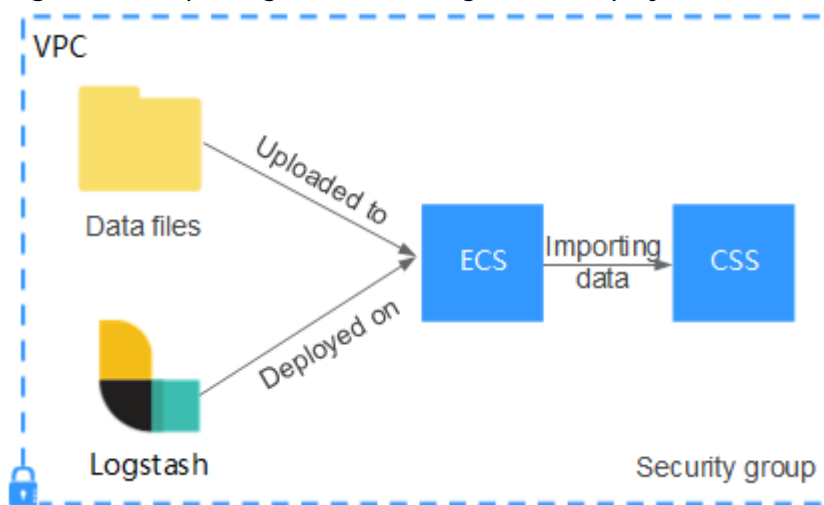
11. In the left navigation pane of the displayed Kibana window, click **Dev Tools**. Click **Get to work** to switch to the **Console** page.
12. On the **Console** page of Kibana, search for the imported data.  
On the **Console** page of Kibana, enter the following command to search for data. View 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 5-2** 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 5-2** 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     |

- 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.
- 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
```



## 5.2 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:

Versions earlier than 7.x

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

Versions later than 7.x

```
PUT /my_store
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "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.):

Versions earlier than 7.x

```
POST /my_store/products/_bulk
```

```
{"index":{}}
```

```
{"productName":"Latest art shirts for women in 2017 autumn","size":"L"}
```

Versions later than 7.x

```
POST /my_store/_bulk
```

```
{"index":{}}
```

```
{"productName":"Latest art shirts for women in 2017 autumn","size":"L"}
```

The command output looks like that in [Figure 5-3](#). If the value of the **errors** field in the result is **false**, the data is successfully imported.

**Figure 5-3** Response message

```

1  {
2    "took": 42,
3    "errors": false,
4    "items": [
5      {
6        "index": {
7          "_index": "my_store",
8          "_type": "products",
9          "_id": "AWTGbHt7BwpN-hb3LKau",
10         "_version": 1,
11         "result": "created",
12         "_shards": {
13           "total": 2,
14           "successful": 2,
15           "failed": 0
16         },
17         "created": true,
18         "status": 201
19       }
20     ]
21   }
22 }

```

## 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**:

Versions earlier than 7.x

```
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"
        }
      }
    }
  }
}'
```

Versions later than 7.x

```
curl -X PUT http://192.168.0.90:9200/my_store -H 'Content-Type: application/json' -d '{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
    "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
```

# 6 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.

Versions earlier than 7.x

```
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 }
```

Versions later than 7.x

```
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 word segmentation, such as tags or enumerated values, as the keyword type.

Versions earlier than 7.x

```
PUT my_index1
{
  "mappings": {
    "my_type": {
      "properties": {
```

```

    "tags": {
      "type": "keyword"
    },
    "full_name": {
      "type": "text"
    }
  }
}
}
}

```

#### Versions later than 7.x

```

PUT my_index1
{
  "mappings": {
    "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.

#### Versions earlier than 7.x

```

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

```

#### Versions later than 7.x

```

PUT my_index2
{
  "mappings": {
    "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.

Versions earlier than 7.x

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

Versions later than 7.x

```
PUT /_template/my_logsa  
{  
  "index_patterns": ["logstasaah-*"],  
  "order": 1,  
  "settings": {  
    "number_of_shards": 1  
  },  
  "mappings": {  
    "properties": {  
      "_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.

Versions earlier than 7.x

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

Versions later than 7.x

```
GET /gb/tweet/_validate/query
{
  "query": {
    "productName": {
      "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.

Versions earlier than 7.x

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

```
}  
}  
}
```

#### Versions later than 7.x

```
GET /gb/tweet/_validate/query?explain  
{  
  "query": {  
    "productName": {  
      "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]"  
}
```

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.

# 7 Customizing Word Dictionaries

---

## 7.1 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 the cluster.

### 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 *Object Storage Service User Guide*. 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.
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.


## 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**.

## 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.

## 7.2 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": "word-1",
      "start_offset": 0,
      "end_offset": 4,
      "type": "CN_WORD",
      "position": 0
    },
    {
      "token": "word-2",
      "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" : [
    {
      "token": "word-1",
      "start_offset" : 0,
      "end_offset" : 4,
      "type" : "CN_WORD",
      "position" : 0
    },
    {
      "token": "word-3",
      "start_offset" : 0,
      "end_offset" : 2,
      "type" : "CN_WORD",
      "position" : 1
    },
    {
      "token": "word-4",
      "start_offset" : 0,
      "end_offset" : 1,
      "type" : "CN_WORD",
      "position" : 2
    },
    {
      "token": "word-5",
      "start_offset" : 1,
      "end_offset" : 3,
      "type" : "CN_WORD",
      "position" : 3
    },
    {
      "token": "word-6",
      "start_offset" : 2,
      "end_offset" : 4,
      "type" : "CN_WORD",
      "position" : 4
    },
    {
      "token": "word-7",
      "start_offset" : 3,
      "end_offset" : 4,
      "type" : "CN_WORD",
      "position" : 5
    },
    {
      "token": "word-2",
      "start_offset" : 5,
      "end_offset" : 8,
      "type" : "CN_WORD",
      "position" : 6
    },
    {
      "token": "word-8",
      "start_offset" : 5,

```



```

    "end_offset" : 7,
    "type" : "CN_WORD",
    "position" : 7
  },
  {
    "token" : "word-9",
    "start_offset" : 6,
    "end_offset" : 8,
    "type" : "CN_WORD",
    "position" : 8
  },
  {
    "token" : "word-10",
    "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"  
        }  
      }  
    ]  
  }  
}
```

# 8 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"
  }
]
}
```



# 9 Managing Clusters

## 9.1 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 9-1** 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.<br><br>If a cluster is in the <b>Unavailable</b> 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 administrator 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 9-2** Cluster storage capacity status description

| Status   | Description   |
|----------|---|
| Normal   | 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 <b>Abnormal</b> due to faults, the storage space status of the cluster is <b>Abnormal</b> .        |

## 9.2 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 9-3](#) 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 9-3** 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 <a href="#">Cluster Status and Storage Capacity Status</a> .   |
| 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 <b>Kibana</b> , <b>View Metric</b> , <b>Modify</b> , <b>Restart</b> , <b>Delete</b> , <b>Custom Word Dictionary</b> , <b>Migrate</b> , <b>Cerebro</b> , and <b>Back Up and Restore</b> . If an operation is not allowed, the button is gray. |

## 9.3 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.

- **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**.

**Table 9-4** 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: <ul style="list-style-type: none"> <li>• <b>Storage Class</b> is <b>Standard</b> or <b>Infrequent Access</b>.</li> </ul>   |
| Backup Path | Storage path of the snapshot in the OBS bucket.   | The backup path configuration rules are as follows: <ul style="list-style-type: none"> <li>• The backup path cannot contain the following characters: \:*?"&lt;&gt; </li> <li>• The backup path cannot start with a slash (/).</li> <li>• The backup path cannot start or end with a period (.</li> <li>• The total length of the backup path cannot exceed 1,023 characters.</li> </ul> |
| 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: <ul style="list-style-type: none"> <li>• <b>Agency Type</b> is <b>Cloud service</b>.</li> <li>• <b>Cloud Service</b> is <b>Elasticsearch</b>.</li> <li>• The agency has the <b>Tenant Administrator</b> permission for the <b>OBS</b> project in <b>Global service</b>.</li> </ul>                |

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.


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.

7. Click **OK**.

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

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.
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.

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.

## 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**.

## 9.4 Modifying Specifications

If the specifications of a cluster cannot meet business requirements, modify its specifications to improve storage and usage efficiency.

### 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.
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.

## 9.5 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.

## 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.
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**.
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.
  - 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.

## 9.6 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**.
  - You can quick restart nodes by **Node type** or **Node name**. If you select **Node type**, then you can select multiple node types to perform quick restart at a time. If you select **Node name**, you can quickly restart only one node at a time.
  - 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 administrator 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**.
  - You can perform rolling restart by **Node type**. Select specific node types for restart.
  - During the rolling restart, the cluster is able to provide services, and only the node that is being restarted is unavailable. If you fail to access a node, you can 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 administrator for troubleshooting.

## 9.7 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](#).

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](#).

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

## 9.8 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**.

## 9.9 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 9-5** Naming rules for a tag key and value

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

## 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** 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.

## 9.10 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 9-6** 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**.

## 9.11 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**.

**Table 9-7** 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. |

| Parameter   | Description   | Remarks  |
|-------------|---|--|
| Backup Path | Storage path of logs in the OBS bucket  | <p>The backup path configuration rules are as follows:</p> <ul style="list-style-type: none"> <li>• The backup path cannot contain the following characters: \:*?"&lt;&gt; </li> <li>• The backup path cannot start with a slash (/).</li> <li>• The backup path cannot start or end with a period (.)</li> <li>• The total length of the backup path cannot exceed 1,023 characters.</li> </ul>     |
| IAM Agency  | IAM agency authorized by the current account to CSS to access or maintain data stored in OBS. | <p>The following conditions must be met for existing IAM agencies or those to be created:</p> <ul style="list-style-type: none"> <li>• Select <b>Cloud Service</b> for the <b>Agency Type</b>.</li> <li>• Select <b>Elasticsearch</b> for <b>Cloud Service</b>.</li> <li>• The agency has the <b>Tenant Administrator</b> permission for the <b>OBS</b> project in <b>Global service</b>.</li> </ul> |

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.

Backed up logs mainly include deprecation logs, run logs, index slow logs, and search slow logs. [Table 9-8](#) lists the storage types of the OBS bucket.

**Table 9-8** 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             |

## 9.12 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** page, view default plugins supported by the current version.

**Table 9-9** Cluster versions supported by default plugins

| Plugin                   | Supported Cluster Version             |
|--------------------------|---------------------------------------|
| analysis-dynamic-synonym | 5.5.1, 6.2.3, 6.5.4, 7.1.1, and 7.6.2 |
| analysis-icu             | 6.2.3, 6.5.4, 7.1.1, and 7.6.2        |
| analysis-ik              | 5.5.1, 6.2.3, 6.5.4, 7.1.1, and 7.6.2 |
| analysis-pinyin          | 5.5.1, 6.2.3, 6.5.4, 7.1.1, and 7.6.2 |
| analysis-poisson         | 6.2.3                                 |
| analysis-stconvert       | 5.5.1, 6.2.3, 6.5.4, 7.1.1, and 7.6.2 |
| lasthit                  | 5.5.1 and 6.2.3                       |
| repository-obs           | 5.5.1, 6.2.3, 6.5.4, 7.1.1, and 7.6.2 |
| vector-search            | 6.2.3                                 |
| opendistro_alerting      | 6.5.4 and 7.1.1                       |
| opendistro_security      | 6.5.4, 7.1.1, and 7.6.2               |
| opendistro_sql           | 6.5.4, 7.1.1, and 7.6.2               |
| analysis-kuromoji        | 6.5.4, 7.1.1, and 7.6.2               |
| analysis-nori            | 6.5.4, 7.1.1, and 7.6.2               |
| ingest-attachment        | 6.5.4, 7.1.1, and 7.6.2               |

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** 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 *Object Storage Service User Guide*. 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.

**Table 9-10** Viewing the plugin information

| Parameter     | Description  |
|---------------|--|
| ID            | Plugin ID, which is automatically generated  |
| Name          | Name of the plugin to be uploaded<br>The plugin name must be unique.   |
| Plugin Status | <p>Plugin status. The options are as follows:</p> <ul style="list-style-type: none"> <li>● <b>To be installed:</b> indicates the plugin has been uploaded but not installed.</li> <li>● <b>Installed and to be effective upon cluster restart:</b> If this status is displayed, restart the cluster for the plugin to take effect.</li> <li>● <b>Installed:</b> If this status is displayed after restarting the cluster, the plugin can work properly.</li> <li>● <b>Uninstalled and to be effective upon cluster restart:</b> If this status is displayed, restart the cluster for the plugin to take effect.</li> <li>● <b>Uninstalled:</b> If this status is displayed after restarting the cluster, the plugin is not available.</li> </ul> |
| Task Status   | <p>Running status of the plugin. The options are as follows:</p> <ul style="list-style-type: none"> <li>● <b>Uploading</b></li> <li>● <b>Installing</b></li> <li>● <b>Uninstalling</b></li> <li>● <b>Deleting</b></li> </ul>   |
| Uploaded      | Time when the plugin is uploaded   |
| Operation     | <ul style="list-style-type: none"> <li>● <b>Install:</b> After uploading the plugin, click <b>Install</b>. <b>Task Status</b> changes to <b>Installing</b>.</li> <li>● <b>Uninstall:</b> If you want to uninstall the plugin, click <b>Uninstall</b>. <b>Task Status</b> changes to <b>Uninstalling</b>. The uninstalled plugin is still displayed in the list. If you want to use it again, install it directly.</li> <li>● <b>Delete:</b> If you want to delete the plugin, click <b>Delete</b>. <b>Task Status</b> changes to <b>Deleting</b>. If you delete the plugin, you need to upload it before installing it again.</li> </ul>   |

 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 9-11** 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 <b>Upload</b> , <b>Install</b> , <b>Uninstall</b> , and <b>Delete</b> .   |
| Task Status    | Status of a task, including <b>Successful</b> , <b>Failed</b> , and <b>Running</b> .  |
| Created        | Time when a task is created   |
| Completed      | Time when a task is completed   |
| Error Message  | <p>Message reported when a task fails. Common error messages are as follows:</p> <ul style="list-style-type: none"> <li>• If the error message is <b>Instance: xx.xx.xx.xx. ERROR: `elasticsearch` directory is missing in the plugin zip.</b>, Check whether the plugin file is a standard .zip file.</li> <li>• If the error message is <b>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</b>, the plugin version does not match the cluster version.</li> <li>• If you still cannot locate the fault, contact technical support.</li> </ul> |

## 9.13 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
}
```

## 9.14 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 9-12** 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<br>Value: <b>true</b> and <b>false</b><br>Default value: <b>false</b>  |
|                             | 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.<br>Unit: s<br>Default value: <b>1,728,000</b>                      |
|                             | 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<br>Value: <b>true</b> and <b>false</b><br>Default value: <b>false</b>   |
|                             | 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<br>Value range: 1 to 100<br>Unit: %<br>Default value: <b>10%</b>   |
| Queue Size in a Thread Pool | thread_pool.bulk.queue_size | Queue size in the bulk thread pool. The value is an integer.<br>Default value: <b>200</b><br>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.<br>Default value: <b>200</b><br>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.<br>Default value: <b>1</b>  |
| Customize   | You can add parameters based on your needs. | Customized parameters<br><b>NOTE</b><br>Enter multiple values in the format as <b>[value1, value2, value3...]</b> .<br>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.

# 10 Monitoring a Cluster

---

## 10.1 Supported Metrics

### Function

This section describes CSS metrics that can be monitored by 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 10-1** 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<br><b>0:</b> All primary and replica shards are allocated. Your cluster is 100% operational.<br><b>1:</b> 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.<br><b>2:</b> 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<br>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<br>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<br>Unit: ms | $\geq 0$ 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                | $\geq 0$       | 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<br>Unit: ms   | $\geq 0$ ms    | CSS cluster                    | 1 minute                     |
| max_jvm_old_gc_count   | Max. JVM Old GC Count      | Maximum JVM Old GC count of nodes in a CSS cluster                  | $\geq 0$       | CSS cluster                    | 1 minute                     |
| total_fs_size          | Total Size of File Systems | Total size of file systems in a CSS cluster<br>Unit: byte           | $\geq 0$ 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<br>Unit: byte                           | ≥ 0 bytes   | CSS cluster                    | 1 minute                     |
| max_cpu_usage                          | Max. CPU Usage                          | Maximum CPU usage of nodes in a CSS cluster<br>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<br>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<br>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            | $\geq 0$    | CSS cluster                    | 1 minute                     |
| total_shards_count       | Shards             | Number of shards in a CSS cluster             | $\geq 0$    | CSS cluster                    | 1 minute                     |
| primary_shards_count     | Primary Shards     | Number of primary shards in a CSS cluster     | $\geq 0$    | CSS cluster                    | 1 minute                     |
| docs_count               | Documents          | Number of documents in a CSS cluster          | $\geq 0$    | CSS cluster                    | 1 minute                     |
| docs_deleted_count       | Deleted Documents  | Number of documents deleted in a CSS cluster  | $\geq 0$    | CSS cluster                    | 1 minute                     |
| nodes_count              | Nodes              | Number of nodes in a CSS cluster              | $\geq 0$    | CSS cluster                    | 1 minute                     |
| data_nodes_count         | Data Nodes         | Number of data nodes in a CSS cluster         | $\geq 0$    | CSS cluster                    | 1 minute                     |
| coordinating_nodes_count | Coordinating Nodes | Number of coordinating nodes in a CSS cluster | $\geq 0$    | CSS cluster                    | 1 minute                     |
| master_nodes_count       | Master Nodes       | Number of master nodes in a CSS cluster       | $\geq 0$    | CSS cluster                    | 1 minute                     |
| ingest_nodes_count       | Client Nodes       | Number of client nodes in a CSS cluster       | $\geq 0$    | CSS cluster                    | 1 minute                     |

## Dimensions

**Table 10-2** Dimension description

| Key        | Value       |
|------------|-------------|
| cluster_id | CSS cluster |

## 10.2 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.

**Table 10-3** 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    |
| Description | Alarm rule description. This parameter is optional.                                 | -             |

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

**Table 10-4** Parameters for configuring alarms

| 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. Currently, the following dimensions are supported: <ul style="list-style-type: none"> <li>• <b>CSS Clusters:</b> Alarm rules are specified by cluster.</li> <li>• <b>CSS Clusters - CSS Instances:</b> Alarm rules are specified by node in a cluster.</li> </ul>  | CSS Clusters         |
| Monitoring Scope | Resource range to which the alarm rule applies. You can select <b>Resource groups</b> or <b>Specific resources</b> .<br>Note: <ul style="list-style-type: none"> <li>• If <b>Resource groups</b> is selected and any resource in the group meets the alarm policy, an alarm is triggered.</li> <li>• If you select <b>Specific resources</b>, select one or more monitored objects and click  to synchronize the monitored object or objects to the dialog box on the right.</li> </ul> | Specific resources   |

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



**Table 10-5** Parameter description

| Parameter          | Description   | Example Value   |
|--------------------|---|-----------------|
| Method             | Select <b>Use template</b> or <b>Create manually</b> as required.<br>If you set <b>Monitoring Scope</b> to <b>Specific resources</b> , you can set <b>Method</b> to <b>Use template</b> . | Create manually |
| Template           | Select the template to be imported.   | -               |
| Alarm Notification | If this function is enabled, specify <b>Validity Period</b> , <b>Notification Object</b> , and <b>Trigger Condition</b> .   | -               |

- d. Click **Create**.

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

## 10.3 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.

# 11 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 11-1** 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 11-2** 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 11-3** Aggregations

| Aggregation  | Example                                      |
|--------------|--|
| avg()        | SELECT avg(age) FROM my-index                |
| count()<br>) | 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 11-4** Include and exclude fields

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

- Functions

**Table 11-5** 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 11-6** 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 11-7** 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](#).

# 12 Querying Cluster Logs

## 12.1 Key Operations Recorded by CTS

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 12-1** 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  |

## 12.2 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*.



# 13 FAQs

## 13.1 What Are Regions and AZs?

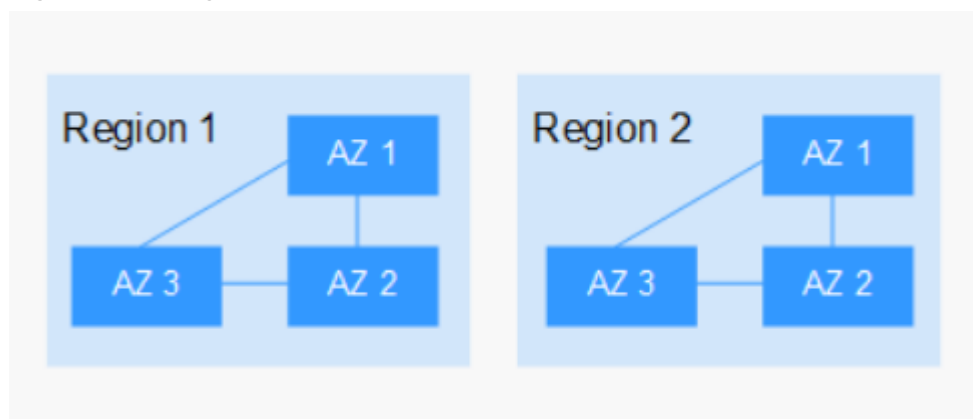
### Regions and AZs

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

- Regions are divided from the dimensions of geographical location and network latency. Public services, such as ECS, EVS, OBS, VPC, Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type only or for specific tenants.
- An AZ contains one or multiple physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow users to build cross-AZ high-availability systems.

**Figure 13-1** shows the relationship between regions and AZs.

**Figure 13-1** Regions and AZs



## Region Selection

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

## AZ Selection

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

- For high DR capability, deploy resources in different AZs in the same region.
- For low network latency, deploy resources in the same AZ.

## Regions and Endpoints

Before using an API to call resources, specify its region and endpoint. Obtain the regions and endpoints from the enterprise administrator.

# 13.2 How Does CSS Ensure Secure Running of Data and Services?

CSS ensures secure running of data and services from the following aspects:

- Network isolation  
The entire network is divided into two planes: service plane and management plane. The two planes are deployed and isolated physically to ensure the security of the service and management networks.
  - Service plane: refers to the network plane of the cluster. It provides service channels for users and delivers data definition, index, and search capabilities.
  - Management plane: refers to the management console. It is used to manage CSS.
- Host security  
CSS provides the following security measures:
  - The VPC security group ensures the security of hosts in a VPC.
  - Using the network access control list (ACL), you can permit or deny the network traffic entering and exiting the subnets.
  - Internal security infrastructure (including the network firewall, intrusion detection system, and protection system) can monitor all network traffic that enters or exits the VPC through the IPsec VPN.
- Data security  
Multiple replicas, cross-AZ deployment of clusters, and third-party (OBS) backup of index data ensure user data security.

# 13.3 Which CSS Metrics Should I Focus On?

The metrics that you need to focus on include the disk usage and cluster health status. You can log in to Cloud Eye and configure alarm prompts according to

actual conditions. If alarms are reported, clear them by taking related measures. For details about how to configure alarms, see *Creating Alarm Rules* in the *Cloud Search Service User Guide*.

**Configuration examples:**

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

**Measures:**

- Upon receiving alarms related to the disk usage, view disk space consumption, check whether data can be deleted from cluster nodes or archived to other systems to release space, or expand the disk capacity.
- If an alarm related to the cluster health status is received, check whether shard allocation is normal, whether shards are lost, and check whether the process is restarted on Cerebro.

## 13.4 Which Storage Options Does CSS Provide?

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

- Supported EVS disk types include common I/O, high I/O, and ultra-high I/O.
- For details about the sizes of EVS disks for different ECSs, see [Constraints](#).

## 13.5 What Is the Upper Limit for the Storage Capacity of CSS?

You can create 1 to 200 nodes during cluster creation. A certain number of disks are mounted to each node, which corresponds to an ECS. You can calculate the total storage capacity of CSS based on the sizes of EVS disks attached to different ECSs. For details about the limitation on EVS disk sizes, see [Constraints](#).

## 13.6 What Can Be the Disk Space of the Requested Cluster Used For?

The following logs and files can be stored in the disks:

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

## 13.7 Which Tools Can I Use to Manage CSS?

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

- **curl**  
cURL is a command-line tool used to perform URL operations and transfer information. It serves as the HTTP client that can send HTTP requests to the HTTP server and receive response messages. cURL is applicable to API debugging. For more information about cURL, visit <https://curl.haxx.se/>.
- **Code**  
You can call APIs through code to assemble, send, and process request messages.
- **REST client**  
Both Mozilla Firefox and Google Chrome provide a graphical browser plugin, that is, REST client, to send and process requests.
  - For Mozilla Firefox, see [Firefox REST Client](#).
  - For Google Chrome, see [Postman](#).

## 13.8 Which Elasticsearch Versions Does CSS Support?

In CSS, Elasticsearch 6.2.3, 6.5.4, 7.1.1, and 7.6.2 and Kibana 6.2.3, 6.5.4, 7.1.1, and 7.6.2 are supported.

## 13.9 Which Methods Can I Use to Access CSS?

You can access CSS using either of the following methods:

- RESTful API
- Transport Client

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

## 13.10 Does CSS Support APIs or Functions of Open-Source Elasticsearch?

Yes. Versions 6.2.3, 6.5.4, 7.1.1, and 7.6.2 are supported. The Elasticsearch Cloud mode is adopted. Kibana can interconnect with Elasticsearch, and functions of open-source Elasticsearch 6.2.3, 6.5.4, 7.1.1, and 7.6.2 are supported.

## 13.11 Can CSS Interconnect with Logstash?

Yes. Logstash 6.2.3, 6.5.4, 7.1.1, and 7.6.2 are recommended. You need to apply for an ECS to install and configure Logstash.

## 13.12 What Should I Do If an ECS Cannot Connect to a Cluster?

Perform the following steps:

1. Check whether the ECS instance and cluster are in the same VPC.
  - If yes, go to [2](#).
  - If not, create an ECS instance and ensure that the ECS instance is in the same VPC as the cluster.
2. View the security group rule setting of the cluster to check whether port **9200** (TCP protocol) is allowed or port **9200** is included in the port range allowed in both the outbound and inbound directions.
  - If yes, go to [3](#).
  - If not, switch to the VPC management console and set the security group rule of the cluster to allow port **9200** in both the outbound and inbound directions.
3. Check whether the ECS instance is added to a security group.
  - If added to a security group, check whether the security group configuration rules meet the requirements. For details, see the description of **Security Group** in the cluster information table in **Clusters**. Then go to [4](#).
  - If not, go to the VPC page from the ECS instance details page, select a security group, and add the security group.
4. Check whether the ECS instance can connect to the cluster.  
*ssh <Private network address and port number of a node>*

### NOTE

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

- If the connection is normal, the network is running properly.
- If the port is unreachable, contact the administrator.

## 13.13 Which Search Functions Does CSS Support?

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

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

## 13.14 Why Do I Fail to Create a Cluster?

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

- Insufficient resource quota. You are advised to apply for sufficient resource quotas.

- The value of **Port Range/ICMP Type** in **Security Group** does not include port **9200**. Modify the security group information or select another available security group.

## 13.15 Why Cannot I Perform Index Backup?

Index backup is implemented by creating cluster snapshots. If you cannot perform index backup, perform the following steps:

### Check Whether the Account or IAM User Has the Index Backup Permission

1. Log in to the IAM management console.
2. Check the user group, to which the account or the IAM user belongs.  
For details, see "Viewing and Modifying User Information" in the *Identity and Access Management User Guide*.
3. Check whether the permissions assigned to the user group contain the following: **Tenant Administrator** for project **OBS** in region **Global service** and **CSS Administrator** in the current region.  
For details, see "Viewing and Modifying User Group Information" in the *Identity and Access Management User Guide*.
  - If neither of the preceding permissions is contained, go to [4](#).
  - If both of the preceding permissions are contained, contact the administrator.
4. Add the following permissions to the user group: **Tenant Administrator** for project **OBS** in region **Global service** and **CSS Administrator** in the current region.  
For details, see "Viewing and Modifying User Group Information" in the *Identity and Access Management User Guide*.

## 13.16 Filebeat Configuration Optimization

### Symptom

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

### Fault Locating

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

### Procedure

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

# Increase the value of **harvester\_buffer\_size** based on actual requirements. This parameter defines the buffer size used by every **harvester**.

```
harvester_buffer_size: 40,960,000
```

# Increase the value of **filebeat.spool\_size** based on actual requirements. This parameter defines the number of log records that can be uploaded by the **spooler** at a time.

```
filebeat.spool_size: 250,000
```

# Adjust the value of **filebeat.idle\_timeout** according to the actual requirements. This parameter defines how often the **spooler** is flushed. After the **idle\_timeout** is reached, the **spooler** is flushed regardless of whether the **spool\_size** has been reached.

```
filebeat.idle_timeout: 1s
```

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

# Set the value of **worker** to the number of Elasticsearch clusters according to the actual situation. The **worker** parameter indicates the number of Elasticsearch clusters. The default value is **1**.

```
worker: 1
```

# Increase the value of **bulk\_max\_size** based on the actual requirements. This parameter defines the maximum number of events to bulk in a single Elasticsearch bulk API index request. The default is **50**.

```
bulk_max_size: 15,000
```

# Adjust the value of **flush\_interval** based on the actual requirements. This parameter defines the number of seconds to wait for new events between two bulk API index requests. If **bulk\_max\_size** is reached before this interval expires, additional bulk index requests are made.

```
flush_interval: 1s
```

## 13.17 What Should I Do If the Access to Kibana Fails?

### Symptom

After I click **Kibana** in the **Operation** column in the row where cluster **Es-event** resides on the **Clusters** page of the CSS management console, the Kibana page fails to be loaded and access to Kibana fails.

### Fault Locating

The browser cache is not cleared.

### Procedure

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 cluster **Es-event** resides and click **Kibana** in the **Operation** column.
4. On the displayed **Kibana** page, press **F12**.

5. Click **Network**, right-click **data:image**, and choose **clear browser cache** from the shortcut menu. In the displayed dialog box, click **OK**. Close the Kibana window.
6. Switch to the **Clusters** page, locate the row where cluster **Es-event** resides and click **Kibana** in the **Operation** column.



---

# A Change History

---

| Released On | Description                         |
|-------------|-------------------------------------|
| 2020-11-07  | This is the first official release. |